

Security of Cyber Physical Systems: Process-Aware Approach

**Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades
Doktor-Ingenieurin (Dr.-Ing.)**

genehmigte Dissertation

von
Marina Krotofil

aus
Poltawa, Ukraine

2023

Tag der mündlichen Prüfung | 10. Februar 2023

Vorsitzender des Prüfungsausschusses	Prof. Dr. habil. Sibylle Fröschle Institut für Secure Cyber-Physical Systems Technische Universität Hamburg
---	---

1. Gutachter	Prof. Dr.-Ing. Dieter Gollmann Technische Universität Hamburg
--------------	--

2. Gutachter	Prof. Dr.-Ing. Felix Freiling Lehrstuhl für IT-Sicherheitsinfrastrukturen Friedrich-Alexander-Universität Erlangen- Nürnberg
--------------	---

DOI: <https://doi.org/10.15480/882.4913>

ORCID: <https://orcid.org/0000-0001-9756-2891>

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Das Werk steht unter der Creative-Commons-Lizenz Namensnennung 4.0 International (CC BY 4.0, <https://creativecommons.org/licenses/by/4.0/legalcode.de>). Ausgenommen von der oben genannten Lizenz sind Teile, Abbildungen und sonstiges Drittmaterial, wenn anders gekennzeichnet.

Acknowledgment

It takes a village to raise a child.
It takes a world to raise a PhD.

My PhD road was nothing but incredible and memorable. I would not want any other! During my research times, I was able to make a lot of unique experiences and get exposed to the brightest minds and mentors globally. I took a conscious decision to build my professional security career by pursuing a doctoral degree under Prof. Dieter Gollmann's supervision. One aspect to be mentioned about his supervising style is non-interference. Best professor ever. He encouraged me to find and follow my own path. It was hard at the beginning, but it became the best experience of my life. I would like to express my gratitude to some of the key individuals who contributed to the existence of this thesis.

Prof. Dieter Gollmann. This thesis, as submitted, would absolutely not be possible without your patience, support and supervision. None of the PhDs stunts is easy, neither was mine. But you never doubted my abilities and encouraged me beyond my own beliefs in myself. You also let me experience industrial and research opportunities outside of our university department. Among everything, I am grateful for the chance to learn from you through observation as I find you to be one of the most incredible security professionals globally.

Prof. Alvaro Cardenas. I perfectly remember the moment when I got inspired by the highest quality of your academic work. Being motivated, I walked the extra mile to be accepted into your research and mentorship circle. I treasure our collaboration very much. You are a big part of this thesis and I am endlessly grateful to you for your guidance.

Prof. Felix Freiling. I am grateful to you for agreeing to be my second supervisor, for the consultations and for the very helpful advice at the early stage of my PhD career. I carefully stored the paper you gave me during one of our first meetings. It served an incredible service in my research two years later and contributed to the chapter in this dissertation.

I am further grateful to other people in somewhat chronological but no particular order.

Dr. Ragnar Shierholz. You are the person who introduced me to the world of Industrial Control Systems and their cyber security challenges. At a time when this research area was still in its infancy and not fashionable, inspired by your professional passion and technical expertise I developed a profound interest in industrial process environments and cyber-physical systems. I am sincerely thankful for your continuous knowledge support and our developed friendship.

Eireann Leverett. I will never forget the day we met. Thank you for introducing me to the "Breakage" world and helping me to unlock an entirely new level of the cyber-physical research world. I am endlessly grateful to you for your continuous encouragement, inspiration (you are so creative in your research!), friendship and support.

Jason Larsen. I can't even begin to describe how grateful I am for your leadership in the field of Industrial Control Systems security and for our collaboration. I know you had only very few spare hours (sometimes minutes!) in your life and you generously supported my research passion with them. My PhD life would not be the same without our technical discussions and without your friendly support when my control loops did not behave well.

Alexander Isakov. Thank you so much for your exceptional analytical thinking, meticulous software development skills and your willingness to walk the extra mile. I am grateful to you for your dedication to my research and for the 150%-quality results I asked for. I will always cherish the memories of our late-night work in the lab and many special moments of mutual support.

Dr. Pawel Gurikov. Thank you for believing that hackers are real and supporting my non-conventional work on exploiting chemical processes. A whole chapter in this thesis reached the required scientific credibility thanks to consultations with you.

My helpful students. I am very thankful to all of my numerous students, who picked me as their supervisor and did incredible research work during my research career. I would like to express huge thanks to **Moritz Obermeier** for turning our first hardware lab experience into a fun adventure and to **Bradley Manning** for our first work on process models when we still did not even know where the journey was taking us. As the lab evolved, more great students joined the crew. A very special and sincere appreciation goes to my student **Alexander Winnicki** who became my valuable long-time collaborator and with whom we published at the most prestigious ICS security conferences.

There are so many other people to thank and opportunities to appreciate. My PhD time would not be as memorable without yearly PhD candidates gatherings organized by Prof. Dieter Gollmann, Prof. Felix Freiling, Prof. Joachim Posegga, Prof. Jörg Schwenk and Prof. Thorsten Holz. The friendships developed during these meetings last till today. Special thanks go to Alexander Galkin, Damiano Bolzoni, Dina Hadziosmanovitch, Mona Lange, Benjamin Green and Ali Abbasi for their collaboration, support and good times. Additionally, I am much obliged to Manuel Atug, Cornelius Ascherman and my dear friend Joe Neesan for their unmatched encouragement to persist and follow my path. I am grateful for research and work opportunities to ABB, Shell and Compass Security and the University of Texas at Dallas. Last but not least I am incredibly grateful to early influencers who inspired me and invited me into their professional circles: Ruben Santamatra, Dale Peterson and Ralf Langner, Joel Langill, Jack Brodsky and Bob Radvanovsky, Chris Blask, Nate Kube and many others.

I am, of course, sincerely grateful to all my former teachers and my family.

My teachers and lecturers. I am in thankful to my Ukrainian school teachers and university lecturers for shaping me into a human being I became. Cyber-Physical Systems research is a multidisciplinary and technically challenging research field. At times I felt being pushed to the limits of my intellectual abilities. Thanks to the education you gave me, I was always able to solve difficult problems and find answers. When I wanted to give up, I remembered all the efforts you put into my formal training and kept pushing.

My parents and family. And of course, I am primary in debt to my loving parents and to my large loving family for always supporting me and raising me as an individual who believes that everything is possible.

Abstract

Cyber-physical systems (CPS) consist of IT or cyber systems “embedded” in applications in the physical world. They combine sensors, actuators, control units, operator consoles, and communication networks. Some of these systems are critical because they are part of an infrastructure critical for society. Critical infrastructure protection had become a high-profile topic in the last decade, partly because the use of the Internet and commodity hardware and software has exposed the industry to security threats from which it has historically been isolated, and partly because the potential impact of attacks on critical infrastructures can no longer be ignored.

Cyber attacks on physical systems are called cyber-physical attacks. The implications of this class of attacks, namely the ability to inflict physical damage, are the main difference that puts them apart from conventional cyber attacks. In this thesis, we show that the security concepts from the IT domain are not sufficient to describe the security needs of cyber-physical systems nor are the defenses effective against cyber-physical attacks. We, therefore, propose to treat the cyber part of a cyber-physical system as a control system and focus on the interfaces between physical space and cyberspace from the attacker’s perspective. We argue that in the CPS domain Defense-in-Depth has to be extended to the level of the physical process itself.

This thesis consists of four parts.

In the first part we address the challenge of lacking realistic large-scale test beds for studying cyber attacks and their effects on physical processes. We present two models of continuous processes, Tennessee Eastman and Vinyl Acetate Monomer, which we enhanced and instrumented with capabilities to conduct cyber-physical security research (Chapter 3).

In the second part, using an example of DoS attacks on sensor and actuator signals, we demonstrate that canonical IT security measures for communication infrastructure are insufficient to defend against attacks that take advantage of control system properties (Chapter 4).

In the third part we introduce a data security property called veracity or trustworthiness. We illustrate that data veracity is not a property guaranteed by any of the familiar IT infrastructure security services and propose a physical process-aware method to detect and locate non-veracious process data in near-real time (Chapter 5).

Lastly, based on the knowledge gained, we propose a model of the cyber-physical attack life cycle and illustrate its applicability on the example of designing persistent economic damage scenario in a Vinyl Acetate Monomer chemical plant (Chapter 6).

We conclude the thesis with the encouragement to further develop process-aware cyber security mechanisms that do not solely rely on IT security approaches. Directions for future research are outlined in the conclusions.

Contents

Contents	vii
List of Figures	xi
List of Tables	xv
Acronyms	xvii
1 Introduction	1
1.1 Cyber-Physical Systems	1
1.2 Cyber-Physical Systems Security	2
1.3 Motivation	2
1.3.1 Research Questions	3
1.3.2 Thesis Contributions and Overview	4
1.3.3 About the Title of this Thesis	6
2 Preliminaries: Process Control Systems	7
2.1 Fundamentals of Process Control	8
2.2 Process Control Systems Architecture	11
2.3 Cyber-Physical Attacks	24
2.3.1 Timing Parameter	25
2.3.2 Safety vs. Security	27
2.4 Industrial Control System Threat Landscape	29
2.5 Conclusion	34
3 Damn Vulnerable Chemical Process Framework	35
3.1 Introduction	35
3.2 Framework Description	36
3.2.1 Tennessee Eastman Process	37
3.2.2 Vinyl Acetate Monomer Process	43
3.2.3 Comparison of TE and VAC Processes	46
3.3 Potential Applications of the Framework	48
3.4 Discussion	56
3.5 Related Work	58
3.6 Conclusions	59
4 Vulnerability of Process Control Systems to Stale Data	61
4.1 Introduction	61
4.2 Susceptibility of Control Equipment to Stale Data	64

4.3	Stale Data Attack	65
4.3.1	Example of Stale Data Impact	66
4.3.2	Selecting Time to Attack	67
4.4	Best Choice and Optimal Stopping Problem	69
4.4.1	Best Choice Problem	69
4.4.2	Peak Detection in a Correlated Time Series	70
4.4.3	Heuristic Approach	72
4.5	Experimental Results	73
4.5.1	Shortest Shutdown Time	73
4.5.2	Attacks on Controller Signals	74
4.5.3	Attacks on Sensor Signals	77
4.5.4	Additional Considerations	81
4.6	Discussion	84
4.7	Related Work	86
4.8	Conclusions	88
5	Data Veracity in Process Control Systems	89
5.1	Introduction	89
5.2	Data Veracity	90
5.2.1	Violation of Data Veracity	91
5.2.2	Ensuring Data Veracity	93
5.3	Sensor Signal Spoofing	94
5.3.1	Sensor Signal Noise Reconstruction with Runs Analysis	95
5.3.2	Sensor Signal Dynamics Reconstruction with Triangle Approximation	96
5.3.3	Construction of Spoofed Sensor Signals	98
5.4	Information-Theoretic Detection Approach	99
5.4.1	Data Preparation	99
5.4.2	Plant-Wide and Sensor Entropy	100
5.4.3	Cluster Entropy	101
5.5	Experimental Results	103
5.5.1	Sensors Clustering	103
5.5.2	Detection of Spoofed Signals	105
5.5.3	Use Case: Powerful Attacker	107
5.5.4	Detection Complexities	109
5.5.5	Increasing Detection Robustness	110
5.6	Discussion	111
5.7	Related work	112
5.8	Conclusions	113
6	Cyber-Physical Attack Lifecycle	115
6.1	Introduction	115
6.2	Classes of Cyber-Physical Attacks	116
6.2.1	Equipment damage	116
6.2.2	Production damage	117
6.2.3	Compliance violation	117
6.3	Cyber-Physical Attack Lifecycle	118
6.3.1	Access	119
6.3.2	Discovery	121

6.3.3	Control	123
6.3.4	Damage	126
6.3.5	Cleanup	131
6.4	Targeted Attack for Production Damage	133
6.5	Achieving Attack Objectives	134
6.5.1	Access	134
6.5.2	Discovery	134
6.5.3	Control	136
6.5.4	Damage	139
6.5.5	Cleanup	143
6.6	Discussion	146
6.7	Related Work	147
6.8	Conclusions	148
7	Conclusions and Future Research	151
7.1	Summary of Contributions	151
7.2	Follow-Up Research	153
7.3	Future Research	155
7.4	Conclusions	157
	Appendix	159
A	Attack on Situational Awareness	159
A.1	Experimental Setup	159
A.2	Experimental Results	161
B	ICS Security Properties	163
B.1	Controllability, Observability and Operability	163
B.2	Ensuring ICS Security	165
C	Data Utility in Process Control Systems	167
D	Sensor Signal Spoofing Algorithms	171
D.1	Runs Algorithm	171
D.2	Triangle algorithm	173
E	Communication Channels Beyond IT	175
E.1	Zones and Conduits	175
E.2	Delivering Attack Payloads via Process Physics	176
E.3	Information Flow via Process Physics	177
	Bibliography	179

List of Figures

1.1	Logical layers of a cyber-physical system	3
2.1	Distinction between various types of control systems	7
2.2	The components of a control loop	8
2.3	Multivariable control loop	9
2.4	PID controller and impact of a derivative component on controller output	9
2.5	Benefits of Advanced Process Control [293]	10
2.6	Purdue reference architecture for process automation	12
2.7	Physical process infrastructure and field equipment [155]	12
2.8	Sensor signal and PLC scan cycle	14
2.9	Example of a tag table in a PLC [3]	15
2.10	Example of Human Machine Interface view [90]	17
2.11	Alarm criticality and consequences cost (source is not specified for sensitivity reasons)	17
2.12	Alarm system generation and response workflow [62]	18
2.13	Data historian view [185]	19
2.14	Open NAMUR reference architecture for IIoT applications [26]	22
2.15	Automation pyramid according to ISA-95	23
2.16	OPC communication topology	23
2.17	Multidisciplinary components of cyber-physical attack	24
2.18	Time constants in process control, based on [288]	25
2.19	Water hammer	26
2.20	Dependability and security attributes, based on [21]	27
2.21	Relationships between security and safety	28
2.22	Historical time of cyber-physical attacks	29
2.23	Architecture of a model 3008 main processor [60]	31
2.24	“Race-to-the-Bottom” exploitation trend in Industrial Control Systems	32
2.25	Industroyer payload components	33
3.1	Plantwide chemical process under control–based on [193]	38
3.2	Different types of sensor signals in the TE process	40
3.3	TE Simulink model enhanced with attack blocks	41
3.4	Functional implementation of integrity attack on sensor signal in TE Simulink model	42
3.5	Economic impact of the attack on purge valve	43
3.6	Vinyl Acetate Monomer plantwide process control structure	44
3.7	Visualization of simulation results	47
3.8	TE process enclave network diagram [40]	49
3.9	Calculation of the response time for the operator, based on [240]	52
3.10	Concealment of attack trace (oscillation) by changing signal sampling frequency	55

3.11	Animation of explosion incident at the DeRidder pulp and paper mill, USA [268] .	57
4.1	Stale data attack on sensor signal	63
4.2	DoS attack on reactor sensor pressure	63
4.3	Frequency deviation due to “frozen” measurement [299]	67
4.4	Different effects of DoS attack on a Reactor Pressure sensor at random times . . .	68
4.5	Example of a sensor signal	68
4.6	Classic Secretary Problem [79]	70
4.7	Illustration of the Secretary Problem solution applied to a sensor signal	71
4.8	Illustration of Outlier Test detection	73
4.9	“Closing” DoS attack on the C -feed valve	76
4.10	Four types of sensor signals in TE process: raw and smoothed ($\mu = 250$) signals .	80
4.11	Distribution of the error in selecting the highest possible sample, without and with peak detection	81
4.12	Impact of the IDV(8) disturbance on process measurement	81
4.13	Impact of the sampling frequency on the sensor signal	82
4.14	Impact of the sampling frequency on the SDT	83
4.15	Detection of the plant state change, $h = 0.005$	83
4.16	Examples of the possible stages for chaining attacks	84
4.17	Data packet with process variables represented in unknown units	84
4.18	Impact of the sampling frequency on signal smoothing	85
4.19	Relationship between process performance and plant profit [184]	86
4.20	Usage of wireless 5G communication in an industrial plant [285]	86
5.1	The Parkerian hexad	90
5.2	Data quality in process control systems	90
5.3	Threat scenario for analog control loop [30]	93
5.4	Attributes of Runs Analysis	96
5.5	Reconstruction of sensor signal noise with the Runs algorithm: learning and synthesis phases	97
5.6	Extraction of the dynamic sensor signal trend with line segment approximation .	97
5.7	Illustration of a sensor signal approximation with a strip of triangles	98
5.8	Steps of the sensor signal spoofing process	99
5.9	Steps of data discretization procedure	100
5.10	Computation of plant-wide and sensor entropy	101
5.11	Detection of the process state change with plant-wide and sensor entropy	102
5.12	Illustration of the approach to computing cluster entropy	103
5.13	Pairwise sensor correlation coefficients	104
5.14	Correlated sensor signals forming a cluster	105
5.15	Detection of the data integrity attack on XMV(10) with PWE and SE	106
5.16	Correlation entropy calculated over non-overlapping time windows	106
5.17	Computation of correlation entropy with sliding time window	107
5.18	Detection of spoofed signal presence in a cluster, with a plant disturbance and in a steady state	108
5.19	Detection of spoofed signals presence in a cluster when all signals are spoofed . .	108
5.20	Detection of spoofed sensor signals in a steady state	109
5.21	Correlation inconsistency	110
5.22	Inaccurate spoofing of a signal by learning from a signal of a different scaling . .	111
5.23	Identification of related/correlated signals	111

5.24	Chemical analyzer signal	112
6.1	Mandiant Attack Lifecycle [213]	115
6.2	Stages of SCADA attacks [188]	118
6.3	Proposed cyber-physical attack lifecycle	119
6.4	Attack path to safety systems in Triton attack [159]	120
6.5	Results from the S7comm asset discovery module from the ICSSPLOIT toolkit [77]	122
6.6	Instrumentation and Control System engineering software [20]	123
6.7	Nonlinear response of a control loop	124
6.8	Parameters which influence control loop response	125
6.9	Visualization of the plant-wide process response to a manipulation of the valve [320]	126
6.10	Tunable parameters of a VFD [30]	127
6.11	P&ID of the ultrafiltration section of a water treatment plant [157]	128
6.12	Differential pressure on a membrane in UF water filter	129
6.13	Hierarchical architecture of control loops	129
6.14	Architecture of C2 and feedback channels in cyber-physical attack life cycle	130
6.15	Indicator of an unsuccessful implosion attack [191]	131
6.16	Socio-technical system	132
6.17	Simplified scheme of the VAC plant	134
6.18	Location of control valves in the reactor unit of the VAC plant	135
6.19	Illustration of process response across the entire plant in VAC process	137
6.20	Types of attacks on process behavior (red line denotes steady state value)	138
6.21	Manipulation of control loop 5 (vaporizer)	138
6.22	Location of chemical analyzers in VAC plant	140
6.23	Comparison of indirect and direct measurements	141
6.24	Code to calculate the concentration change of chemical components in the reactor	141
6.25	Vinyl acetate production, 24 hours	142
6.26	Increasing variation of reactor exit temperature caused by attacks on different control loops	144
6.27	Analysis of the physical processes in the reactor	145
6.28	Relationship between TSD and cyber-physical attack lifecycle	148
7.1	Snippet of a control logic for a traffic light system [178]	155
7.2	Construction of equipment failure indicators [251]	156
7.3	Visualization of defense-in-depth concept in cyber-physical systems	157
A.1	Testbed and IDS architecture	160
A.2	Modbus client on the PLC	160
A.3	Data inconsistency detection	161
B.1	Security properties for control systems	164
C.1	Data flow of a single sensor through the levels of ICS reference architecture [123]	168
C.2	Flow-induced vibration in piping systems and its detection	169
C.3	Inconsistency of control equipment configuration	170
E.1	Delivery of attack payload via process physics and violation of the network security zoning architecture	176
E.2	Pump monitoring application by Flowserve [177]	177
E.3	Propagation of alarms and deteriorated process state	178

List of Tables

2.1	Impact of alarm rate on operator acceptability and performance, per 10 min in steady state, based on [128]	18
3.1	Valves and measured variables	39
3.2	Process operating constraints [87]	39
3.3	Complexity comparison of the processes	47
3.4	Comparison of process models features; * enabled by us	48
4.1	Simulation results of the attacks on MV	75
4.2	Simulation results of the approaches to the Best Choice Problem solution for the attacks on PV	78
4.3	Aggregated comparison between the approaches	79
4.4	Comparison between the approaches, steady state	80
4.5	Comparison between the approaches, $F_s=100$ s/h	82
5.1	Illustration of the discretization process	100
6.1	Relative paracetamol prices [286]	117
6.2	Classes of cyber-physical attacks and their major impacts	118
6.3	Sensitivity of control loops	139
6.4	Activation of alarms	139
6.5	Categorization of control loops based on damage potential	143

Acronyms

ACE	Area Control Error
ABS	Antilock Braking System
ADC	Analog-to-Digital Converter
AI	Analog Input
AIChE	American Institute of Chemical Engineers
APC	Advanced Process Control
ASIC	Application-Specific Integrated Circuits
ATT&CK	Adversarial Tactics, Techniques & Common Knowledge
BPCS	Basic Process Control System
C2	Command & Control
CBD	Cloud-Based Design
CBM	Cloud-Based Manufacturing
CE	Continental Europe
CI	Critical Infrastructure
CIA	Confidentiality, Integrity, Availability
CIP	Critical Infrastructure Protection
CISA	Cybersecurity and Infrastructure Security Agency
CO	Controller Output
COO	Controllability, Observability, Operability
CPS	Cyber-Physical System
CPU	Central Processing Unit
CSB	Chemical Safety Board
CUSUM	Cumulative Sum
DC	Direct Current
DBOM	Data Bill of Material
DCS	Distributed Control System
DEFRA	Department for Environment, Food and Rural Affairs
DM	Decision Maker
DMZ	Demilitarized Zone

DNP3	Distributed Network Protocol
DoS	Denial of Service
DP	Differential Pressure
DVCP	Damn Vulnerable Chemical Process
DVL	Damn Vulnerable Linux
DVWA	Damn Vulnerable Web Application
EFF	Reactor Efficiency
EMI	Electromagnetic Interference
ERP	Enterprise Resource Planning
ESP	Electronic Security Perimeter
EWS	Engineering Workstation
FCE	Final Control Element
FEHE	Feed-Effluent Heat Exchanger
FMEA	Failure Mode and Effect Analysis
FT	Flow Transmitter
GUI	Graphical User Interface
HART	Highway Addressable Remote Transducer
HAZOP	Hazard and Operability Study
HMI	Human Machine Interface
IaaS	Infrastructure as a Service
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
ICS	Industrial Control System
ICSS	Integrated Control and Safety System
ID	Identifier
IDS	Intrusion Detection System
IIoT	Industrial Internet-of-Things
I/O	Input/Output
IoT	Internet-of-Things
IP	Internet Protocol
ISA	International Society for Automation
ISO	International Organization for Standardization
IT	Information Technology
LFC	Load Frequency Controller
LOPA	Layers of Protection Analysis

MES	Manufacturing Execution System
MITM	Man-in-the-Middle
MM	Magnitude of Manipulation
MOMS	Manufacturing Operations Management Systems
MPC	Model Predictive Control
M&TE	Maintenance and Test Equipment
MV	Manipulated Variable
NAMUR	Normenarbeitsgemeinschaft für Mess- und Regeltechnik
NCUSUM	Non-parametric CUSUM
NERC	North American Electric Reliability Corporation
NIST	National Institute of Standards and Technology
NOA	NAMUR Open Architecture
NS	Non-Selection
OLE	Object Linking and Embedding
OPC	OLE for Process Control
OPC UA	OPC Unified Architecture
OS	Operating System
OSINT	Open Source Intelligence
OT	Operational Technology
OT	Outlier Test (Chapter 4)
PA	Periodic Attack
PCS	Process Control System
PERA	Purdue Enterprise Reference Architecture
PHA	Process Hazards Analysis
PID	Proportional, Integral, Derivative
P&ID	Piping & Instrumentation Diagram
PIMS	Plant Information Management System
PLC	Programmable Logic Controller
PoC	Proof of Concept
PV	Process Variable
PWE	Plant-Wide Entropy
QRA	Qualitative Risk Analysis
RAT	Remote Access Trojan
R&D	Research and Development
RDP	Remote Desktop Protocol
RT	Recovery Time

RO	Reverse Osmosis
RTD	Resistance Temperature Detector
RTDB	Real Time Database
RTOS	Real-Time Operating System
RTU	Remote Terminal Unit
SaaS	Software as a Service
SBOM	Software Bill of Material
SCADA	Supervisory Control and Data Acquisition
SDT	Shutdown Time
SE	Sensor Entropy
SIL	Safety Integrity Level
SIS	Safety Instrumented System
SISO	Single-Input, Single-Output
SoC	System on a Chip
SOC	Security Operations Center
SP	Setpoint
SP	Secretary Problem (Chapter 4)
SQL	Structured Query Language
SSA	Steady state attack
STS	Socio-Technical System
SWaT	Secure Water Treatment
TCP	Transmission Control Protocol
TE	Tennessee Eastman
TSD	Timing and State Diagrams
TT	Temperature Transmitter
TTA	Time to Attack
TTP	Tactics, Techniques and Procedures
TTS	Time to Select
TW	Time Window
TWE	Time-Window Entropy
VAC	Vinyl Acetate
VAM	Vinyl Acetate Monomer
VFD	Variable Frequency Drive
VT	Variable Table
UDP	User Datagram Protocol
UF	Ultrafiltration
UV	Ultraviolet

Chapter 1

Introduction

No problem can be solved from the same level of consciousness that created it.

Albert Einstein

1.1 Cyber-Physical Systems

Advances in computing and networking have added new capabilities to physical systems that could not be feasibly added before. This has led to the emergence of engineered systems called cyber-physical systems (CPS): systems where the events in the physical world are managed with the help of modern advances in computation and control. Complex machines such as aircraft or robots, building automation systems, smart cities and smart grids, railways and agricultural systems, medical devices and industrial infrastructures, in general, are examples of cyber-physical systems. Some of these industrial processes are categorized as critical infrastructures (CI) because societal well-being depends on their reliability (e.g., water and power utilities). On the other side of the spectrum of the CPS definition are smaller appliances and gadgets such as consumer electronics and wearables widely known as the Internet of Things (IoT) devices. Instead of being an inseverable part of a larger cyber-physical ecosystem, they are stand-alone devices that are programmed for certain clearly defined applications and rely on Internet connectivity for continuous transmission of data for analysis and feedback. In our thesis, we consider larger cyber-physical systems such as industrial infrastructures due to their higher complexity.

Cyber-physical systems were mapped as a research area in a series of National Science Foundation (USA) workshops starting in 2006. This is also when the term *cyber-physical systems* first emerged. In CPS physical processes affect computations and vice versa with computations and physical processes being so tightly integrated that it is not possible to identify whether behavioral attributes are the result of computations, physical laws, or both working together. In the words of Edward A. Lee, one of the pioneers of the cyber-physical systems discipline [196]:

As an intellectual challenge, CPS is about the intersection, not the union, of the physical and the cyber. It combines engineering models and methods from mechanical, environmental, civil, electrical, biomedical, chemical, aeronautical and industrial engineering with the models and methods of computer science.

As stated in [70], CPS design is subject to a wide range of physical requirements, such as dynamics, power and physical size as well as to systems-level requirements, such as safety, security and fault tolerance. While security is part of the requirements, it is deeply embedded into

engineering specifications. Security is not seen as “cybersecurity”, a property that concerns the confidentiality, integrity and availability of information systems and has no intrinsic connection with physical processes, but as a novel discipline of cyber-physical security – an integration of physical/engineering and information sciences.

1.2 Cyber-Physical Systems Security

Perceived and real security threats affecting cyber-physical systems have been attracting considerable attention in the media, among decision makers, regulators, and in the research community. Cyber-physical systems, by their very nature, cause effects in the physical world, which in some cases can have disastrous physical consequences. On one hand, this is an issue that had to be dealt with already before physical systems were connected to cyberspace. Well-designed systems would have been deployed with appropriate safety measures in place. Conceivably, those measures can restrain cyber-physical attacks once they have transited from cyberspace into the physical domain. On the other hand, those countermeasures were designed under certain assumptions, e.g., physical security protecting access to premises or independence of component failures. Conceivably, those assumptions get invalidated once modern Information Technology (IT) systems get integrated with existing physical plants.

The concern for physical consequences puts cyber-physical systems security apart from the science of *information security*. Integrating modern IT systems with existing physical systems exposes these installations to new security threats. Some of these threats are well-known in IT security and countermeasures have been studied at length. Those threats are new only because of a new application area. Other threats may indeed be specific to cyber-physical systems. This thesis aims at making a distinction between “old” security issues in new settings, and new security aspects intrinsic to cyber-physical systems that would establish cyber-physical security as an object of study in its own rights.

1.3 Motivation

The primary focus of academia and industry has been on securing the communication infrastructure and hardening control systems. There is a large body of literature on how to adapt existing IT security methods to the characteristic features of the control domain. However modern malware for persistent attacks may now be equipped with “trusted” certificates, travel in USB sticks and laptops of “trusted” parties, carry zero-days exploits and rootkits, and propagate through “trusted” security updates. It is becoming increasingly difficult to prevent, detect and to halt these attacks based solely on technical security measures deployed in the cyber layer.

It is often claimed that “once communications security is compromised the attacker can do whatever she wants”. These are presumptuous claims. The attacker may well be able to inject any input she wants but this does not necessarily amount to being able to influence processes in the physical world at will. The processes and their actuators have to be properly understood. Process physics and built-in safety measures might get in the way of the attacker. To address the limitations of defending a system using only IT methods, a new line of research has focused on understanding the adversary’s interactions with the physical system. Analyzing the effects of attacks in the process control domain is a growing area of research.

Figure 1.1 shows the logical layers of a cyber-physical system. While compromising the cyber layer is essential, the attacker needs to interact with the control system to achieve the desired

outcome in the physical world. Studying such interactions from the attacker perspective and understanding needed defenses is the focus of this thesis.

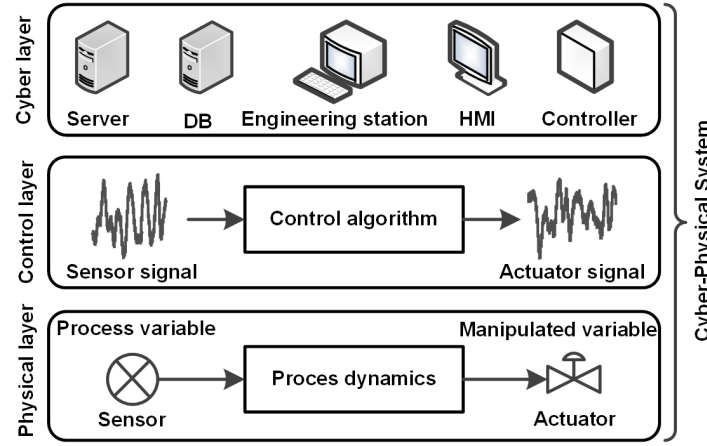


Figure 1.1: Logical layers of a cyber-physical system

1.3.1 Research Questions

To keep our focus of this thesis on the topic of CPS-specific security issues, we assume an attacker capable of obtaining the required level of access to the control system environment. Industrial equipment and environments are commonly characterized by a weak security posture, which makes our assumption reasonable. We will focus on the aspect of attacker interaction with the control system and physical process. However, to keep our research pertinent and real, we consider relevant aspects of the supporting cyber infrastructure whenever necessary and appropriate. One of the key challenges in conducting this line of research is the lack of accessible cyber-physical environments for experimentation and testing. Therefore our first natural research question was about exploring engineering disciplines such as process engineering and control systems for models of the physical processes (under control) and evaluating their suitability for conducting cyber-physical security research.

Research Question One *Can the available research models of physical processes be used to study the impact of cyber attacks on physical processes and their supporting control system?*

We narrowed our search to the chemical engineering field and chemical plant models. What makes chemical plants excellent case studies? Chemical plants are large physical processes with very complex non-linear interdependencies. Their models include disturbances with a wide spectrum of dynamic behaviors. Also, plant models include simulation of controllers. Chemical plant models have challenging objective functions to maintain: optimal production, safety and minimization of production costs. Those functions also allow to numerically evaluate the success of an attack from several perspectives.

Research Question Two *Are there attack vectors and/or attack scenarios that are unique to cyber-physical systems? Are traditional IT security defenses adequate to thwart CPS-specific attack techniques?*

The art of target exploitation is about finding a previously unconsidered attack scenario(s) and thus not yet preventable/detectable with security controls. Cyber-physical systems are real-

time applications that must respond to external stimuli within a finite and predictable period to remain controllable and productive. The effectiveness of such responses depends on the timely availability of process state information and the punctual dispatch of accurate control commands. By taking advantage of sensing and control action scheduling the attacker can orchestrate the dynamic behavior of the physical process to her advantage. While conventional IT network security measures, if implemented, prevent highjacking and modification of communication traffic, they do not deal with the timeliness of data packet arrival.

Research Question Three *Are there data security requirements specific to cyber-physical systems? Are traditional IT data security controls adequate to meet CPS-specific data security needs?*

In the IT domain, data and information security are predominately discussed in the context of the Confidentiality, Integrity, Availability (CIA) model where each component represents a fundamental objective of information security and is used to guide the development of data security policies in an organization with associated security controls. In the IT world, information security requirements only address the data which are already part of the digital infrastructure, the validity or truthfulness of the data is outside of the CIA scope. Capturing process data or information about the state of the physical process is a core function of a cyber-physical system. Consequently ensuring the data received truthfully captures the aspects of the physical world relevant to the application at hand becomes a crucial fundamental security requirement in the cyber-physical discipline.

Research Question Four *Is the cyber-physical attack lifecycle different from the traditional IT attack lifecycle or kill chain? What are the essential attack steps to achieve the desired impact in the physical world?*

Adversarial studies constitute an important part of cyber security defense strategies. Attacker actions are often illustrated in form of a cyber kill chain or attack lifecycle to better understand how to impede adversarial chances to achieve the desired outcome of a conducted attack. In the IT domain, the primary focus of such studies is on understanding attacker activities related to intrusion and maintaining persistent access. The ways of achieving the attack objective are typically not detailed due to their assumed common knowledge. To achieve an effect in the physical world, the attacker has to find ways of manipulating physical processes in the target environment, a skill not commonly found or understood in the IT security community. Insights into cyber-physical attack steps are required to understand the detours an attacker may have to take to accomplish her mission in the cyber-physical domain.

1.3.2 Thesis Contributions and Overview

We start the thesis by introducing the concept of cyber-physical systems and explaining the challenges of securing these systems in **Chapter 1**. We formulate our motivation and research questions. Parts of this chapter were previously published in the following papers:

- Cyber-Physical Systems Security, Dieter Gollmann, Marina Krotofil. Book chapter in The New Codebreakers – Essays Dedicated to David Kahn on the Occasion of His 85th Birthday, 2016
- Securing Industrial Control Systems, Marina Krotofil, Klaus Kursawe, Dieter Gollmann. Book chapter in Security and Privacy Trends in the Industrial Internet of Things, 2018

We then introduce background information on process control and Process Control Systems, necessary to understand the contributions of the thesis in **Chapter 2**. The introduced preliminaries are essential for the comprehension of the discussion when interpreting research results and conclusions in particular. Parts of the Section on Safety Systems were published in the following papers:

- Industrial control systems security: What is happening? Marina Krotofil, Dieter Gollmann. Proceedings of 11th IEEE International Conference on Industrial Informatics, INDIN 2013, Bochum, Germany, July, 2013
- Are You Threatening My Hazards?, Marina Krotofil, Jason Larsen. Advances in Information and Computer Security, IWSEC 2014, Lecture Notes in Computer Science, vol 8639, Hiroasaki, Japan, August, 2014

In **Chapter 3**, we present two models of chemical processes used as testing environments in our research. We first explain what makes these models suitable for cyber-physical security research and then present the modifications to the original models to enhance their research value and usage convenience. The content of this chapter was introduced in the following publications:

- The Process Matters: Ensuring Data Veracity in Cyber-Physical Systems, Marina Krotofil, Jason Larsen, Dieter Gollmann. Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS 2015, Singapore, April 14-17, 2015
- Cyber-Physical Systems Security: Experimental Analysis of a Vinyl Acetate Monomer Plant, Marina Krotofil, Alexander Isakov, Alexander Winnicki, Pavel Gurikov, Jason Larsen, Dieter Gollmann. Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, CPSS 2015, Singapore, April 14 - March 14, 2015.
- Rocking the Pocket Book: Hacking Chemical Plants for Competition and Extortion, Marina Krotofil, Jason Larsen, Alexander Isakov, Alexander Winnicki, Pavel Gurikov, Dieter Gollmann. White Paper, DEF CON 23, Las Vegas, NV, USA, August 6-9, 2015.

We then discuss the limitations of IT security controls to prevent and detect CPS-specific attacks in **Chapter 4**. We first present a novel exploitation method called “Stale Data Attack” which exploits real-time requirements of CPS and then show the non-trivial amount of uncertainties the attacker faces when conducting an attack on an unfamiliar physical process. We show that by adding additional barriers to the exploitation chain it is possible to reduce the attractiveness of certain attack scenarios even to capable threat actors due to a high risk of failure. The content of this chapter appeared in the following publications:

- Resilience of Process Control Systems to Cyber-Physical Attacks, Marina Krotofil, Alvaro A. Cárdenas. Secure IT Systems – Proceedings of 18th Nordic Conference, NordSec 2013, Ilulissat, Greenland, October 18-21, 2013. Springer LNCS, volume 8208: 166–182.
- Timing of Cyber-Physical Attacks on Process Control Systems, Marina Krotofil, Alvaro A. Cárdenas, Kishore Angrishi. International Conference on Critical Infrastructure Protection VIII, ICCIP 2014 – 8th IFIP WG 11.10 , Arlington, VA, USA, March 17-19, 2014
- CPS: Driving Cyber-Physical Systems to Unsafe Operating Conditions by Timing DoS Attacks on Sensor Signals, Marina Krotofil, Alvaro A. Cárdenas, Bradley Manning, Jason Larsen. Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8-12, 2014

- Vulnerabilities of Cyber-Physical Systems to Stale Data – Determining the Optimal Time to Launch Attacks, Marina Krotofil, Alvaro A. Cárdenas, Jason Larsen, Dieter Gollmann. International Journal on Critical Infrastructure Protection (IJCIP), Vol.7 (2014): 213-232.

In **Chapter 5** we discuss a data security property intrinsic to cyber-physical systems called veracity. Process data originate in the physical world and constitute a root of trust. If incorrect, a control system is incapable of guaranteeing the correct performance of the process. We first propose an attack scenario in which we forge sensor signals inside of the sensor's microcontroller and then propose a process-aware detection method with entropy metrics based on plausibility and consistency checks. The content of this chapter appeared in the following publication:

- The Process Matters: Ensuring Data Veracity in Cyber-Physical Systems, Marina Krotofil, Jason Larsen, Dieter Gollmann. Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIA CCS 2015, Singapore, April 14-17, 2015

We then propose a cyber-physical attack lifecycle as a framework to design and evaluate attacks directed at physical processes in **Chapter 6**. In the example of a Vinyl Acetate Monomer plant we illustrate the applicability of the proposed attack lifecycle and provide detailed insights into attacker activities when achieving the desired attack outcome of persistent economic damage to the production process. The content of this chapter appeared in the following publication:

- Cyber-Physical Systems Security: Experimental Analysis of a Vinyl Acetate Monomer Plant, Marina Krotofil, Alexander Isakov, Alexander Winnicki, Pavel Gurikov, Jason Larsen, Dieter Gollmann. Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, CPSS 2015, Singapore, Republic of Singapore, April 14 - March 14, 2015.

To conclude the thesis, we first briefly discuss further research results related to the research objective of this thesis and which were not included in the thesis to keep its size manageable. We then propose future directions of cyber-physical research (**Chapter 7**).

Note on images used in this thesis

Some figures used in this thesis are meant to provide an overall visual impression about the subject to the reader as opposed to providing details. Such figures can contain elements that may appear unreadable. Comprehension of these elements is not necessary for the understanding of the thesis.

1.3.3 About the Title of this Thesis

When we started research for this thesis, we kept an open mind about which research results would be possible to achieve. The main objective was to separate our line of thinking from the conventional cyber security concepts and focus on the control and physical components of cyber-physical systems. As much as possible, we kept our research fundamental in nature so that the results would be applicable to a wider range of cyber-physical systems and not only to the selected use-case of continuous processes such as chemical plants. Because the main attack objective considered in this thesis was to cause a specific or targeted impact on the physical process, we chose the “process-aware approach” as the main theme of this dissertation.

Chapter 2

Preliminaries: Process Control Systems

IIoT means you license a pump, and it phones home regularly to make sure you made your monthly payments.

Michael Toecker

Industrial control system (ICS) also known as Operational Technology (OT) is a collective term used to describe different types of control systems and associated instrumentation, which includes end-point devices, systems, networks, applications and controls used to operate and/or automate industrial processes. Process Control Systems (PCS) is a special case of Industrial Control Systems, which refers to monitoring and managing continuous or batch processes such as chemical plants or water utilities in order to ensure the desired output.

Depending on the complexity of a process and the spread of its supporting infrastructure, the process control function may be implemented with various types of control systems. The distinction between the major types of control systems is visualized in Figure 2.1 and can be summarized as follows:

- Programmable Logic Controller (PLC): Typically used to control a small-size/low complexity process or a unit of a larger process. The control function of a PLC may or may not be monitored by a human operator;
- Distributed Control System (DCS): Typically used for a large-scale, complex process in a single location. DCS includes an integrated control center for supervisory process control by human operators;
- Supervisory Control and Data Acquisition (SCADA): Typically used to control a complex process distributed via a large geographic area and may include several manned and unmanned control rooms/centers.

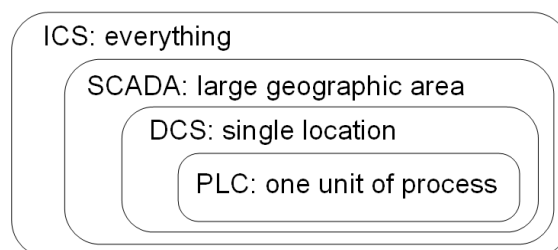


Figure 2.1: Distinction between various types of control systems

In practice, SCADA systems have evolved over years and become very similar to DCS in functionality. It is not uncommon that both types of control systems are used in the same industrial site, supplemented with PLC-based controls to administer small supporting functions. In this thesis we focus on the architectures and terminology closely related to distributed control systems due to their prevalent usage in the (petro)chemical sector and other large-scale continuous processes.

2.1 Fundamentals of Process Control

Process control is an umbrella term for both an engineering discipline and engineered infrastructures which include various systems and software that exert control over production processes. Control systems include sensors and data processing electronic units, actuators, networks to connect equipment, and algorithms to relate process variables to product attributes.

In the process industry *process* refers to the methods of changing or refining raw materials to create an end product. *Control* refers to the methods that are used to control process variables when manufacturing a product. This is done for three major reasons: (1) Reduce variability, (2) Increase efficiency, (3) Ensure safety. The first two points are important for plant economy. Reduced variability lowers operational costs and ensures consistent quality of the end product. Efficiency refers to the accurate maintenance of optimal production conditions to decrease the production bill. Precise control is important for preventing runaway processes and ensuring safe operations.

The complexity of modern production processes is usually simplified by dividing the control load into subsystems containing separate control loops. Control loop is a fundamental building block of industrial control systems. A block diagram of a basic feedback control loop is shown in Figure 2.2.

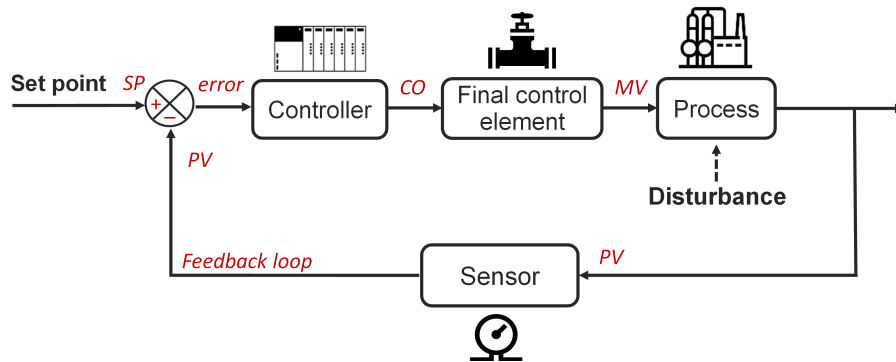


Figure 2.2: The components of a control loop

A control loop monitors the production process via sensors deployed around the production infrastructure and interacts with the process through actuators also called final control elements (FCE). The autonomous control function over a process is achieved through a specialized computing element called controller. Configuration of the control loop starts with a decision on a *set point* (SP) which is the desired value of a certain process parameter, e.g., a tank level L . Level L is called *process variable* or *process value* (PV) and must be kept as close to the setpoint as possible by the means of control methods (minimization of *error*). Level L can be measured directly in which case the PV may be called *measured variable*, or indirectly by measuring two process variables, e. g., in- and out-flows. Process variables are fed into a controller containing a control algorithm based on a complex set of equations. The controller calculates the offset between SP and PV and generates an actionable controller output (CO) signal to the FCE. The actuator then adjusts the *manipulated value* (MV), e.g, flow F to bring the process closer to the SP.

In practice, control loops can be complex. More common are multivariable or advanced control loops in which each MV depends on two or more of the process variables as shown in Figure 2.3. The strategies for holding basic and multivariable control loops at setpoints are not trivial, and the interactions of numerous setpoints in the overall process control scheme can be subtle and complex. Process interactions may cause loop interactions via hidden feedback control loops and may result in control loops instability. For this reason, heavy control loop couplings among subsystems are avoided.

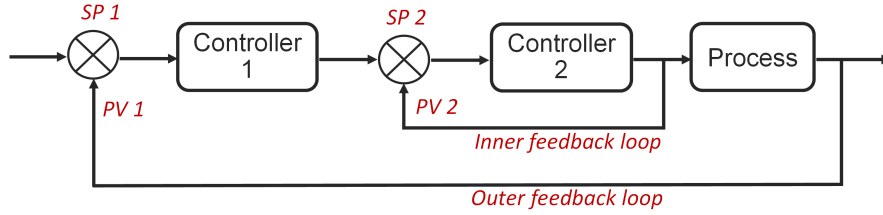


Figure 2.3: Multivariable control loop

A variety of process controllers are used to manage processes, e.g., On-Off controllers. However, the most common control algorithm used in industrial applications is a so called PID controller.

Definition: A Proportional–Integral–Derivative (PID) controller is a control loop mechanism employing feedback that is calculated according to the following formula:

$$u(t) = K_p e(t) + K_i \int_0^t e(t') dt' + K_d \frac{de(t)}{dt}, \quad (2.1)$$

where K_p , K_i and K_d are non-negative coefficients for the Proportional, Integral, and Derivative terms respectively.

The block diagram of the PID controller described with Equation 2.1 is shown in Figure 2.4a. The controller continuously calculates an error value $e(t)$ as the difference between a desired setpoint $SP = r(t)$ and a measured process variable $PV = y(t)$: $e(t) = r(t) - y(t)$, and applies a correction based on proportional, integral, and derivative terms. The controller attempts to minimize the error $e(t)$ over time by adjustment of a control variable or controller output $u(t)$ to a new value determined by a weighted sum of the control terms.

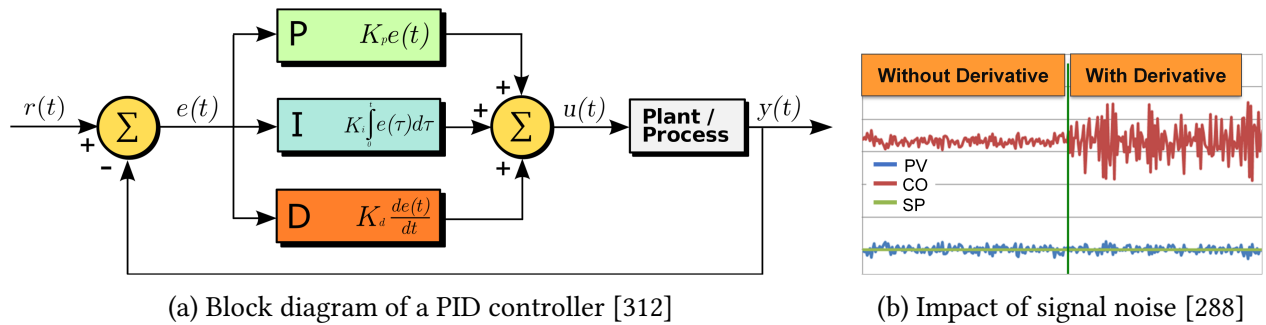


Figure 2.4: PID controller and impact of a derivative component on controller output

The PID coefficients are tuned (or weighted) to adjust their effect on the process. Controller tuning allows for optimization of a process and minimizes the error between the variable of the

process and its set point. Along formal approaches for controller tuning such as the Ziegler-Nichols and Cohen-Coon methods [288], a “trial and error” approach is often used due to its simplicity. In this method, the proportional action is the main control, while the integral and derivative actions subsequently refine it. First, the proportional gain is increased until the output of the loop oscillates around the setpoint. Once P has been set to obtain a desired fast response, the integral term is increased to stop the oscillations.

Even though the full PID controller in theory delivers the most accurate controller output, most practical control systems use very small or zero derivative coefficients K_d . The derivative response is proportional to the rate of change of the process variable and causes the controller output to decrease if the process variable is increasing rapidly. Because of its inherent properties, the derivative response is highly sensitive to noise in the process variable signal as shown in Figure 2.4b. If the sensor feedback signal is noisy, the derivative response can make the control system unstable. For this reason, PI controllers are the most frequently used in the industry.

An industrial plant may have thousands of measurements and control loops. *Plantwide process control* involves systems and strategies required to control an entire plant consisting of many control loops and interconnected unit operations with the emphasis on structural decisions. The structural decisions include the selection/placement of actuators and measurements as well as the decomposition of the overall problem into smaller sub-problems [192, 211]. There are two main approaches to the problem: a *mathematically-oriented* approach and a *process-oriented* approach. It was acknowledged in [192] that the mathematically-oriented approach to plant-wide control configuration is difficult to implement practically, both because of the size of the problem and the large cost involved in making a precise problem definition, which would include a detailed dynamic and steady state modeling (see also Chapter 3). An alternative to the mathematical approach is a design procedure based on heuristic rules founded on experience and understanding of the process (process-oriented approach). One of the prominent heuristic approaches was proposed by Luyben et al. [211]. The approach is composed of nine steps and centers around the fundamental principles of plantwide control: energy management; production rate; product quality; operational, environmental and safety constraints; liquid-level and gas-pressure inventories; the makeup of reactants; component balances; and economic or process optimization.

While basic process controls are designed and built with the process itself to facilitate fundamental operational requirements, Advanced Process Controls (APC) are typically added subsequently, often incrementally over the course of many years, to address particular performance or economic improvement opportunities in the process. Figure 2.5 illustrates the economic benefits of advanced process control.

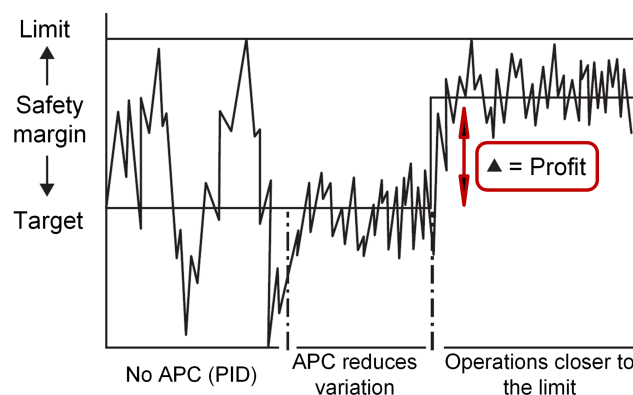


Figure 2.5: Benefits of Advanced Process Control [293]

One of the frequently employed APC methods is Model Predictive Control (MPC), a class of advanced process controllers capable of utilizing system information through a well-developed model and real-time process measurements to predict the future trajectory of the process. This is achieved through complex predictive modeling, which then allows the control system to take corrective action in advance, to ensure the process remains within the optimal trajectory in the future. A key feature of MPC is that future process behavior is predicted using a dynamic model and available measurements. Due to the requirement of having a high-precision process model, MPC is predominately used in control loops critical to plant economy and safety.

2.2 Process Control Systems Architecture

The first generation of process control systems used to be pneumatic and based on manually operated relay systems. Later control systems evolved into analog electronic systems which allowed to automate the majority of labor-intensive tasks. In the 1970s a new generation of computerized control systems emerged resulting in a new discipline called computer-aided manufacturing. Over the past decades, these systems evolved in parallel with general-purpose computing technologies to become a highly sophisticated distributed ecosystem of applications and hardware systems. Because computers allowed to automate manual tasks, modern process controls are also called *automation systems* or simply *automation*.

For many years control systems used to be air-gapped with process management being executed locally within the boundaries of individual production sites. With more data becoming available and increased complexity of equipment and process control approaches the need for interconnection of process control and enterprise networks as well as for third-party remote access has emerged. In the 1990s, T. J. Williams, a member of the Purdue University Consortium for Computer Integrated Manufacturing, published the Purdue Enterprise Reference Architecture (PERA) that provided guidance on the interface design between process control functions and enterprise functions [319]. On the premise of the PERA model, the International Society of Automation (ISA) released a layered network model which described common vendor-independent information flows which can be applied in any manufacturing or process industry production architecture known also as the ISA-95 international standard [63]. It quickly became a de-facto standard for operational technology professionals to think about, design and implement industrial control systems.

To address cyber security concerns and ensure secure integration of OT and IT environments ISA developed ISA-99 [64] which is currently known as the IEC-62443 series of standards on the cyber security of Industrial Automation and Control Systems (IACS) [65]. The collection of IEC-62443 standards and technical reports is organized into four general categories called General, Policies and Procedures, System and Component. Part IEC-62443-1-1 of the standard specifies how IACS assets should be organized into network layers based on device function and requirements as shown in Figure 2.6. The industrial network architecture also involves provisioning of the OT Demilitarized Zone (DMZ) zone to securely segregate the OT environment from the untrusted enterprise IT network and the Internet.

The Purdue model and the IEC-62443 reference architecture are often referenced synonymically due to their direct relation. They are regarded as industry best practice and are a widely adopted concept model for the logical network segmentation of industrial control systems. Below we give a summarization of each network level, including example devices and systems, and their functions.

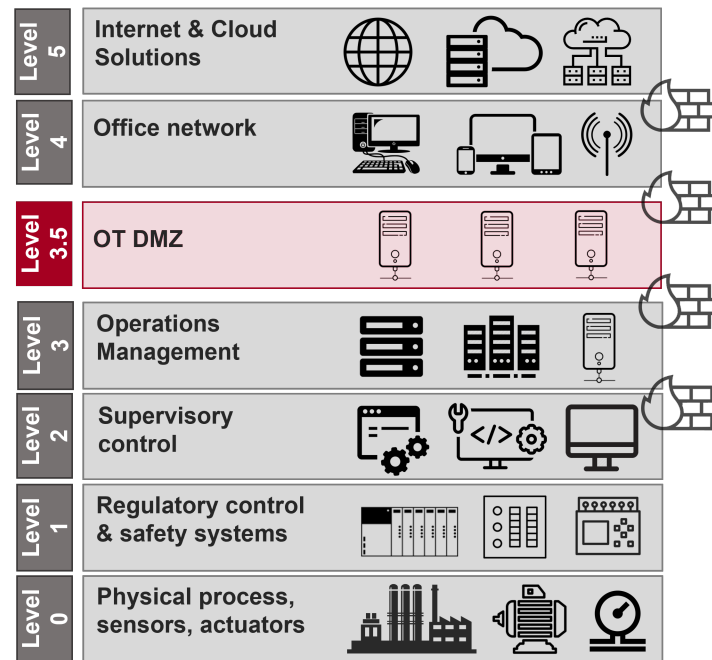


Figure 2.6: Purdue reference architecture for process automation

Level 0: Physical Process, Sensors and Actuators

The lowest level of the automation hierarchy includes the physical process and supporting physical infrastructure, and defines activities involved in sensing and manipulating the physical processes. This level is also called *Field Level*. Correspondingly, sensors and actuators are often collectively called *field instrumentation*. Figure 2.7 gives a view of a physical level infrastructure from a water treatment environment.

Process equipment (static). This group of process equipment includes static process infrastructure such as pipes, various vessels and storage tanks, reactors, distillation columns, filtering and absorption equipment, ovens, etc.

Process equipment (dynamic). This group of equipment includes equipment which, if actuated/powered, is capable of executing dynamic/kinetic behaviors. Examples include pumps, compressors, furnaces, fans, conveyers, robots, etc.



Figure 2.7: Physical process infrastructure and field equipment [155]

Sensing equipment. In broad terms, this group of equipment captures various parameters of the physical world. Examples include various types of sensors such as temperature, pressure or flow; proximity, fire and gas detectors; chemical/quality analyzers, etc.

Actuating equipment. An actuator converts a control signal into a mechanical action. Examples include valves, electric motors, steam or gas turbines, internal combustion engines, etc. Strictly speaking, although valves are frequently seen as monolithic elements, practically they consist of two elements - a mechanical part of the valve and devices designed to automatically control/actuate (valve actuators) and monitor the position of valves in relation to their open or closed positions (valve positioners). Similarly, pumps are often seen as actuating equipment despite their actuating function being dependent on the accompanied motor.

In the distant past field instruments were 3-15 psi “dumb” pneumatic-based devices that performed control via local single-loop controllers. Information was shown locally with gauges and recorded manually with pen and paper by a technician. In special cases local chart recorders were used when data needed to be saved for analysis. A small degree of intelligence and ability to communicate with remote controllers was next added to field instruments in the form of a 4-20 mA analog signal that represents 0 to 100% of process variable or controller output. Currents less than 3.8 mA or more than 20.5 mA are seen as indicators of a fault and with that provide rudimentary diagnostic functionality. With the introduction of the HART protocol it became possible to transmit some data and settings over the 4-20 mA analog signal with up to 1200 bps speed in half-duplex mode. A major step forward occurred when microprocessors became small and robust enough for installation in field instruments, enabling local conversion of an analog signal into its digital representation suitable for transmission over a network. Added computing capabilities allowed instruments to manage other tasks such as calibration and complex self-diagnostics and with that transforming them into *smart instruments*.

Modern smart sensors (also called transmitters) can send back multiple readings along with alarm conditions. Valves now have computing and data storage capabilities to calculate and retain in the local data history a current valve signature of pressure vs. stem travel, compared with the signature when the valve was installed, and provide diagnostic information or alarms based on the detected differences. Despite the popularity of smart sensors, cheap simpler sensors are still often used for monitoring on major equipment. For instance, a standard electric motor may now have over a dozen of sensors providing real-time measurements to determinate the health of equipment. These data is sent to an upper layer of the control network architecture where it is fed into an application for predictive maintenance and other purposes.

As the extent of field data consumption increased, field data started to be called telemetry data. Telemetry engineers are responsible for the accuracy of data during their acquisition in the field, in transit and storage. It is worth mentioning that raw sensory data rarely can be used directly. The electrical output of a sensing element is usually small in value and has non-idealities such as offset, sensitivity errors, non-linearities, noise, etc. Therefore, raw sensor or, more precisely, transducer output is subjected to signal conditioning such as amplification, filtering, range matching, etc. Acquired signal conditioning may happen directly in the sensor or in external dedicated signal processing modules. The example of a conditioned and digitized signal is shown in Figure 2.8a. Sensor signals are combined together, aggregated with other sensor signals to extract additional new information about process performance and state of the plant. To drive efficiencies, sensors are increasingly communicating directly with their vendor monitoring software in the cloud via cellular networks.

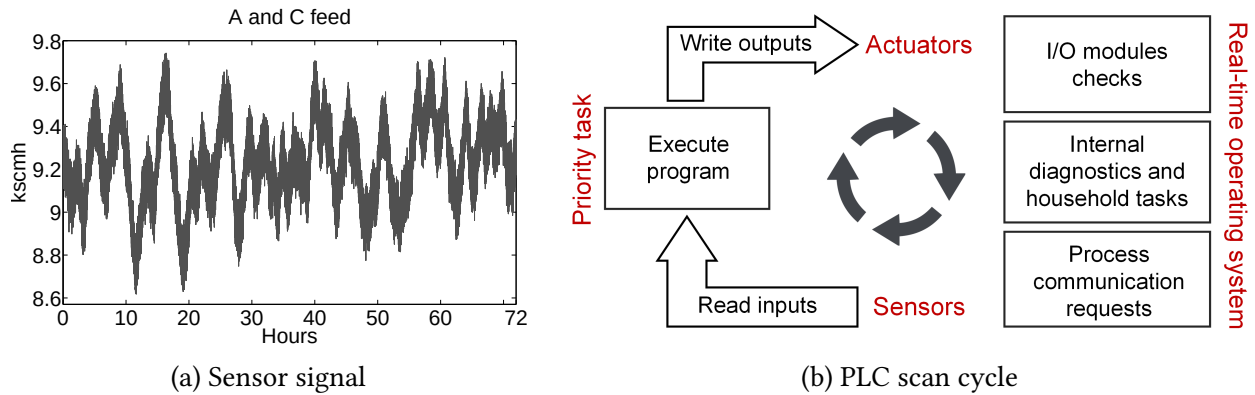


Figure 2.8: Sensor signal and PLC scan cycle

Level 1: Regulatory Control

The level immediately adjacent to Field Level hosts controlling equipment responsible for monitoring operating parameters and controlling functions of the physical process and equipment. This level is called Regulatory or Basic Control. The types of control equipment include PLCs, Variable Frequency Drives (VFD), dedicated PID controllers, Remote Terminal Units (RTU), etc. Controllers interface field instruments via analog or digital Input/Output (I/O) modules. In this section we will use the example of a Programmable Logic Controller to describe the major concepts related to regulatory control.

The main work flow of the programmable controller consists of two parts: the process of self-diagnosis and communication response and the execution process of the *user program* also called *control logic* as shown in Figure 2.8b. This work flow is repeated several times a second and is called *scan cycle*. In relation to process control, the scan cycle consists of three steps: reading inputs, solving control logic and writing outputs. The PLC queries each input card and saves the readings in a variable table (VT). VT contains all the variables needed by the control logic: setpoints, counters, timers, inputs and outputs. During the program scan cycle, every change in the I/O of the PLC is ignored until the next program scan cycle. The PLC executes the user program one instruction at a time and when the program execution completes, the outputs are updated using the temporary values from VT. The PLC updates the status of the outputs and restarts the scan cycle by starting a self-check for faults. The PLC scan cycle is an infinite loop which runs until reboot or power off.

Execution of the scan cycle takes some amount of time. This time is called the *scan time* and often amounts to milliseconds (ms). Usually industrial environments mandate real-time control over an industrial process. Failing to execute control operations in a timely manner may result in the failure of an industrial process, which leads to unacceptable consequences. To overcome this problem, the majority of PLCs are equipped with Real Time Operating Systems (RTOS) to execute their tasks in a predictable manner and within strict time constraints.

User program or control logic, as the name suggests, defines what operations should happen, when and under which conditions. It also contains so called interlocks which define mutually exclusive conditions to prevent undesired (harmful) states of the process. Industrial controllers are programmed by control engineers. The code can be developed in-house but mostly frequently it is developed by external engineering companies (third-party subcontractors). The source codes of control logic are stored on the Engineering Workstation and/or on data servers or data shares. The

Default tag table									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervis...	Comment
1	inputs	Byte	%I0.0	☐	☒	☒	☒		Slot for inputs
2	Sensor1	Bool	%I0.0	☐	☒	☒	☒		Sensor Conv 1
3	Sensor2	Bool	%I0.1	☐	☒	☒	☒		Sensor Conv 2 for small and big boxes
4	Sensor5	Bool	%I0.2	☐	☒	☒	☒		Sensor Conv 2 for big boxes
5	Sensor3	Bool	%I0.3	☐	☒	☒	☒		Sensor for Conv 3
6	Sensor4	Bool	%I0.4	☐	☒	☒	☒		Sensor for Conv 4
7	Sensor6	Bool	%I0.5	☐	☒	☒	☒		Conveyor 2 is turned
8	Sensor7	Bool	%I0.6	☐	☒	☒	☒		Conveyor 2 is in home pos.
9	outputs	Byte	%Q0.0	☐	☒	☒	☒		Slot for outputs
10	MotorControl1	Bool	%Q0.0	☐	☒	☒	☒		Conveyor 1 moves forward
11	MotorControl2	Bool	%Q0.1	☐	☒	☒	☒		Conveyor 2 moves forward
12	MotorControl3	Bool	%Q0.2	☐	☒	☒	☒		Conveyor 3 moves forward
13	TurnControlFwd	Bool	%Q0.3	☐	☒	☒	☒		Conveyor 2 turns forward
14	TurnControlRev	Bool	%Q0.4	☐	☒	☒	☒		Conveyor 2 turns reverse

Figure 2.9: Example of a tag table in a PLC [3]

IEC standard 61131-3:2013 [61] outlines five PLC programming languages. These are split into two categories: (1) Graphical (Ladder Diagram, Function Block Diagram, and Sequential Function Chart) and (2) Text Based (Instruction List, Structured Text). Although these languages are vendor and application agnostic, vendor specific language subsets are common as well. An example of a ladder diagram with a function block for a Siemens PLC can be seen in Figure A.2. The control logic is compiled with vendor-proprietary compilers and uploaded to a PLC for execution in form of bytecode or binary code.

One unique characteristic seen across industrial control systems is their use of points - a concept not applied in conventional IT systems. Points are responsible for all aspects of an ICS, essentially denoting a data source or a controllable function. Each point is assigned a unique ID, e.g., I1.1, DB1.DBX1.1, etc. ID values can be input channels, memory locations or other process specific variables. Point IDs are supplemented with more logical “friendly” names (e.g., Valve 1) called tags. Tag names are descriptive and given to points for operator convenience. In large facilities tag naming/numbering typically follows a site-specific nomenclature in form of a “code” (e.g., TMHS KQP536NR) and tag description is provided as a separate attribute. Tags typically encode some meaningful data like plant area code, equipment type, equipment tag, unique sequence number and similar. Points can be soft or hard. A hard point denotes a physical input or output to/from sensors and actuators. A soft point denotes results derived from mathematical calculations and logic actions. From a software perspective, a tag is considered much the same as a variable name. In essence, points and their associated tags link external devices and process parameters with process control applications, and can have either Read, Write, or Read/Write permissions. Figure 2.9 shows an example of a tag table in a PLC.

Controllers communicate with field instruments and with each other via real-time communications protocols. Many industrial protocols were initially designed to communicate serially over RS-232/485 physical connections at low speeds, but have since evolved to operate over Ethernet networks using routable protocols such as TCP/IP and UDP/IP. A good example of this evolution would be the Modbus protocol which evolved from its serial version Modbus RTU to the Ethernet version called Modbus TCP. For many years Modbus held the title of the most widely used network protocol in industrial applications. With time a variety of proprietary and functionally richer protocols such as EtherNet/IP, PROFINET and EtherCAT began to prevail.

An interlock functionality mentioned above can be also seen as a basic safety feature that makes the state of two mechanisms or functions mutually dependent. In most applications, an interlock is used to help prevent a machine from harming its operator or damaging itself by preventing one element from changing state due to the state of another element, and vice versa. For instance the

pump should not run if there is no flow or if lubricating material is not supplied. Interlocks can be not only programmable but also consist of any electrical, electronic, or mechanical devices or systems such as the Safety Instrumented Systems discussed below.

Safety Systems

Besides basic process control systems (BPCS) Levels 0-1 may also include an independent control system to carry out safety functions. Safety Instrumented Systems (SIS) provide dedicated fail-safe operations of high reliability. They are used to protect against catastrophic consequences such as equipment damage, severe environmental impact and personal injuries/casualties by shutting down the production application. SIS typically consist of sensors and logic functions that detect dangerous conditions and final elements, such as valves, that are operated to achieve a safe state. SIS are categorized by a Safety Integrity Level (SIL) – a performance measure that is related to the probability that the safety instrumented function will not work when challenged (when needed). The required SIL may be determined from Hazard and Operability Studies (HAZOP), Failure Mode and Effect Analysis (FMEA), Layers of Protection Analysis (LOPA), risk graphs, and other methods. The higher the severity of hazards, the higher the SIL level of the safety controller must be.

Historically, safety systems were hardwired and segmented off from the main control system. During recent years SIS evolved to the point where they now include programmable electronic technology and an IP stack, which allows for over-the-network changes to the functionality of these system. To maintain high reliability of an SIS, it is advisable to segregate BPCS and SIS functions with the needed communication conduit being tightly configured. As best practice, the data from safety systems is first communicated to a regulatory controller via a point-to-point communication link (Modbus RTU or Modbus TCP) and then forwarded to the systems in the upper layers of the control architecture. Some vendors offer Integrated Control and Safety System (ICSS) solutions which combine elements of process control and functional safety into a single architecture. Usage of such architectures is typically driven by the cost reduction of plant engineering and subsequent ease of maintenance through integrated tools.

Level 2: Supervisory control

Individual controllers contain only a small portion of control logic and therefore can control the process to achieve a local optimum at most. As the title suggests, this level is involved into supervising the control function over the whole plant, typically done by human operators and engineers. The main component of this level is the control room (or control center) where operators monitor and control the plant through Human Machine Interface (HMI) software which contains a graphical representation of the process as shown in Figure 2.10. If several different types of control systems are used in the plant, their data are jointly displayed in the control room. The supervisory control function is performed 24/7 in shifts. Whenever a process deviates too much and requires correction, operators may change a setpoint, manually operate actuators and even overwrite interlock conditions.

The operators are notified of abnormal process conditions or equipment malfunctions that require actions to be taken by an application called alarm system. Alarm management is the most critical function of the supervisory control as the potential consequences of an unresolved abnormal situation can be dramatic, ranging from significant financial losses to casualties as shown in Figure 2.11. The alarm systems can include both the basic process control system and the safety instrumented system, each of which uses measurements of process conditions and logic to generate

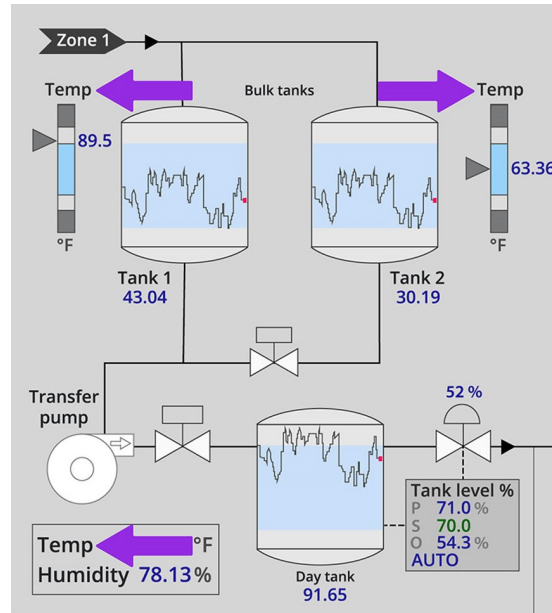


Figure 2.10: Example of Human Machine Interface view [90]

alarms. Packaged systems such as fire and gas systems or complete packaged units are typically included into overall alarm management systems. Figure 2.12 illustrates the process of alarm generation, visualization and logging in the alarm system. Most frequently alarm information is communicated to the operator via an HMI, usually a computer screen or an annunciator panel. The number of alarms to be triaged per a period of time is strictly regulated to maintain appropriately sufficient cognitive function by the operators. In process industries the acceptable alarm rate is equal to 6-7 alarms per hour or 1-2 per 10 min as shown in Table 2.1, with critical alarms not exceeding 10% of the total number of alarms [128].

Response Class		Available Response Time	PRIORITY CLASS				
	SHORT	< 5min	L	M	E	E	E
	MEDIUM	5 – 15 min	L	M	M	E	E
	LONG	> 15 min	L	L	M	M	E
Consequence Category	ECONOMICS		No/Slight Effect (< 10k)	Minor Effect (10k – 100k)	Medium Effect (100k – 1M)	Major Effect (1M – 10M)	Extensive (>10M)
	HEALTH & SAFETY		No/Slight Injury	Minor Injury	Major Injury	Single Fatality	Multiple Fatalities
	ENVIRONMENT		No/Slight Effect	Minor Effect	Local Effect	Major Effect	Massive
CONSEQUENCE CLASS			Negligible	Low	Medium	High	Extreme

Figure 2.11: Alarm criticality and consequences cost (source is not specified for sensitivity reasons)

The other systems found at the supervisory level include typical components of a Distributed Control System: various applications for configuration, maintenance and diagnostics of the control system, control logic and equipment, databases, communication and batch management servers, engineering desktops and others. A key element in the control system infrastructure is an Engineering Workstation (EWS), a high-reliability and sometimes ruggedized computing platform equipped with software applications and tools to program and modify the industrial process. It

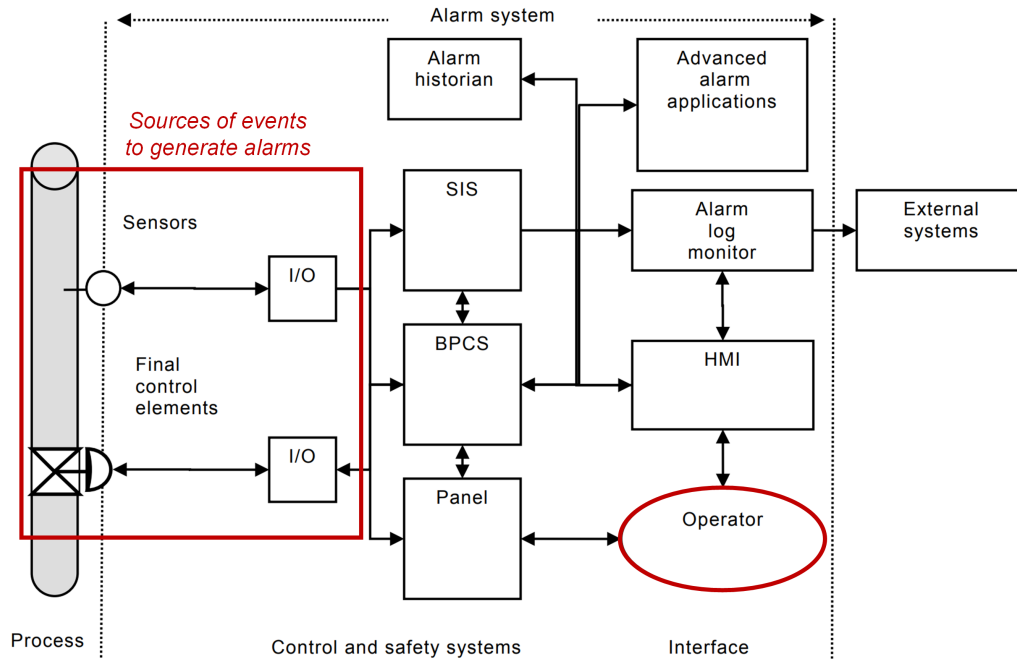


Figure 2.12: Alarm system generation and response workflow [62]

Average Alarm Rate	Acceptability	Performance and risk
< 10	Very likely unacceptable	Inefficient / High Risk
5 – 10	Likely over-demanding	Medium performance & risk
2 – 5	Possibly over-demanding	
1 – 2	Manageable	
> 1	Very likely acceptable	Efficient / Low Risk

Table 2.1: Impact of alarm rate on operator acceptability and performance, per 10 min in steady state, based on [128]

includes specialized tools needed to directly communicate with, configure, and update the primary control equipment such as PLC, BPCS, SIS, intelligent electronic devices (IED) and others. Due to the nature of its tasks, an EWS contains significant amounts of sensitive documentation specific to the process and infrastructure design, configuration, and plant operations. It is also frequently used to store the historic information related to process configuration change. Another key element of the control system is a purpose-built real-time database (RTDB) for short-term storage of process data to be consumed by other applications. The process RTDB may also include a local instance of the data historian application, a specialized software that collects point values, alarm events, batch records, and other information from industrial devices and systems for long-term storage. Additional systems may include asset management solutions, local backup servers and any other system which is required to support plant management operations.

Level 3: Operations Management

At this level reside systems that support functions involved in managing the work flows to produce the desired end products. Examples include Manufacturing Execution Systems (MES) and Manufacturing Operations Management Systems (MOMS). These systems take orders and

business data from the corporate systems to manage production scheduling and dependencies, ensure production capacity optimization and schedule dispatching of the finished goods. This level is also sometimes called “shared services” because it hosts applications and services jointly used by multiple plants such as centralized backup and asset management systems, network management functions, file servers and others.

Among the pillar systems hosted at this level is the Plant Information Management System (PIMS) also known as historian. Figure 2.13 shows the Graphical User Interface (GUI) of one of the most widely used historian software, OSIsoft PI. A PIMS collects and integrates information about production processes and infrastructures from a multitude of different sources, including local historian instances from Level 2. Historians use specialized algorithms for lossy and lossless data compression to reduce storage footprint as well as specialized software to enrich data with context and visualize statistical trends.



Figure 2.13: Data historian view [185]

It might not always be obvious whether a certain system belongs to Level 2 or Level 3. In general, no system or application which is directly involved in or depended upon for process control should be hosted at this level. In other words in case of issues with the L3 infrastructure, it should be possible to disconnect this network segment from the rest of the OT infrastructure without immediately affecting the process control function. For this reason an essential service such as historical storage is provisioned at both L2 and L3. However, because L3 hosts such services like scheduling systems and material flow applications, disruptions at the Operations Management level can lead to hours or days of interferences in production and enterprise processes, with potential for considerable revenue loss. In some instances it might be required to shutdown the manufacturing processes.

OT Zone

Levels 0-3 of the network reference architecture are collectively called *OT zone* or *Industrial zone*. The rise of automation leading to higher efficiencies has created an increased need for bidirectional data flows between OT and IT systems. The OT zone is required to be protected from the Internet-enabled networks such as enterprise layer and any other “untrusted” environments like third-parties. This is achieved through provisioning of a dedicated perimeter network also known as Demilitarized Zone which acts as a conduit system between the OT zone and other external environments.

Level 3.5: OT DMZ

OT DMZ is a recent addition to the Purdue model. Similar to a conventional IT DMZ, the OT DMZ is a buffer zone where services and data can be shared between two networks belonging to different trust zones. OT DMZ is the only non-functional level and provides an interface where IT and OT worlds “converge”. For this reason, the OT DMZ level is frequently counted as 3.5 to highlight its extension to L3. Among key functions of the DMZ is to enable communication between OT and enterprise applications in order to exchange manufacturing and resource data. For this purpose the historian from L3 is fully or partially mirrored to a historian instance at the DMZ. The DMZ typically also hosts patch and antivirus servers as well as applications for secure file exchange and remote access to OT assets for third parties.

When correctly implemented, no communication connection should directly traverse the OT DMZ, all traffic should originate/terminate in the DMZ. For instance historian data from L3 are first sent to the historian instance in the DMZ and then sent to L4 for consumption by business applications. Similarly, all communication flows from the IT network or Internet first land in the DMZ and are then routed further via newly established sessions.

Level 4: Enterprise network

Level 4 or the Enterprise level, is where IT systems, applications and functions exist tasked with the management of the entire industrial systems estate and business processes. This is the level that provides business direction and orchestrates operations with data flowing up from the shop floor (L1) and decisions flowing down from the boardroom. Examples include Enterprise Resource Planning (ERP) systems that drive long-term plant production schedules, material use, shipping, inventory levels and logistics processes, business-to-business and business-to-customer applications and services as well as various business application such as emailing and voice communication, document management and data centers. This level also hosts various supporting functions such as advanced data analytics and associated applications, Research and Development (R&D), Security Operations Center (SOC) and others.

