Spring 2018

An approach for formal analysis of the security of a water treatment testbed

Sai Sidharth Patlolla

Department:

AN APPROACH FOR FORMAL ANALYSIS OF THE SECURITY OF A WATER TREATMENT TESTBED

by

SAI SIDHARTH PATLOLLA

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

COMPUTER SCIENCE

2018

Approved by

Dr. Bruce McMillin, Advisor
Dr. Sanjay Madria
Dr. Jennifer Leopold

ABSTRACT

This thesis focuses on securing critical infrastructures such as chemical plants, manufacturing units, and power generating plants against attacks that disrupt the information flow from one component to another. Such systems are controlled by an Industrial Control System (ICS) that includes controllers communicating with each other, and with physical sensors and actuators, using a communications network.

Traditional security models partition the security universe into two worlds, secure and insecure, but in the real world the partitions often overlap and information is leaked even through the physical observation which makes it much harder to analyze a Cyber physical system (CPS). To overcome these, this thesis focus on the Multiple Security Domain Nondeducibility (MSDND) model to identify the vulnerable points of attack on the system that hide critical information as in the STUXNET virus rather than theft of information. It is shown how MSDND analysis, conducted on a realistic multi-stage water treatment testbed, is useful in enhancing the security of a water treatment plant. Based on the MSDND analysis, this thesis offers a thorough documentation on the vulnerable points of attack, invariants used for removing the vulnerabilities, and suggested design decisions that help in developing invariants.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Bruce McMillin for the continuous support of my master's study and research and for his immense knowledge, motivation, enthusiasm, and patience. His guidance helped me through all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master's study. Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Sanjay Madria and Dr. Jennifer Leopold, for their insightful comments and encouragement and for the inputs they provided that helped me widen my research from various perspectives.

I would like to thank my fellow research students Anusha Thudimilla, Prakash Rao Dunaka, and Uday Ganesh Kethineni for their input, feedback, and cooperation. Without their passionate participation and input, this project could not have been successfully completed. In addition, I would like to express my gratitude to some of the staff of the Computer Science department Dawn Davis for responding to all my queries. Also, I would like to thank my friends for accepting nothing less than excellence from me. Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study. Thank you.

Very special gratitude goes out to National Institute of Standards and Technology, grant number 60NANB15D236 and with support from the Missouri S&T Intelligent Systems Center and the U.S. National Science Foundation, award number CNS-1505610 for helping and providing the funding for the work.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

NOMENCLATURE

	Description
AIT	Analyzer Indicating Transmitter
BIT	Belief, Information transfer and Trust
CPS	Cyber Physical Systems
$DPIT$	Differential Pressure Indication Transmitter
ICS	Industrial Control System
$LIT101$	Level Indicating Transmitter in Process 1
LS	Level Switch
$MV101$	Motorized Valve 101 in Process 1
$P1$	Process 1
$P101$	Pump 101 in Process 1
PIT	Pressure Indicating Transmitter
PLC	Programmable Logic Controller
$SCADA$	Supervisory Control And Data Acquisition
$T101$	Tank 101 in Process 1
	Greek
$\oplus$	XOR

ϕ	A Question that should be evaluated
	Subscripts
$P1\_SD^i$	Security Domain i in Process 1
s_x	State x
SD^i	Security Domain
$V_x^i(w)$	Valuation Function to know the status of x from SD^i, this assigns a truth value

1. INTRODUCTION

Cyber Physical Systems [Dunaka and McMillin, 2017], such as water treatment and power systems, are the pillars of sustainability for any working community. These systems are a combination of a control system and instrumentation used for process control. In general these systems are implemented by Supervisory Control And Data Acquisition (SCADA) or distributed control system (DCS) and programmable logic controllers (PLC). These systems are used in chemical plants, manufacturing units, power generation, oil and gas systems. These systems are also called Industrial Control Systems.

An ICS [Stouffer *et al.*, 2011] consists of physical, control, and network devices (Figure 1.1). In general, control devices are PLCs. The PLCs in an ICS can be viewed collectively as a distributed control system that transforms the state of the process through the use of sensors and actuators. The state of the physical process is collected by sensors and sends to controllers through a communication channel. A controller calculates the control command based on the control logic in controller and sends it to actuators, which eventually change the physical process. There could be more than one stage in a single ICS plant and each stage controlled by its own PLC. It is a distributed control system. Different controllers communicate through a network in order to know the state of the other parts of the system. SCADA, HMI, engineering workstations, and a historian are connected to the network for monitoring purposes.

A successful cyber attack [ics, 2016] on such plants could de-stabilize an entire community. Recent increase in successful cyber physical attacks on public infrastructure [Cobb, 2016, LIPOVSKY, 2016, Weinberger, 2011], and other mostly unsuccessful attempts [ics, 2016], have raised the importance of security analysis of an industrial CPS. There exists a variety of commercial products available for cyber attack prevention and detection that include firewalls and intrusion detection systems. However, attackers are often bypassing

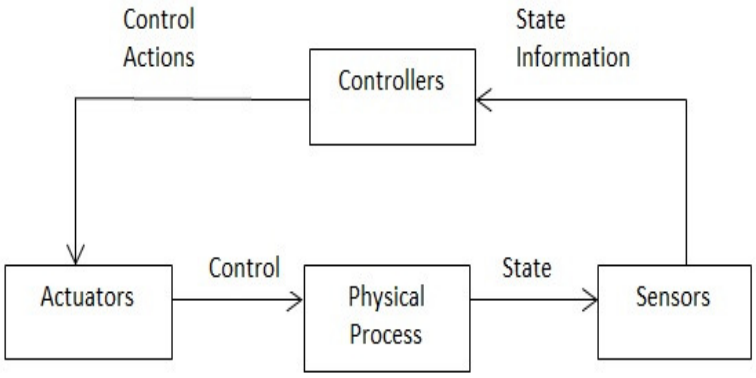

Figure 1.1. High level view of a Cyber Physical System (CPS).

these defense mechanisms by exploiting software and hardware vulnerabilities or through social engineering. It therefore becomes important to look for ways of detecting process anomalies in an ICS caused by an attacker who has gained unauthorized entry.

A recent survey on industrial SCADA systems shows that attackers are trying to attack 20% of SCADA computers [Adepu and Mathur, 2016b, securelist.com, 2017]. The threat landscape is shown in Figure 1.2 [securelist.com, 2017]. Hence, it is critical to analyze the vulnerability of such plants and recommend actions to improve the plant design. This work proposes, and evaluates, an approach for such analysis. The entire work reported here was conducted on a water treatment testbed which serves as a miniature version of a real water treatment plant to perform experiments and improve security.

A typical water treatment plant consists of multiple stages. Each stage in the treatment process either removes impurities from the incoming water or adds chemicals to prepare for the next stage. This study was conducted on a water treatment plant testbed consisting of six stages for water purification. Each stage has several sensors, actuators and PLCs that communicate with each other, and with other PLCs, in different stages to make the system work efficiently and effectively. Overall, there are 42 sensors and actuators distributed across different stages of the plant.