Level 5: Cloud applications

Level 5 or Cloud level does not officially exist in the Purdue or IEC 62443 reference architecture. However, we added it to illustrate two ongoing trends:

- Increasing desire of enterprises to reduce on-premise hardware footprint and reduce infrastructure maintenance cost. Various Software-as-a-Service (SaaS) applications such as Office 365 and modern endpoint security solutions are the examples of cloud-hosted business applications. In this regard, Level 5 could be seen as an extension of the office network;

- Digitalization of the industrial automation to achieve the next level of efficiency and operational excellence. This trend is also widely known as Industry 4.0 [249], where production and asset data are being directly fed into so-called data lakes and cloud-hosted applications for analysis and gathering insights.

In the past, collected data were stored in on-premise servers or data centers and consumed locally for local decision-making. Later data from various sites started to be aggregated at the enterprise level for enterprise-wide usage and decision making. With a wider adoption of cloud technologies and the evolvment of advanced machine learning techniques for big data analytics, data storage increasingly began to move to data lakes for ease of data sharing and simplified management of data access by the enterprise users, partners, third-party service providers and customers.

It would not be possible to cover all the benefits of industrial digitalization. Among the most mature and widely adopted examples of cloud-based application is Predictive Maintenance [254]. This is a proactive maintenance strategy that utilizes condition monitoring applications to detect various deterioration signs, anomalies, and equipment performance issues to estimate when a piece of equipment might fail so that maintenance work can be performed ahead of a potential breakdown. For this purposes complex pieces of equipment are instrumented with additional sensors to gather additional asset data as discussed in Appendix E. Instrumentation of assets with additional sensors is sometimes called asset digitization. Asset data is then used to create predictive models to detect signs of asset degradation.

Among emerging trends are Cloud-Based Design (CBD) and Cloud-Based Manufacturing (CBM) which refer to a distributed collaborative cloud-based engineering design services and manufacturing model that exploits on-demand access to a shared collection of distributed manufacturing resources to form temporary, reconfigurable production lines in response to demand [321].

Transformation of the Purdue Model

Asset monitoring and plant optimization applications with data flows utilizing Internet-based communication and being consumed by cloud-based software platforms has received the name of Industrial Internet of Things (IIoT) [32]. With the added sensors for asset and process monitoring, the network bandwidth of the BPCS became insufficient to transfer large volumes of high-resolution telemetry data. As a consequence, it was decided to send IIoT data via dedicated communication channels which included edge processing devices and communication gateways. Naturally, the Purdue reference architecture does not provide any guidance neither about how such IIoT communication architecture should be implemented nor how it should be integrated with the main OT infrastructure. To standardize IIoT implementation, NAMUR, an International User Association of Automation Technology in Process Industries, came up with the NAMUR Open Architecture (NOA) which provides guidance on the architectural design of IIoT data flows and intercommunication between IIoT and the core process control infrastructure. Figure 2.14 shows such a design for L0-L3/L4 for the reference architecture.

In the future the current Purdue model will continue to be challenged. With the industry's growing desire to reduce infrastructure deployment and maintenance costs, L3 and L2 computing resources are increasingly hosted in the cloud in form of Infrastructure-as-a-Service (IaaS) [59]. This trend will inevitably compel deployment of a DMZ zone in the cloud as well, calling for rethinking security requirements and the architectural design of the Purdue reference model. In the future, with L3 and L2 applications being consumed as SaaS offerings, the Purdue architecture will transform even further.

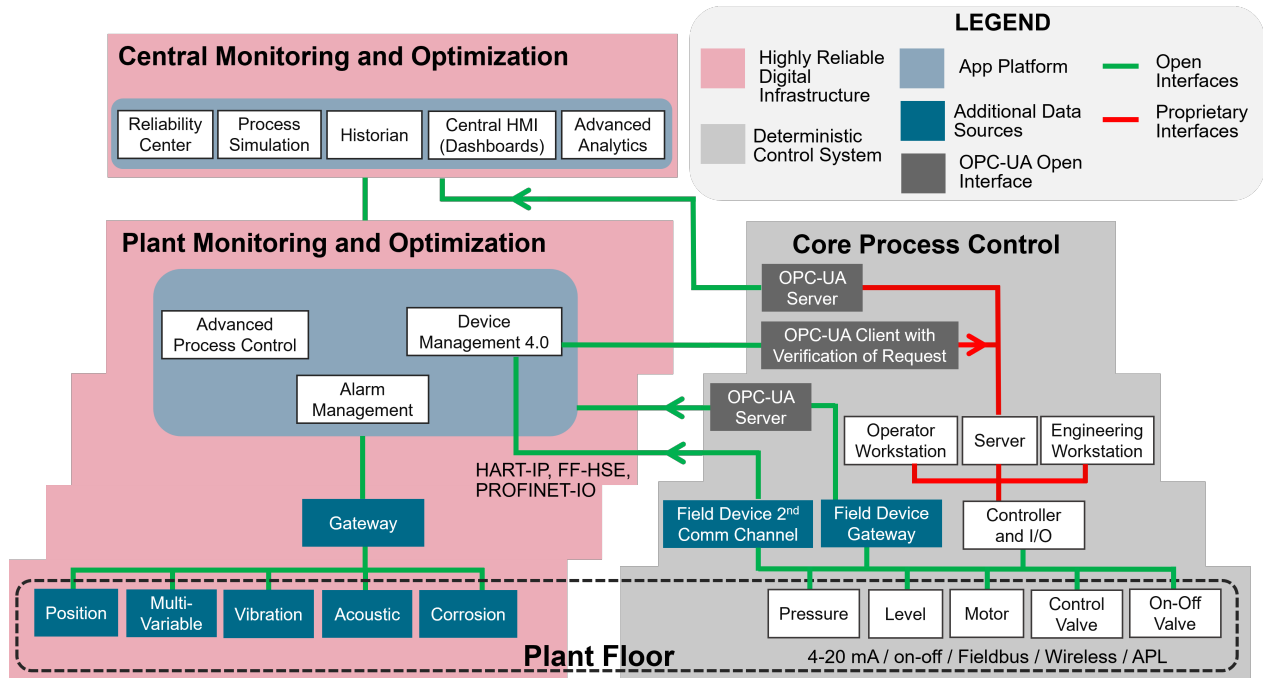


Figure 2.14: Open NAMUR reference architecture for IIoT applications [26]

Data processing in industrial automation

With the introduction of computer-aided manufacturing, data became the most essential ingredient of automation and processing this data into information became a substantial task of control systems. The key to handling information was the establishment of a transparent data flow inside an automation system with a strict subdivision of the data processing into a hierarchical model which resulted in an architecture known as automation pyramid [63] and shown in Figure 2.15. Established by the ISA-95 work group, this model became a foundation for the Purdue reference architecture discussed above. Allocation of devices to automation levels highlights the varying time constraints associated with data collection and response. The direction of the process data flow is bottom-up, while management and control data flows from top to bottom, with each layer adding latency, predominately related to data aggregation and analysis strategies.

At the level of regulatory/local control (L1) data has momentary significance and is used merely for real-time process corrections. Data exchange between field instrumentation and control equipment is happening in milliseconds or seconds intervals and data processing mostly addresses the quality of the sensed signals. While industrial controllers mostly operate on *raw data*, the systems in the upper layers of the automation pyramid operate on the *information* extracted from the data. Field data are contextualized for extracting useful observations by the means of various data processing and analysis techniques such as filtering, conditioning, aggregation, transformation, correlation and others. Supervisory control is about short-term tactical control by process operators and medium-term strategic monitoring by process supervisors and process managers. Process information is used for displaying trends, alarm management and overall situational awareness with data significance lasting in sec-min for operators and min-hrs for supervisors. Incorrect processing of process data may result in partial or complete loss of situational awareness as further discussed in Chapter 5 and Appendix C. Information at the level of operations management is concerned with correcting plant performance drifts and strategic long-term planning.

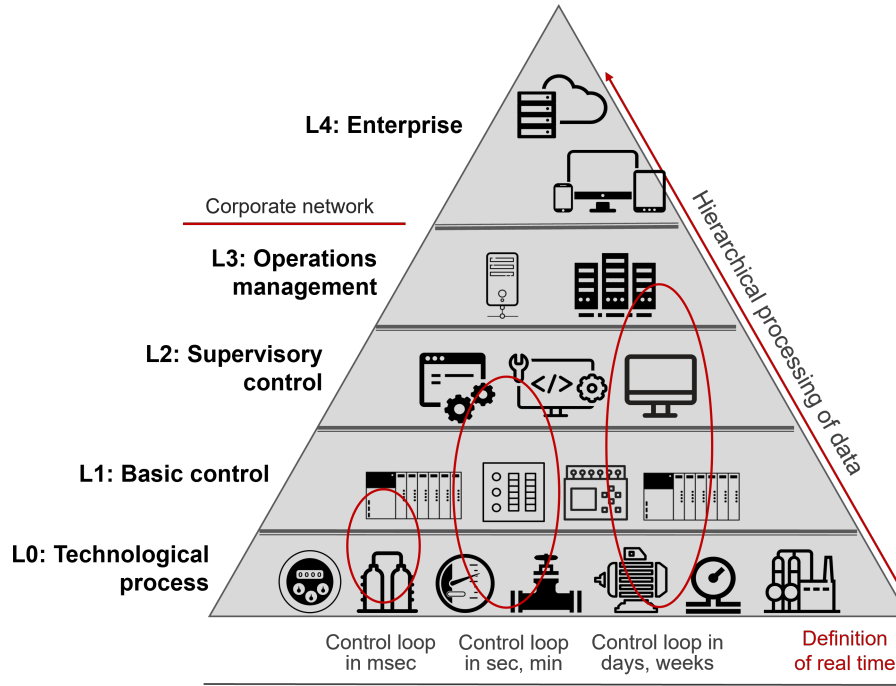


Figure 2.15: Automation pyramid according to ISA-95

In the past reliance of industrial equipment and software on proprietary protocols posed significant challenges to centralized acquisition of process data. To bridge the “communication gap”, an industrial automation task force consisting of industrial vendors and end users developed an OPC standard for communication interoperability in 1996 [106]. OPC stands for Open Platform Communications or OLE (Object Linking and Embedding) for Process Control in the initial version of the protocol name. OPC serves as a bridge between both Windows-based software applications such as historians and various process control hardware as well as control equipment of different vendors. Figure 2.16 illustrates the OPC communication topology for data acquisition. To date, OPC has become one of the most widely used interfaces for accessing process data not only in OT but also in IIoT applications and allows building multi-vendor architectures in a “plug and play” fashion. Thus OPC’s latest version OPC UA (Unified Architecture) is adaptable to different transport layers depending on the application-specific requirements and use cases, and offers vertical and horizontal integration, including field, edge and cloud. Also, OPC UA makes use of a universal Quality-of-Service (QoS) concept, which includes real-time communication capabilities with guaranteed bandwidth and low latencies.

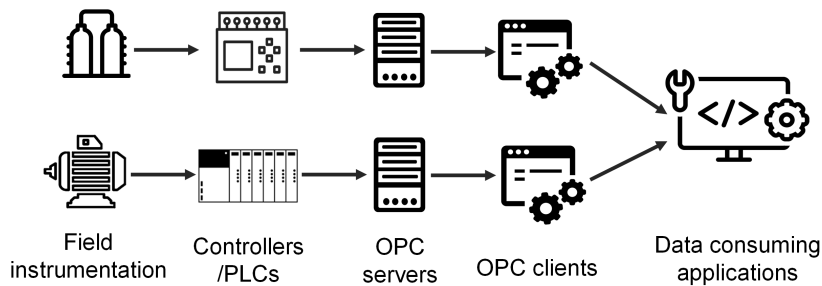


Figure 2.16: OPC communication topology

2.3 Cyber-Physical Attacks

ICS have traditionally been designed for dependability, durability, economic efficiency and safe use. Pervasive computerization and automation of control systems has enabled vertical and horizontal systems integration and introduced cyber interdependencies [262], which became a source of disturbances which are unusual and difficult to foresee. Subtle control loops, cascading failures and malware propagation were the price for increased efficiency. Technology has improved efficiency but at the same time has become a source of concerns. In the past few decades plants have undergone tremendous modernization. What used to be a panel of relays became an embedded computer. What used to be a simple analog sensor is now an IP-enabled smart transmitter [227] with multiple wired and wireless communication modes, a large number of configuration possibilities, and even a web-server so that maintenance staff can calibrate and setup the device without approaching it. While security engineers try to limit the attack surface, vendors keep introducing novel opportunities for remote exploitation of the physical processes and equipment.

Pervasive “cyberfication” became a source of concerns about plant vulnerability to both random cyber failures and security attacks. On one hand embedded computers have enabled the governing of physical applications to achieve desired outcomes. On the other hand physical systems can be instructed in the same way to perform actions that are not intended. As a result software code which does not inherently possess tangible force can potentially acquire destructive capacity through the ability to instruct physical systems to malfunction. Cyber attacks on physical systems are correspondingly called *cyber-physical attacks*. The implications of this class of cyber attacks, namely the ability to inflict physical damage, is the main difference between cyber-physical and conventional cyber attacks.

What is not always understood is that breaking into a cyber-physical system and taking over its component(s) is not enough to achieve a desired physical impact. After all, breaking into a system is not the same as breaking a system. Executing successful cyber-physical attack requires a body of knowledge that is different from what is commonly employed in the information technology field. In particular, the attacker needs to understand signal processing [306], control principles [288], the physics of process behavior [211], the mechanics and failure conditions of the equipment [36, 188, 289], etc. Moreover, different types of CPS are subjected to fundamentally dissimilar failure modes the conditions of which first need to be discovered [36, 188, 289]. Figure 2.17 illustrates the distinction between the two constituents of cyber-physical attacks based on the attack objective. In the following sections we describe some intrinsic aspects of cyber-physical exploitation as well as the current threat landscape and state of threat actor abilities to execute cyber-physical attacks.

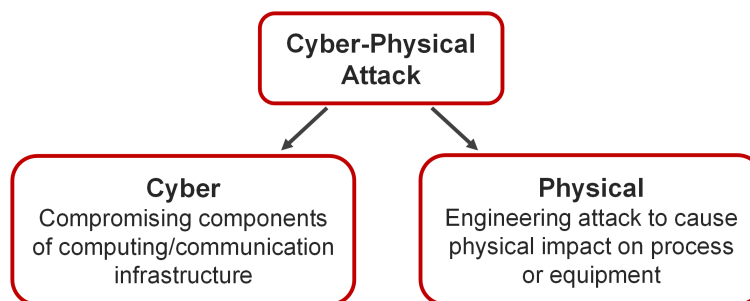


Figure 2.17: Multidisciplinary components of cyber-physical attack

2.3.1 Timing Parameter

Cyber attacks in the IT domain do not generally depend on timing aspects. Exceptions being instances such as race conditions, Time-of-Check to Time-of-Use vulnerabilities, or cross-site scripting attacks that rely on getting access to session cookies before they expire, where attackers need to make sure that their attacks occur within a tight window of time. In cyber-physical systems, however, timing plays a major role as the physical state of the system evolves continuously over time and some states might be more vulnerable to attacks than others. Timing also characterizes the sensitivity of a cyber-physical system, e.g., it may take minutes for a chemical reactor to burst [271], hours to heat a water tank or burn out a motor, and days to destroy centrifuges [187]. Understanding the timing parameters of the physical processes does not only allow an attacker to construct a successful attack but also to maximize the desired impact (e.g., long-term damage to the system).

The dynamic evolution of process variables can be described with a simple model consisting of process gain, dead time, and time constant. The process gain describes how much the process will respond to a change in controller output, while dead time and time constant describes how quickly the process will respond (Figure 2.18). Precisely, dead time describes how long it takes before a process begins to respond to a change in controller output, and the time constant describes how fast the process responds once it has begun changing. Controlling the processes with large time constants is a challenging task causing operator stress and fatigue [214]. The described timing parameters are not only important for the design of a control algorithm but also for the attacker to design an effective attack.

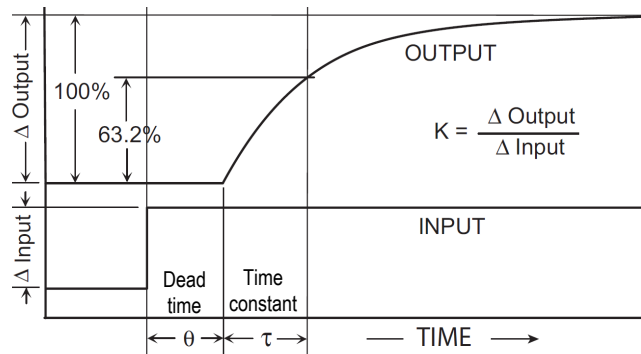


Figure 2.18: Time constants in process control, based on [288]

Attackers and researchers have shown numerous ways of compromising and controlling the digital systems involved in process control [1, 55, 94, 159, 275], hereby raising the possibility that catastrophic failures in the physical world can result from taking control of those digital systems. When an attack transitions from control of a digital system to control of a physical process, physics becomes the controlling factor instead of the digital rules encoded into the microcontroller.

A single bit flip can engage the burner under a tank of water, but the water will still take hours to heat regardless of the state of the controller outputs. Changing the state of the outputs does not immediately put the process into a vulnerable state. An attacker needs to take into account the timing and state of the system and act when the process is in the vulnerable state.

The key questions are when the process will reach the vulnerable state and how long it will take after the start of the attack to achieve the attacker's goals. Since the exact answers to these

questions may not be known to the attacker, there may be considerable uncertainty with regard to the timing of the attack. Feedback loops are capable of governing the process when it is operated within expected boundaries. However, as the process is pushed further away from the normal conditions, the effectiveness of the control loops becomes less known. Correspondingly, timing parameters become more uncertain and the process may enter an uncontrollable state. Moreover, the field instrumentation is calibrated to measure the process within certain predefined limits and may be inadequate to describe the process when it operates outside its configured region [270].

Consider a piping infrastructure in a plant. Fluid dynamics plays an important role in the exploitation of the piping systems. Pipes are typically designed to the maximum expected design pressure of the system, but a water hammer may amplify the system pressure by as much as six to ten, or more, times the original, intended design pressure and cause pipe rupture [197]. A classic water hammer or a hydraulic shock is caused when a fluid in motion is forced to stop or change direction suddenly, e.g., when a valve closes suddenly at an end of a pipeline system (Figure 2.19). The resulting pressure surge travels upstream at a sonic velocity causing pressure change in the pipe. Certain timing parameters such as speed and frequency of the valve closing, period and velocity of the wave propagation must be considered by an attacker while designing the damage scenario on the piping system.

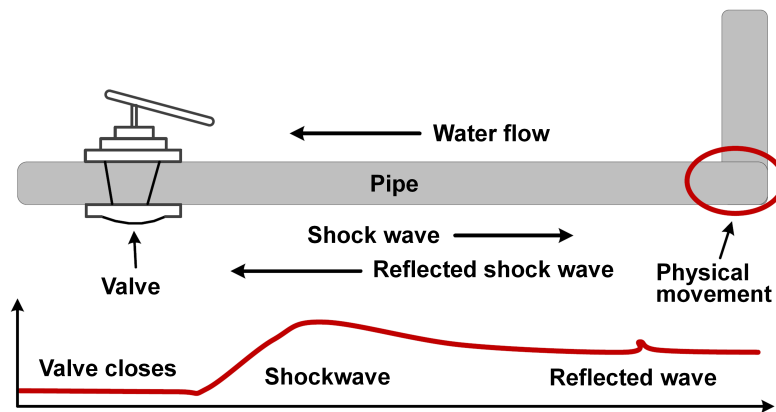


Figure 2.19: Water hammer

Every physical phenomenon in nature is best experienced in moderation and their extremes may yield disruptions and damages. Resonance and vibrations are common threads which run through almost every branch of physics and mechanics. Taken to an extreme, they occasionally cause a bridge to collapse, a helicopter to fall apart and other detrimental effects. Understanding the nature of their occurrence and susceptibility of the physical and mechanical systems to their exposure is the foundation for many classes and types of cyber-physical attacks [188]. The equations to compute conditions for the occurrence of unwanted forces and determination of their unsafe levels are available in the literature [197, 289]. However, the design and execution of such attacks in the cyberspace would require strategic and timely manipulation of the process variables.

To give an example of the above mentioned attack scenarios, we consider a catastrophic accident at Sayano-Shushenskaya Dam, Russia's Largest Hydroelectric Power Station in 2009 (75 killed) [136]. It was caused, among other reasons, by excessive vibrations which wore down the bolts that kept the turbine in place and by a consequent water hammer which ripped the turbine out of its seat. Additionally, another time-related parameter was involved in the accident. The working life defined by the manufacturer for the turbine was specified as 30 years. At the moment of the accident the age of the turbine was 29 years and 10 months. Vibrations and shocks degrade equipment robustness over time. Therefore the exploitation regime of an aged machinery must be

selected with care in order not to overstress it. Fragility of the equipment at the end of its designed exploitation time can be abused by the attacker.

2.3.2 Safety vs. Security

In the physical world safety is a primary concern. In the context of CPS, safety systems have the critical function of detecting dangerous or hazardous conditions and taking actions to prevent catastrophic consequences on the users and the environment. The industrial control community has substantial experience in identifying and addressing potential hazards and operational problems in terms of plant design and human error, used to minimize the effects of atypical situations and to achieve a safe outcome from a situation that could have resulted in a major accident. Even before somebody is allowed to visit a plant, they usually must watch a safety training video. Many industrial companies have a large screen displaying the number of days elapsed since the last safety accident.

However, the evolution of safety systems is largely built on the ability to interconnect systems and to automate notifications and alarms in the event of safety breaches. As a result, safety systems became vulnerable to cyber attacks. In the past the relationship between safety and security was studied in the context of dependable computing (Figure 2.20). Compared to previous work on determining common approaches to safety and security, which had its focus on IT or system-design, see, e.g., [199, 241], we include the underlying physical processes in our considerations.



Figure 2.20: Dependability and security attributes, based on [21]

Both cyber security and safety have distinct histories and have developed their own bodies of work. In both disciplines basic concepts have developed into a language that can be used to describe best practices. However, the current efforts to secure critical infrastructures have used the language of cyber security drawing little from the language of safety. Architectures are most often described in terms of security boundaries and not in terms of hazards. This cyber-oriented view of the world has been codified into standards and regulations governing process control.

One regulation illustrating this point are the NERC CIP standards [239]. Under this regulation a control system is broken down into a set of “control centers”. The communications between control centers and to outside entities defines an electronic security perimeter (ESP). Not all control centers are required to be defended. Simple tests are used to determine whether a particular control center is required to be defended to achieve compliance with the standard. However, most of the tests are cyber-oriented. The only safety-oriented test is that the control system should have the ability to shed 300 MW of load. All other hazards such as bringing a generator out of phase [327] or energizing a line during maintenance work are not considered by the standard.

The NIST 800-53a standard has a similar flavor [152]. Its general hardening recommendations such as password lengths are applied broadly to the devices used in process control. The standard is meant to be applied to all industrial processes without any additional considerations for the specific product being produced or manufactured.

In both cases, there is no need for the implementer to understand the inherent hazards of the system. Hazards are simply seen as part of the nameless devices residing at the lowest level of the control systems architecture, whereas cyber security controls are implemented as a barrier on top of these devices, frequently in the form of firewalls. The danger of this line of thinking is that all parts of the process are often grouped together into a single security compartment without any regard to how the parts of the process interact with each other and, specifically, how these interactions are being key factors to the safety of process

Safety and security are sometimes described as two sides of the same coin [72]. If the attacker can compromise safety systems through cyberspace and prevent them from performing their intended protection function, a security incident may lead directly to a catastrophic event. Security and safety are interconnected but both have different missions and employ different vocabularies. In order to understand their “separation of concerns”, in the simplest way, the relationship between security and safety can be explained with the diagram shown in Figure 2.21. If the attacker manages to breach a security perimeter and gains access to the plant infrastructure, there are still several layers of safety protection in place to prevent harm to the process, equipment, personnel and environment.

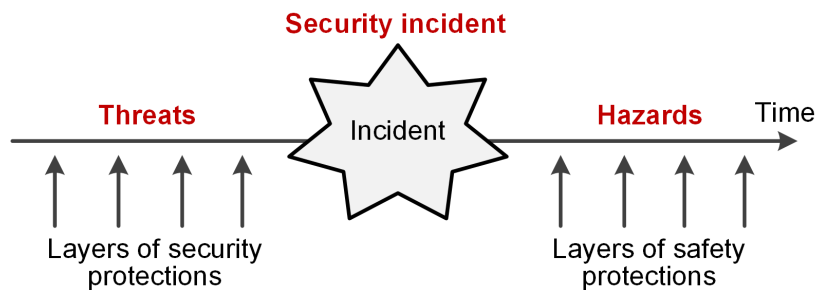


Figure 2.21: Relationships between security and safety

From a safety point of view, any device that can fail is an additional risk. A firewall might physically catch fire, or it might be misconfigured and suppress important messages. As emergency cases are rare, uncommon messages will be sent especially often in cases when they are important. Message encryption makes it harder to monitor the communication channels. Message authentication failures might refuse valid messages. Key management may introduce further potential points of failure.

Safety and security are related and require similar mindsets, e.g., investing resources for something not to happen, but there are also some fundamental differences in the way of thinking. In a safety system, failures are usually assumed to be accidental. Failure probabilities of individual components are thus by and large independent. If the probability of an engine failure in an airplane is 1 in 100.000 flights, the probability of two engines failing is 10^{-12} ; this is then the failure probability for a plane that can land safely with one engine. In security, failures are not independent. An attacker attempting to manipulate one device is likely to also attack the fallback. An attacker’s next step may *depend* on what has been observed so far. Events deemed so unlikely that they are simply not accounted for, such as a name containing valid SQL commands, may be triggered intentionally by an attacker.

Finally, security and safety have opposing update regimes. IT security employs *frequent updates* such as installing patches, upgrading firmware or adding new firewall rules to react to new *security advisories*. On the other hand, an engineering system once approved is expected to *remain safe if left untouched*. Any change in software or operational practices must be followed by an extensive

safety revision. Failing to do so can result in casualties. For example, after an update of the SAP-based maintenance software at DuPont without a subsequent safety review, an alarm notifying on a due date for a hose change disappeared. As a result, a hose used to transfer phosgene from a cylinder to a process wore out and catastrophically failed spraying a worker in the face resulting in his death [273].

With the increasing digitalization of the safety systems and their interconnectivity with other systems such DCS and historians, there is a growing risk for these systems to be compromised and not being able to provide their protective function. This will be further discussed in the next section.

2.4 Industrial Control System Threat Landscape

In the IT domain, the ultimate objective of the attackers is often to obtain data (e.g., intellectual property or specific records). Misuses in the OT domain are much more concerning as the attackers' goal is to cause an impact in the physical world. The concerns are exacerbated by the fact that most control equipment, systems and protocols either lack fundamental security controls or do not have them properly configured. As compensating security measure, it is possible to segregate OT equipment from the high-risk networks and Internet via the OT DMZ. However, many industrial environments either do not have this network layer or have it implemented with very limited capability. As a result, susceptibility of the OT zone to human-assisted or malware compromise remains high. Figure 2.22 shows the timeline of publicly-known attacks which had manifested in physical consequences. We added two targeted reconnaissance campaigns because their activities were directed at collecting information related to execution of cyber-physical attacks.

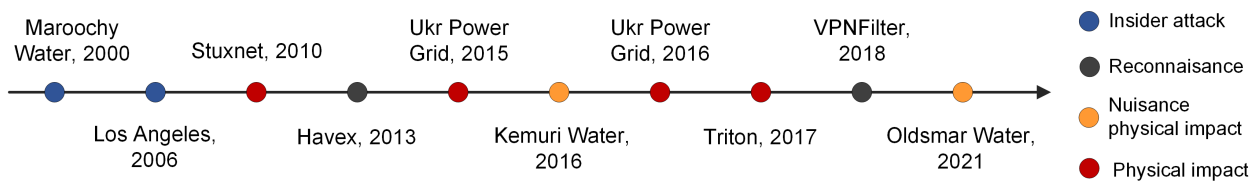


Figure 2.22: Historical time of cyber-physical attacks

In general, the concept of control systems misuse to achieve specific physical impact is not novel. Before becoming a target of external threat actors, control systems were subject to insider attacks. Among the most frequently discussed cases is an attack on the Maroochy Shire sewage treatment process in Queensland, Australia in 2000. A disgruntled contractor who had installed the radio-controlled waste management system, used commercial radios to interact with control equipment and caused millions of litres of raw sewage to spill out into local parks and rivers, contaminating creek water and causing death to local marine life [287]. In 2006 two traffic engineers in Los Angeles remotely accessed the city's traffic control system and changed the duration of the red light at four busy intersections located near freeways and major destinations in Los Angeles as part of a labor union protest. They also inserted a code that prevented an easy fix of the hack. It took a few days to resolve the traffic gridlock caused [122]. As can be noticed from these two examples, insider attacks can be especially damaging thanks to intimate knowledge of the systems possessed by the insider employees.

The epoch of external targeted attacks on control systems started with Stuxnet, malware that was designed to limit the pace of the Iranian uranium enrichment capabilities [98]. Among the

distinctive features of Stuxnet is the amount of multi-national efforts invested into its development [330] which makes it a special case not only in the ICS space but also in the overall history of cyber security. The cyber-physical payload of Stuxnet was implemented in two distinctive variants [187]. The first payload was designed to over-pressurize centrifuges by keeping exhaust valves within the centrifuge cascades closed. This attack scenario turned to be not sufficiently reliable. The second payload variant targeted the centrifuges' rotor speeds to over-speed centrifuge rotors and to take them through their critical (resonance) speeds/frequencies. While Stuxnet did not significantly set back the Iranian program due to its premature discovery, it opened the Pandora box of cyber-physical attacks and prompted other nation state threat actors to ramp up their cyber activities in the ICS space.

Not long after the Stuxnet discovery, industrial organizations primarily in the USA and Europe were targeted with a malware campaign which hypothetically could have been a reconnaissance stage preceding a Stuxnet-like sabotage attack. A malware named Havex was designed to search for OPC servers and collect information about connected field instrumentation [148]. While no malicious code targeted at process manipulation was discovered, with the obtained process configuration data such capability could have been added to Havex with just a few lines of code. In another instance, an attack campaign targeted a wide range of networking devices in 2018 included components for detecting and logging Modbus TCP traffic [294]. It is not uncommon to relay industrial traffic over the Internet without protection [236]; the attackers used this knowledge to identify sources and destination of such traffic together with process data payloads.

In the safety discipline a *near miss* is an event not causing harm, but having the potential to cause injury or ill health [146]. Two cyber-physical attacks in 2016 and 2021, both on water utilities, could have potentially caused significant harm to the population but resulted only in nuisances. Partly due to attacker "incompetences" and partly due to the safety measures in place [96, 201, 313]. In both cases threat actors exploited weaknesses in the Internet-facing systems, obtained remote access to the control applications and modified the ratios of chemical components entering the water treatment process. In both cases process manipulations were promptly identified and reverted to normal values. While there was no danger to public safety due to strict water quality controls before water leaves for public use, the attackers could cause damage to process equipment through chemical contamination. The amateur and naive way of conducting these attacks suggests opportunistic attackers who were not driven by a deterministic objective of achieving specific outcome.

Although opportunistic attacks may potentially result in harmful outcomes, high-precision attack are much more concerning. Besides Stuxnet, there were three other targeted cyber-physical attacks with physical consequences in past years, all happening in a short period between 2015-2017:

- Attacks on three power substations in Ukraine, 2015; malware family – BlackEnergy3 [91];

On December 23rd, 2015 a major coordinated attack on electrical utilities in three different regions in Ukraine left hundreds of thousands of inhabitants without electricity for about six hours in total. The attackers compromised remote access to the operator console in the OT network, and used HMI software to open circuit breakers. Simultaneously, in one of the regional utilities, the attackers updated Serial-to-Ethernet converters with malicious firmware to make them inoperable. This ensured that even if the operator workstations were recovered, it would not be possible to re-close the breakers in the affected substations remotely, requiring manual remediation. Only thanks to the timely restoration of power supply the attack did not result in casualties or other significant consequences.

- Attack on a power substation in Ukraine, 2016; malware family – Industroyer [329];

Just before midnight on December 17th, 2016 a 330kV substation north of the Ukrainian capital Kiev went offline, affecting the population in a part of Kiev and a surrounding area. Forensic investigation has discovered malicious software on several SCADA servers which was used to disconnect circuit breakers via scripted networking commands. In contrast to the attack in 2015, the final subversion activities were not executed manually by human threat actors but instead scripted into an autonomous malware framework. As during the previous incident, the power supply was manually restored in timely fashion, avoiding serious consequences.

- Attack on Safety Instrumented System in a refinery in Saudi Arabia, 2017; malware family – TRITON [159].

While both attacks on the Ukrainian power grid could have potentially resulted in significant impact on the civilian population via collateral consequences, the TRITON attack on the Triconix safety controller directly undermined the safety of human workers in the victim facility. TRITON is a passive software implant with read, write and execute capabilities, residing in the controller’s memory and activated by a special network packet. While TRITON was discovered at the stage of its installation, once successfully installed the implant allows to remotely reprogram safety system with an attacker-defined payload. In the worst-case scenario the TRITON capability could be extended to controlling I/O functions and denying safety shutdown functionality during hazardous events. Figure 2.23 shows the block diagram of the 3008 Triconex electronic board with main and I/O processors located on the same board and communicating with each other via shared memory. With this hardware design, passing commands to safety instrumentation could be implemented with minimal efforts.

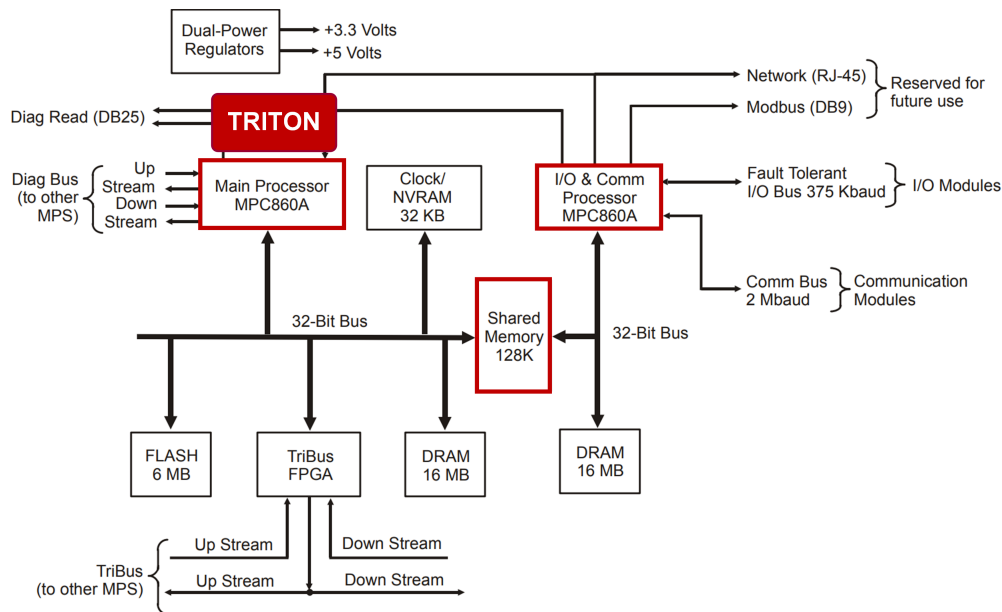


Figure 2.23: Architecture of a model 3008 main processor [60]

When mapping the attacks discussed above against a traditional layered representation of an ICS network, it becomes apparent that cyber-physical exploitation tactics are following an evolutionary path, clearly picturing the ongoing “race-to-the-bottom” trend between ICS threat actors and defenders. This trend refers to the tendency of the attackers to move their exploits one

layer down as soon as security controls are introduced at some layer of the computer or network architecture. While OT asset owners begin to harden operator consoles and embrace ICS network monitoring solutions, the attackers are already moving their exploits into the control equipment as illustrated in Figure 2.24.

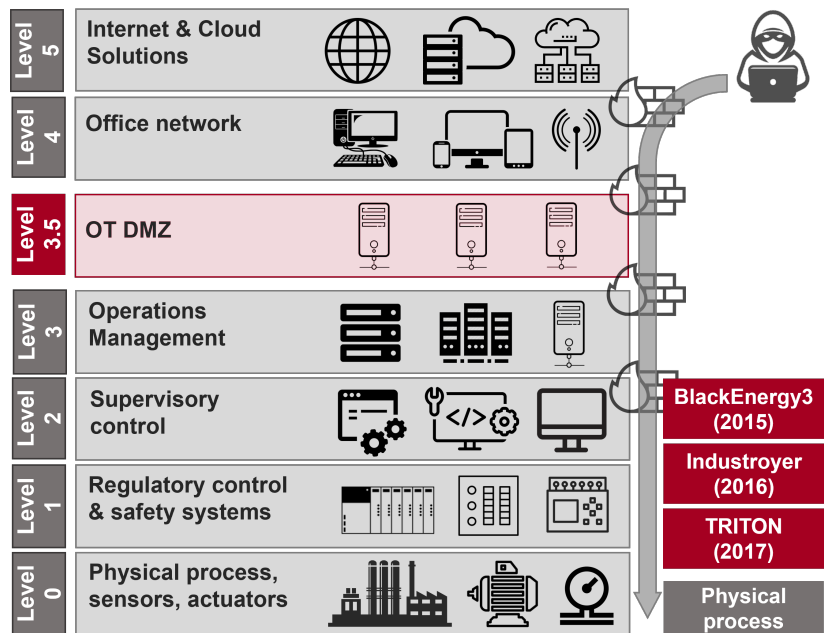


Figure 2.24: “Race-to-the-Bottom” exploitation trend in Industrial Control Systems

In the 2015 attack on the Ukrainian power grid, the attackers leveraged unauthorized access to the HMI via remote access software to open the breakers and de-energize the substation. With that, the attack was executed at the level of supervisory control by compromising more familiar Windows-based IT systems. However, executing ICS attacks at this level is less reliable due to readily-available security controls for Windows systems. In the 2016 power grid incident the attackers launched their exploits at the industrial protocol layer. Before the 2015-2016 events very few industrial organizations invested in visibility solutions for their control networks. This is why Industroyer activities were not detected at the time of the attack. With increased awareness of security risks and OT network monitoring solutions readily available on the market, investing in offensive capabilities at the level of control networks has become less practical. In the 2017 TRITON attack, the attacker moved their exploit all the way into the control equipment at the regulatory control level. The reason for this strategy is the lack of exploit mitigation and detection capabilities in most of the embedded components deployed within the ICS and a lack of tools to support compromise assessment and forensic analysis of these systems. Therefore, in the foreseeable future embedded exploits are likely to remain undetected for an extended period. Also, the lack of international regulations in using cyber warfare against critical and vital infrastructures does not discourage the execution of cyber-physical attacks, further motivating threat actors to continue with the race.

It is not uncommon for sensing devices and even actuators to be accessible either physically or via exposed communication channels. Examples include wireless sensing devices located too close to the physical fence of an industrial facility or devices placed in accessible places. The attackers can take control over such end-point devices and use their connectivity to a wider network to compromise systems located in the upper layers of the Purdue reference architecture. This control-systems specific exploitation scenario is sometimes called “hacking upstream”. In one instance

the attackers were able to exfiltrate a large volume of data from a casino by compromising an Internet-connected thermostat in the casino's aquarium [217]. In another example, the researcher demonstrated how the attacker can trigger software vulnerabilities in an ERP system such as SAP in a multistage attack which starts with compromising HART transmitters [29]. Industrial facilities in remote locations which rely on radio transmission lines are especially susceptible to such attack scenarios [209].

Every industrial environment is unique. Creating customized payloads is laborious and not scalable. It is natural for the attackers to search for (1) design payloads that require little proprietary knowledge about the target environment (e.g., configuration parameters, understanding of the process data), (2) reusable cyber-physical attack scenarios. The second variant of the Stuxnet payload is an early example of a reusable damage scenario. Its payload targets a common weakness of all rotating equipment: namely, equipment overstress through operation at harmful frequencies. With minor modification, this payload can be re-used against other types of rotating equipment such as motors or turbines. The Industroyer malware is among the more recent examples of re-usable cyber-physical payloads. Its attack logic is relatively simple: find all or selected controllable signals on RTUs and OPC servers and change their status to off. The attackers used publicly available protocol specification documentation to discover the needed points in the control equipment instead of relying on the substation's configuration information, which makes the Industroyer attack scenario applicable to any substation worldwide. To maximize their success chances, the attackers implemented attack logic in four different modules for four widely used industrial protocols in the electrical sector as shown in Figure 2.25. A launching module was pre-programmed to execute all four payloads simultaneously at the predefined time. This success maximization strategy can be compared to a heap spray attack on computer systems. The Industroyer attack framework is also extensible in that it can be easily enhanced to support additional protocols.

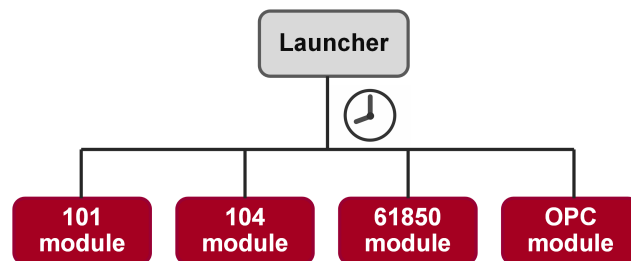


Figure 2.25: Industroyer payload components

Previously, limited access to industrial equipment, software and proprietary engineering documentation were thought to be the main barrier to entering the ICS exploitation field. The situation has changed significantly over the past decade. If previously one would need to be a legitimate buyer to buy certain types of industrial equipment such as safety systems, nowadays such equipment and related software can be freely purchased at e-commerce platforms such as eBay [92] or Ali Baba [129]. Also restricted and proprietary documentation such as various engineering documentation, equipment manuals and development guides, specification sheets, trainings, purchasing documentation and others are frequently uploaded to public file repositories such as Scribd [151]. Industrial software executables, firmware for industrial equipment, control logic, configuration files are often uploaded to Virus Total [304] and GitHub [150]. It is also not uncommon when engineers and subcontractors expose sensitive documentation via improperly configured file servers [235].

To make use of the vulnerabilities discovered in the ICS equipment, they need to be weaponized. There is a growing effort in the security community to produce exploits for publicly disclosed

ICS vulnerabilities for both research and commercial purposes. This includes stand alone exploits, e.g., as Metasploit modules [255] and commercial frameworks such as SCADA+ Pack [119], which includes both public and 0-days vulnerabilities in one exploit pack. To close the gap in lacking tools for ICS exploitation, researchers and security practitioners developed repositories with ICS attack instruments, e.g., [77, 242, 279, 296]. Additionally, a significant effort was dedicated to developing password bruteforces and creating libraries of the default/commonly used credentials for industrial equipment [76, 78, 276, 302]. As anything publicly available, such repositories of attack tools lower the bar for threat actors with an interest in ICS exploitation. It is worth mentioning that it took only a few months until a detailed technical analysis of Industroyer [55] became publicly available and just a couple of weeks for the TRITON attack framework to be uploaded to the GitHub repository after incident disclosure [88]. Although pulling off successful cyber-physical attacks still requires a significant amount of specialized skills and expertise, the accessibility of tools and exploit modules is making it easier for threat actors with limited resources to bridge their knowledge gap.

Although ransomware became a prominent threat to ICS environments in the past few years [246], we do not consider this threat in this thesis because its impact on the physical process is either collateral, caused by poor or missing segmentation between OT and IT networks [35], or tailored to increase the chance of receiving the ransom by stopping some critical monitoring and controls services [37]. We limit our scope to targeted cyber-physical threats directed at causing a pre-defined impact in the physical world.

2.5 Conclusion

With the introduction of computer-aided manufacturing, significant effort was put into the standardization of industrial architectures and communication flows resulting in concepts like the automation pyramid and the Purdue reference architecture. With the growing business need to exchange data with enterprise systems and users as well as provide remote access to complex pieces of machinery, the industry has focused on finding best practices for securing automation systems from potential external threats and avoiding any disruptions in control systems functioning. However, being a conservative industry, the adaptation of security practices in automation has been much slower than the desire of the business for the better economic performance of the plants, resulting in increasing discrepancies between technological advancements such as the usage of cloud technologies and efforts to identify and mitigate associated security risks.

The Industrial Control Systems threat landscape has changed dramatically over the past few years. New threats have emerged, exacerbating global concerns brought about by Stuxnet, the first known cyber-physical attack. Although modernization and wider connectivity introduced opportunities for remote exploitation of control systems, attempts to disrupt a process without clearly understanding the consequences of the attack actions on the physical process are likely to result in a minor nuisance instead of an actual disruption. In order to achieve the desired cyber-physical effect, the attacker requires a body of knowledge that is different from what is commonly employed in the IT field. For instance, it takes a specially-crafted attack to bring the process into a needed vulnerable state or the attacker needs to wait for a process to be in a particular state before executing her attacks. In the pre-Stuxnet era, ICS exploitation was considered a boutique skill mostly possessed by state-sponsored threat actors. However, the previous status quo has been challenged over the past decade thanks to wider accessibility of OT equipment, open-source attack tools, and information, allowing less resourceful threat actors to participate in the ICS domain in support of economic, political and military interests.

Chapter 3

Damn Vulnerable Chemical Process Framework

The secret to getting ahead is getting started.

Mark Twain

3.1 Introduction

One of the challenges of cyber-physical security research is the lack of realistic large-scale testbeds for studying complex cyber attacks and their effects on physical processes. Well-known realistic testbeds are Secure Water Treatment (SWaT) [218] and Electric Power and Intelligent Control (EPIC) [267], hosted by the iTRUST research center at the Singapore University of Design and Technology (SUTD). These testbeds were conceived, designed and built collaboratively by SUTD faculty members, international consultants, and engineers from public utilities in Singapore. Each testbed mimics a smaller in scale realistic industrial process and is built from popular models and brands of industrial equipment. The high fidelity of these experimental environments is certainly very beneficial to multidisciplinary researchers when conducting cyber-physical security experimentation. However, building and maintaining such testbeds requires specialized personnel and significant financial investments. Thus, at the time of writing, external users were charged an hourly rate of \$900 (\$400 for internal students) for testbed usage [155]. Provided that scholarly research requires extensive experimentation for statistical analysis and discovering edge-cases, research on a realistic testbed can prove economically prohibitive. Going further, additional repair expenses are inevitable if the testbed is damaged as a result of a successful experiment aimed at physical disruption. Additionally, unsafe experiments are not permitted by safety regulations. This means that testing on the realistic testbeds is limited to a set of well-calculated experimental use cases, which do not jeopardize the safety and physical integrity of the equipment. Clearly, this restriction imposes further significant limitations when conducting research on realistic testbeds. Out of these considerations, a more affordable and safe way of conducting cyber-physical security experimentation is using simulated process models.

In the process engineering domain, developing models of physical processes is an integral part of a cost-effective R&D process. Process simulation is a model-based representation of chemical and other processes in software, intending to optimize designs and operating procedures of plants and individual units. Process simulation software has long been instrumental to industrial and academic research groups in aiding “trial and error” experimental approaches. Also, models allow for close

collaboration between chemical, control and process engineers. While the chemical engineers model complex thermodynamic interactions within the plant, control and process engineers devise suitable control algorithms and plantwide control schemes and optimize equipment design. The ability of the control system to support plant chemistry is then tested in closed-loop simulations.

Developing process models is time-consuming and costly, therefore the majority is kept proprietary. Conceivably, running complex process simulations requires highly-specialized software, which is also often kept proprietary or available with expensive licenses. Out of these considerations, several open-source models of realistic industrial plants were proposed in the process control community (e.g., [11, 25, 51, 87]) to focus on industry-relevant problems and to allow the comparison of research results. The research question we pursue is whether these models could be adapted to study the various aspects of cyber-physical system security.

In this thesis we consider two public process models, *Tennessee Eastman* and *Vinyl Acetate Monomer*. We chose these models for their size and complexity. Both models are large-scale nonlinear dynamic models, which can be simulated in Matlab [219]. While Matlab is not an open-source software, it is one of the most popular software environments for engineers and scientists, and it is more affordable than other commercial dynamic process simulators.

In 1993 Down and Vogel published a model of a Tennessee Eastman (TE) industrial chemical process for the purpose of developing, studying and evaluating process control technologies [87]. For many years their model has proved to be beneficial to the process control community serving as a realistic check on the industrial and practical relevance of novel process control solutions. Many publications have appeared about the TE process which has become one of the most popular research process models.

One disadvantage of the TE model is that the details about chemical reactions and involved equipment are omitted. The dynamic behavior of the plant was provided in the form of a simulation in Fortran code accompanied by a flowsheet, a steady state material balance, and a qualitative description of the key process characteristics. As a result, this implementation does not allow studying *specific* research questions related to the modeled process. To address the continued interest among researchers to have additional industrial benchmark processes, Luyben and Tyreus published details of an industrial process for the production of Vinyl Acetate (VAC) also known as Vinyl Acetate Monomer (VAM) in 1997 [210]. This model involves real non-ideal chemical components, a realistically large process flow sheet and consists of several standard unit operations that are typical of many chemical plants with the recycle stream and energy integration. In this regard, the VAC process model is a step ahead compared to the TE process model.

In this chapter we describe how we enhanced the models to support cyber-physical security experimentation and which research questions can be effectively studied with process simulators. Consequently, in the following chapters, we show how we used the proposed experimental framework in our research.

3.2 Framework Description

Damn Vulnerable Chemical Process (DVCP) is an open-source framework developed for cyber-physical security experimentation based on the two above mentioned models of chemical plants [179, 180]. The naming of the framework follows the naming of two well-known “damn vulnerable” frameworks from the IT security domain, namely Damn Vulnerable Linux (DVL) [277] and Damn Vulnerable Web Application (DVWA) [295]. These frameworks are intentionally vulnerable Linux distribution and web applications, created for the purpose of studying a variety of security topics and legally practicing exploitation techniques in a controlled environment.

Since the control strategies for both plant models were designed without considering any

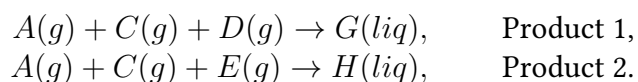
potential malicious interaction with the plant, it makes them a suitable test case for researching what it takes to convert a cyber attack into a successful cyber-physical attack. The frameworks can be utilized for Red-Team (offensive) type of research questions such as the discovery of novel attack vectors, designing individual attack instances and complex coordinated attacks. Consequently the “defenders” can study the resilience of processes to cyber attacks, develop and evaluate risk assessment methods, robust control algorithms, attack detecting techniques, process-aware authentication methods, etc. The framework can be used as a standalone experimental environment or as a physical layer in a distributed industrial control system testbed [39].

3.2.1 Tennessee Eastman Process

The Tennessee Eastman (TE) test problem is an industry-based chemical process (the entire plant) from the Eastman Chemical Company, where the components, kinetics and operating conditions have been modified for proprietary reasons. This process was first proposed in 1990 at the annual meeting of the American Institute of Chemical Engineers (AIChE) and it remains among the most frequently studied chemical processes to date.

Process description

The plant produces two liquid (*liq*) products from four fresh gaseous (*g*) reactants involving two irreversible exothermic reactions composed of chemicals *A*, *C*, *D*, *E*, *G* and *H*:



The process has five major operation units: the reactor, the product condenser, a vapor-liquid separator, a recycle compressor and a product stripper as shown in Figure 3.1. The gaseous reactants and products are not specifically identified. Feed *C* is not pure and consists of 48.5% *A* and 51% *C*.

The gas phase reactions are catalyzed by a substance dissolved in the liquid phase in the reactor. The reactor is pressurized and relies on an internal cooling system to remove the heat produced by the reactions. The byproducts and inerts are purged from the system in the vapor phase using a vapor-liquid separator, products *G* and *H* exit the stripper base and are separated in a downstream refining section. The plant has 12 degrees of freedom (valves) for manipulation. In total 41 measurements are involved in process monitoring. The first 22 measurements are continuous, the rest are sampled composition analyses from chromatographs with a delay of 0.1 or 0.25 hours, depending on the process variable.

The system can be operated in six distinctive modes which are determined by *G/H* mass ratios. Mode 1 is a base case with $G/H = 50/50$. The goal of the plant operation is to maintain desired production rates and product composition within $\pm 5\text{mol}\%$ while keeping other variables within specified operational limits. The process control goal is to minimize variability and absorb disturbances. If the process exceeds its safety limits, it automatically shuts down. The initial problem statement does not give recommendations about how the plant should be controlled. Instead control engineers are challenged to come up with their own control strategies, e.g., [193, 211, 222, 261]. The original TE process has more degrees of freedom (valves) than necessary for control and the engineers are free to decide which ones to utilize. The resulting control structures are usually designed to meet a specific control objective, e.g., optimal steady state, maximum rejection of process disturbances, ensuring on-demand production rate, adapting to an on-supply reactants rate or a combination of few objectives. Optimization of the control strategy is typically subject to several constraints and an optimal solution is not always feasible.

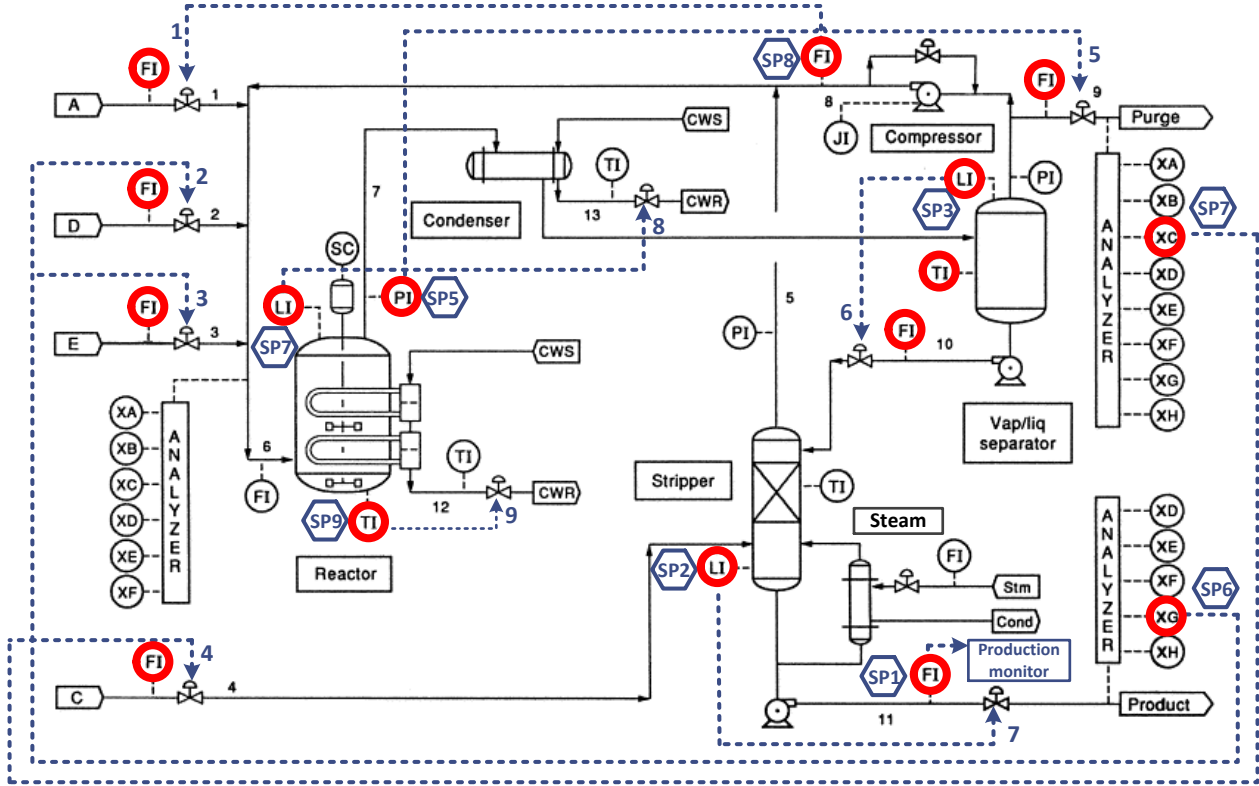


Figure 3.1: Plantwide chemical process under control–based on [193]

Matlab/Simulink Model Description

For our experiments we selected the TE Matlab model developed by Ricker [259] as its implementation was well documented. The model utilizes the self-optimizing control strategy proposed in [193]. The simulated control scheme consists of 18 proportional-integral (PI) controllers, 16 process measurements denoted as $XMEAS\{1-5;7-12;14;15;17;31;40\}$, nine valves denoted as $XMVs\{1-4;6-8;10;11\}$ (three valve positions are constant), and nine setpoints which form eight multivariable control loops and one single feedback control loop as specified in Table 3.1. There are two auxiliary control loops for improved management of the production rate. The first loop generates a contributory control value which is used in the calculations of the $XMV\{1-4;6;7\}$. A second auxiliary loop is used to calculate additional control values for D - and E -feed rates, depending on the G mol% in the product flow (stream 11).

The full notation and units of process characteristics, including operation constraints, can be found in [87]. The default simulation time of a single experiment is 72 hours. There are two metrics readily available to evaluate the result of plant operations: product quality defined as G mol% in the product flow and operating costs. Additionally, it is possible to measure time to plant shutdown due to exceeding safety constraints. Each simulation run generates in total 53 plots: 41 $XMEAS$, 11 XMV and operating costs. Additionally, a real-time production monitor is available.

The process description includes a flowsheet, steady state material and energy balance as well as the operational constraints of both the optimal steady state operations and the process shutdown limits (safety limits) [87]. Depending on her goal (economic impact or physical disruption), the attacker would have to violate these specified constraints.

By default, the model operates in a steady state (without disturbances). The original problem statement includes typical set points and load changes, which along with other listed disturbances

MV	Valve	Variable 1	Variable 2
XMV(1)	D-feed rate	% G in product	FI (stream 2)
XMV(2)	E-feed rate	% G in product	FI (stream 3)
XMV(3)	A-feed rate	Recycling rate	FI (stream 1)
XMV(4)	C-feed rate	% C in purge	FI (stream 4)
XMV(5)	Compressor (recycle)	Constant	Constant
XMV(6)	Purge flow rate	Reactor pressure	FI (stream 9)
XMV(7)	Separator underflow	Separator level	FI (stream 10)
XMV(8)	Stripper underflow	Stripper level	FI (stream 11)
XMV(9)	Steam feed rate	Constant	Constant
XMV(10)	Reactor cooler	Reactor temperature	—
XMV(11)	Condenser cooler	Reactor level	TI (stream 13)
XMV(12)	Agitator speed	Constant	Constant

Table 3.1: Valves and measured variables

Process variable	Normal operating limits		Shutdown limits	
	Low limit	High limit	Low limit	High limit
Reactor pressure	none	2895 kPa	none	3000 kPa
Reactor level	50% (11.8m ³)	100% (21.3m ³)	2.0m ³	24.0m ³
Reactor temperature	none	150°C	none	175°C
Product separator level	30% (3.3m ³)	100% (9.0m ³)	1.0m ³	12.0m ³
Stripper base level	30% (3.5m ³)	100% (6.6m ³)	1.0m ³	8.0m ³

Table 3.2: Process operating constraints [87]

illustrate different aspects of TE process operations. From the attacker point of view, the most interesting process information is the operation constraints presented in Table 3.2. In total there are 20 disturbance modes IDV{1-20} and four set point changes which can be selectively included in the simulation. This information is valuable to the attacker as any change in operations causes variations in process behavior which in turn is reflected in the process measurements (sensor signals).

All process measurements include Gaussian noise with standard deviation typical of the measurement type. Depending on the noise level and shape of the signal, sensor signals in the TE process can be roughly divided into four distinct groups (Figure 3.2). Type 1 is characterized by few distinct peaks and low noise level. Type 2 is distinguished by multiple noisy signal peaks. Type 3 can be described as a very noisy variation of Type 1 signals. The Type 4 signals distinguish themselves by the overall slow signal amplitude change with high amplitude noise. Depending on the process state, the noise floor of sensor signals may differ substantially. For instance, A feed flow F_A is of Type 4 in a steady state, of Type 2 under disturbance IDV(11) and of Type 1 with disturbance IDV(8) active. This information might not be known to the attacker upfront and contributes to her uncertainty.

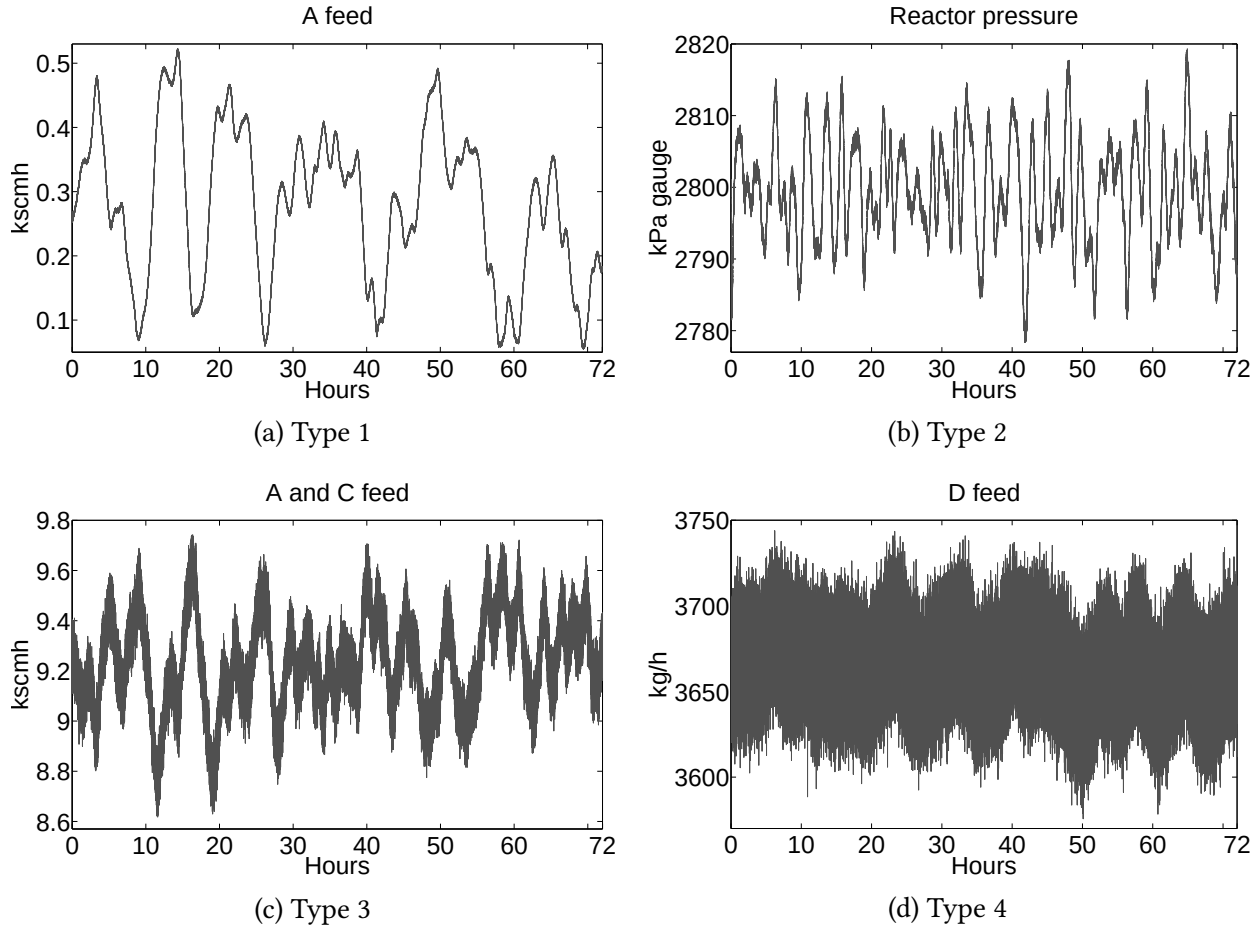


Figure 3.2: Different types of sensor signals in the TE process

Model Enhancements to Support Cyber-Physical Security Research

To begin with, we redesigned the initial model of the TE process by Ricker [259] to include simulation of typical attacks classes such as data integrity and Denial of Service (DoS) attacks on the process variables (sensor signals) and the manipulated variables (controller output to actuators). Instrumentation of the TE Simulink model with attack blocks is shown in Figure 3.3. Note that the Simulink model is multi-layered and care must be taken to ensure correct implementation and propagation of the attack parameters through the hierarchy of the layers. A functional implementation of the integrity attack on sensor signal in an individual controller is illustrated in Figure 3.4. We also implemented output and storage of the attack data in the work space for further analysis and visualization. Additionally, we enhanced the Simulink model with a graphical user interface for a convenient setup of the desired attack parameters. For instance, for an integrity attack on sensor signals, the user can choose attack value, attack time and its duration (predefined or random) as well as the frequency of the attacks (single or periodic). We also implemented an option of feeding a predefined sensor values stream from a file. The user may run several attacks in parallel or sequentially (chain several attacks).

We made a few further improvements to the Simulink model of the process to enhance the statistical evaluations of the experiments. The initial model did not allow any randomness in the simulations to guarantee repeatability of the disturbances to plant operations. Therefore, each

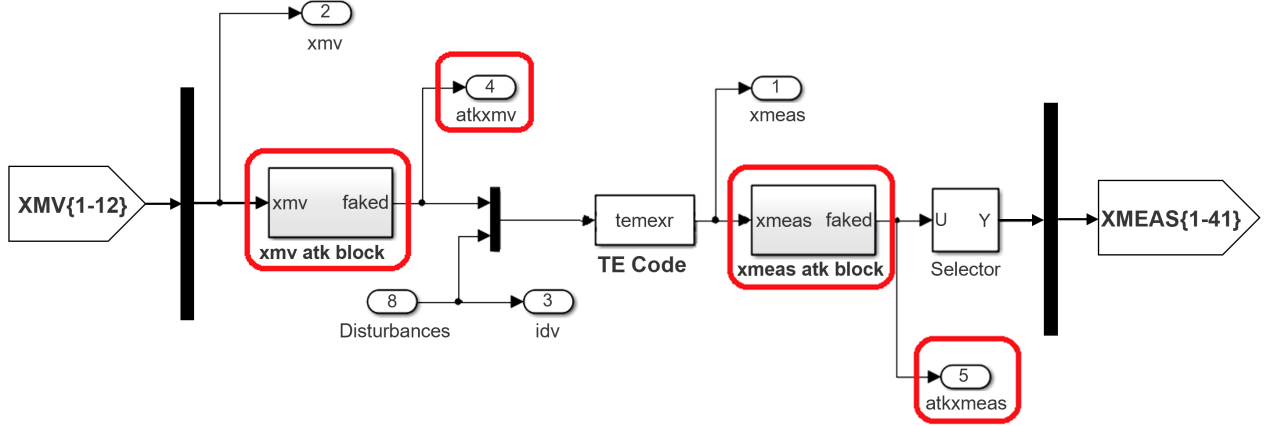


Figure 3.3: TE Simulink model enhanced with attack blocks

simulation run produced identical results. It was a significant disadvantage of the TE model as it was not possible to statistically evaluate the impact of the attack and defense strategies. We modified the original code by generating a new seed for the random number generator for each simulation run while preserving the underlying dynamics of process behavior. If needed, the user can switch off the randomness or set a specific seed. Also, in the original implementation, the process data is downsampled to 100 samples per hour (s/h) before being stored in the Matlab workspace. Downsampling of sensor signals significantly changes the noise profile of raw signals. Therefore, we implemented the output of the raw sensor data with a sampling rate of $f_s=2000$ s/h which is used during the actual real time simulation runs. Because the sampling frequency of sensor signals has significant impact on the attacker strategies, we implemented the possibility to set up a custom f_s value.

Evaluation metrics are indispensable for measuring any activity. In process control, optimizing operating costs is among the prime goals along ensuring safety. Therefore, we introduced an additional metric to measure attack success from the monetary standpoint, which is described below. The economical performance of the TE plant can be estimated twofold. Firstly, via operating costs, expressed in $\$/h$ and calculated as [87]:

$$\begin{aligned} & (\text{purge costs})(\text{purge rate}) + (\text{product stream costs})(\text{product rate}) + \\ & (\text{compressor costs})(\text{compressor work}) + (\text{steam costs})(\text{steam rate}) = \text{total costs.} \end{aligned} \quad (3.1)$$

Operational costs are primarily determined by the loss of raw materials in the purge stream. The purge rate has the greatest impact on cost due to material losses (cost of chemicals being purged) and costs of running the compressor, which cycles non-condensed components back to the reactor feed. Any attack which has an impact on the purge flow rate has a proportional impact on the production costs.

Another economic indicator is product quality $G \text{ mol}\%$ in the product flow. Extensive experimentation has shown that very few attacks have an impact on product composition that could be considered as harmful. This indicates that the control configuration implemented in the TE Matlab model is robust against disturbances that could impact required product composition. For this reasons, we introduced an additional metric – *product throughput* – to estimate the influence of an attack on the final product throughput:

$$F_{strip} m^3/h \times G \text{ mol}\% = G m^3/h. \quad (3.2)$$

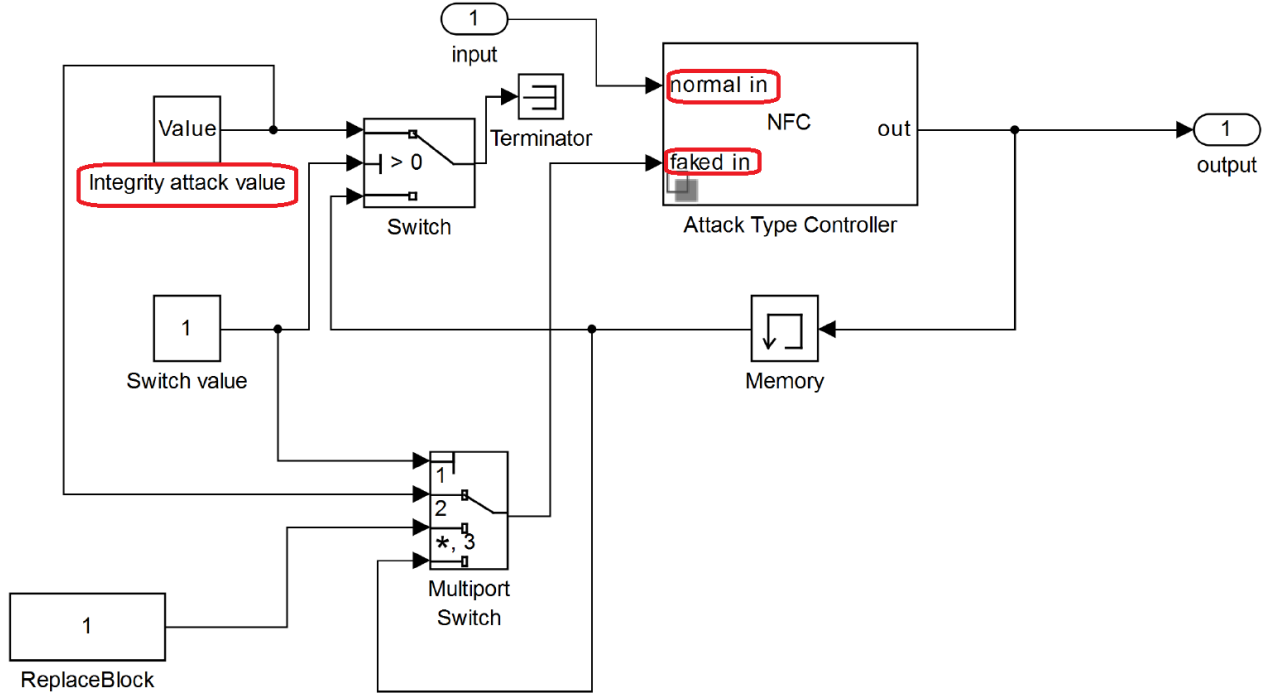


Figure 3.4: Functional implementation of integrity attack on sensor signal in TE Simulink model

To illustrate the usage of this metric, we executed an attack on the purge valve, making it to open. The attack resulted in lowered reactor pressure and reduced reactor efficiency. As shown in Figure 3.5a, high losses in the purge expectedly resulted in a significant increase of operational costs (Eq. 3.1). Due to decreased reactor pressure, the feed of the reactants was regulated to the lower rate, resulting in the decreased throughput of the final product as shown in Figure 3.5b (Eq. 3.2).

Attack modeling

The adversary's goal is to cause tangible impact on the process, either on its safety or on its economy. At the level of the regulatory or supervisory process control the attacker can either tamper with the sensor readings or modify the manipulated values issued by the controller via a *data integrity attack* (man-in-the-middle (MITM), packet injection, replay, etc.) or deny communication between process and controller via a *DoS attack* on the communication network or equipment (flood, starvation, packet drop, packet delay, etc.).

Let $S_i(t)$ be a PV or MV sample at time t , where $0 \leq t \leq T$, and T the duration of the simulation; time is *discrete*. The attack interval T_a is arbitrary and is limited by the simulation run time T . In our setting, we simulate the manipulated signal S'_i during a data integrity attack as follows:

$$S'_i(t) = \begin{cases} S_i(t), & \text{for } t \notin T_a \\ S_i^a(t), & \text{for } t \in T_a, \end{cases}$$

where $S_i^a(t)$ is the modified PV or MV measurement (attack value).

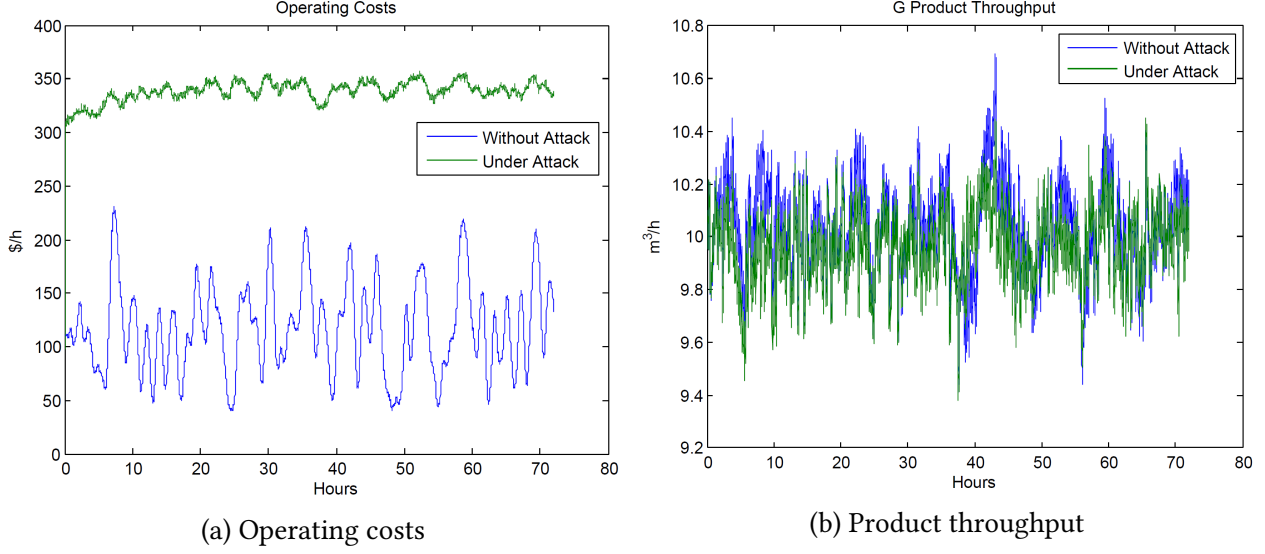


Figure 3.5: Economic impact of the attack on purge valve

During a DoS attack on a sensor signal new sensor measurements do not reach the controller. During a DoS attack on a controller signal new manipulated variables do not reach an actuator. Translated into the real world scenario, the controller's input register assigned to storing the measurements of a particular sensor will not be overwritten by a fresh value during the next control cycle run as would happen in a normal case. If $X_i(t)$ is a measurement of sensor i , $Y_j(t)$ is a manipulated variable for actuator j and the attack starts at time t_a , we have:

$$X_i^a(t) = X_i(t_a - 1) \text{ and } Y_j^a(t) = Y_j(t_a - 1),$$

where X_t^a is the stale data reading (the last process value received by the controller before the DoS attack) and Y_t^a is stale manipulated variable (the last MV received from the controller before the DoS attack).

In the context of Process Control Systems DoS attacks are similar to integrity attacks. The only difference is in *how* the attack value is brought about: by choosing a DoS approach the attacker has to attack at a *specific time* (e.g., when a valve is all the way open or closed). The advantage of the DoS attacks is that they can be used to manipulate the process even if control traffic is authenticated and the integrity of data packets is protected. This is because during DoS attacks, malicious actors simply delay or drop data packets instead of modifying them. This will be further discussed in Chapter 4.

3.2.2 Vinyl Acetate Monomer Process

Vinyl acetate (VAC) monomer is a commodity chemical and is an essential building block used in a wide variety of industrial and consumer products. VAC is a key ingredient in resins, intermediates used in paints, adhesives, coatings, textiles packaging, automotive plastic fuel tanks and many other final products. Detailed information about the product, including regulatory, health, environmental, and physical hazard information can be, for instance, found on the web page of Dow Chemical Company [298]. Note, that in this thesis we adopted vinyl acetate (VAC) abbreviation instead of

more commonly used vinyl acetate monomer (VAM) because of the original naming of this process in the introductory paper [51].

Process Description

In the VAC process, there are ten basic unit operations, which include a vaporizer, a catalytic plug flow reactor, a feed-effluent heat exchanger (FEHE), a separator, a gas compressor, an absorber, a carbon dioxide CO_2 removal system, a gas removal system, a tank for the liquid recycle stream, and an azeotropic distillation column with a decanter.

The route for vinyl acetate manufacturing used in the process model is the same as employed in today's manufacturing and involves seven chemical components. Ethylene C_2H_4 , oxygen O_2 , and acetic acid HAc are provided as both fresh and recycled feeds and are converted into the vinyl acetate with water H_2O and carbon dioxide CO_2 as byproducts. The fresh C_2H_4 stream contains an inert component C_2H_6 . The following reactions take place in the reactor:

Main reaction: $\text{C}_2\text{H}_4 + \text{CH}_3\text{COOH} + \frac{1}{2} \text{O}_2 \rightarrow \text{CH}_2=\text{CHOCOCH}_3 + \text{H}_2\text{O}$,

Side reaction: $\text{C}_2\text{H}_4 + 3 \text{O}_2 \rightarrow 2 \text{CO}_2 + 2 \text{H}_2\text{O}$.

The reactor contains tubes packed with a catalyst. Both reactions are highly exothermic and require tight control of the reactor cooling. The side reaction of ethylene combustion to CO_2 is highly undesirable as it lowers the conversion and complicates the removal of the reaction heat. Details of the ethylene combustion kinetics in the synthesis of vinyl acetate are presented in [137].

The reactor effluent is sent to the separator, where gas and liquid are separated. The vapor from the separator goes to the compressor and the liquid stream becomes a part of the feed to the distillation column. The gas from the compressor is recycled back to the reactor through the absorber and the CO_2 removal system. The liquid products, VAC and water, are withdrawn from the decanter. Figure 3.6 illustrates the process flowsheet with correspondingly control structure.