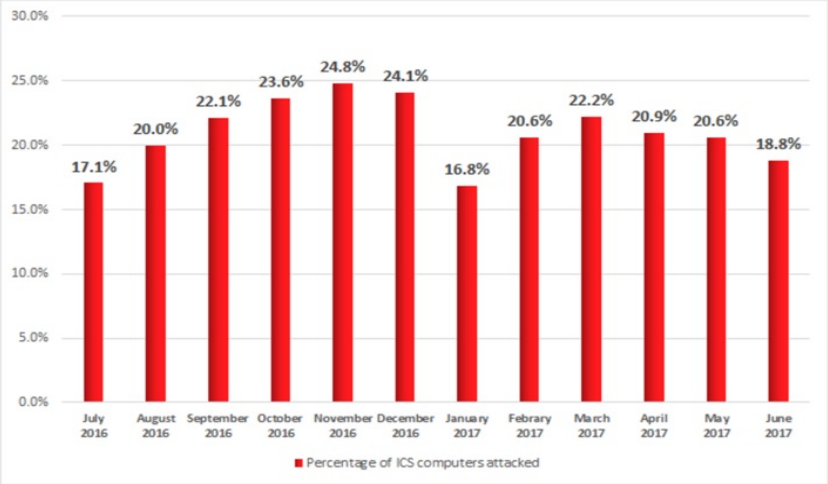

Figure 1.2. Percentages of ICS systems attacked.

In any CPS there is information flow across its components for the controllers to coordinate and perform the intended tasks. This flow of information can be disrupted by sending false values to other components. Here, "false value" refers not only to the manipulation or stealing of information but simply hiding of critical information. Such attacks on the information flow cannot be identified unless there is an independent way to derive the true state of the component. This work examines the general security attributes related to each component of the system, each with their own security domain, using Multiple Security Domain Nondeducibility (MSDND) [Howser and McMillin, 2014] models and Belief, Information transfer and Trust (BIT)[Liau, 2003][Liau, 2005] logic.

2. SYSTEM MODEL

The system consists of six stages (Figure 2.1), also referred to as processes [Mathur and Tippenhauer, 2016]. There is an operational PLC and a backup PLC at each stage that controls the flow of water and water purification. The primary function of Process 1 is to keep the water always available in the tank for use in subsequent processes. This is done with the help of a motorized valve which opens to let the water in and a pump to send water to the next process. Water from Process 1 is fed through a chemical dosing system at Process 2 which doses chemicals for maintaining the pH and oxidation reduction potential (ORP: a measurement that indicates the degree to which a substance is capable of oxidizing or reducing another substance) and conductivity of water. Process 3 contains an Ultrafltration (UF) unit. Here water is sent through the UF membranes to remove micrometer sized impurities. The output of UF is passed through Process 4, an ultraviolet chlorine destruction Unit which removes free chlorine from the water; this removal is necessary before Process 5 (reverse osmosis process) as the free chlorine present in the water could damage the RO membranes. In addition to removing free chlorine, Sodium bisulphate ($NaHSO_3$) is added to the water, when necessary, to control the ORP. Lastly, the water from the RO unit has two paths to go to Process 6. Pure water is sent to the RO permeate treatment system and impure water, also known as reject, is sent to the UF backwash system. The water from the UF backwash system is used for cleaning the UF membranes every 30 minutes or when the differential pressure across the UF membrane is greater than a preset. In the RO-CIP system, the water is a mixture of water from RO permeate and NaOCl from Process 2.

A cyber physical attacker model [Adepu and Mathur, 2016a] for industrial control systems such as this, consists of intentions. Attacker intentions include component damage, changing properties of the system, and performance degradation. Several attacks have been

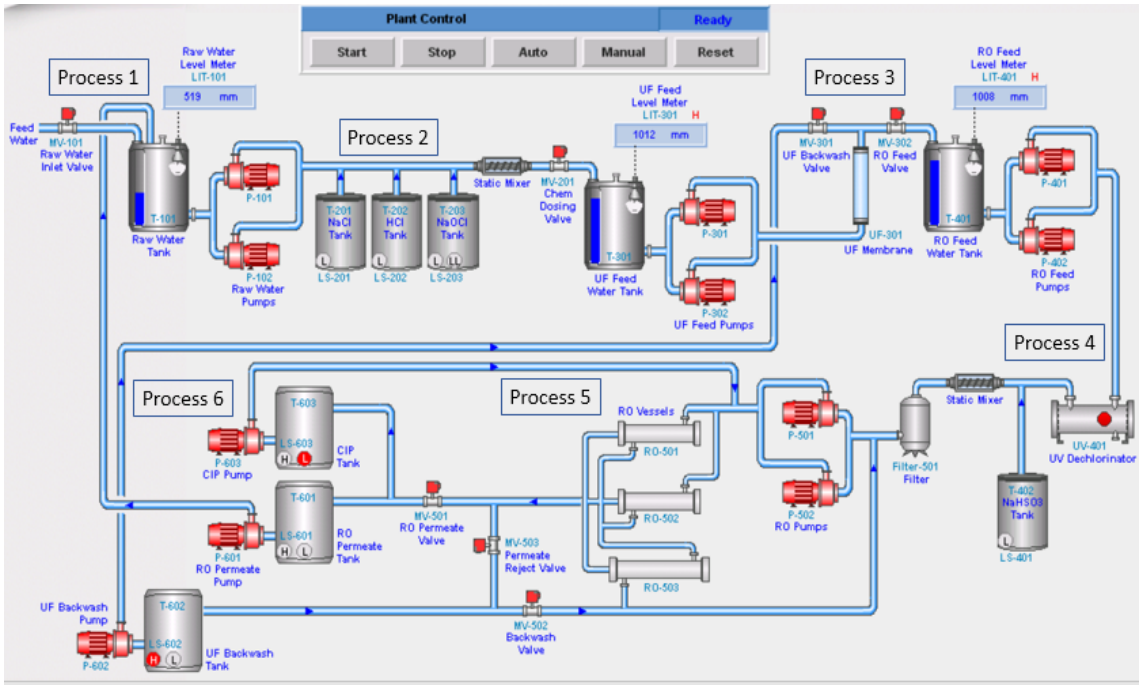

Figure 2.1. Architecture of the testbed.

launched on different sensor measurements and actuator measurements. Some specific scenarios were designed and attacks were launched on a real-time operational water treatment [Mathur and Tippenhauer, 2016] plant called Secure Water Treatment Testbed.

It is assumed that the attacker has the capability to enter into the system using vulnerabilities in the system and through social engineering. The attacker is capable of performing attacks such as STUXNET [Weinberger, 2011] and Ukraine power blackout [LIPOVSKY, 2016]. This work does not focus on how the attacker is entering into the system. An attacker has access to communication channels in the plant network. In general attacker has ability to modify the network packets in the communication channels.

PLCs use sensors such as flow indication transmitters (FIT) and level indication transmitters (LIT). These sensors are located across all the processes to monitor the water flow. Actuators such as motorized valves (MV) and pumps (P), are used to control the flow of water across processes. These sensors, actuators and PLCs are the most vulnerable points for cyber, as well as physical, attacks which hide critical information flowing between

either a sensor and PLC, or between a PLC and an actuator. It is easy for one component to believe in the truth value of information coming from the other. For example, an intruder can attack a component such as sensor or actuator and always send incorrect values to the PLC regardless of the actual values. To avoid such an attack, the MSDND model is used to reveal where the vulnerabilities lie.

3. PROBLEM STATEMENT

This work models Stuxnet-like [Chen, 2010] attacks on the water treatment system using MSDND to locate points of vulnerability. The focus of such attacks is to hide critical information rather than steal it. Once into the system; viruses that aim at hiding information stay dormant and learn the behavior of the system before corrupting the information. There are two basic ways to hide this information: make it impossible to evaluate the desired question, say ϕ, or to disrupt the actual valuation function to return an unreliable valuation of the question ϕ. It is *bad* for the system if it is MSDND secure with respect to integrity since by the definition of MSDND the observer does not have valuation functions for the states of the system, i.e. one cannot determine the truth value of a system state. However, it is *good* for the system with respect to confidentiality because an observer will not be able to know changes made to the system. Thus, given a system the problem is to identify all such "good" and "bad" paths. This paper proposes the use of MSDND as an approach to solve this problem and make design recommendations.

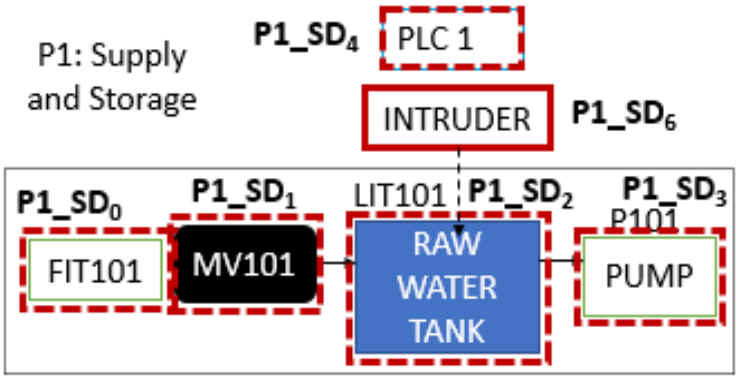

Figure 3.1. Process 1 with Security Domains.

3.1. ATTACKS

Process 1 (P1) as shown in Figure 3.1 consists of MV101 (motorized valve), raw water tank and P101 (pump) which serve the purpose of pumping the water to the next stages of the water system. PLC 1 makes sure that there is enough water in the tank at any time to be pumped to next stage by looking at the level indication transmitter (LIT) sensor which is mounted over raw water tank T101. When it senses the water level is L (low) or L (very low) it opens valve MV101 to let water into the tank. Similarly, if the water level is H (high) or HH (very high), PLC 1 will turn on pump P101 to pump water from the tank thus not draining or overflowing the tank.

Consider an attack on LIT101 of P1 where the goal of the attacker is to overflow tank T101. The intruder always injects a lesser value to LIT101 irrespective of its real value expecting that the PLC 1 will open MV101 to fill the tank and eventually causing an overflow. To apply the MSDND security model, initially, the set of all components are partitioned into security domains as shown in Figure 3.1. Next the information flow paths for each security domain are checked against MSDND for security. Formally, this process is illustrated in Section 5 using BIT logic.

3.2. ATTACK TOOL

Researchers have developed a tool named SWaTAssault [Urbina *et al.*, 2016] to aid in launching attacks on the water treatment system. This tool enables the launch of various types of attacks such as MITM, command injection, and stealthy replay attacks on Level 0 and Level 1 networks. The launch is carried out by programmatically overriding and manipulating packets between PLCs, sensors, and actuators. Figure 3.2 shows the dashboard screen shot of the implemented invariants with their violation status when an attack happens.

Level 0 is the communication channel between Remote Input Output (RIO) and PLC. RIO receives the sensors signals from the physical process. Level 1 is the communication channel among different PLCs and engineering workstation.

3.3. PROVERIF

ProVerif is a tool for automatically analyzing the security of cryptographic protocols. Support is provided for, but not limited to, cryptographic primitives including: symmetric and asymmetric encryption; digital signatures; hash functions; bit-commitment; and non-interactive zero-knowledge proofs. ProVerif is capable of proving reachability properties, correspondence assertions, and observational equivalence. These capabilities are particularly useful to the computer security domain since they permit the analysis of secrecy and authentication properties. Moreover, emerging properties such as privacy, traceability, and verifiability can also be considered. Protocol analysis is considered with respect to an unbounded number of sessions and an unbounded message space. The tool is also capable of attack reconstruction: when a property cannot be proved, ProVerif tries to reconstruct an execution trace that falsies the desired property.

The primary goal of ProVerif is the verification of cryptographic protocols. Cryptographic protocols are concurrent programs which interact using public communication channels such as the Internet to achieve some security-related objective. These channels are assumed to be controlled by a very powerful environment which captures an attacker with "Dolev-Yao" capabilities. Since the attacker has complete control of the communication channels, the attacker may: read, modify, delete, and inject messages. The attacker is also able to manipulate data, for example: compute the i^{th} element of a tuple; and decrypt messages if it has the necessary keys. The environment also captures the behavior of dishonest participants; it follows that only honest participants need to be modeled. ProVerif's

Figure 3.2. Implemented Invariants of Process 1 with Violation Status.

input language allows such cryptographic protocols and associated security objectives to be encoded in a formal manner, allowing ProVerif to automatically verify claimed security properties.

In this thesis, Bruno Blanchet's ProVerif [Blanchet, 2008] automates the MSDND process for water treatment testbed using observational equivalence and integrity properties, in addition to that it verifies the correctness of the system with the help of proofs. The ProVerif proofs are presented in Section 5

4. RELATED WORK

This paper focuses primarily on information flow disruption rather than theft of information. Research related to this aspect of cyber security is summarized below.

Challenges in addressing safety and security against cyber attacks are addressed in [Cardenas *et al.*, 2008]. Lee [Lee, 2008] presented cyber physical systems from an embedded systems point of view and described the problems in computing and networking for the design of CPS. Humayed et al. [Humayed *et al.*, 2017] surveyed literature on cyber physical systems security, and presented an orthogonal framework that consists of security, components, and system perspectives. They focused mainly on four CPS systems such as ICS, smart grids, medical devices, and smart cars. This paper presents threats, known attacks, vulnerabilities and security aspects to those vulnerabilities.

"Researchers model cyber attacks on cyber physical systems in different ways: [Cárdenas *et al.*, 2011] modeled deception attacks that include surge, bias and geometry. In [Kwon *et al.*, 2013], attacks have been modeled as noise in sensor data. The work described in [Gao *et al.*, 2010], focuses on the impact of cyber attacks on water treatment behavior and procedures to launch the attack. This paper also includes a neural network based system to detect anomaly detection due to exploits on modbus. The remainder of the related work is closely related to the current manuscript. Investigation of cyber attacks [Adepu and Mathur, 2016b] on a water treatment system was studied, this works considered impacts in three different domains: 1) impact on components of water system, 2) impact on water properties such as pH, ORP and conductivity, and 3) impact on water system performance. A complementary approach to this current manuscript is based on learning is proposed by [Krotofil *et al.*, 2015] to determine anomalous behavior within a plant, which considering

information flow as a key element." This content is contributed by Adepu Sridhar and Dr. Aditya Mathur, who were the coauthors for DSN conference paper with myself as the main author.

4.1. NONDEDUCIBILITY(ND) [Sutherland, 1986]

Nondeducibility was introduced by Sutherland in an attempt to model information flow in a partitioned model. The partitions are divided into two sets, these sets are usually labeled as high and low with information restricted to one side of the partition or the other. Information that cannot be deduced from the other side of the partition is said to be nondeducibility secure. However, the partitions must be absolute and the partition is necessarily simplistic. Overlapping security domains present difficulties for ND as do information flows which cannot be evaluated because the model lacks the required valuation functions. However the restrictions of Sutherland's ND model made it difficult to model critical infrastructures like ICS, transportation systems etc. The motivation to model security for these critical infrastructures and to have much more refined control over the information being transferred and to deal with multiple physical and cyber components at a time led to the development of the Multiple Security Domain Nondeducibility model.

4.2. MULTIPLE SECURITY DOMAIN NONDEDUCIBILITY

[Howser and McMillin, 2013b] Critical infrastructures have complex interaction between the physical and cyber components, when such a system is divided into security domains, the domains often overlap or a security domain is entirely contained in another which makes it harder for a traditional security model to capture the information flow among different components that might lead to a vulnerability. To overcome these limitations present in the traditional models, MSDND security model is used. MSDND is not a high/low hierarchy model, but is instead a partitioning model. MSDND does not depend upon

examining two domains on any relationship between those domains such as low and high or left and right. The domains in question might be wholly contained in the other, they might overlap, or they might be disjoint.