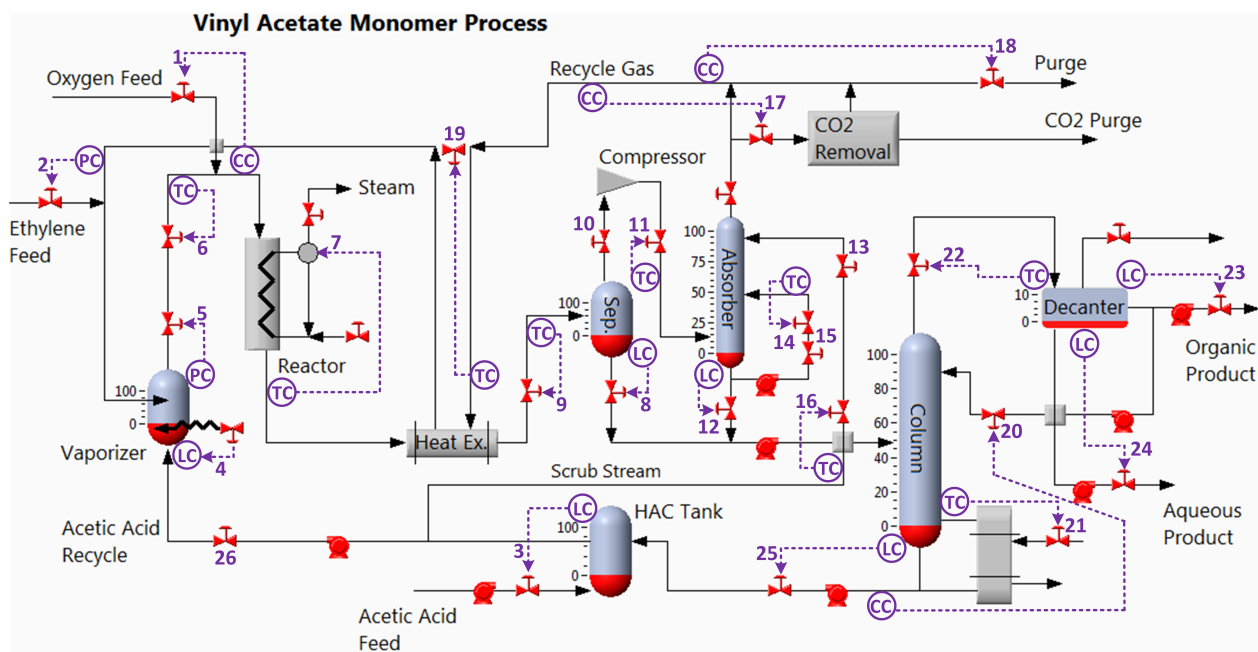


Figure 3.6: Vinyl Acetate Monomer plantwide process control structure

Detailed process description, including the reaction rate expressions, steady state process data and the major aspects of each unit operation are covered in [210] and Chapter 10 in [83]. To protect the proprietary information of any specific VAC production facility, the kinetic data, process flowsheet information, equipment data and modeling formulation in the published process came from sources in the open literature, e.g., in [210] and references therein.

Safety constraints

Two key safety constraints exist in the process. Exceeding either of the safety limits will shut down the process via interlocks:

- O₂ concentration must not exceed 8 mol% anywhere in the gas recycle loop to remain outside the explosivity envelope of ethylene. More on the limits of oxygen concentration in gas mixtures can be found in [331].
- The pressure in the gas recycle loop and distillation column must not exceed 965 kPa (140 psi) because of the mechanical construction limit of the vessels.

Operating constraints

The process constraints must be maintained to ensure efficient production without interruptions for maintenance. They are specified as the following upper and lower bounds for key process variables:

- The peak reactor temperature along the length of the tube must remain below 200°C to prevent mechanical damage to the catalyst, which would require shutdown and catalyst exchange.
- Liquid levels in the vaporizer, separator, absorber base, distillation column base, and decanter must operate within the limits of 10-90%.
- Reactor inlet temperature and the hot side exit temperature from the heat exchanger must remain above 130°C to avoid condensation of liquid.
- Organic phase in the decanter must contain less than 600 mol/million of acetic acid to prevent product contamination.
- The VAC composition in the bottoms stream must remain below 100 mol/million to prevent polymerization and fouling in the reboiler and vaporizer.

In contrast to the TE test process, the VAC model is not accompanied with the operating objective costs function for process control optimization. Instead, the economic objective was formulated as balancing trade-offs in maximizing vinyl acetate production and recovery while minimizing the carbon dioxide production and energy consumption.

Control Model

Similarly to the authors of the TE process, Luyben and Tyreus proposed a set of control requirements but did not suggest any process control scheme, challenging the research community to come up with their own control approaches, e.g., [83, 211, 245, 278]. The majority of the control design implementations were kept proprietary due to high modeling costs. The process model includes 246 states, 26 manipulated variables $XMV\{1-26\}$, and 43 measurements $XMEAS\{1-43\}$. A complete description of the process model formulation, assumptions, and implementation can be found in [51]. The process model utilizes a control scheme proposed in [211]. Figure 3.6 depicts the location of the control loops. The numbering of the control loops follows the numbering in Appendix 2 in [51]. Some manipulated variables are fixed, their control loops are not shown.

Model Enhancements to Support Cyber-Physical Security Research

The original authors of the Vinyl Acetate plant built a rigorous nonlinear dynamic model of the process to verify the feasibility of simulating the plant. Details on the assumptions and details of the modeling are described in Section 5 in [210]. Initially, the simulation model was implemented in TMODES, DuPont’s in-house dynamic simulation environment, which was not available for public use. In several academic works on vinyl acetate, process models were implemented in specialized commercial simulation tools such as HYSYS [68], Visual Modeler [278], Aspen Plus [278] and others. To make the process model available for a wider range of users McAvoy et al. have developed a simulation model of VAC for Matlab [52] which we adopted for our experimental framework. Both the steady state and dynamic behavior of the Matlab model were designed to be close to the TMODES model.

Originally, process equations had been modeled in Matlab and then translated into C-routines. The C codes were compiled into “MEX functions” and could be called within the Matlab environment. A separate m-file was responsible for the control of the VAC process (scheduling of the C-routines) with a developed multiloop Single-Input, Single-Output (SISO) controller architecture. Additional four Matlab routines were developed for plotting the results of the simulations. No simulation data were output to the workspace for further analysis. Also, the initial model did not provide any interface to the code of the VAC process mode. Therefore, any manipulations of the model inputs (e.g., setpoint or controller update) had to be carried out directly in the C code, requiring its re-compilation.

Considering the number of variables in the complex VAC process and the inconvenience of manipulating the process within the source code, we have developed a Simulink model of the process. Initially we had developed a user interface and attack codes without building a Simulink model. Several months of the experimentation have revealed limitations of this approach. Simulink [220] provides an interactive, graphical environment for modeling, simulating, and analyzing the dynamic systems at any level of details. Simulink models are compact enough to be understood with moderate effort. Additionally, the interactive nature of Simulink allows easy experimenting by changing the model and its parameters, and immediately observing what happens. Thus, modeling of attacks on a selected process component can be done easily by adding a function block with several lines of code. Such functionality suits well the “what if” nature of cyber-physical exploitation.

To allow users to benefit from our experience we designed the Simulink Model for the VAC process similar to the one used in the TE model. We also named manipulated variables as X MVs and controlled variables as X MEAS to maintain consistent notation between the TE and VAC Simulink models. Similar to the TE model, we instrumented the VAC Simulink model with a user interface for a convenient update of simulation parameters and setting up attacks on individual components. We also implemented an output of the simulated data to the workspace for further analysis and enabled their automatic visualization (Figure 3.7). Besides, we fixed several implementation mistakes in the process code and made several improvements to its control model to make the process more stable.

In general, the advantage of the VAC model is that it consists of realistic components. Therefore, one can easily modify and enhance the process model with further simulation code, e.g., change catalyst decay conditions or add dynamics of the pumps.

3.2.3 Comparison of TE and VAC Processes

To better understand the advantages and disadvantages of both process models, we compare them based on essential characteristics and numerical parameters. Table 3.3 compares the complexity between TE and VAC processes. Involvement of the real non-ideal chemical components and the

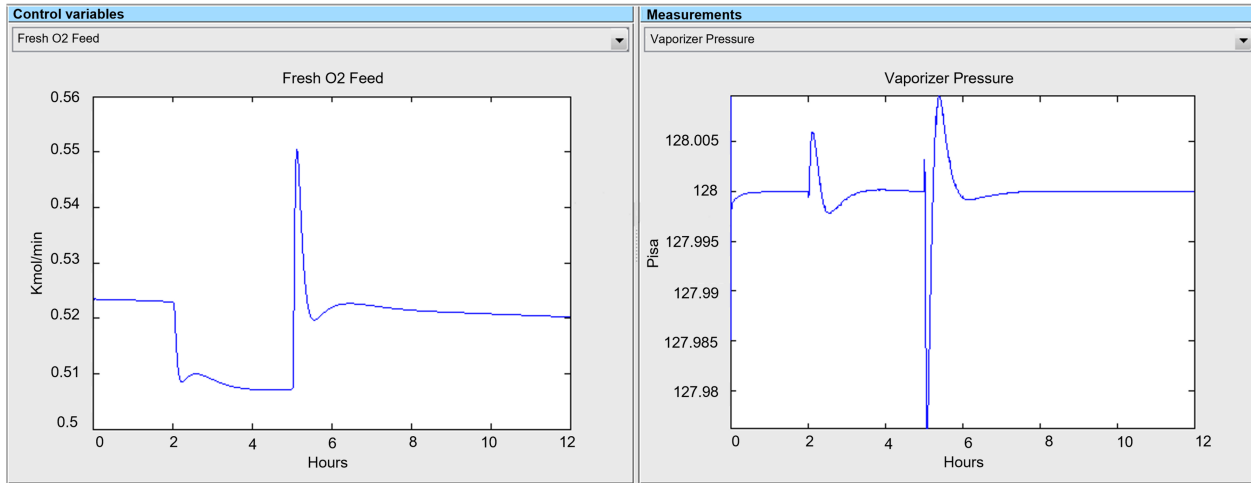


Figure 3.7: Visualization of simulation results

size of the process makes Vinyl Acetate a more challenging process control problem compared to Tennessee Eastman.

	TE	VAC
# sections	5	10
# states	50	246
# measurements	41	43
# MV	12	26
# modes	6	1
# setpoint changes	4	2
# disturbance modes	20	5
Process complexity	Medium	High

Table 3.3: Complexity comparison of the processes

Table 3.4 compares different features of the TE and VAC process model implementations. The availability of literature on the VAC process specifics allows to design high-precision *targeted* attacks. In contrast, the absence of a prior information about the TE process makes it an excellent case study for emulating “grey-box” exploitation use cases. In addition, the TE model includes a wide choice of the disturbances with both plantwide and local effects. The disturbances add uncertainties to the process behavior and can make exploitation more challenging. Note that since the exact chemistry of the TE process is unknown, there is no literature on the process details. However, there is a large number of works on the topic related to TE process control and safety. In contrast, VAC is a real process. There is a large body of literature on the specifics of the individual unit operations and chemical reactions.

In [211] Luyben et al. specify a number of complexities and interesting dynamic effects of the VAC process. The Vinyl Acetate process quickly becomes unstable if pushed outside of the steady state conditions. In contrast, controlling the TE process is rather straightforward and the process is more robust to malicious manipulations. Both processes have control loops with small and large time constants and exhibit fast and slow dynamics, depending on the change in the operating conditions.

	TE	VAC
Chemicals and reactions are specified	-	+
Equipment is specified	-	+
Safety constraints	+	+
Operating constraints	+	+
Operating cost function	+	-
Sensor signal noise	+	-
Process randomness	*	-
Predefined process code	+	-
Literature on the process	-	+

Table 3.4: Comparison of process models features; * enabled by us

3.3 Potential Applications of the Framework

As was mentioned above, the TE and VAC models can be used as standalone testbeds, encompassing physical and regulatory layers (L0-L1) of the process automation reference architecture (Figure 2.6). With respect to the cyber-physical system model (Figure 1.1), the DVCP framework covers physical and control layers as a stand-alone framework. If combined with hardware controllers and a distributed control infrastructure, which represents the cyber layer, the DVCP can serve as the physical layer of the cyber-physical system.

DVCP as part of the (distributed) control infrastructure

In our initial approach, we used the TE model as part of the distributed control infrastructure consisting of a hardware PLC and an HMI to study intrusion detection possibilities with a network-based Intrusion Detection System (IDS). As a Proof of Concept (PoC) we implemented an attack on situational awareness by the process operator. This type of attack was realized in Stuxnet [187]. The details of our setup and the results are presented in Appendix A.2. Since the goal of our research had been to focus on regulatory (control) and physical layers, it became apparent that adding additional hardware to the model did not add much value to the experiments but increased troubleshooting and maintenance efforts. This is why we opted for using the DVCP framework in a standalone mode.

Among notable examples of the TE process model being used as part of a distributed ICS architecture is the cybersecurity performance testbed developed at the National Institute for Standard and Technology (NIST) [39, 40]. The main objective of the testbed design was “to emulate real-world industrial systems as closely as possible without replicating an entire plant or assembly system”. Therefore, the testbed utilizes a variety of IP-routable and non-IP-routable industrial protocols, and typical industrial systems such as PLCs, OPC servers, historians and others. The testbed was conceived to serve as a guide on how to implement security safeguards effectively without negatively affecting process performance (hence, the usage of realistic hardware). A secondary objective was to measure the performance of industrial control systems while undergoing a cyber-attack, with resiliency being a central focus of the experimentation.

To evaluate the applicability of specific security controls to different industrial processes, the physical layer of the NIST testbed is decoupled from the rest of the distributed control infrastructure and constitutes separate enclaves for individual physical processes. Figure 3.8 shows a high-level architecture of the testbed with the Tennessee Eastman process.

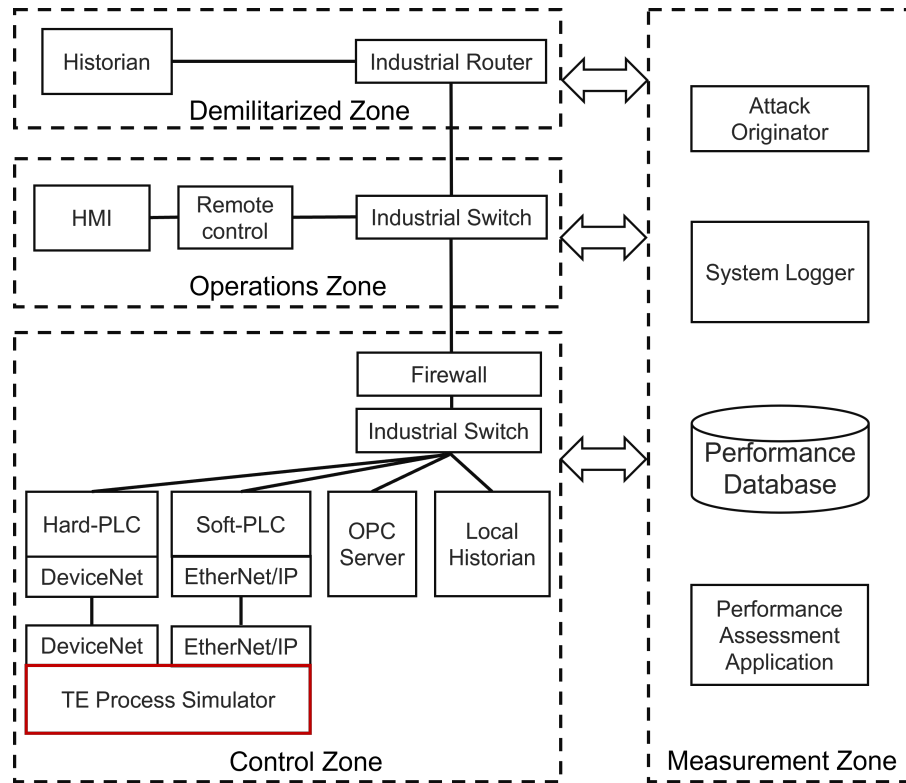


Figure 3.8: TE process enclave network diagram [40]

DVCP as standalone testbed

Working with the combination of control equipment/protocols and the physical process can be arduous. Once a cyber vulnerability is understood it is easier to *abstract* its potential impact and to apply it directly to the process model. For example, in Chapter 4 we demonstrate the susceptibility of physical processes to “stale data” brought about by DoS attacks on communication links between the controller and the process. We first present specific design weaknesses of the control equipment which would allow to launch DoS attacks and make the control equipment operate on stale data. After that, we demonstrate how those vulnerabilities can be made exploitable and study in depth the impact of different process parameters on the attacks’ success. In Chapter 5 we discuss Larsen’s algorithms for sensor signal spoofing and how they can be coded into a rootkit in the sensor’s firmware [189]. We apply these algorithms to spoofing sensor signals in the TE process model and develop a process-aware approach to detecting such spoofed signals. In this way, by abstracting from the real sensor microcontrollers we could concentrate on the factors that make spoofing challenging at the process level (in contrast to firmware rootkitting challenges), e.g., sensor signals parameters such as sensor noise, sampling frequency, types of dynamic process behavior, etc.

Since process models include controllers, the simulator allows the implementation of network-
ing attacks (e.g., Man-in-the-Middle) and attacks on controllers (e.g., false control logic upload) directly in the Simulink model. Simulated process data can be saved into a file (e.g., in .csv format) and used for testing malicious exploitation algorithms, exploits on the PLC, anomaly detection algorithms, etc.

Research questions and applications

Below we present examples of the research questions which can be studied using the DVCP framework. Both offensive and defensive security strategies can be discovered, designed and evaluated. Besides, the framework can be effectively used for educational and training purposes.

Attacker strategies. Understanding attacker Tactics, Techniques and Procedures (TTP) is the first step toward designing defensible environments and effective preventive and reactive security controls. An attacker who gained access to a control system of the physical plant of interest may initially have a poor cognitive awareness of plant susceptibility to various attack types and strategies. For instance, in the example of the Advanced Lead-Cool Fast Reactor European Demonstrator (ALFRED), it was shown that attacks on actuators pose safety risks to the entire system along with attacks on the lead temperature sensor. In contrast, the state of the process is negligibly affected by attacks that alter the values of PI gains [315]. This effect is explained by the PI controller's capability of regulating the errors of controlled variables close to zero even if the gain values are changed to three or four orders of magnitude larger than the reference setting.

Game theory is frequently applied to capture the interactive decision making process between the attacker (offense) and control system (defense). Both the offense and the defense actors are considered to be intelligent rational decision makers pursuing the optimization of their payoff functions. For instance, Barreto et al. [23] illustrate a differential game between an adversary and a controller in which the attacker manipulates a set of sensor/control signals, while the controller responds by tuning the remaining system parameters to negate the attack or minimize its impact.

It was previously shown that risk-taking and risk-averse attacker would employ different attack strategies [42, 316, 322]. Risk-averse attackers may refrain from certain attack actions due to the possible consequences of detection. One possible approach to avoid detection is employing stealth/covert attack techniques as presented in [15]. In the example of a water canal system, it was shown that a knowledgeable attacker can successfully steal water from the canal while bypassing detection by existing diagnostic tools designed for addressing random faults. In general, an attacker may apply different strategies to designing stealth attacks to adapt to detection schemes [43, 274]. Other attack techniques include various types of deception attacks, fault injections, false signals injections, replay attacks, spoofing of sensor and actuator signals, DoS/jamming attacks, transduction attacks, and others. A bibliographical study of different attack techniques from the control theory perspective can be found in [274]. Chapter 6 of this thesis presents different TTPs throughout the entire cyber-physical attack life cycle. Further adversarial TTPs can be found in [69]. It is worth noting that attack strategies can be categorized and studied from the logical layers of a cyber-physical system perspective (Figure 1.1). Examples of attack techniques in the control layer include stale data [30], skip frequency [328] and various types of covert attacks [274]. Attack techniques in the physical layer can be found in [30, 188]. Going further, the adversarial strategies can be studied from the attacker goal perspective such as maximizing economic impact or expected downtime.

Risk assessment. Existing processes and control strategies were not designed and built with security in mind. By studying process reactions to unanticipated conditions brought about by malicious manipulations, defenders can better understand the bottlenecks and other weaknesses in process design and control configuration. Such analysis, in turn, can help to better estimate attack consequences and associated risks. Consider the following example. A chemical reactor is typically the heart of an industrial process and might be a priority target for the adversary. In the context of the Tennessee Eastman process, a straightforward attack would be forging pressure sensor readings as $P_{\text{reac}}^{\text{low}}$ to provoke pressure rise. However, the response to this attack has slow dynamics and it takes more than eight hours to raise the pressure to the unsafe limit. To accelerate

the attack, a more knowledgeable attacker might prefer to attack a temperature sensor because the rate r of the reaction depends on the temperature T in an exponential fashion [211]:

$$r = A_f e^{-E_a/RT} f(C_i).$$

A small increase in temperature causes an unproportionally big increase in pressure. The TE process does not have an integrated heat exchange and pressure is controlled by the gas purge valve which is very small and therefore not effective in controlling a rapid pressure rise. It is apparent that P_{reac} exhibits a slow dynamic under one attack and a fast dynamic under another attack. Identification of control loops, which may cause quick process state changes if deliberately manipulated, is the first step in the cyber-physical risk assessment. In Chapter 4 of this thesis we propose an automated approach to evaluating the sensitivity of individual control loops to DoS attacks on sensors and actuators. Control loops that lead to a safety compromise in minutes or have a substantial impact on plant economy and stability are priority candidates for close monitoring and tight control methods. Haller and Genge [135] conducted a sensitivity analysis of the Vinyl Acetate process to a wide range of interventions (including attacks). The obtained results were then used for optimal placement of the limited number of IDS agents so that the coverage of the most sensitive variables is maximized. A similar analysis was conducted by Wang et al. [316] with the goal of optimal allocation of defense resources. A comprehensive survey of existing risk assessment methods for ICS and SCADA in the context of different physical processes can be found in [54].

Safety measures. Safety systems have the critical function of detecting dangerous or hazardous conditions and taking actions to prevent catastrophic consequences to people, equipment, and the environment. Process Hazards Analysis (PHA) or HAZOP studies are typical assessment methods used for the identification and evaluation of issues that may lead to hazardous situations. Consequently, Quantitative and Qualitative Risk Analysis (QRA) is used to assess the actual risk with measurable criteria like financial exposure or the probability of failure. The industrial control community has accumulated substantial experience in identifying and addressing potential hazards and operational problems in terms of plant design and human error. However, in light of today's cyber security threat landscape, it becomes essential to add cyber security considerations to the hazard analysis. For instance, Hashimoto et al. [139] proposed a safety-oriented approach to security zoning for control network using detectability and reachability matrices.

Financial impact. Operational targets and security requirements may conflict and have to be considered in conjunction. In many processes, the optimal reaction rate is achieved when the reactor is operated at high pressure. For instance, for the Tennessee Eastman process it was shown that the optimal operating steady state condition for the reactor pressure P_{reac} is as close as possible to the upper shutdown limit of 3000kPa and for the reactor level L_{reac} to its lower bound [260]. In this case, the attacker will be able to bring the system into an unsafe state quickly. To ensure secure operations it would be desirable to maintain a sufficient safety margin. However, maintaining a safety margin for P_{reac} of at least 100 kPa is equivalent to a 5% increase in operating costs [261]. This is a useful optimization problem to solve. The requirements on safety margins will also depend on how fast process operators and control system can detect and resolve a potentially unsafe situation. An extensive study on the premise of a desalination water treatment plant has shown that attacks with a seemingly negligible immediate physical impact may cause significant financial losses over a longer period of time [253].

Operator response. The requirement for better human responses to abnormal situations is a recognized industry-wide problem [2]. Many safety accidents have previously happened because of the late identification of critical process degradation and/or wrong operator response. Human operators react to alarms. One of the key elements of alarm activation configuration is the time

needed for the operator to respond to an alarm (Figure 3.9). Once the alarm is acknowledged by the operator, the clock is started. The operator has a predetermined time window to understand the alarm and select the course of action. Malicious process manipulation may invalidate the assumptions about the time needed for the operator to correctly identify corrective actions.

There are several approaches to assisting operators when responding to alarms. For example, automated algorithms for detecting spoofed process measurements could expedite the operator's analysis of data trustworthiness. One such approach is presented in Chapter 5 of this thesis. Additionally, the operators could be trained to recognize abnormal process behavior, which is potentially caused by an intentional manipulation and call for assistance. The operators may further be trained on how to best respond to potential attacks. For instance, if a certain controller is suspected of being compromised, the operators may assess the stability of the process and temporarily disconnect suspected equipment from the network. Process simulators such as DVCP can be used to test operator reaction procedures.

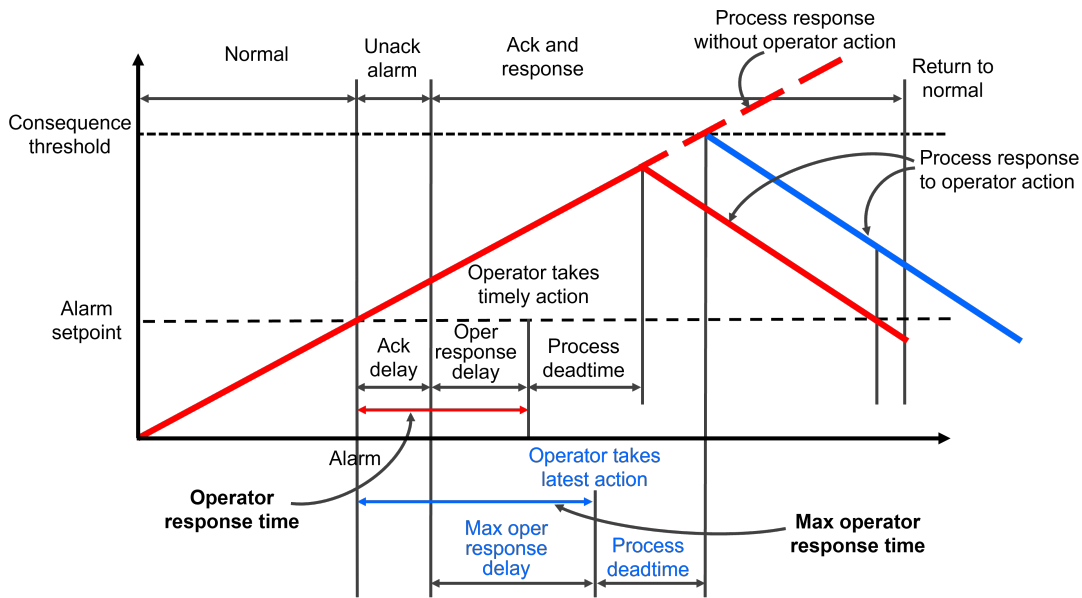


Figure 3.9: Calculation of the response time for the operator, based on [240]

Evaluation metrics. Established performance evaluation metrics exist in both the process control (e.g., [40]) and IT infrastructure domains (e.g., [56]). However, there is no established set of technical indicators for assessing the impact of offensive and defensive security approaches on the performance of a physical process. To challenge the status quo, the National Institute of Standards and Technology has initiated a project aimed at identifying suitable metrics for measuring the impact of security technologies on the performance of a process under protection [56]. A comprehensive set of metrics to estimate and quantify the intrinsic resilience of a control system is proposed by Zhu et al. [53]. In Chapter 4 of this thesis we propose several metrics to evaluate attacker strategies (e.g., Time to Attack) and to measure the impact of attacks on the process performance (Shutdown Time). In a work related to ours, Hau et al. [140] measure process resilience to attacks with Impact Ratio and Time-to-Critical-State metric. Jairo and Cárdenas [116] proposed a novel metric for assessing an anomaly detection algorithm that is based on the maximum impact an adversary can cause while remaining undetected (security vector) and on the expected time between false alarms (usability vector). This metric has been adopted in several academic works. In general, as stated by Jairo et al. [117], there are no common evaluation metrics used across multiple papers on cyber-physical security and more research is needed on this topic.

Process control security properties. The current practices in information security are strongly influenced by communication security. The CIA triad is also the basis of the ubiquitous claim that a core difference between “traditional” IT security and cyber-physical systems security is the importance of availability. While traditionally confidentiality is supposedly the most important and availability least, in the cyber-physical world the order is reversed. It is true that cyber-physical systems often have to meet hard real-time requirements, a topic rarely studied in “traditional” security research. This requirement can, for example, lead to a readjustment of priorities in the vulnerability scoring system as discovered by Khalili et al. [165]. The distinction is thus much more about hard versus soft real time than about availability versus confidentiality (as those terms are understood in IT security).

To meet the unique needs of securing cyber-physical systems, we propose to treat their core as a control system and not as an IT infrastructure. Consequently, we propose *Controllability*, *Observability*, and *Operability* (Appendix B.2) as the security objectives of a control system, whilst the much-used CIA triad captures the security requirements on the supporting IT infrastructure. Consider a contamination intrusion on a water distribution system. After the water is already contaminated, IT security methods are inapplicable for detecting this threat. Diao and Rauch [82] use controllability analysis in order to identify optimal sensor placements for timely detection of the contamination event and identification of the contamination source. Feng and Tesi [99] propose an approach to designing DoS-resilient control architectures for networked systems. They use impulsive controllers with dynamical observers and a state resetting mechanism to maximize the frequency and duration of the DoS attacks tolerance under which closed-loop stability is not destroyed. Impaired observability of the system state may affect the control system’s ability to conclude that an attack is happening or has happened. Damodaran and Rowe [75] introduce Process Model Inconsistency (PMI) to describe the discrepancy between the observations made in the known process model or process model developed ahead of a process being engineered and the ground truth process model or model of realistic process behavior. Detection of PMI is imperative for process security. For instance, when a transition of process state forced by the attacker is observed, such a transition may be flagged as an error or ignored, depending on the implementation of the known process model. In Chapter 6 of this thesis we show that impaired observability is similarly detrimental to the attack execution. One of the approaches to improve the survivability of physical processes under cyber-attacks is resilience-aware network segmentation. As shown by Genge and Siaterlis [112], such network design can significantly improve the tolerance period that would give operators more time to intervene. In such a hybrid strategy the control and network configurations are beneficially considered in conjunction.

Security-aware control strategies. Establishing control objectives is the first step in a plantwide control design procedure [211]. A viable control strategy not only satisfies operational and economic goals but is ideally also able to absorb the greatest anticipated disturbances, including those which are caused by intentional manipulations. Therefore the requirements related to the security aspects of plant operation should be determined upfront and included among the control objectives. In practice, it is hardly possible to design a single control structure capable of accommodating all operational goals. Therefore often one or more alternative control strategies are developed in parallel to compensate for the weaknesses of the other control schemes. One of the most widely applied techniques for alternative control is the usage of an *override controller* [206]. This controller can take command over a manipulated variable away from the main controller when otherwise the process would exceed a certain process limit. Such adaptive control keeps the plant running although perhaps at a suboptimal level. In the Tennessee Eastman process, an attack L_A^{low} is identical to the disturbance IDV(6). No basic *regulatory* control strategy can successfully reject this disturbance. Therefore most of the developed control structures are modified with overrides

to handle this situation [211, 261]. The approach of using overrides can be similarly applied to compensate for the other process impairments caused by the cyber-attacks described above. For instance, An and Yang [16] propose a controller model, which guarantees that the closed-loop system is asymptotically stable and the state constraints are not violated even in the presence of time-varying sensor and actuator attacks. Configuration and tuning of the control system also play an important role in process resilience under attack. Genge et al. [113] studied remotely executed attacks directed at opening or closing control valves. It was shown that controller tuning such as PLC task scheduling and mechanical properties of valves such as speed play an important role in process robustness and stability. These parameters can be considered at the design stage and configured accordingly in order to increase process resilience under cyber attacks and to make remote exploitation harder or even impossible.

Process-aware security controls. One of the major differences between digital and physical systems is that physics cannot be obscured or spoofed by the attacker. Therefore, knowledge about the underlying physical properties of the process under protection can be incorporated into the design of security controls. For instance, Mo et al. [233] proposes an approach to authenticating the correct control system operations called “physical watermarking”. Its utility lies in the ability to allow physical authentication of physical components. By injecting a known noisy input into a physical system, it is expected that the effect of this input can be found in the measurement of the true output. If an attacker is unaware of this physical watermark, the system cannot be adequately emulated because the attacker is unable to consistently generate the component of the output associated with this known noisy input. Consequently, the watermark acts as a physical nonce, forcing an attacker to generate outputs unique to the given inputs at a chosen time. Ahmed et al. [4, 7, 243] illustrate how physical invariants derived from process models and noise profiles of both sensors and process can be used to identify a wide range of cyber-physical attacks. In Chapter 5 we show how known correlations between certain process variables can be used to detect spoofed measurements in clusters of related sensor signals. A novel problem of covert communication via acoustic emissions by field instruments [175] and sensor signal measurements [142] has been recently proposed in the literature. The need for detecting physics-based covert channels necessitates the development of “next-generation” defenses, beyond previously suggested anomaly and intrusion detection systems.

Intrusion/anomaly detection. Intrusion/anomaly detection plays an important role in the early identification of a possible ongoing attack. In cyber-physical systems, it is often challenging to distinguish a well-designed cyber-physical attack from natural upsets and accidents. As a result, intrusion detection systems may falsely identify a disturbance as an attack and vice versa, resulting in false positives and false negatives. Detection of anomalies caused by natural events and intentional manipulations is studied disjointedly by the engineering and cyber security communities. Therefore, there are currently no established approaches to telling apart disturbances and attacks. This represents an important research problem. In general, intrusion detection can be implemented in any layer of a cyber-physical system as well as at the interfaces between the layers. One of the most common approaches to intrusion detection in industrial control systems is the monitoring of networking traffic (cyber layer). For instance, Yüksel et al. [325] use semantic information about the protocol such as the type and interpretation of a field and expected or observed interdependencies between fields, to construct features that capture the meaning of messages rather than only their format. Examples of attack detection at the control layer can be found in [45] and [232]. Detection in the process layer relies on process models. McEvoy and Wolthusen [224] detect attacks hidden in the sensor signals noise using proxy measurements. In another work, the same authors [223] propose an approach for identifying malicious activity that involves the use of a path authentication mechanism in combination with state estimation

for anomaly detection. This approach reasons conjointly over computational structures (cyber layer), and physical states (physical layer). As a standalone testbed, the DVCP framework is best suitable for designing and testing anomaly detection solutions in control and physical layers (e.g., see Chapter 5 of this thesis). A comprehensive survey of physics-based attack detection can be found in [310]. Further detection approaches based on the engineering methods from control and estimation theories are presented in [84].

Cyber-physical forensics. One of the emerging research areas is forensics-readiness of cyber-physical systems [13, 127, 230]. Process data represent an important source of forensic evidence when investigating targeted cyber-physical attacks. For instance, Alrimawi et al. [13] describe a hypothetical attack in which the attackers damage data servers by disabling HVAC systems and the fire detector. In order to confuse the investigators, the attacker also spoofs stored temperature measurements. Consequently, misled investigators believed that the servers were damaged because of a hardware failure. Despite its simplicity, this example brings up an important security issue of preserving forensic evidence in sensor signals. In Chapter 2 it was discussed that the fidelity of process data is decreasing as they traverse from the physical to higher layers of the Purdue network reference architecture. The common source of the stored process data for the majority of industrial applications is a historian. However, historians typically rely on (lossy) data compression for storage space optimization and “unimportant” data is removed. The removed data are lost forever and missing data is interpolated, resulting in a loss of data fidelity. An explanation of how extrapolation may result in missing precision in capturing events sequence and can make the root-cause analysis of process upset difficult or impossible is given in [80]. A pertinent research question is to determine the optimal relationship between data compression parameters while preserving sufficient fidelity of process data for forensic investigation. In Chapter 5 we provide examples of how the modification of data processing parameters may result in missing or incorrect information about the process state. When the forensic investigators analyze an incident, it is not immediately known to them whether critical information is missing in the data (e.g., certain frequencies are filtered out) or a certain measurement is incorrect.

Figure 3.10 shows how the change of sensor sampling frequency (blue signal) allows an attacker to conceal process oscillations (red signal) and impair forensic investigation. Currently, there are no methods to identify whether some information is potentially missing in sensor signals. This is a research question that deserves attention, especially when forensically sound data are needed for legally-bounded decisions.

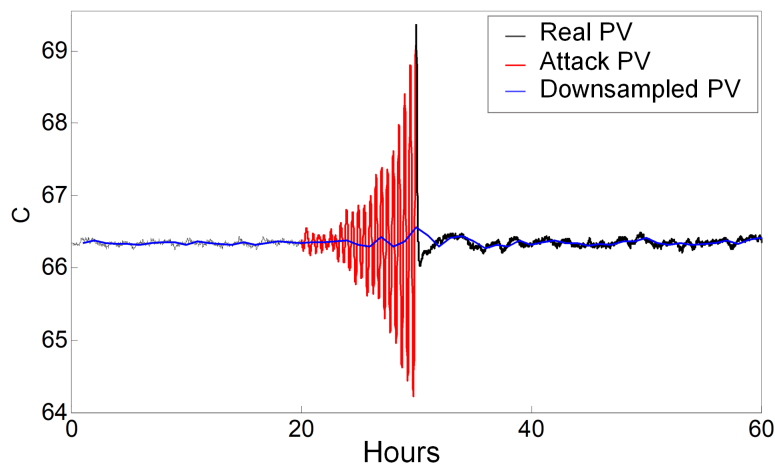


Figure 3.10: Concealment of attack trace (oscillation) by changing signal sampling frequency

Education. IT security professionals predominately lack practical experience with operational processes and industrial control systems. Getting access to realistic industrial or lab environments is not easy. Even with access to real productive equipment, proper planning, approval and support of the responsible personnel are required for conducting even a single experiment. Hardware-based table-top test models may include small-scale physical processes such as a conveyer belt or robotic hand [71, 104]. However, such models lack the complexity of physical interdependencies and control-loop couplings. The DVCP framework provides a realistic and visual experience of interacting with complex large-scale physical processes and their control architecture. Using its GUI, an individual without an engineering background can run process models under natural disturbances and different types of attacks and immediately observe the effects. The research questions include developing an effective educational curriculum for the IT community to attain necessary knowledge from chemical, process, control and mechanical engineering disciplines as well as studying various adversarial Tactics, Techniques and Procedures, including those presented in this thesis. The IT professionals can learn to work with engineering documentation such as Piping and Instrumentation Diagram and recognize potentially exploitable plant design weaknesses. For instance, a combination of a valve and a pump is a good candidate for a water hammer attack. In the next step, the trainees can evaluate the practical exploitability of the identified design flaw with additional process information. For instance, the pipe might be too thin and/or the valve too slow, impairing the attacker's ability to invoke a dangerous pressure transient. Harmonization of IT and OT vocabularies and mindsets in pursuit of an IT/OT convergence through the development of joint approaches and knowledge base is another research question. An efficient way to identify commonalities and differences in job vocabularies and mindsets is through joint activities such as tuning security alerts and response to an OT security incident. This could be achieved by constructing and executing a Red Team - Blue Team exercise with a multidisciplinary IT-OT team, following the cyber-physical attack life cycle steps presented in Chapter 6 of this thesis.

3.4 Discussion

Due to their predictive accuracy, process models play a central role in the innovation, design and operation of industrial processes. They provide decision support for better, faster and safer experimentation and help to reduce uncertainty. Although the Tennessee Eastman and Vinyl Acetate models mimic most of the properties and challenges of the continuous chemical processes, the cyber-physical security research community would benefit from further models of industrial processes, especially from different domains. However, as was mentioned at the beginning of this chapter, due to the significant efforts required to develop and validate such models, most of the models are either commercial or kept proprietary. One possible solution could be a collaboration between academia and industry. However, such cooperation will require extensive support from industrial scientists, calculated in man-hours. Until there will be an economic incentive for the industry to develop cyber-physical security methods, it can be challenging to convince companies to invest in such partnerships.

Lack of access by research groups to large-scale realistic or simulated processes can be partially addressed by publishing representative data sets, which contain attack traces or attack-alike anomalies, e.g., [156, 182, 183]. As a primary benefit, such benchmark data sets would enable evaluation and quality comparison of different cyber-physical security solutions. Among the obvious disadvantages of research on the premise of data sets is its limitation to the defensive type of research questions.

In 2005, the U.S. Chemical Safety Board (CSB) introduced an approach to safety incident analysis in form of computer animations. A core part of each video is a 3D computer animation sequence that

recreates a complex chemical accident along with erroneous human actions and decisions [145]. Figure 3.11 illustrates vapor flow between the foul condensate tank and turpentine-stripping column at the DeRidder pulp and paper mill, Louisiana USA. Computer animation techniques turned out to be so effective in conveying incident findings, that CSB videos are now being used for training, hazard analysis, and plant engineering worldwide. It is not surprising that the compelling way of presenting process dynamics and sequence of events in large facilities in form of animations has been adopted in the ICS security field. For example, Formby et al. [105] presented an open-source 3D visualisation of the simplified model of the Tennessee Eastman process based on the Unity 3D game engine. To ensure simulation realism, the authors purchased pre-made models of a warehouse environment, industrial reactor, pipes, and valves to recreate a realistic chemical manufacturing facility.

Combined with process simulation, the framework allows users to practice common attacks such as command injection, Man-in-the-Middle, and buffer overflows, and visually see the impact of their attacks in the 3D visualization. In another example, Kaspersky Lab developed a demonstration stand based on a 3D visualisation of the full TE model to illustrate the effectiveness of their early anomaly detection solution [194]. Due to the visual appeal and pronounced learning effect of 3D visualization, computer gaming approaches have already found their way into commercial ICS security products such as training platforms [301]. We expect to see a wider utilization of 3D visualization in the future.



Figure 3.11: Animation of explosion incident at the DeRidder pulp and paper mill, USA [268]

Research testbeds are typically either small in scale or recreate specific properties or parts of the environment being analyzed. Therefore, most research testbeds are not suitable for activities like full-fledged penetration testing or determining the value of enterprise grade security solutions. To address these needs, organizations worldwide have established specialized cyber security centers called “cyber ranges” that replicate real-world networked IT environments and allow them to conduct security exercises in controlled settings. Cyber ranges can be virtualised/emulated, hardware-based or hybrid. With the growing number and severity of threats to industrial control systems, it was only a matter of time till the first commercial and academic ICS cyber ranges emerged [118, 134, 164, 172]. Note that there is no clear differentiation between the terms “lab”, “testbed”, and “cyber range”. For example, Mudassar et al. [324] provide a comprehensive overview of cyber ranges and testbeds without clearly differentiating between the two, whereas some of the national experimental testbeds are called labs. Although cyber ranges are meant to be

more practical than research oriented, they could provide researchers with alternative sources of experimental environments.

Despite their realism, cyber ranges are generic environments and do not accurately replicate the specifics of individual assets or network topology in a given organization. Especially in the ICS domain, production processes frequently rely on customized machinery and a large variety of specialized assets. To allow companies to study the behaviors of these physical assets under different conditions, a new technology has emerged called “digital twin” [126]. This concept is based on modeling assets with all their geometrical data, kinematics, data flows and logical behavior using digital tools. The major difference between a simulation and a digital twin is their positioning within the product life cycle. A digital twin replicates an already existing physical object while a simulation is used as early as when the product development stage begins. In the context of production plants, digital twins can be seen as dynamic process simulators and provide status data of the production process and physical assets. The recent advancements in computational technologies, big data analytics and artificial intelligence have accelerated progress in digital twin technology, particularly in the context of Industry 4.0. A comprehensive academic and industrial overviews of the digital twin technology and its future can be found in [67, 109, 256]. In the ICS security field, digital twin technology has been utilized to address the challenge of security assessments. As a rule, activities like penetration testing or evaluation of defensive solutions are not allowed to be conducted in live environments. Bitton et al. [27] introduced an automated method for inferring the specification of a digital twin to aid the security evaluation of a specific industrial environment. The fidelity and implementation of the digital twin are then optimized for specific security tests that need to be conducted. Similar to generic process models, digital twins could greatly support academic research.

3.5 Related Work

In the past years, industrial and academic research communities have made significant progress in designing and implementing industrial control system testbeds. A comprehensive overview of 41 testbeds developed worldwide was compiled by Ani et al. [18]. The majority of the testbeds are focused on maximizing the coverage of the ICS architectural components as described in NIST 800-82 [291]. These components are physical process, field devices, communication gateway and control center. In contrast, in our experimental environment we focus on maximizing the coverage of physical process and control algorithm complexities.

The high-fidelity full-scale testbeds with complex physical processes are SWaT [218] and EPIC [267]. The realism of these testbeds allowed researchers to develop a suite of applicable attack tools and invent unique defense approaches [154]. However, the main disadvantage of these testbeds is their high cost and need for maintenance personnel. In two other large-scale high-fidelity testbeds developed at NIST [40] and the European Commission’s Joint Research Centre [284], some architectural components are simulated or emulated. Even though this reduced the capital and operational costs, these testbeds still require dedicated manpower for their continuous support. In contrast, the DVCP framework is of low maintenance.

The original code of the Tennessee Eastman process has been recently updated by Bathelt et al. [24]. They improved the predictability of model execution, making it solver-independent as well as enhancing model abilities, such as faster simulation time, random errors, and output of additional, partly internal, process values to monitor disturbances and output of reactions. The introduction of additional process measurements can be easily implemented in DVCP and is especially beneficial when evaluating attack execution and detection. Capaci et al. [41] have further improved the revised TE model by adding sources of variation to sensor signals to enable statistical

analysis of the results, similar to the implementation of TE in DVCP. Candell et al. [40] suggested to add an additional simulation block between controllers and process physics to simulate the effects of the communication channel such as packet error rate or delay variation. This could be a useful addition to the DVCP framework as network performance may invalidate the configuration of control loops. This will be further discussed in Chapter 4.

The ability of the Tennessee Eastman model to support applied cyber-physical security experimentation was demonstrated by Kaspersky Lab [103, 194, 195]. They developed machine learning approaches for detecting anomalies in process data and identifying potential cyber-physical attacks [194]. The developed solution was first validated with the TE simulator [103] and later successfully implemented in the Russian TANECO oil refinery [195].

Formby et al. [105] virtualized the Tennessee Eastman model and developed a 3D visualization of its dynamic behavior. The authors converted simulation code from the original Fortran and Matlab into a standalone C++ program, thus making the simulator accessible to a wider audience (no license fees required). Martin-Villalba et al. [215] implemented Modelica libraries of the simplified and full TE models, further improving the accessibility of the TE simulator.

As mentioned at the beginning of this chapter, open-source process models are scarce. Examples of models that have been used in academic cyber-physical security research include [102, 225, 253]. Siaterlis and Genge [284] proposed an Experimentation Platform for Internet Contingencies (EPIC) which included four process models: an oil-fired power plant, a chemical plant (Tennessee-Eastman), a railway system consisting of ten train models, and a power grid IEEE 39-bus system model. Although some of these models are either small in scale, or do not simulate dynamic behavior (steady state models), or rely on rather simple control logic, they allow researchers to study aspects specific to different industrial domains and cyber-physical processes.

Further research questions which could be studied on the premise of the DVCP framework can be found in a comprehensive overview of CPS security issues by Alguliyev et al. [10] and in a Survey of Surveys on security and privacy issues in CPS by Giraldo et al. [115].

3.6 Conclusions

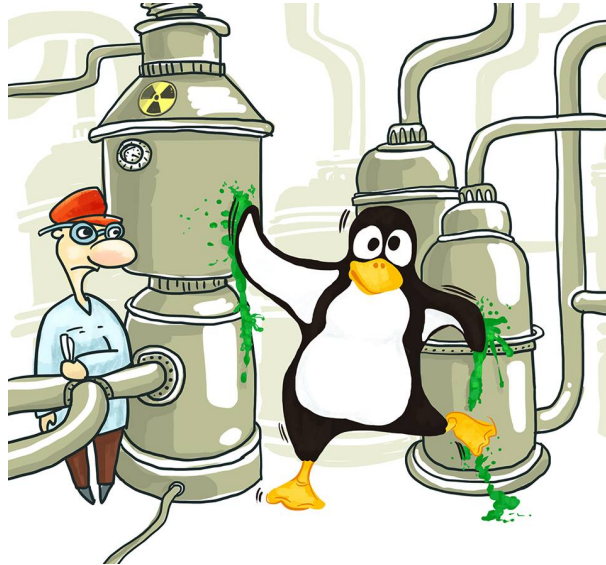
Software-based testbeds are frequently seen as less credible for scientific research. This opinion is predominately caused by the perceived reduced complexity of the simulated environments and assumed limited implementation of real-world behaviors. While this assessment might be valid for certain research areas or specific research questions, in the process- and control engineering fields, simulation-based testing environments have become the primary type of experimental testbeds. With the help of specialized software, engineers can model and simulate dynamic behaviors of physical processes and equipment with the required degree of fidelity. This allows for conducting cost-effective and safe “trial and error” type of R&D activities.

This chapter presented the Damn Vulnerable Chemical Process framework, a software-based experimental platform for cyber-physical security research. The framework includes simulated models of two chemical processes the Tennessee Eastman and Vinyl Acetate Monomer. Since the plant design and control strategies for both models were designed “without security in mind”, it makes them suitable for researching what it takes to convert a cyber attack into a successful cyber-physical attack. Owing to an elaborate emulation of the physical and control layers of a cyber-physical system, the framework allows to abstract from the research factors specific to cyber layer and concentrate on the aspects which are intrinsic to control- and process engineering.

The past few years were marked by the active development of various types of experimental, testing and visualization platforms to support the growing needs of the ICS security industry in research, education and testing. The emerging trends include 3D visualization and gamification,

cyber ranges and digital twins. It is expected that these technologies will further stimulate progress in the cyber-physical security discipline. Providing academia with access to commercial facilities could prove to be mutually beneficial to both communities.

To highlight the importance of using both IT and OT approaches to cyber-physical security, we designed a representative logo of the Damn Vulnerable Chemical Process framework.



Meaning of the logo. It is frequently argued that the insecurity of OT environments is a result of reliance on outdated versions of Windows OS and that hardened Linux-based hosts would be a more secure alternative. This logo suggests that no brand of OS (including Linux) is effective against process-oriented attacks. In the next chapters we will show that mitigating such attacks requires engineering methods from the process and control engineering domains.

Chapter 4

Vulnerability of Process Control Systems to Stale Data

Security is always excessive until it's not enough.

Robbie Sinclair

4.1 Introduction

One of the growing research areas of securing cyber-physical systems is the work on threat models that consider an adversary who can manipulate sensor or actuator signals in order to drive the physical process under control to an undesired state. While most of the work in this area explores the implications of manipulating signals directly, e.g., by modifying the payload of network packets in unprotected industrial protocols [223, 309] or replaying packets [101, 144], we are focusing on scenarios where direct packet manipulation/injection is not possible or challenging. Such scenarios include, among others, proprietary protocols with missing parsers and protected communication streams.

Whereas encryption of the control traffic is not always desired for ease of troubleshooting or due to strict hard real-time control requirements, the necessity for the authenticity and integrity mechanisms to prevent data spoofing, modification, and replay attacks was recognized by the community. In the power sector it resulted in the development of secure authentication features for the application layer of the DNP3 protocol [85]. Although the security controls provisioned do not allow an attacker to modify the communication flow (e.g., change packet payload or inject messages), it is still possible to execute a targeted attack on the underlying physical process at the protocol level by exploiting fundamental principles of control systems functionality.

We consider a scenario where the direct manipulation of data in transit is not feasible because the integrity of communications between field devices and controllers is protected by message authentication codes and the attacker does not have the key material. The attacker can still set process variables in controller memory to the desired values by strategically executing some form of a DoS attack. The controllers typically store sensor signals in dedicated memory registers that are overwritten when new values arrive. When communication is jammed, the register assigned to store values from a particular sensor cannot be updated with fresh values. Therefore, the last process values that reached the controller before the DoS condition will be used for the duration of the attack. As a result, the controller will calculate its outputs based on stale readings. Similarly, if a controller signal (manipulated variable) does not reach an actuator due to interrupted or delayed

communication, the actuator remains in its last state which was set during the previous control cycle (e.g., motor on or valve 80% open).

In this chapter we investigate the impact of different forms of DoS attacks (jamming, packet drop, packet delay, etc.) to make a control system operate on stale process data. When timing DoS attacks strategically, the attacker can manipulate a process similar to data integrity attacks. In cyber-physical systems with hard real-time requirements process data may only be valid for a short time and become irrelevant if arriving just few milliseconds too late. The same applies to the scheduling of control actions. Therefore timeliness must be protected and “stale” data should be recognized. The danger of undetected stale data usage in industrial facilities was acknowledged in [226]. The article with the title “Stale Data, Process Running” specifically mentions that degraded or failed communication between critical systems has previously led to stale HMI data and to malfunctioning and even failure of the industrial systems.

Exploiting Process Control Features

As discussed in Section 2.2, most Programmable Logic Controllers operate in a scan cycle where the logic in the controller uses the last saved input buffers obtained from sensor readings to calculate control commands for the actuators. If sensor readings cannot be updated because of interruptions on the communication line, the last value received will be the one used by the PLC in subsequent scan cycles until a fresh value arrives. In the professional control engineering jargon, such a situation is sometimes called as “frozen” input.

This specific property of industrial controllers can be abused by the attacker to make the controller operate on invalid process data as illustrated in Figure 4.1. By “freezing” a process value of interest, the attacker can deceive the controller about the current state of the process. This in turn will provoke compensating actions which may bring the process into a state desired by the attacker, e.g., an unsafe state such as overpressure. The attacker may also jam communication between the controller and the actuator. In this case, the output buffer of the PLC will keep being updated by the control logic, but the actuator will remain in its last state before the DoS attack.

Figure 4.2 illustrates the execution of a DoS attack on a reactor pressure sensor for a duration of 10 hours (from 10 to 20 hours). By executing a DoS attack, the attacker “locked” a specific pressure measurement in the controller memory, which is *above* the set point. In order to decrease the pressure and bring the process closer to the set point, the controller issues an opening command to the purge valve. Figure 4.2a shows the real reactor pressure as being measured by the sensor. As can be seen, the pressure is continuously decreasing due to the wrong compensating reaction. We call this compensation action “wrong” because it is not being executed based on the true process value: the state of the control logic does not match the state of the process. Figure 4.2b shows the position of the purge valve. For the duration of the attack, the purge valve receives an opening command in each scan cycle. Note, that because the same controller output was generated in each cycle (constant difference between PV and SP), the valve opening curve is linear. Upon attack completion, the controller receives a true pressure measurement (low pressure) and issues a full close command to the purge valve. The valve remains closed for some time to let pressure build up in the reactor.

While the concept of stale data is simple, a strategic execution of a Stale Data attack is not. The effect of a DoS attack will depend on the state of the input or output controller registers right before the DoS attack as well as on the overall state of the process. The attacker’s challenge therefore consists of identifying the *right* time to attack. If the DoS attack is not timed properly, the use of stale data by the controller would have a limited impact on the process. However, if the attacker is able to launch the DoS attack at the correct time, the use of stale data can drive the

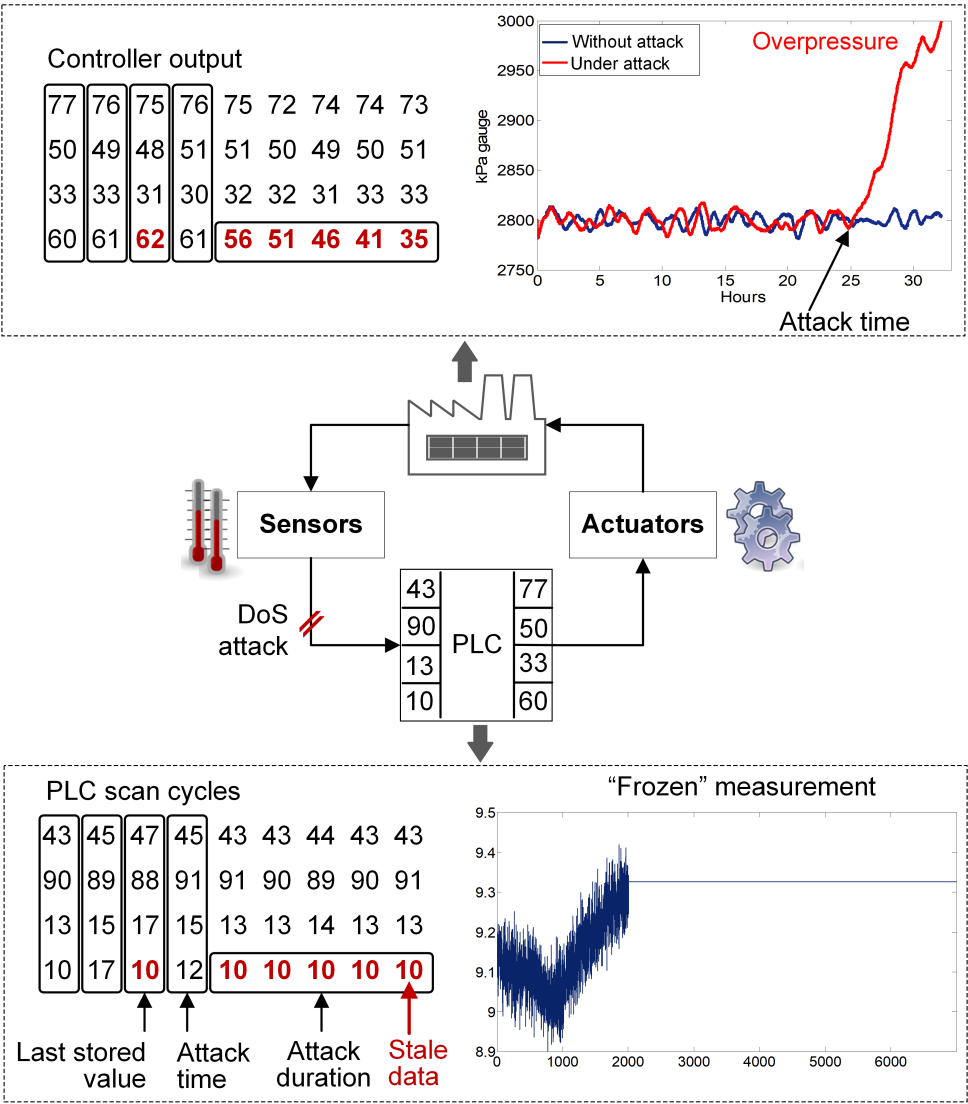


Figure 4.1: Stale data attack on sensor signal

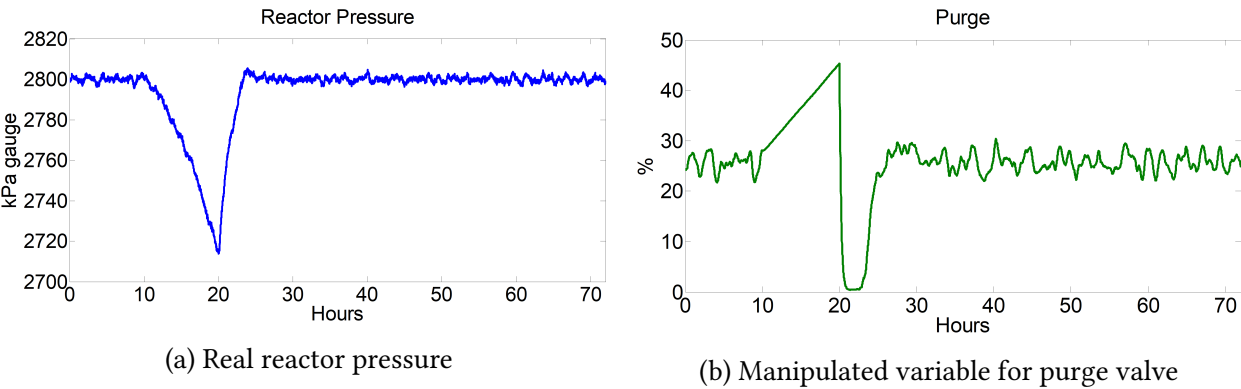


Figure 4.2: DoS attack on reactor sensor pressure

system to a state wanted by the attacker.

To begin with, the attacker needs to identify an overall more vulnerable state of the process or more vulnerable parts of the process. For instance, a process in a steady state is typically robust to disturbances. However, there could be control loops which are sensitive to malicious control and may lead to destabilization or ineffectiveness of the process through a control loop coupling effect even in a steady state. After identifying a sensor or actuator signal of interest, the attacker needs to identify vulnerable PV or MV. For instance, an attack might only be effective if a valve is locked in a certain position or the “frozen” process variable is above (or below) a certain threshold. Stale value will also define the time needed to achieve the desired physical effect, which may range from minutes to hours.

4.2 Susceptibility of Control Equipment to Stale Data

Almost every device in process control equipment needs to deal with stale data but this is rarely documented in the open literature, including user manuals. In many environments the bandwidth required to report the current state of all points (field devices) in the system exceeds the capacity of the communications channel. This is particularly prevalent when the data is communicated over long-haul serial communications. Open-source and proprietary protocols both describe themselves as “report by exception” meaning that data is reported to the server only when it has changed significantly. In many cases the amount of change required for the process value to be reported is configurable but the exact rules may still be opaque.

Idaho National Labs and other facilities conducted extensive testing of ICS equipment [238]. During those tests a number of scenarios were found where stale data could be maintained in a system for indefinite periods. No general technique is known that is applicable to every device so we describe some of the implementation details that have previously allowed for stale data.

The most common root cause of stale data is when the protocol stack is implemented in a stateless manner. The master maintains a set of timers for the data tied to individual data points or a set of data points assigned to a class. When data is received by exception, the timer is restarted. If the data changes often enough, the master never sends a poll request. If the timer expires, the point is placed into a point list to be requested during the next poll. In many cases, the timer is reset when the poll request is sent with the assumption that the point value will be returned in the next reply. In other cases, the timer is only reset upon a successful receipt of the updated data. In most cases, no alerts are generated if the data does not arrive. Also, if a TCP session is maintained without transferring any data, the sensor is never reported as offline.

Also more complex attacks have been shown possible. Thus, in many implementations the server listens on both the UDP and TCP ports. Messages received via TCP, UDP, or serial communication were routed to the same processing code without checking which communications mechanism sent the message. An exception report could arrive via the UDP port, but would be merged with the ones received via an existing TCP session. In cases when sequence numbers were used to detect missing and retransmitted data, empty acknowledgments with higher sequence numbers could be sent via UDP to increment the internal sequence number of the controller. When a legitimate message arrived at the TCP port, it would be discarded as a retransmission.

Furthermore, a number of examples can be formed, where interfering with a data channel is possible but fully viewing and manipulating the data channel is not. The most common case is networking equipment, where the attacker only has access to the administrative interface, but not to the data plane. For example, in common routers there is a “fast path” where packets are routed in custom designed application-specific integrated circuits (ASIC), and a “slow path”, where packets are transferred to the CPU. Access to the administrative interface can give code execution on the

device, but only packets transiting the slow path can be observed and manipulated. If all packets are requested by the slow path, the CPU is saturated resulting in a DoS on the communications link. If packets are sampled periodically, they can be examined which allows an attacker to perform a DoS at an opportune time. Similarly, keeping queuing buffers in the routers/switches occupied can result in dropped packets.

In [1] Abbasi et al. describe an approach to exploiting pin management operations in System on a Chip (SoC) integrated circuits, which are frequently used in industrial controllers. The attacker can take advantage of absent hardware interrupts for Pin Multiplexing and Pin Configuration operations when changing the configuration of the I/O pins. By tampering with *read* and *write* operations, the attacker can set I/O registers to the values of her choice without being detected. For a Stale Data attack on the actuator, the attacker can reconfigure the output pin as an input. As a result, the write operation by the controller will not succeed and the actuator will remain in its last position. To make the controller operate on stale sensor measurements, the attacker can reconfigure the input pin as an output and write a value of her choice into the physical address of the pin (e.g., last received sensor value).

Any of the above described techniques and many others could be used to make a process operate on stale data. As noted above the value of the stale data makes a difference in the final outcome of the attack. In case, when the traffic is encrypted, process state can be obtained from other sources, historian being the easiest option. Note, that there will be a certain delay between capturing process state by the sensor and update of the historian. This may impact the precision of selecting the “attack value”. In this chapter we consider an attacker who can observe raw sensor signals (PVs) and controller outputs (MVs).

4.3 Stale Data Attack

We consider an adversary, capable of launching an attack somewhere in the communication infrastructure at the level of regulatory control with the goal of *bringing a process into an undesirable state*. In the cyber domain, the attacker can tamper with process management (sensor signals) or modify manipulated variables issued by controllers. The attacker might either directly set the state of an actuator to influence the dynamics of the process, or deceive a controller about the current state of the process by means of a Stale Data attack so that the compensating actions taken have a harmful effect.

The basic research question is thus how (partial) compromise of the communication infrastructure can translate into control over physical processes. An attacker who has full control over the IT infrastructure, a complete view and understanding of the process, and who can manipulate any signals at will, would have the same control over the physical process as the legitimate operators. We are interested in the case where the attacker’s capabilities are more restricted.

Considering how such attacks are conducted in the cyber domain, they may be referred to as *DoS attacks* on sensor or actuator signals. However, considering their impact in the process domain, they may be called *Stale Data attacks*.

To a certain degree, in the context of industrial control systems, DoS attacks are similar to data integrity attacks with the main difference being that the adversary cannot directly set the “attack value”. Instead, the adversary can decide on the timing of an attack, such as its starting time t_a , to select an attack value of interest.

The advantage of the DoS attacks is that they can be used to manipulate the process even if control traffic is authenticated and the integrity of data packets is protected.

This is because during DoS attacks, malicious actors simply delay or drop data packets instead of modifying them.

We consider an attacker that can read a sensor or a controller signal for some given process or manipulated variable and has to decide when to “freeze” that value in the controller or actuator memory. To increase impact, the attack should be launched when the signal is jammed at a value of maximal benefit for the attacker. Under the assumption that freezing values in the normal operational range has limited negative impact on a process, freezing values, which significantly deviate from the set point (outliers), would be the attacker’s objective. Since the attacker needs to make her decision in real time, her task can be stated as a sequential decision problem. The attacker is presented with the candidate process or manipulated values at each time instance, and has to decide immediately on whether to launch an attack or not.

4.3.1 Example of Stale Data Impact

On 10 January 2019, 21:02 CET, the Continental Europe Power System which stretches across 26 countries registered for nine seconds the largest absolute frequency deviation since 2006. Among the main causes of the incident was a failure of a communication line, which resulted in stale data being used for various calculations [299].

The long-lasting steady-state frequency deviation (average 30 MHz) was caused by the drop of a telecommunication line between substation St. Peter (APG, Austria) and substation Simbach (TenneT, Germany), which stopped the measurement value transmission to TenneT Control Center South. The “frozen” (stale) measurement values in the Load Frequency Controller (LFC) of TenneT DE led to an error in the calculation of Area Control Error (ACE) for TenneT DE and subsequently in the German LFC block. The ACE error produced an incorrect power export/import imbalance of up to 1,000 MW. Since measured values are only transmitted if there is a change from the previous value, the fact of missing measurement updates was not immediately obvious.

At the time of the telecommunication failure on 9 January at 13:25, wind energy generation in Germany was *very high*, reaching 34 GW, which resulted in export of 723 MW via the four interconnectors between TenneT DE and APG. Afterwards, the measurement values were no longer transferred to the LFC of TenneT DE and the reading remained “frozen” at the last delivered value. On late 9 and during 10 January, the wind energy production in Germany *decreased significantly*, down to 4 GW. As a result, the export to Austria was strongly reduced and even led to an import from Austria of up to 330 MW on the four tie-lines with TenneT DE. The difference between real import/export (yellow curve in Figure 4.3) and wrong measurement (green line in Figure 4.3) values caused an “unreal additional power flow” from the TenneT control area towards APG. This resulted in a wrongly calculated surplus for the TenneT DE control area. Consequently, the LFC applied negative control power, which caused in reality a shortage in the TenneT DE control area. Due to incorrect values, this shortage was seen as a properly balanced situation. The wrongly calculated export (too high) led to a positive ACE (wrong) and to negative control power demand (also wrong).

This error “MW-value” (Equation 4.1) which was missing in the TenneT DE control zone caused a proportional frequency deviation of about 0 to 60 MHz between 9 January, 13:25 and 11 January, 09:37. The same frozen and incorrect values were transmitted by TenneT DE to Amprion, where they were used to calculate the import/export of control block Germany with the same result for the control block as for the control area. German energy demand and German ACE were also falsely calculated as balanced.

$$\text{Export(schedules)} = \text{Export(measurement)} = \text{Export(true)} + \text{Error} \quad (4.1)$$

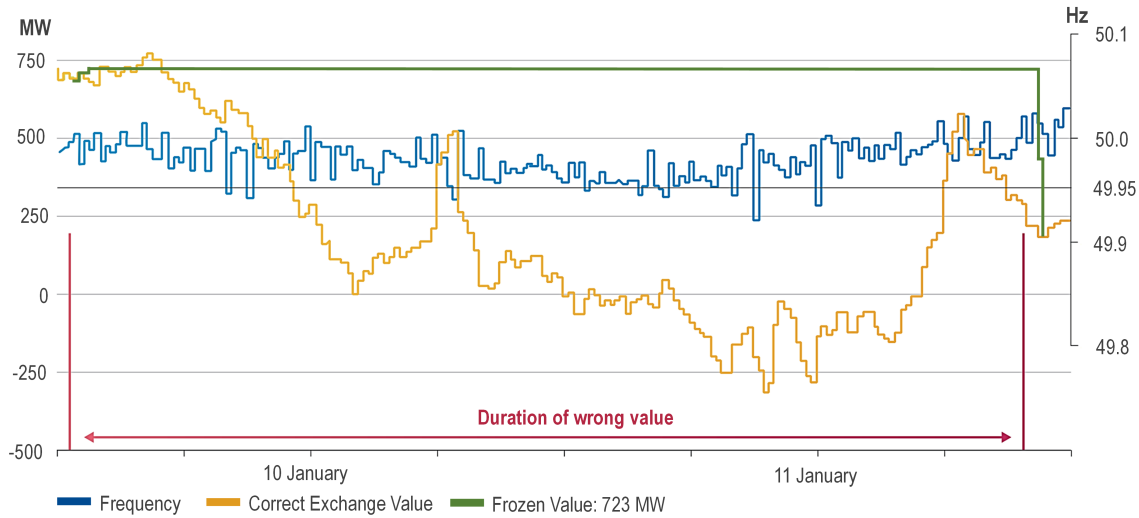


Figure 4.3: Frequency deviation due to “frozen” measurement [299]

Upon incident investigation, it became apparent that the current monitoring tools and alarm systems are not adequate for the kind of error described in this section. Therefore, all relevant stakeholders were recommended to check the existing definition and implementation of fail-safe measurement and telecommunication standards, for all interconnector values used by LFC across Continental Europe (CE). Also, it was decided to define and implement control system functionality standards to detect “frozen” LFC values across CE.

Even though the incident described in this section was not caused by a malicious event, it vividly illustrates how a potential Stale Data attack on a single measurement can cause a significant deterioration of a large-scale cyber-physical system. Therefore, the prevention and detection of such attacks must be considered by the engineers and operators of the industrial facilities.

4.3.2 Selecting Time to Attack

The timing of a DoS attack t_a plays an important role in the attack outcome. Figure 4.4 demonstrates the outcome of the DoS attacks on the reactor pressure sensor at a random time for a duration of $T_a = 10$ h. As can be observed, depending on t_a the impact varies from negligible (Figure 4.4a) to a safety shutdown (Figure 4.4d). Thus, an attack in Figure 4.4a has limited impact because the stale value was very close to the set point. In contrast, the impact in Figure 4.4d resulted from an attack at one of the lowest peaks of the signal. Evidently, the attacker has to develop a strategy to determine the optimal attack time t_a to achieve a desired impact.

In general, the further the process variable, chosen in an attack, is beyond a certain threshold, the higher the impact may be. Since the ongoing attack can be discovered over time, an attacker might want to reduce the chance of detection and reach the attack objective as quickly as possible. Therefore, the adversary may aim at launching the attack at a time the process or manipulated variable of interest reaches its local maximum/minimum (a more vulnerable state).

The attacker then faces the following problem: given a time-series that exhibits a sequence of peaks and valleys of different amplitude (Figure 4.5), she has to select one of the peaks in real time to launch her attack. If the attacker strikes too soon, she might miss the opportunity of having a higher impact on the system if she had waited longer (i.e., if the process variable would reach a more favorable value later). However, if the attacker waits too long, the process variable might not reach a more vulnerable state than previously observed, and the damage would be limited.

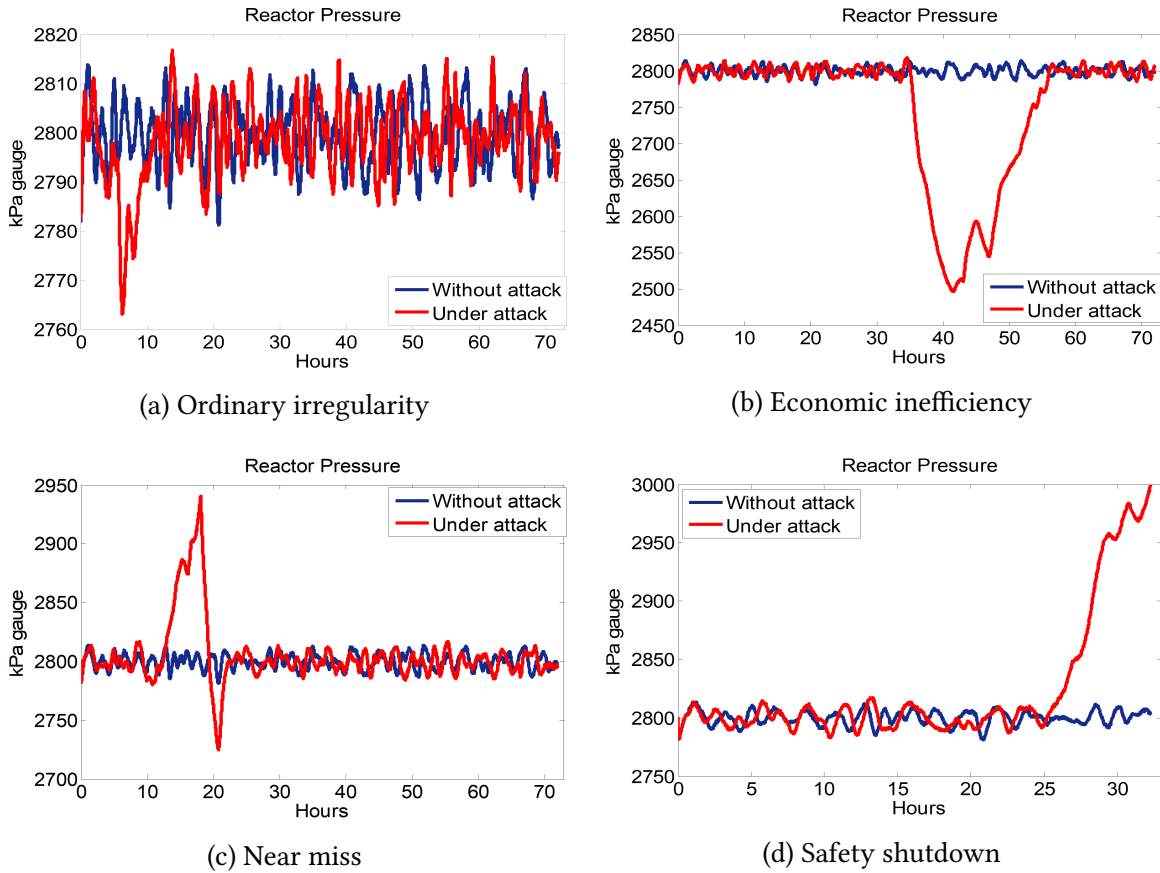


Figure 4.4: Different effects of DoS attack on a Reactor Pressure sensor at random times

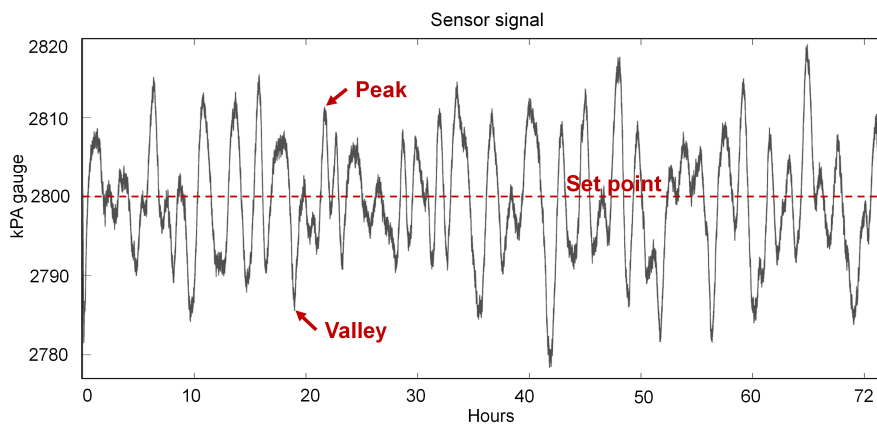


Figure 4.5: Example of a sensor signal

The problem of selecting a good time to attack can be framed as an optimal stopping problem. This class of problems is concerned with the challenge of picking the time to take a particular action based on sequentially observed random variables in order to maximize an expected payoff. The subclass of optimal stopping decisions where the binary decision to either stop or continue depends only on relative ranks is modeled as Best Choice Problem, also known as the Secretary Problem [107].

The problem the attacker has to solve is non-trivial in many practical situations as sensor signals and manipulated variables can be noisy and have sudden fluctuations. Therefore, we consider four types of sensor signals (see Figure 3.2) to make the selection problem more real and challenging. We then explore different stopping rules and compare their efficacy for timing attacks. Our results show how different stopping rules may be selected by risk-taking and risk-averse adversaries (those who prefer to attack as soon as a reasonable attack opportunity emerges).

4.4 Best Choice and Optimal Stopping Problem

This section provides the theoretical background necessary to formulate the research problem. The assumption is that the attacker has to perform an analysis in real time with little to no prior knowledge. Moreover, the attacker has to pull off the attack using devices with limited computational resources and limited memory (embedded systems). Therefore, the solution must involve fast and computationally light on-line algorithms for sensor signal analysis and decision making.

4.4.1 Best Choice Problem

In the standard version of the Best Choice Problem, a finite and known number n of items (or alternatives) are presented to a decision maker (DM) sequentially, one at a time, in random order. Time is discrete. At any time, the decision maker is able to rank all the observed items in terms of their desirability or quality. For each observed item, the decision maker must either accept it, in which case the search process terminates, or reject it, upon which the next item in the random order is presented and the decision maker faces the same problem as before. The decision maker's objective is to maximize the probability of selecting the best item from the pool of n available items.

The classical Best Choice Problem is formulated in terms of a hiring manager identifying the best secretary interviewed from among all the applicants; this version is called the Secretary Problem (SP). The problem formulation involves six assumptions [100]:

- Only one position available.
- The number of applicants n is finite and known in advance to the decision maker.
- The n applicants are interviewed sequentially, one at a time, in a random order; consequently, each of the $n!$ possible orders is equally likely.
- The decision maker can rank all the n applicants from best to worst without ties. The decision to either accept or reject an applicant at a given time is based only on the relative ranks of the applicants interviewed up to that point in time.
- An applicant who is rejected cannot be recalled later.
- The decision maker is satisfied with nothing but the best.

The payoff for the decision maker is one, if the best of the n applicants is chosen; otherwise, the payoff is zero. Note that an applicant is accepted only if she is the best applicant relative to all those who have been already interviewed (Figure 4.6). Such a relatively best applicant is called a *candidate*.

The stopping rule suggested for the Best Choice Problem is the following: do not make an offer to the first n/e candidates (where e is Euler's number) and after that make an offer to the first candidate whose value exceeds the values of all the candidates seen so far (or take the last applicant if this never occurs). In other words, the algorithm starts with a *learning phase*, during



Figure 4.6: Classic Secretary Problem [79]

which the decision maker interviews n/e candidates and sets an *aspiration level* that is equal to the highest value seen during the learning phase. After that, the decision maker should hire the first candidate who exceeds the aspiration level. Following the optimal stopping rule allows to select the best candidate with at least $(1/e)$ probability.