There exists some world with a pair of states $s_x \wedge s_y$ where one must be true and the other false (exclusive OR), but an entity i has no valuation function for those states. In security domain SD^i, i simply cannot know which state is true and which is false.

MSDND(ES) = $\exists w \in W \vdash [(s_x \vee s_y)] \wedge \sim(s_x \wedge s_y) \wedge [w \models (\nexists V_x^i(w) \wedge \nexists V_y^i(w))]$

An equivalent formula is

MSDND(ES) = $\exists w \in W \vdash [(s_x \oplus s_y)] \wedge [w \models (\nexists V_x^i(w) \wedge \nexists V_y^i(w))]$

If a security domain is MSDND secure then it is bad, as an observer cannot evaluate the status of that security domain. Similarly, if a security domain is Not MSDND secure then it is good for the system. These statements hold when we check the integrity of the system.

4.3. VALUATION FUNCTION

$V_x^y(\phi)$ represents valuation function of boolean x in domain y. A valuation function is a function which assigns a truth value to question ϕ in state x with respect to the security domain y.

4.4. SECURITY DOMAIN (SD^I) [Howser and McMillin, 2013a]

The event system divides the system into multiple security domains SD^i as viewed by each entity i in the model. These security domains may or may not overlap with each other. An entity i is any part of the system that is capable of independent observation or action. Security domains of Process 1 in the water system are shown in Figure 3.1 with rectangular boxes with dotted lines. Each security domain overlaps with security domain of another component at the PLC, as PLC controls the information flow among all the components.

4.5. BIT LOGIC

BIT logic was introduced by Liau [Liau, 2003][Liau, 2005] to formally reason about belief, information transfer and trust when dealing with cyber entities. While it was developed primarily for handling trust in database and distributed systems, BIT logic is useful for describing CPS, especially when humans are involved. Before BIT logic, social engineering attacks could only be described by a narrative in imprecise language. With BIT logic, spoofing and other unwanted behavior is described with simple, formal proofs. BIT logic is designed to reason about the belief and trust an entity i has in information from an entity j, e.g. the belief and trust an operator has in the reading from a monitoring station.

- $T_{i,j}\ \phi$, defines the trust i has in a report from j that ϕ is true.
  - In the proofs presented in Section 5 we use a similar notation, for example, $T_{6,2}LIT101$; to indicate that the security domain 6 trusts the LIT101 value sent by the security domain 2.
- $B_i\ \phi$, defines the belief by i that ϕ is true; it does not matter if ϕ is true or not, i believes it to be true.
  - Example, $B_6 I_{6,2} LIT101$; indicates that the security domain 6 believes LIT101 value that it received from security domain 2 is true.
- $I_{i,j}\ \phi$, defines the transfer of information directly from one agent to another, that is j reported to i that ϕ is true.
  - Example, $I_{6,2}LIT101$; This simply means that information regarding LIT101 is sent to security domain 6 from security domain 2.

BIT logic is used in the further sections to clearly specify the information transfer between components, demonstrate how an intruder sitting in between the components get access to the information and finally show how the PLC is made to believe manipulated information.

4.6. INVARIANTS

An *invariant* is a property that remains unchanged when a specified transformation is applied. An invariant is experssed as a logical predicate on a system state. Invariant coded thus must not change its truth value during plant operation. An axiomatic basis for the truth of invariants on cyber physical systems was first proposed in [Owicki and Gries, 1976]. The invariants that are considered in this paper are derived by considering the physical properties of a process and from [Mathur and Tippenhauer, 2016].

4.7. EXECUTION MONITORS

Some research is being done in implementing execution monitors such as the Shadow Security Unit (SSU [Cruz *et al.*, 2015]) in ICS. The SSU is attached in parallel to Remote terminal units (RTUs) or PLCs, being able to capture and decode the SCADA protocol information flow, correlating this information with the status of the physical I/O modules that interface with sensors and actuators on the field. This enables the possibility of implementing a redundant security-checking mechanism that follows a black box approach regarding the analysis of the monitored device behavior. Coupling MSDND and a few techniques from SSU along with the ground truths encapsulated as invariant equations can further reduce the bounds on parameters measured in a water treatment plant and also more accurately determine a corrupted information path. A ground truth refers to information provided by direct observation as opposed to information provided by inference. If the invariant is violated, the monitor raises an exception such as that shown in Figure 3.2.

5. WORKING OF MULTIPLE SECURITY DOMAIN NONDEDUCIBILITY

MSDND analysis can be done to various cyber-physical systems to identify if there is a vulnerability or not. Here, we use BIT (Belief, Information and Trust) logic to formally show the exchange of information between components and world states, when these put together in MSDND equation reveal if there exist a vulnerability. BIT logic is especially helpful in writing these proofs as this conveys the message without any ambiguity.

Below, two scenarios with respect to an attack on LIT101 (Figure 3.1) are presented which will help the reader understand the MSDND proofs using BIT Logic.

5.1. THE LEVEL OF THE WATER IN TANK $LIT101$ IN SECURITY DOMAIN $P1\_SD2$ IS MSDND SECURE WITHOUT INVARIANTS UNDER AN ATTACK

The level of the water in tank is normal implies LIT101 = true. During the attack phase, a virus in $P1\_SD6$ receives sensor reports and always reports to the PLC in $P1\_SD4$ an LIT value lesser than the actual value. Thus the virus has corrupted the information path between the sensor and the PLC.

1. LIT101 = true; level of the tank is normal
2. w $\models V_{LIT101}^{P1\_SD6}$(w) = true; the reading is normal and the valuation function in world w is true
3. $I_{6,2}$ LIT101; Sensor reports to virus
4. $B_6 I_{6,2}$ LIT101; Virus believes sensor report
5. $T_{6,2}$ LIT101; Virus trusts the sensors
6. $B_6 I_{6,2}$ LIT101 $\wedge$ $T_{6,2}$ LIT101 $\rightarrow$ B_6 LIT101; Virus believes the reading

7. $I_{4,6}$ ~LIT101; Virus always sends incorrect readings

8. $B_4 I_{4,6}$ ~LIT101; PLC believes incorrect readings

9. $T_{4,6}$ ~LIT101; PLC trusts reports

10. $B_2 I_{4,6}$ ~ LIT101 $\wedge$ $T_{5,6}$ ~LIT101 $\rightarrow$ B_5 ~LIT101; PLC believes readings are correct

11. w $\models V^{P1\_SD4}_{\sim LIT101}$(w) = true; $V^{P1\_SD4}_{\sim LIT101}$(w) always returns true

MSDND(ES) = $\exists$ w $\in$ W $\vdash$ [($S_{LIT101} \oplus S_{\sim LIT101}$] $\wedge$ [w $\models$ ($\nexists$ $V^{P1\_SD4}_{\sim LIT101}$(w) $\wedge$ $\nexists$ $V^{P1\_SD4}_{LIT101}$(w))]

Since $B_4 I_{4,6}$LIT101 $\wedge$ $T_{4,6}$LIT101 $\rightarrow$ B_4LIT101, the PLC believes the lie told in step 7 in all cases. Therefore, unknown to entities in $P1\_SD^4$, $V^{P1\_SD4}_{LIT101}$(w) and $V^{P1\_SD4}_{\sim LIT101}$(w) cannot be evaluated. Therefore LIT101 is MSDND secure from $P1\_SD^4$.

5.2. IN THE PRESENCE OF AN INVARIANT ON WATER FLOW, WE GET ANOTHER INFORMATION PATH TO KNOW THE STATUS OF SECURITY DOMAIN. AN INVARIANT USED HERE WILL VERIFY ITSELF WITH THE STATUS OF OTHER COMPONENTS

The level of water in the tank can be estimated using flow meters FIT101 and FIT201. These FITs measure the rate of flow of water into the tank and water leaving the tank. By subtracting outflow from the inflow and multiplying it with a constant, current estimate of LIT101 can be obtained. The invariant equation considered here is:

$$LIT\_Est : x(k+1) - x(k) = \alpha(u_i(k) - u_o(k)) \tag{5.1}$$

1. LIT101 = true; level of the tank is normal

2. w $\models V^{P1\_SD6}_{LIT101}$(w) = true; the reading is normal the valuation function in world w is true

3. $I_{6,2}$ LIT101; Sensor reports to virus

4. $B_6 I_{6,2}$ LIT101; Virus believes sensor report
5. $T_{6,2}$ LIT101; Virus trusts the sensors
6. $B_6 I_{6,2}$ LIT101 $\wedge$ $T_{6,2}$ LIT101 $\rightarrow$ B_6 LIT101; Virus believes the reading
7. $I_{4,6}$ ~LIT101; Virus always sends incorrect readings
8. $B_4 I_{4,6}$ ~LIT101; PLC believes interface report
9. $T_{4,6}$ ~LIT101; PLC trusts reports
10. $B_2 I_{4,6}$ ~ LIT101 $\wedge$ $T_{5,6}$ ~LIT101 $\rightarrow$ B_5 ~LIT101; PLC believes readings are correct
11. w $\models$ $V^{P1\_SD4}_{\sim LIT101}$(w) = true; $V^{P1\_SD4}_{\sim LIT101}$(w) always returns true
12. $\sim LIT101_{LIT\_Est} \implies$ ~LIT101; from assumption and invariant (5.1)
13. I_{4,LIT_Est}LIT101; PLC reads the invariant
14. $B_4 I_{4,LIT_Est}$LIT101; PLC believes the invariant
15. T_{4,LIT_Est}LIT101; PLC trusts the invariant
16. $B_4 I_{4,LIT_Est}$LIT101 $\wedge$ T_{4,LIT_Est}LIT101 $\rightarrow$ B_4LIT101; PLC believes readings are correct and normal
17. $S_{LIT\_Est} \wedge S_{LIT101} = S^*$; System is working normally if and if only this is true
18. w $\models$ $V^{P1\_SD4}_{LIT101}$ (w) = true

MSDND(ES) = $\exists$ w $\in$ W $\vdash$ [($S^* \oplus S_{\sim LIT101}$)] $\wedge$ [w $\models$ ($\exists$ V^{SD4}_{LIT101}(w))]

$V^{P1\_SD4}_{LIT101}$(w) exists: can be evaluated from the invariant, which contradicts the second part of MSDND definition.

Therefore the system is not MSDND secure, and a potential threat can be detected. This is good for the plant and bad for the attacker.

There are level indication transmitters in several processes of the water system, using LIT_Est invariant we can break the MSDND security. Invariants are present for the flow indication transmitter, pumps, and motorized Valves for all the processes. The BIT Logic for these components is similar hence they are tabulated only once Section 6.

The first line of Table 7.2 contains the result of implementing the LIT101 proof. Similarly, the remainder of Table 7.2 contains the results of testing the remaining processes [SUTD, 2016]. Table 7.1 summarizes the vulnerabilities in each of the processes and their mitigation.

5.3. PROVERIF CODE - WHEN THE INFORMATION PATH BETWEEN LIT101 AND PLC IS CORRUPTED

```
(* Creating a free channel for message passing between PLC and LIT101 *)
free c:channel.
(* Initializing messages that are needed to be passed*)
free LIT101:bitstring [private].
free Request:bitstring[private].
(* Querying to see if the attacker can get any of the messages*)
query attacker(LIT101).
(* Starting a Process *)
let PLC =
(* Sending Request on public channel c *)
out(c, Request);
(* Receiving the LIT101 value *)
in( c, LIT101_val:bitstring );
0.
(* Process ends with 0 and new Process begin *)
(* Process 1 at T101 *)
```

```
let T101 =
(* Receiving the Request for sending back LIT101 Value *)
in( c,PLC_Req:bitstring );
if PLC_Req = Request then
(* sending back LIT101 Value *)
out(c, LIT101).
(* This initiates the Verification *)
process ((!PLC) | (!T101))
```

Result:

– Query not attacker(LIT101[])

Completing...

Starting query not attacker(LIT101[])

goal reachable: attacker(LIT101[])

RESULT not attacker(LIT101[]) is false.

Since the LIT101 value is passed through a public channel c, the attacker can read and manipulate the value, hence, the final result obtained is not attacker is false that means it is insecure.

5.4. PROVERIF CODE - WHEN THE INFORMATION PATH BETWEEN LIT101 AND PLC IS CORRUPTED IN THE PRESENCE OF AN INVARIANT

```
(* Defining own types *)
type FIT101.
type P101.
type LIT101.
```

```
(* The private channels cannot be accessed by the attacker *)
free c:channel[private].
free ch:channel.
free mon:channel[private].
free lit101:LIT101 [private].
free fit101:FIT101 [private].
free p101:P101 [private].
free Request:bitstring[private].
(* Here is the function that returns the value predicted by the invariant *)
fun Inv(FIT101,P101,LIT101):LIT101.
(* Process 1 at PLC *)
let PLC =
out(c, Request);
in(ch, lit101_val:LIT101);
(* The choice keyword checks for the observational equivalence *)
out(mon, choice[lit101_val, Inv(fit101,p101,lit101)]);
0.
(* Process 2 at T101 *)
let T101 =
in( c, PLC_Req:bitstring );
out(ch, lit101);
0.
(* The result of observational equivalence is sent to Monitor *)
(* Process 3 at Monitor *)
let Monitor=
in(mon, lit_ind:LIT101);
0.
```

process ((!PLC) | (!T101))

Result:

– Observational equivalence

Termination warning: v_103 <> v_104 & & attacker2(v_102,v_103) & & attacker2 (v_102,v_104) -> bad

Selecting 0

Termination warning: v_106 <> v_107 & & attacker2(v_106,v_105) & & attacker2 (v_107,v_105) -> bad

Selecting 0

Completing...

Termination warning: v_103 <> v_104 & & attacker2(v_102,v_103) & & attacker2 (v_102,v_104) -> bad

Selecting 0

Termination warning: v_106 <> v_107 & & attacker2(v_106,v_105) & & attacker2 (v_107,v_105) -> bad

Selecting 0

RESULT Observational equivalence is true (bad not derivable).

As we are using Invariant to verify the value of LIT101, the result obtained is true, that means the attacker cannot distinguish the value of LIT101 with invariant value.

6. MSDND PROOFS

The proofs in this section are similar to the ones in Section 5. The proofs for components in other six processes of Secure Water Treatment Testbed (Figure 6.1-6.4) are Tabulated (Table 6.1-6.10) in this Section. Let us consider the first two entries of Table 6.1, which considers FIT101 (Figure 3.1) to explain the scenario of the attack and how the invariants used help break the MSDND.

The first column represents the actual value of the sensor or an actuator, here, the value of sensor FIT101 is 5.88. The second column mentions the change in the sensor value similar to what an attacker might do, in this case, the value of FIT101 is changed from 5.88 to 2.00. The third column shows an invariant, if an invariant exist. The fourth column provides the MSDND proofs similar to the one explained in Section 5. The fifth column justifies if it is MSDND secure or not by checking if there exist a valuation function or not, here, the value of FIT101 is MSDND secure. The last column mentions the impact on the water treatment plant caused by the attack.

In the second row, the invariant is included, which helps in finding the value of FIT101. The invariant included is shown in third column. In this case, the result obtained is Not MSDND secure, which is good for the system.