It has been recognized that the classical assumptions place more constraints on observation and selection than apply in practice [114]. Relaxing one or more assumptions to produce a more realistic formulation of the problem has attracted the attention of the research community. In this chapter, we consider the classical solution and a recent result, which assumes that the order in which the candidates arrive is not completely random, but has a probability distribution satisfying a hazard rate condition [212]. This assumption, which is commonly employed in engineering applications, states the following: given that the value of a candidate is not less than y , the likelihood that it is equal to y increases as y increases. For example, Gaussian, uniform and exponential distributions satisfy this property. Under these assumptions, it has been shown that the learning period can be shortened from n/e to $n/\log(n)$, meaning that a much smaller number of candidates has to be interviewed to set the optimal aspiration level and have a comparable probability of success. Since the probability of an attack being detected increases with time, having a shorter learning phase is beneficial to the attacker.

Figure 4.7 illustrates the formulation of the Secretary Problem in the context of a sensor signal in a chemical process. During the learning phase (called the *observation window*), the attacker finds the maximum (or minimum) value achieved by the process measurement. The *attack window* corresponds to the selection phase, where the attacker finds the first value greater than the value recorded in the observation window. Note that, in this case, the maximum value selected is lower than the overall maximum.

It is important to note that contrary to the independence of items assumed in the Best Choice Problem theory, the samples in sensor signal time series are correlated. Therefore, even if the attacker detects a value that is greater than that observed during the learning phase, she may be better off trying to determine if the signal is currently increasing and attack when the process measurement reaches its local peak. The next section discusses peak detection algorithms for determining if a signal has an upward or downward trend. Noteworthy, our empirical analysis of noisy signals has revealed that assuming a certain level of sample independence guarantees that a selection process based on the Best Choice Problem performs well for these type of signals.

4.4.2 Peak Detection in a Correlated Time Series

In a time series of measurements of a physical phenomenon, each sample X_i is heavily correlated with the next sample X_{i+1} . Thus, if a process variable is increasing, it would be unlikely for it to drop radically in the next time instant. According to the solution of the Secretary Problem, upon completion of the learning phase, the attacker should select the first sample whose value exceeds the aspiration level. By doing so, the attacker may miss the opportunity to select an even

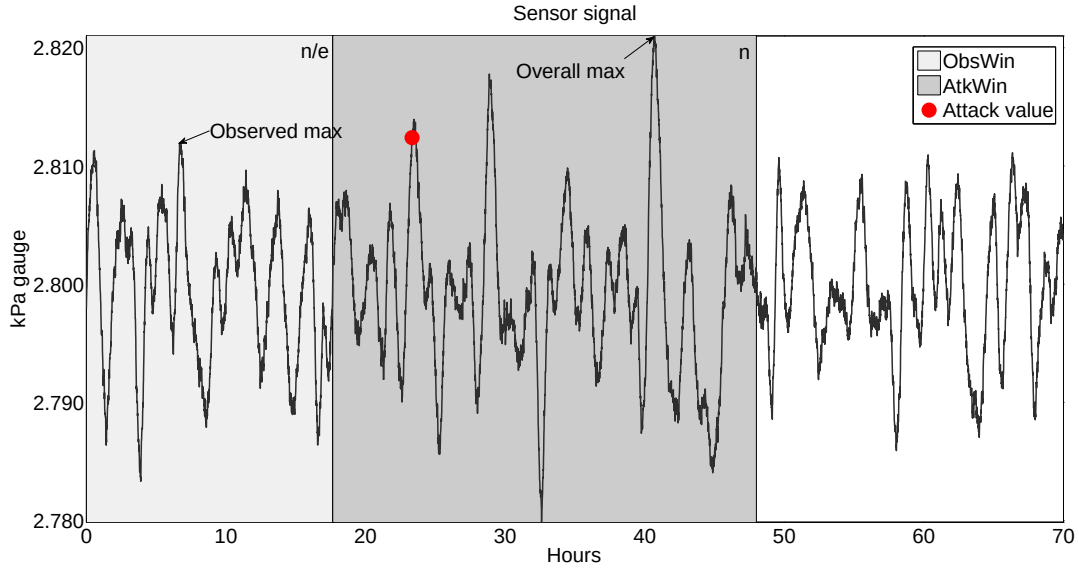


Figure 4.7: Illustration of the Secretary Problem solution applied to a sensor signal

“better” sample. For instance, in the case of an upward trend, the process measurement would keep increasing until it reaches its local peak, which will be the better time to make a selection. Therefore, in contrast to the static choice rule, the attacker may incorporate expectations about the future in the decision making process. The problem of identifying a signal peak is exacerbated by the fact that process variables are *noisy* and that an high amplitude sample might be followed by a sample of lower amplitude, followed by an even larger sample.

Forward Looking Search

To solve the peak detection problem, in our initial approach we applied a low-pass filter to sensor signals to smooth short-term fluctuations and highlight longer-term trends. In this case, the choice between stopping or continuing the search at sample X_i is determined by the difference between the stopping value and the continuation value X_{i+1} . This allows a peak to be identified as soon as a downward trend in a smoothed signal is detected (e.g., three consecutive measurement drops).

We use simple “moving average” as a low-pass filter, i.e. the mean of the previous n data points. We compute the moving mean of a signal as following. If m is the *smoothing interval* and s is a signal sample, then the moving average is given by:

$$\hat{s}_n = \begin{cases} \hat{s}_{n-1} - \frac{s_{m-n}}{n} + \frac{s_m}{n}, & \text{for } n > m, \\ \frac{n-1}{n} \cdot \hat{s}_{n-1} + \frac{s_n}{n}, & \text{for } n < m. \end{cases}$$

One of the side effects of signal smoothing is the delay of the smoothed signal with respect to the original signal by $(m - 1)/2$ samples. In financial applications, it is recommended to average the same number of values before and after where the average is calculated to avoid shifting of data. However, this is not possible during real-time analysis. Note, that noisy signals may require longer smoothing intervals to improve the signal-to-noise ratio. However, the cost of improved signal characteristic is a longer delay in peak detection; meaning that, when the smoothed signal reaches its peak, the real measurement has already begun to decay.

Change Detection

Another approach to peak detection is to apply a quickest change detection theory. Change detection tests are statistical techniques that help to identify possible drifts, abrupt and sudden changes in data series at an unknown time. Cumulative sum (CUSUM) is one of the most commonly used algorithms for change detection problems. Most optimal stopping rules are parametric and require a priori knowledge about signal statistics and the size of the expected shift. However, in many practical cases, the underlying distribution and the magnitudes of shifts are unknown. In such cases, usage of the non-parametric scheme (NCUSUM) is a better choice.

Assume that the sensor time series is represented by X_i . The goal is to identify the peak of this time series as soon as possible. To do this, we define a new time series S_i . We use two NCUSUM statistics for each signal: (i) S_i^+ to detect a local maximum in the signal; and (ii) S_i^- to detect a local minimum. The NCUSUM is initialized with $S_0^+ = 0$ and $S_0^- = 0$, and is updated as follows:

$$\begin{aligned} S_i^+ &= \max(0, |X_{i-1} - X_i| + S_{i-1}^+), \\ S_i^- &= \max(0, |X_i - X_{i-1}| + S_{i-1}^-). \end{aligned}$$

An alarm is triggered whenever $S_i > h$, where h is a threshold that can be used as a parameter to control the trade-off between the rate of false alarms and the delay in detecting the local maximum or minimum of the signal. Although the NCUSUM statistic is general and can be applied to any signal, the threshold h has to be learned for different types of sensor signals.

4.4.3 Heuristic Approach

Although the Secretary Problem is based on sound mathematical theory, one of the disadvantages of its optimal solution is the large number of non-selections (the last sample in the time series is taken because the attacker runs out of time). In such cases, the attacker has to be satisfied with a non-optimal value or has to start the search afresh. To avoid non-selections, we propose an alternative heuristic solution, which we call the *Outlier Test* (OT).

In process control applications, it is not uncommon to assume that process measurements have Gaussian distributions characterized by a mean μ and standard deviation σ . These parameters can be estimated for each signal during the observation period and be used by the adversary in the attack window to detect outliers.

According to the “three-sigma rule” in a normal distribution, nearly all the values (99.73%) lie within three standard deviations of the mean. In this case, the attacker should select the first sample in the attack window such that $X_i > \mu + 3\sigma$ or $X_i < \mu - 3\sigma$. Whereas the “three-sigma rule” threshold permits the selection of better samples, it can result in a longer search time. The sigma threshold can be lowered so that the selection can be performed faster if the attacker has a time deadline. A standard score curve can be used to determine the optimal Outlier Test threshold h . Figure 4.8 illustrates the application of the Outlier Test with $\sigma = 2.7$.

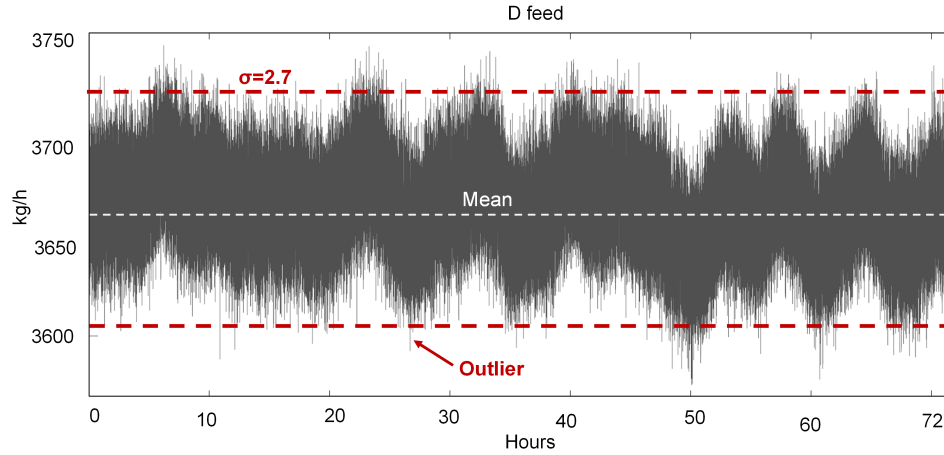


Figure 4.8: Illustration of Outlier Test detection

4.5 Experimental Results

For the empirical evaluation of our attack we use the Tennessee Eastman model from Section 3.2.1. The TE model does not simulate start-up and shutdown procedures, instead its execution starts with predefined base values. The “warm-up” phase of the plant lasts for about two hours and is excluded from our analysis. Design and implementation of DoS attacks follow the mathematical description in Equations 3.2.1 and 3.2.1.

We use a default simulation time of 72 hours. Unless specified differently, the experimental results are conducted with IDV(8) which stands for a variation in the reactor feed. This disturbance is successfully absorbed by the process and does not impact operations. However, it causes noticeable unpredictable deviations in sensor signals, which we would like to take into account as a complicating attack factor. All statistical results are averaged based on 50 simulation runs. The 95% confidence intervals are calculated using Student’s t -distribution and included selectively to provide a general sense of confidence interval range.

4.5.1 Shortest Shutdown Time

In our scenario, we assume an attacker whose goal is to force the physical process into an unsafe state and cause its shutdown. To evaluate the result of such an attack, we use the Shutdown Time (SDT) metric, which is the length of time that the process keeps running after the attack is started and until it shuts down upon exceeding the safety constraints.

In order to be able to compare the results of the proposed approaches, we first obtain the reference value of the *shortest possible* shutdown time for each attack on an individual sensor or valve. The reference values are selected by examining the overall lowest (min) and highest (max) process values $X_i^a(t)$ and manipulated variables $Y_j^a(t)$ in each individual simulation run and using them as stale values. $X_i^a(t)$ and $Y_j^a(t)$ are defined correspondingly as follows:

$$X_i^{min}(t) = \min_{t \in T} X_i(t) \text{ and } X_i^{max}(t) = \max_{t \in T} X_i(t);$$

$$Y_j^{min}(t) = \min_{t \in T} Y_j(t) \text{ and } Y_j^{max}(t) = \max_{t \in T} Y_j(t).$$

These reference SDTs can be considered as optimal because the intended attack impact can be achieved as quickly as possible. However, these attacks are practically infeasible to execute because

the attacker has to analyze the signals and launch the attacks in real time. We refer to these shutdown times as *optimum*.

4.5.2 Attacks on Controller Signals

Process control is achieved by the adjustment of the available degrees of freedom (manipulated variables). Therefore, *direct* attacks on a manipulated variable would be a logical first choice for the attacker. We analyze a scenario where the attacker attempts to fix a valve in its *lowest* possible position (optimum). The results of the simulations are summarized in Table 4.1. The table does not include the results for XMV5;9;12 because the positions of these valves are fixed in the process model and they are not used in the simulated control scheme.

As can be seen, under IDV(8), the Tennessee Eastman process is resilient to attacks on the *D*- and *E*-feed flow valves because no attack instances move the process to an unsafe state. At the same time, the process exhibits partial resilience to attacks on XMV(10) and is vulnerable to attacks on the remaining actuators. Except for the attacks on the condenser cooling water flow XMV(11), it takes a considerable amount of time to move the process to an unsafe state by means of a DoS attack on an actuator. This is not a beneficial outcome for the attacker because the plant operator may notice the process changes and respond with the appropriate countermeasures. Going further, it becomes even more difficult to disrupt a process in the steady state (Table 4.1).

The attacker may also influence the state of a valve *indirectly*. As described in Section 3.2.1, all the measured variables in the simulated model are computed based on two sensor measurements, except for XMV(10), which is a single-variable control loop. By executing a strategic DoS attack on the corresponding sensor signals, an attacker can deceive the controller about the actual state of the process and make it outputting a manipulated variable value that is desired by the attacker. As Table 4.1 suggests, the attacker can achieve a process disruption goal much faster in this manner. Although the result may appear to be counterintuitive, it has a logical explanation based on process control practice.

Among important control objectives for most continuous processes is to minimize valve movement to prevent wear and minimize local disturbances. Hence, in most cases, valve positions do not fluctuate much. Also, a valve is typically never kept in the fully open or fully closed position. As previously discussed, the attacker can only fix the valve in a position limited by the lowest observed MV. Since a valve state does not typically deviate much from its normal operating conditions, in general, an attack on a manipulated variable does not pose a danger to the process. The shutdown occurs due to the loss of a degree of freedom. As process dynamics evolve, the valve under attack remains non-responsive to the process control scheme, which eventually leads to an overall plant imbalance and to a “loss of control” situation.

In contrast, attacks on sensors not only preserve the degrees of freedom, but also allow an attacker to control valves at will. Sometimes, the attacker may even achieve the fully open or fully closed valve positions. In this case, the process will exceed the normal operating conditions and achieve an unsafe state much faster.

Figure 4.9 illustrates a “closing” DoS attack on the *C*-feed valve. First, we conduct a ten-hour-long F_C^{min} DoS attack directly on manipulated variable XMV(4) to fix the valve at one of its low states (Figure 4.9a). Because the valve remains in a fixed position, the F_C sensor keeps measuring a constant flow rate (Figure 4.9b). At the same time, the controller keeps generating the manipulated variable in accordance with the overall process dynamics. As can be seen in Figure 4.9a, the controller output quickly saturates, indicating that the process has reached a critical imbalance in the stoichiometric amounts of components *A* and *C*. Nevertheless, the process is able to withstand the attack and return to normal operation in about five hours (Figure 4.9b). In contrast, a deception

XMV	XMEAS	With IDV(8)						Steady State					
		Attack on XMEAS			Attack on XMV			Attack on XMEAS			Attack on XMV		
		SDT,h	Conf.Int.	SDT,h	Conf.Int.	#SDT		SDT,h	Conf.Int.	SDT,h	Conf.Int.	#SDT	
(1)	(40) (2)	- 3.69	- [3.63;3.75]	-	-	-		- 3.59	- [3.57;3.62]	-	-	-	
(2)	(40) (3)	- 2.83	- [2.76;2.90]	-	-	-		- 2.94	- [2.91;2.97]	-	-	-	
(3)	(5) (1)	9.17 -	[8.28;10.07] -	11.48	[10.36;12.60]	50		8.53 -	[8.49;8.56] -	-	-	-	
(4)	(31) (4)	7.97 1.71	[7.23;8.71] [1.56;1.85]	8.06	[7.48;8.63]	50		- 1.32	- [1.32;1.33]	53.99	[48.82;59.14]	7	
(6)	(7) (10)	8.56 -	[8.05;9.05] -	8.31	[7.93;8.70]	50		8.80 -	[8.77;8.83] -	-	-	-	
(7)	(12) (14)	5.50 11.58	[5.34;5.66] [11.27;11.89]	7.02	[6.77;7.28]	50		9.36 10.18	[9.29;9.43] [10.09;10.27]	33.89	[13.26;34.88]	42	
(8)	(15) (17)	6.39 1.33	[6.16;6.62] [1.11;1.54]	5.50	[5.29;5.72]	50		14.56 1.25	[14.42;14.68] [1.10;1.40]	18.89	[18.61;19.18]	50	
(10)	- (9)	- 1.34	- [1.30;1.38]	18.78	[6.39;31.05]	8		- 1.36	- [1.34;1.37]	29.38	[27.35;31.39]	35	
(11)	(8) (11)	2.38 -	[2.28;2.47] -	0.65	[0.63;0.67]	50		3.02 -	[2.98;3.05] -	0.76	[0.74;0.78]	50	

Table 4.1: Simulation results of the attacks on MV

attack F_C^{max} attack on the F_C sensor (Figure 4.9c) results in the rapid closure of the C -feed valve, which leads to a prompt process shutdown (Figure 4.9d) – process simulations end about 3 hours after attack launch.

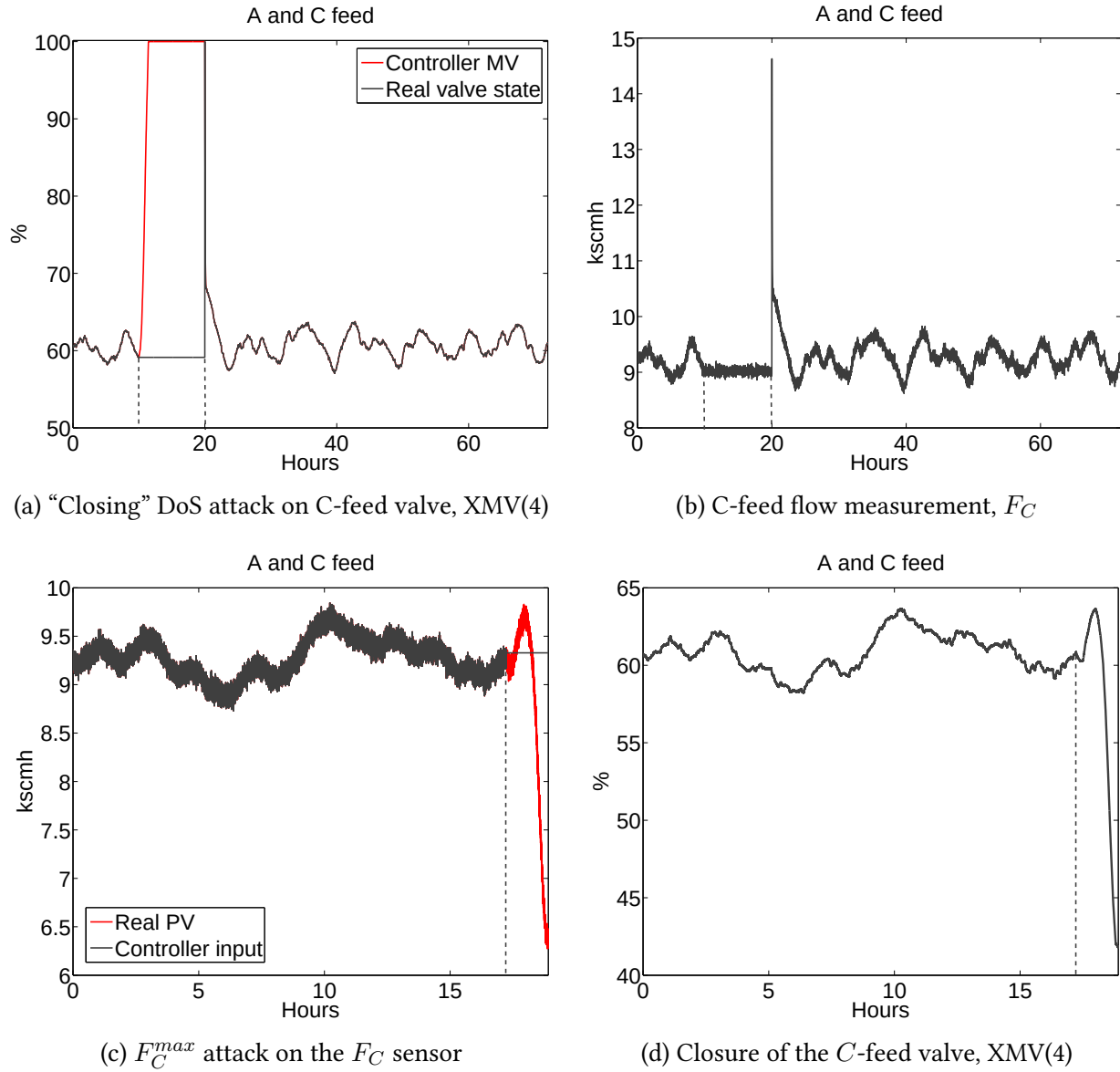


Figure 4.9: “Closing” DoS attack on the C -feed valve

Clearly, attacks on manipulated variable (i.e., actuators) are less rewarding than attacks on sensor signals. In order to achieve faster and more reliable results, the attacker should opt for DoS attacks on process measurements (i.e., sensors). It is important to note that attacks on certain sensors do not move the process to an unsafe state. This will be covered in more detail in the next section. Also, note that the shutdown times #SDT for the attacks on sensor signals are not specified because all the attacks, except those on XMEAS(1), caused process shutdowns. Nevertheless, non-critical process safety attacks on actuators may have negative impacts on the actuator mechanisms and on the process itself. As a controller output saturates, a “full-action” (full-open or full-close) command is sent to an actuator, causing significant equipment overstress and process disturbances.

In the context of the example illustrated in Figure 4.9, a full-open command is sent to a valve (Figure 4.9a). This evokes a rapid change in the valve position, overloading its mechanics and resulting in a temporary process upset (Figure 4.9b). In many cases, rapid movement of valves may also cause a highly disruptive cavitation process (formation of vapor cavities in a liquid) [197].

4.5.3 Attacks on Sensor Signals

This subsection analyzes the attacker's prospects of selecting the highest possible *process value* in real time by applying the strategies described in the preceding sections. First, we determine the shortest possible shutdown time or optimal SDT. The analysis of XMEAS 31;41 is omitted because these are not real-time measurements. Also, the results for XMEAS{10;11} are not included because no attacks on these sensors move the system to an unsafe state. As seen in Table 4.2, the sensitivity of control loops to stale data attacks varies greatly with shutdown times ranging from minutes to more than ten hours. The process is partially resistant to the F_A^{min} attack because only five attack instances trigger process shutdowns. At the same time, attacks P_{reac}^{max} and F_A^{max} do not drive the process to an unsafe state. Therefore, an attacker who plans to attack, say, the reactor pressure, should strike only at the minimum peaks.

Stopping Problem Approach Without Peak Detection

To begin with, the attacker has to decide on two parameters in the context of the Secretary Problem, namely, the number of samples or alternatives n to be considered, and the duration of the learning phase. For simplicity, we measure n in hours. In this case, for a time frame of 48 h, the number of alternatives is equal to $48 \times f_s$. We test both lengths of the learning window n/e and $n/\log(n)$. There is no specific requirement on the size of the learning window for the Outlier Test. We keep it set to n/e to obtain results comparable with the the "classical" secretary approach. Additionally, the Outlier Test requires a decision on the detection threshold; we set it to $h = 2\sigma$, which accounts for 95.45% of the data set.

We evaluate the approaches using two metrics: (i) error in selecting the best value in the attack window, in %; and (ii) number of non-selections (NS) – when no sample in the attack window is better than the aspiration value, and the last sample in the series is taken. To evaluate the influence of the error in selecting the best value, we also measure the shutdown time for each attack instance. This helps with estimating the sensitivity of the control loops to Stale Data attacks. Table 4.2 summarizes the results of the simulations.

Although the classical Secretary Problem solution with learning window size n/e results in the lowest error when selecting the best sample, its shortcoming is a high number of non-selections. This side effect is disadvantageous to the attacker who has to choose a clearly suboptimal candidate (last sample in the attack window) or decide to not launch the attack. Cutting the learning window to $n/\log(n)$ or applying the Outlier Test substantially reduces the number of non-selections, albeit at the cost of selecting a lower attack value and increased shutdown time. As Table 4.2 indicates, most of the control loops are sensitive to the magnitude of the selected attack value. A longer shutdown time is unfavorable to the attacker because plant operators may take corrective measures to bring the process back to its steady state.

A further feature of the classical SP solution is at odds with the attacker's requirements because it prescribes a long learning phase before selecting a candidate. The length of the observation window together with the time to selection (TTS) constitute the time to attack (TTA). The longer the TTA, the greater the chance for the attacker to be detected, including being accidentally discovered. Table 4.3 presents a comparison of the approaches based on the aggregated (averaged) results from Table 4.2 along with TTA and TTS results. Although the Outlier Test and Secretary Problem

XMEAS	Variable name	Optimum	Secretary, n/e			Secretary, n/log(n)			Outlier Test, $h = 2\sigma$		
		SDT,h	SDT,h	Error,%	NS,%	SDT,h	Error,%	NS,%	SDT,h	Error,%	NS,%
(1)	min max A-feed rate	22.22	-	13.96	40	-	42.85	12	-	18.74	70
(2)	min max D-feed rate	5.15 3.69	5.93 3.85	8.52 6.38	26 20	6.05 4.27	19.57 13.64	0 4	6.72 6.36	42.93 42.15	0 0
(3)	min max E-feed rate	4.29 2.83	4.46 3.51	8.13 7.82	38 36	4.55 3.32	17.65 19.43	2 2	5.26 4.44	44.87 45.18	0 0
(4)	min max C-feed rate	1.05 1.78	1.78 2.38	12.44 18.83	34 44	1.99 1.93	35.80 34.45	4 8	2.07 2.20	29.15 28.44	12 6
(5)	min max Recycle flow	4.39 9.17	5.44 9.90	17.12 25.12	32 40	8.73 10.11	39.17 42.66	2 0	6.77 10.17	28.92 35.44	14 12
(7)	min max Reactor pressure	8.56	8.2	23.41	34	8.21	39.20	8	8.51	25.93	8
(8)	min max Reactor level	2.37 2.73	2.90 3.08	14.57 11.13	34 32	3.31 3.40	30.05 24.20	2 0	3.96 3.71	44.98 37.69	0 0
(9)	min max Reactor temperatur.	1.34 0.65	1.39 0.69	4.06 3.74	30 40	1.45 0.70	13.54 13.56	8 6	1.75 3.71	45.25 37.69	0 0
(12)	min max Separator level	5.50 3.49	6.92 4.28	17.10 15.97	36 30	8.74 5.86	40.67 41.60	0 2	8.15 5.96	31.18 36.10	0 4
(14)	min max Separator underflow	12.03 11.58	12.10 11.13	17.50 20.22	34 26	11.88 11.69	39.16 40.70	0 2	12.58 12.15	34.01 34.33	0 2
(15)	min max Stripper level	6.39 6.35	7.96 8.54	20.28 23.76	20 34	11.06 13.05	49.43 56.36	4 2	9.70 9.71	35.63 34.01	14 10
(17)	min max Stripper underflow	1.86 1.33	2.33 2.36	4.89 7.07	18 24	3.37 3.10	13.91 19.99	4 2	6.75 7.14	42.77 43.30	0 0

Table 4.2: Simulation results of the approaches to the Best Choice Problem solution for the attacks on PV

with $n/\log(n)$ approaches both demonstrate roughly similar results with regard to the accuracy of selecting the maximum value and the time before making a selection, the latter approach benefits from a drastically smaller TTA with a very small observation window. It is, therefore, more advantageous for the attacker to use this approach.

	Error	TTS	TTA	NS
SP, n/e	13.78 %	11.52 h	27.17 h	37.77 %
SP, $n/\log(n)$	31.26 %	7.63 h	9.61 h	3.36%
OT, $h = 2\sigma$	36.93 %	5.32 h	20.98 h	3.9 %

Table 4.3: Aggregated comparison between the approaches

Stopping Problem Approach With Peak Detection

To increase the prospects for selecting the best possible attack value, the attacker may add a peak detection step to the attack strategy. The forward looking search approach uses a simple moving average to smooth sensor signals and filter out high frequency noise (Figure 4.10). We focus on four types of signals, with different noise floors to compare between peak selection approaches.

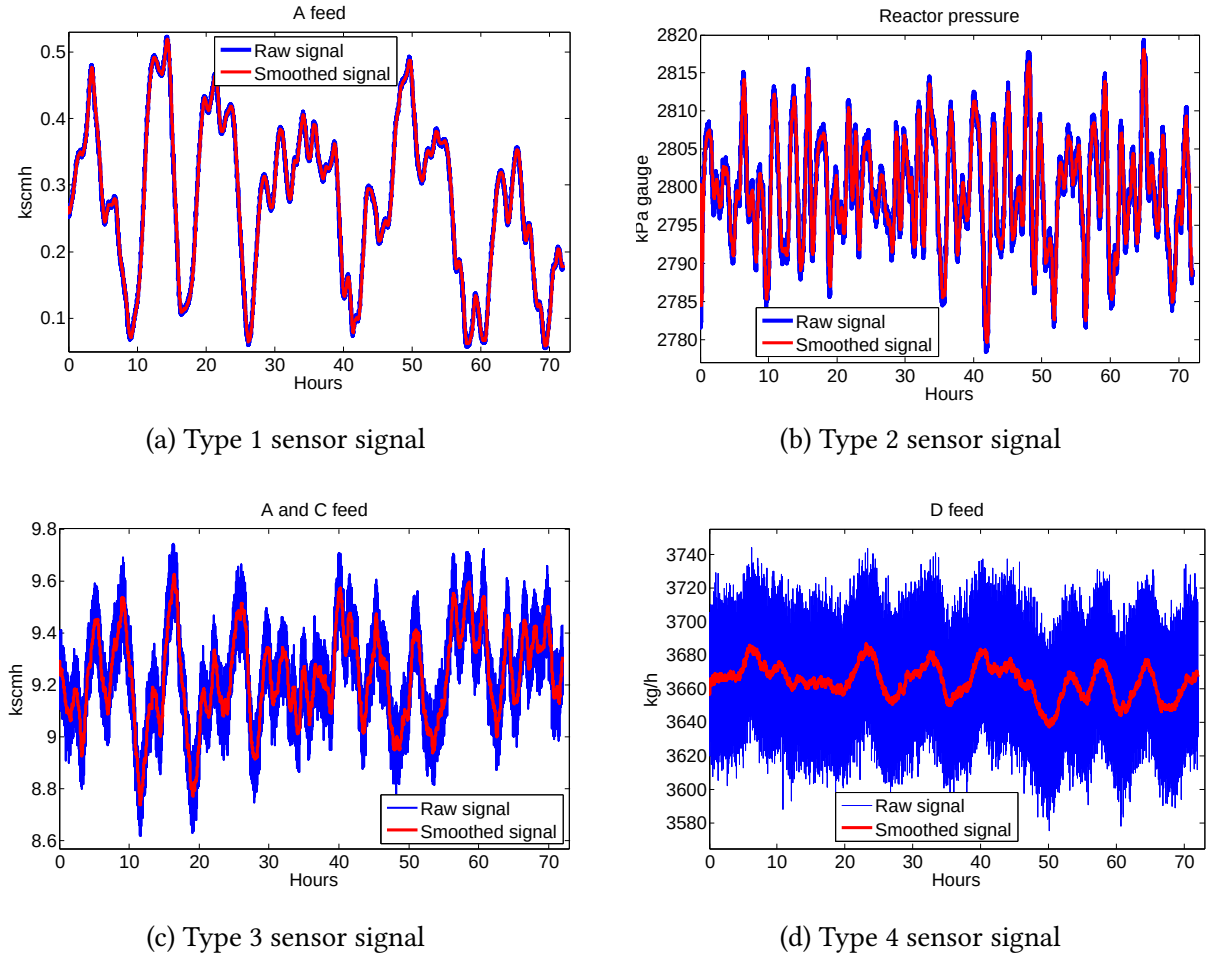
The threshold in the CUSUM algorithm depends on the scale of the change that an attacker would like to detect. Thus, higher thresholds should be selected for noisy signals than for less noisy signals. We determined the optimal thresholds to be $h = 0.005$ for Type 1 and 2, and $h = 0.015$ for Type 3. There are no distinct peaks in the signals of Type 4. We compare the peak detection approaches with the classic SP solution as it delivers the best results in selecting the max value (without peak detection). Figure 4.11 illustrates the comparison of the results in the form of a histogram, which represents the distribution of the errors in selecting an optimal sample.

As can be seen, peak detection approaches deliver superior results compared to the Secretary Problem algorithm for sensor signals of Type 1 (first row in the Figure 4.11). This is because the SP search terminates too early, before reaching the peak, yielding a high error in selecting the max value. At the same time, the low noise level allows for accurate detection of the signal peak.

In contrast, peak detection does not perform well in the case of noisy signals. Thus, noise corrupts the CUSUM statistics, resulting in a high false detection rate. Although smoothing reduces signal noise, the selected raw sample is usually suboptimal due to noise uncertainty. On the other hand, the SP solution delivers lower errors for noisy signals because it sets up the aspiration value based on the best outlier value in the observation window. Note that while enhancing the Secretary Problem approach with peak detection may, in certain cases, substantially increase the attacker's ability to select the best value, any peak detection approach would need to be tuned to match the sensor signal characteristics for it be useful.

Attacks in a Steady State

One of the primary goals of process control is to keep the process as close as possible to its optimal steady state. In the Tennessee Eastman plant, the process variables in steady state do not deviate much from the set points and accurately follow Gaussian distributions (Figure 6.20a). Most of the sensor signals are of Type 4, with a few signals of Type 3. The results in Table 4.4 show that, relative to each other, all the approaches demonstrate performance similar to the use case in Section 4.5.3, the only exception being the NS statistic for the Outlier Test. Even with a higher threshold of $h = 3.5\sigma$ (0.999% of the data set), the Outlier Test delivers a diminishingly small number of non-selections. Also, in the steady state, the observation window for the Outlier Test


 Figure 4.10: Four types of sensor signals in TE process: raw and smoothed ($\mu = 250$) signals

can be reduced substantially. For most sensor signals, it is sufficient to observe only five hours of samples in order to accurately identify μ and σ . Under these conditions, the TTA for the Outlier Test becomes comparable to that for the Secretary Problem approach with the reduced learning window $n/\log(n)$.

Due to the minimal variations in process measurements, all approaches demonstrate lower errors when selecting the best possible value; but for the same reason, the selected attack values are also (much) lower. Moreover, in the steady state, the process is more resilient to attacks on certain control loops. For example, it is no longer possible to move the process to an unsafe state by attacking XMEAS(1), and the shutdown times for attacks on XMEAS{5;12;17} increase more than twofold. Therefore, it is less rewarding for the attacker to strike when the process is at steady state.

	Error	TTS	TTA	NS
SP, n/e	3.20 %	2.87 h	28.53 h	37.09 %
SP, $n/\log(n)$	9.09 %	8.39 h	10.37 h	8.64 %
OT, $h = 3.5\sigma$	7.82 %	6.39 h	22.02 h	1.5 %

Table 4.4: Comparison between the approaches, steady state

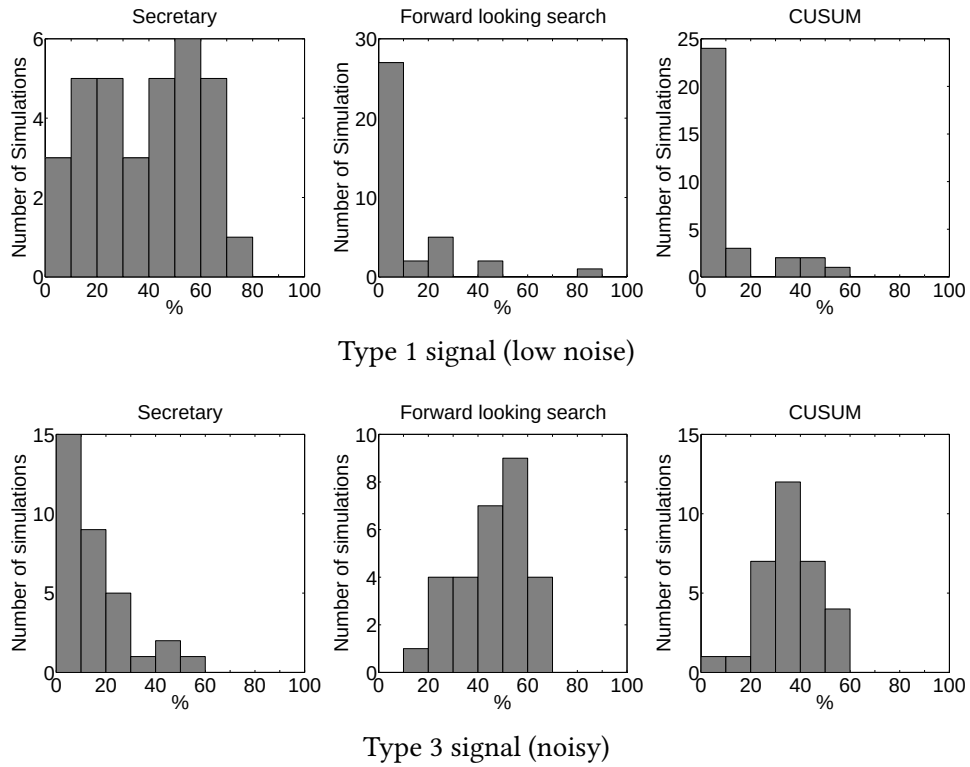


Figure 4.11: Distribution of the error in selecting the highest possible sample, without and with peak detection

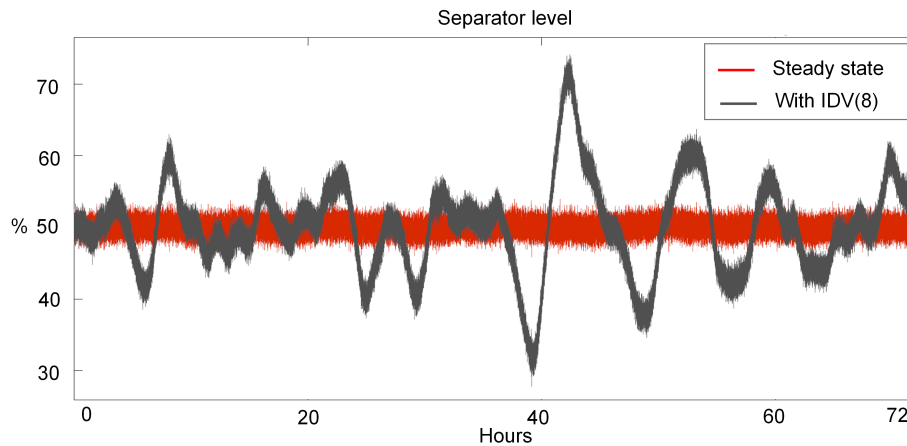


Figure 4.12: Impact of the IDV(8) disturbance on process measurement

4.5.4 Additional Considerations

After gaining access to the control system, the attacker faces a number of uncertainties. Therefore hitting at outright attack success might be challenging. This section discusses several considerations related to attack design and execution.

Impact of the Sampling Frequency

The sampling rate of a sensor signal changes its noise profile. In order to complete the analysis, we also investigated the impact of f_s . As anticipated, the lower signal sampling rate results in a lower error in selecting the best sample due to the reduced noise level (Table 4.5). However, the selected attack value is also lower compared with the attack on the signal sampled at a higher rate; this is due to the reduced number of high amplitude outlier samples (Figure 4.13). Depending on the control loop sensitivity, we could observe an increase in shutdown time of up to three hours.

	Error	TTS	TTA	NS
SP, n/e	4.33 %	10.73 h	28.38 h	37.72 %
SP, $n/\log(n)$	12.81 %	6.70 h	10.97 h	8.09 %
OT, $h = 2\sigma$	12.55 %	3.95 h	21.59 h	0.29 %

Table 4.5: Comparison between the approaches, $F_s=100$ s/h

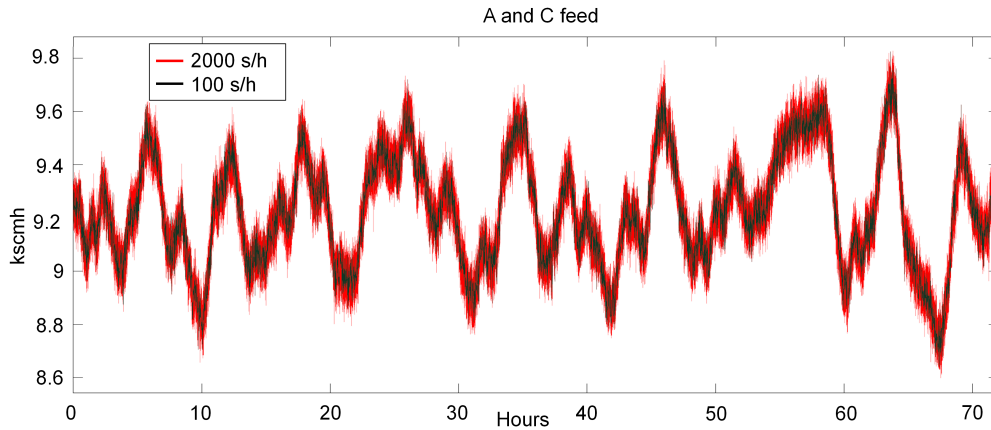


Figure 4.13: Impact of the sampling frequency on the sensor signal

Figure 4.14 demonstrates the dependence of the shutdown time on the sampling frequency of the signal in the case of the F_{strip}^{max} attack. As can be observed, even a slight increase in sampling frequency (until 250 s/h) gives the attacker 1.34 hours of advantage in achieving the attack goal.

Detection of Plant State Change

The reference value learned in the observation window is only valid for a particular plant state. In practice, a process periodically undergoes changes in its operating conditions due to updated setpoints, change in operating modes of production loads, disturbances, etc. The attacker needs to be able to detect such changes quickly in order to adapt her attack strategy to the new circumstances. Figure 4.15 demonstrates the detection of the A/C -feed ratio change using the CUSUM algorithm. With a threshold $h = 0.001$, the change can be detected in 8.5 min and with $h = 0.005$ in 12.5 min. Once the change is detected, the attacker can either reset the learning phase immediately or wait for some time to see if the process state keeps changing.

One of the challenges in detecting a state change in a plant is the uncertainty in the magnitude and location of its manifestation. In the case of the Tennessee Eastman process, most of the mode, load and step changes have explicit influences only on certain control loops while the changes in other control loops happen in a less distinctive way and may be unnoticeable to the attacker.

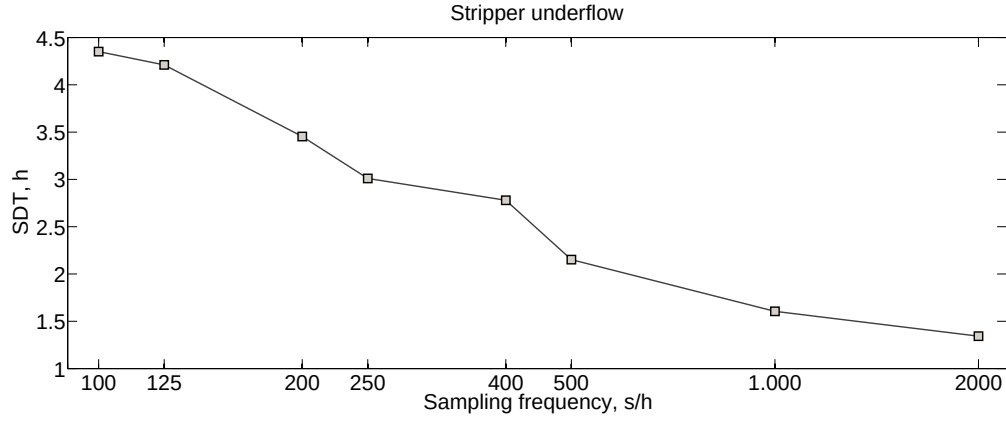
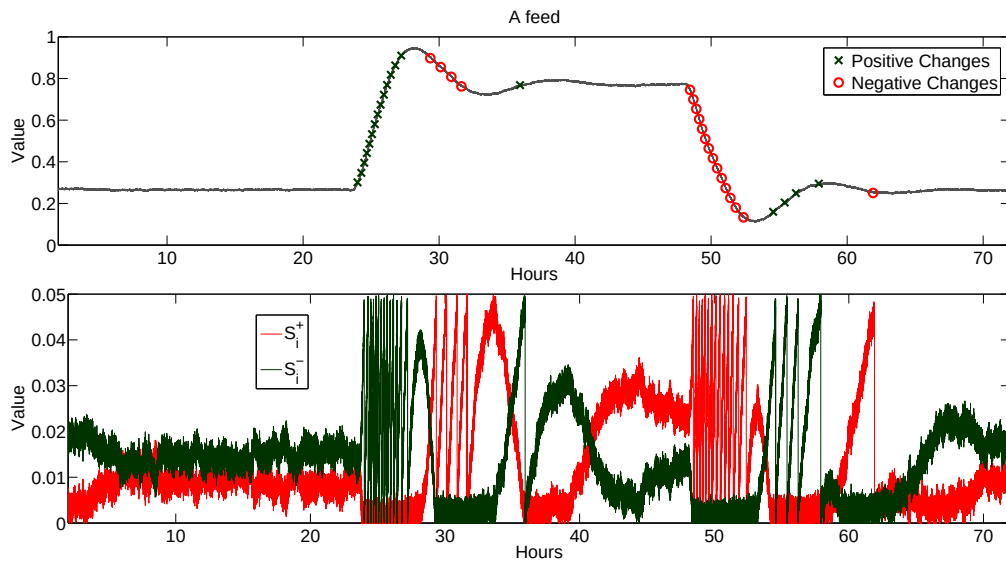


Figure 4.14: Impact of the sampling frequency on the SDT


 Figure 4.15: Detection of the plant state change, $h = 0.005$

Chaining DoS Attacks

Even after executing a successful DoS attack at steady state, it may take a long time to move the process to an unsafe state. In order to achieve a shorter shutdown time, the attacker would have to first disturb the process to cause greater deviations in the process measurements so that a better (higher or lower) attack value could be selected. Causing a plantwide disturbance might be difficult, but the attacker can “chain” two DoS attacks to accomplish her attack goal. For example, the F_{sep}^{min} attack on the separator underflow causes an oscillation effect on the separator level (Figure 4.16a). After 30 min, the separator level L_{sep} reaches 30%. The attacker can then use the CUSUM algorithm to detect the change and launch a DoS attack on L_{sep} when L_{sep} reaches its lowest value. In this way, shutdown can be reached in 3.43 hours compared with 12.03 hours if the attacker were to execute a direct attack on the separator level sensor at steady state. Similarly, the L_{sep}^{min} attack causes an immediate step change in the separator underflow (Figure 4.16b), which can be quickly and accurately detected and leveraged to launch a subsequent DoS attack. The attacker could also chain a DoS attack on a sensor with a subsequent attack on an actuator.

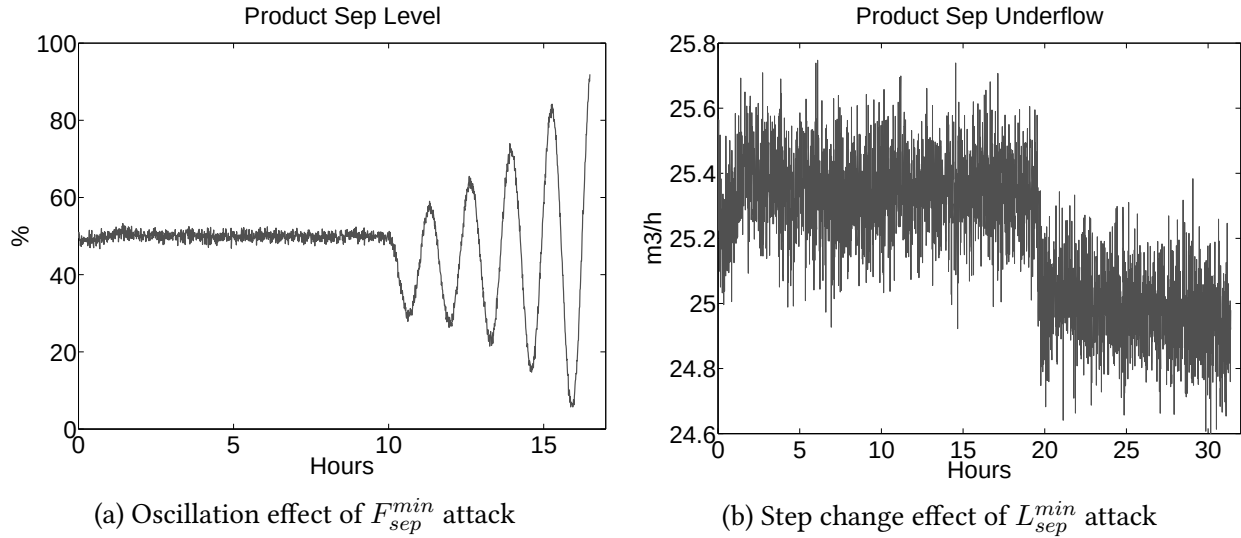


Figure 4.16: Examples of the possible stages for chaining attacks

4.6 Discussion

After accessing the process measurements, the attacker neither knows the maximum possible process variable range nor the sensitivity of the process to the magnitudes of the manipulations. More importantly, the attacker has no knowledge about the time constants of the individual control loops and the extent of disturbance propagation. This means that the attacker is uncertain about which attack value to choose and the period of time to carry out the attack. The latter is crucial knowledge, especially when planning concealment activities. Therefore, to maximize the impact and minimize the attack duration, the attacker should attempt to select the highest or lowest value possible from the observed data range. This strategy will be also useful in cases when the scaling or units of data are unknown, e.g., as shown in Figure 4.17.

434	1.070135	10.85.64.50	10.21.81.252	DNP 3.0	162 from 16 to 1024, len=255, Unconfirmed User
553	1.131345	10.85.64.50	10.21.81.252	DNP 3.0	112 from 16 to 1024, Response
740	1.447104	10.21.81.252	10.85.64.50	DNP 3.0	78 from 1024 to 16, Read, Internal Indications
749	1.510921	10.85.64.50	10.21.81.252	DNP 3.0	75 from 16 to 1024, Response

```

▷ Object(s): Binary Input With Status (Obj:01, Var:02) (0x0102), 40 points
▲ Object(s): 16-Bit Analog Input (Obj:30, Var:02) (0x1e02), 70 points
  ▷ Qualifier Field, Prefix: None, Range: 8-bit Start and Stop Indices
  ▷ [Number of Items: 70]
  ▲ Point Number 0 (Quality: Online), Value: 1678
    [Point Index: 0]
    ▷ Quality: Online
    Value (16 bit): 1678
  ▷ Point Number 1 (Quality: Online), Value: 1358
  ▷ Point Number 2 (Quality: Online), Value: 1760
  ▷ Point Number 3 (Quality: Online), Value: 1677
  ▷ Point Number 4 (Quality: Online), Value: 74
  ▷ Point Number 5 (Quality: Online), Value: 103
  ▷ Point Number 6 (Quality: Online), Value: 25
    
```

Figure 4.17: Data packet with process variables represented in unknown units

Characteristics of sensor signals in the same plant can be very dissimilar. Taking into account the amount of uncertainty involved when designing an attack, an attacker would be better off to

choose scale-independent and non-parametric algorithms. Although we have selected approaches with few tuning parameters, the attacker would still require a learning phase to roughly determine the types of signals, their distributions and the noise levels. The noise level play an important role when deciding on the most effective algorithm to selecting attack value and tuning certain parameters such as thresholds used in the Outlier Test and the CUSUM algorithm.

Sampling frequency turned to be one of the most crucial parameters that determines attack success. As mentioned in Section 4.5.4, f_s influences the noise profile and, therefore, can directly impact the selected attack values and shutdown times. Also, knowledge of the sampling frequency is crucial to determine the optimal smoothing interval; the lower the f_s value, the smaller the smoothing interval should be. Figure 4.18 shows the raw XMEAS(12) signal sampled at 2,000 s/h and its corresponding smoothed signal with $m = 250$. If the raw signal had been sampled at 100 s/h, its smoothed signal would have been significantly distorted.

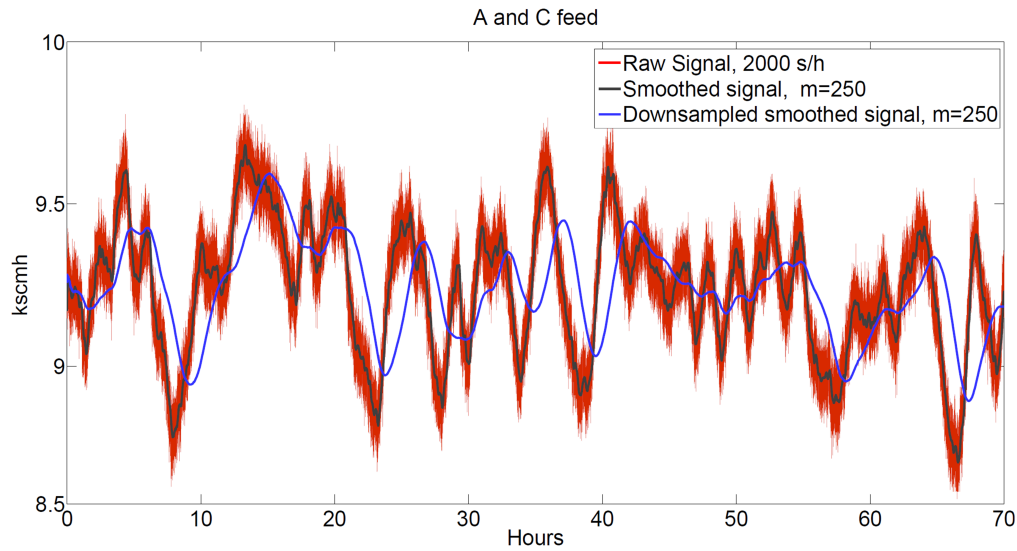


Figure 4.18: Impact of the sampling frequency on signal smoothing

Destructive capabilities in the cyber-physical domain predominantly exist in relation to a specific target. Thus, knowledge on process dynamics such as the results from Table 4.2 are only valid for a specific control architecture of the process. A separate analysis would have to be conducted for the other control schemes developed for the Tennessee Eastman process. The attacker could do excellent homework and design a good portion of the attack in advance. However, it would be still necessary to tune the attack to the individual dynamic behavior of the plant through reconnaissance activities such as stimulating the process by changing configuration parameters or modifying manipulated variable while observing the system’s reaction as, e.g., presented in [320]. From the defensive point of view, short-term process deviations caused by attacker “tests” can be detected by process-aware anomaly detection solutions [45, 223].

While in this chapter we considered attack scenarios in which an attacker aims at causing significant unwanted impact on the physical process (e.g., unsafe state), DoS attacks are also effective for execution of so-called “economic damage” attacks. Figure 4.19 illustrates various cost elements in relationship to process efficiency. As can be seen, even insignificant deviations from the optimal operating conditions may impact plant profitability. Therefore also non-strategic and short-time DoS attacks can have an unwanted effect on plant operations.

A large body of literature is dedicated to understanding the best strategies for causing DoS attacks on various communication protocols [53, 110, 203, 207, 208]. DoS can rarely be achieved

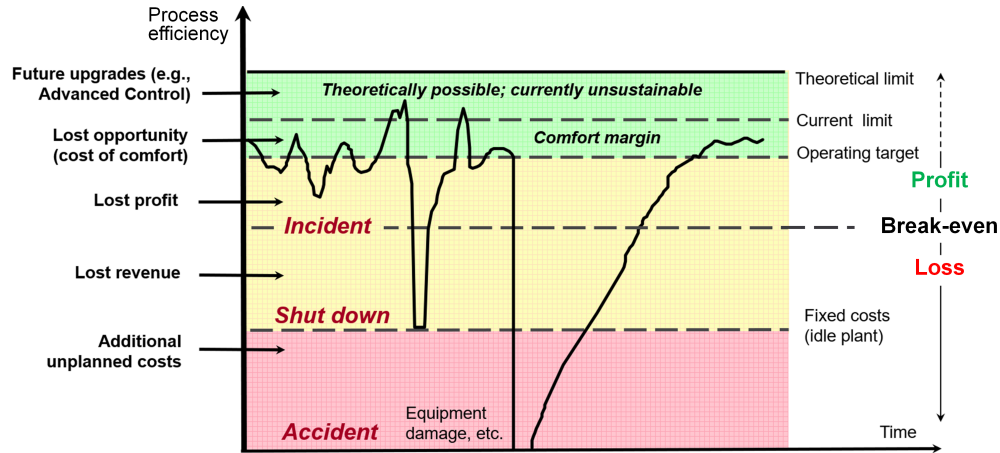


Figure 4.19: Relationship between process performance and plant profit [184]

instantaneously. Typically it takes some to achieve a long enough packet delay or ensuring no successful packet retransmission. Therefore, during practical Stale Data attack implementation it might be required to optimize (decrease) a delay between selecting stale data and achieving DoS to maximize attack impact. A significant advantage of DoS attacks is that complete parsers for proprietary protocols are not required to launch an attack.

The issue of stale data may become especially prominent with the increasing usage of wireless communications such as 5th generation mobile radio (5G) in plants (Figure 4.20). Even though 5G technology significantly improves important parameters for industrial networks such as latency and jitter compared to previous generations of cellular networks, the attacker may strategically jam communication links in a temporary fashion or increase latency.

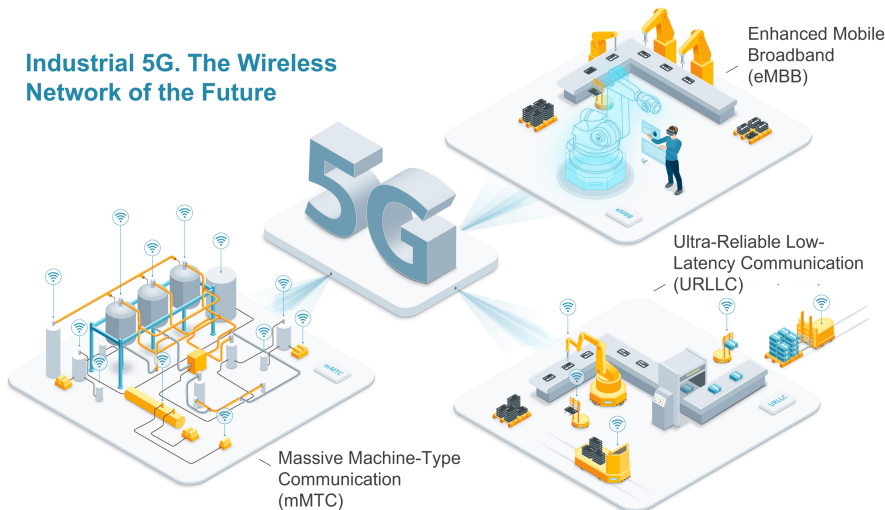


Figure 4.20: Usage of wireless 5G communication in an industrial plant [285]

4.7 Related Work

Physical processes and their states are inherently time dependent. Researchers have previously demonstrated the important role of timing parameters in cyber-physical security. Genge and

Siaterlis [112] have shown that process-aware segmentation of a control network increases survivability and extends time to process shutdown. Also, Genge et al. [111] experimentally demonstrated that PLC task scheduling and speed of valve have a major impact on the outcome of remotely executed attacks. Criticality of the synchronization to the stability of the electric power grid is discussed by Amelot et al. [14]. It was shown that accurate timing is critical when aggregating data from multiple sources and supplying them to computational models. Delays in receiving certain data may significantly impair situation intelligence and increase risks of adverse impacts.

In our work we consider an attacker who carries out a DoS attack for a certain period of time, needed for moving the process into a desired state. Larsen [190] investigates the impact of the strategic short-term delays of process measurements to destabilize control loops. Specifically, he showed how data packet delays invalidated the dead time constant and subsequently invalidate the tuning of the PID controller. The same author applied a triangle approximation algorithm for detecting certain process conditions in sensor signals in [189]; this method can be also used for sensor signal peak detection.

A comprehensive overview of the recent research efforts on networked control systems under denial-of-service attacks can be found in [50]. It was concluded that attack-resilient control and communication systems and prevention and detection approaches for DoS attack have not yet been sufficiently explored. Few works exist on detecting DoS attacks in the Tennessee Eastman process [158, 166]. Both approaches employ general data analysis strategies which makes such anomaly detection approaches applicable to a wider range of physical process. DoS attacks on other types of cyber-physical systems such as cars and underwater sensor networks can be found in [171] and [332] respectively.

Different approaches to identifying parts of the process and equipment which are more sensitive to cyber attacks were studied in [5, 135, 315]. Haller and Genge [135] used the change in the standard deviation of observable variables as a metric to measure the sensitivity of individual control loops to a set of attacks. Using cross-correlations, the authors determined the groups of observed variables with similar sensitivity to prioritize deployment of a limited number of IDS sensors. The experiments discussed in [5] had shown that it is more advantageous to the attacker to jam (wireless) communication links at Level 0 of the Purdue network reference architecture than at Level 1 based on damage potential and detectability metric. Wang et al. [315] applied a Monte Carlo-based framework to a model of a lead-cooled fast nuclear reactor to identify components which are most vulnerable to cyber attacks. Using a safety margin metric the authors compared impacts caused by attacks on sensors, actuators and PID controllers. In contrast to our work, attacks on actuators were found to have the most detrimental effect. This is because valve failures were modeled as a fail-stuck to a random magnitude of actuation from a uniform distribution. In real world, valve position is kept close to its typical operational value most of the time to minimize process fluctuations. This is why DoS attacks on actuators have less detrimental impact than attacks on sensors.

In work closest to ours, Cárdenas et al. [44] used a simplified model of the Tennessee Eastman process to compare sensitivity of the control loops to data integrity and DoS attacks on sensor signals. The authors were not able to bring the TE process into an unsafe state with DoS attacks and concluded that they have limited impact. The impression was likely caused by the lack of variations in sensor signals. The simplified TE model includes only steady state and does not include sensor noise, therefore the attack (stale) values were very close to the set point. In our work we have shown, that strategic execution of DoS attacks on signals with variable dynamic are as effective as data integrity attacks.

4.8 Conclusions

Cyber-physical systems are real-time applications which depend on the timely availability of process state information and punctual dispatch of control commands. By taking advantage of sensing and control actions scheduling, the attacker can orchestrate the dynamic behavior of the physical process to her advantage. While conventional IT network security measures, if implemented, prevent highjacking and modification of communication traffic (data integrity attacks), they do not deal with the timeliness of data packet arrival caused by strategically executed DoS attacks. We called this class of attack Stale Data. In this chapter we studied the effect of DoS attack on a control system and the dynamic behavior of an underlying physical process.

Firstly, we showed that the effect of DoS attacks is similar to data integrity attacks with the only difference being in the range of attack values equal to the range of process values observed in the legitimate traffic. Although this is not necessarily a limitation, the range of attack values can be expanded by chaining DoS attacks on different control loops. Secondly, we investigated the effect of DoS attacks on individual control loops. By adopting a risk assessment approach and a metric of Safety Shut Down time, we numerically evaluated various attacker strategies when selecting the maximum attack value on sensor and actuator signals. We used the Tennessee Eastman challenge process to illustrate our approach. However, our methodology can be applied to any cyber-physical system.

Our empirical results reveal the execution of cyber-physical attacks is associated with a great deal of uncertainty about the behavior of individual control loops under attack. It is not evident which attack value to choose, for how long to carry out the attack to achieve the desired impact and what would be the extent of process disturbance propagation. The latter is crucial knowledge, especially when planning concealment activities. Additionally, attack strategies deliver radically different results when applied to different types of process measurements. To reduce the need for preliminary reconnaissance activities the attacker may opt for approaches to signal processing with minimal tuning parameters. Last but not least, we observed that attacks on actuators are less effective than attacks on sensor signals due to much lower variability in the manipulated variables.

With the increasing usage of various wireless communication (including 5G) and drone technology becoming widely available to attackers, jamming attacks with stale data impact may become of significant concern. Sensitive control loops should be especially closely monitored for delays, stale data should be discarded and missing data compensated with alternative sources of data, e.g., with MPC data. Threat modeling for industrial equipment design must include stale data attack scenarios.

Chapter 5

Data Veracity in Process Control Systems

Don't trust your inputs.

“Writing secure Code” by Howard &
LeBlanc

5.1 Introduction

With the progressive digitization of industrial control systems, process data collection, processing, storing and presentation became an essential task of the control infrastructures with process data becoming one of the “assets” to manage. Control system applications supervise all of the functions and operations in a plant by taking in real-time data from the field instrumentation and using the trends extracted from the process data to make decisions about each process unit performance. When speaking about software applications, they are seen as applications that run on top of an untrusted IT infrastructure. Input validation is performed to ensure that only properly formed data is entering the application workflow, preventing the unexpected outcome of code execution. Physical processes similarly rely on the process data being appropriate and not harmful to the physical application.

Security properties should fit the application's needs. For a long time it was thought that if data security policies were guided by the Confidentiality, Integrity, Availability triad, the data was ‘secure’. With the change in the volume of data being produced and the way how and where data was stored and used, the CIA model began to be seen as limiting. In 2002 Donn Parker extended the CIA model to better describe data security requirements as shown in Figure 5.1 [248].

Parker's six attributes of information security are non-overlapping in that they refer to unique aspects of information with utility or data usefulness being the only property that is not necessarily binary in nature. Not only utility can have degrees of usefulness, it is also highly contextual and can only be evaluated in relation to a specific application.

In process automation, field devices such as sensors are considered to be monolithic units that are fully *trusted* [237] and the data originated by the devices are correspondingly considered *trustworthy*. In many cases the controller and operator can observe a physical process only through process measurements, therefore their accuracy is critical. As discussed in Section 2, acquired sensor signals pass through a variety of functions that process them in a variety of ways such as amplification, scaling, conversion, filtering, aggregation and normalization to name a few. Furthermore, data sources are combined through computation formulas prior to being consumed by additional control logic and applications. In essence, data processing is conducted to provide

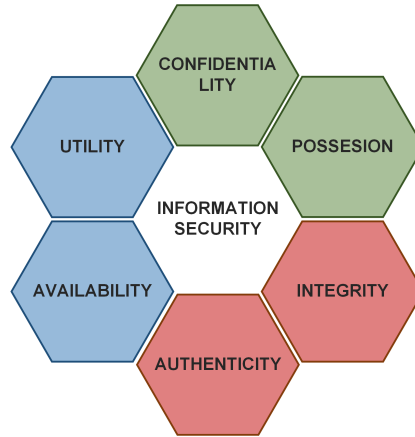


Figure 5.1: The Parkerian hexad

usable, i.e., actionable information, based on the requirements defined by data consuming circuits/devices/applications at each stage in a data chain. Any error in data acquisition or processing along data routes harbors the potential to degrade and even lose visibility of the process state. This is why the understanding of data sources and their pathways is essential to the assessment of the undesirable impact on process operations, caused by errors or intentional manipulation of data streams. In process control, data condition/content is addressed in the context of “data quality” or “data reliability”. Telemetry engineers are frequently responsible for this task.

Data utility refers to the data accuracy and usefulness as a result of its transformation within the computing infrastructure. Examples of data utility and the impact of its violation in the context of industrial infrastructures are discussed in Appendix C. The utility does not address how trustworthy or accurate the source of the data is, and whether the information extracted from such data is correctly representing the reality. Process data are originated in the physical process and measured by the sensors. It would be prudent to establish methods able to determine if the data received truthfully capture the attributes of the physical world relevant to the application under control. This leads to a new security requirement called veracity, or trustworthiness of data. The relationship between data veracity and data utility is shown in Figure 5.2.

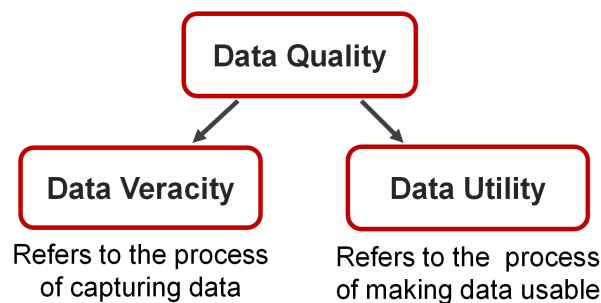


Figure 5.2: Data quality in process control systems

5.2 Data Veracity

The word veracity derives from the Latin adjective *vērāx* – “truthful”, which in turn comes from the earlier *vērū* – “true”. Initially, the property of veracity gained its popularity in the area of

criminology – the ability to detect whether a witness is veracious or not [200]. The term veracity is used both in relation to actors such as witnesses and their statements. Borselius extended veracity to the property of mobile agents namely that they will not knowingly communicate false information [31].

We refer to statements about aspects relevant in a given application domain as *assertions*. Assertions may be true or false. In this thesis we use the definition of veracity proposed in [120].

Veracity: *The property that an assertion truthfully reflects the aspect it makes a statement about.*

Restating, data veracity is the degree of accuracy or truthfulness of a data set. Veracity is not a property guaranteed by any of the familiar IT infrastructure security services. Veracity refers to aspects outside the IT infrastructure: the adversary is not an entity launching an attack in the infrastructure but an entity making false assertions. The data are already false when passed to the infrastructure. “Trusted” sensor data just because they have been digitally signed do not help when the sensor readings are incorrect or have been manipulated before being submitted to the communication protocol stack. Veracity is a new security property relevant at the application level.

There is an old engineering aphorism that says “*one must measure a process in order to control it*”, which is still as current today as it was in the past. A corollary to this saying is that trustworthy process measurement information is required so that users feel comfortable enough to risk making tighter control improvements [227]. For data to be regarded as trustworthy, particularly to support the decision-making process, it is vital that data accurately reflect the reality. However, data veracity is a complex property that deals with uncertainties in data brought about by a diversity of causes with biases, noise, abnormalities and missing values being a few examples. Enforcing veracity can be challenging. Quoting the words of Prof. Vijay Varadharajan in [28]:

Every physical and virtual device in the IoT infrastructure generating huge quantities of data presents immediate and direct consequences. Just because data are accessible doesn’t mean it’s trustworthy or reliable for making decisions, or even ethical to access and use it.

To secure an application, one thus may have to go beyond securing the infrastructure and in addition provide mechanisms for detecting false data. In this section we first examine examples of how data veracity can be impacted in process control applications before focusing on a specific adversarial example of veracity compromise. We then propose an approach for detecting non-veracious sensor signals in near-real time.

5.2.1 Violation of Data Veracity

In many cases the operator can only observe the process through sensors and must have faith that the sensors describe the true underlying physical reality. This faith can be used by an attacker even when error handling is in place and used to mathematically solve the error in the system. For example, in the power grid complementary disturbances caused in adjacent nodes are canceled out as a residual error in a state estimator.

Violation at the Process Level

The lowest level of manipulation possible is the actual physics of the process itself. The most intuitive attack scenario is a manipulation of the environment around a sensor. This attack vector

is especially applicable to sensors that can be “approached” by the attacker such as weather and ambient sensors or proximity and object detection sensors. Increasingly, data veracity is becoming of concern in the automotive sector. Modern cars are equipped with a large number of sensors some of which are exposed to the outer surroundings without physical protection. Shoukry et al. [283] showed that by modifying the physical environment around analog sensors such as Antilock Braking System (ABS) it is possible to alter wheel speed measurements and potentially induce life threatening situations. ABS relies on magnetic-based wheel speed sensors which are exposed to an external attacker from underneath the body of a vehicle. By placing a thin electromagnetic actuator near the ABS wheel speed sensors, an attacker can inject magnetic fields to both cancel the true measured signal and inject a malicious signal, thus spoofing the measured wheel speeds.

Industrial processes are typically located within the protective boundaries of a physical fence or similar and therefore are not easily approachable by attackers. However, it is possible to achieve an adversarial impact on data veracity by creating special conditions in the physical process itself via remote attacks. In continuous processes, even if two sensors are segregated electronically, the process physics may connect them creating a “data flow”. When the logic of a field device operates on a particular datum, that datum may actually be an aggregate of other data even if that value was directly measured from the process. In such case, the “unseen” process data may have a negative impact on the control loop performance. Consider a pressure measurement. Depending on the process, changes in pressure may be the result of temperature, flow, volume, or reaction speed. A simultaneous change in temperature and volume may be incorrectly interpreted as a change in reaction speed. Incorrect interpretation of unmeasured quantities (here: reaction speed) is a frequent concern in process control.

Consider another example of a process unit that produces ammonia with pressure in a vessel being maintained with an inflow pump. Maintaining the right ammonia pressure is critical to the financial health of the plant, therefore its control loop is closely monitored. A second pump is responsible for the outflow of ammonia from the vessel but is not considered critical to the plant economy. It may be possible to set up a standing wave between the two pumps that has a direct impact on the ability of the first pump to perform its function. In that case, the unseen state of the second pump is critical to the functionality of the first pump even though there are no electronic data flows between these two control loops. A variation of this attack scenario has been implemented on a realistic hardware setup to degrade the performance of an analog pump as described in Appendix E.3. By modifying the position of the inlet valve, an attacker was able to cause a cavitation process which impacted the working of the pump up to full stoppage of the flow.

Although data veracity attacks at the layer of the physical process are hard to detect and diagnose, they are also much harder to execute from the design and attack reliability standpoints. During such attacks both the operator and the adversary operate in conditions of limited visibility of the underlying process physics. Depending on the attack scenario, prior testing on a mockup system may be required to determine the exact attack parameters.

Violation at the Sensor Level

Input devices such as sensing devices are inherently susceptible to attacks with unvalidated, unwanted inputs. Attackers can exploit the physics of materials to fool sensors into becoming unintentional receivers of malicious signals to manipulate sensor output or induce intentional errors. This class of attack is called a transduction attack [108]. This is one of the most frequent attack scenarios covered in the CPS security research literature. For instance, Tu et al. [305] implemented an attack on temperature sensors that exploits an unintended rectification effect in analog amplifiers that can be induced by injecting electromagnetic interference (EMI) at a certain

wavelength. The attack scenario bypassed conventional noise filtering and generated a controllable DC (direct current) voltage offset at the analog-to-digital converter (ADC) input. Due to the altered offset, the resistance of a resistance temperature detector (RTD) was converted into a wrong albeit “valid” value within the operating range. As stated by the researchers, “from meters away or an adjacent room, an attacker could trick the internal control system of an infant incubator to heat or cool to unsafe temperatures.”

In another example, Bolshev et al. [30] exploited the ADC’s input signal sampling process in industrial equipment. Specifically, the authors discussed a scenario of injecting a specially crafted analog signal which would be converted into different digital values by the serially connected equipment on an analog line. In the architecture depicted in Figure 5.3, a PLC responsible for the regulatory control sends an analog control command to the actuator (e.g., a motor) whereas a so-called “safety PLC” performs checks of that command for unsafe conditions. It was shown that it was possible to generate an analog signal that would be converted into different digital values by different equipment. For instance, the actuator converted the analog signal into 1.5 V (ON command) and the safety PLC output 0 V (OFF command on the HMI). If such an attack is successfully executed on the plant floor, the operator would lose awareness of the true state of the process and could make wrong and potentially harmful control decisions. Similarly, the safety protection would not engage due to a wrong notion of process state. The perceived threat of the transduction attacks to the industrial-grade sensing equipment has grown such that in one instance U.S. ICS-CERT warned manufacturers of the related hardware design flaws in a security advisory [73].

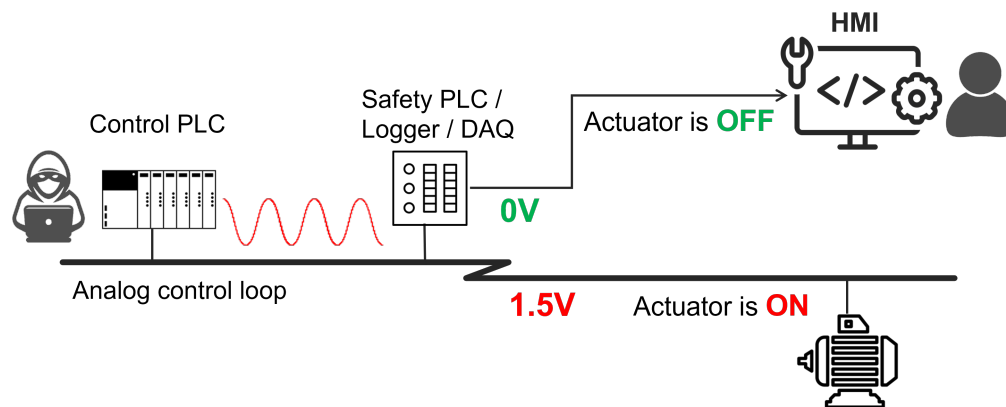


Figure 5.3: Threat scenario for analog control loop [30]

While the implementation of the above scenarios requires close proximity or physical access to the target environment for attack execution, the evolution from simple analog and digital sensors to sophisticated “smart” transmitters makes it possible to execute an attack on data veracity directly on the sensing device, and with that to undermine the assumed trust in field equipment. Modern smart sensors are equipped with microprocessors and IP-based communication capabilities making them attractive targets for remote and supply chain attacks. Violation of data veracity directly on the sensor through its compromise is a scenario considered in this chapter.

5.2.2 Ensuring Data Veracity

The definition of veracity does not differentiate between assertions that are false on purpose and assertions that are false by accident. This distinction does not matter for the application; the application would react to a false assertion in the same way in both cases. The distinction matters

in the design of countermeasures. For instance, redundancy can be used for accidental violations but not for malicious violations. In this regard, Borselius [31] commented that veracity cannot be effectively enforced in mobile agent systems as the required redundancy for such a system is likely to make the system useless.

For security purposes, data veracity can be achieved in a few ways. First, if the environment warrants a sufficient degree of physical protection tamper-resistant “trustworthy” sensors can be deployed, i.e. reliance on physical security. Second, hardware and/or software-based validation of the input signals may help to reject malicious signals or filter out harmful signal components (e.g., high frequencies). Third, if the environment and input signals cannot be easily controlled, countermeasures can take the form of plausibility or consistency checks on the received sensor inputs.

Redundancy and consistency checks such as majority voting have been used for detecting accidental sensor failures. Since the defenses against intentional attacks cannot be built on the basis of statistical independence, countermeasures may take the form of plausibility and/or consistency checks. In this case, models of the physical space under observation are used to judge to which extent individual sensor readings are consistent with the overall state of the system derived from all the readings. It is possible to further model the relationship between different aspects of the physical process (e.g., temperature and pressure) in order to detect spurious sensor readings or flag implausible readings as suspicious. Changes in the plant configuration are not required to implement such countermeasures which makes them practical.

5.3 Sensor Signal Spoofing

If achieving the desired physical impact takes more time than the expected operator’s response, the attacker may need to spoof the process state in an intelligent way for the duration of the attack to prevent the operator from taking corrective measures and bringing the process back to its good state. Additionally, if the operator attributes the process behavior to unnatural causes, she can initiate an immediate incident investigation. Spoofed process measurement will then significantly obstruct the forensic investigation and root-cause analysis.

Believable falsification of sensor signals is a matter of science as much as art. In theory, it is possible to falsify a sensor signal that would be undetectable given enough time and CPU power. In real world scenarios the attacker is constrained by time, needed engineering efforts and other factors. The core computers of an industrial control system are capable of significant computations, but the individual measuring instruments are almost universally based on small microcontrollers that have limited computing and storage resources.

The techniques used to spoof a signal are, generally speaking, a tradeoff between storage and computation. At one end of the spectrum, an attacker can use a “record-and-playback” approach. The sensor signals are recorded at a convenient time and then played back during the attack. Depending on the type of the process (continuous or batch), data from a previous day or a previous batch is recorded and played back during the attack. This approach to spoofing was used in Stuxnet [187]. However, this method may require a large amount of storage depending on the volume of process data that needs to be spoofed. At the other end of the spectrum, the attacker can build a model of the process and then run that model in a process. This is an onerous task that requires significant computational and storage resources [314]. Also, it requires a large body of detailed knowledge about the process that the attacker has to gain directly from the process she is trying to spoof.