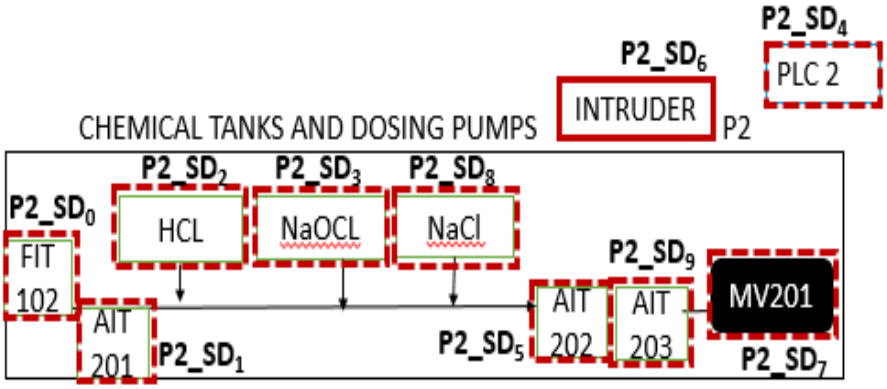

Figure 6.1. Process 2 with Security Domains.

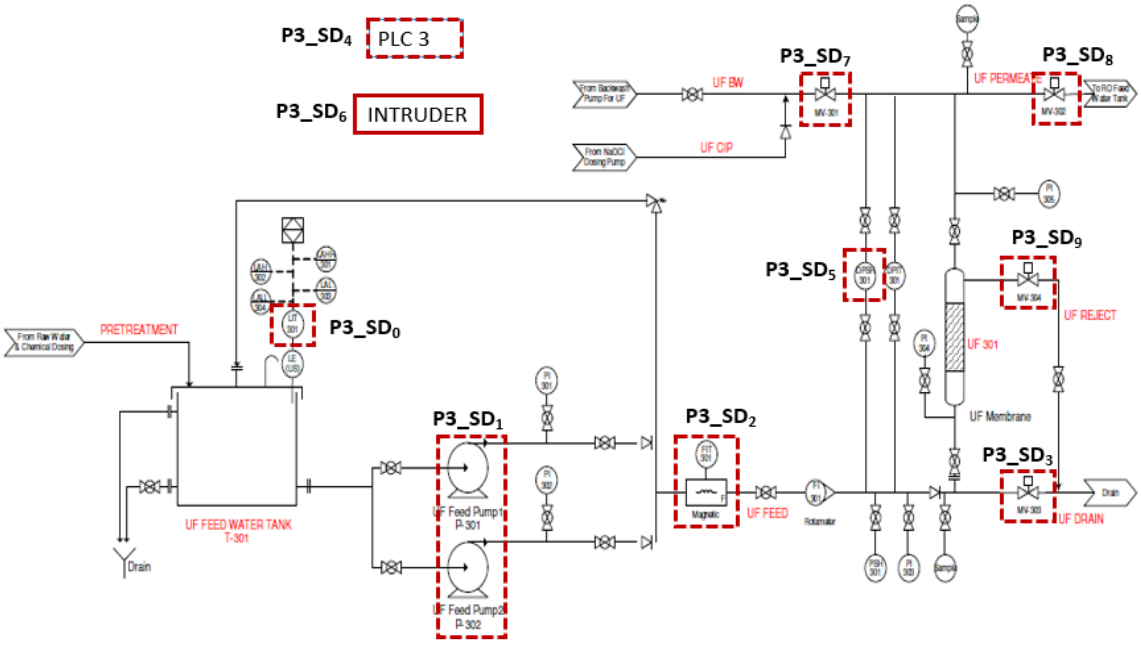

Figure 6.2. Process 3 with Security Domains.

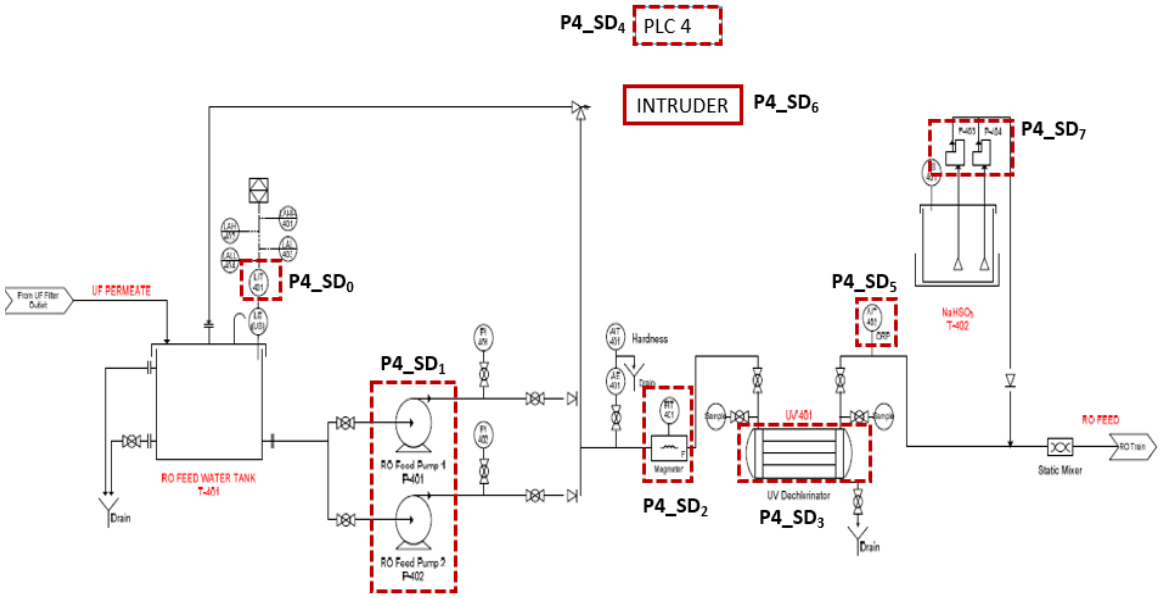

Figure 6.3. Process 4 with Security Domains.

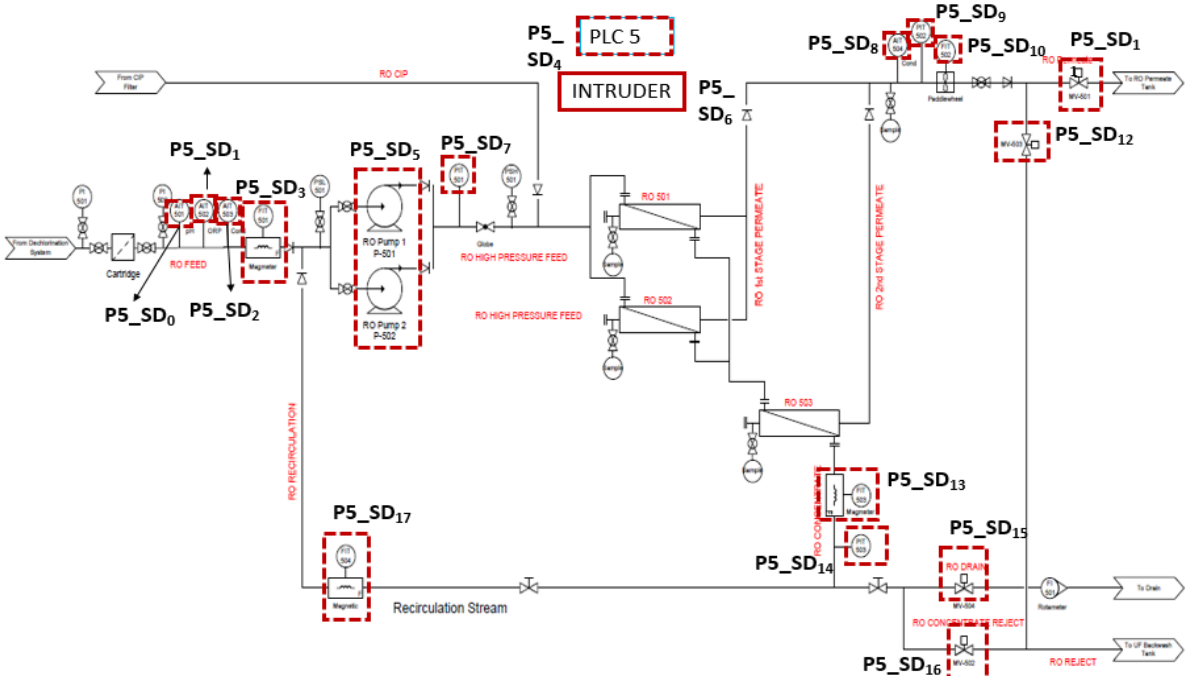

Figure 6.4. Process 5 with Security Domains.