As an alternative, we consider a spoofing technique that can be run on the small microcontrollers used in smart sensors. It is based on a set of lightweight real-time algorithms first introduced

by Larsen [189]. In his approach, Larsen builds a small model of an individual sensor signal, which allows accurate recreation of sensor signal noise (Runs Analysis algorithm) and process measurement dynamics (Triangle Approximation algorithm) with the primary goal to deceive human operators as well as pass basic sanity/validity checks of sensor values such as statistical uncertainty validation [141]. We implemented Larsen's algorithms and applied them to sensor signals in the TE process model. We verified the suitability of the approach and parametrized the sensor spoofing algorithms to make them applicable to spoofing different types of sensor signals.

5.3.1 Sensor Signal Noise Reconstruction with Runs Analysis

The technique described in this section is used to replicate the noise floor of a sensor signal. The design goal is to produce sensor noise that is believable to a human observer in the smallest footprint possible. It means accurately reproducing noise amplitude and the number of visible spikes and gaps from the original data.

The technique is based on Bradley's Runs Test for comparing random sequences from his collection of the Distribution-Free Statistical Tests [33]. The algorithm counts the number of runs in two pseudo-random sequences and then compares the distribution of those runs as a measure of the similarity of the underlying randomness. The algorithm counts the runs in the sensor noise treating it like a pseudo-random sequence to build a distribution. It then uses that distribution to generate sensor noise that mimics the original. The approach works across a wide variety of sensor signals and does not require the attacker to have *a-priori* knowledge of the noise parameters.

The technique is best described by example. Consider the following sequence as if it was a list of consecutive samples from a sensor:

[33.47, 34.73, 37.77, 36.56, 35.31, 30.15, 39.64, 35.08, 37.22, 34.18, 30.84, 38.47, 35.9, 31.12, 32.19, 30.39, 35.39, 31.08, 37.15, 34.24, 36.63, 37.4, 31.49, 37.11, 38.27, 39.97, 34.16, 38.85, 32.6, 32.69, 32.64, 35.41, 35.03, 39.33, 37.45, 31.69, 38.27, 35.59, 34.08, 34.5, 38.66, 38.78, 34.28, 35.81, 37.64, 30.84, 36.97, 34.99, 30.01, 30.31, 38.91, 38.45, 31.57, 34.46, 33.03, 31.19, 34.69, 36.89, 35.64, 39.18, 36.45, 32.99, 31.66, 38.62, 36.76, 35.93, 35.28, 36.58, 35.37, 33.57, 30.16, 33.77, 33.88, 33.81, 37.65, 32.41, 33.44, 38.27, 33.06, 38.74, 35.62, 34.95, 35.65, 32.35, 39.38, 30.62, 38.47, 35.38, 39.1, 30.11, 37.73, 39.8, 36.52, 33.4, 39.54, 36.57, 37.36, 32.85, 38.66, 35.9 ...]

The technique first counts the number of increasing or decreasing values in a row called "runs up" or "runs down". The subsequence [33.47, 34.73, 37.77] increases over three consecutive numbers and can be described as a "run up of 3" or simply "3". The next subsequence [36.56, 35.31, 30.15] decreases over three consecutive numbers compared to the last number 37.77 from the previous run and can be described as a "run down of 3" or simply "-3". Note, that the first subsequence does not have a "reference number" and its run is computed in relation to the first and last numbers in the subsequence. The run can be further described by the total distance traveled. The subsequence [33.47, 37.77] travels $37.77 - 33.47 = 4.30$ units and can be described with the shorthand (3;4.3). The subsequence [36.56, 30.15] travels $37.77 - 30.15 = 7.62$ units and can be described with the shorthand (-3;7.62). The original sequence can be converted into the runs notation as follows:

[(3;4.30), (-3;7.62), (1;9.49), (-1;4.56), (1;2.14), (-2;6.36), (1;7.63), (-2;7.35), (1;1.07), (-1;1.80), (1;5.00), (-1;4.31), (1;6.07), (-1;2.91), (2;3.16), (-1;5.91), (3;8.48), (-1;5.81), (1;4.69), (-1;6.25), (1;0.09), (-1;0.05), (1;2.77), (-1;0.38), (1;4.30), (-2;7.64), (1;6.58), (-2;4.19), (3;4.70), (-1;4.50), (2;1.83), (-1;6.80), (1;6.13), (-2;6.96), (2;8.90), (-2;7.34), (1;2.89), (-2;3.27), (2;5.70), (-1;1.25), (1;3.54), (-3;7.52), (1;6.96), (-3;3.34), (1;1.30), (-3;6.48), (2;3.72), (-3;1.47), (2; 5.86), (-1;5.21), (1;5.68), (-2;3.79), (1; 0.70), (-1;3.30), (1;7.03), (-1;8.76), (1; 7.85), (-1;3.09), (1;3.63), (-1;8.99), (2;9.69), (-2;6.40), (1;6.14), (-1;3.09), (1;0.79), (-1;4.51), (1;5.81), (-1;2.76) ...]

The described procedure is applied to a certain number of consecutive values in the so-called *learning phase*. The average distance moved for each run length can be described with a single distribution (Figure 5.4a). This distribution describes the dynamic nature of the input sensor noise and can be used to generate sensor noise that looks similar to real noise for the human eye. The algorithm constructs the noise as a series of line-connected segments from the distribution where the X value of the line segment is the length of the run (Figure 5.4b) and the Y value of the line segment is the average distance traveled. Each line segment is not equally represented in the input data and should not be equally represented in the spoofed data. Thus, a run of +2 is far more common than a run of +5 in most data sets. The ratio of +2 to +5 can be determined by simply counting their occurrences in the input data set.

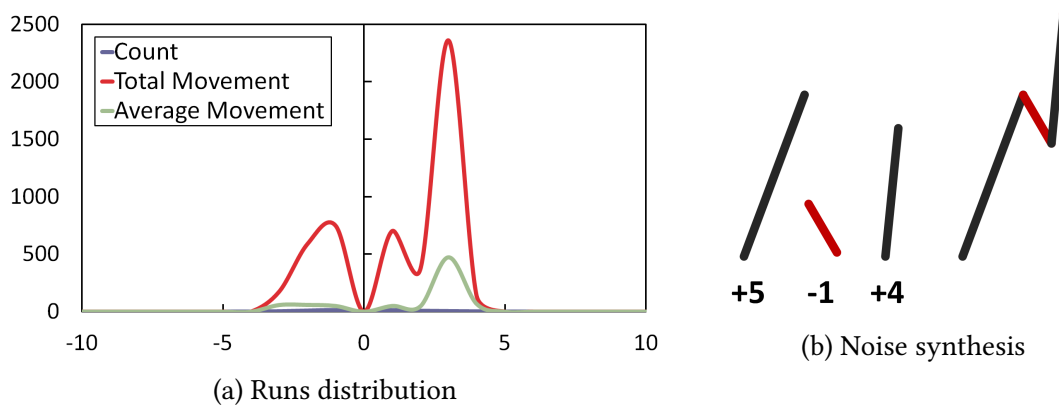


Figure 5.4: Attributes of Runs Analysis

The noise generation algorithm is simple. A line segment is chosen from either the positive half or the negative half of the distribution at random. The random choice is weighted so that the ratio of +2 to +5 is maintained. The random noise should not drift too far from the sensor reading shown to the operator. This is accomplished by choosing from the positive runs when the value is lower than desired and the negative runs when the value is higher than desired.

Figure 5.5 visualizes the results of the Runs algorithm. The original noise shown in Figure 5.5a is used as the training set. As can be observed in Figure 5.5b the noise features important to the human eye such as noise floor width, distribution of spikes and gaps are generated in a believable way. The complete pseudo code of the Run Analysis algorithm is given in Appendix D.1. In exceptional cases, when high-precision spoofing is required, measuring Signal-to-Noise Ratio (SNR) can further help with parametrizing the algorithm to ensure higher accuracy. The described approach to sensor noise synthesis can be implemented in a small code footprint. Larsen optimized the assembly code for the ARM microcontroller to fit both the code and data for the entire algorithm into approximately 400 bytes [189].

5.3.2 Sensor Signal Dynamics Reconstruction with Triangle Approximation

While sensor noise is a result of process physics and imperfections of the sensing hardware, the feedback mechanism of control loops induces dynamic process behavior that is visible to the operators through fluctuations in sensor signals. In a steady state, many sensor signals do not deviate much from a set point and look more like a flat line with a sensor noise (Figure 5.6a). If the

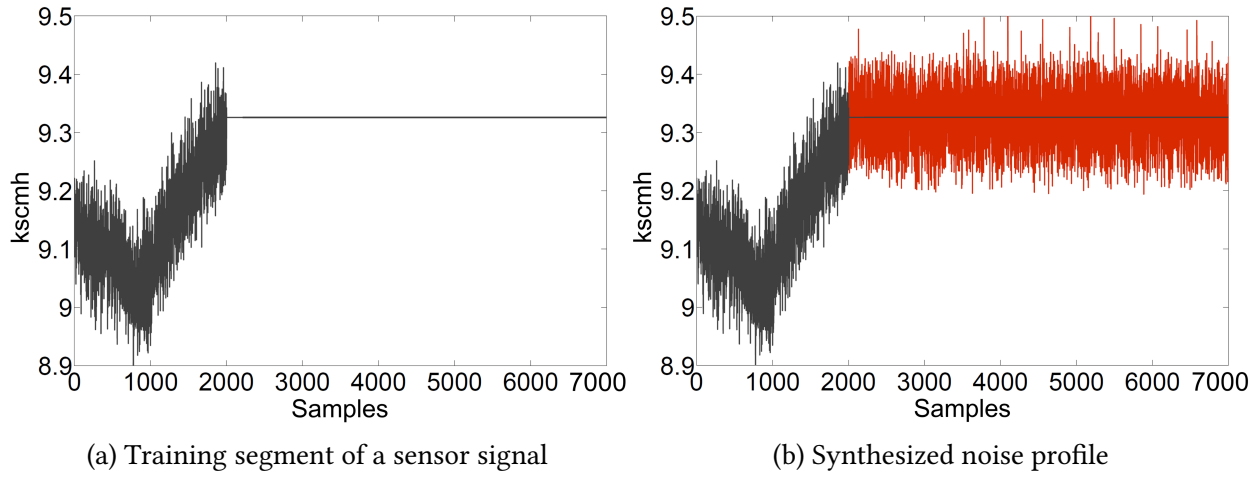


Figure 5.5: Reconstruction of sensor signal noise with the Runs algorithm: learning and synthesis phases

process under observation is naturally dynamic or the operator changes operational settings and moves the process into a new state, the attacker will need to re-construct the believable dynamic behavior of sensor signals.

As noted above, the microcontrollers used to measure process state in field equipment have limited processing power. Even if it would be possible to build a simplified model of the process to reconstruct the dynamic behavior of the process, such models are usually too complex to implement on the limited resources available in smart transmitters.

The raw units coming from the sensory package may be in an arbitrary scale as well as measured in various units. Therefore, the attacker needs a scale-independent technique for reconstructing dynamic sensor signal behavior. In our approach, we approximate the dynamic nature of the physical process with line segments. Figure 5.6b shows an arbitrary sensor signal with a series of line segments fitted over the top. The line segments provide a sufficiently accurate estimation of the measured process variable.

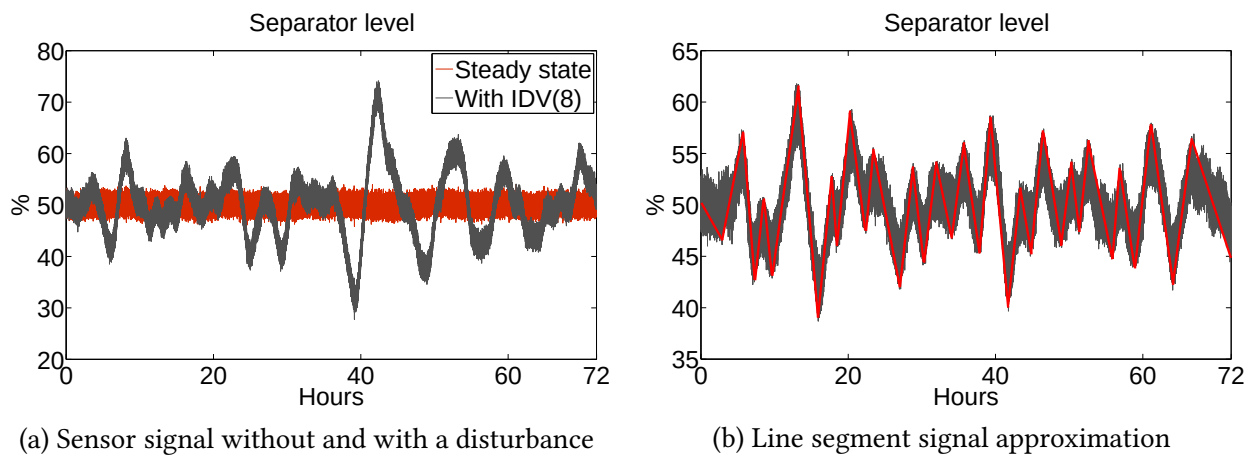


Figure 5.6: Extraction of the dynamic sensor signal trend with line segment approximation

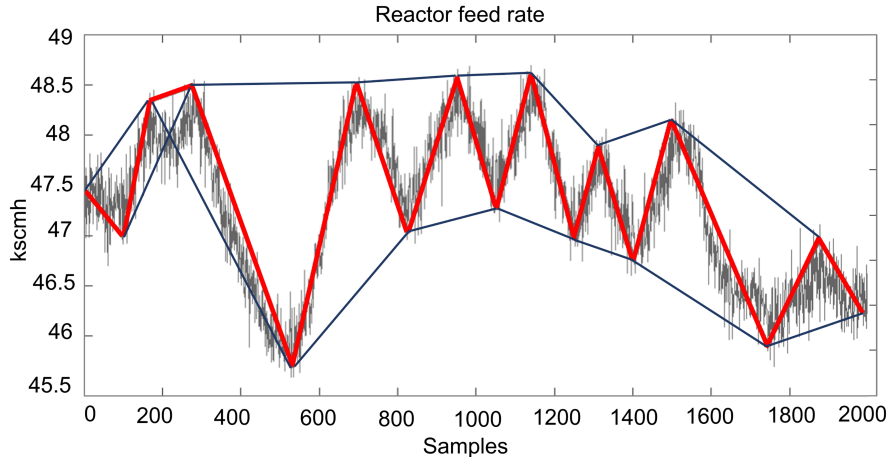


Figure 5.7: Illustration of a sensor signal approximation with a strip of triangles

Conceptually approximation is based on a triangle starting at the vertex declared at the first value and growing to the right. Two rays are sent out from the vertex until too many points fall outside the triangle described by the two rays. In that case, a next vertex is declared, the angles of the rays are adjusted and the process starts over. The extracted line segments can then be converted into a triangle strip as shown in Figure 5.7. The complete pseudo code of the Triangle Approximation algorithm is given in Appendix D.2. The algorithm is adaptive in the sense that it will generate more line segments when the data are changing rapidly and fewer line segments when the data are level. Depending on the number of line segments needed to be stored in memory, the triangle algorithm requires approximately 700 bytes of combined code and data [189].

Inherently noisy signals can make use of the triangle algorithm difficult, yielding unsatisfactory results. Therefore, very noisy signals must be smoothed first to filter out high-frequency noise (short-term fluctuations) and highlight underlying long-term trends. To smooth the signal we use a simple “moving average”, i.e., the mean of the previous n data points. The correct smoothing parameter must be determined by the attacker in the target environment depending on the sampling frequency of the signal and the noise level of the discovered sensor signals.

5.3.3 Construction of Spoofed Sensor Signals

We apply the above algorithms to spoof sensor signals from the TE process. As mentioned in Section 3.2.1, sensor signals in the TE process include realistic models of signal noise and can be roughly divided into four distinct groups depending on their noise level and the dynamic shape of the signal (Figure 3.2).

Figure 5.8 demonstrates the steps of the proposed real-time signal spoofing approach in the example of a Type 3 signal (signal with few distinct peaks and significant noise level). The spoofing process starts with a learning phase when the attacker determines sensor noise characteristics, extracts the dynamic behavior of the sensor signal in the form of line segments and determines the signal mean. During the actual spoofing phase, the attacker combines line segments in a random fashion placing them around the learned mean value in such a way that the synthesized signal evenly fluctuates around the mean (Figure 5.8a). Sensor noise is then added on top of the line segment-approximated signal (Figure 5.8b).

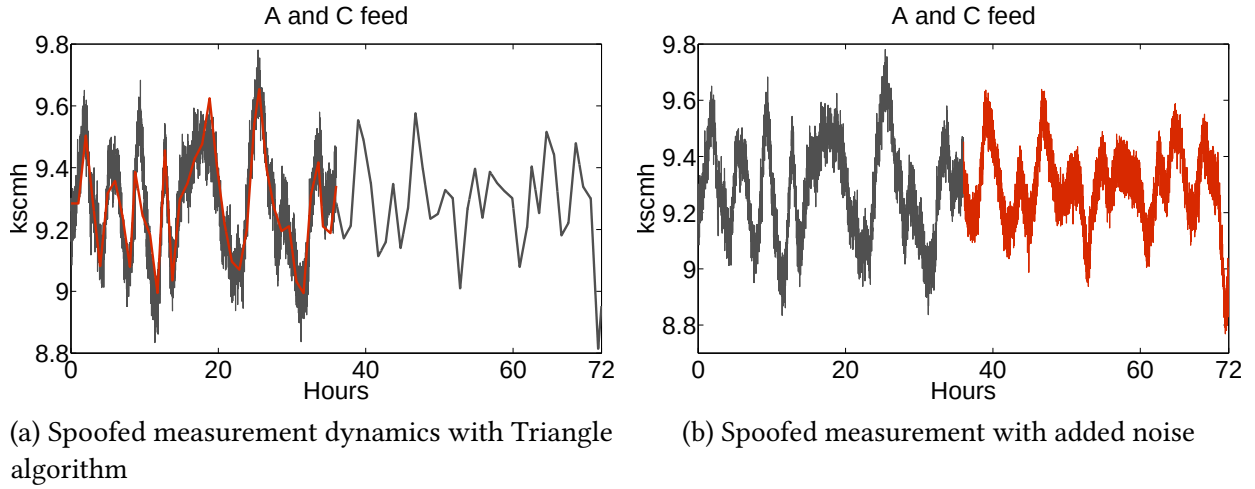


Figure 5.8: Steps of the sensor signal spoofing process

5.4 Information-Theoretic Detection Approach

Entropy is a metric from communication theory, originally intended for analyzing data signals [280]. It is an abstract measure of the randomness or information contained in a data set (continuous or discrete). The more randomness the data contains, the higher its entropy is. The essential advantage of the entropy metric is that it can be applied to distinct data sets with different semantic meanings, without requiring to have proper knowledge or understanding of these meanings. In this work we investigate the applicability of the entropy metric for detecting inconsistent and/or implausible sensor signal(s) in a cluster of correlated signals.

Definition: The entropy of a random variable X that takes values from the finite alphabet $C_X = \{x_1, x_2, \dots, x_m\}$, where each data item belongs to a character set $x \in C_X$ is defined as:

$$H(X) = \sum_{x \in C_X} P(x) \log_a \frac{1}{P(x)}, \quad (5.1)$$

where $P(x)$ is the probability of x in X .

The base a of the logarithm specifies the unit of the information measure. We set $a = 2$ to measure entropy in bits.

5.4.1 Data Preparation

Process measurements from the TE model are given in double-precision format. With this level of precision, it is unlikely that there are two or more identical occurrences of data items matching in all fifteen decimal places. Consequently, each data item of the set is unique, yielding a characteristic probability distribution: In a set N of cardinality n , each $x \in N$ has the probability of occurrence equal to $1/n$. As a result, the data sets from two different simulations without discretization will have identical entropy because of identical probability distribution functions. Therefore, in order for entropy to yield meaningful and comparative results, we first further discretize the simulated data. The data preparation process is partially based on [6] and is visualized in Figure 5.9.

In the approach from [6], the normalized data are mapped into one of 64 uniformly sized bins. This quantization method turned out to be unsuitable for the sensor data from the TE process

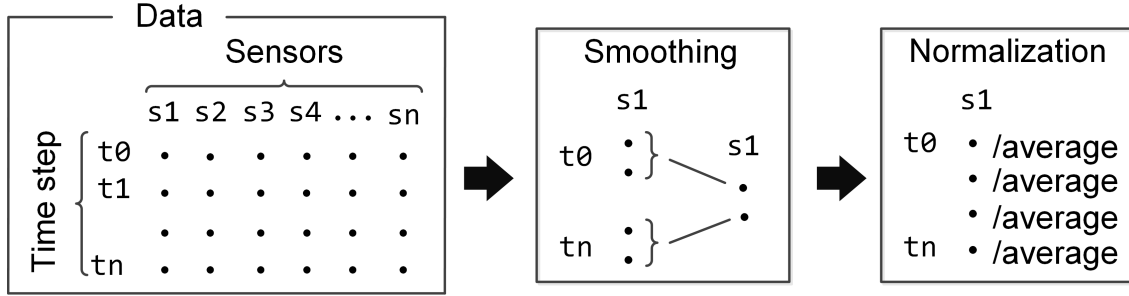


Figure 5.9: Steps of data discretization procedure

in the presence of attacks as the limited amount of bins resulted in a large number of mapping collisions. Therefore we derived our own discretization function f without limiting the number of bins. For all $z \in \mathbb{R}$ we define $\lfloor |z| \rfloor = x \in \mathbb{N}_0$, and $\lfloor 10 \cdot (|z| - x) \rfloor = y \in \mathbb{N}_0$ and derive $f(z) \in \mathbb{Z}$ as follows:

$$f(z) = \begin{cases} -(10 \cdot x + y), & \text{for } z < 0 \\ 0, & \text{for } z = 0 \\ 10 \cdot x + y, & \text{for } z > 0 \end{cases}$$

where $\lfloor a \rfloor$ is the floor function $\lfloor a \rfloor = \max\{b \in \mathbb{Z} | b \leq a\}$.

Table 5.1 provides an example of the discretization mapping. All values from a certain range are mapped into a corresponding bin. For instance, all values in the range $[0-0.1]$ are mapped into bin $f(z) = +1$.

z	x	y	$f(z)$
..1.3	1	3	+13
..1.2	1	2	+12
...			
..0.1	0	1	+1
..0	0	0	0
..-0.1	0	1	-1

Table 5.1: Illustration of the discretization process

5.4.2 Plant-Wide and Sensor Entropy

Once the simulation data are preprocessed, the entropy can be computed according to Equation 5.1 either (i) for all sensors at each time step yielding a plant-wide entropy (PWE) value or (ii) for an individual sensor over a period of time yielding a sensor entropy value (SE) as visualized in Figure 5.10.

The plant-wide entropy is particularly effective in detecting single or multiple anomalies, which have a large-scale impact on the whole system and affect multiple sensor measurements. Figure 5.11a depicts two sensor signals from the TE process, XMEAS{7;10}, in a steady state which were temporarily affected by the plant-wide disturbance IDV(8). As can be seen in Figure 5.11b, the plant-wide entropy has changed substantially during the disturbance period causing an overall rise of the mean entropy value (red line) in comparison to the steady state plant-wide entropy

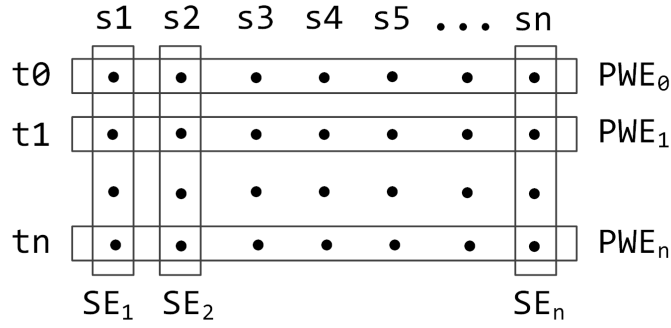


Figure 5.10: Computation of plant-wide and sensor entropy

(green line). Although the plant-wide entropy can indicate that there is a system-wide change in a measurement set, it does not provide any information on the source of the disturbance and does not reveal the affected sensors.

In contrast, sensor-specific entropy is not only useful in detecting changes in the process state but also allows the identification of affected sensors and locating the source of the issues. Figure 5.11c suggests that the most affected sensors are XMEAS{1;10} which can be attributed with certainty to the disturbance in the reactor unit. However, sensor-specific entropy is rendered ineffective if the attacker manages to spoof attack-indicating sensor signal(s) in a way that their entropy will not deviate from its normal value.

5.4.3 Cluster Entropy

Disturbances (of both natural and malicious causes) in real complex systems rarely influence the measurements of a single sensor. Instead, they have a correlated influence on multiple parts of the process (multiple sensors). Discovering these correlations allows the identification of clusters of related sensors, namely groups of measurements whose dynamic behavior is tightly correlated over time.

The hypothesis we investigate is that spoofed signals while appearing genuine at first glance, will not be correlated with the rest of the signals in the cluster. We call this type of detection *plausibility check*.

The viability of such a hypothesis is based on the limitations of the attacker. Firstly, the attacker must know which sensors form a cluster. Since this information is not known a priori, the attacker does not know which sensor signal to match to remain correlated to other signals in the cluster. Additionally, the attacker cannot observe the veracious signals and reproduce a correlated signal simultaneously in real-time because this would require her to look into the future.

We calculate *correlation* or *cluster entropy* to *detect* whether a cluster contains spoofed (uncorrelated) sensor signal(s). The approach to computing correlation/cluster entropy is based on [6] and is illustrated in Figure 5.12. First, the correlation matrix is computed between each pair of sensors. Then the sensors are clustered by partitioning the correspondent adjacency graph. Within each cluster, the correlation coefficient is computed for each sensor within a specific non-overlapping time window. The entropy of the correlation indices can be used to determine time series when one or a few sensors in the cluster become uncorrelated.

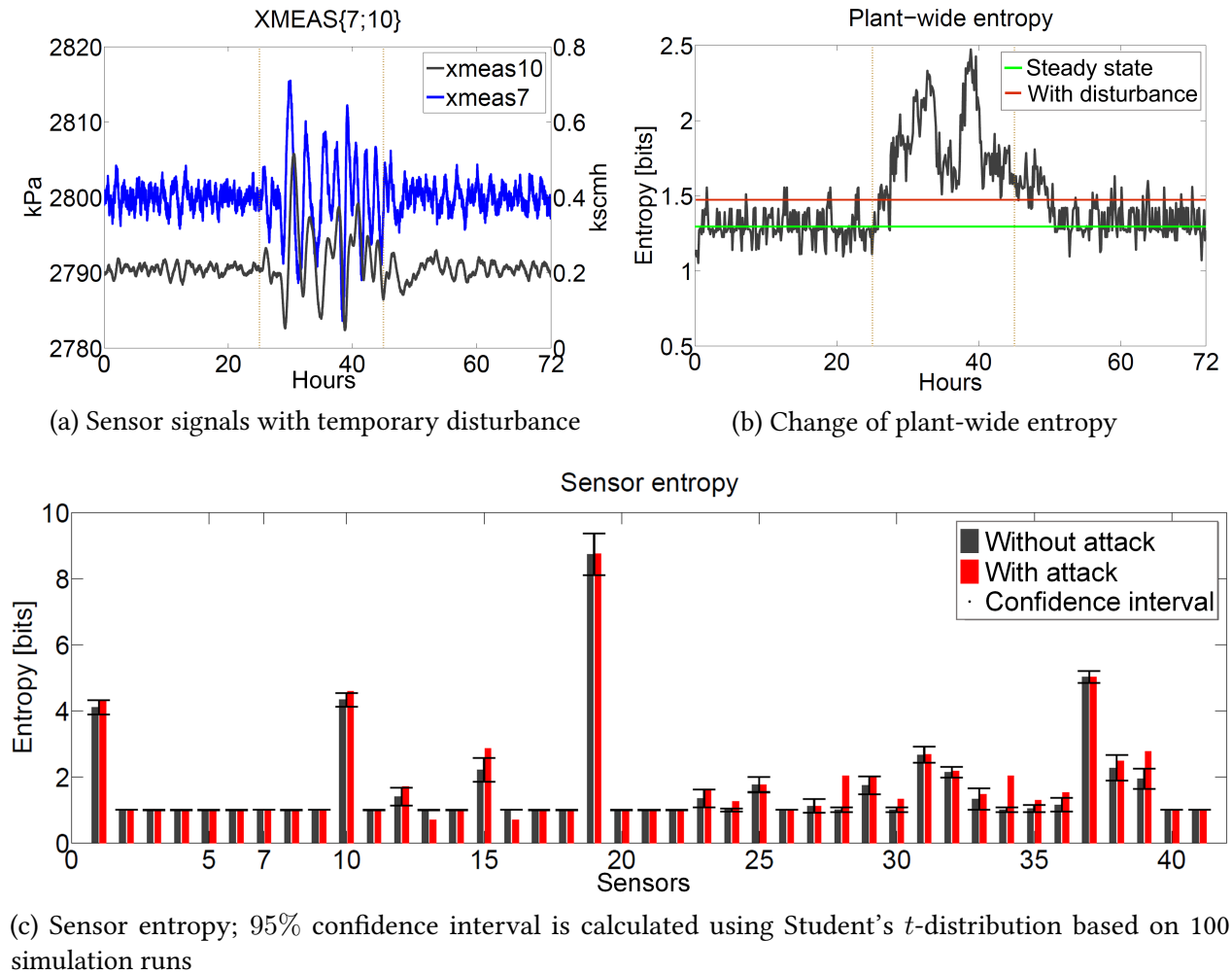
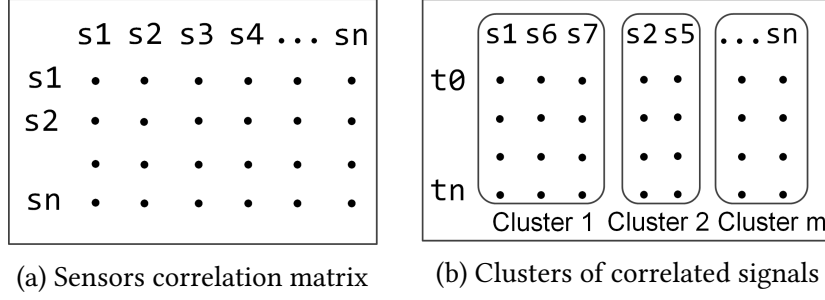


Figure 5.11: Detection of the process state change with plant-wide and sensor entropy



	s1 s6 s7	Corr1	Corr6	Corr7	Correlation entropy
t0	9 9 9 9 9 9 9 9 9	High	High	High	Low
t4	5 5 2 5 5 7 9 9 9	High	High	Low	Higher
tn	9 9 9 3 3 3 3 3 3	High	High	High	Low

(c) Change of correlation entropy in a cluster over time; t is the time stamp of individual sensor readings

Figure 5.12: Illustration of the approach to computing cluster entropy

5.5 Experimental Results

To evaluate our detection approach we conduct an experimental analysis for two operational modes: (i) steady state and (ii) with disturbances. We use similar simulation setup parameters as described in Chapter 4. Specifically, we enable disturbance IDV(8) to illustrate the ability to spoof different shapes and types of sensor signals and compare detection effectiveness for steady state and highly fluctuating process behavior.

5.5.1 Sensors Clustering

As stated in Section 5.4.3, sensors from the same cluster are more correlated with each other than with sensors from the other clusters. To identify signals that form a cluster, it is necessary to compute pairwise correlation coefficients between all the sensor signals first. We identify the distance between two sensors with the help of the Pearson correlation coefficient r [250] that takes values between $+1$ and -1 , where $+1$ is a total positive correlation, 0 is no correlation, and -1 is a total negative correlation. The result of this operation is a $n \times n$ distance matrix, where $n = 41$ is the number of process measurements in the TE process.

The pairwise linear dependencies between the sensors can be visualized with the help of a scatter plot (Figure 5.13). Correlations equal to $+1$ or -1 correspond to data points lying exactly on a line (like in the highlighted red squares). The direction of the slope (ascending or descending) determines the sign of the correlation (positive or negative). The further data points scatter from the line, the smaller the correlation coefficient is. Horizontal lines usually stand for undefined correlation. There are no clear guidelines for the interpretation of the correlation coefficients. Instead, the evaluation depends on the context and purpose of the experiment. For

instance, a correlation of 0.8 may be very low when verifying a physical law using high-quality instruments whereas it may be regarded as very high in social sciences where there might be a greater contribution from complicating factors.

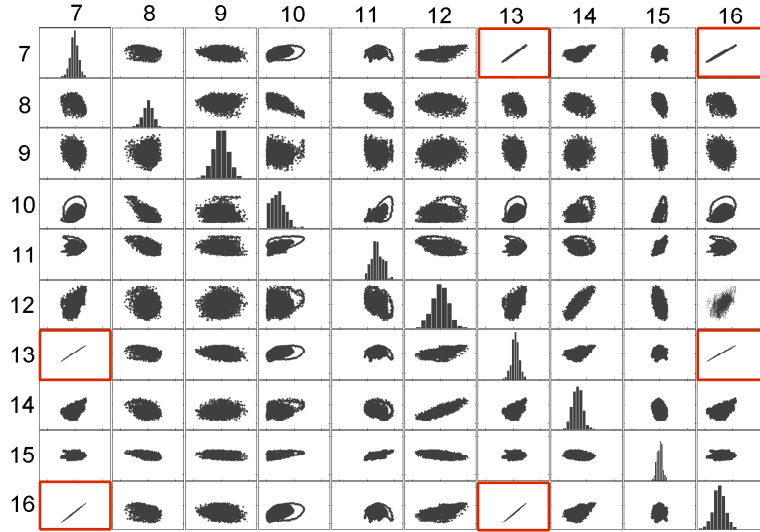


Figure 5.13: Pairwise sensor correlation coefficients

Graph partitioning

The correlation matrix represents an adjacency graph, where sensors that are closer together are linked by edges of greater weight. In this case, the identification of sensor clusters takes the form of graph partitioning. Since edge weights represent the correlation between sensors, the objective of the partitioning would be to produce a clustering that has strongly weighted edges between nodes in the same cluster and weakly weighted edges between nodes belonging to different clusters.

We have used METIS [160], an open-source toolkit for graph partitioning, which includes several multilevel graph partitioning algorithms [163]. METIS allows partitioning an unstructured graph into a user-specified number k of parts using either the multilevel recursive bisection [161] or the multilevel k -way partitioning [162] paradigms. We opted for the latter because it partitions the graph evenly and provides additional partitioning optimization capabilities. In particular, the algorithm tries to minimize the aggregate weight between the partitions. This objective is commonly referred to as the edge-cut.

One of the challenges was to identify the optimal number of partitions or clusters, k . In the context of the TE process, partitioning into smaller clusters delivered better detection results. We identified the optimal number of partitions to be $k \approx 10 - 13$. To ensure accurate clustering we verified partitioning results by checking against the correlation matrix (or scatter plot) and by inspecting sensor signals visually. Thus, METIS had identified sensors XMEAS{7;13;16} as belonging to the same cluster. The correctness of this result can be confirmed with the scatter plot on Figure 5.13 (red squares) and with the plots of the corresponding process measurements. As can be seen in Figure 5.14, the signals are indeed strongly correlated (correlation coefficient ≈ 0.95).

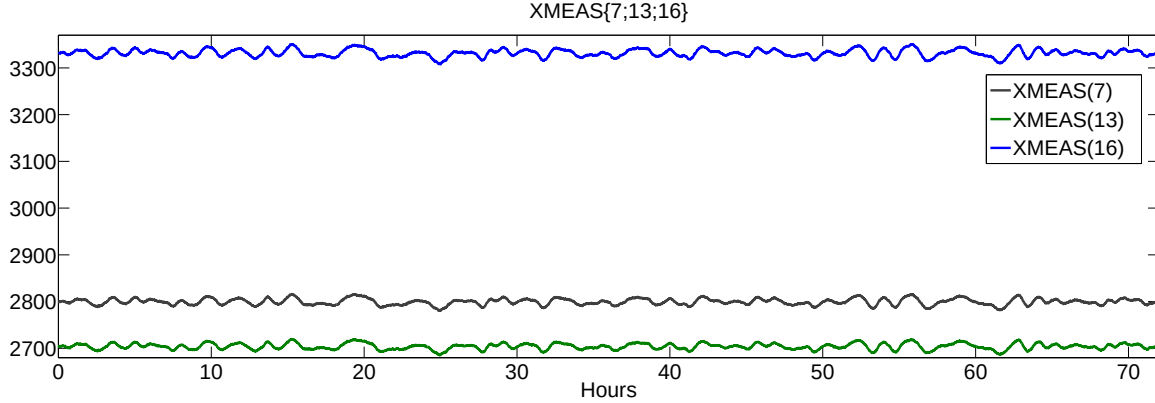


Figure 5.14: Correlated sensor signals forming a cluster

5.5.2 Detection of Spoofed Signals

To execute an attack on the reactor unit, we conduct a data integrity attack on the purge valve XMV(10) forcing it to open. Such an attack decreases reactor pressure. The attacker hides the real reactor pressure state from the operator by spoofing the correspondent process measurement, XMEAS(7), and making it look normal as depicted in Figure 5.15a. The attack window is indicated by vertical lines. Whereas the increase in the plant-wide entropy (red line in Figure 5.15b) indicates a significant change in the plant state, sensor entropy in Figure 5.15c cannot provide needed information about the source of the problem as the entropy of the spoofed sensor XMEAS(7) does not exceed the normal value.

In contrast, cluster entropy (Figure 5.16a) demonstrates an increase in the correlation entropy indicating a (more) non-correlated behavior of the signals within the cluster in the “attack”-window (indicated by vertical lines). The time window (TW) for computing entropy was set to $TW = 45$ min or 75 samples. Additionally, we had to perform a slight smoothing of sensor signals with a moving average to reduce high-frequency noise in data. This step was necessary as heavy noise conditions complicated the extraction of the correlation coefficients.

Note that even outside of the attack window the correlation entropy varies between 0 and 1. This shortcoming is the result of the entropy being computed over non-overlapping time windows. Thus, even few non-correlated signal samples cause an increase in correlation entropy in that time window. The situation is even worse in case of weakly correlated sensors. Figure 5.16b demonstrates the detection of a spoofed signal in a cluster formed by XMEAS{5;6;23} with $r \approx 0.85$.

As can be seen, not only non-correlated samples of the more weakly correlated veracious signals cause higher entropy level outside of the attack window, but we also did not clearly detect the attack. Non-correlated samples of the spoofed signals became distributed over several time windows without causing clear increases of the entropy values in the attack window. To mitigate the weakness of detection with non-overlapping time-window cluster entropy, we apply a sliding window-entropy detection.

Detection with Sliding Time-Window Entropy

The principle of computing sliding time-window entropy (TWE) is illustrated in Figure 5.17, where t as before stands for the time stamps of the sensor readings. The entropy is computed for a predefined time window. However, at each step the time window slides only one sample further, i.e., one sample is discarded and one sample is added to the window. The advantage of this approach is that it allows computing entropy over larger time windows. In this case even if a single sample

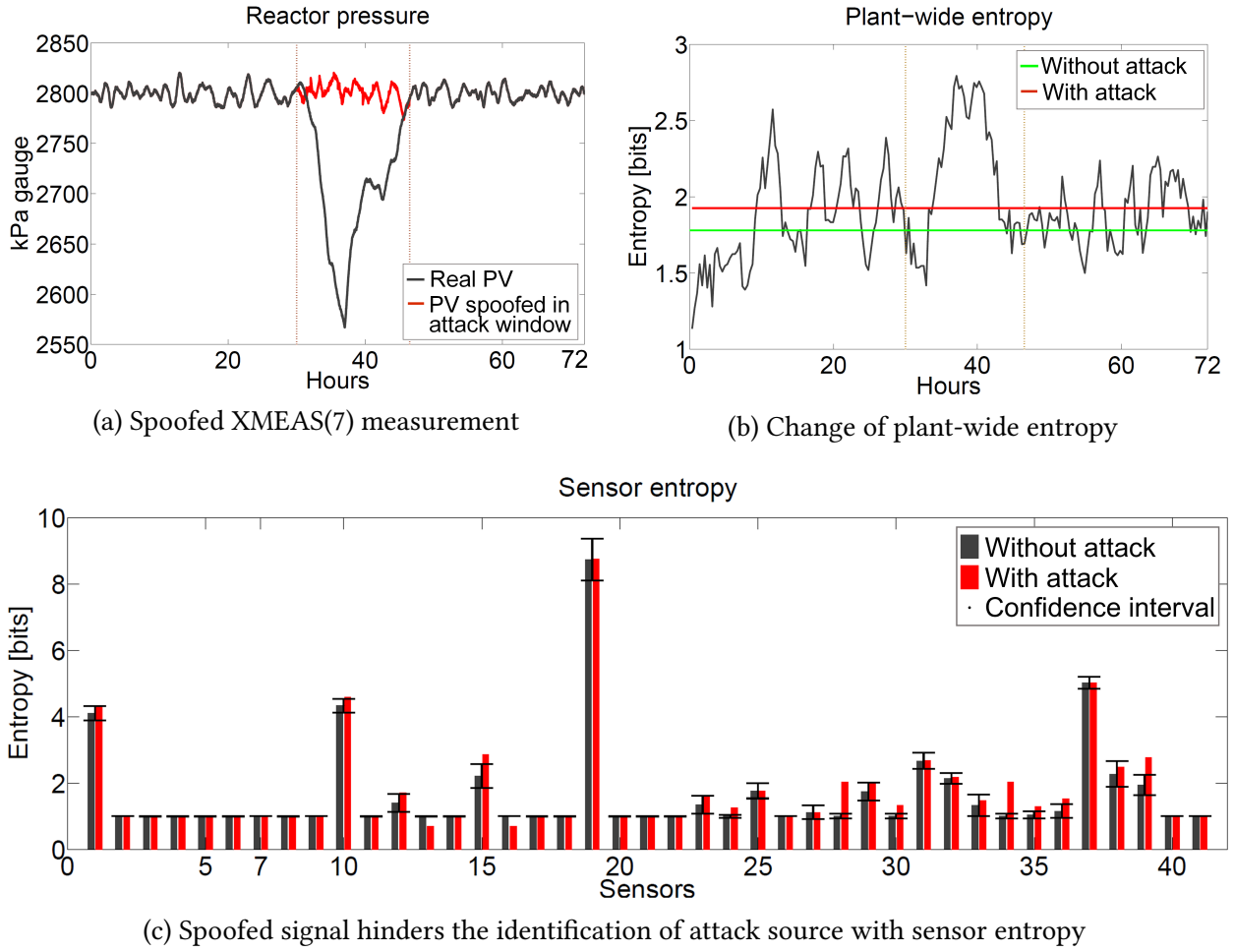


Figure 5.15: Detection of the data integrity attack on XMV(10) with PWE and SE

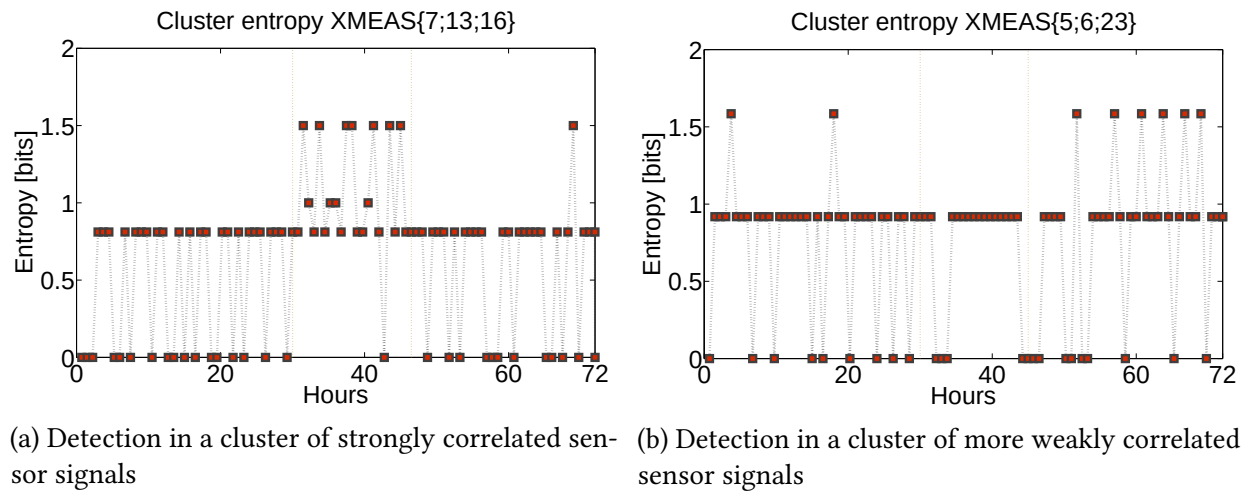


Figure 5.16: Correlation entropy calculated over non-overlapping time windows

is uncorrelated due to inherent random noise in the sensor readings it has little impact on the overall entropy in the entire window. What is more important, at some point the time window for computing entropies will largely overlap with the attack window (in red) enabling unambiguous detection of the presence of an uncorrelated signal in a cluster.

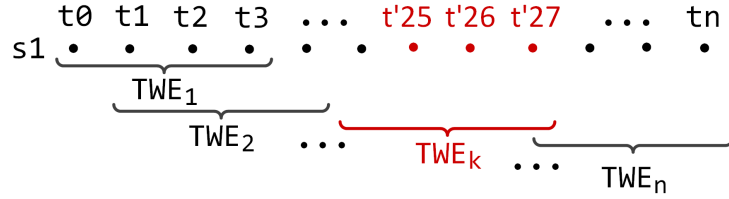


Figure 5.17: Computation of correlation entropy with sliding time window

Figure 5.18 demonstrates detection of spoofed signals with sliding time-window entropy in cluster XMEAS{5;6;23}, with disturbance IDV(8) active (Figure 5.18a, 5.18c) and in a steady state (Figure 5.18b, 5.18d). The time window is set to $TW = 12$ hours. Spoofed signals are shown in red. It should be noted, that this method is characterized by a slight detection delay. This is because for the correlation entropy to increase it is required to have a certain dominant number of uncorrelated samples in a cluster. For the same reason cluster entropy remains higher for some time after the spoofing attack ends.

5.5.3 Use Case: Powerful Attacker

We now assume a worst case scenario where the attacker (i) knows which sensor signals form a cluster, (ii) spoofs all the signals making them looking authentic as shown in Figure 5.19a. In this scenario, while all signals in a cluster are *plausible*, they will not be *consistent* because they lack correlation. This assertion is based on the assumption of the attacker's inability to reconcile signal spoofing processes in real time. Thus, the attacker synthesizes spoofed signals by assembling together line segments "learned" for these particular type and scale of signals. To synthesize correlated signals, the attacker must be able to pick "correlated" line segments out of a bin of segments for each signal. This turns the attacker's task into a sequential object inspection problem, which is not solvable in real time. Lack of correlation between the spoofed signals can be detected by computing the time-window entropy of the corresponding cluster (Figure 5.19b). We call this type of spoofed signals detection *consistency check*.

Further, we investigate the detection of (all) spoofed signals in a cluster when the process is in a steady state. As mentioned earlier, most sensor signals in the TE process in a steady state do not deviate far from their set points and look more like flat lines with added noise. Computing the correlation entropy of such signals requires significant signal smoothing to uncover underlying (long-term) signal dynamics.

Figure 5.20a depicts real and spoofed signals of XMEAS(20) from the cluster XMEAS{11;18;20;22}. The rest of the signals are also accurately spoofed but are not shown in the figure to preserve visual discernibility. As can be seen in Figure 5.20b, without smoothing it is not possible to determine correlation coefficients of the steady noisy signals. In contrast, prior signal smoothing allows detection of inconsistency even in seemingly flat signals (Figure 5.20c). The selection of the appropriate smoothing parameter m depends on the noise profile and sampling frequency of the signal. In our experiment we set $m = 850$.

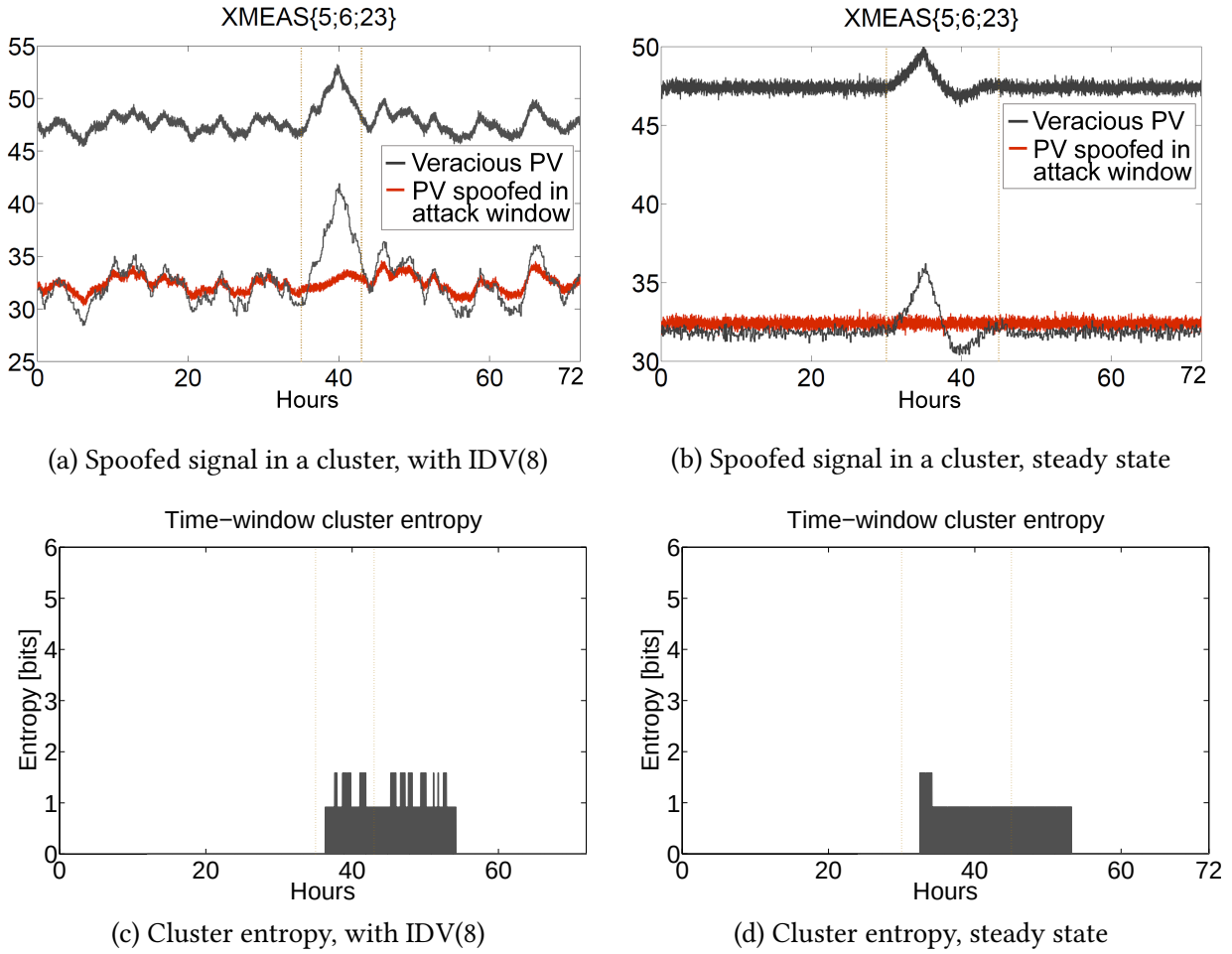


Figure 5.18: Detection of spoofed signal presence in a cluster, with a plant disturbance and in a steady state

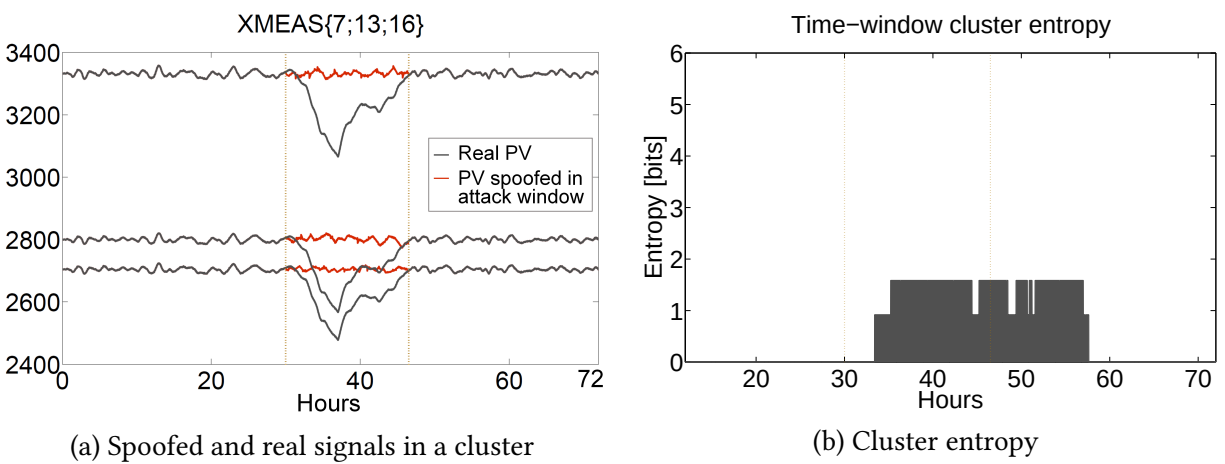
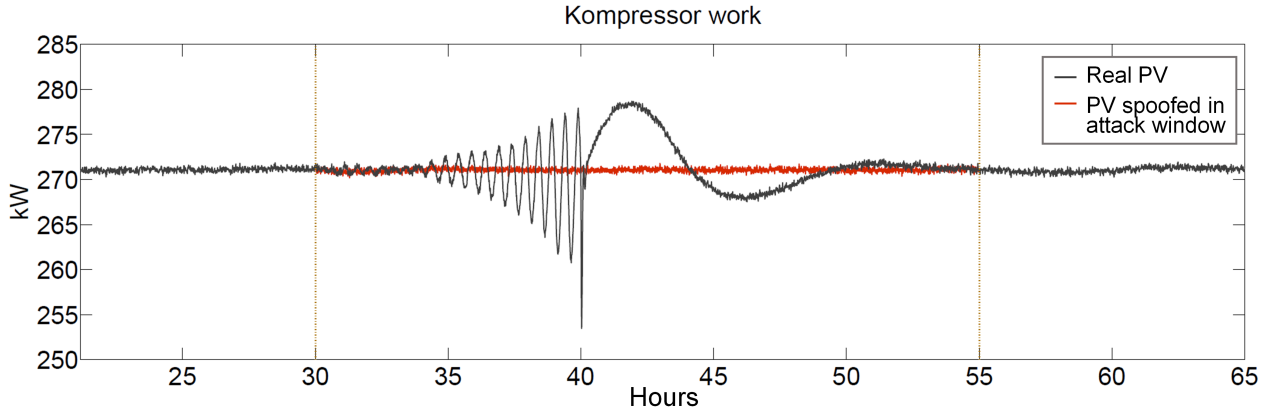
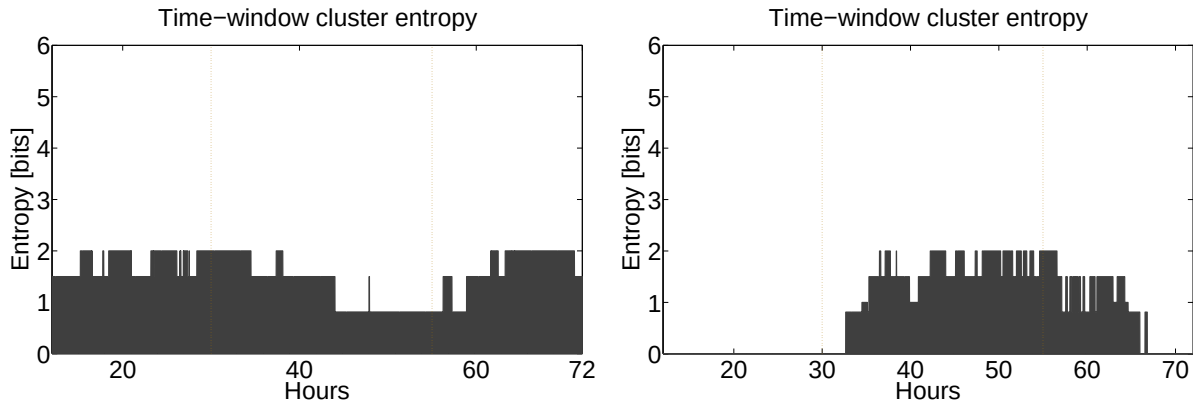


Figure 5.19: Detection of spoofed signals presence in a cluster when all signals are spoofed



(a) Spoofed and real signal XMEAS(20), steady state



(b) Cluster entropy without smoothing

(c) Cluster entropy with smoothing

Figure 5.20: Detection of spoofed sensor signals in a steady state

5.5.4 Detection Complexities

We verified the effectiveness of our approach by testing the detection of spoofed signals (i) during different operational conditions and (ii) under several attack scenarios. Although the approach delivers robust performance, inaccurately identified clusters may impede its effectiveness.

During our analysis we discovered that pairwise correlation coefficients within a cluster may change depending upon the state of the process. As can be seen in Figure 5.21a all sensors in the cluster formed by XMEAS{4;5;6;23} demonstrate homogeneous correlational relationships under disturbance IDV(8). However, signal XMEAS(4) becomes uncorrelated to the rest of the signals under disturbance IDV(1) which stands for a change in reactants A/C feed ratio (Figure 5.21b, in red squares). Exhaustive analysis of this cluster showed that XMEAS(4) did not belong to the cluster in the majority of scenarios. Therefore we excluded it from the cluster.

In contrast, the cluster formed by XMEAS{7;13;16} has consistently demonstrated strong dependencies between the sensors (Figure 5.21c, 5.21d). Finding such clusters of sensors with stable correlations is essential for avoiding/reducing false positive alarms.

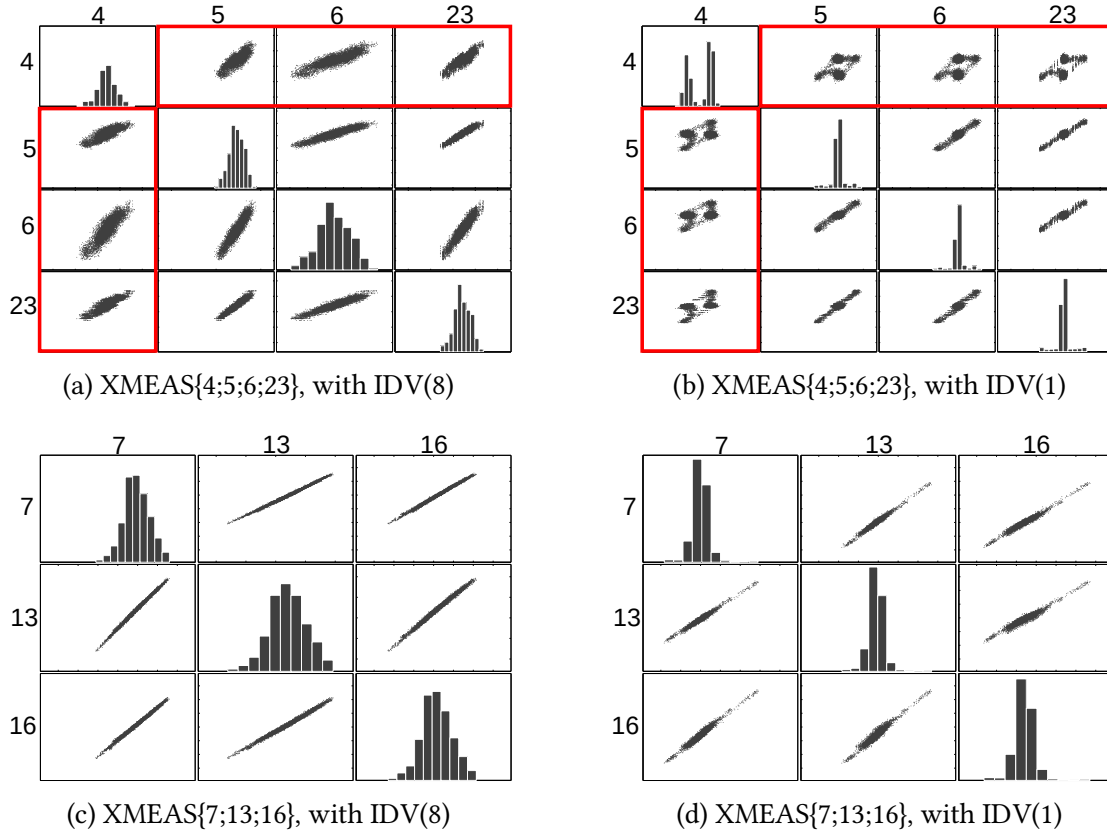


Figure 5.21: Correlation inconsistency

5.5.5 Increasing Detection Robustness

Although signals in cluster XMEAS{7;13;16} always remain correlated, a powerful attacker can undermine the detection capability in this cluster. This cluster is formed by three signals identical in terms of signal type (Type 3) and signal scale. As a result, the attacker only needs to spoof one signal which she can re-use (re-play) to spoof other signals as well. In this case, all three signals in the cluster will be both *plausible* and *correlated*. Logically, our approach cannot detect such an attack.

To increase detection robustness it is, therefore, advisable to form clusters consisting of different types of signals and/or of different scales. Figure 5.22a depicts two signals XMEAS{20;22} of the same type (Type 1) but having different scales. These signals belong to the same cluster and we assume an attacker using XMEAS(22) to extract signal characteristics for a later simultaneous spoofing attack. As can be seen in Figure 5.22b, the attacker is not successful due to the differences in signal scaling. The line segments extracted from signal XMEAS(22) are too short to believably reproduce XMEAS(20).

Although accounting for scaling and adding additional algorithms for spoofing different types of signals are solvable problems, they force the attacker to account for more edge cases and write larger code (which may not eventually fit into a small microcontroller), and in general exert more effort to perform an attack.

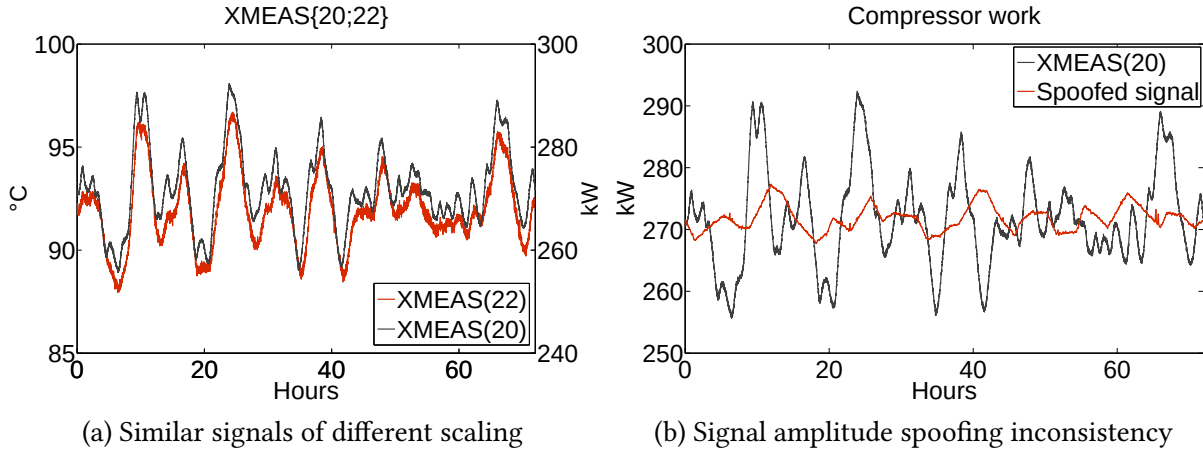


Figure 5.22: Inaccurate spoofing of a signal by learning from a signal of a different scaling

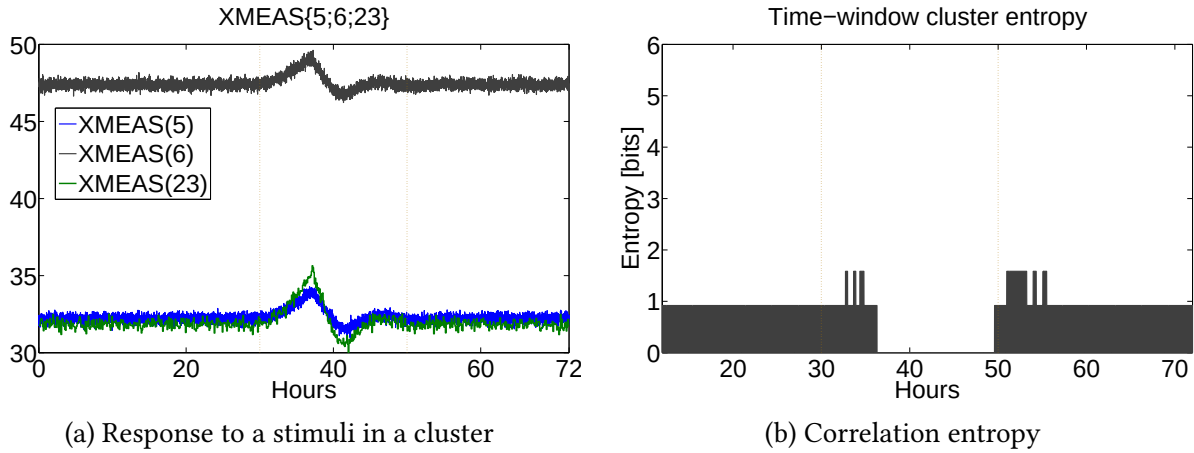


Figure 5.23: Identification of related/correlated signals

5.6 Discussion

In the course of our experiments, selection of the right approach to clustering sensor signals turned to be the most crucial and time-consuming process. We found that this process delivers most accurate results with human assistance and thorough testing of correlations under different operational conditions. With hundreds or thousands of sensors in a facility it could be an onerous task which would benefit from a reliable automated solution. However, the clustering process is sustainable in that once it is done, it need not be revisited or changed often.

In the context of the TE process, most signals do not show clear dependencies in the steady state due to flat dynamics. In order to identify the distances between the signals the system needs to be given a stimulus to observe which signals respond to the input in a correlated manner (Figure 5.23a). Verification of the cluster can be automated with “negative” detection. Thus, we verified clusters by applying temporary disturbances (both natural and intentional) and measured correlation entropy. If signals react to the stimuli identically, the correlation entropy will be zero (Figure 5.23b). Note that we do not smooth signals for this type of detection.

To increase detection robustness it is not only advisable to include signals of different types and scales but also signals from distant locations. Attackers generally use malicious code in places where there is a single point of data manipulation. A group of sensors located on a single bus can be manipulated at their common interface. Therefore forming clusters of sensors on different buses provides more protection by reducing the chance of having all sensors in the cluster spoofed.

In (petro)chemical facilities detection robustness can be further improved by including measurements of product variables or quality indices. These are offline data obtained through the analyzer equipment by performing a chemical analysis of the liquid or gaseous components. Typically the analyzers are placed at the critical process locations to control the chemical composition of the process flows. Each analyzer is characterized by its sampling frequency (frequency of taking a probe from the stream) and by the dead time (time between taking a sample and when the analysis is complete). Because of the dead time, such signals have a step-like form (Figure 5.24). In case it is possible to ensure the security of the analyzing process, signals of quality indices can be used as “known good” signals in clusters.

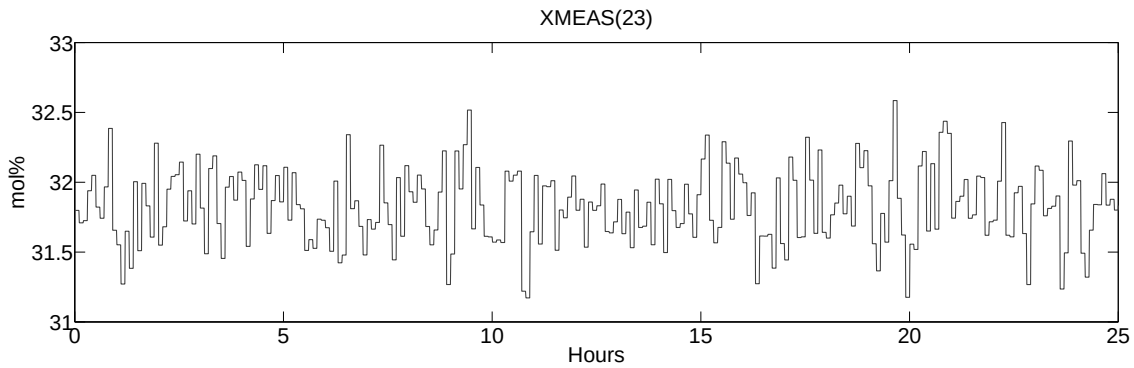


Figure 5.24: Chemical analyzer signal

There are three analyzers in the TE process generating 19 signals $XMEAS\{23-41\}$ with lag times from 10 to 25 minutes. The majority of them are strongly correlated with the other real-time process measurements. Particularly, we tested our approach on the cluster $XMEAS\{5;6;23\}$ which includes signal $XMEAS(23)$ from the analyzer that measures $mol\%$ of component A in the reactor feed (Figure 5.24).

5.7 Related work

With the introduction of new types of sensors and smart instrumentation, the volume of process data has been steadily increasing. Predictably, plant data began being treated by methods from big data science. A comprehensive study of big data veracity and how it could be evaluated is discussed by Rubin and Lukoianova [266]. With the growing awareness of data trustworthiness criticality, veracity began to be considered across various disciplines. The challenges of storing, managing, processing, analyzing, visualizing and verifying the quality of geospatial data are examined by Li et al. [202]. Data handling and data veracity issues in the healthcare sector are highlighted by Raghupathi and Raghupathi [252]. Data veracity in intelligent transportation systems and a comprehensive taxonomy of threats to data veracity are covered by Staron and Scandariato [290]. We discuss veracity in the context of industrial cyber-physical systems such as continuous production processes.

The effectiveness of the entropy-based attack detection in in-vehicle networks is explored by Mueter and Asaj [234], a work which inspired our research. In the work closest to ours, Agogino

and Tumer [6] investigate the applicability of entropy-based anomaly detection techniques to detecting and locating anomalies in the space shuttle main engines. We have extended the proposed approach by introducing sliding time-window anomaly detection which allowed us to significantly increase the accuracy of detection. Methods of clustering industrial process data and comparison of their effectiveness can be found in works by Kiss et al [167, 168]. Sensors signals are grouped based on their association with specific operating states. In contrast, in our approach, we identify clusters of correlated sensor signals. Similar to our observations, the researchers highlighted the importance of accurately tuning the detection parameters when using clustering algorithms. An approach to plausibility checks in automotive systems using immanent signals, derived from hard-wired sensors is discussed by Ring et al. [263]. Similar to our conclusion, the authors recommend the usage of hard-to-spoof signals (e.g., hard-wired signals) to increase the reliability of plausibility checks.

A variety of methods to detect the presence of malicious signals is discussed in the literature. Wang et al. [317] developed an intrusion detection approach for false data injection attacks which relies on the correlation graphs (for system variables) and state relation graphs (for system states). The proposed approach is heavily dependent on knowledge of the possible and invalid system states. McAvoy and Wolthusen [224] describe an anomaly detection approach based on proxy measurements to detect attacks disguised in the sensor signals noise. They compare the predicted system output using real-time numerical simulation along with the actual output to detect attacker manipulations. Both works rely on having a plant model for detection. The complexity of modeling continuous processes and in particular of a plant-wide process dynamic [22] limits the application of such approaches. In contrast, our method does not require knowledge of the plant states. Linda et al. [204] propose an approach for detecting falsified control systems states with the help of encrypted and thus secure (“known”) sensor measurements. By relying on a small set of randomly selected sensors, the algorithm imposes little additional computational and bandwidth costs. Whilst this approach is feasible, the advantage of the entropy-based approach is that it does not require changes to existing IT operational practices like introducing encryption. Shoukry et al. [282] propose an approach to reconstructing the state of a differentially flat system when a subset of sensors is corrupted by the attacker. A process-aware approach with the use of invariant equations based on the physical and chemical properties of the process and a Multiple Security Domain Nondeducibility (MSDND) framework to detect when a sensor signal is being maliciously manipulated is presented by Dunaka and McMillin [89]. Although the usage of invariants offers accurate detection of anomalies, it relies on specific process information of individual plants. Our approach is plant knowledge agnostic.

5.8 Conclusions

Cyber-physical systems security is looking at applications with observations and effects in the physical world that rely on services provided by an IT infrastructure. Securing the IT infrastructure protects against an adversary who can manipulate data as it is processed in the infrastructure, it does not protect against an adversary feeding misleading data to the infrastructure, possibly by manipulating sensors. In this chapter we investigated the issue of data veracity in the context of process control systems.

No process is fully instrumented and at best sensors can provide only an approximation of the process. Therefore the accuracy and reliability of sensor readings are paramount. In process control applications sensor data generally are considered trustworthy. We have implemented a set of techniques to synthesize believable but fake sensor signal which can fit into a microcontroller embedded in a measuring instrument. No canonical IT security solution can guarantee that the

inputs from a sensor faithfully represent the reality. To overcome the limitations of securing process data with information security approaches we propose a process-aware detection technique based on the correlation entropy in a cluster of the related sensors. The detection technique takes the form of plausibility and consistency checks. We have empirically verified the effectiveness of our approach during different operational conditions of the plant. The cluster entropy method has demonstrated robust performance for different types and the number of spoofed signals. However, care must be taken when forming the clusters as improperly formed clusters may result in false positives.

Note, that we did not investigate the impact of timeliness on data veracity. Our approach detects veracity violation in near real-time, as soon as data become available in the infrastructure. We assume no malicious threat actor is delaying the data with, e.g., DoS attacks as discussed in the previous chapter. In fast processes with short time constants, acquired process values may only be valid for a short time and delayed data should be used with care in plausibility checks.

Chapter 6

Cyber-Physical Attack Lifecycle

There is no security on this earth; there is only opportunity.

Douglas MacArthur

6.1 Introduction

Threat intelligence and adversarial studies constitute an important part of cyber security defense strategies. Understanding attack strategies and tools allows defenders to proactively implement effective security controls and monitoring tactics. The attacks on selected organizations or systems are not executed at random. An attacker goes through a sequence of steps to successfully infiltrate a network and, e.g., exfiltrate data from it. An “attack lifecycle” or “kill chain” [147] is a common method to describe the process of conducting cyber attacks. One of the popular attack models is the attack lifecycle proposed by Mandiant [213]. Figure 6.1 shows the major attack stages, from initial reconnaissance where the attacker conducts research on a target to the final attack mission. Note that at each stage the adversaries have specific goals and will use appropriate methods to accomplish their goals. If one of the attack vectors and/or methods fails and the attack progress is impeded, the adversary would iterate the stage until success or circle back to a previous stage.

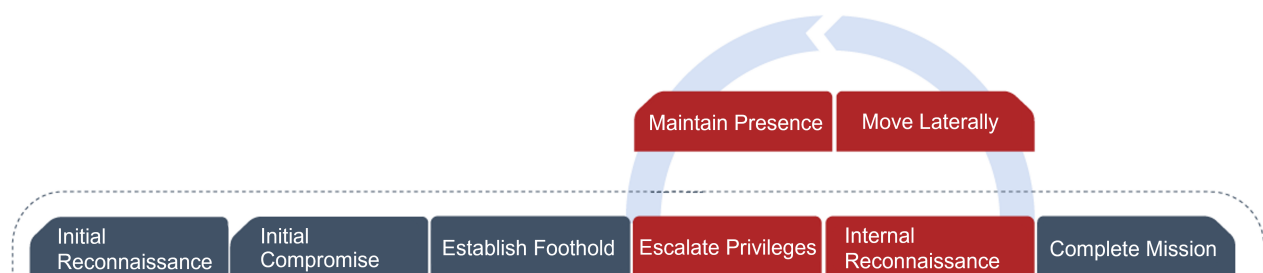


Figure 6.1: Mandiant Attack Lifecycle [213]

The attack steps leading to the final stage of the attack lifecycle “Complete Mission” are required to obtain persistent control over the victim’s infrastructure. This sequence of steps is generic and applicable to most organizations and targets. In the final stage, the attackers act upon their motivation and take the required action to achieve their planned mission. The attack mission can be simple or complex and may consist of additional steps. In information security, the attacker’s

goal may vary from stealing financial assets or exfiltrating sensitive information like intellectual property to extorting the victim organization or destroying networking equipment.

In the context of cyber-physical systems, the attacker aims at the interaction with the physical process to achieve a desired outcome in the physical world. This may include denial of control loop communication, degradation of equipment performance, or causing process upsets. When weaponizing a buffer overflow in a software application, the shellcode is constructed to make the target system perform actions desired by the attacker. Similarly, cyber-physical payloads must contain a set of instructions that manipulate the process, and the choice of instructions depends on the specific impact the attacker wants to have on the process.

Industrial environments are complex and their comprehension requires multidisciplinary knowledge. In this chapter we consider the physical part of the attack mission and propose a cyber-physical attack lifecycle model. Similar to the “traditional” IT attack lifecycle, our framework describes the attack steps necessary to design and execute a cyber-physical attack. We examine the hurdles an attacker might face when trying to manipulate physical processes, using a realistic simulation model of a Vinyl Acetate plant as a case study. We demonstrate a complete attack, from start to end, directed at prolonged/persistent economic damage to a production site while avoiding the attribution of production loss to a cyber event. Such an attack scenario could be used, e.g., by a manufacturer aiming at putting competitors out of business or as a pressing argument in an extortion attack.

6.2 Classes of Cyber-Physical Attacks

Modern industrial plants face multiple challenges such as the delivery of products at consistent quality and low cost, management of plant dynamics altered by material recycling and energy integration, strict adherence to environmental and safety regulations, and having a sufficient degree of flexibility to handle fluctuations such as production rate changes in response to changing market demand and feed quality. Maintaining these complex requirements is the responsibility of an efficient and reliable process control infrastructure. Due to a high level of interdependencies and efficiency requirements, any malicious intervention may result in significant consequences to plant performance and compliance. This knowledge can be exploited by the attacker.

The process of designing a cyber-physical attack scenario starts with an attack objective or a precisely desired impact. In general, the effects of cyber-physical attacks can be classified into three groups. Admittedly, the classes outlined are interrelated as damage of one kind may lead to another kind of damage. For example, production can be disrupted through the breakage of equipment. Runaway reactions can cause serious safety accidents and equipment breakage. However, the distinction between attack objectives allows for a better judgment of how different damage types can be maximized and provides guidance for more accurate risk assessment.

6.2.1 Equipment damage

This class of attacks aims at physical damage of equipment or infrastructure (e.g., pipes or valves). Classes of physical damage can be found in [188]. In general, equipment damage can be achieved under two conditions.

Overstress of equipment. Every piece of equipment wears out or breaks at the end of its expected lifecycle. Prolonged overstress of equipment can accelerate this process. An example is wear-off attacks on valves due to unstable process control. This type of attack was implemented in the second version of the Stuxnet worm [187].

Violation of safety limits. Violation of the equipment's safe operating/design typically results in rapid deterioration of the equipment state and its subsequent damage. This scenario was implemented by engineers at Idaho National Labs to remotely destroy a power generator [327]. This attack idea was also realized in the first version of Stuxnet [187]. Equipment safety limits are public information and are readily available in books, professional articles and equipment manuals. For instance, safety limits of piping infrastructures and related equipment can be found in [197].

6.2.2 Production damage

Instead of breaking equipment, an attacker can target the production process to degrade the product or make production more expensive. Attacks on production can be divided into three groups.

Product quality and production rate. The attacker may reduce the value of the product or make it unusable. Every product has its specification and its market price varies with the quality grade, car fuel being a familiar example. In certain instances, the price of a product may rise exponentially with product purity. Table 6.1 presents relative prices for paracetamol. As can be seen, not achieving the desired product quality can result in significant revenue losses. The similarly detrimental economic impact can be caused by reducing the production rate.

Purity	Price, Euro/kg
98%	1.0
99%	5.0
100%	8205.0

Table 6.1: Relative paracetamol prices [286]

Operating costs. After the process is configured and tuned, the operator's primary task is to keep the process as close as possible to the economically optimal operating conditions. Every plant has an objective cost function consisting of several components which impact the operating costs, e.g. as described in Equation 3.1. It may be loss of raw materials in the purge, premature deactivation of the catalyst, or increased energy usage. Increased operating costs reduce revenue and plant profitability.

Maintenance efforts. The attacker can impact a production process by increasing the maintenance workload. Maintenance includes troubleshooting process disturbances and equipment malfunctions. For example, the rapid operation of a flow valve causes a damaging cavitation process. Cavitation eventually wears off the valve and leads to leaks, requiring valve replacement. Additionally, the bubbling of liquid results in turbulent liquid flows and substantially complicates process control, requiring the involvement of external specialists.

6.2.3 Compliance violation

Industrial sectors tend to be strongly regulated to ensure employee safety and protect the environment. Non-compliance can result in fines and bad publicity (unlike economic attacks whose effect can be kept internal to a company).

Safety. Most damaging would be attacks on occupational and environmental safety as they may result in lethal accidents and serious environmental damage. This type of attack may also yield unpredictable collateral damage, resulting in an extended damage scope.

Environmental pollution. A less serious environmental impact is described as pollution. An attack targeted at exceeding regulatory pollution limits such as concentration and/or volume of

gaseous emissions or insufficiently treated wastewater discharge will induce fines, and recurrent offenses can even lead to plant shutdown. A negative impact on reputation may be a further consequence.

Contractual agreements. Some industries may be required to adhere to legally binding production schedules, vaccine production being one example. Reactions to outbreaks of a disease often lead to political and public pressure to supply needed pharmaceuticals in a timely fashion. Missing delivery schedules may cause contractual sanctions and negative publicity.

Classes of cyber-physical attacks, their categories and predominant impact are summarized in Table 6.2.

Equipment damage	Production Damage	Compliance violation
Equipment overstress Safety limits violation	Product quality Production rate Operating costs Maintenance efforts	Safety Pollution Contractual agreements
Downtime	Financial	Publicity

Table 6.2: Classes of cyber-physical attacks and their major impacts

6.3 Cyber-Physical Attack Lifecycle

The stages of a SCADA attack with a goal to achieve physical impact were first presented in 2006 by Larsen as shown in Figure 6.2. However, Larsen did not provide descriptions of attacker activities related to each stage or supplied any other additional information.

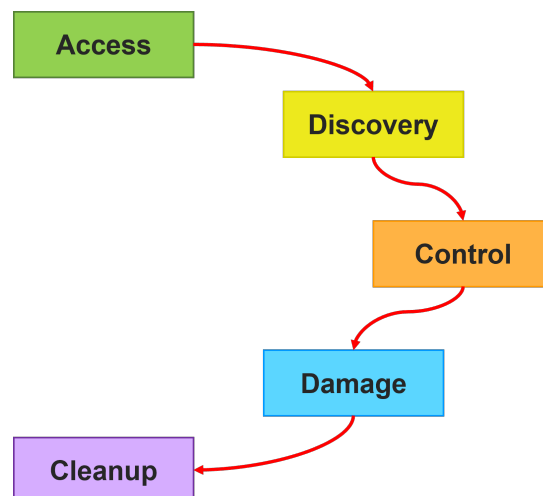


Figure 6.2: Stages of SCADA attacks [188]

In the absence of supporting information, we decided to follow the proposed sequence of attack stages and used our judgment to identify necessary attacker activities at each of the attack stages. Through experimental work and empirical observations, we were able to arrive at a description of the attack stages and enhanced the initial model to better describe essential attack steps. Specifically, we added *Prevent Response* and *Obtain Feedback* steps as independent attack sub-stages due to their

significance in planning and executing cyber-physical attacks. Figure 6.3 illustrates the proposed cyber-physical attack lifecycle.

In this chapter we describe the stages of the cyber-physical attack lifecycle and illustrate the applicability of the proposed framework in the example of a targeted attack on a chemical plant with the objective of causing a prolonged negative economic impact. A developed set of attack payloads can be delivered as a series of remotely executed commands/scripts or programmed into an autonomous attack payload.

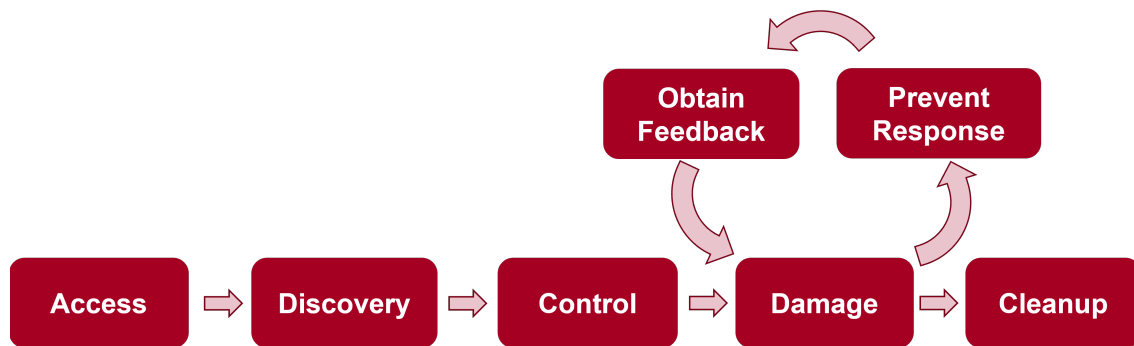


Figure 6.3: Proposed cyber-physical attack lifecycle

6.3.1 Access

Access is the stage closest resembling traditional intrusion attacks in the IT domain. In general, the attacker requires code executing capability in the victim's industrial control network to manipulate the process and thus has to find a way in. Even though this stage is largely the same as intruding into any other network, the attacker may utilize knowledge about the ICS ecosystem to find alternative ways of obtaining access.

Industrial control systems are increasingly getting connected to corporate networks and to external organizations like regulatory bodies, vendors and third-party providers, making them reachable from the Internet. Additionally, in some cases, ICS equipment is directly connected to the Internet for ease of remote access. Any of these data flows can be a potential way into a control network or provide direct access to control equipment. Among the latest known examples of successful execution of the Access stage is the Triton attack where a suspected state-sponsored threat actor obtained remote access to a Safety Instrumented System in a petrochemical facility [159]. Figure 6.4 illustrates the lateral movement of the attacker in the victim environment. The attackers obtained remote access credentials of an internal employee and took advantage of the Remote Desktop Protocol (RDP) access to the control system network from the Demilitarized Zone. Further, they leveraged architectural weaknesses to gain access to the SIS engineering station and attempted to implant a Remote Access Trojan (RAT) into the memory of an SIS controller.

Although the threat actor behind the Triton attack has shown that compromising individual industrial facilities is possible given sufficient time and effort, the execution of such operations does not scale easily. It took the attack team 12 months to obtain access to the assets needed in the target facility. In March 2018, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) published an advisory about threat actors conducting multi-stage intrusion campaigns which used so-called "staging targets" to obtain access to the intended targets [149]. The staging targets are typically trusted third-party suppliers and partners with relatively insecure networks that are selected because of their pre-existing relationships with the intended targets. The threat actors have been using these third-party networks to host attack tools and eventually pivot into the infrastructures

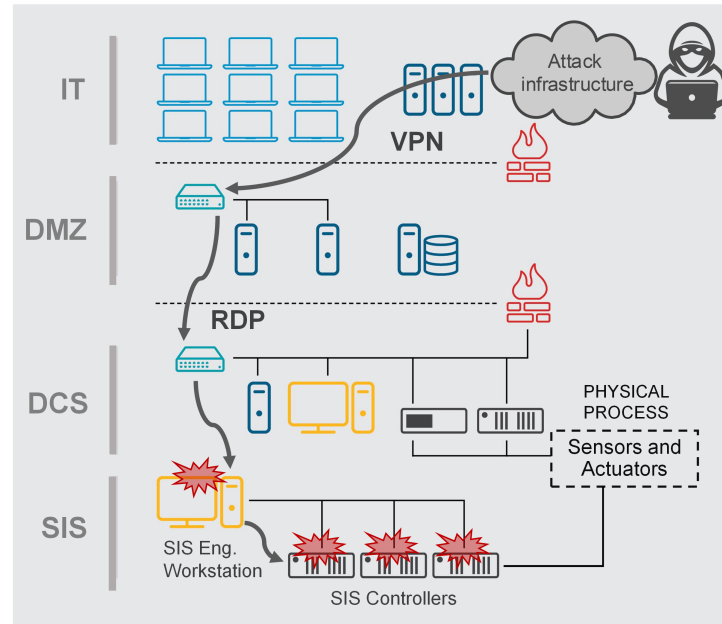


Figure 6.4: Attack path to safety systems in Triton attack [159]

of their intended critical infrastructure targets. The advisory does not elaborate on whether the attackers were interested in specific industrial organizations or were acting opportunistically with the objective of obtaining access to as many industrial organizations as possible. One downside of such intrusion campaigns is that initially unintended parties can become opportunistic targets and collateral victims simply because of being customers or partners of “staging targets”.

Supply chain attacks in the industrial domain can be traced to at least 2012 with examples being the complete compromise of SCADA software vendor Telvent [173] and the distribution of malware that was digitally signed by the whitelisting software vendor Bit9 [174]. In one of the most successful supply chain campaigns, Havex, the attackers trojanized legitimate software update installers on at least three ICS vendor web sites [97, 148]. Once installed, Havex attempted to call back to a Command and Control server (C2). When successful, malware would send back information gathered about the victim environment and download further attack instruments (see Section 2.4). While Havex appeared to only conduct reconnaissance activities, its level of access to the control networks would have allowed an attacker to interact with the physical process and execute cyber-physical attacks of high precision.

Due to their size, complexity and interdisciplinary nature, many industrial environments rely on a large number of integrators, engineering contractors and subcontractors throughout the entire lifecycle of their operational sites. These organizations hold a vast amount of engineering documentation such as plant designs and network diagrams, calculations and project files which are indispensable when designing a targeted cyber-physical attack making these organizations valuable attack targets. In the past threat actors have already conducted large-scale operations to obtain access to such organizations [49].

Despite growing awareness of cyber security threats and risks to the process control infrastructures, thousands of ICS assets are being exposed to the Internet without proper security measures [17, 231] helping attackers to gain access to the control devices and applications. Researchers routinely conduct industry- and country-focused as well as worldwide scans to quantify the online exposure of ICS devices and identify exploitable and weakly protected systems [47, 236, 258, 323]. Among frequently identified issues are poor configurations, usage of insecure protocols, missing

security patches and insufficient level of access control (e.g., absent or default passwords). The attackers could use public device search engines, e.g. [46, 216] or develop their own scripts to search and identify ICS devices of interest. For instance, threat actors may search for PLCs with specific protocol support [198], exposed control panels or HMIs [143] or complete systems such as wind turbines [300]. Once the devices of interest are discovered, threat actors can further use open-source intelligence (OSINT) methods to geolocate devices and identify host organizations.

6.3.2 Discovery

This stage refers to discovering information about the physical process of interest from the documentation and through traditional network-based reconnaissance activities. In the context of a production site, the attacker must reconstruct the layout and configuration of the plant and how it carries out its functions, find process design weaknesses or flaws which could be taken advantage of and identify the exploitability of the related ICS components. Below we describe the types of activities at the Discovery stage.

Industrial facilities are usually uniquely customized and even operational sites within the same industrial organization can be very dissimilar. This does not only relate to the sector-specific aspects and architecture of the installation but also to the proprietary configuration and operational methods. Therefore offensive capabilities in the process control domain exist in relation to a specific target, which must be scoped by the attacker upfront. Without detailed knowledge about the victim environment, it is unlikely that an attacker can achieve more than a nuisance. For instance, simply trying to destroy a process by overheating a tank will probably only result in exercising the emergency shutdown logic and the pressure relief valves.

There are several data sources that describe the production process. The attacker may first study general information on the chemistry, kinetics, and thermodynamics of the physical processes of interest. This can be done by consulting the open literature as well as proprietary information of process design companies. Regulatory filings describe the inner workings of safety or environment-related subsystems. Engineering diagrams may be stored in cloud-based change management systems. Operator consoles offer a human-readable overview of process design and operating constraints [149].

Espionage and OSINT. Most data related to industrial processes are proprietary in nature and are therefore not easily available to a wider audience. This has been realized by attackers almost two decades ago manifesting in non-stop massive espionage campaigns against ICS-related research institutions, industrial organizations and suppliers. Early examples are [12, 57, 221, 257] and lately [48, 149, 264]. The attackers appeared to be especially interested in design documents, formulas, manufacturing processes and research materials, engineering diagrams, HMI screens and other related information.

At the same time, with the wider usage of the Internet, more and more information became available on the web, presenting new challenges for critical infrastructures. Open-source intelligence, which refers to legal information gathering from public sources, became a popular method of intelligence collection. Konstantinou et al. [170] showed how public information can be utilized in order to identify critical regions of the smart grid and which specific electrical utilities need to be targeted to achieve large-scale power outages. Further, the authors provided examples of public sources which may reveal the technologies (e.g., communication protocols, models of controllers, etc.) employed in the target utilities. While it remains uncertain how a specific substation was selected to carry out the Industroyer attack on the Ukrainian power grid, there is a possibility that the affected substation was targeted because of its detailed network diagram, an HMI screen with the electrical connections and switchgear locations were publicly available on the web page of its

contractor [93]. Additionally, several official videos [66, 307] showed the exact brands and models of the control and protective equipment used in the substation together with an overview of the field equipment and the supporting OT infrastructure.

Virus Total [304], an online public service for analyzing files for malicious content, became a popular resource among organizations and individuals to evaluate suspicious files. However, in some cases, service users upload sensitive and even confidential files without understanding that other users can search for and download uploaded files. In 2018 the security company FireEye conducted an analysis of the OT-related files uploaded to Virus Total and discovered a vast amount of highly-sensitive engineering, contractual and purchasing documentation [326]. In a similar effort, security company Otorio discovered thousands of project files (control logic) uploaded to Virus Total [86]. Besides actual process automation logic, project files may contain network configuration, screen definitions, hardware and software configurations, and therefore could be of great assistance in developing targeted attack payloads. To confirm their hypothesis, the researchers created tools for automated parsing and analysis of project files and illustrated the process of deducing tailored damage payloads.

Information gathered through espionage and OSINT efforts can be both target-specific and broad to support a general learning curve. These activities can be carried out as a preliminary step that precedes targeting as well as happening in parallel to or after the Access stage.

Reconnaissance. Similar to intrusion into IT networks, the attacker needs to discover the target plant from the networking perspective, enumerate important file shares, HMIs, engineering workstations and control equipment. While many of the reconnaissance objectives can be achieved with traditional IT reconnaissance means, discovery and enumeration of the ICS assets require specialized tools which collect relevant asset information such as shown in Figure 6.5. In the past years researchers published several public repositories with ICS reconnaissance tools for popular brands of control equipment [77, 279, 296]. Also, threat actors were observed to design such capabilities for their attack campaigns with an OPC scanner in the Havex attack and asset discovery modules in the Industroyer and Triton exploitation frameworks being a few examples.

Order Code	Module Type Name	Firmware Version	Module Name	Serial Number	Rack/Slot
6ES7 412-2EK06-0AB0	CPU 412-2 PN/DP	V 6.0.3		SVPF126xxxx	0/3

Figure 6.5: Results from the S7comm asset discovery module from the ICSSPLOIT toolkit [77]

Process Comprehension. Plants can be highly proprietary. Even if restricted to the exact same chemical process, different engineers would make different choices when selecting vendors for pumps and valves. That, in turn, would influence the sizing and placement of various pipes and holding tanks. That in turn would change the way the plant is controlled, the design of the control loops and the programming of the controllers.

There exists little documented knowledge with regard to the collection of information about a target process, its analysis, and inclusion within a targeted attack. “Process comprehension” is a term coined in [124] to describe the understanding of system characteristics and components responsible for the safe delivery of a service (e.g., treatment of water). This includes all relevant physical and computational attributes. This is the most difficult and time-consuming activity in the Discovery stage. The inability to obtain vital contextual information can become a major hurdle for attackers and result in the inability to compile a final attack payload.

Among the most essential plant design documents are Piping and Instrumentation Diagrams (P&ID), Flow Diagrams, One-Line Diagrams which often contain information on safety interlocks,

Cause & Effect Diagrams, Cable Schedule Diagrams, Project Interconnection Diagrams and others. Instrument I/O Lists contain the listing of instruments with vital information such as the type of instruments and their location in the process, range of set points, instrument tags and loop numbers, service descriptions, etc.

Instrument Datasheet				PRESSURE TRANSMITTER							
1	Tag No.			01-PT-510				Manufacturer: Yokogawa			
2	Loop Service			Reactor01-R-510				ModelNo: EJA110A			
3	P&ID No:	Line Number		01-220-004		01-P007-80-B1					
4	Area Classification			Zone 1, GrIIC, T3							
5	Ingress Protection			IP 57							
6	PROCESS CONDITIONS										
7	Fluid	State		HC		Vapour		Process Design Conditions			
8	Pressure	Normal	Max	1450 Kpag		1650 Kpag		Design Pressure	Min/Max	-	-
9	Temperature	Normal	Max	100 C		149 C		Design Temperature	Min/Max	-	-
10	TRANSMITTER										
11	Instrument Range	LRV / URV / Units		-0.5	14	MPa		Output Signal Type	4-20 mA		
12	Callibration Range	LRV / URV / Units		0	1700	KPag		Protocol/Version	HART		
13	Accuracy			+/- 0.75% of span					Burnout	Downscale	
14	Elevation	Supression		-		-		Installation Style	Horizontal Impulse		
15	LP Proc. Conn.	HP Proc. Conn.		1/4" NPT-F(Vent to Atm)			1/4" NPT-F		Mounting	Via Manifold	
16	Conduit Connection	Power Supply		2xM20 Female			Nominal 24VDC IS		Other	See Note 6	
17	Housing	Paint		Low Copper Cast Alu			Epoxy Resin Coating		Tag Plate	SS304 Permanent	
18											
19	ELEMENT										
20	Element Type	Element Material		DP Capsule		SUS316L		Temperature Limits	Min/Max	-40 C	-
21	Measurement (Gauge / Abs / Vac etc)			Gauge				Pressure Limits	Min/Max	-	-
22	Body Material	Body Rating		SCS14A		16 Mpa					
23	Bolts	Seals		SUS630		Teflon Coated SUS316					
25	Fill Fluid			Silicone Oil							

Figure 6.6: Instrumentation and Control System engineering software [20]

In the past such documentation was stored in form of standalone files on data servers, in change management systems and in the engineers' personal computers. The attacker would need to compromise and search multiple hosts to collect the necessary information. In response to the growing complexity of control and field instrumentation, demand for operational efficiency and digitalization trends, ICS vendors introduced dedicated applications for centralized management of process engineering efforts and documentation storage. Figure 6.6 shows a user interface for a pressure transmitter in an Instrumentation and Control Systems Engineering application from Aveva. The consolidated data include instrument service (Reactor control loop), tag number in the control logic, process conditions in which it is intended to be operated, instrument calibration and range, vendor and model, associated equipment and instrument tag on the P&ID diagram and even a user name of the maintenance personnel who has permissions to modify the instrument configuration. An example described in [124] illustrates non-trivial challenges faced by the attacker in achieving the level of process comprehension needed for achieving tailored physical impact by means of the network- and host-based MITM attacks. Access to engineering applications that concentrate large volumes of data on the process infrastructure would significantly reduce attacker efforts in locating and collating the information required for process comprehension.

6.3.3 Control

Some information about the process can be studied statically, but other information can only be determined empirically via dynamic testing. No engineering diagram is detailed enough to accurately predict the travel time of a shock wave down a pipe to the accuracy needed to, e.g., set

up a resonance wave to cause a water hammer effect (see Figure 2.19). Hence such data must be extracted from the live process. The control stage studies what actuators of interest do and the side effects of their manipulation.

In dynamic systems such as cyber-physical systems, the values of process variables change with time according to the laws of physics. The transitioning of a process from one state to another is in most cases not instantaneous and adheres to the well-known fact that “things take time”. In the control stage, the attacker tries to discover the dynamic behavior of the process which can be described in the form of a simple differential equation $dy/dt = f(y, u)$, where u is an independent variable and y is the dependent variable which are related by cause-and-effect relationships.

Most physical processes are inherently non-linear systems system in which the changes in the output are not proportional to the change of the inputs. Depending on the conditions, changes in process variables over time may be unpredictable and counterintuitive to a non-expert observer as can be seen in Figure 6.7. This is why making a process misbehave in a predictable way can be a non-trivial task. In general, even with a comprehensive knowledge of the exact plant configuration, it is not always possible to correctly determine whether a certain process response is related to a fundamental property of the plant or is a result of a specific attack parameter choice.

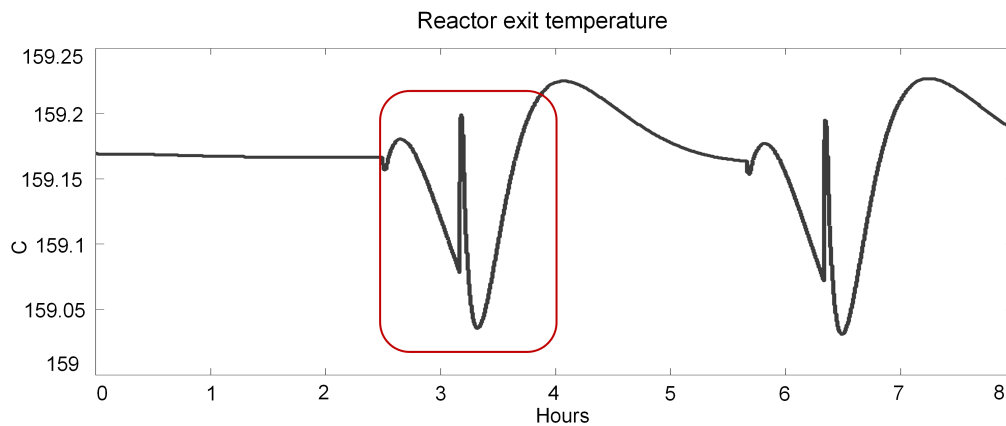


Figure 6.7: Nonlinear response of a control loop

An attacker facing an unknown process is challenged by a lack of knowledge about the tuning and responses of control loops. Ultimately, the attacker may apply trial-and-error process probing, where errors should not trigger alarms as they may attract the operator’s attention. To date, there are no well-established solutions to the problem of how to probe a given control loop with unknown responses without raising alarms. A rare exception is a work by Winnicki et al. [320] which offers a general approach to stealthy control loop testing. Note that simply knowing operational and safety thresholds is not sufficient as the non-linear responses of control loops can trigger alarms when not expected. Therefore, understanding basic factors that influence the dynamic effects of process behavior as shown in Figure 6.8 is fundamental to process control analysis. Obtaining such information is part of the process comprehension process at the Discovery stage.

Attack objectives may require the attacker to push a process beyond its normal operating conditions. Testing the performance of the control loop in abnormal conditions is essential for attack success. Every controller is configured, tuned and tested to perform within certain operational conditions and range of input signals, and may not be effective in different operating conditions. For instance, the dynamic behavior of a process in a temperature range from 140 to 150°C may be different from when it is further heated to 160°C. If the control algorithm had never been tested for its performance in atypical temperature ranges, its control commands may result in erratic and even out-of-control process behavior.

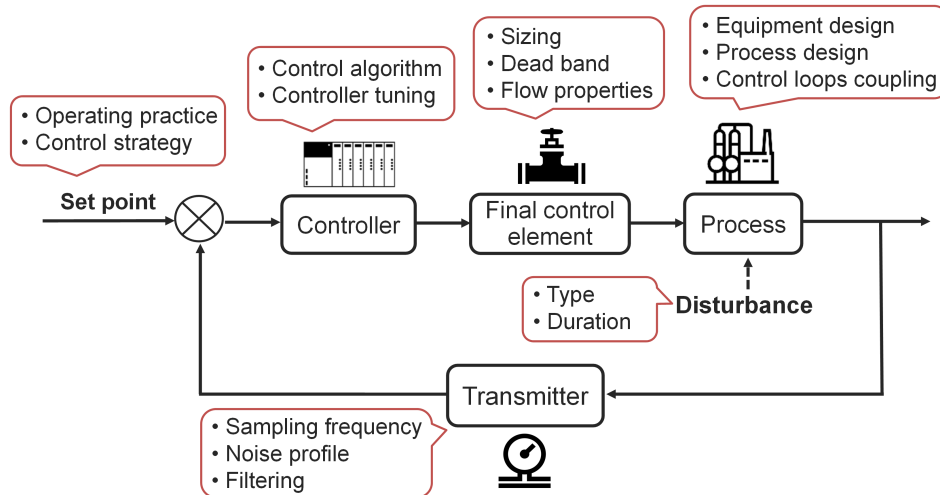


Figure 6.8: Parameters which influence control loop response

Control loop coupling is another challenge that adds uncertainty to interpreting the process response. For two control loops to operate successfully in tandem each loop must “know” what the other is doing. Otherwise trying to achieve their respective objectives the two control loops may act against the interest of the other. This phenomenon is known as loop interaction. Some control loop interactions occur naturally as a result of their physical and chemical makeup. Some loop interactions may arise as a consequence of process design. Typical examples are heat integration and recycle streams which create the potential for disturbance propagation and the alteration of dynamic system behavior.

In continuous processes even distant parts of a process may be connected together in physical relationships. Increasing the temperature in a reactor may distort the chemical reaction. Improperly reacted chemicals will propagate downstream in form of imbalanced flow composition and cause irregularities in the refining section of a plant. The attacker thus needs to develop knowledge of all side effects of process manipulation as well as understand the extent of process upset propagation. In essence, the attacker needs to “reverse engineer” the physics of the plant and determine a model of its dynamic behavior. Given the size and complexity of industrial plants, manual acquisition and systematization of the dynamic process behavior can be prohibitively arduous. A semi-automated method for identification, cataloging and visualization of process dynamics by capturing sensor readings on-the-fly is proposed in [320]. Figure 6.9 shows a proposed visualization of the plant-wide response to a crafted impulse in the example of the Tennessee Eastman process. The purpose-built algorithm extracts behavioral patterns from sensor measurements with the help of lightweight data approximation procedures paired with estimation methods.

The ability to control the actuators does not guarantee the ability to achieve control over the process. Adjusting one part of the process for malicious purposes may aggravate the process state in other parts of the plant. For instance, it may be possible to turn off a pump, but as a side effect the pressure would build up quickly in an upstream pipe rendering the process inoperable. Additionally, an action taken in one logical layer of the cyber-physical system model might be prevented by a system design in another layer. Instructing a pump to run while the flow is off with a network packet in the cyber layer may be prevented by a hardware interlock in the physical layer. Note that such physical protections are not always specified on the engineering diagrams. The control stage also involves the study of timings. For instance, if the damage occurs in seconds, a safety shutdown minutes later will not stop the attacker.

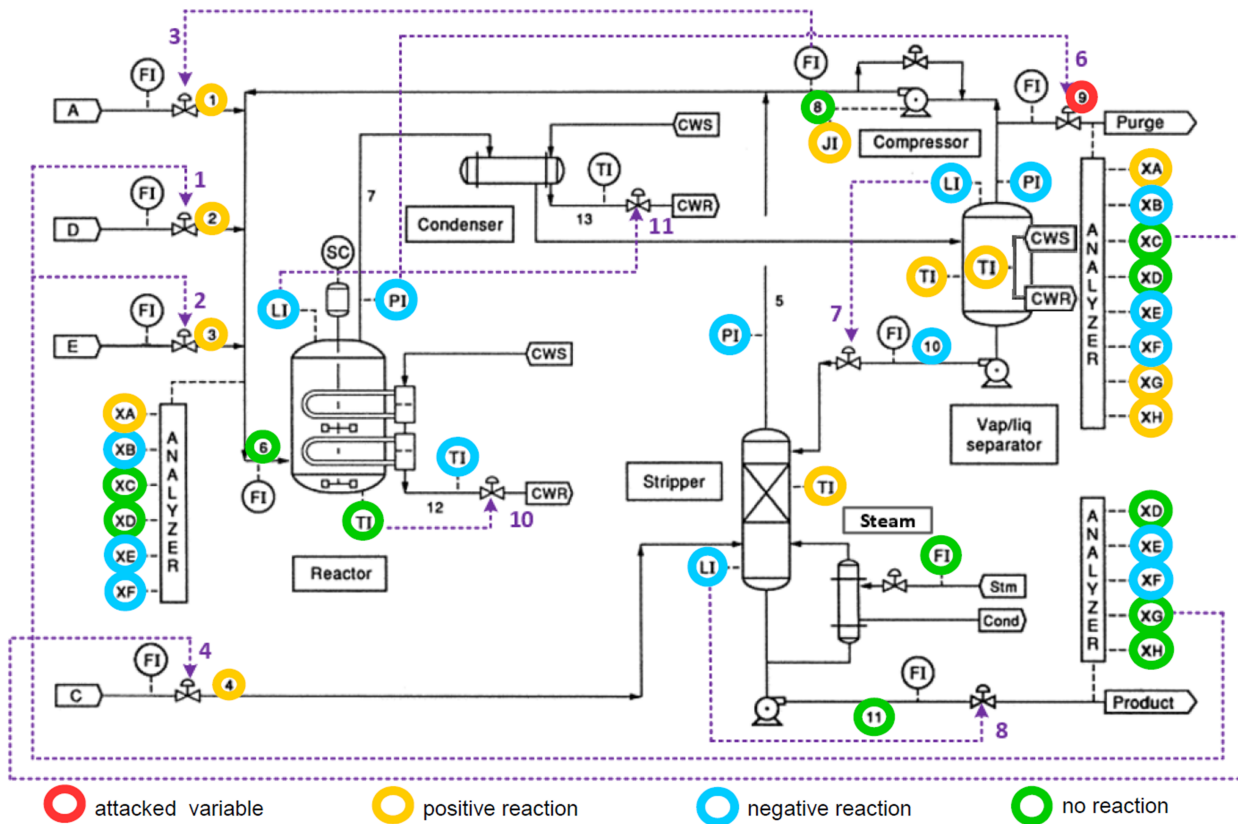


Figure 6.9: Visualization of the plant-wide process response to a manipulation of the valve [320]

The control stage is completed when the attacker identifies at least a minimum set of controls that can be applied in the Damage stage to obtain attack objectives. Note that successful control over the process will not necessarily result in the attacker's ability to cause the desired damage. This will be further discussed in the next section.

6.3.4 Damage

Once the attacker understands the process and how to control it, she needs to decide on the specifics of achieving her attack goals. It often requires the input of subject matter experts to assess the full range of possibilities. However, physical damage scenarios may not come easily even to a process engineer's mind. Engineers asked to attack a process tend to come up with what is known as "salty cookie" scenarios. A group of engineers asked to attack a cookie factory all hit on variations of putting too much salt in the cookies so they would become unpalatable. However, in a real-world situation, the actual damage to the food factory was caused by the simultaneous presence of too much product in the pipes and a failed emergency flushing system. As a result, the pipes got clogged and had to be physically replaced after both water pressure and chemical means of clearing them failed.

Accident data can be a good starting point when brainstorming ideas for causing damage. If a particular type of process has gone wrong in the past by accident, it stands to reason that an attacker may be able to make the process fail in a similar way by design. There is a number of freely available (e.g., [269]), commercial or membership-based (e.g., [244]) incident databases with detailed investigation information.

Some of the damage scenarios are straightforward such as operating rotating equipment at

its natural frequency at which a mechanical resonance and equipment vibrations may occur. Vibrations significantly reduce the expected equipment life span and lead to equipment breakage. The skip frequency parameters in the Variable Frequency Drives are used to set up a frequency band through which the drive output may pass, but never continuously operate in it to protect equipment from harmful operating conditions. Many VFDs allow a remote user to both read and modify the skip bands without authentication (Figure 6.10). The attacker strategy for causing vibration damage is quite simple: read the skip band information from the VFD, modify the skip band value to a new range, and set the output frequency of the drive to the center of the previously-observed skip band, i.e., to the dangerous natural vibration frequency. Many VFDs allow this complete set of changes to be made while the equipment is still running.

Code	Name	Logic address	Access	Type
TDC1	IDC injection time	16#28A2 = 10402	R/W	UINT(Unsigned16)
JOG	Jog assignment	16#2B66 = 11110	R/WS	WORD (Enumeration)
JGF	Jog frequency	16#2B67 = 11111	R/W	UINT(Unsigned16)
PS2	2 preset speeds	16#2C89 = 11401	R/WS	WORD (Enumeration)
PS4	4 preset speeds	16#2C8A = 11402	R/WS	WORD (Enumeration)
PS8	8 preset speeds	16#2C8B = 11403	R/WS	WORD (Enumeration)
SP2	Preset speed 2	16#2C92 = 11410	R/W	UINT(Unsigned16)
SP3	Preset speed 3	16#2C93 = 11411	R/W	UINT(Unsigned16)
SP4	Preset speed 4	16#2C94 = 11412	R/W	UINT(Unsigned16)
SP5	Preset speed 5	16#2C95 = 11413	R/W	UINT(Unsigned16)
JPF	Skip frequency	16#2C25 = 11301	R/W	UINT(Unsigned16)
PIF	PID: PI function feedback assignment	16#2E7D = 11901	R/WS	WORD (Enumeration)
TLS	Time limited speed (LSP)	16#2DB5 = 11701	R/W	UINT(Unsigned16)
FBS	PID Feedback scale factor	16#2E7F = 11903	R/W	UINT(Unsigned16)

Figure 6.10: Tunable parameters of a VFD [30]

It is important to note that cyber-physical exploitation is not necessarily achieved by delivering bits of malicious exploit over electronic media. Consider an electrical analog motor. Is it not “hackable” in the traditional way as it does not have a programmable element in it. However, engineers use Digital Maintenance and Test Equipment (M&TE) for motor configuration, calibration and diagnostics. M&TE includes firmware, which is potentially exploitable. Thus, it may be possible to place malicious code on the M&TE used when conducting a motor test. This code could be created so as to report a bad motor as being good and/or a good motor as being bad [303]. As a result, a facility may incur replacement costs before the end of the useful life of a motor, or downtime costs from an unhealthy motor failing during operation.

Not all potential damage scenarios and their successful realization will result in an effective attack. Consider a water treatment plant [218]. The process begins with the inflow of raw water to be treated through an inlet pipe. Firstly, the water is treated with chemical disinfectants such as chlorine, sodium hypochlorite and others to eliminate pathogens such as bacteria and viruses. In the next step, the water is filtered with ultrafiltration (UF) membrane filters to eliminate unwanted inclusions. Filtered water is de-chlorinated using ultraviolet (UV) lamps and then fed into a Reverse Osmosis (RO) system for final filtering. The filtered water from the RO unit is supplied to consumers and the reject water is stored in the UF backwash tank. A backwash process cleans the membranes in UF at preset times or conditions.

We assume that the attacker’s goal is to damage the UF filter which would result in an extended water supply interruption and monetary loss due to filter replacement. Filter membranes are sensitive to several hazards such as grease or overpressure. This information can be found in the UF filter operating or service manuals, e.g., [130, 131]. While increasing the number of grease

particles in the water via a remote cyber attack can be difficult or impossible to achieve, creating overpressure conditions appears to be a more realistic scenario. The attacker can turn to the engineering documentation of a water plant to work out an attack that causes overpressure. For instance, it can be discovered from a P&ID diagram such as found in [157] and highlighted in Figure 6.11 that there are two water inflows in the filter section: treated water flow and backwash flow. It is logically understood that these are mutually exclusive flows that should never be allowed to flow at the same time. However, the attacker may find a way to overcome the restraining interlock and make two flows run through the filter simultaneously, causing high pressure inside of the filter. Note that Figure 6.11 shows the potential point of attack but is not intended to explain the entire design of the water treatment plant.

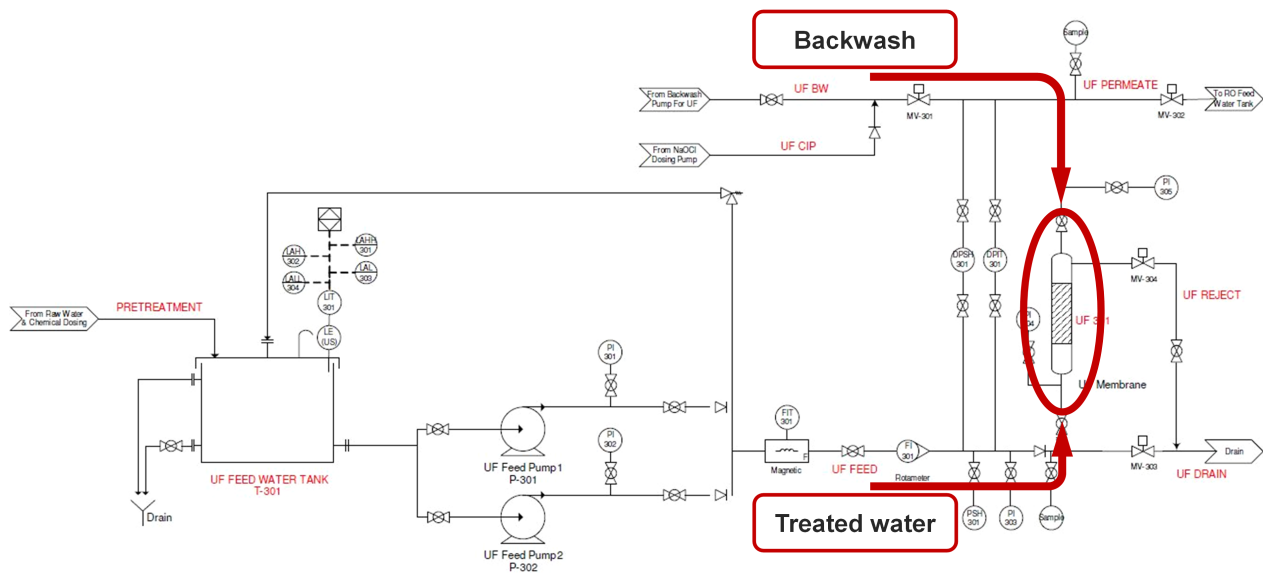


Figure 6.11: P&ID of the ultrafiltration section of a water treatment plant [157]

To practically implement this potential attack scenario the attacker needs to analyze the PLC code. There are three conditions that can trigger the backwash process, each guided by a state machine in a PLC: a preset timer (every 30 minutes), UF filter differential pressure (DP) ≥ 40 kPa or plant shutdown. In collaboration with the iTRUST research center [155] we implemented an attack based on a preset timer and enabled the concurrent flow of both streams through the filter. The result of our experiment is shown in Figure 6.12. The normal average UF filter DP is $\approx 12 - 13$ kPa. The maximum differential pressure we were able to achieve was ≈ 98 kPa (1 bar). According to the filter documentation, this pressure is not enough to damage filter membranes. This example illustrates that the successful implementation of a potentially viable attack will not necessarily cause the damage intended. In such a situation the attacker has two options. The attacker may return to the Control stage and examine whether the flow properties (e.g., velocity) of both or any of the two flows can be modified to achieve the desired pressure in the filter. Alternatively, the attacker may return to the Discovery stage and work out new potential scenarios to disrupt the water treatment process.

An attacker is not restricted to a single scenario and different attacks are not mutually exclusive. It may be possible to attack one part of the process and then later attack another part based on the success of the first. An attacker may never have full knowledge of the process and the environment. It may also be impossible for the attacker to fully test her code before deploying it. Chaining together multiple attacks into a single payload maximizes the chance that one of them will have the desired effect. Note that while we adhered to the usage of “damage” throughout the chapter, it

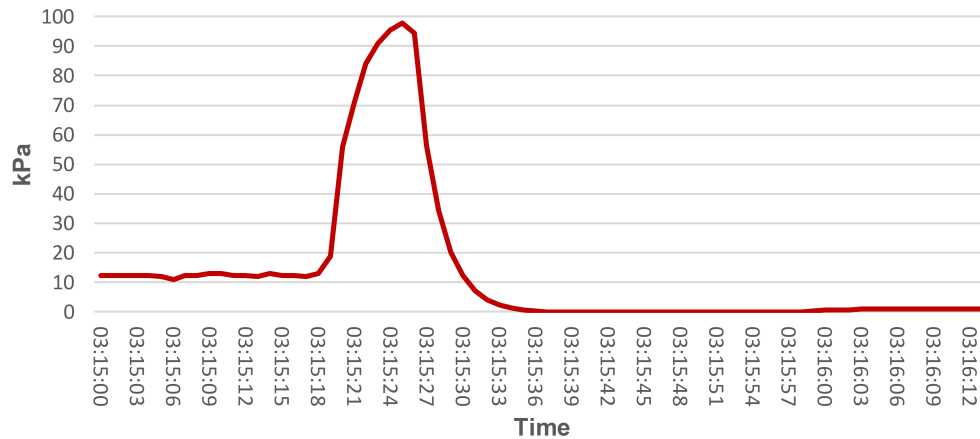


Figure 6.12: Differential pressure on a membrane in UF water filter

would be useful to distinguish between “damage” and “degradation” attack strategies. Degradation is easier to achieve and therefore such attack scenarios pose a higher risk.

Prevent Response

The main task of the control system ecosystem is to keep the process state as close to its set points as possible. The attacker will inevitably move the process away from its optimum, “provoking” compensating reactions. During the attack, the control system and the attacker compete for control over the process. Therefore, the attacker may require to obstruct the control system from “performing its duties”. In this attack stage it is important to remember that the control loop is not a single control loop but rather a hierarchical structure of control loops as shown in Figure 6.13.

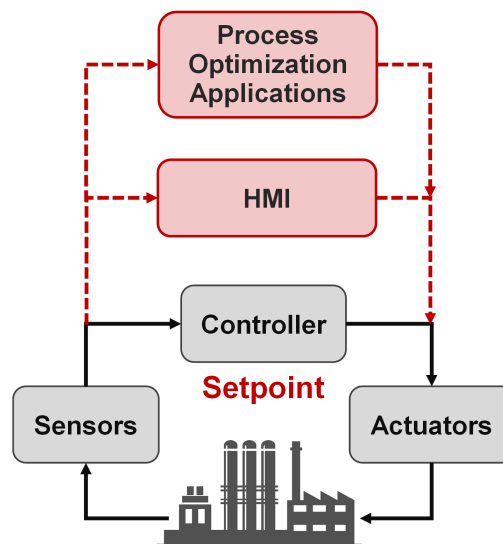


Figure 6.13: Hierarchical architecture of control loops

The control ecosystem includes the control logic, process operators and all other applications which contribute to control decisions, e.g., advanced process control or equipment monitoring applications. The attacker may need to hide attack artifacts across all control loops to be successful. For instance, it may be sufficient to keep the process operator’s attention away to prevent a

timely response. However, process optimization applications may automatically act on suboptimal operational process data and, e.g., calculate new process set points or adjustments to relevant valve positions to balance out process inefficiency.

Approaches to preventing a control (eco)system response include but are not limited to techniques of spoofing process measurements as discussed in Section 5.3, poisoning of Historians with pre-calculated data, various methods of alarm suppression (alarm hiding, relaxation, shelving and others), avoidance of alarm activation and disruption of feedback and control communication channels. An attacker may have to use different strategies across different hierarchical control loops.

Process state and alarm information propagate not only vertically but also horizontally, affecting process variables both upstream and downstream. To identify the *extent* and *duration* of process upset propagation, the attacker needs to see the process status and alarms for the related process unit(s) or even the entire plant. Good places to access such information could be historian applications. They are also often mirrored in near real-time to the corporate network so that they can be queried by various regulatory processes.

Note that in the experiment with the UF filter described above, the maximum differential pressure was achieved in 8 sec. If the high pressure damages membranes instantaneously, such an attack may not even need concealment measures as the operators would have very little time to acknowledge, react and prevent the accident. However, if damaging the filter takes some time, the operators may intervene and the attack may fail. Therefore, timing parameters of the physical process need to be taken into consideration when designing a cyber-physical attack.

Obtain Feedback

There may be several competing attack designs with varying degrees of execution reliability. Therefore, the attacker may want to develop some kind of measure or metric to choose between them. To monitor and manage attack execution as well as measure attack success (or failure), the attacker needs to establish a feedback loop to observe the process under attack. This need is directly correlated to the core principle of process control that the process needs to be observable to be controllable (manageable). This principle is also similar to the utilization of a Command & Control communication channel in attacks on IT infrastructures. Depending on the attack scenario and scope of access obtained, the attacker may need to obtain a feedback loop between the placed exploit and the process itself and/or coordinate attack activities between several implants. For instance, observation of state A in component B needs to trigger payloads X, Y, Z. This could be seen as a C2 mechanism for embedded implants based on process state detection. In the event of remote access to the target facility, the attackers may use familiar C2 capabilities to relay feedback information into their infrastructure on the Internet and to monitor and adjust attack execution in near real-time (Figure 6.14). Previously, threat actors showed the capability to obtain persistent remote access to SCADA servers [55] and safety systems [159].



Figure 6.14: Architecture of C2 and feedback channels in cyber-physical attack life cycle

The target plant may not have been designed in an “attacker-friendly” way and may not measure the values needed to monitor attack performance. As a result, the attacker may face the challenge of lacking sensors or other measuring instruments needed to monitor her attack. Additionally, process information may spread across disparate subsystems forcing the attacker to invade a greater number of devices. Generally speaking, there are two types of measurements available to the attacker:

- “*Technician*” measurement is a qualitative measure, e.g., whether some process value is decreasing or increasing. A so-called “sensor proxy” can be used to measure an aspect of the process where direct measurement is not possible. For instance, pressure (proxy measurement) in some cases can be used to infer temperature (measurement of interest). Similarly, the temperature in some cases can be used to estimate flow.
- “*Engineering*” measurement is a quantitative measure, e.g., by how much some process value is decreasing or increasing. When exploiting a process, attack scenarios requiring an engineering answer are harder to implement as they often rely on data unique to the plant and its current operating mode.

In the case when observation of the process state is not achievable via available measurements, the attacker may use alternative techniques which include but are not limited to the usage of proxy sensors (see Section 6.5.4), detection of process state (Section 4.5.4), reading the status of the state machine in the PLC code, building a process model, estimations and engineering calculations (see Section 6.5.4).

Figure 6.15 illustrates the execution of an implosion attack on a metal barrel. There is no “roundness” sensor to directly measure the success or failure of the attack, requiring finding alternative ways to monitor what is happening with the barrel. In this case a pressure sensor is used to detect the engagement of a vacuum breaker which opens an air vent and allows air into the system. Figure 6.15 shows how pressure starts to raise upon vacuum breaker engagement, signaling the unsuccessful outcome of the attack.

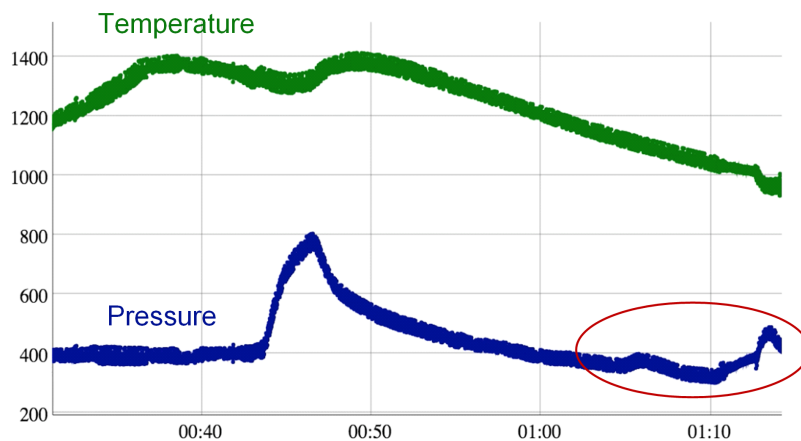


Figure 6.15: Indicator of an unsuccessful implosion attack [191]

6.3.5 Cleanup

In traditional IT attacks the goal is often to remain undetected. In most process control scenarios, this is not an option. If a piece of equipment is damaged or if a plant suddenly becomes less

profitable, a team of experts will be dispatched to investigate. A cyber-physical attack changes things in the real world that cannot be obscured by, e.g., erasing or modifying log files.

The cleanup phase is about creating a *forensic footprint* for investigators by manipulating the process, historical data and logs such that the investigation draws wrong conclusions. The goal is to get the attack blamed on operator error or equipment failure instead of a cyber event.

An example of a cleanup phase would be to show the operator a process out-of-control, making her take appropriate actions. When investigators ask the operator if she manually manipulated the process when it malfunctioned, she would answer in the affirmative. If malicious plant upsets are regularly timed so that they fall in a particular employee's shift, one might end up investigating the employee rather than the process.

The human operator is one of the key elements in monitoring industrial processes. Having a human operator in the control loop (Figure 6.16) turns the process control system from a pure cyber-physical system into a socio-technical system (STS). To take advantage of the operator's "vulnerabilities" the attacker needs to understand the specifics of the operator's job and act according to identified weaknesses in the operator's attention, judgment process or standard procedures he has to follow.

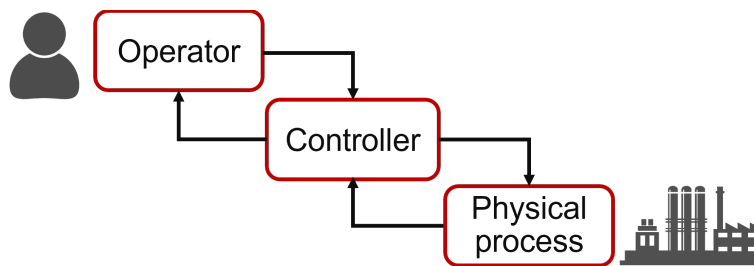


Figure 6.16: Socio-technical system

The investigator may be persuaded that a change is environmental in nature if attack patterns are timed to rainy or hot days. Replaced equipment is another good candidate for assigning blame. Most chemical plants are harsh environments and components fail regularly. If an attacker either intentionally makes a controller inoperable or waits for a component to fail naturally before beginning with her cyber-physical attack, the investigator may believe that process issues result from the suboptimal tuning of a new component instead of a cyber attack. The list of potential events that can be used as a decoy for an attack includes but is not limited to:

- Change in operating conditions: change of set point, change of raw material supplier, new equipment, etc.,
- Maintenance work: scheduled or unscheduled,
- Specific events: change in weather conditions, a particular operator on duty, etc.

An attacker may also hinder an immediate root-cause analysis of a malicious process upset by causing a so-called alarm flood [265]. An alarm flood is an event of too many alarms being raised in a short period of time, making recognition of the causal alarms difficult or impossible. Similarly, the attacker may intentionally bring about more critical alarms requiring immediate operator attention and with that taking their focus away from the ongoing attack.

6.4 Targeted Attack for Production Damage

In this section we consider an attack scenario of a negative impact on plant economics over an extended period. At the outset of attack planning, the attacker needs to consider key factors which influence attack design decisions.

One way to *influence plant profitability* is to make a process inefficient by inducing process disturbances and/or provoking control loops instabilities to increase maintenance efforts. However, creating loop instabilities can be a risky option as in the case of unskillful manipulations, the process can become completely uncontrollable and result in plant shutdown. Plant shutdown is risky for the attacker as it may instigate an investigation. For this reason, an attacker would want to remain in control of the process and to adjust disruptive actions over time.

To *persist* with her malicious objective, the attacker would want to attract as little attention from process operators as possible, e.g., by suppressing alarm activation on the operators' screens. However, alarm suppression adds complexity (and cost) to attack design. Out of this consideration we decided to manipulate the process without triggering alarms to save efforts on alarm suppression or process measurements spoofing. Another condition of persistence is avoiding attribution of the prolonged process misbehavior to malicious causes. This can be achieved by, e.g., timing the attacks to specific events in order to misdirect operators' suspicion as discussed in the previous section.

Preliminary Analysis of VAC Process

For our experimental work we use the Vinyl Acetate Monomer process model due to its complexity and known realistic components. This allows us to investigate the attack execution process close to the real-world conditions.

Economic damage to a plant can be achieved in several ways. To narrow down the most effective attack scenario the attacker can first take a bird's-eye view on the plant structure. The VAC plant can be roughly divided into three parts: reaction, refinement and finished product output as shown in Figure 6.17. Distilled products represent the most valuable commodities leaving a refinery or chemical plant. Therefore maximum economic damage could be achieved by destroying the pipe that carries the final product into the storage vessels. This attack is certainly effective, but due to clear physical evidence of its consequences it will also be noticed and fixed quickly. For a prolonged damaging effect the attacker needs less conspicuous scenarios.

The refinement section is responsible for VAC distillation to ensure a final product meeting rigid specifications. The refinement process consists of multiple units so the attacker has many opportunities to tamper with the process, but the operator also has many opportunities to notice changes and take compensating actions. Moreover, the operator may send an impure product for an extra cycle of refinement. In contrast, upsetting the reaction process in the reaction unit reliably yields reduced production of a useful product. This is why we focus attack efforts on the reactor unit.

Most factors influencing the reaction process fall under the following general three categories:

- Catalyst deactivation caused by high temperatures and other factors,
- Reduction of reaction rate due to wrong ratio and/or preconditioning of the chemical components,
- Reduction of the primary reaction due to wrong material and energy balance.

While the catalyst deactivation attack vector could potentially offer the most reliable reduction of production rate and the largest potential for a negative economic impact, we were not able to

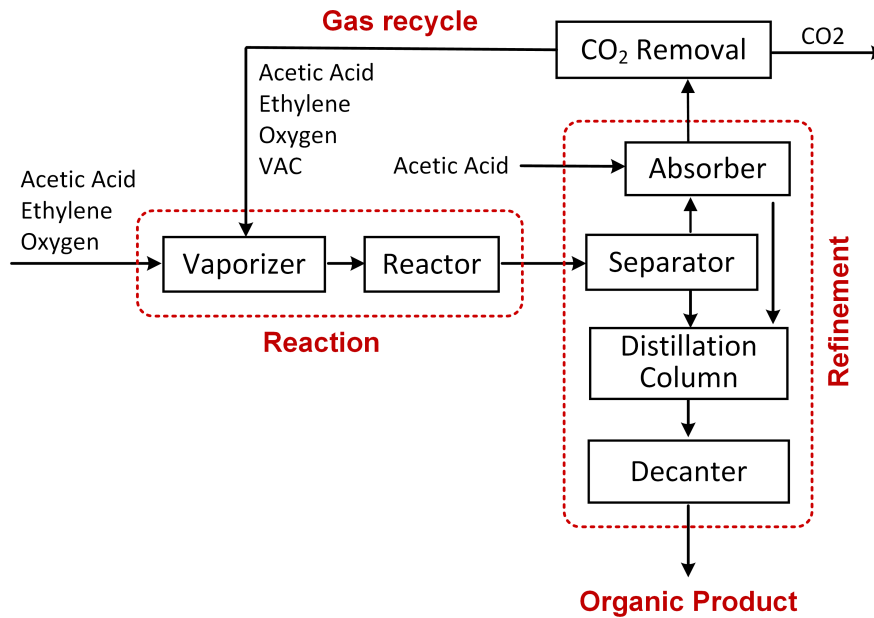


Figure 6.17: Simplified scheme of the VAC plant

realize this scenario in our simulations. Further details can be found in Section E.3. Therefore we were left with two scenarios for reducing the reaction rate. Both scenarios naturally occur as a result of manipulating actuators in the refinement session and bringing the process out of the optimal production conditions.

6.5 Achieving Attack Objectives

After deciding on a general approach to achieving the attack goal, the attacker proceeds with designing a detailed attack plan. Following the stages described in Section 6.3, the attacker develops a set of attack instances which can be scripted into individual payloads to form an overall attack framework/toolkit or included in a single final payload.

6.5.1 Access

When working with a model, it is easy to look at a process environment as static and isolated. When considering an attack on a VAC process somewhere in the world, one must remember that this is a real facility with real-world needs. As any other production facility, a VAC plant would rely on dedicated teams of subcontracting workers, engineers, consultants vendor- and third-party service providers to support plant operation, maintenance and troubleshooting all year around. Additionally, a large number of data needs to be exchanged between corporate and third parties networks for predictions, equipment maintenance and hardware, software and security updates. An attacker can take advantage of any of the above mentioned interactions to obtain access to the plant, specifically to the reactor unit.

6.5.2 Discovery

The chemical approach to VAC manufacturing is not a trade secret. There is a wealth of information on the process itself and how it is typically implemented in a factory. We assume that the attacker

has already obtained knowledge about the process such as presented in Section 3.2.2. Since we are working with the process model which does not include networking components, the discovery activities are directed at the process itself. Specifically, we concentrate on discovering measurements and actuators available to the attacker at the reactor unit as well as on a high-level understanding of the inner workings and configuration of the target plant.

Six sensors, XMEAS{1-6} and one flow composition analyzer are available to the attacker at the reactor unit, vaporizer{P;L;T}, heater exit{T}, reactor exit{T;F}, and molar concentrations of the seven chemical elements in the reactor feed stream from the analyzer, XMEAS{37-43}. Specifically, the O_2 concentration is used to monitor the hazardous conditions related to the explosivity of ethylene in the presence of oxygen.

There are seven degrees of freedom XMV{1-7} available for control in the reactor unit, three reactants fresh feeds $\{O_2; C_2H_4; HAc\}$, two valves to control vaporize{heater; vapor exit}, reactor preheater valve, and steam drum valve to control the reactor temperature. The location of the valves in the reactor unit is shown in Figure 6.18. All valves except XMV(3) control effects within the reactor unit itself. However, part of the acetic acid inflow controlled by XMV(3) is also sent into the vaporizer (reactor unit) and into the absorber (refinement section). Also, acetic acid comes from a supply tank. It means any attack on this feed will be buffered by the acetic acid holdup in the tank. Additionally, we discovered a disparity between process documentation and its software implementation. Specifically, inflow of ethylene XMV(2) is used to control the pressure in the absorber and not the pressure in the gas recycle loop. Thus any manipulation of this control loop will have effects on both the reactor and refinement sections. For the attacker this poses an additional requirement to closely monitor process state in the refinement section when manipulating XMV(2).

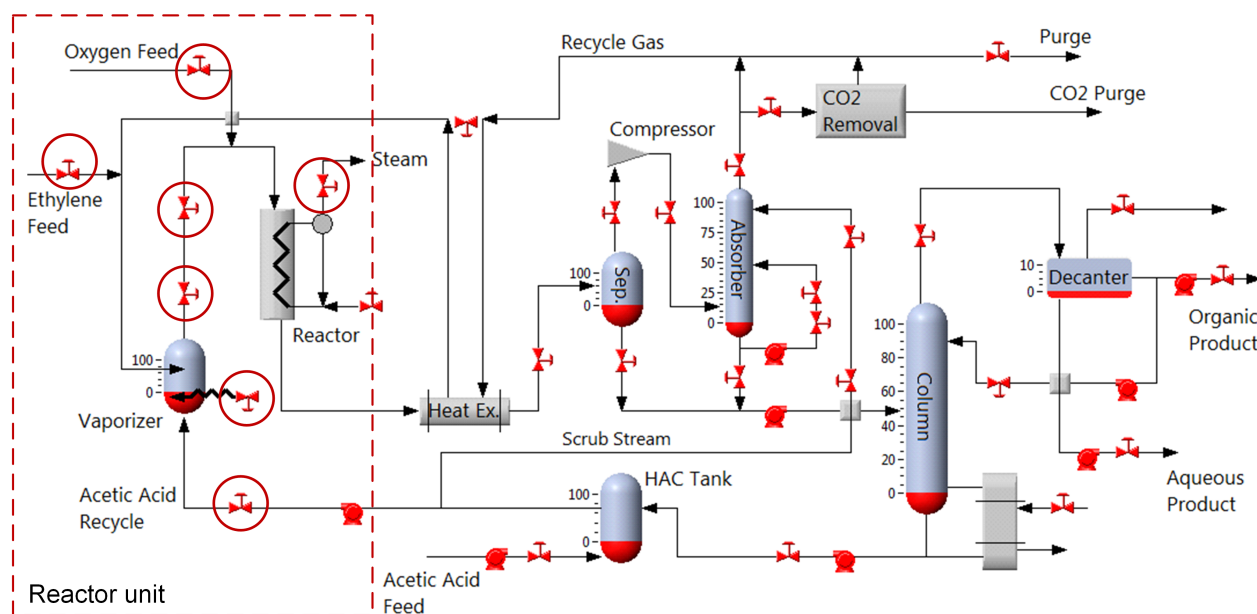


Figure 6.18: Location of control valves in the reactor unit of the VAC plant

A sensor measuring a process variable important for safety or operating constraints will have an alarm or interlock set at certain operating ranges. A quick search of chemical engineering journals turns up information on such constraints and specifics of the VAC process (see Section 3.2.2). Exact numeric limits can be discovered in operator screens, controller logic, one-line diagrams, etc. The basic plumbing of the process can be understood from its flow diagram (Figure 3.6). In a real world

scenario, this information would have to be gathered from configuration files and other sources as described in Section 6.3.2.

6.5.3 Control

At the control stage the attacker explores dynamic process behavior by triggering system responses. Typically, digital controllers are designed based on process models and with very few exceptions the designs begin with the specification of some desired closed loop properties. Collecting such information in advance at the Discovery stage may improve attacker understanding of the observed process reactions and accuracy of system identification.

Figure 6.19 illustrates VAC process response to an attack on the heater exchange and the propagation of the resulting disturbance across the plant. Even though the overall plant-wide process response can be considered as consistent, the responses of individual control loops are very dissimilar. Some of the simulation results of the VAC plant to setpoint changes are described in [211]. It was specifically mentioned that some dynamic behaviors of this process are not intuitively obvious. As discussed in Section 6.3.3, this is due to the unique hardware plant setup, configuration of each control loop and control loop coupling.

While the attacker may apply many different attack patterns, in this work we consider the two following attacks:

- **Steady state attacks (SSA)** – step-like attacks which bring the process into the new state and leave it there (Figure 6.20a). This attack is similar to a setpoint change.
- **Periodic attacks (PA)** – recurring attacks interleaved with natural process recovery phases (Figure 6.20b). This attack is similar to causing a short disturbance.

A large part of the control phase is concerned with mapping out the dependencies between each actuator and all of the downstream measurements. Mapping of the dependencies can be done through detailed modeling of the process (resource and time consuming) or observed on the live process. One possible approach is to make small changes to the process and note process response as it propagates through the various sensors. Whilst working with a model one can safely apply any input to the process and observe process response, including alarm generation and propagation. When working with the actual process, the attacker would need to make minor changes and then extrapolate how large a change was necessary to cause an unwanted result while avoiding alarm activation.

We applied the following strategy to discovering dynamic process behavior. We first identified the steady state MV values. We then increased or decreased MV by approximately 1% for 30 sec and observed the process response measured by sensors. Depending on the response we increased the magnitude and duration of the manipulation and monitored the process variables for reaching operational or safety constraints.

Steady state attacks. Not all actuators are suitable for carrying out steady state attacks. Manipulation of $XMV\{4;5\}$ moves the process to its operational or safety constraints within a short time (from minutes to few hours) even if adjusted only slightly.

Periodic attacks. This attack scenario can be seen as pulse-width modulation of a steady state attack in which pulse amplitude represents *attack value* (position of the valve), pulse width stands for *attack duration* and inter-pulse distance is *process recovery time*. Examining the sensitivity of control loops to periodic attacks turned out to be laborious and challenging due to the large number of attack parameters.

One of the challenges we faced was management of the recovery (inter-attack) phase. Initially we directly set the position of the valves to their steady state values or lower. However, this strategy was unsuccessful as the majority of control loops continued drifting towards their operational and

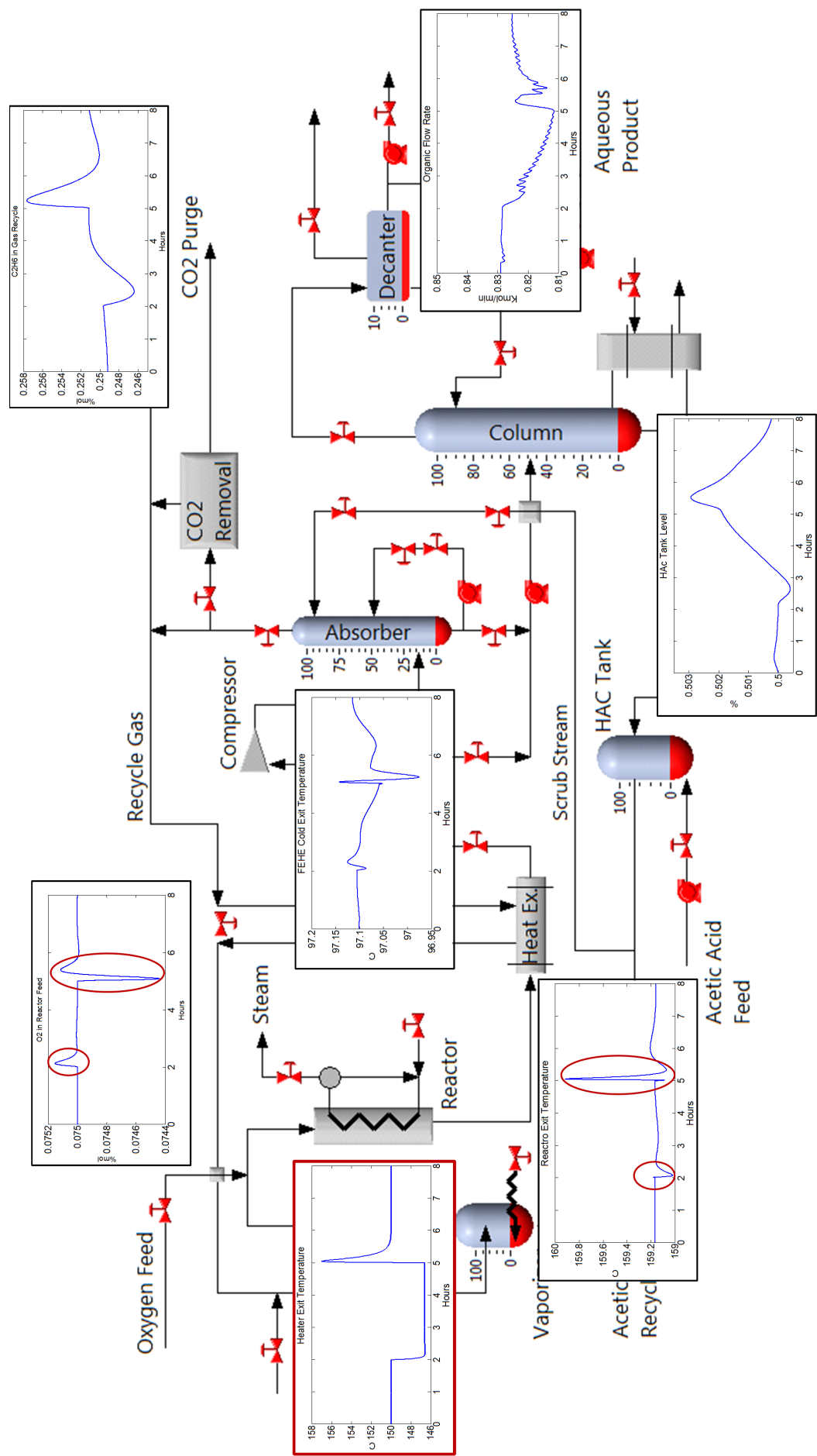


Figure 6.19: Illustration of process response across the entire plant in VAC process

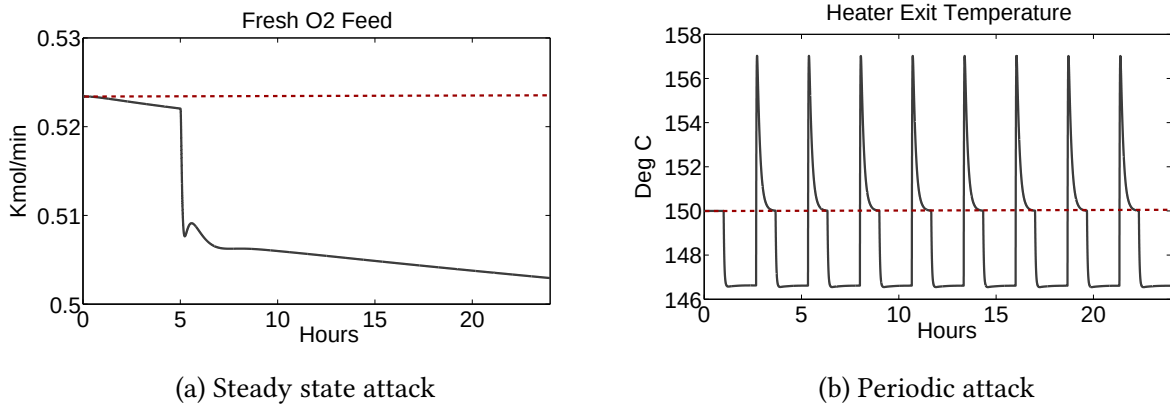


Figure 6.20: Types of attacks on process behavior (red line denotes steady state value)

safety constraints. This was due to the fixed MV values which removed the ability of the valve to adjust dynamically to the process state. Therefore we decided to leave administration of the process recovery phase to the controllers.

Another challenge was control loop ringing while manipulating XMV(5), which controls the vaporizer exit flow. Sometimes, digital controllers produce a control signal that oscillates with decreasing amplitude around the final equilibrium level (Figure 6.21a). This phenomenon is known as “ringing” and is caused by negative real controller poles. Ringing is an unwanted effect as it increases the wear and tear of controller components and can cause system instability within a multi-loop environment. Manipulation of XMV(5) in the negative direction (decreasing its value) causes a large overshoot which leads to process instability. Note, that even though the ringing appears to be of a very insignificant amplitude in the vaporizer pressure control loop (Figure 6.21a), it causes a ringing impact of ratio 1:150 in the vaporizer exit flow control loop (Figure 6.21b).

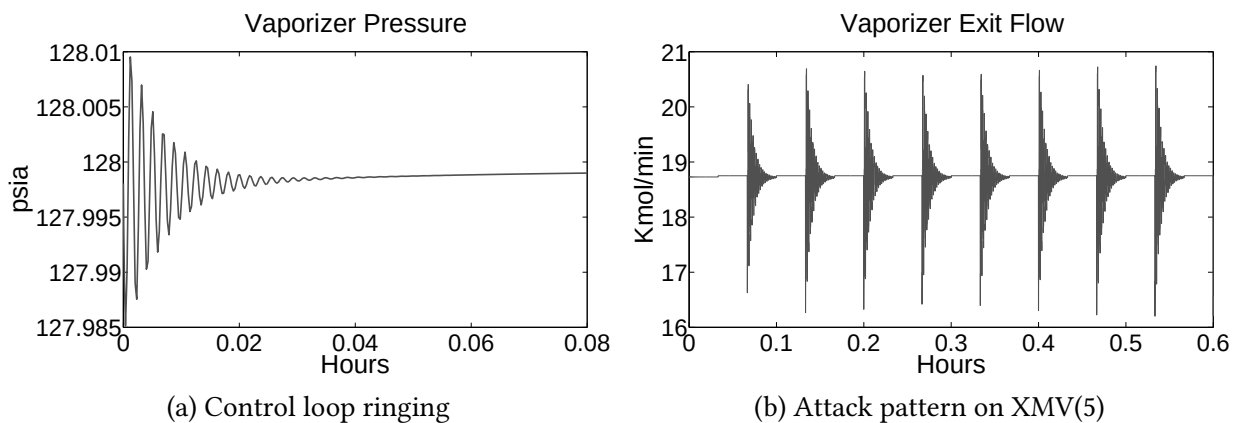


Figure 6.21: Manipulation of control loop 5 (vaporizer)

Despite an unwanted ringing effect in the XMV(5) control loop, we still pursued this attack as reducing the inflow of reactants would result in decreased production of VAC. To overcome the above challenge we decided to take advantage of the negative compensation reaction in the process recovery phase. In our approach we slightly increased the flow for 1-2 minutes and let

the process recover for 2 minutes or longer (Figure 6.21b). In the recovery phase the controller decreased the flow to bring the controlled variable to its set point. In this way we were able to achieve a reduced flow (when averaged over time). In total, there are three control loops in the reactor unit which can become unstable under certain attack parameters. Those are $\text{XMV}\{2;4;5\}$.

Outcome of the control stage

We have exhaustively tested all control loops for their sensitivity against a large number of attack settings combinations {attack value; attack duration; recovery time}. Although we could establish a good mental model of process behavior, we needed to find a systematic way to categorize controlled loops. We chose two parameters:

- **Sensitivity to magnitude of manipulation (MM)** – how much can we change the process in response to control loop manipulation;
- **Required recovery time (RT)** – if the process can recover in a time equal to the attack time or shorter we consider such control loops of low sensitivity.

Table 6.3 gives the results of our analysis. Sensitive control loops are riskier to operate and therefore less suitable for reliable control from the attacker perspective.

Sensitivity	MM	RT
High	$\text{XMV}\{1;5;7\}$	$\text{XMV}\{4;7\}$
Medium	$\text{XMV}\{2;4;6\}$	$\text{XMV}(5)$
Low	$\text{XMV}(3)$	$\text{XMV}\{1;2;3;6\}$

Table 6.3: Sensitivity of control loops

We finalized our testing with the analysis of alarm activation withing the plant in response to SSA and PA attacks as shown in Table 6.4. Additionally, for each manipulated variable in the reactor section we noted upper limits of all possible attack parameters which would allow to manipulate the process without triggering an alarm (not included into the thesis due to space considerations).

Alarm	SSA	PA
Gas loop O_2	$\text{XMV}(1)$	$\text{XMV}(1)$
Reactor feed T	$\text{XMV}(6)$	$\text{XMV}(6)$
Reactor T	$\text{XMV}(7)$	$\text{XMV}(7)$
FEHE effluent	$\text{XMV}(7)$	$\text{XMV}(7)$
Gas loop P	$\text{XMV}\{2;3;6\}$	$\text{XMV}\{2;3;6\}$
HAc in decanter	$\text{XMV}\{2;3;7\}$	$\text{XMV}(3)$

Table 6.4: Activation of alarms

6.5.4 Damage

In the previous stage we evaluated the potential to control the process. In the damage phase the attacker uses identified controls to achieve the desired damage scenario. This stage is similar to the “what occurs if” approach in a HAZOP analysis (“what happens if this valve is closed”).

As mentioned above, we opted for attack strategies that do not activate alarms. This means that the overall process state would remain within the “normal” operating envelope, however, the production economics or equipment operating conditions would deteriorate (Figure 4.19). Therefore, the results obtained in *Control* stage will be used in the *Prevent Response* sub-stage to avoid detection and prevent immediate corrective actions.

For reliability, the attacker must choose one or more attack scenarios to deploy in the final payload. This could be arbitrarily chosen based on gut feeling. However, given the amount of effort it takes to mount a real attack, an attacker may use a metric to measure her success. For an economic attack a plausible metric would be the amount of *monetary loss* to the victim. In order to determine monetary loss one needs to measure how much of vinyl acetate is produced in the reactor.

Obtaining Feedback

To measure the exact amount of a specific chemical in a production stream two measurements are needed: the total flow (FT) and the molar concentrations of the total flow composition. What would immediately catch the attacker’s eye is the absence of an analyzer in the reactor exit stream (Figure 6.22).

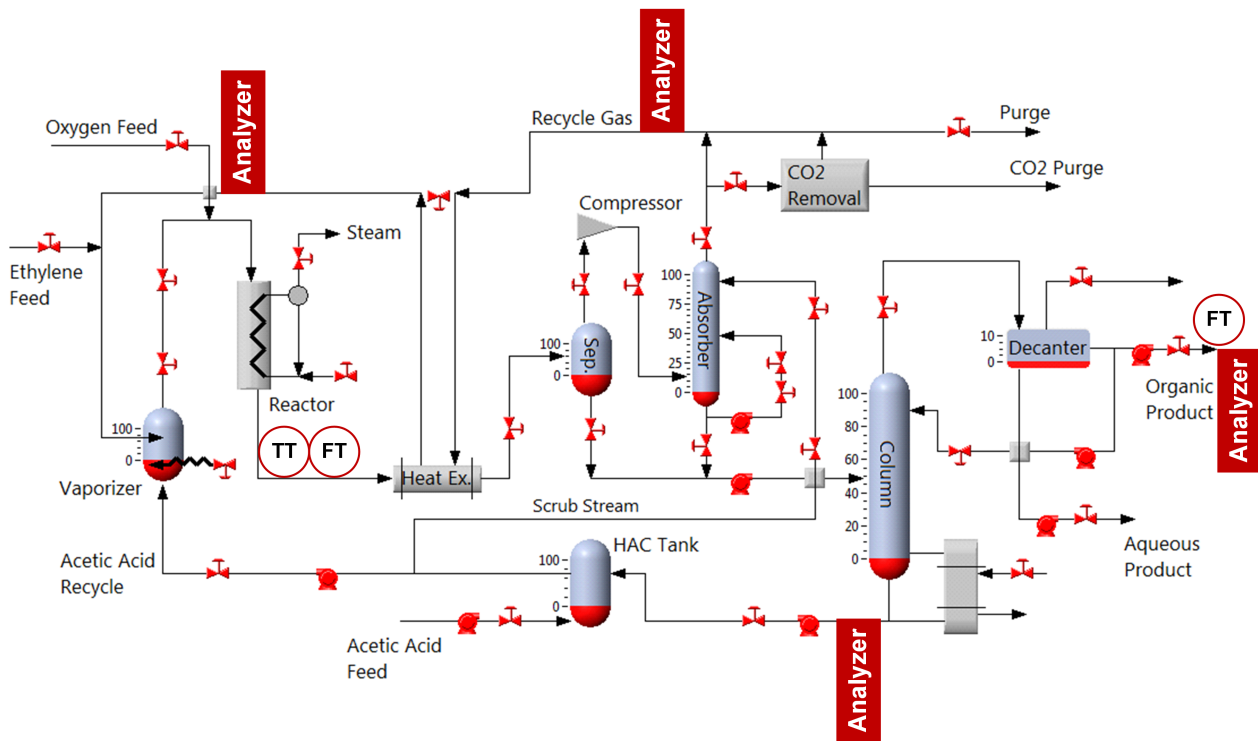


Figure 6.22: Location of chemical analyzers in VAC plant

Chemical composition analysis systems are expensive; their installation must be justified by important considerations such as safety or significant product quality improvement. This is a significant disadvantage for the attacker as she will not be able to directly obtain measurements of the molar concentration of the produced vinyl acetate in the reactor outflow. Therefore the attacker cannot immediately obtain engineering measurements to monitor and measure attack execution.