Table 6.1. MSDND Proof for FIT101 in Process 1 (Figure 3.1)

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$FIT101$ = 5.88	$w \models V^{P1\_SD6}_{FIT101}(w)$ = $true$ $\sim (FIT101 = 5.88)$ $\Rightarrow FIT101 = 2.00$ $\equiv\sim FIT101$		$I_{6,0}FIT101$; $B_6 I_{6,0}FIT101$; $T_{6,0}FIT101$; $B_6 I_{6,0}FIT101 \wedge T_{6,0}FIT101 \rightarrow B_6 FIT101$; $w \models V^{P1\_SD6}_{FIT101}(w) = true$; $I_{4,6} \sim FIT101$; $B_4 I_{4,6} \sim FIT101$; $T_{4,6} \sim FIT101$; $B_4 I_{4,6} \sim FIT101 \wedge T_{4,6} \sim FIT101 \rightarrow B_4 \sim FIT101$; $w \models V^{P1\_SD4}_{\sim FIT101}(w) = true$;	$\exists w \in W \vdash$ $[(S_{FIT101} \oplus$ $S_{\sim FIT101})] \wedge$ $[w \models$ $(\nexists V^{P1\_SD4}_{\sim FIT101}(w))]$	$Sensor$ $Failure$
$FIT101$ = 5.88	$w \models V^{P1\_SD6}_{\sim FIT101}(w)$ = $true$	$P1\_INV1$: MV101- OPEN $\Rightarrow FIT101 > \delta$	$I_{4,6} \sim FIT101$; $B_4 I_{4,6} \sim FIT101$; $T_{4,6} \sim FIT101$; $B_4 I_{4,6} \sim FIT101 \wedge T_{4,6} FIT101 \rightarrow B_4 \sim FIT101$; $\sim P1\_INV1 \implies \sim FIT101$; $I_{4,P1\_INV1} FIT101$; $B_4 I_{4,P1\_INV1} FIT101$; $T_{4,P1\_INV1} FIT101$; $B_4 I_{4,P1\_INV1} FIT101 \wedge T_{4,P1\_INV1} FIT101 \rightarrow B_4 \sim FIT101$; $S_{P1\_INV1} \wedge S_{FIT101} = S^*$; $w \models V^{P1\_SD4}_{FIT101}(w) = true$	$\exists w \in W \vdash [(S^* \oplus$ $S_{\sim FIT101})] \wedge [w \models$ $(\exists V^{P1\_INV1}_{\sim FIT101}(w))]$	$Safe$

Table 6.2. MSDND Proof for P101 in Process 1 (Figure 3.1)

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$P101$ $= ON$	$w \models V_{P1\_SD1}^{P1\_SD6}(w)$ $= true$ $\sim (P101 = ON)$ $\Rightarrow P101 = OFF$ $\equiv\sim P101$		$I_{6,3}P101$; $B_6 I_{6,3}P101$; $T_{6,3}P101$; $B_6 I_{6,3}P101 \wedge T_{6,3}P101 \rightarrow B_6 P101$; $w \models V_{P101}^{P1\_SD6}(w) = true$; $I_{4,6} \sim P101$; $B_4 I_{4,6} \sim P101$; $T_{4,6} \sim P101$; $B_4 I_{4,6} \sim P101 \wedge T_{4,6} \sim P101 \rightarrow B_4 \sim P101$; $w \models V_{\sim P101}^{P1\_SD4}(w) = true$;	$\exists w \in W \vdash [(S_{P101} \oplus S_{\sim P101})] \wedge [w \models (\nexists V_{\sim P101}^{P1\_SD4}(w))]$	$Infrast-$ $ructure$ $Damage$
$P101$ $= ON$	$w \models V_{\sim P101}^{P1\_SD4}(w)$ $= true$	$P1\_INV3 :$ $P101 - On$ $\Rightarrow FIT201 > \delta$	$I_{4,6} \sim P101$; $B_4 I_{4,6} \sim P101$; $T_{4,6} \sim P101$; $B_4 I_{4,6} \sim P101 \wedge T_{4,P1\_INV3} \sim P101 \rightarrow B_4 \sim P101$; $\sim P1\_INV3 \implies \sim P101$; $I_{4,P1\_INV3} \sim P101$; $B_4 I_{4,P1\_INV3} \sim P101$; $T_{4,P1\_INV3} \sim P101$; $B_4 I_{4,P1\_INV3} \sim P101 \wedge T_{4,P1\_INV3} \sim P101 \rightarrow B_4 \sim P101$; $S_{P1\_INV3} \wedge S_{P101} = S^*$; $w \models V_{P101}^{P1\_SD4}(w) = true$	$\exists w \in W \vdash [(S^* \oplus S_{\sim P101})] \wedge [w \models (\exists V_{\sim P101}^{P1\_INV3}(w))]$	$Safe$

Table 6.3. MSDND Proof for AIT201 of Process 2 (Figure 6.1)

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$AIT201 = 265$	$w \models V^{P2\_SD6}_{AIT201}(w) = true$ $\sim (AIT201 = 265) \Rightarrow AIT201 = 165 \equiv\sim AIT201$		$I_{6,1}AIT201$; $B_6 I_{6,1} AIT201$; $T_{6,1} AIT201$; $B_6 I_{6,1} AIT201 \wedge T_{6,1} AIT201 \rightarrow B_6 AIT201$; $w \models V^{P2\_SD6}_{AIT201}(w) = true$; $I_{4,6} \sim AIT201$; $B_4 I_{4,6} \sim AIT201$; $T_{4,6} \sim AIT201$; $B_4 I_{4,6} \sim AIT201 \wedge T_{4,6} \sim AIT201 \rightarrow B_4 \sim AIT201$; $w \models V^{P2\_SD4}_{\sim AIT201}(w) = true$;	$\exists w \in W \vdash [(S_{AIT201} \oplus S_{\sim AIT201})] \wedge [w \models (\nexists V^{P2\_SD4}_{\sim AIT201}(w)]$	$Chemical\ Imbalance$
$AIT201 = 265$	$w \models V^{P2\_SD6}_{\sim AIT201}(w) = true$	$P2\_INV1 : AIT503 - High \Rightarrow P201, 2Off$	$I_{4,6} \sim AIT201$; $B_4 I_{4,6} \sim AIT201$; $T_{4,6} \sim AIT201$; $B_4 I_{4,6} \sim AIT201 \wedge T_{4,6} t \rightarrow B_4 \sim AIT201$; $\sim P2\_INV1 \implies \sim AIT201$; $I_{4,P2\_INV1} AIT201$; $B_4 I_{4,P2\_INV1} AIT201$; $T_{4,P2\_INV1} AIT201$; $B_4 I_{4,P2\_INV1} AIT201 \wedge T_{4,P2\_INV1} AIT201 \rightarrow B_4 \sim AIT201$; $S_{P2\_INV1} \wedge S_{AIT201} = S^*$; $w \models V^P 2\_SD4_{AIT201}(w) = true$	$\exists w \in W \vdash [(S_{AIT201} \oplus S_{\sim AIT201})] \wedge [w \models (\exists V^{P2\_INV1}_{\sim AIT201}(w))]$	$Safe$