The rate of the reaction can be qualitatively determined from the reactor exit temperature (TT). This would provide an attacker with a technician measurement. A decrease in temperature signals

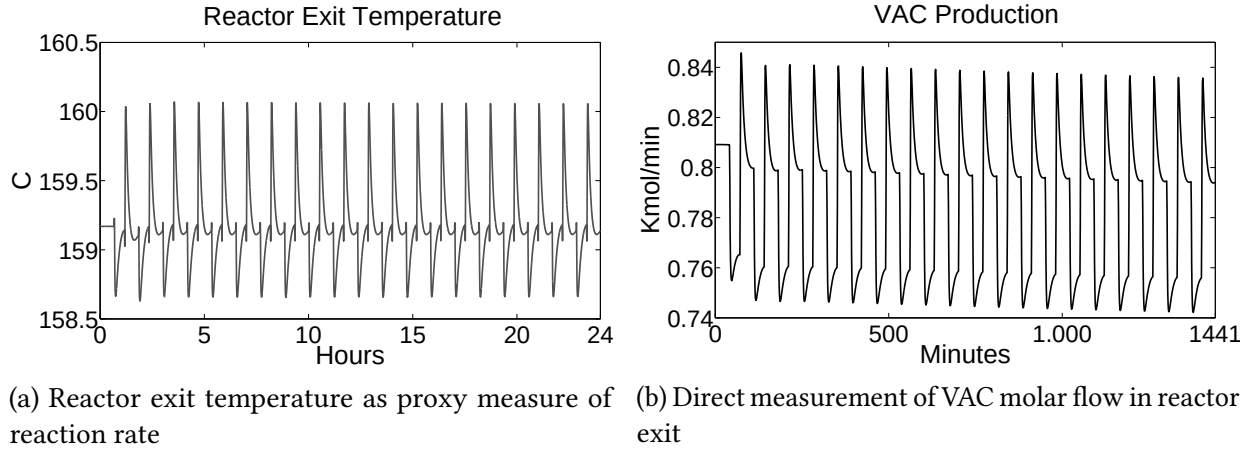


Figure 6.23: Comparison of indirect and direct measurements

that less reaction is happening in the reactor, so less product is being produced (Figure 6.23a). This measurement can be sufficient to determine whether a specific attack has an effect on the reaction rate, but it does not allow to quantify the effect of an attack and select the most effective one. Looking at the process flowsheet, the only location where the attacker would be able to determine the exact amount of VAC produced is the decanter exit. However, this number would be available to the attacker only after hours, at the end of the refinement phase. This may not be a satisfactory option. Moreover, this would require the attacker to exploit additional network segments and devices. In our analysis we have not found a way to solve this challenge.

In the real world such an obstacle would force the attacker to look into controller code or search for process models in the test plant. In our case, we decided to look into the implementation code of the process model. In particular we were interested in the state variables in the reactor unit, used in the internal computations of the process model. We were able to locate "concentration" variables of seven chemical components in ten sections of the reactor as shown in Figure 6.24.

```
//This is the main file for the reactor
//The plug-flow reactor is modeled as a 10-section distributed model
/*NR: section number + 1 */
/*calculate derivatives*/
for (n=1;n<NR;n++)
{
    /*Use component balance to get concentration change: */
    /*dC/dt=-delta(C*v)/deltaZ+sum(vij*rj) */
    /*Use single backward */
    C_O2_t[n-1]=(-(C_O2[n]*v[n]-C_O2[n-1]*v[n-1])/dz + Coeff1[0]*r_all[n][0]+Coeff2[0]*r_all[n][1])/cata_porosity;
    C_CO2_t[n-1]=(-(C_CO2[n]*v[n]-C_CO2[n-1]*v[n-1])/dz + Coeff1[1]*r_all[n][0]+Coeff2[1]*r_all[n][1])/cata_porosity;
    C_C2H4_t[n-1]=(-(C_C2H4[n]*v[n]-C_C2H4[n-1]*v[n-1])/dz + Coeff1[2]*r_all[n][0]+Coeff2[2]*r_all[n][1])/cata_porosity;
    C_VAc_t[n-1]=(-(C_VAc[n]*v[n]-C_VAc[n-1]*v[n-1])/dz + Coeff1[4]*r_all[n][0]+Coeff2[4]*r_all[n][1])/cata_porosity;
    C_H2O_t[n-1]=(-(C_H2O[n]*v[n]-C_H2O[n-1]*v[n-1])/dz + Coeff1[5]*r_all[n][0]+Coeff2[5]*r_all[n][1])/cata_porosity;
    C_HAc_t[n-1]=(-(C_HAc[n]*v[n]-C_HAc[n-1]*v[n-1])/dz + Coeff1[6]*r_all[n][0]+Coeff2[6]*r_all[n][1])/cata_porosity;
    Q_rct[n]= UA*(Tg[n]-Shell_T); /*kcal/min m^3*/
    Tg_t[n-1]=1/(cata_porosity*CCP[n] + cata_heatcapacity *cata_bulk_density)*(-FCP[n]*(Tg[n]-Tg[n-1])/dz -
    - r_all[n][0]*E_r1-r_all[n][1]*E_r2-Q_rct[n]);
};
```

Figure 6.24: Code to calculate the concentration change of chemical components in the reactor

Through extensive analysis of the obtained data and the working principle of the plug flow reactor employed in the VAC plant we could determine that the concentration of chemicals in the tenth section would be the same or about the same as in the reactor outflow. However, the obtained numerical values were of an unknown unit measure and very small, e.g., 0,00073; 0,00016; 0,0007. The sum of all concentrations values did not sum up to one/hundred or to the total flow.

After further investigations we concluded that molar concentrations of chemical components can be computed according to the formula:

$$MOL_{comp}(t) = \frac{CONC_{comp}(t)}{\sum_{comp} CONC_{comp}(t)},$$

where $CONC_{comp}(t)$ are the concentrations of the individual chemical components and $comp$ is the index of a current component.

We verified the numbers obtained with those provided in Table 5 in [210]. Since the total reactor exit flow is directly measured in the plant, we could compute the amount of vinyl acetate produced as:

$$Ouflow_{comp}(t) = MOL_{comp}(t)[\%mol] \times F_{reac}[Kmol/min],$$

where F_{reac} is measured reactor outflow, XMEAS(6).

Figure 6.23b shows the production of vinyl acetate. The rate of VAC production indeed coincides with the reactor outflow temperature profile. Knowing the molar production of VAC we could finally quantify production loss in dollar equivalents as:

$$Cost = VAC_{out}[Kmol] \times 86.09[g/mol] \times 0.971[\$/kg],$$

where $86.09[g/mol]$ is the VAC molar weight and $0.971[\$/kg]$ is the VAC price as given in [210].

To verify the numbers obtained we compared the amount of VAC produced in the reactor over a time period (Figure 6.25a) with the amount of VAC leaving the factory as final organic product (Figure 6.25b); the numbers matched.

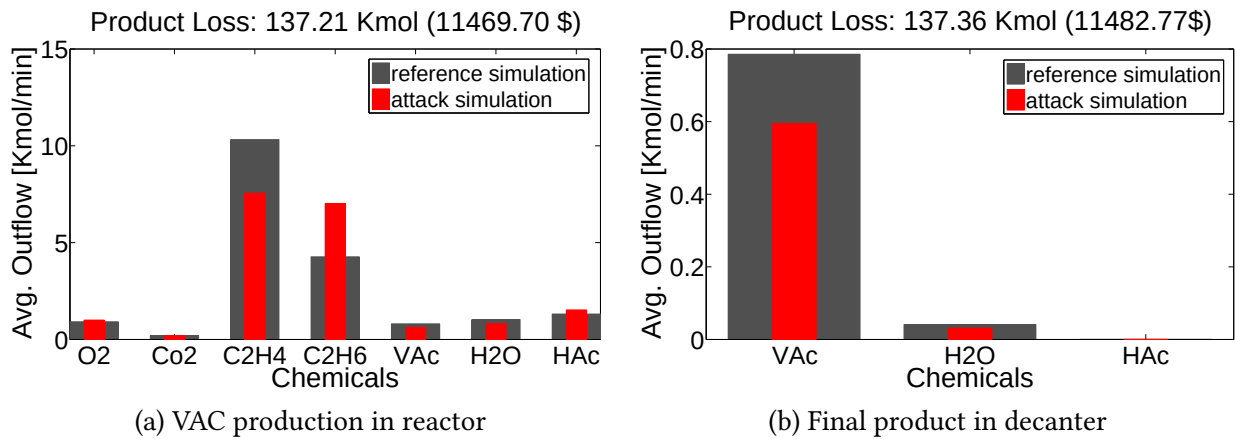


Figure 6.25: Vinyl acetate production, 24 hours

Evaluation of Attack Scenarios

With a suitable metric for evaluating the impact of the attack we could finally start deciding on the most effective attacks. We have established the reference value of the steady-state production to determine loss (or gain) as a result of the attack. The categorization of the control loops based on their economic damage potential is given in Table 6.5. Note, that certain attacks have caused an increase of the vinyl acetate produced. This does not necessarily mean that the overall financial gain of the plant as such increased as production may have yielded increased operating costs and thus may resulted in revenue loss. However, we did not pursue this investigation. Therefore we did not consider attacks causing product gain as “successful”.

Production loss	SSA	PA
High, $\geq 10.000\$$	XMV(2)	XMV{4;6}
Medium, $5.000\$ - 10.000\$$	XMV{6;7}	XMV{5;7}
Low, $2.000\$ - 5.000\$$	–	XMV(3)
Negligible, $\leq 2.000\$$	XMV{1;3}	XMV{1;2}

Table 6.5: Categorization of control loops based on damage potential

Note that attacks on XMV(1), oxygen feed, only have little attack potential, but also that this control loop easily becomes unstable. In addition, this control loop must be manipulated with great care as it quickly reaches its safety limit. Among all XMVs we marked XMV(1) as of least use.

Outcome of the damage stage

The damage phase is concluded with a portfolio of attacks which can be deployed at any opportune time. By scheduling attack value, attack duration and process recovery time we can control the amount of economic damage we would like to bring about. Note that our analysis has only determined how much money will be lost. Ideally economic damage numbers should be multiplied by the chance of success so that a risky high-damage scenario can be compared with a low-risk low-damage scenarios. Precise risk metrics may never be available, but in general attacks that require manipulating more components are considered a higher risk. Attacks that require an engineering answer (high-precision attack scenario) are more complex to design and execute and therefore are riskier than attacks only requiring a technician answer. Additionally, attacks that must hit a particular measure value or fail are riskier than attacks that simply get more effective the closer they are to the optimal value. Finally, the cost of attack implementation should be considered and compared against attack “gain”.

6.5.5 Cleanup

In our attack scenario we were aiming for attacks that do not move the process towards operational limits or unsafe conditions. Since the attack execution does not trigger alarms, the operator may not notice immediately that the process has drifted from its economically optimal operating state. However, process operators may get concerned after noticing a persistent decrease in VAC production and may try to fix the problem. There can be numerous reasons for a process upset and operators are used to them. In this section we discuss how to influence the operators’ beliefs about what is happening with the process.

An industrial process, just like software, has to be debugged when it malfunctions. If the attacker changes her attacks based on the debugging efforts of the maintenance engineers, future

attacks may be attributed to the efforts of the engineer rather than a cyber attacker. Figure 6.26 illustrates four different attacks which show themselves as symmetric fluctuations of different amplitudes in reactor exit temperature. These attacks are chained together and rotated in response to debugging efforts such as sensor calibration or reactor troubleshooting.

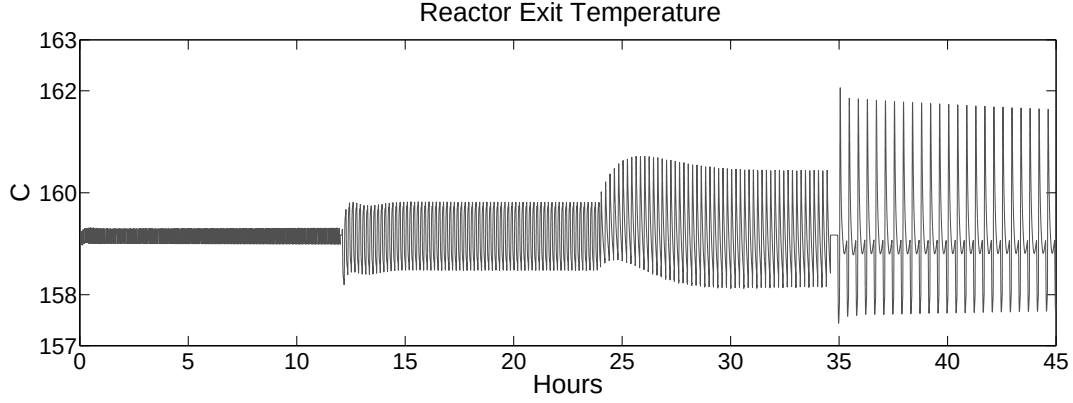


Figure 6.26: Increasing variation of reactor exit temperature caused by attacks on different control loops

If a reactor is suspected to malfunction, a group of experts will be invited to investigate. It is not possible to see directly into the reactor. The investigators will apply specific metrics allowing them to evaluate the chemical processes in the reactor and –hopefully– determine potential reasons for the deterioration of reactor efficiency. They will then schedule maintenance work in accordance with identified issues. The attacker can execute different attacks which have the same effect on specific chemical processes in the reactor making engineers believe that their maintenance efforts are not bringing the expected results.

Typical examples of such metrics would be selectivity and conversion rate. Selectivity is a metric to control catalyst activity. Catalyst selectivity determines the fraction of the ethylene consumed that makes the desired VAC product or, in other words, how much (in percent) of the primary reaction has been induced by the catalyst:

$$SEL(t) = \frac{VAC_{out}(t)}{VAC_{out}(t) + 0.5 \times CO2_{out}(t)} \times 100,$$

where $VAC_{out}(t)$ and $CO2_{out}(t)$ are molar flows of the respective chemical components in the reactor outflow.

Conversion determines the fraction of the chemicals consumed (converted into product and byproducts) during the reaction. This metric is informative in several ways. For instance, there are certain safety limits and best-practice conversion rates, which should not be exceeded. Thus, the reduced conversion rate of acetic acid and increased conversion of ethylene suggest an increase in the amount of the undesired secondary reaction. Conversion is computed as

$$CONV_{comp}(t) = \frac{COMP_{in}(t) - COMP_{out}(t)}{COMP_{in}(t)} \times 100,$$

where $COMP_{in}(t)$ and $COMP_{out}(t)$ are the molar masses of the chemical components in the reactor in- and outflows.

In addition, we introduced a metric to measure reactor efficiency (EFF). It computes how much molar mass of acetic acid has reacted, and compares this value to the amount of reacted ethylene.

Since the reaction ratio of ethylene and acetic acid in the primary reaction is 1:1, the amount of reacted acetic acid is equal to the amount of correctly reacted ethylene. Relating this value to the amount of total reacted ethylene indicates the percentage of the primary reaction. Efficiency allows similar conclusions as catalyst selectivity, however, it is calculated based on the converted reagents rather than on the produced products:

$$EFF(t) = \frac{HAC_{in}(t) - HAC_{out}(t)}{C2H4_{in}(t) - C2H4_{out}(t)} \times 100,$$

Figure 6.27 illustrates the processes in the reactor during the attack on XMV(2). We decrease ethylene feed at time $t = 120$ minutes.

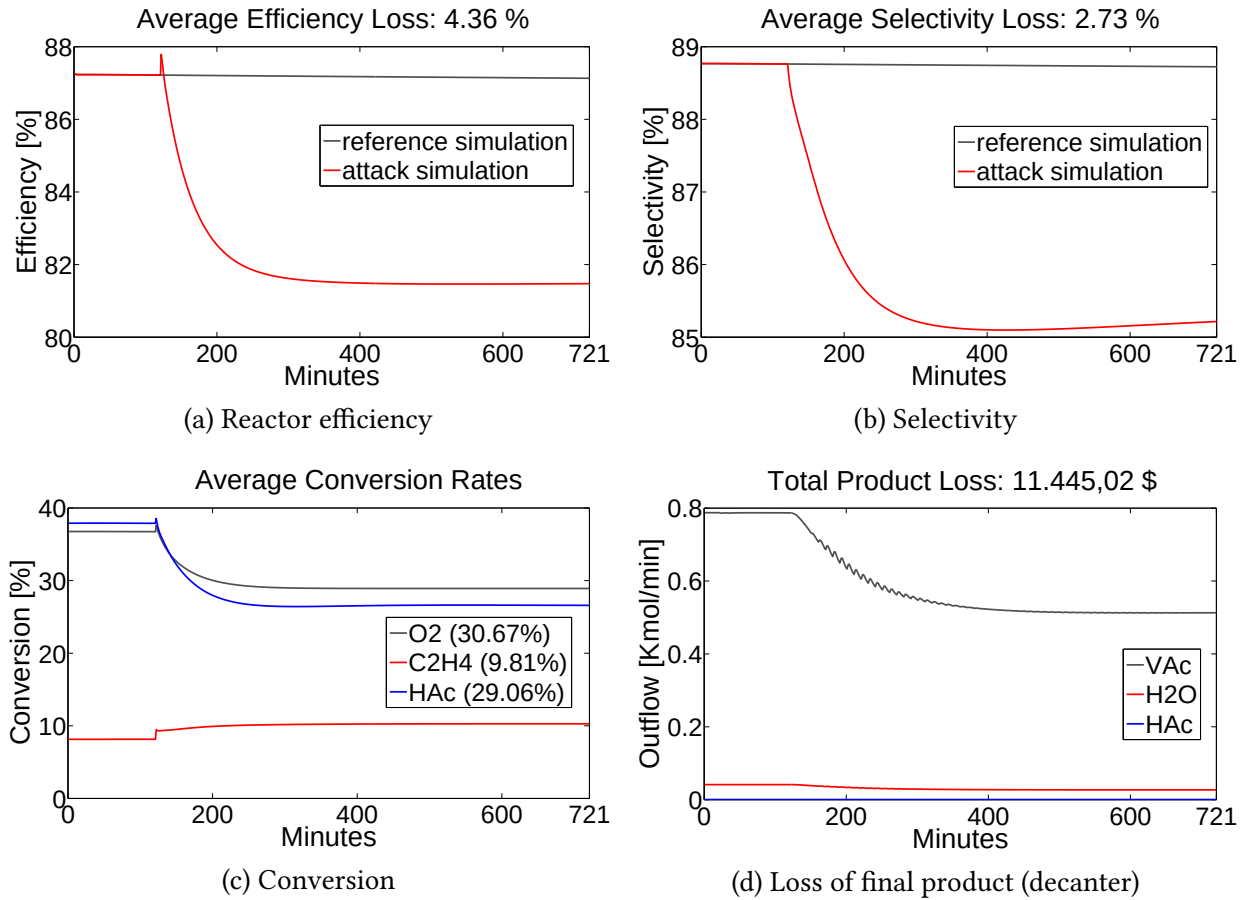


Figure 6.27: Analysis of the physical processes in the reactor

One can see how the attack affects the ratio between primary and secondary (ethylene combustion) reaction: the percentage of the primary reaction drops from 87% to under 82% and the amount of the secondary reaction increases by 4.32% on average (Figure 6.27a). Selectivity also drops to a lower level (Figure 6.27b). Since selectivity is calculated based on the ethylene consumed in both primary and secondary reactions, we can conclude that an increase of the secondary reaction has a stronger effect on the reagents consumed than it has on the products produced. In other words, the consumption of reagents is the more expressive metric in this case. Figure 6.27c plots the conversion rates for the main reagents in the reactor. Ideally, the conversion rate of acetic acid is $\approx 37\%$, around 2% higher than the oxygen conversion. However, due to the attack, the conversion

rate of acetic acid drops beneath oxygen conversion, indicating that the reaction kinetics have changed. This is because the newly induced secondary reaction also consumes oxygen (even more than the primary reaction) At the same time ethylene consumption has increased. Therefore we can conclude that the amount of the primary reaction has decreased (less acetic acid is converted), and the amount of ethylene combustion has increased. The result of the attack on XMV(2) is a significant reduction in the production of the final product (Figure 6.27d).

6.6 Discussion

Our initial attacker model restricts the attacker from triggering operational constraints. In reality, the attacker may suppress alarms while supplying the operator with good process values, e.g., using sensor signal spoofing techniques presented in Section 5.3. We intentionally chose not to activate alarms to understand attacker challenges when discovering dynamic process behavior in a stealthy manner. This turned out to be a laborious manual process with currently no established methods in the public literature to automate or support the testing and cataloging of dynamic process responses. We examined whether it could be beneficial for the attacker to violate operational constraints to cause more damage. Whereas we could almost double the loss with steady state attacks, the increase of damage in periodic attacks was modest. In certain cases violation of operational alarms eventually moved the process into an unsafe state, triggering process shutdown and attack interruption. It is, therefore, important for the attacker to monitor the plant state at all times to avoid unintentional plant shutdown. Not only does such an event slow down testing and/or execution of attack scenarios, it may also prompt an investigation. Thus an investigation of plant shutdown in the Triton attack resulted in the discovery of attack tools and triggered a full-fledged incident response [159].

The attacker is not almighty and some damage scenario might be unachievable for various reasons. For instance, initially, we attempted to achieve economic damage via catalyst deactivation. A chemical catalyst is a substance that increases the rate of a reaction without itself being consumed. It is frequently made of precious metals and therefore tends to be expensive. Catalyst productivity reduces over time requiring its replacement. The typical catalyst lifetime of VAC catalyst is 1-2 years [83] after which plant profitability decreases. Preferred operation conditions are temperatures around 150-160°C, higher temperatures within reactor tubes promote catalyst activity decay. Hot spots above 200°C lead to permanent catalyst deactivation requiring a costly production shutdown and catalyst replacement [210]. We were not able to realize this damage scenario as we were not able to overheat the reactor for long enough to damage the catalyst. As described in Appendix E.3, hot reactor outflow caused condensation in the downstream heat exchanger which quickly led to the development of unsafe operational conditions and plant shutdown. This example also illustrates why the control stage precedes the damage stage – designing a specific damage scenario prior to testing control loops for controllability conditions may result in attack failure during execution.

Because dynamic process behavior cannot be deduced from the plant documentation or control logic and can only be measured on a live process, the Control stage provides an opportunity for the defenders to detect anomalous short-term process fluctuations or unusual operations on equipment. Additionally, implementing defense-in-depth security controls and reducing the threat of unauthorized remote access to live process may significantly limit attacker abilities to develop viable attack scenarios. Besides, due to the high effort and extensive knowledge required overall to engineer and implement tailored damage attacks, it is more likely for threat actors to search for so-called “easy button” attack options. These are simple attack scenarios with easily identifiable damage conditions (e.g., operating rotating equipment at skip frequency) and reliable attack

outcome (e.g., equipment damage due to vibrations). The Industroyer attack can be considered an “easy button” because it simply turns off all discovered control signals [55] resulting in the predictable outcome of substation shutdown. Control logic inherently contains information about exceptions and unsafe conditions making it a prime candidate for discovering “easy button” attack ideas.

As discussed in Section 3.1 building and maintaining realistic testing environments is a significant financial investment. It has been suggested in [125, 247] that both attacks on the Ukrainian power grid were live drill operations by a state-sponsored threat actor to test cyber-physical sabotage capabilities. The chance exists that in the future threat actors will continue using real-world ICS environments for capability development and testing purposes. Understandingly, industrial plants with weak digital perimeters are more prone to becoming such testing environments.

With the growing awareness of cyber-security threats, industrial organizations start building stronger digital perimeters requiring attackers to search for alternative pathways into the operational environments. Project files are considered to be trusted files coming from trusted sources like subcontractors or service providers. Such files are typically directly loaded to the control network and executed with high privileges. Gurkin [132] has shown that it is possible to embed complex exploits such as callback trojan or execution of OPC methods into the project file code itself which will be invisibly launched upon project startup. The emergence of such innovative methods for delivering attack code calls for the development of additional security controls such as ICS-aware sandbox applications.

6.7 Related Work

The overall concept of describing targeted cyber attacks in form of an attack lifecycle is relatively new. It was first introduced by researchers from Lockheed Martin Corporation in 2011 under the Kill Chain trademark name [147]. In 2015 SANS Institute adapted the Lockheed Martin Cyber Kill Chain model to the ICS needs and proposed The Industrial Control System Cyber Kill Chain [19]. The model consists of two phases. Stage 1: Cyber Intrusion Preparation and Execution and Stage 2: ICS Attack Development and Execution. While this model comprehensively covers cyber activities of a targeted cyber-physical attack, the physical part of the attack is represented as a single stage called *Attack Development and Tuning*. In contrast, the cyber-physical attack lifecycle proposed in this chapter focuses on the physical part of the attack and details the steps needed for achieving an objective of the above mentioned stage.

In the same year, Hahn et al. [133] proposed their variation of Lockheed Martin’s Kill Chain model applied to the cyber-physical systems. Specifically, they added *Perturb Control* and *Physical Objective* stages to describe attack activities in the physical and control layers of the cyber-physical system. This work is the closest to ours. However, this work focuses on the formal descriptions of attack activities from the control theory standpoint whereas our framework is focused on the practical execution of an attack in real-world production environments.

In 2016 Larsen [191] described a process of developing a cyber-physical exploit chain with the help of Timing and State Diagrams (TSD). The sequence of the exploit building activities can be directly mapped to the stages of the cyber-physical attack lifecycle as shown in Figure 6.28. Larsen’s approach is complimentary to the work presented in this chapter as it allows to compare different attack design decisions in terms of development effort and execution reliability.

Finally, in 2019 MITRE Corporation published the ICS ATT&CK knowledge base of the tactics and techniques that adversaries can use when attacking an industrial control systems [69]. While ICS ATT&CK provides a useful summary of known ICS attack techniques, the focus of the framework is on the execution and detection of the attacks in the cyber layer. In contrast, the

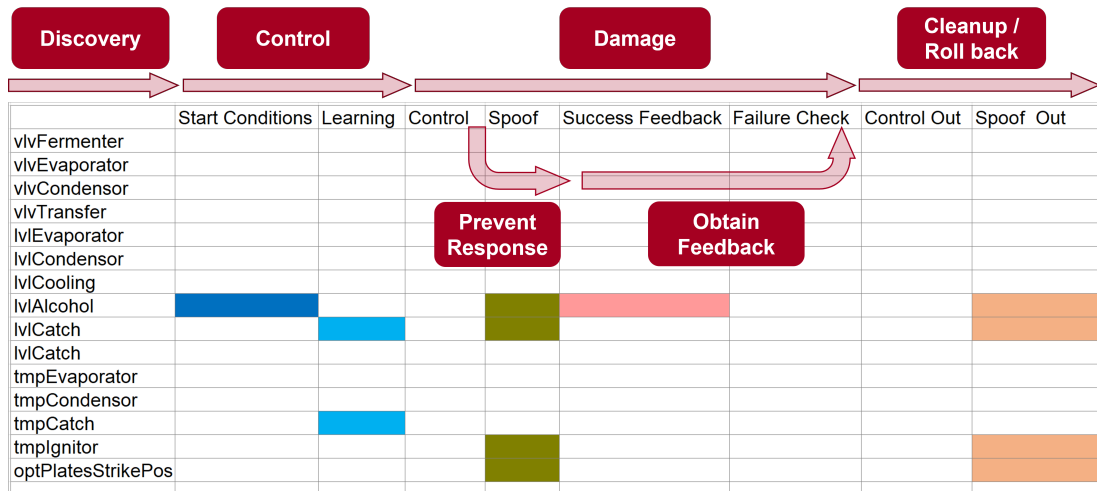


Figure 6.28: Relationship between TSD and cyber-physical attack lifecycle

framework proposed in this chapter puts greater focus on which attack tactics and techniques need to be implemented in the control and physical layers. The MITRE ICS ATT&CK was later expanded by Menendez [229] to better fit the attack lifecycle in the context of the electrical sector. However, the focus of the framework remained on the cyber aspects of ICS attacks.

The proposed stages of cyber-physical attacks were tested by Krotofil and A. D. [178] on the example of a traffic light system, illustrating its applicability to discrete cyber-physical systems. As an application of moderate complexity, traffic light systems have a relatively small and constant process state space which can be fully tested by the attacker in the Control stage. Once useful controls are identified and tested, the attacker can develop a portfolio of ready-to-use attack instances and apply them in the Damage stage at an opportune time. Exploits can be combined into a final payload depending on desired attack outcome. Examples of damage scenarios include large-scale gridlocks to cause economic damage or short-term traffic congestions to slow down the arrival of vital services such as police or ambulance.

6.8 Conclusions

Cyber-physical systems security concerns specifically attacks that cause physical impact. To achieve such an impact, the attacker has to find ways of manipulating the physical processes in the target environment. Cyber attacks on control networks may allow the attacker to obtain access to sensor measurements and manipulate instructions sent to actuators. However, to appreciate the effect of such manipulations the attacker has to understand the physical part of her target. Similar to attacks executed in the IT domain, a cyber-physical attack goes through several attack stages before achieving the desired damage. Along the cyber-physical attack lifecycle, some activities may resemble well-known IT exploitation methods and some will require expertise on the physical part of the cyber-physical system, expertise not commonly found in the IT security community. Depending on the attack scenario, state of the process and encountered challenges, the attacker may skip certain stages or may need to circle back to previous stages.

The cyber-physical attack lifecycle provides greater insights into the detours an attacker may have to take to reach her goal, the types of data sources to consult to achieve the required level of process comprehension and which vital reconnaissance tasks need to be executed on the live process. We illustrated the utility of the proposed framework in the example of a simulated model

of the Vinyl Acetate Monomer plant and demonstrated non-trivial hurdles to overcome at each attack stage. It took almost one and half years to finish this research undertaking from the initial familiarization with the VAC process to obtaining final results.

From the attacker's perspective understanding the desired state of a physical process that achieves certain attack goals, and knowing how to reach that state are two distinct problems. The process is not designed for the attacker. The attacker may be impeded by automatic safety interlocks and/or may not have access to observations that allow her to monitor the effect of her actions. Due to the significant effort required to execute sophisticated high-precision attacks such capabilities are likely to remain in the hands of state-sponsored threat actors with sufficient resources and reserved for critical targets. In contrast "easy button" attacks are more likely to be executed on a larger scale.

Successful execution of cyber-physical attacks relies on access to specific information sources and assets. Understanding attacker needs assists defenders with identifying "crown jewels" assets such as historians, specialized engineering applications, OPC servers and engineering workstations and strategizing defense and detection efforts.

Chapter 7

Conclusions and Future Research

The important thing is not to stop questioning.

Albert Einstein

Below we outline the contributions of our work and further work done on the basis of the thesis contributions. We then discuss future research directions.

7.1 Summary of Contributions

The overarching objective of this thesis was to improve discerning cyber-physical systems security from the control system and physical process standpoints as opposed to the security of the supporting IT infrastructure. Specifically, we focused on better understanding the threats and exploitation techniques as well as security requirements and defenses specific to cyber-physical systems. We aimed to keep our research fundamental in nature so that our results could apply to a wider range of cyber-physical systems. In the following, we summarize our contributions in relation to the initial research questions.

Research Question One *Can the available research models of physical processes be used to study the impact of cyber attacks on the physical processes and their supporting control system?*

Among the main challenges of cyber-physical security research is the requirement for testing environments which accurately represent the physical process and its control infrastructure and allow to experiment with failure conditions. In this thesis we explored the feasibility of utilizing process models used in process and control engineering for CPS research needs. Although the existing models on their own are of limited use, we showed that they can be adapted to facilitate a cyber-physical security analysis. We focused on two publicly available models of chemical plants: Tennessee Eastman and Vinyl Acetate Monomer. First, in Chapter 3 we described how the models were enhanced and instrumented with means to enable attacker interactions and allow qualitative and quantitative evaluation of the bespoke interactions. We then illustrated the practical utility of the models to cyber-physical research in Chapters 4, 5 and 6.

Research Question Two *Are there attack vectors and/or attack scenarios that are unique to cyber-physical systems? Are traditional IT security defenses adequate to thwart CPS-specific attack techniques?*

For valid historical reasons, industrial control systems were designed with safety in mind, focusing on their ability to withstand a variety of random faults. This means, that even if typical IT security controls are added, for instance, to the design of an industrial protocol stack, they may not guarantee defenses against an intelligent attacker who exploits vulnerabilities in the control layer of a cyber-physical system. In Chapter 4 we analyzed the mechanics of Denial-of-Service attacks and their impact on the control function of cyber-physical systems. We showed that DoS attacks are similar to data integrity attacks in their effect. However, their execution does not require packet manipulation and can be performed even on authenticated and integrity-protected traffic. We then showed how to strategically select attack value out of observed process values and how to chain several DoS attacks to achieve the desired attack outcome, not achievable by a single DoS attack on an individual control loop.

Research Question Three *Are there data security requirements specific to cyber-physical systems? Are traditional IT data security controls adequate to meet CPS-specific data security needs?*

Data quality plays a paramount role in the correct, efficient and safe functionality of a cyber-physical system. On one hand, it is critical to correctly configure all data processing points along a data flow to preserve data utility and the ability to extract correct informational insights from the process data. On the other hand, ensuring the correct or truthful acquisition of process data is the most fundamental security requirement for process data. The latter requirement is not discussed in the context of IT applications and as a result security approaches to guarantee data veracity are not part of the standard IT security toolkit. To investigate this research questions we first illustrated the importance of data veracity to CPS applications and provided examples of how data veracity can be compromised in Chapter 5. We then proposed a specific attack scenario to violate data veracity and proposed a plant-agnostic detection method based on entropy metrics in form of sensor signal consistency and plausibility checks.

Research Question Four *Is the cyber-physical attack lifecycle different from the traditional IT attack lifecycle or kill chain? What are the essential attack steps to achieve the desired impact in the physical world?*

An attacker targeting a remote process may not immediately be gifted with the required knowledge of the process and the means to manipulate it. From the attacker's perspective understanding the desired state of a physical process that achieves certain attack goals, and knowing how to reach that state are two distinct problems. Similar to sophisticated attacks executed in the IT domain, a cyber-physical attack goes through several attack stages before achieving the desired damage effect. The defense goal is to understand attacker opportunities in the context of the exploited environment and figure out effective measures to interrupt the attack kill chain. In Chapter 6 we described the structure of a cyber-physical attack with associated attack tactics, techniques and procedures, especially focusing on those which are different from the conventional TTPs known in the IT domain. In the example of an attack scenario aimed at causing persistent economic damage to a chemical plant, we provided a stepwise illustration of a cyber-physical attack and highlighted pivotal attainments required at each attack stage, disruption of which may significantly complicate the attacker's ability in achieving their attack objective or even interrupt the attack kill chain.

7.2 Follow-Up Research

The foundational results presented in this thesis provided inspiration for follow-up CPS security research related to the research topics addressed in this thesis.

Academic papers

- A Rising Tide: Design Exploits in Industrial Control Systems, M. Krotofil, A. Bolshev, J. Larsen, R. Wightman. 10th USENIX Workshop on Offensive Technologies, WOOT 2016, Austin, TX, USA, August 8-9, 2016

This paper formulated the concept of design exploits in industrial control systems, which remain unaddressed in public risk scenario approaches. The paper provided examples of design flaws in physical, control, and cyber layers of a CPS successful exploitation of which would result in a physical impact on the underlying physical process.

Contribution: Main author and co-researcher.

- Achieving ICS Resilience and Security Through Granular Data Flow Management, B. Green, M. Krotofil, D. Hutchison. Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy, CPS-SPC 2016, Vienna, Austria, October 28, 2016

This paper illustrated how a granular representation of data flows and their data processing points allows for a much more accurate security and risk assessment of industrial process control infrastructures.

Contribution: Co-author and co-researcher.

- Cyber-Physical System Discovery: Reverse Engineering Physical Processes, A. Winnicki, M. Krotofil, D. Gollmann. Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security, CPSS@AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2, 2017

An attacker facing an unknown process is challenged by the lack of knowledge about the tuning of its control loops and its responses. This work illustrated the feasibility of reverse engineering a controlled physical process from observations of responses to crafted impulses and cataloging the dynamic behavior of the process.

Contribution: Co-author, co-researcher, supervisor of Master's Thesis.

- On the Significance of Process Comprehension for Conducting Targeted ICS Attacks, B. Green, M. Krotofil, A. Abbasi. Proceedings of the 3rd Workshop on Cyber-Physical Systems Security and Privacy, CPS-SPC 2017, Dallas, TX, USA, November 3, 2017

Due to the lack of basic in-built security mechanisms across the majority of ICS protocols and equipment, they are often considered to be an easy target, even for low skilled threat actors. On the example of widely applied Man-in-the-Middle attack scenarios, this work illustrated the non-trivial reconnaissance efforts an attacker must invest before compiling the payload for targeted physical impact while avoiding detection.

Contribution: Co-author and co-researcher.

- PCaaD: Towards Automated Determination and Exploitation of Industrial Processes, B. Green, W. Knowles, M. Krotofil, D. Prince, N. Suri. Computers & Security, August 8, 2021

This work demonstrated the feasibility of designing a fully automated, stealthy, targeted attack payload for arbitrary industrial processes through the exploitation of current PLC programming practices and library of functions code reuse patterns.

Contribution: Co-author and co-researcher.

Conference talks (selected)

- Hack Like a Movie Star: Step-by-Step Guide to Crafting SCADA Payloads for Physical Attacks with Catastrophic Consequences, M. Krotofil, A. D. Zeronights, Moscow, Russia, November 25-26, 2015

In this talk we verified the applicability of our cyber-physical attack lifecycle on the control system of a crossroad traffic light. By following the attack stages we successfully created a portfolio of attack instances which could be used to achieve a desired damage scenario, e.g., cause gridlock or setup a green wave at an opportune time.

- Never Trust your Inputs: Causing “Catastrophic Physical Consequences” from the Sensor (or How to Fool ADC). A. Bolshev, M. Krotofil. Black Hat Asia 2016, Singapore, Republic of Singapore, March 30, 2016

The talk illustrated the need of hardware- and software-based input sanitization/validation in the cyber-physical systems on the example of compromised analog sensor signals and their impact on the outcome of analog-to-digital conversion.

- “Man-in-the-SCADA:” Anatomy of Data Integrity Attacks in Industrial Control Systems. M. Krotofil, C. Sistrunk. Black Hat Asia 2017, Singapore, Republic of Singapore, March 30, 2017

The talk accentuated the challenges of targeted data integrity attacks in industrial control systems and illustrates the requirement for control-system-specific reconnaissance activities and strong engineering knowledge for attack design. This talk also explained the challenges of maintaining data utility property in ICS environments.

- Evil Bubbles or How to Deliver Attack Payload via the Physics of the Process. M. Krotofil. Black Hat USA 2017, Las Vegas, USA, July 26, 2017

The talk introduced the concept of a physical process being a communication media for process equipment. Process physics can be used to deliver an attack payload even between equipment from different security zones.

- TRITON: How it Disrupted Safety Systems and Changed the Threat Landscape of Industrial Control Systems, Forever. M. Krotofil, A. Carcano, Y. Dragoni. Black Hat USA 2018, Las Vegas, USA, August 8, 2018

Following the TRITON incident, this talk presented a scenario for a physical damage attack and its implementation in form of a shellcode as a payload for TRITON malware. The execution of the payload was illustrated using a real-world Triconix safety controller.

- A Diet of Poisoned Fruit: Designing Implants and OT Payloads for ICS Embedded Devices. J. Wetzles, M. Krotofil. TROOPERS 2019, Heidelberg, Germany, March 19, 2019

This talk provided additional information about attacker activities at each stage of cyber-physical attack lifecycle and how they could be implemented in the embedded systems such as controllers or transmitters.

- If I were MITRE ATT&CK Developer: Challenges to Consider when Developing ICS ATT&CK – Attacker perspective. M. Krotofil. The 7th International Kaspersky Industrial Cybersecurity Conference 2019, Sochi, Russia, September 19, 2019

This presentation discussed the non-trivial challenges of capturing attack techniques from cyber, control and physical layers of a cyber-physical system in a single framework.

7.3 Future Research

Cyber-physical security is still an evolving field with many questions yet to be answered. In this section we summarize possible future research directions.

Chapter 3 presented two models of continuous processes to conduct cyber-physical experimentation. To gain a better understanding of the attack surface of cyber-physical systems from various industries requires more public models of realistic processes, e.g., discrete event-driven process models such as manufacturing or logistics processes. When applying the results from Chapter 6 to a model of a traffic light system (discrete process), we noted that exploitation of discrete systems can be less challenging than exploiting continuous processes due to a finite and deterministic space of possible process states [178]. Figure 7.1 shows an excerpt from a control logic where traffic light (TL) states are predefined by a fixed logic based on Boolean operators AND and OR. This suggests that discrete systems, especially with a relatively small state space, are more promising targets for high-precision attacks. Moreover, it is possible to create a portfolio of attack scenarios and exploits that can be applied to an arbitrary system of a similar kind with moderate efforts.

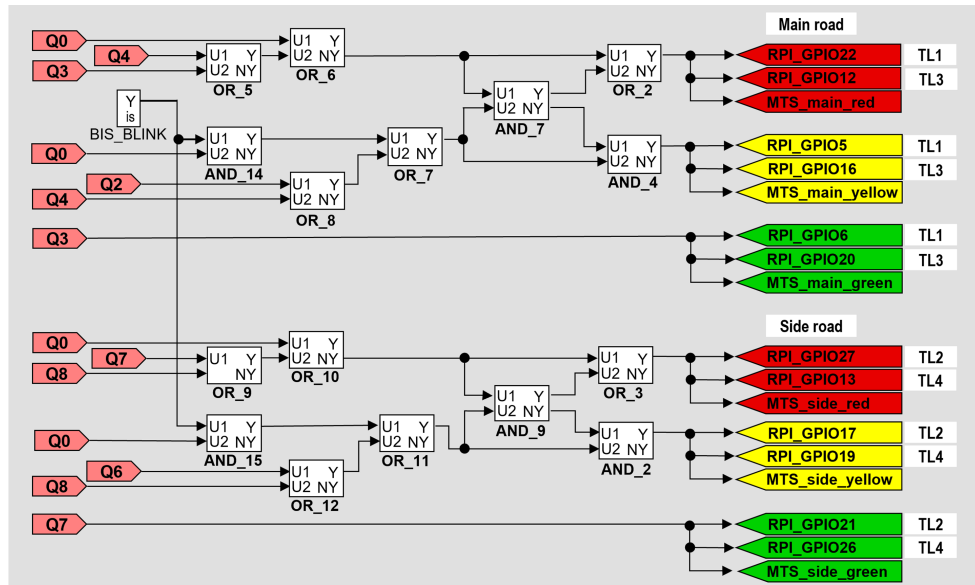


Figure 7.1: Snippet of a control logic for a traffic light system [178]

Cyber-physical security research on a diversity of models may reveal synergies in attacker efforts, such as learning curves and economy of scale when designing attack tools, and inform defenders about more likely attack scenarios and opportunities to interrupt the attack kill chain. Further, when prioritizing defense efforts it is important to have metrics to measure the severity of attack impact. Individual industrial domains and cyber-physical applications are subject to unique performance targets. Typically security risks and attack consequences are expressed as business impact, measured in financial numbers. However, such a measure is not granular enough to differentiate between the criticality of various attack scenarios. To provide an example, Urdaneta et al. [311] show the advantage of evaluating attack strategies on a traffic light system employing traffic performance measures such as travel time and queue length.

Chapter 4 presented a novel attack vector specific to control systems called Stale Data. We focused on determining the time to execute a DoS attack so that the desired physical effect would be achieved as soon as possible. Alternatively, the attacker can take advantage of the timing

parameters of the PID controller tuning and invalidate the controller configuration by strategically delaying sensor values, causing oscillations, instability, high variation, and other unwanted effects as illustrated by Larsen [190]. Further research is required into adversarial strategies for DoS attacks to learn about suitable preventive and reactive defense methods. Possible topics include the design of communication libraries that are resilient to different scenarios of Stale Data attack as well as DoS tolerant control algorithms and fault-tolerant controllers, able to support the execution of the security-aware control schemes.

Chapter 5 examined the data security property called veracity. This security requirement is not typically studied in information security as data content and context are not considered in the threat model. Further research is required into approaches for violating data veracity and their impact on physical applications. One such example are so called zero-dynamics attacks (ZDA) which target the internal dynamics of a system and remains stealthy to detectors that monitor system inputs and outputs [308]. Better awareness of relevant threats and security requirements is essential to formulating pertinent security policies and guiding the implementation of effective security controls in an organization [58]. Additionally, at the moment there are no open-source or commercial tools to assist with the discovery and visualization of the attack surface for both data veracity and utility. The development of approaches to capturing the inventory of data flows and detailing data process points is another research question to address. We call this concept the Data Bill of Materials (DBOM), echoing the concept of the Software Bill of Materials (SBOM) in software applications [297].

Chapter 6 proposed our cyber-physical attack lifecycle. Being a composite process, future research directions can be formulated for each of the attack stages. We will focus on the Damage stage. Among fundamental research directions is the need for theoretical and experimental studies of various damage scenarios. Since the initial taxonomy of physical damage approaches presented by Larsen in 2007 [188], there were only limited works on this topic. This is related to the requirement of specialized knowledge and the high costs of physical testbeds. Research into high-fidelity digital twins may enable progress in this line of research. Progress in TTPs specific to the Damage stage will in turn create the knowledge needed for devising patterns or indicators of ongoing physical attacks, possibly with AI methods as shown in Figure 7.2.

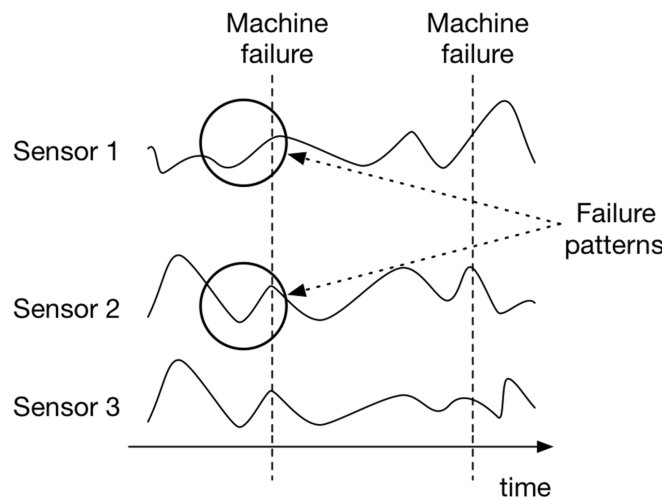


Figure 7.2: Construction of equipment failure indicators [251]

In Appendix E.3 we show how process physics can be used as an attacker instrument and carry attack payloads, and argue that ICS security standards should expand their scope to the physical processes layer. However, despite the importance of detecting ongoing cyber-physical attacks

at the level of process physics (last line of defense), such attacks should be ideally prevented or detected in the upper layers of the Purdue reference architecture due to the significant cost of their consequences and hazardousness potential.

7.4 Conclusions

Cyber-physical systems are engineered systems that are built from and depend upon the seamless integration of computational and physical components. While advances in computing technologies gave industry new opportunities and functionalities for interacting with the physical world, they have also changed the attack surface and introduced new forms of attacks. In contrast to IT security which can trace its roots back several decades, the discipline of CPS security is comparatively young, and the research community has not yet achieved the same clarity about relevant threat scenarios and best security practices. While compromising computing infrastructure and communication channels is a powerful facilitator for launching attacks aimed at disrupting physical processes, the damage from an attack will be limited if the attacker is unable to manipulate the control system in a way needed to achieve the desired physical outcome.

The attacker is not bound to adhere to assumptions made by the defender about possible adversarial behavior. In fact, it is an effective attack strategy to search for assumptions that can be violated, including previously unconsidered attack scenarios. We have shown that the security concepts from the IT domain are not sufficient to describe security demands on cyber-physical systems nor are the defenses effective against all attacks which are part of the cyber-physical kill chain. We, therefore, propose to treat the cyber part of a cyber-physical system as a control system instead and focus on the interfaces between physical space and cyberspace. For the attacker, the control system is a weird machine [34] commandeered by unconventional inputs used in ways not intended by the designer. To constrain the attacker, systems ultimately have to be designed and operated in a way so that as little as possible can be done with them beyond their intended use. This may include replacing critical digital components with immutable alternatives. The goal of this thesis was to show that defense efforts should extend to the design of process control applications and process physics itself as visualized in Figure 7.3

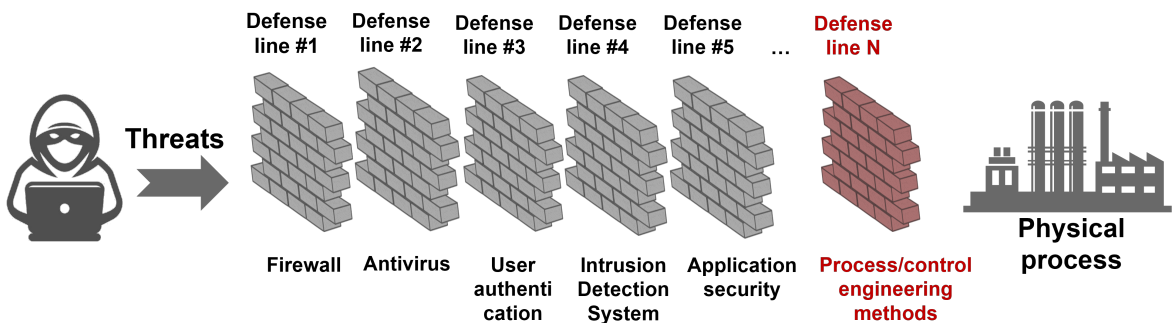


Figure 7.3: Visualization of defense-in-depth concept in cyber-physical systems

We conclude with the encouragement to further develop process/physics-aware cyber security mechanisms that do not solely rely on IT security approaches from the digital space. This, for instance, includes redesigning cyber-physical systems to make them more robust to the compromised physical environment (both in immediate proximity and at a distance). CPS should be further enhanced with an enlarged forensic footprint to enable improved situational awareness in the presence of attacks. Once physical and/or IT barriers are compromised, adversaries are potentially

able to take control of the physical process. While the latter is a non-trivial task and requires expert knowledge, it would be negligent to assume that attackers are unable or not incentivized enough to obtain and apply the required knowledge. Technical means should be complemented with the appropriate international regulations where the use of cyber-physical attacks against civilian critical infrastructures should be prohibited to preclude potential humanitarian crises.

Appendix A

Attack on Situational Awareness

Industrial process dynamics are monitored around the clock by operators via an HMI console. Upon observing an undesired process behavior, an operator takes corrective measures to bring the process back into its steady state. Moreover, if the operator attributes the disturbances to causes, she can initiate an immediate incident investigation. Out of these considerations, the attacker might prefer to hide the real field data from the operator for the duration of the attack (see Section 6.5.5). This type of attack was realized in Stuxnet [187].

Let the attacker's goal be to raise the pressure in a reactor to an unsafe limit without the operator's awareness. One of the possible scenarios to "blind" the operator is to record steady state process data and replay them back during the attack. Spoofing process data impedes process observability and may result in loss of *situational awareness* by the operator. This is one of the most dangerous attacks on process control. If the attacker manages as well to manipulate the safety limit value or suppress the safety system's communication link, the reactor can actually explode and injure personnel in its vicinity [272].

A.1 Experimental Setup

To model and detect this type of attacks, we have implemented an experimental environment in a form of a hybrid process control environment (real hardware, simulated process) as depicted in Fig. A.1. For the industrial hardware we used the Siemens SIMATIC Starter Kit, consisting of an S7-1200 PLC and the KTP400 HMI Starter Kit. We opted for Modbus/TCP as the industrial communication protocol due to its open-source nature. The role of the PLC is to poll selected process variables to display them in the HMI and to forward the setpoints to the process. The Modbus protocol utilizes a Client-Server communication model. Therefore it is required to install Modbus Server and Client on the PLC. The configured Modbus client Function Block is shown in Figure A.2. To enable communication between the simulated process and the HMI/PLC, we used the open-source library libmodbus [292]. The detection logic was implemented as a separate software module.

A.2 Experimental Results

We implemented an attack on the observability of the pressure process variable via manipulation of the PLC code. During the initial stage of the attack, the PLC records process measurements during normal plant operations. When the attack begins, the PLC sends stored data to the HMI whereas the real field data remains undisclosed. To detect that spoofed data are being sent to the HMI, we implemented an experimental IDS engine. We monitor data flows between the process and the PLC and between the PLC and the HMI. Any discrepancy in the process value between the indicated data flows will indicate an *attack on data consistency*. To watch over the specified data flows on one hand we query the output registers of the PLC for the data which should be displayed on the HMI. On the other hand, we capture the traffic between the process and the PLC. If an inconsistency in the data beyond a certain threshold is detected as shown in Figure A.3, an alarm is generated by the Alarm Manager on the IDS.

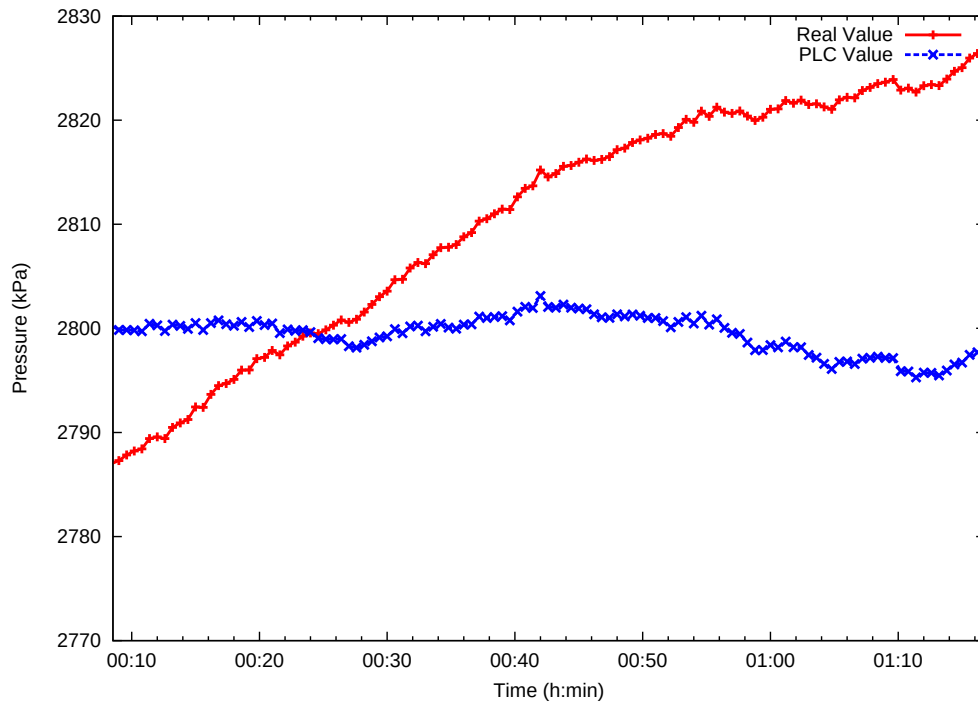


Figure A.3: Data inconsistency detection

The described attack scenario and detection approach are relatively straightforward. However, our main objective was to verify the hypothesis that the usage of real physical hardware for control and monitoring does not always add value when studying the behavior of the physical process under a cyber attack. Depending on the research questions, the usage of process models with the integrated implementation of control algorithms provides enough fidelity. This is why, for the purpose of our research, we decided in favor of a software-based testbed.

Appendix B

ICS Security Properties

The CIA triad was conceived for communication systems (ISO 7498-2:1989) in the 1980s, i.e., for a *digital infrastructure*. ICS are engineered physical systems. CIA in the original sense certainly applies to the supporting IT infrastructure, i.e., to protect data in transit and storage. An ICS is, however, more than its IT infrastructure. It is a control system, *nomen est omen*. Even if one would try to adapt the CIA triad to control systems and find satisfactory explanations, terms would be overloaded with different interpretations in different parts of a system. For example, should an up-and-running but uncontrollable process be termed ‘available’?

The alternative is to approach control system security from the first principles. To do so, one needs a *threat model*, just as it is the case in the IT domain. The threat model has to capture the fundamental possibilities of the attacker. What can she do to a control system?

B.1 Controllability, Observability and Operability

Control is the adjustment of available degrees of freedom (manipulated variables) to assist in achieving an acceptable operation of the process. In order to control a process, one needs to measure (observe) it first. If the state of the process cannot be determined due to incorrect or incomplete process measurements, one is effectively controlling a fiction, i.e., an arbitrary process in an arbitrary state, but not the actual process. This applies both to the attacker and the defender. To ensure ‘secure control’ one thus needs process-oriented security metrics and properties.

An attacker must be able to *observe* the process to be able to derive its *state*. To do “whatever she wants with a given system”, she must make sure that she controls the system. Her goal is to acquire controls that are sufficient for bringing the process into a state she desires, and at the same time to take control away from the control system installed and from the operator. To illustrate these concepts, we refer to a security incident at Kemuri Water Company (KWC) reported in 2016 [313], where hackers had manipulated the valves controlling the flow of chemicals, though to no particular effect.

In at least two instances, they [attackers] managed to manipulate the system to alter the amount of chemicals that went into the water supply and thus handicap water treatment and production capabilities so that the recovery time to replenish water supplies increased. Fortunately, based on alert functionality, KWC was able to quickly identify and reverse the chemical and flow changes, largely minimizing the impact on customers [313].

While the attackers were potentially in a position “to do whatever they wanted”, they did not obtain observability of the system to measure the effect of their attack. Also, the attackers did

Controllability	Observability	Operability
Ability to bring the process into a desired state • Feasibility – The process in a controllable state (there is a control sequence which can bring process into an intended state) • Awareness – The sequence of the control commands known to the operator	Ability to determine process state and maintain situational awareness • Data quality and availability – Data trustworthiness (veracity) – Integrity and availability of data in transit and storage • Sufficiency – Measuring all necessarily quantities at the right locations – Ability to interpret the measurements	Ability of the plant to achieve acceptable operations • Resilience – Ability to maintain optimal operations under attack • Survivability – Ability to maintain operations albeit at suboptimal level • Graceful degradation – Ability to maintain limited plant functionality to achieve safe shut down

Figure B.1: Security properties for control systems

not obtain full controllability of their target and could not prevent human intervention which thwarted the attack execution

Observability and controllability are thus crucial for control systems and are therefore core ICS security objectives. These terms were formally defined for linear dynamic systems by Kalman in the 1960s. They can be informally defined as follows.

- *Controllability*: A system is *controllable* if and only if for any two (valid) states A and B , there exists an input and a time t , so that the input takes the system from initial state A to state B by time t .
- *Observability*: A system is *observable* if and only if the value of its initial state can be determined from the system output observed up to some given time t .

It is particularly desirable that a system always remains controllable at least to a degree where it can be brought to a *safe state*.

The third relevant property is *operability*. Operability stands for the ability of a plant to achieve acceptable operations [281]. What is acceptable depends on the situation and includes the ability of a plant to perform satisfactorily under conditions different from its nominal design conditions. E.g., during a process upset or an attack, a plant may operate inefficiently from a cost standpoint, but operability would be acceptable for the given situation.

Operability can be further refined into resilience, survivability, and graceful degradation. For a process to be resilient, the physical design and the control strategies have to be designed so that they can withstand malicious process manipulations. *Resilience* is thus the ability to absorb process disturbance caused by attacks. *Survivability* is the ability to maintain plant operations albeit at a suboptimal level, e.g., with higher production costs or reduced product output, but the plant continues delivering its services. *Graceful degradation* means that a system is capable of maintaining limited but safe functionality and can be safely shut down.

The three security properties for process control then abbreviate to COO or CO₂ (Figure B.1). Note that the effects of CIA type of attacks on ICS are popularly described as “loss of view” and “loss of control”, which may exaggerate their effect. Even if the operator is blinded due to a modified view at the HMI, a process may still be observable through data in the historian or any database that stores process data, or through the input data in controller memory and physical gauges. Similarly, if the attacker manages to compromise one or several sensor readings, the system may

still remain observable through proxy sensors, i.e., related or correlated sensors, or through derived process variables. From the attacker's angle, in order to make the operator lose view, she has to determine an attack strategy that ensures loss of observability by the operator and the control system, i.e., their inability to determine the true state of the process.

B.2 Ensuring ICS Security

Ensuring observability is related to sensor placement methods and derived measurement algorithms from the process and control theory [281]. Currently, processes are not designed to support observability for systems under attack, and no approaches are used to determine spoofed process views. Process data may be false at source (manipulated process, faulty or manipulated sensor) or become so in transit or storage (tampered data). The first case refers to data veracity and is traditionally discussed in terms of *data quality*. The second case refers to data integrity and is mostly discussed in terms of *data security*. With the extension of the attacker model to physical processes and sensors, veracity becomes a critical security property for process data. Research questions related to observability as a security objective therefore include:

- Do we measure the right process variables at the right places to be able to detect possible cyber-physical attacks? An attack could be observed by measured or derived process variables.
- Ability to interpret the measurements by the operator and identify a possible ongoing attack, type of attack, its possible root cause, etc.
- Ensuring veracity of process data and detection of spoofed sensor readings to ensure trustworthy observation of the process state.

When considering the latter security question, ICS specific security controls can take the form of data consistency or plausibility checks directly through correlated sensors, or indirectly through the proxy and related sensors or additional calculations, see, e.g., [89, 181]. Section E.1 gives a further example.

Similar considerations apply to controllability. An attacker managing to break a valve will not necessarily cause a loss of control for the operator or the control system; processes are built to be robust and recoverable. Control algorithms or the operator will adjust the process control to account for a lost degree of freedom (valve). From the offensive security standpoint, on one hand, the attacker needs to:

1. Identify controls that would allow her to bring the process into a desired state. Can the process even be moved into a state wanted by the attacker?
2. Ensure that both control system and operator lose controllability of the process, i.e., prevent their access to controls that would allow them to intervene in the attack or to bring the process into a normal state. Alternatively, an attacker can design an attack that cannot be mitigated with the available controls.

From the defensive security standpoint (process design), ensuring controllability implies, e.g., adding additional degrees of freedom (actuators) and overriding controllers to preserve controllability during process upsets caused by an attack. For instance, Barreto et al. [23] formulated a research question about defending the most critical actuators. They first identified inputs that

could be compromised by the attacker in order to manipulate more process states with less effort and ranked them accordingly. The defender should distribute their defense budget based on the ranking of each actuator. Alcaraz and Lopez [9] studied control capacity and process restoration possibilities from the *structural controllability point of view*. Structural controllability is an evolution of traditional control theory that focuses on providing the concept of “controllability” through graph-based structures. Additionally, the authors suggest a solution for the reliable propagation of alarms when a situation threatening structural controllability is detected.

Appendix C

Data Utility in Process Control Systems

Utility stands for data usefulness. Reduced data utility may result in the loss of vital or critical information resulting in significant consequences. The utility of process data can be impaired due to different causes. Consider a use case of obligatory regulatory monitoring of certain process parameters for conformity to certain operational requirements. For instance, in the UK regulators such as the Department for Environment, Food and Rural Affairs (DEFRA) require data that identifies wastewater treatment facilities' conformity to predetermined operational restrictions such as quality and quantity of final effluent [81]. Data must be collected from the appropriate sources within the operational process, conforming to any defined time constraints. While the regulatory instructions impose specific operational requirements to the process data, the same data may also be used for performance analysis, and by decision makers at the board level, to determine the financial viability of operational sites, and for identification of required investments to streamline processes. Furthermore, the same data may also be applied to the remote alarm monitoring of unmanned operational processes. In some cases, this remote alarm management may form part of an organizational safety reporting process. It is evident that without complete knowledge of all stakeholders of the individual data sources and understanding of their needs on data content, there is a high risk of losing or misrepresenting critical process information due to erroneous configuration of data processing points.

Figure C.1 presents the flow of data from a single sensor to the remote access connectivity residing within the corporate level of the OT reference architecture. When an instrumentation engineer recalibrates or replaces a sensor, she will likely know to liaise with a PLC configuration engineer to account for the new properties of sensor signals within the PLC logic, therefore maintaining accurate control of the physical process. This fulfills the primary requirement of sensors, that of providing accurate input to operational decision making applications such as control logic. However, when re-configuration requirements arise in relation to neighboring devices such as in-line monitors or data historians if their configuration is left unadjusted, regulatory, alarm monitoring, safety, and performance analytics data may become compromised. In the given example, awareness of sensor data consumption beyond the PLC could fall outside the scope of an instrumentation engineer's role. As a result, while the data processing formulas in the PLC's control logic were modified to maintain the accuracy of calculations, the other data processing points across the data path were not, leading to the corruption of calculations and stored values in the historian server.

When a situation like this occurs, from the initial suspicions around data quality, historian support engineers would drill down into complex mathematical calculations to identify the root cause of spurious data. It is not infrequent that the calculations are derived from up to 30+ operational tags (signal inputs), therefore the investigation process could prove to be time-consuming.

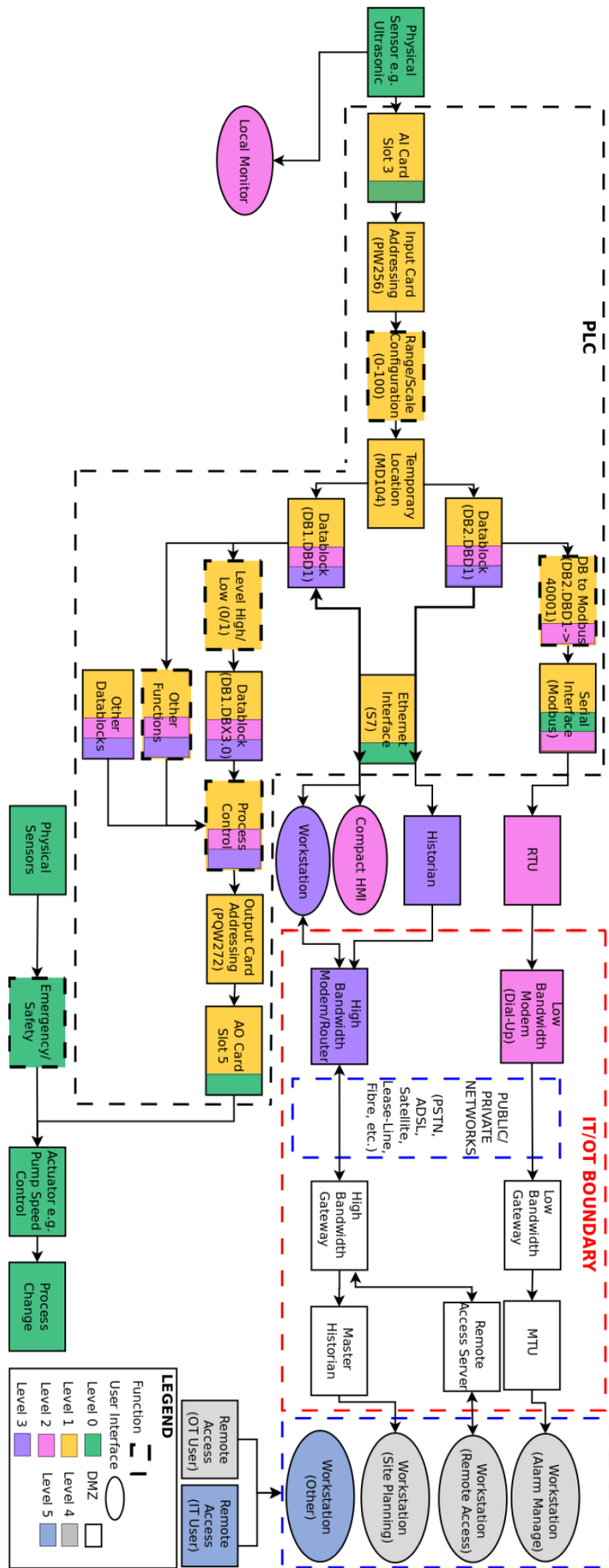


Figure C.1: Data flow of a single sensor through the levels of ICS reference architecture [123]

Furthermore, if data is processed at multiple points in the system, the visibility into data processing points may end quickly, requiring interaction with Level 1 control engineers to better understand any changes further downstream, across suspected data flows. It is evident that preserving data utility from the source and along its path is critical for ensuring business continuity and reducing troubleshooting efforts.

Examples of Data Utility Impact

In his public blogpost, Weiss [318] described a real-world use-case in which two *identical* plants reported to the plant vendor that one plant experienced a flow-induced vibration issue (Figure C.2a) and was operated at a reduced load, while the other plant did not have a vibration problem and was operated at full power. The latter plant ultimately reported significant damage to the nuclear fuel system caused by vibration. It was discovered during the investigation that both plants had flow-induced vibration issues. The indicator of this physical problem was a resonant frequency detectable by the plants' in-core instrumentation recorders. However, one chart recorder (Figure C.2b) reported the problem while the chart recorder in the other plant had the sensor signal filter altered to eliminate the higher frequency noise in the signal that was indicative of the flow-induced vibration. This resulted in the "filtered" plant operating in unsafe conditions, without the operators' awareness.

In this example, while the sensor signal was still veracious and authentic, and its filtering was done by the authorized personnel, the filtered content of the sensor signal did no longer provide one of its intended functions, i.e., vibration monitoring. Erroneous changes of sensor signal processing parameters are frequent causes of unintentional data utility corruption. Data utility differs from data integrity in the sense that the data itself is not manipulated but the way of extracting information from the data is altered.

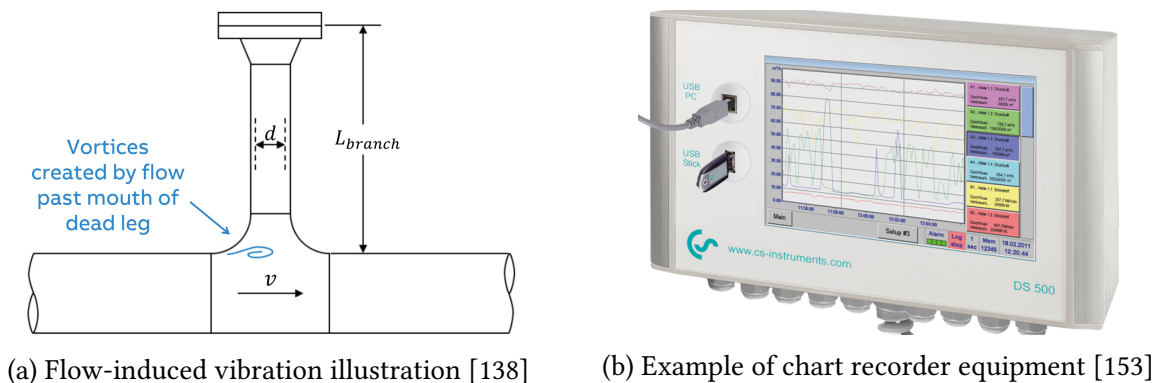


Figure C.2: Flow-induced vibration in piping systems and its detection

In another example [95] a control engineer in a pumping station responsible for PLC programming used the 0 to 4095 analog-to-digital converter counts range to convert the 4-20 mA analog signal from the measuring instrument whereas the engineer responsible for the telemetry used the 819 to 4095 range on the RTU. The 819 counts offset was introduced to detect signal underrange as shown in Figure C.3. As a result, when the PLC received zero pressure (4 mA), it converted it to 0 counts whereas the RTU program output 819 counts for zero pressure. The SCADA read process data from the RTU whereas the HMI at the site received data from the PLC. Consequently, the mismatch in data scaling leads to inconsistencies in data readings across

different applications. It is apparent that such inconsistencies in converting measured data into digital counts and subsequently into engineering units may result in undesired consequences.

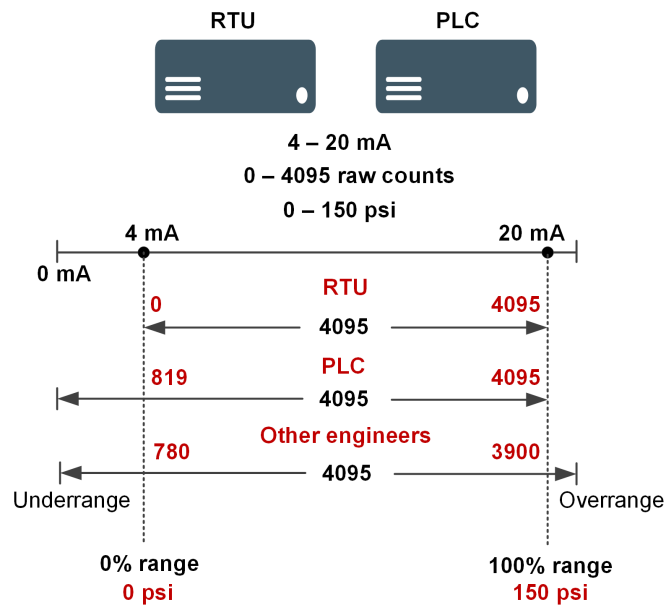


Figure C.3: Inconsistency of control equipment configuration

In process automation, field devices such as sensors are considered to be monolithic units that are fully trusted [237] and the data originated by the devices are correspondingly considered trustworthy. To provide useful measurement units, an analog signal must be scaled first. The accuracy of the scaling is validated through *calibration*. Improperly calibrated measuring instruments can be a source of safety accidents. An example is an explosion at BP Texas City Refinery (15 people killed, 180 injured) [270]. The root cause of that tragedy was critical alarms and control instrumentation providing false indications. Due to calibration errors the splitter tower level indicator showed that the tower level was declining when it was actually overfilling with flammable liquid hydrocarbons. As a result, the operator kept filling the tower. A chain of further events eventually led to an explosion. Hence, manipulation of instrument calibration is a potentially dangerous attack vector.

Calibration systems are increasingly transforming from stand-alone systems or work processes to software-based integrated solutions [8]. Although the automation of the calibration processes and tight integration into the documentation and maintenance management systems was driven by the desire to reduce human error, the integration also introduced an easy-to-exploit and dangerous attack vector. Quoting a software-induced incident revealed by safety systems vendor HIMA [176]:

Due to an unknown bug in the engineering software, all scaling of the SIS AI (analog input) got altered from 0 to 100% automatically. The altered value got loaded and activated automatically based on in unknown bug at the same system.

In the past, sensors used to be purely analog and the measured analog signal was converted into a digital signal by the controller. These days it is more common for a microcontroller to be embedded into the sensor itself [227]. The presence of a digital stage in the sensor allows attack scenarios where a sensor behaves maliciously but always passes calibration tests.

Appendix D

Sensor Signal Spoofing Algorithms

D.1 Runs Algorithm

Algorithm 1 Runs Analysis

```
1: procedure EXPLORE ▷ 1: analyse phase
2:    $signal \leftarrow$  signal to analyse

3:   while not an end of signal do
4:     while moving up do
5:        $runs ++$  ▷ count positives moves
6:        $value = sum(changes)$  ▷ positive steps change
7:       if direction change then ▷ save results
8:          $positivesruns(runs) ++$ 
9:          $positivesvalues(runs) = value$ 

10:    while moving down do
11:       $runs ++$  ▷ count negatives moves
12:       $value = sum(changes)$  ▷ negative steps change
13:      if direction change then ▷ save results
14:         $negativesruns(runs) ++$ 
15:         $negativesvalues(runs) = value$ 

16:    if no change then
17:       $nils ++$  ▷ count nils

18:   return  $runs, values, nils$ 

19: procedure PLAYBACK ▷ 2: playback phase
20:    $runs \leftarrow$  runs from previos step
21:    $values \leftarrow$  values from previos step
22:    $state \leftarrow$  current value
23:    $target \leftarrow$  target value ▷ can be a single value or a vector
```

Algorithm 1 Runs Analysis - Continued

```
24:   allpositives = sum(positivesruns)
25:   allnegatives = sum(negativesruns)
26:   for needed number of samples do
27:     if state < target then                                     ▷ current under target value
28:       rand = random(allpositives)
29:       for all sort(negativesvalues, desc) do
30:         move = move + current
31:         if move > random then
32:           i = indexof(current)
33:           move = positivvalues(i)/positivesruns(i)
34:           break
35:     else                                                         ▷ current above target value
36:       rand = random(allnegatives)
37:       for all sort(negativevalues, desc) do
38:         move = move + current
39:         if move > random then
40:           i = indexof(current)
41:           move = negativesvalues(i)/negativesruns(i)
42:           break
43:       state = state + move
44:       signal(end + +) = state
45:   return noised signal
```

D.2 Triangle algorithm

Algorithm 2 Triangles

```

1: procedure EXPLORE ▷ 1: analyse phase
2:    $signal \leftarrow$  signal to analyse
3:    $window \leftarrow$  learning window
4:    $noiselvl \leftarrow$  noise parameter

5:    $step = window * 10$ 
6:    $topslope = -999.99$ 
7:    $bottomslope = 999.99$ 
8:   while not an end of signal do
9:     if first elements then
10:       $current = value$ 
11:       $index = 1$ 
12:      while  $index < window$  do ▷ learning phase of  $i - th$  bucket
13:         $upperslope = (current - (last + noiselvl)) / index$ 
14:         $lowerslope = (current - (last - noiselvl)) / index$ 
15:        if  $upperslope > topslope$  then
16:           $topslope = upperslope$ 
17:        if  $lowerslope < bottomslope$  then
18:           $bottomslope = lowerslope$ 
19:        if  $index = window$  then ▷ end of learnig phase for  $i - th$  bucket
20:           $project = current$ 
21:           $slope = ((bottomslope - topslope) / 2) + topslope$ 
22:           $continue$ 
23:        while  $index < window$  do ▷ projecting phase of  $i - th$  bucket
24:           $project = project + slope$ 
25:          if  $current > (project + noiselvl)$  then
26:             $residual = current - (project + noiselvl)$ 
27:          if  $current < (project - noiselvl)$  then
28:             $residual = val - (project - noiselevel)$ 
29:          if  $residual <> 0$  and  $noiserun == 0$  then
30:             $noiserun = index$ 
31:           $drift = drift + residual$ 
32:          if  $abs(drift) > (2 * noiselvl)$  then
33:            go to clean
34:        if  $index > step$  then ▷ end of projecting phase for  $i - th$  bucket
35:           $project = current$ 
36:           $slope = ((bottomslope - topslope) / 2) + topslope$ 
37:          go to clean

```

Algorithm 2 Triangles - Continued

```

38: clean:                                                                                                     ▷ clean up
39:     index = 0                                                                                             ▷ go to the next bucket
40:     topslope = -999.99
41:     bottomslope = 999.99
42:     drift = 0
43:     if noiserun == 0 then                                                                                   ▷ missed the noise
44:         noiselvl* = 0.9
45:     if noiserun < (step/8) then                                                                           ▷ too far in the noise
46:         noiselvl* = 1.1;
47:     noiserun = 0
48: output:                                                                                                   ▷ make step output
49:     slopes(end + 1) = current - slopes(end)
50:     steps(end + 1) = index - steps(end)
51:     return slopes, steps

52: procedure PLAYBACK                                                                                       ▷ 2: playback phase
53:     slopes ← slopes from previous step
54:     steps ← steps from previous step
55:     state ← current value
56:     target ← target value                                                                                   ▷ can be a single value or a vector

57:     for needed number of samples do
58:         positiveslopes = slopes( where slopes > 0)
59:         positivesteps = steps( where slopes > 0)
60:         negativeslopes = slopes( where slopes < 0)
61:         negativesteps = steps( where slopes < 0)
62:         for needed number of samples do
63:             if state < target then                                                                                   ▷ current under target value
64:                 slope = random(positiveslopes)
65:                 currentsteps = positivesteps(slope)                                                                 ▷ corresponding steps
66:             else                                                                                                   ▷ current above target value
67:                 slope = random(negativeslopes)
68:                 currentsteps = negativesteps(slope)                                                                 ▷ corresponding steps
                                                                                                       ▷ divide current bucket into equally parts
69:                 minibuckets = [state, state + slope]/currentsteps
70:                 result = result concat minibuckets
71:                 state = result(end)
72:             if length(result) > needed then
73:                 result = result(1 - needed number of samples)
74:     return trianled signal

```

Appendix E

Communication Channels Beyond IT

In an IT system, components are designed to interact with each other only via information channels that are part of the IT system, the notable exception being the human user. In IT security research it is well-known that this is an abstraction that can be violated by using *covert channels* [186]. Covert channels use measurable effects such as memory usage or response time to convey information from one process to another, thus bypassing the process isolation implemented in the IT system. Covert channels had been a concern in military IT systems already in the 1970s. *Side channels* based, e.g., on execution time or power consumption can leak information about the internal state of a system. Timing channels have played a part in the recent Meltdown [205] and Spectre [169] attacks on modern CPUs.

In a process control system, the process itself can become both a communication channel and an attack vector. *Transduction attacks* exploit the physics of a sensor to manipulate its output or induce intentional errors. For example, acoustic interference can use the resonant frequency of a sensor to create fake output [108].

With access to a large number of power consumers, electric heaters or vehicles, or laser printers, for example, an attacker can bring and keep down an electricity grid without touching any of the grid's information systems. The attacker modulates the power consumption, of these devices over the internet at a rate too fast for the grid's protection mechanisms to react, pushing the grid into unstable states that trigger automated load shedding or tie-line tripping [74].

E.1 Zones and Conduits

The ISA 99/IEC 62443 set of standards [64] introduces the concept of zones and conduits as a way to segment and isolate the various sub-systems in a control system. A zone defines a grouping of logical or physical assets that share common security requirements based on factors such as criticality and consequence. Equipment in a zone has a security level capability. If the capability level is not equal to or higher than the required level, then extra security measures, such as implementing additional technology or policies, must be taken. Any communication between zones must be via a defined conduit. A *conduit* is a path for the flow of information between two zones. It provides security functions that allow different zones to communicate without introducing security threats to each other.

Consider now a piping infrastructure at an industrial facility. Once two devices are inserted into a process, they can become related to each other by the physics of that process. The physical process then becomes a communication medium and may be used for delivering malicious payloads even if the devices are segregated electronically. With that, the IEC 62443 approach to defining zones and conduits is violated when an attacker can use equipment in a lower security zone to

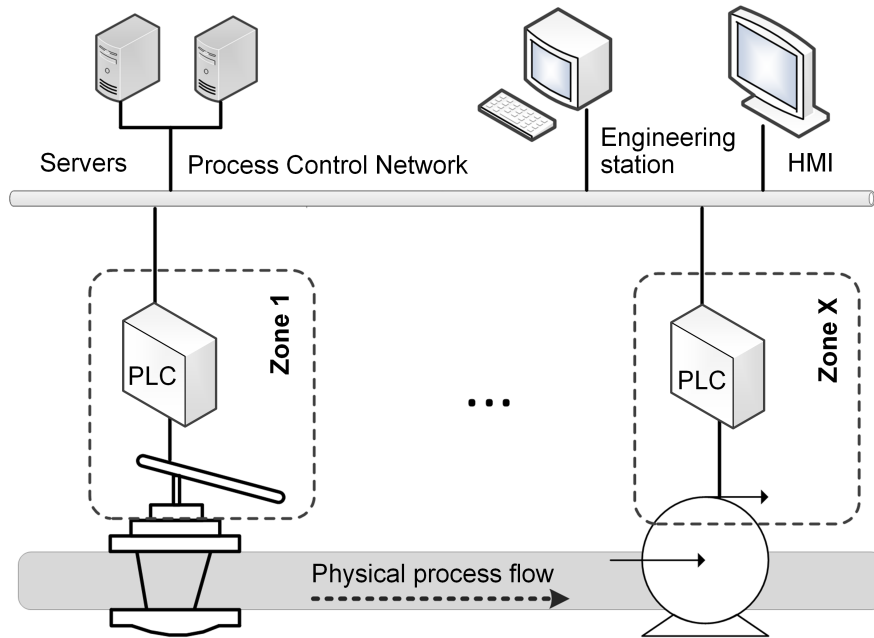


Figure E.1: Delivery of attack payload via process physics and violation of the network security zoning architecture

deliver an attack payload to piece of equipment in a more secure zone via process physics, rather than in electronic form via a configured conduit (Figure E.1). The security zoning analysis must thus be extended to an analysis of the physical process and its component interactions. Although a physical process is a conduit that cannot be eliminated, it can be monitored and attack propagation can be detected and mitigated.

E.2 Delivering Attack Payloads via Process Physics

Results from a physical experiment were presented at Black Hat 2017 using a setup where a valve and a pump did not have any electronic communication between them [177]. The goal of the attack was to damage the pump. The pump was analog and did not include exploitable IT components. Instead, the attacker exploited the positioner of an electronic valve to manipulate the valve position to cause cavitation in the pump. The damaging payload took the form of cavitation bubbles which were successfully delivered to the pump via process physics (fluid in the pipe).

Although the cavitation process harms the pump, breaking a pump in this way takes some time. To prevent operator response, the attacker may want to implement routines to spoof the reading of the flow sensor and the status of the valve positioner to make the process state appear normal. Detecting an ongoing cavitation attack is related to the research question on ensuring observability outlined in Appendix B.

In this experiment, even though the attacker was able to cause a loss of view with respect to the operator, she could not achieve the loss of process observability. The pump was instrumented with a few additional analog sensors (i.e., vibration, temperature) for pump diagnostic (Figure E.2). By manually collecting pump diagnostic data and using pump documentation, the operator could establish that the process state was not consistent with the pump state. Furthermore, using pump documentation and correlating pump measurements, it was possible to derive the true flow meter reading and the state of the valve positioner. Since analog sensors were only accessible locally, their

measurements could be trusted. In geographically distributed control systems, it was previously suggested to protect the security of a few known physical measurements with cryptographic means and use these measurements for detecting falsified states of the system [228].

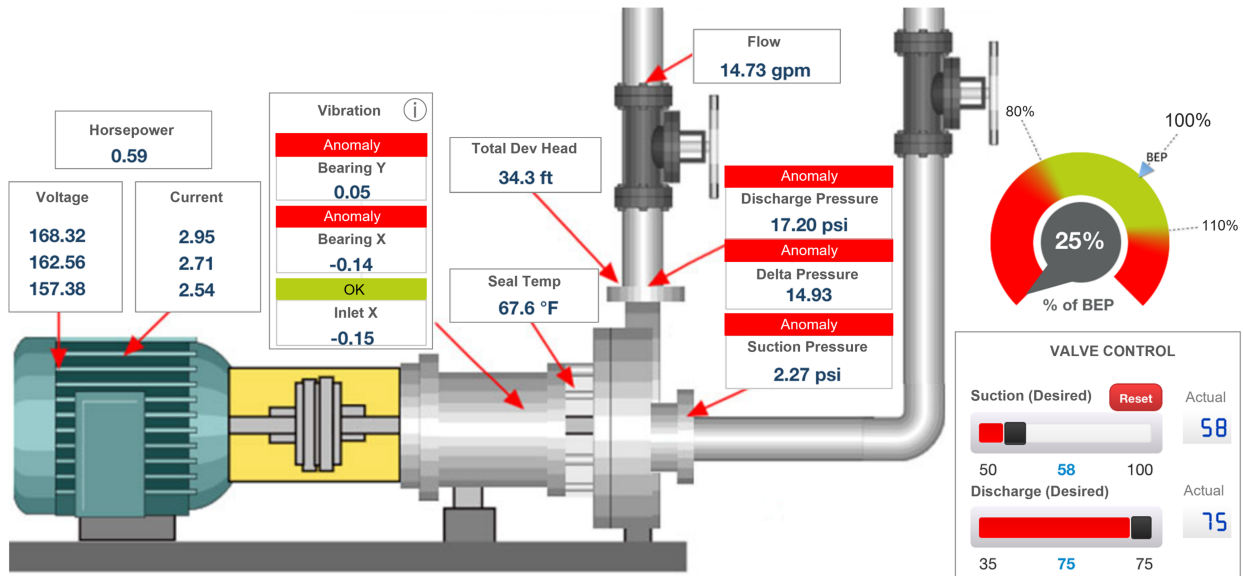


Figure E.2: Pump monitoring application by Flowserve [177]

Also, aggregated diagnostic data could be sent to the cloud via independent one-directional communication links and become accessible for the analysis in real time [38]. It was previously argued that IIoT technology introduces additional attack surfaces to OT. While this is true, IIoT can also become part of the solution when addressing OT security.

E.3 Information Flow via Process Physics

Moreover, a physical process may constitute an *information flow* channel. Such effects have been demonstrated in a Matlab model of a process for producing vinyl acetate [121], where the attacker attempted to deactivate a catalyst by raising the temperature in the reactor in a plant, without being detected. According to the process design, the reactor outflow passed through a heat exchanger to cool down before being further fed into a piping infrastructure. Due to the increased temperature in the reactor, reactor outflow was unusually hot, yielding a condensation problem in the heat exchanger and causing a “condensation” alarm.

To conceal the ongoing attack, the attacker needed to implement a routine to suppress the digital alarm in the heat exchanger control loop. Suppressing the digital alarm resulted in hiding information about the deteriorated state of the physical process from the operator and from the controller but did not change the physical state of the process. The “unhealthy” process state propagated into a neighboring plant section causing a low liquid level in an absorber vessel. Even though the absorber alarm was also suppressed to prevent compensating actions from the operator, the degraded state of the process kept propagating downstream, eventually reaching an unsafe limit in the distillation column causing a safety shutdown of the plant. Even though the flow of the digital alarms over the communication link was interrupted, the plant subsystems continued to communicate with each other via the physics of the process. This is a natural information flow in physical processes that cannot be prevented. However, it is important to be aware of such information flows as they may be used by attackers for coordinating attack activities. By

implementing routines for detecting specific process states, the attacker can effectively coordinate attack activities between multiple malicious implants throughout the plant without sending messages via digital communication links.

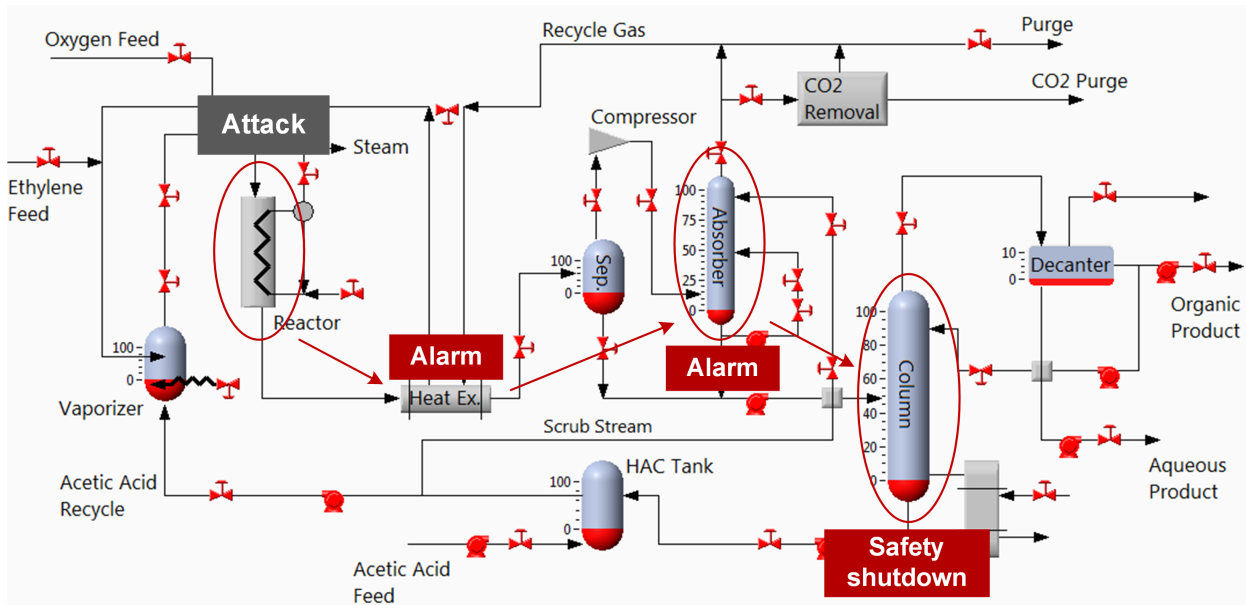


Figure E.3: Propagation of alarms and deteriorated process state

Bibliography

- [1] Ali Abbasi, Majid Hashemi, Emmanuele Zambon, and Sandro Etalle. Stealth Low-Level Manipulation of Programmable Logic Controllers I/O by Pin Control Exploitation. In *International Conference on Critical Information Infrastructures Security*, pages 1–12. Springer, 2016. 25, 65
- [2] Abnormal Situation Management (ASM) Consortium. Official website. <https://www.asmconsortium.net>. Retrieved: June, 2013. 51
- [3] Simumatik Academy. Simumatik Open Emulation Platform: User Manual OEP. https://simumatik.com/learn/ThirdParty/PLC/siemens_TIA. Retrieved: Nov, 2019. xi, 15
- [4] S. Adepu and A. Mathur. Using Process Invariants to Detect Cyber Attacks on a Water Treatment System. In *31st IFIP International Information Security and Privacy Conference (SEC)*, volume AICT-471 of *ICT Systems Security and Privacy Protection*, pages 91–104, 2016. 54
- [5] S. Adepu, J. Prakash, and A. Mathur. WaterJam: An Experimental Case Study of Jamming Attacks on a Water Treatment System. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 341–347, July 2017. 87
- [6] A. Agogino and K. Tumer. Entropy based anomaly detection applied to space shuttle main engines. In *Aerospace Conference, 2006 IEEE*, pages 1–7, 2006. 99, 101, 113
- [7] Chuadhry Mujeeb Ahmed, Martin Ochoa, Jianying Zhou, Aditya P. Mathur, Rizwan Qadeer, Carlos Murguia, and Justin Ruths. NoisePrint: Attack Detection Using Sensor and Process Noise Fingerprint in Cyber Physical Systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, page 483–497, 2018. 54
- [8] Raimo Ahola. The future of calibration is integration. *InTech*, 61(2):44, 2014. 170
- [9] Cristina Alcaraz and Javier Lopez. A cyber-physical systems-based checkpoint model for structural controllability. *IEEE Systems Journal*, 2017. 166
- [10] Rasim Alguliyev, Yadigar Imamverdiyev, and Lyudmila Sukhostat. Cyber-Physical Systems and Their Security Issues. *Computers in Industry*, 100:212–223, 2018. 59
- [11] Emad Ali. Understanding the Operation of Industrial MSF Plants Part I: Stability and Steady-State Analysis. *Desalination*, 143(1):53 – 72, 2002. 36
- [12] Dmitri Alperovitch. Revealed: Operation Shady RAT. Technical report, McAfee, 2011. 121
- [13] F. Alrimawi, L. Pasquale, and B. Nuseibeh. Software Engineering Challenges for Investigating Cyber-Physical Incidents. In *2017 IEEE/ACM 3rd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*, pages 34–40, May 2017. 55

- [14] Julien Amelot, Dhananjay Anand, Thomas Nelson, Gerard Stenbakken, Ya-Shian Li-Baboud, and James Moyne. Towards Timely Intelligence in the Power Grid. In *44th Annual PTTI Meeting*, 2012. 87
- [15] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber Security of Water SCADA Systems—Part I: Analysis and Experimentation of Stealthy Deception Attacks. *IEEE Transactions on Control Systems Technology*, 21(5):1963–1970, September 2013. 50
- [16] Liwei An and Guang-Hong Yang. Improved Adaptive Resilient Control against Sensor and Actuator Attacks. *Information Sciences*, 423:145 – 156, 2018. 54
- [17] Oxana Andreeva, Sergey Gordeychik, Gleb Gritsai, Olga Kochetova, Evgeniya Pot-seluevskaya, Sergey I. Sidorov, and Alexander A. Timorin. Industrial Control Systems and Their Online Availability. Technical report, Kaspersky Lab, 2016. 120
- [18] Uchenna D. Ani, Jeremy M. Watson, Benjamin Green, Barnaby Craggs, and Jason Nurse. Design Considerations for Building Credible Security Testbeds: A Systematic Study of Industrial Control System Use Cases. *arXiv*, 1911.01471, 2019. 58
- [19] Michael J. Assante and Robert M. Lee. The Industrial Control System Cyber Kill Chain. Technical report, SANS Institute, 2015. 147
- [20] Aveva. Aveva Instrumentation Software. <https://www.aveva.com/en/products/electrical-instrumentation>. Retrieved: Nov, 2019. xiii, 123
- [21] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1:11–33, 2004. xi, 27
- [22] T. Backx, O. Bosgra, and W. Marquardt. Integration of Model Predictive Control and Optimization of Processes. Technical report, The Chair of Process Systems Engineering, University of Aachen, 2000. 113
- [23] Carlos Barreto, Alvaro A Cárdenas, and Nicanor Quijano. Controllability of Dynamical Systems: Threat Models and Reactive Security. In *International Conference on Decision and Game Theory for Security*, pages 45–64. Springer, 2013. 50, 165
- [24] Andreas Bathelt, N. Lawrence Ricker, and Mohieddine Jelali. Revision of the Tennessee Eastman Process Model. *IFAC-PapersOnLine*, 48(8):309 – 314, 2015. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015. 58
- [25] R.D. Bell and K.J. Åström. *Dynamic Models for Boiler-Turbine aAlternator Units: Data Logs and Parameter Estimation for a 160 MW Unit*. 1987. 36
- [26] Jonas Berge. Implementing the NAMUR Open Architecture (NOA). <https://www.linkedin.com/pulse/implementing-namur-open-architecture-noa-jonas-berge>, 2019. Retrieved: Feb, 2020. xi, 22
- [27] Ron Bitton, Tomer Gluck, Orly Stan, Masaki Inokuchi, Yoshinobu Ohta, Yoshiyuki Yamada, Tomohiko Yagyu, Yuval Elovici, and Asaf Shabtai. Deriving a Cost-Effective Digital Twin of an ICS to Facilitate Security Evaluation. In *Computer Security*, pages 533–554. Springer International Publishing, 2018. 58

- [28] Irena Bojanova and Jeffrey Voas. Securing the Internet of Anything (IoA). *IEEE Computer Society*, 11 2015. 91
- [29] Alexander Bolshev. ICSCorsair: How I will PWN your ERP through 4-20 mA Current Loop. Black Hat USA, <https://www.blackhat.com/docs/us-14/materials/us-14-Bolshev-ICSCorsair-How-I-Will-PWN-Your-ERP-Through-4-20mA-Current-Loop.pdf>, 2014. Retrieved: Nov, 2019. 33
- [30] Alexander Bolshev, Jason Larsen, Marina Krotofil, and Reid Wightman. A Rising Tide: Design Exploits in Industrial Control Systems. In *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, Austin, TX, Aug 2016. USENIX Association. xii, xiii, 50, 93, 127
- [31] Niklas Borselius. Mobile agent security. *Electronics & Communication Engineering Journal*, 14(5):211–118, 2002. 91, 94
- [32] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry*, 101:1–12, 2018. 21
- [33] J. V. Bradley. *Distribution-Free Statistical Tests*. Prentice Hall, 1968. 95
- [34] Sergey Bratus, Michael Locasto, Meredith Patterson, Len Sassaman, and Anna Shubina. Exploit programming: From buffer overflows to weird machines and theory of computation. *USENIX: ;LOGIN;*, 2011. 157
- [35] Bill Briggs. Hackers hit Norsk Hydro with ransomware. The company responded with transparency. <https://news.microsoft.com/transform/hackers-hit-norsk-hydro-ransomware-company-responded-transparency>, 2019. Retrieved: Nov, 2019. 34
- [36] J. T. Broch. *Mechanical Vibration and Shock Measurements*. Bruel & Kjer, 1984. 24
- [37] Nathan Brubaker, Daniel Kapellmann Zafra, Keith Lunden, Ken Proska, and Corey Hildebrandt. Financially Motivated Actors Are Expanding Access Into OT: Analysis of Kill Lists That Include OT Processes Used With Seven Malware Families. <https://www.fireeye.com/blog/threat-research/2020/07/financially-motivated-actors-are-expanding-access-into-ot.html>, 2020. Retrieved: Nov, 2020. 34
- [38] Brett Burger. How IIoT Monitoring Improves Pump Maintenance. <https://www.flowcontrolnetwork.com/how-iiot-monitoring-improves-pump-maintenance>, 2016. Retrieved: Nov, 2019. 177
- [39] Richard Candell, Dhananjay Anand, and Keith Stouffer. A Cybersecurity Testbed for Industrial Control Systems. In *Proceedings of the 2014 Process Control and Safety Symposium*, 2014. 37, 48
- [40] Richard Candell, Timothy Zimmerman, and Keith Stouffer. Industrial Control System Cybersecurity Performance Testbed. Technical report, NIST, 2015. xi, 48, 49, 52, 58, 59
- [41] Francesca Capaci, Erik Vanhatalo, Murat Kulahci, and Bjarne Bergquist. The revised Tennessee Eastman process simulator as testbed for SPC and DoE methods. *Quality Engineering*, 31(2):212–229, 2019. 58

- [42] A. A. Cárdenas, S. Amin, and S. Sastry. Secure Control: Towards Survivable Cyber-Physical Systems. In *2008 The 28th International Conference on Distributed Computing Systems Workshops*, pages 495–500, June 2008. 50
- [43] Alvaro Cárdenas. *Cyber-Physical Systems Security Knowledge Area*. The Bristol University, 2019. 50
- [44] Alvaro A. Cárdenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar Sastry. Attacks Against Process Control Systems: Risk Assessment, Detection, and Response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 355–366. ACM. 87
- [45] Alvaro A. Cárdenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar Sastry. Attacks against process control systems: risk assessment, detection, and response. In *ASIACCS'11*, pages 355–366, 2011. 54, 85
- [46] Censys. Censys search engine. <https://censys.io>. Retrieved: Dec, 2019. 121
- [47] J.M. Ceron, J.J.Chromik, J.J.C. Santanna, and A. Pras. Online Discoverability and Vulnerabilities of ICS/SCADA Devices in the Netherlands. Technical report, Universiteit Twente, 2019. 120
- [48] Kaspersky ICS CERT. MontysThree: Industrial Espionage with Steganography and a Russian Accent on Both Sides. <https://ics-cert.kaspersky.com/reports/2020/10/08/montysthree-industrial-espionage-with-steganography-and-a-russian-accent-on-both-sides/>, 2020. Retrieved: Dec, 2020. 121
- [49] Kaspersy Lab ICS CERT. Spear phishing attack hits industrial companies. <https://ics-cert.kaspersky.com/alerts/2016/12/16/spear-phishing-attack-hits-industrial-companies>, 2016. Retrieved: Feb, 2017. 120
- [50] Ahmet Cetinkaya, Hideaki Ishii, and Tomohisa Hayakawa. An Overview on Denial-of-Service Attacks in Control Systems: Attack Models and Security Analyses. *Entropy*, 21(2), 2019. 87
- [51] Rong Chen, Kedar Dave, Thomas J. McAvoy, and Michael Luyben. A Nonlinear Dynamic Model of a Vinyl Acetate Process. *Industrial & Engineering Chemistry Research*, 42(20): 4478–4487, 2003. 36, 44, 45
- [52] Rong Chen, Kedar Dave, Thomas J. McAvoy, and Michael Luyben. Code repository: A Nonlinear Dynamic Model of a Vinyl Acetate Process. <http://www.isr.umd.edu/~mcavoy/VAC%20Material>, 2003. Retrieved: June, 2013. 46
- [53] Peng Cheng, Heng Zhang, and Jiming Chen. *Cyber Security for Industrial Control Systems: From the Viewpoint of Close-Loop*. CRC Press, 2016. 52, 85
- [54] Yulia Cherdantseva, Pete Burnap, Andrew Blyth, Peter Eden, Kevin Jones, Hugh Soulsby, and Kristan Stoddart. A Review of Cyber Security Risk Assessment Methods for SCADA Systems. *Computers & Security*, 56:1 – 27, 2016. 51
- [55] Anton Cherepanov. WIN32/INDUSTROYER: A new threat for industrial control systems. Technical report, ESET, 2017. 25, 34, 130, 147

- [56] Elizabeth Chew, Marianne Swanson, Kevin Stine, Nadya Bartol, Anthony Brown, and Will Robinson. NIST SP 800-55, Performance Measurement Guide for Information Security. *National Institute of Standards and Technology*, 2008. 52
- [57] Eric Chien and Gavin O’Gorman. The Nitro Attacks: Stealing Secrets from the Chemical Industry. Technical report, Symantec, 2011. 121
- [58] D. D. Clark and D. R. Wilson. A comparison of commercial and military computer security policies. In *IEEE Symposium on Security and Privacy*, pages 184–184, 1987. 156
- [59] A.W. Colombo, T. Bangemann, S. Karnouskos, J. Delsing, P. Stluka, R. Harrison, F. Jammes, and J.L. Lastra. *Industrial Cloud-Based Cyber-Physical Systems*. Springer, 2014. 21
- [60] United States Nuclear Regulatory Commission. Planning and Installation Guide for Tricon v9–v10 Systems. <https://www.nrc.gov/docs/ML0932/ML093290420.pdf>, 2009. Retrieved: Dec, 2017. xi, 31
- [61] IEC/SC 65B committee. IEC 61132-3:2013 - Programmable Controllers - Part 3: Programming Languages. Standard, International Electrotechnical Commission, 2013. 15
- [62] ISA 18.2 committee. IEC 62682:2015 - Management of Alarms Systems for the Process Industries. Standard, International Electrotechnical Commission, 2015. xi, 18
- [63] ISA-95 Committee. Enterprise-Control System Integration. <https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa95>, . Retrieved: Nov, 2019. 11, 22
- [64] ISA-99 Committee. Industrial Automation and Control Systems Security. <https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa99>, . retrieved: Nov, 2019. 11, 175
- [65] ISA-99 Committee. IEC 62443 - International Series of Cybersecurity Standards for Operational Technology in Automation and Control Systems. Standard, International Electrotechnical Commission, 2010. 11
- [66] Kyiv Energy Construction Company. Reconstruction of the substation "Severnaya". <https://www.youtube.com/watch?v=VLwgprHrqMI>, 2014. Retrieved: Feb, 2020. 122
- [67] Industrial Internet Consortium. *Journal of Innovation: Digital twin*, volume 12. November 2019. 58
- [68] J.P. Contreras, J.C. Naranjo, S. Ramirez, D.M. Martinez, and I.D. Gil. Vinyl Acetate from Ethylene, Acetic Acid and Oxygen Industrial Plant Simulation. In *Proceedings of the Computing and Systems Technology Division, American Institute of Chemical Engineers (AIChE) Annual Meeting*, AIChE, pages 249–259, 2009. 46
- [69] MITRE Corporation. MITRE ATTCK Matrix for ICS. <https://attack.mitre.org/matrices/ics/>. Retrieved: Nov, 2019. 50, 147
- [70] CPS Steering Group. Cyber-Physical Systems Executive Summary. http://iccps.acm.org/2013/_doc/CPS-Executive-Summary.pdf, 2008. Retrieved: Dec, 2013. 1
- [71] LJ create. The Control and Instrumentation Education. <http://www.ljcreate.com/us/programs/engineering/control-and-instrumentation>. Retrieved: Nov, 2019. 56

- [72] John Cusimano and Eric Byres. Safety and Security: Two Sides of the Same Coin. *Control-Global*, 2010. 28
- [73] U.S. Cybersecurity and Infrastructure Security Agency. MEMS Accelerometer Hardware Design Flaws. <https://www.cisa.gov/uscert/ics/alerts/ICS-ALERT-17-073-01A>, 2017. Retrieved: Jan, 2022. 93
- [74] Adrian Dabrowski, Johanna Ullrich, and Edgar R Weippl. Grid Shock: Coordinated Load-Changing Attacks on Power Grids: The Non-Smart Power Grid is Vulnerable to Cyber Attacks as Well. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 303–314. ACM, 2017. 175
- [75] Suresh K. Damodaran and Paul D. Rowe. Limitations on Observability of Effects in Cyber-Physical Systems. In *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security*, HotSoS '19, 2019. 53
- [76] dark lbp. S7 Bruteforce Modules. <https://github.com/dark-lbp/isf/tree/master/icssexploit/modules/creds>, 2017. Retrieved: Feb, 2020. 34
- [77] dark lbp. Industrial Control System Exploitation Framework. <https://github.com/dark-lbp/isf>, 2017. Retrieved: Nov, 2019. xiii, 34, 122
- [78] dark lbp. Credentials wordlists. <https://github.com/dark-lbp/isf/tree/master/icssexploit/wordlists>, 2017. Retrieved: Feb, 2020. 34
- [79] DataGenetics. The Secretary Puzzle. <http://datagenetics.com/blog/december32012/index.html>. Retrieved: Nov, 2019. xii, 70
- [80] dataPARC. Process Data Compression: Why It's A BAD Idea. <http://www.dataparc.com/2016/01/12/process-data-compression-why-its-a-bad-idea>, 2016. Retrieved: Nov, 2019. 55
- [81] Food Department for Environment and Rural Affairs. Waste water treatment in the United Kingdom – 2012. Implementation of the European Union Urban Waste Water Treatment Directive – 91/271/EEC. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/69592/pb13811-waste-water-2012.pdf, 2012. Retrieved: Nov, 2019. 167
- [82] Kegong Diao and Wolfgang Rauch. Controllability analysis as a pre-selection method for sensor placement in water distribution systems. *Water Research*, 47(16):6097 – 6108, 2013. 53
- [83] Alexandre C. Dimian and Costin Sorin Bildean. *Chemical Process Design: Computer-Aided Case Studies*. WILEY-VCH Verlag GmbH & Col, 2008. 45, 146
- [84] Derui Ding, Qing-Long Han, Yang Xiang, Xiaohua Ge, and Xian-Ming Zhang. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275:1674 – 1683, 2018. 55
- [85] DNP3 Users Group. Further Information Regarding the Release of DNP3 Secure Authentication Version 5 (SAv5). <http://www.dnp.org/DNP3Downloads/DNP320SAv5FurtherInfoAnnouncement20111201.pdf>, 2011. Retrieved: May, 2014. 61

- [86] Matan Dobrushin and Yoav Flint Rosenfeld. Project RunAway, 2019. CS3sthlm, <https://cs3sthlm.se/program/presentations/matan-yoav>, 2019. Retrieved: Nov, 2019. 122
- [87] J. J. Downs and E. F. Vogel. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245–255, 1993. xv, 36, 38, 39, 41
- [88] Marcin Dudek. TRITON original samples. https://github.com/MDudek-ICS/TRISIS-TRITON-HATMAN/tree/master/original_samples, 2017. Retrieved: Feb, 2020. 34
- [89] P. R. Dunaka and B. McMillin. Cyber-Physical Security of a Chemical Plant. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pages 33–40, 2017. 113, 165
- [90] Gregory Duranso. Development of High-Performance HMI Philosophy. <https://realpars.com/hmi-philosophy>, 2021. Retrieved: Sept, 2022. xi, 17
- [91] E-ISAC. Analysis of the Cyber Attack on the Ukrainian Power Grid. https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf, 2016. Retrieved: Nov, 2019. 30
- [92] eBay Inc. eBay e-commerce platform. <https://www.ebay.com>. Retrieved: Feb, 2020. 33
- [93] LLC "NTK ENPASELECTRO". Completed projects: Control systems. <https://enpaselectro.com/ru/tehnologii/sistemy-upravleniya.html>. Retrieved: Feb, 2020. 122
- [94] Éreann Leverett and Ried Wightman. Vulnerability Inheritance Programmable Logic Controllers. *GreHack'13*, 2013. 25
- [95] Adriana Erickson. RTU/PLC Communication. <https://control.com/forums/threads/rtu-plc-communication.2951>, 2001. Retrieved: Nov, 2019. 169
- [96] Jack Evans. Someone tried to poison Oldsmar's water supply during hack, sheriff says. <https://www.tampabay.com/news/pinellas/2021/02/08/someone-tried-to-poison-oldsmars-water-supply-during-hack-sheriff-says>, 2021. Retrieved: March, 2021. 30
- [97] F-Secure. Havex Hunts for ICS/SCADA Systems. <https://www.f-secure.com/weblog/archives/00002718.html>, 2014. Retrieved: July, 2014. 120
- [98] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32.Stuxnet Dossier. Symantec, https://www.wired.com/images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf, 2011. Retrieved: Feb, 2020. 29
- [99] Shuai Feng and Pietro Tesi. Resilient Control under Denial-of-Service: Robust Design. *Automatica*, 79:42 – 51, 2017. 53
- [100] Thomas S. Ferguson. Who Solved the Secretary Problem? *Statistical Science*, 4(3):282–296, 1989. 69
- [101] Riccardo M.G. Ferrari and André M.H. Teixeira. Detection and Isolation of Replay Attacks through Sensor Watermarking. *IFAC-PapersOnLine*, 50(1):7363 – 7368, 2017. 20th IFAC World Congress. 61

- [102] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. Multivariate Industrial Time Series with Cyber-AttackSimulation: Fault Detection Using an LSTM-basedPredictive Data Model. *arxiv*, 1612.06676, 2016. 59
- [103] Pavel Filonov, Fedor Kitashov, and Andrey Lavrentyev. RNN-based Early Cyber-Attack Detection for the Tennessee Eastman Process. *arXiv*, 1709.02232, 2017. 59
- [104] Fischertechnik. Fischertechnik Training Models. <https://www.fischertechnikwebshop.com/en-gb/fischertechnik-training-models-en-gb>. Retrieved: Nov, 2019. 56
- [105] David Formby, Milad Rad, and Raheem Beyah. Lowering the barriers to industrial control system security with GRFICS. In *2018 USENIX Workshop on Advances in Security Education (ASE 18)*. USENIX Association, 2018. 57, 59
- [106] OPC Foundation. What is OPC? <https://opcfoundation.org/about/what-is-opc>. Retrieved: Feb, 2020. 23
- [107] PR Freeman. The secretary problem and its extensions: A review. *International Statistical Review/Revue Internationale de Statistique*, pages 189–206, 1983. 68
- [108] Kevin Fu and Wenyuan Xu. Risks of Trusting the Physics of Sensors. *Communications of ACM*, 61(2):20–23, Jan 2018. 92, 175
- [109] Aidan Fuller, Zhong Fan, and Charles Day. Digital Twin: Enabling Technology, Challenges and Open Research. *arXiv*, 1911.01276, 2019. 58
- [110] Jianlei Gao, Senchun Chai, Baihai Zhang, and Yuanqing Xia. Research about DoS Attack against ICPS. *Sensors*, 19:1542, 03 2019. 85
- [111] B. Genge, C. Siaterlis, and M. Hohenadell. Impact of network infrastructure parameters to the effectiveness of cyber attacks against industrial control systems. *IjCCC*, 7(4):673–686, 2012. 87
- [112] Béla Genge and Christos Siaterlis. An Experimental Study on the Impact of Network Segmentation to the Resilience of Physical Processes. In *LNCS*, volume 7289, pages 121–134. 2012. 53, 87
- [113] Béla Genge, Christos Siaterlis, and Marc Hohenadel. Impact of Network Infrastructure Parameters to the Effectiveness of Cyber Attacks Against Industrial Control Systems. *International Journal of Computers Communications & Control*, 2012. 54
- [114] John Gilbert and Frederik Mosteller. Recognizing the Maximum of a Sequence. *Journal of the American Statistical Assosiation*, pages 35–73, 1966. 70
- [115] J. Giraldo, E. Sarkar, A. A. Cárdenas, M. Maniatakos, and M. Kantarcioglu. Security and Privacy in Cyber-Physical Systems: A Survey of Surveys. *IEEE Design Test*, 34(4):7–17, 2017. 59
- [116] Jairo Giraldo and Alvaro A. Cárdenas. A New Metric to Compare Anomaly Detection Algorithms in Cyber-Physical Systems. In *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security, HotSoS '19*, 2019. 52

- [117] Jairo Giraldo, David Urbina, Alvaro Cárdenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. *ACM Comput. Surv.*, 51(4), 2018. 52
- [118] Vincenzo Giuliano and Valerio Formicola. ICSrange: A Simulation-based Cyber Range Platform for Industrial Control Systems. *arXiv*, 1909.01910, 2019. 57
- [119] GLEG. SCADA+ Pack - 0day exploits. http://gleg.net/agora_scada.shtml. Retrieved: Nov, 2019. 34
- [120] Dieter Gollmann. Veracity, Plausibility, and Reputation. In *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems*, volume 7322 of *LNCs*, pages 20–28. 2012. 91
- [121] Dieter Gollmann, Pavel Gurikov, Alexander Isakov, Marina Krotofil, Jason Larsen, and Alexander Winnicki. Cyber-physical systems security: Experimental Analysis of a Vinyl Acetate Monomer Plant. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, pages 1–12. ACM, 2015. 177
- [122] Shelby Grad. Engineers who hacked into L.A. traffic signal computer, jamming streets, sentenced. <https://latimesblogs.latimes.com/lanow/2009/12/engineers-who-hacked-in-la-traffic-signal-computers-jamming-traffic-sentenced.html>, 2009. Retrieved: Nov, 2019. 29
- [123] Benjamin Green, Marina Krotofil, and David Hutchison. Achieving ICS Resilience and Security through Granular Data Flow Management. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, CPS-SPC '16, page 93–101, 2016. xiii, 168
- [124] Benjamin Green, Marina Krotofil, and Ali Abbasi. On the Significance of Process Comprehension for Conducting Targeted ICS Attacks. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)*, CPS '17, pages 57–67. ACM, 2017. 122, 123
- [125] Andrew Greenberg. How an Entire Nation Became Russia's Test Lab for Cyberwar. <https://web.archive.org/web/20170624161851/https://www.wired.com/story/russian-hackers-attack-ukraine>, 2017. Retrieved: Nov, 2019. 147
- [126] Michael Grieves and John Vickers. *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*, pages 85–113. 08 2017. 58
- [127] G. Grispos, W. B. Glisson, and K. R. Choo. Medical Cyber-Physical Systems Development: A Forensics-Driven Approach. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 108–113, July 2017. 55
- [128] Alarm Systems User Group. EEMUA 191. Alarm systems - A guide to design, management and procurement. Standard, EEMUA, 1999. xv, 17, 18
- [129] Alibaba Group. Alibaba e-commerce platform. <https://www.ebay.com>. Retrieved: Feb, 2020. 33
- [130] Gruenbeck. Operating instructions: Ultrafiltration system GENO-Ultrafil 450/900 with GENO-matic. 2017. 127

- [131] GRUNDFOS. AQpure UF, Modular water treatment system. Service instructions. 2010. 127
- [132] Yuriy Gurkin. SCADA Projects - Hackers' Point of View. ZeroNights, <https://2018.zeronights.ru/wp-content/uploads/materials/21-SCADA-projects-from-the-point-of-view-of-hackers.pdf>, 2018. Retrieved: Dec, 2018. 147
- [133] Adam Hahn, Roshan K. Thomas, Ivan Lozano, and Alvaro Cárdenas. A Multi-Layered and Kill-Chain based Security Analysis Framework for Cyber-Physical Systems. *International Journal of Critical Infrastructure Protection*, 11:39 – 50, 2015. ISSN 1874-5482. 147
- [134] Bil Hallaq, Andrew Nickolson, R. Smith, Leandros Maglaras, Helge Janicke, and Kevin Jones. CYRAN: A Hybrid Cyber Range for Testing Security on ICS/SCADA Systems. 07 2016. 57
- [135] P. Haller and B. Genge. Using Sensitivity Analysis and Cross-Association for the Design of Intrusion Detection Systems in Industrial Cyber-Physical Systems. *IEEE Access*, 5:9336–9347, 2017. 51, 87
- [136] F. Hamill. Sayano Shushenskaya accident – presenting a possible direct cause. *Plant Power and Dam Construction: online magazine*, 2010. URL <http://www.waterpowermagazine.com/features/featuresayano-shushenskaya-accident-presenting-a-possible-direct-cause>. Retrieved: December, 2013. 26
- [137] Y.-F. Han, D. Kumar, C. Sivadinarayana, and D.W. Goodman. Kinetics of ethylene combustion in the synthesis of vinyl acetate over a Pd/SiO₂ catalyst. *Journal of Catalysis*, 224(1):60 – 68, 2004. 44
- [138] Chris Harper. AIV and FIV in Pipelines, Plants, and Facilities. volume Volume 1: Pipelines and Facilities Integrity of *International Pipeline Conference*, 09 2016. 169
- [139] Yoshihiro Hashimoto, Takeshi Toyoshima, Shuichi Yogo, Masato Koike, Takashi Hamaguchi, Sun Jing, and Ichiro Koshijima. Safety securing approach against cyber-attacks for process control system. *Computers & Chemical Engineering*, 57:181 – 186, 2013. 51
- [140] Zhongyuan Hau, John H. Castellanos, and Jianying Zhou. Evaluating Cascading Effects of Attacks on Resilience of Industrial Control Systems: A Design-Centric Modeling Approach. *arXiv*, 1905.03156, 2019. 52
- [141] Manus Henry and Graeme Wood. Sensor validation: principles and standards. *ATP International*, 3:39–52, 04 2005. 95
- [142] Amir Herzberg and Yehonatan Kfir. The Chatty-Sensor: A Provably-Covert Channel in Cyber Physical Systems. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, page 638–649, 2019. 54
- [143] Stephen Hilt, Numaan Huq, Vladimir Kropotov, Robert McArdle, Cedric Pernet, and Roel Reyes. Exposed and Vulnerable Critical Infrastructure: Water and Energy Industries. Technical report, TrendMicro, 2018. 121
- [144] Andreas Hoehn and Ping Zhang. Detection of replay attacks in cyber-physical systems. In *2016 American Control Conference (ACC)*, pages 290–295, 07 2016. 61
- [145] Daniel M. Horowitz and Sandy Gilmour. CSB safety videos: A new and effective communication tool for accident investigation findings. *Process Safety Progress*, 26(3):177–179, 2007. 57

- [146] HSE. Accidents and investigations. <https://www.hse.gov.uk/toolbox/managing/accidents.htm>. Retrieved: Feb, 2020. 30
- [147] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Technical report, Lockheed Martin Corporation, 2011. 115, 147
- [148] ICS-CERT. ICS Focused Malware. <https://us-cert.gov/ics/alerts/ICS-ALERT-14-176-02A>, 2014. Retrieved: Feb, 2015. 30, 120
- [149] ICS-CERT. Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors. <https://www.us-cert.gov/ncas/alerts/TA18-074A>, 2018. Retrieved: March, 2018. 119, 121
- [150] GitHub Inc. GitHub: Internet hosting for software development. <https://github.com>, . Retrieved: Feb, 2020. 33
- [151] Scribd Inc. Scribd digital library. www.scribd.com, . Retrieved: Nov, 2019. 33
- [152] Joint Task Force Transformation Initiative. NIST SP 800-53A Rev.1. Guide for Assessing the Security Controls in Federal Information Systems and Organizations. *National Institute of Standards and Technology*, 2010. 27
- [153] CS INSTRUMENTS. Intelligent chart recorder DS 500 for compressed air and gases. <https://www.cs-instruments.com/products/d/chart-recorder/ds-500-intelligent-chart-recorder>. Retrieved: Nov, 2019. 169
- [154] iTRUST Research Centre. Publications. <https://itrust.sutd.edu.sg/research/publications>, . Retrieved: Nov, 2019. 58
- [155] iTRUST Research Centre. Garden of Testbeds: Booking Requests. https://itrust.sutd.edu.sg/itrust-labs_overview, . Retrieved: Nov, 2019. xi, 12, 35, 128
- [156] iTRUST Research Centre. iTRUST Labs Datasets. https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info, 2018. Retrieved: Nov, 2019. 56
- [157] iTRUST Research Centre. Secure Water Treatment (SWaT) Testbed. <http://locnuoctienthanh.net/document/water/ReverseOsmosisPID.pdf>, 2018. Retrieved: Feb, 2020. xiii, 128
- [158] M. Iturbe, J. Camacho, I. Garitano, U. Zurutuza, and R. Uribeetxeberria. On the Feasibility of Distinguishing Between Process Disturbances and Intrusions in Process Control Systems Using Multivariate Statistical Process Control. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 155–160, June 2016. 87
- [159] Blake Johnson, Dan Caban, Marina Krotofil, Dan Scali, Nathan Brubaker, and Christopher Glycer. Attackers Deploy New ICS Attack Framework “TRITON” and Cause Operational Disruption to Critical Infrastructure. <https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html>, 2017. Retrieved: Nov, 2019. xiii, 25, 31, 119, 120, 130, 146

- [160] Karupis Lab. METIS - Serial Graph Partitioning and Fill-Reducing Matrix Ordering. <http://glaros.dtc.umn.edu/gkhome/views/metis>, 2013. Retrieved: July, 2014. 104
- [161] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998. 104
- [162] G. Karypis and Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48:96–129, 1998. 104
- [163] George Karypis and Vipin Kumar. Multilevel graph partitioning schemes. In *The 24th International Conference on Parallel Processing*, pages 113–122, 1995. 104
- [164] Georgios Kavallieratos, Sokratis K. Katsikas, and Vasileios Gkioulos. Towards a cyber-physical range. In *Proceedings of the 5th on Cyber-Physical System Security Workshop, CPSS '19*, pages 25–34, 2019. 57
- [165] Abdullah Khalili, Ashkan Sami, Mahdi Azimi, Sara Moshtari, Zahra Salehi, Mahboobe Ghiasi, and Ali Akbar Safavi. Employing Secure Coding Practices into Industrial Applications: A Case Study. *Empirical Software Engineering*, 21, 12 2014. DOI 10.1007/s10664-014-9341-9. 53
- [166] Istvan Kiss, Haller Pirooska, and Adela Beres. Denial of Service Attack Detection in Case of Tennessee Eastman Challenge Process. *Procedia Technology*, 19:835–841, 12 2015. DOI 10.1016/j.protcy.2015.02.120. 87
- [167] István Kiss, Bela Genge, Pirooska Haller, and Gheorghe Sebestyén. Data clustering-based anomaly detection in industrial control systems. In *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 275–281, 2014. 113
- [168] István Kiss, Bela Genge, and Pirooska Haller. A Clustering-Based Approach to Detect Cyber Attacks in Process Control Systems. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pages 142–148, 2015. 113
- [169] Paul Kocher, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. *arXiv preprint arXiv:1801.01203*, 2018. 175
- [170] C. Konstantinou, M. Sazos, and M. Maniatakos. Attacking the smart grid using public information. In *2016 17th Latin-American Test Symposium (LATS)*, pages 105–110, 2016. 121
- [171] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462, May 2010. 87
- [172] Eduard Kovac. Accenture Launches Cyber Ranges for Industrial Companies. <https://www.securityweek.com/accenture-launches-cyber-ranges-industrial-companies>, 2018. Retrieved: Nov, 2019. 57
- [173] Brian Krebs. Chinese Hackers Blamed for Intrusion at Energy Industry Giant Telvent. <https://krebsonsecurity.com/2012/09/chinese-hackers-blamed-for-intrusion-at-energy-industry-giant-telvent/>, 2012. Retrieved: Sept, 2012. 120
- [174] Brian Krebs. Security Firm Bit9 Hacked, Used to Spread Malware. <http://krebsonsecurity.com/2013/02/security-firm-bit9-hacked-used-to-spread-malware/>, 2013. Retrieved: Dec, 2013. 120

- [175] P. Krishnamurthy, F. Khorrami, R. Karri, D. Paul-Pena, and H. Salehghaffari. Process-Aware Covert Channels Using Physical Instrumentation in Cyber-Physical Systems. *IEEE Transactions on Information Forensics and Security*, 13(11):2761–2771, Nov 2018. 54
- [176] Marina Krotofil. What I Learned about ICS Security from Cyber-Physical Hacking. DHS IC-SJWG, <https://www.slideshare.net/MarinaKrotofil/dhs-icsjwg-2015v3>, 2015. Retrieved: Nov, 2019. 170
- [177] Marina Krotofil. EVIL BUBBLES or How to Deliver Attack Payload via the Physics of the Process. Black Hat USA, <https://www.blackhat.com/docs/us-17/wednesday/us-17-Krotofil-Evil-Bubbles-Or-How-To-Deliver-Attack-Payload-Via-The-Physics-Of-The-Process.pdf>, 2017. Retrieved: Nov, 2019. xiii, 176, 177
- [178] Marina Krotofil and A. D. Hack Like a Movie Star: Step-by-step guide to crafting SCADA payloads for physical attacks with catastrophic consequences. ZeroNights, <https://repo.zenk-security.com/Conferences/ZeroNights/12-Krotofil.pdf>. Retrieved: Feb, 2020. xiii, 148, 155
- [179] Marina Krotofil and Alexander Isakov. Damn Vulnerable Chemical Process - Tennessee Eastman Process. <http://github.com/satejnik/DVCP-TE>, 2015. Retrieved: Jan, 2015. 36
- [180] Marina Krotofil and Alexander Isakov. Damn Vulnerable Chemical Process - Vinyl Acetat Monomer. <http://github.com/satejnik/DVCP-VAM>, 2015. Retrieved: Jan, 2015. 36
- [181] Marina Krotofil, Jason Larsen, and Dieter Gollmann. The process matters: Ensuring data veracity in cyber-physical systems. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 133–144. ACM, 2015. 165
- [182] Kaspersky Lab. Gas-Oil heating loop dataset. https://kas.pr/ics-research/dataset_ghl_1, 2016. Retrieved: Nov, 2019. 56
- [183] Kaspersky Lab. Tennessee Eastman Process Dataset. <https://box.kaspersky.com/d/56ad570fca>, 2017. Retrieved: Nov, 2019. 56
- [184] Jason Laberge. Introduction to ASM. <https://www.asmconsortium.net/resources/presentations/Pages/default.aspx>, 2008. Retrieved: Nov, 2019. xii, 86
- [185] Canary Labs. A Guide to the Best Data Historian Software: A Review of the Canary Historian Versus Rockwell FactoryTalk or OSIsoft Pi. <https://blog.canarylabs.com/2016/06/27/a-guide-to-the-best-data-historian-software-a-review-of-the-canary-historian-versus-rockwell-factorytalk-and-osisoft-pi>, 2016. Retrieved: Nov, 2019. xi, 19
- [186] Butler W Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, 1973. 175
- [187] Ralf Langner. To kill a centrifuge. Technical report, Langner Communications, 2013. 25, 30, 48, 94, 116, 117, 159
- [188] Jason Larsen. Breakage. Black Hat Federal, <http://www.orkspace.net/secdocs/Conferences/BlackHat/Federal/2008/SCADA%20Security.pdf>, 2007. Retrieved: May, 2013. xiii, 24, 26, 50, 116, 118, 156

- [189] Jason Larsen. Miniaturization. Black Hat USA, <https://www.blackhat.com/docs/us-14/materials/us-14-Larsen-Miniturization-WP.pdf>, 2014. Retrieved: Feb, 2015. 49, 87, 95, 96, 98
- [190] Jason Larsen. The Stale Data Problem. SANS ICS Summit, <https://www.sans.org/event-downloads/43347/brochure.pdf>, 2016. Retrieved: Nov, 2019. 87, 156
- [191] Jason Larsen. Hacking Critical Infrastructure Like You're Not a N00b. RSA, <https://www.slideshare.net/cisoplatform7/hacking-critical-infrastructure-like-youre-not-a-n00b>, 2016. Retrieved: Nov, 2019. xiii, 131, 147
- [192] Truls Larsson and Sigurd Skogestad. Plantwide control – A review and a new design procedure. *Modeling, Identification and Control*, 21(4):209–240, 2000. 10
- [193] Truls Larsson, Kristin Hestetun, Espen Hovland, and Sigurd Skogestad. Self-optimizing control of a large-scale plant: The Tennessee Eastmann process. *Ind. Eng. Chem. Res.*, 40(22): 4488–4901, 2001. xi, 37, 38
- [194] Andrey Lavrentyev. MLAD: Machine Learning for Anomaly Detection. Kaspersky Lab, <https://ics-cert.kaspersky.com/reports/2018/01/16/mlad-machine-learning-for-anomaly-detection>, 2018. Retrieved: Nov, 2019. 57, 59
- [195] Andrey Lavrentyev. Monitoring technological processes at TANECO oil refinery with Kaspersky MLAD. Technical report, Kaspersky Lab, 2019. 59
- [196] Edward A. Lee. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors*, 15(3):4837–4869, 2015. 1
- [197] Robert A. Leishear. *Fluid Mechanics, Water Hammer, Dynamic Stresses, and Piping Design*. ASME, 2013. 26, 77, 117
- [198] Eireann Leverett and Reid Wightman. Vulnerability inheritance programmable logic controllers. In *Proceedings of the Second International Symposium on Research in Grey-Hat Hacking*, 2013. 121
- [199] N. G. Leveson. *Engineering a Safer World: Systems Thinking Applied to Safety*. The MIT Press, 2012. 27
- [200] Timothy Levine, Hee Park, and Steven McCornack. Accuracy in Detecting Truths and Lies: Documenting the "Veracity Effect". *Communication Monographs - COMMUN MONOGR*, 66: 125–144, 06 1999. 91
- [201] John Leyden. Hacker jailed for revenge sewage attacks. https://www.theregister.com/2016/03/24/water_utility_hacked, 2016. Retrieved: Nov, 2019. 30
- [202] Songnian Li, Suzana Dragicevic, Francesc Antón Castro, Monika Sester, Stephan Winter, Arzu Coltekin, Christopher Pettit, Bin Jiang, James Haworth, Alfred Stein, and Tao Cheng. Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:119–133, 2016. 112
- [203] Y. Li, L. Shi, P. Cheng, J. Chen, and D. E. Quevedo. Jamming attack on Cyber-Physical Systems: A game-theoretic approach. In *2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, pages 252–257, May 2013. 85

- [204] Ondrej Linda, Milos Manic, and Miles McQueen. Improving Control System Cyber-State Awareness Using Known Secure Sensor Measurements. In *Critical Information Infrastructures Security*, volume 7722 of *LNCS*, pages 46–58. 2013. 113
- [205] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown. *arXiv preprint arXiv:1801.01207*, 2018. 175
- [206] Bela G. Liptak. *Instrument Engineers' Handbook, Vol. 2: Process Control and Optimizatiol*. CRC Press, 2005. 53
- [207] Z. Lu, X. Lu, W. Wang, and C. Wang. Review and Evaluation of Security Threats on the Communication Networks in the Smart Grid. In *2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE*, pages 1830–1835, Oct 2010. 85
- [208] Z. Lu, W. Wang, and C. Wang. From Jammer to Gambler: Modeling and Detection of Jamming Attacks Against Time-Critical Traffic. In *2011 Proceedings IEEE INFOCOM*, pages 1871–1879, April 2011. 85
- [209] Lucas Apa and Carlos Mario Penagos Hollman. Compromising Industrial Facilities from 40 Miles Away. Technical report, IOActive, 2013. 33
- [210] Michael L. Luyben and Björn D. Tyréus. An industrial design/control study for the vinyl acetate monomer process. *Computers & Chemical Engineering*, 22(7–8):867 – 877, 1998. 36, 45, 46, 142, 146
- [211] William L. Luyben, Bjorn D. Tyreus, and Michael L. Luyben. *PlantwideProcess Control*. McGraw-Hill, 1998. 10, 24, 37, 45, 47, 51, 53, 54, 136
- [212] Mohammad Mahdian, Randolph Preston McAfee, and David Pennock. The secretary problem with a hazard rate condition. In *Internet and Network Economics*, pages 708–715. 2008. 70
- [213] FireEye Mandiant. Mandiant Attack Life Cycle. <https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/ds-threatspace.pdf>, 2021. Retrieved: Nov, 2021. xiii, 115
- [214] Sam Mannan. *Lees' Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control*, volume 1. Butterworth Heinemann, 2005. 25
- [215] Carla Martin-Villalba, Alfonso Urquia, and Guodong Shao. Implementations of the Tennessee Eastman Process in Modelica. *IFAC-PapersOnLine*, 51(2):619 – 624, 2018. 9th Vienna International Conference on Mathematical Modelling. 59
- [216] John C. Matherly. SHODAN: Industrial Control Systems. <https://www.shodan.io/explore/category/industrial-control-systems>. Retrieved: Dec, 2019. 121
- [217] Lee Mathews. Criminals Hacked A Fish Tank To Steal Data From A Casino. <https://www.forbes.com/sites/leemathews/2017/07/27/criminals-hacked-a-fish-tank-to-steal-data-from-a-casino/#656917b732b9>, 2017. Retrieved: Nov, 2019. 33
- [218] A. P. Mathur and N. O. Tippenhauer. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, 2016. 35, 58, 127

- [219] MathWorks. Matlab. <https://www.mathworks.com/products/matlab.html>, . Retrieved: Nov, 2013. 36
- [220] MathWorks. Simulink. <https://www.mathworks.com/products/simulink.html>, . Retrieved: Nov, 2013. 46
- [221] McAfee Foundstone Professional Services and McAfee Labs. Global Energy Cyberattacks: "Night Dragon". Technical report, McAfee, 2011. 121
- [222] T.J. McAvoy and N. Ye. Base control for the Tennessee Eastman problem. *Computers & Chemical Engineering*, 18(5):383 – 413, 1994. 37
- [223] Thomas McEvoy and Stephen Wolthusen. A Plant-Wide Industrial Process Control Security Problem. In *CIP V*, volume 367, pages 47–56. 2011. 54, 61, 85
- [224] Thomas Richard McEvoy and Stephen D. Wolthusen. Detecting Sensor Signal Manipulations in Non-Linear Chemical Processes. In *CIP IV*, pages 81–94, 2010. 54, 113
- [225] McEvoy, Thomas Richard and Wolthusen, Stephen D. Trouble brewing: Using observations of invariant behavior to detect malicious agency in distributed control systems. In *Critical Information Infrastructures Security*, pages 62–72, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 59
- [226] Jim McGlone. Stale Data, Process Running. <https://www.kenexis.com/stale-data-process-running>, 2014. Retrieved: Nov, 2019. 62
- [227] Craig McIntyre. Using Smart Instrumentation. *Plant Engineering: online magazine*, 2011. <http://www.controleng.com/single-article/using-smart-instrumentation/a0ec350155bb86c8f65377ba66e59df8.html>. 24, 91, 170
- [228] Miles McQueen and Annarita Giani. ‘Known Secure Sensor Measurements’ for Critical Infrastructure Systems: Detecting Falsification of System State. In *International Workshop on Software Engineering for Resilient Systems*, pages 156–163. Springer, 2011. 177
- [229] Aaron Flecha Menendez. CAFFEINE: Attack Framework for Electric Sector. https://cybercamp.es/sites/default/files/contenidos/videos/adjuntos/cybercamp19_afm_caffeine.pdf, 2019. Retrieved: Nov, 2019. 148
- [230] Dustin Mink, Alec Yasinsac, Kim-Kwang Raymond Choo, and William Bradley Glisson. Next Generation Aircraft Architecture and Digital Forensic. In *AMCIS*, 2016. 55
- [231] Ariana Mirian, Zane Ma, D. Adrian, Matthew Tischler, Thasphon Chuenchujit, Timothy M. Yardley, R. Berthier, J. Mason, Zakir Durumeric, J. A. Halderman, and M. Bailey. An Internet-wide view of ICS devices. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 96–103, 2016. 120
- [232] Yilin Mo, R. Chabukswar, and B. Sinopoli. Detecting Integrity Attacks on SCADA Systems. *Control Systems Technology, IEEE Transactions on*, 22(4):1396–1407, 2014. 54
- [233] Yilin Mo, S. Weerakkody, and B. Sinopoli. Physical Authentication of Control Systems: Designing Watermarked Control Inputs to Detect Counterfeit Sensor Outputs. *Control Systems, IEEE*, 35(1):93–109, 2015. 54

- [234] M. Mueter and N. Asaj. Entropy-based anomaly detection for in-vehicle networks. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 1110–1115, 2011. 112
- [235] Sheppard Mullin. Power Company Slammed With Hefty 2.7M Fine After Data Breach. <https://www.eyeonprivacy.com/2018/03/power-company-regulator>, 2018. Retrieved: Feb, 2020. 33
- [236] M. Nawrocki, T. C. Schmidt, and M. Waehlich. Uncovering Vulnerable Industrial Control Systems from the Internet Core. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2020. 30, 120
- [237] Lee Neitzel and Bob Huba. Top ten differences between ICS and IT cybersecurity. *InTech*, 61(3):12–18, 2014. 89, 170
- [238] Trent Nelso and May Chaffin. Common cybersecurity vulnerabilities in industrial control systems. *Control Systems Security Program*, 2011. 64
- [239] NERC. Critical Infrastructure Protection Standards. <http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx>. Retrieved: Nov, 2014. 27
- [240] Ian Nimmo and Stephen Maddox. A lasting plan for managing alarms. <https://www.controlglobal.com/articles/2015/a-lasting-plan-for-managing-alarms>, 2015. Retrieved: Nov, 2019. xi, 52
- [241] T. Novak and A. Gerstinger. Safety- and Security-Critical Services in Building Automation and Control Systems. *IEEE Transactions on Industrial Electronics*, 57(11):3614–3621, 2010. 27
- [242] NullArray. s7scan. <https://github.com/NullArray/AutoSploit>, 2018. Retrieved: Nov, 2019. 34
- [243] Martin Ochoa and Mujeeb Ahmed Chuadhry. Sensor and Process Fingerprinting in Industrial Control Systems. Black Hat USA, <https://i.blackhat.com/USA-19/Wednesday/us-19-Ochoa-Sensor-And-Process-Fingerprinting-In-ICS.pdf>, 2019. Retrieved: Nov, 2019. 54
- [244] American Institute of Chemical Engineers. Process Safety Incident Database. <http://www3.aiche.org/PSID/Home.aspx>. Retrieved: November, 2019. 126
- [245] D. G. Olsen, W. Y. Svrcek, and B.R. Young. Plantwide Control Study of a Vinyl Acetate Monomer Process Design. *Chemical Engineering Communication*, 192(10):1243 – 1257, 2005. 45
- [246] Danny Palmer. Ransomware attacks are now targeting industrial control systems. <https://www.zdnet.com/article/ransomware-attacks-are-now-targeting-industrial-control-systems>, 2019. Retrieved: Nov, 2019. 34
- [247] Donghui Park, Julia Summers, and Michael Walstrom. Cyberattack on Critical Infrastructure: Russia and the Ukrainian Power Grid Attacks. <https://jsis.washington.edu/news/cyberattack-critical-infrastructure-russia-ukrainian-power-grid-attacks>, 2017. Retrieved: Nov, 2019. 147
- [248] Donn B. Parker. Toward a New Framework for Information Security. In Seymour Bosworth and M. E. Kabay, editors, *The Computer Security Handbook*, chapter 5. John Wiley & Sons, 4 edition, 2002. 89

- [249] Ulyana A. Pozdnyakova, Vyacheslav V. Golikov, Irina A. Peters, and Irina A. Morozova. *Genesis of the Revolutionary Transition to Industry 4.0 in the 21st Century and Overview of Previous Industrial Revolutions*, pages 11–19. 2019. 21
- [250] W. H. Press, B. P. Flannery, S. A Teukolsky, and W. T. Vetterling. *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. Cambridge University Press, 1992. 103
- [251] MANTIS Project. Deep learning for predictive maintenance. <http://www.mantis-project.eu/deep-learning-for-predictive-maintenance>. Retrieved: Nov, 2019. xiii, 156
- [252] Wullianallur Raghupathi and Vijju Raghupathi. Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, 2(1), 2016. 112
- [253] Prashant Hari Narayan Rajput, Pankaj Rajput, Marios Sazos, and Michail Maniatakos. Process-Aware Cyberattacks for Thermal Desalination Plants. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, Asia CCS '19, page 441–452, 2019. 51, 59
- [254] Yongyi Ran, Xiaoxia Zhou, Pengfeng Lin, Yonggang Wen, and Ruilong Deng. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *arXiv*, 1912.07383, 2019. 21
- [255] Rapid7. Vulnerability Exploit Database. <https://www.rapid7.com/db/?type=metasploit>. Retrieved: Feb, 2020. 34
- [256] Adil Rasheed, Omer San, and Trond Kvamsdal. Digital Twin: Values, Challenges and Enablers. *arXiv*, 1910.01719, 2019. 58
- [257] ESET Research. ACAD/Medre.A: 10000's of AutoCAD Designs Leaked in Suspected Industrial Espionage. Technical report, ESET, 2012. 121
- [258] Kudelski Security Research. 6 Months of ICS Scanning. <https://research.kudelskisecurity.com/2017/10/24/6-months-of-ics-scanning>, 2017. Retrieved: Feb, 2018. 120
- [259] N. Lawrence Ricker. Tennessee Eastman Challenge Archive. <http://depts.washington.edu/control/LARRY/TE/download.html>. Retrieved: May, 2013. 38, 40
- [260] N. Lawrence Ricker. Optimal steady-state operation of the Tennessee Eastman challenge process. *Computers & Chemical Engineering*, 19(9):949 – 959, 1995. 51
- [261] N. Lawrence Ricker and J.H. Lee. Nonlinear Model Predictive Control of the Tennessee Eastman Challenge Process. *Comp. & Chem. Engineering*, 19(9):961 – 981, 1995. 37, 51, 54
- [262] S.M. Rinaldi, J.P. Peerenboom, and T.K. Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems, IEEE*, 21(6):11 –25, 2001. 24
- [263] Martin Ring, Reiner Kriesten, and Frank Kargl. Plausibility Checks in Electronic Control Units to Enhance Safety and Security. *International Journal On Advances in Security*, 10(1), 06 2017. 113

- [264] Control Risks. Iranian Cyber Espionage Group Targets Suppliers of Industrial Control Systems. <https://www.controlrisks.com/our-thinking/insights/iranian-cyber-espionage-group-targets-suppliers-of-industrial-control-systems>, 2020. Retrieved: Dec, 2020. 121
- [265] Douglas H. Rothenberg. *Alarm Management for Process Control: A Best-Practice Guide for Design, Implementation, and Use of Industrial Alarm Systems*. Momentum Press, 2009. 132
- [266] Victoria Rubin and Tatiana Lukoianova. Veracity Roadmap: Is Big Data Objective, Truthful and Credible? *Advances In Classification Research Online*, 24(1):4–15, 2014. 112
- [267] Adepu S., Kandasamy N.K., and Mathur A. EPIC: An Electric Power Testbed for Research and Training in Cyber Physical Systems Security. In *Security of Industrial Control Systems and Cyber-Physical Systems (CyberICPS 2018)*, volume 11387 of *LNCS*, pages 20–28. 2018. 35, 58
- [268] U.S. Chemical Safety and Hazard Investigation Board. Videos: Safety Incidents Animation. <https://www.csb.gov/videos>, . Retrieved: Nov, 2019. xii, 57
- [269] U.S. Chemical Safety and Hazard Investigation Board. Completed investigations. <https://www.csb.gov/investigations/completed-investigations>, . Retrieved: November, 2019. 126
- [270] U.S. Chemical Safety and Hazard Investigation Board. BP America Refinery Explosion: Final Investigation Report. <https://www.csb.gov/bp-america-refinery-explosion>, 2007. Retrieved: May, 2013. 26, 170
- [271] U.S. Chemical Safety and Hazard Investigation Board. T2 Laboratories Inc. Reactive Chemical Explosion: Final Investigation Report. <https://www.csb.gov/t2-laboratories-inc-reactive-chemical-explosion>, 2009. Retrieved: May, 2013. 25
- [272] U.S. Chemical Safety and Hazard Investigation Board. Runaway: Explosion at T2 Laboratories, investigation video. <http://www.youtube.com/watch?v=C561PCq5E1g>, 2009. Retrieved: June, 2015. 159
- [273] U.S. Chemical Safety and Hazard Investigation Board. DuPont Corporation Toxic Chemical Releases: Investigation Report. <https://www.csb.gov/dupont-corporation-toxic-chemical-releases>, 2011. Retrieved: May, 2013. 29
- [274] Helem S. Sanchez, Damiano Rotondo, Teresa Escobet, Vicenc Puig, and Joseba Quevedo. Bibliographical review on cyber attacks from a control oriented perspective. *Annual Reviews in Control*, 48:103 – 128, 2019. 50
- [275] Ruben Santamarta. Here be Backdoors: A Journey into the Secrets of Industrial Firmware. Black Hat USA, https://media.blackhat.com/bh-us-12/Briefings/Santamarta/BH_US_12_Santamarta_Backdoors_Slides.pdf, 2012. Retrieved: December, 2012. 25
- [276] SCADAStrangeLove. SCADAPASS. <https://github.com/scadastrangelove/SCADAPASS/blob/master/scadapass.csv>, 2016. Retrieved: Feb, 2020. 34
- [277] Thorsten Schneider. Damn Vulnerable Linux. <https://www.linux.com/tutorials/damn-vulnerable-linux>, 2010. Retrieved: Nov, 2019. 36

- [278] Hiroya Seki, Morimasa Ogawa, Toshiaki Itoh, Shigeki Ootakara, Hisashi Murata, Yoshihiro Hashimoto, and Manabu Kano. Plantwide Control System Design of the Benchmark Vinyl Acetate Monomer Production Plant. *Computers & Chemical Engineering*, 34(8):1282 – 1295, 2010. 45, 46
- [279] Kaspersky Lab Security Services. s7scan. <https://github.com/klseccservices/s7scan>, 2018. Retrieved: Nov, 2019. 34, 122
- [280] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communications*. University of Illinois Press, 1949. 99
- [281] Mahdi Sharifzadeh. Integration of process design and control: A review. *Chemical Engineering Research and Design*, 91(12):2515–2549, 2013. 164, 165
- [282] Y. Shoukry, P. Nuzzo, N. Bezzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada. Secure State Reconstruction in Differentially Flat systems under Sensor Attacks Using Satisfiability Modulo Theory Solving. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3804–3809, 2015. 113
- [283] Yasser Shoukry, Paul Martin, Paulo Tabuada, and Mani Srivastava. Non-Invasive Spoofing Attacks for Anti-Lock Braking System. In *Cryptographic Hardware and Embedded Systems, CHES 2013*, pages 55 – 72. Springer, 2013. 92
- [284] Christos Siaterlis and Béla Genge. Cyber-Physical Testbeds. *Commun. ACM*, 57(6):64–73, June 2014. 58, 59
- [285] Siemens. Industrial 5G – The Wireless Network of the Future. <https://press.siemens.com/global/en/feature/industrial-5g-wireless-network-future>, 2019. Retrieved: Nov, 2019. xii, 86
- [286] Sigma-Aldrich. Paracetamol. <https://www.sigmaaldrich.com/DE/de>. Retrieved: Jan, 2015. xv, 117
- [287] Tony Smith. Hacker jailed for revenge sewage attacks. https://www.theregister.com/2001/10/31/hacker_jailed_for_revenge_sewage, 2001. Retrieved: Nov, 2019. 29
- [288] Jacques F. Smuts. *Process Control for Practitioners*. OptiControls Inc, 2011. xi, 9, 10, 24, 25
- [289] Anthony Sofronas. *Case Histories in Vibration Analysis and Metal Fatigue for the Practicing Engineer*. John Wiley & Sons, 2012. 24, 26
- [290] M. Staron and R. Scandariato. Data veracity in intelligent transportation systems: The slippery road warning scenario. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 821–826, 2016. 112
- [291] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn. NIST special publication 800-82, revision 2: Guide to industrial control systems (ICS) security. *National Institute of Standards and Technology*, 2014. 58
- [292] Stéphane Raimbault. libmodbus: Open source library. <http://libmodbus.org/>. Retrieved: June, 2013. 159

- [293] O. Taha and M. Rashid Khan. Advanced process control for clean fuel production: smart plant of the future. In *Advances in Clean Hydrocarbon Fuel Processing*, Woodhead Publishing Series in Energy, pages 496–508. Woodhead Publishing, 2011. xi, 10
- [294] Talos. New VPNFilter malware targets at least 500K networking devices worldwide, 2018. <https://blog.talosintelligence.com/2018/05/VPNFilter.html>, 2018. Retrieved: Nov, 2019. 30
- [295] DVWA team. Damn Vulnerable Web Application (DVWA). <https://github.com/ethicalhack3r/DVWA>, 2010. Retrieved: Nov, 2019. 36
- [296] ICSMASTER Security Team. Industrial Security Exploitation Framework. <https://github.com/w3h/isf>, 2017. Retrieved: Nov, 2019. 34, 122
- [297] National Telecommunications and Information Administration. Software Bill of Materials. <https://ntia.gov/SBOM>, 2020. Retrieved: June, 2022. 156
- [298] The Dow Chemical Company. Product safety assessment: Vinyl acetate. <http://www.dow.com/productsafety/finder/vinyl.htm>. Retrieved: June, 2014. 43
- [299] The European Network of Transmission System Operators for Electricity. Continental Europe Significant Frequency Deviations - January 2019. Technical report, ENTSO-E, 2019. xii, 66, 67
- [300] Thomas Brewster. Hundreds Of Wind Turbines And Solar Systems Wide Open To Easy Exploits. <https://www.forbes.com/sites/thomasbrewster/2015/06/12/hacking-wind-solar-systems-is-easy>, 2015. Retrieved: Feb, 2015. 121
- [301] ThreatGEN. ThreatGEN: Cybersecurity Gamification Platform. <https://threatgen.com/>. Retrieved: Nov, 2019. 57
- [302] Aleksandr Timorin. SCADA tools. <https://github.com/atimorin/scada-tools>, 2014. Retrieved: Feb, 2020. 34
- [303] Michael Toecker. Didgital Maintenance and Test Equipment and Impact on Control Systems Security. <https://www.slideshare.net/MichaelToecker/maintenance-and-test-equipment-cyber-security>, 2015. Retrieved: March, 2016. 127
- [304] Virus Total. Virus Total online service. www.virustotal.com. Retrieved: Nov, 2019. 33, 122
- [305] Yazhou Tu, Sara Rampazzi, Bin Hao, Angel Rodriguez, Kevin Fu, and Xiali Hei. Trick or Heat? Manipulating Critical Temperature-Based Control Systems Using Rectification Attacks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS '19*, page 2301–2315. Association for Computing Machinery, 2019. 92
- [306] George Tzagkarakis, Grigorios Tsagkatakis, Daniel Alonso, Eugenio Celada, Cesar Asensio, Athanasia Panousopoulou, Panagiotis Tsakalides, and Baltasar Beferull-Lozano. *Signal and Data Processing Techniques for Industrial Cyber-Physical Systems*, pages 181–226. 10 2015. 24
- [307] NPC Ukrenergo. Press tour on substation 330 kV "Severnaya". <https://www.youtube.com/watch?v=AUoiKZBqIo0>, 2015. Retrieved: Feb, 2020. 122

- [308] David Urbina, Jairo Giraldo, Alvaro Cárdenas, Junia Valente, Mustafa Faisal, Nils Tippenhauer, Justin Ruths, Richard Candell, and Henrik Sandberg. Survey and New Directions for Physics-Based Attack Detection in Control Systems. Technical report, NIST, 2016. 156
- [309] David Urbina, Jairo Giraldo, Nils Ole Tippenhauer, and Alvaro Cárdenas. Attacking Fieldbus Communications in ICS: Applications to the SWaT Testbed. In *Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016*, pages 75–88, 01 2016. 61
- [310] David Urbina, Jairo Giraldo, Alvaro Cárdenas, Junia Valente, Mustafa Faisal, Justin Ruths, Niles O. Tippenhauer, Henrik Sandberg, and Richard Candell. A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. *ACM Comput. Surv.*, 51(4):1–36, 2018. 55
- [311] Marielba Urdaneta, Antoine Lemay, Nicolas Saunier, and Jose Fernandez. A Cyber-Physical Testbed for Measuring the Impacts of Cyber Attacks on Urban Road Networks. In *Critical Infrastructure Protection XII*, pages 177–196, 2018. 155
- [312] Arturo Urquizo. PID controller overview. https://commons.wikimedia.org/wiki/File:PID_en.svg, 2011. Retrieved: Nov, 2019. 9
- [313] Verizon. Data breach digest. Scenarios from the field. https://maritimecyprus.com/wp-content/uploads/2016/03/verizon_data-breach-digest_en-1.pdf, 2016. Retrieved: Dec, 2017. 30, 163
- [314] A. Vodencarevic, H. Kleine Buning, O. Niggemann, and A. Maier. Identifying behavior models for process plants. In *Emerging Technologies Factory Automation, 2011 IEEE 16th Conference on*, pages 1–8, 2011. 94
- [315] Wei Wang, Antonio Cammi, Francesco Di Maio, Stefano Lorenzi, and Enrico Zio. A Monte Carlo-based exploration framework for identifying components vulnerable to cyber threats in nuclear power plants. *Reliability Engineering System Safety*, 175:24 – 37", 2018. 50, 87
- [316] Wei Wang, Francesco Di Maio, and Enrico Zio. Adversarial Risk Analysis to Allocate Optimal Defense Resources for Protecting Cyber-Physical Systems from Cyber Attacks. *Risk Analysis*, 39(12):2766–2785, 2019. 50, 51
- [317] Yong Wang, Zhaoyan Xu, Jialong Zhang, Lei Xu, Haopei Wang, and Guofei Gu. SRID: State Relation Based Intrusion Detection for False Data Injection Attacks in SCADA. In *Computer Security – ESORICS 2014*, volume 8713 of *LNCS*, pages 401–418. 2014. 113
- [318] Joe Weiss. Marina Krotofil’s presentation on how to hack a chemical plant and it’s implication to actual issues at a nuclear plant. <https://www.controlglobal.com/blogs/unfettered/marina-krotofil-presentation-on-how-to-hack-a-chemical-plant-and-its-implication-to-actual-issues-at-a-nuclear-plant>, 2015. Retrieved: Nov, 2019. 169
- [319] Theodore J. Williams. The Purdue Enterprise Reference Architecture. *Computers in Industry*, 24(2–3):141–158, Sep 1994. ISSN 0166-3615. 11
- [320] Alexander Winnicki, Marina Krotofil, and Dieter Gollmann. Cyber-Physical System Discovery: Reverse Engineering Physical Processes. In *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, CPSS ’17, pages 3–14, 2017. ISBN 978-1-4503-4956-7. xiii, 85, 124, 125, 126

- [321] Dazhong Wu, David W. Rosen, Lihui Wang, and Dirk Schaefer. Cloud-Based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation. *Computer-Aided Design*, 59:1–14, 2015. 21
- [322] L. Xiao, D. Xu, N. B. Mandayam, and H. V. Poor. Attacker-Centric View of a Detection Game against Advanced Persistent Threats. *IEEE Transactions on Mobile Computing*, 17(11): 2512–2523, Nov 2018. 50
- [323] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. Characterizing industrial control system devices on the Internet. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–10, 2016. 120
- [324] Muhammad Mudassar Yamin, Basel Katt, and Vasileios Gkioulos. Cyber ranges and security testbeds: Scenarios, Functions, Tools and Architecture. *Computers & Security*, 88:101636, 2020. 57
- [325] Ömer Yüksel, Jerry den Hartog, and Sandro Etalle. Reading between the Fields: Practical, Effective Intrusion Detection for Industrial Control Systems. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing, SAC '16*, page 2063–2070, 2016. 54
- [326] Daniel Zafra. Fantastic Information and Where to Find It: A Guidebook to Open-Source OT Reconnaissance. <https://www.virusbulletin.com/uploads/pdf/magazine/2019/VB2019-Kapellmann-Zafra.pdf>, 2019. Retrieved: Dec, 2019. 122
- [327] M. Zeller. Myth or reality - does the aurora vulnerability pose a risk to my generator? In *Protective Relay Engineers, 2011 64th Annual Conference for*, pages 130 –136, 2011. 27, 117
- [328] Kim Zetter. An Easy Way for Hackers to Remotely Burn Industrial Motors (2016). <https://www.wired.com/2016/01/an-easy-way-for-hackers-to-remotely-burn-industrial-motors>, 2016. Retrieved: Nov, 2019. 50
- [329] Kim Zetter. Analysis of the Cyber Attack on the Ukrainian Power Grid. https://www.vice.com/en_us/article/bmvkn4/ukrainian-power-station-hacking-december-2016-report, 2017. Retrieved: Nov, 2019. 31
- [330] Kim Zetter. Revealed: How a secret Dutch mole aided the U.S.-Israeli Stuxnet cyberattack on Iran. <https://news.yahoo.com/revealed-how-a-secret-dutch-mole-aided-the-us-israeli-stuxnet-cyber-attack-on-iran-160026018.html>, 2019. Retrieved: Feb, 2020. 30
- [331] Isaac A. Zlochower and Gregory M. Green. The limiting oxygen concentration and flammability limits of gases and gas mixtures. *Journal of Loss Prevention in the Process Industries*, 22 (4):499–505, 2009. 45
- [332] Michael Zuba, Zhijie Shi, Zheng Peng, and Jun-Hong Cui. Launching Denial-of-Service Jamming Attacks in Underwater Sensor Networks. In *Proceedings of the Sixth ACM International Workshop on Underwater Networks, WUWNet '11*, 2011. 87