Table 6.4. MSDND Proof for P201-208 and DPIT in Process 3 (Figure 6.2)

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
P201-8 = $OPEN$	$w \models V^{P2\_SD6}_{P201-8}(w)$ $= true$ $\sim (P201 - 8$ $= OPEN)$ $\Rightarrow = CLOSE$ $\equiv \sim P201 - 8$		$I_{6,Pump}$P201-8; $B_6 I_{6,Pump}$P201-8; $T_{6,Pump}$P201-8; $B_6 I_{6,Pump}$P201-8$\wedge T_{6,Pump} P201 - 8 \rightarrow B_6 P201 - 8$; $w \models V^{P2\_SD6}_{P201-8}(w) = true$; $I_{4,6}$ ~P201-8; $B_4 I_{4,6}$ ~P201-8; $T_{4,6}$ ~ P201-8; $B_4 I_{4,6}$ ~P201-8$\wedge T_{4,6}$ ~P201-8$\rightarrow$ B_4 ~P201-8; $w \models V^{P2\_SD4}_{\sim P201-8}(w) = true$;	$\exists w \in W \vdash$ $[\,(S_{P201-8} \oplus$ $S_{\sim P201-8})\,] \wedge$ $[\,w \models$ $(\nexists V^{P2\_SD4}_{\sim P201-8}(w))\,]$	$Explosion$ $result$ $here$
			$PROCESS$ 3		
$DPIT = 20$	$w \models V^{P3\_SD4}_{DPIT}(w) = true$ $\sim (DPIT = 20)$ $\Rightarrow DPIT = 40$ $\equiv \sim DPIT$		$I_{6,5} DPIT$; $B_6 I_{6,5} DPIT$; $T_{6,5} DPIT$; $B_6 I_{6,5} DPIT \wedge T_{6,5} DPIT \rightarrow B_6 DPIT$; $w \models V^{P3\_SD6}_{DPIT}(w) = true$; $I_{4,6} \sim DPIT$; $B_4 I_{4,6} \sim DPIT$; $T_{4,6} \sim DPIT$; $B_4 I_{4,6} \sim DPIT \wedge T_{4,6} \sim DPIT \rightarrow B_4 \sim DPIT$; $w \models V^{P3\_SD4}_{\sim DPIT}(w) = true$;	$\exists w \in W \vdash [\,(S_{DPIT} \oplus S_{DPIT})\,]$ $\wedge [\,w \models (\nexists V^{P3\_SD4}_{\sim DPIT}(w))\,]$	$Sensor$ $-failure$

Table 6.5. MSDND Proof for MV302, MV303

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$MV302 = OPEN$ $MV303 = CLOSE$	$w \models V^{P3\_SD4}_{\sim MV302,3}(w) = true$ $\sim (MV302 = OPEN)$ $\Rightarrow MV302 = OPEN$ $\sim (MV303 = CLOSE)$ $\Rightarrow MV303 = OPEN$ $\equiv\sim MV303$		$I_{6,8}MV302$; $B_6 I_{6,8} MV302$; $T_{6,8}MV302$; $B_6 I_{6,8} MV302 \wedge T_{6,8} MV302 \rightarrow B_6 \sim MV302$; $w \models V^{P3\_SD6}_{MV302}(w) = true$; $I_{4,6}MV302$; $B_4 I_{4,6} MV302$; $T_{4,6}MV302$; $B_4 I_{4,6} MV302 \wedge T_{4,6} MV302 \rightarrow B_4 MV302$; $w \models V^{P3\_SD4}_{MV302}(w) = true$; $I_{6,3}MV303$; $B_6 I_{6,3} MV303$; $T_{6,3}MV303$; $B_6 I_{6,3} MV303 \wedge T_{6,3} MV303 \rightarrow B_6 MV303$; $w \models V^{P3\_SD6}_{MV303}(w) = true$; $I_{4,6} \sim MV303$; $B_4 I_{4,6} \sim MV303$; $T_{4,6} \sim MV303$; $B_4 I_{4,6} \sim MV303 \wedge T_{4,6} \sim MV303 \rightarrow B_4 \sim MV303$; $w \models V^{P3\_SD4}_{\sim MV303}(w) = true$;	$\exists w \in W \vdash [\,(S_{MV303} \oplus S_{\sim MV303})\,]$ $\wedge [\, w \models (\nexists V^{P3\_SD4}_{\sim MV303}(w)\,]$ $\exists w \in W \vdash [\,(S_{MV302} \oplus S_{\sim MV302})\,]$ $\wedge [\, w \models (\nexists V^{P3\_SD4}_{\sim MV302}(w)\,]$	$Water$ $Wastage$

Table 6.6. MSDND Proof for MV302, MV304

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$MV302 = OPEN$ $MV304 = CLOSE$	$w \models V^{P3\_SD4}_{\sim MV302,3}(w) = true$ $\sim (MV302 = OPEN)$ $\Rightarrow MV302 = OPEN$ $\sim (MV304 = CLOSE)$ $\Rightarrow MV304 = OPEN$ $\equiv\sim MV304$		$I_{6,8}MV302$; $B_6 I_{6,8} MV302$; $T_{6,8} MV302$; $B_6 I_{6,8} MV302 \wedge T_{6,8} MV302 \rightarrow B_6 \sim MV302$; $w \models V^{P3\_SD6}_{MV302}(w) = true$; $I_{4,6} \sim MV302$; $B_4 I_{4,6} \sim MV302$; $T_{4,6} \sim MV302$; $B_4 I_{4,6} MV302 \wedge T_{4,6} MV302 \rightarrow B_4 MV302$; $w \models V^{P3\_SD4}_{\sim MV302}(w) = true$; $I_{6,3}MV304$; $B_6 I_{6,3} MV304$; $T_{6,3} MV304$; $B_6 I_{6,3} MV304 \wedge T_{6,3} MV304 \rightarrow B_6 MV304$; $w \models V^{P3\_SD6}_{MV304}(w) = true$; $I_{4,6} \sim MV304$; $B_4 I_{4,6} \sim MV304$; $T_{4,6} \sim MV304$; $B_4 I_{4,6} \sim MV304 \wedge T_{4,6} \sim MV304 \rightarrow B_4 \sim MV304$; $w \models V^{P3\_SD4}_{\sim MV304}(w) = true$;	$\exists w \in W \vdash [\,(S_{MV304} \oplus S_{\sim MV304})\,]$ $\wedge [\, w \models (\nexists V^{P3\_SD4}_{\sim MV304}(w)\,]$ $\exists w \in W \vdash [\,(S_{MV302} \oplus S_{\sim MV302})\,]$ $\wedge [\, w \models (\nexists V^{P3\_SD4}_{\sim MV302}(w)\,]$	$Water$ $Wastage$

Table 6.7. MSDND Proof for UV-Dechlorination Unit and P403, P404 in Process 4 (Figure 6.3)

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$UV-D=On$	$w \models V_{UV-D}^{P4\_SD4}(w) = true$ $\sim (UV-D=On)$ $\Rightarrow UV-D=Off$ $\equiv\sim UV-D$		$I_{6,5}UV-D$; $B_6 I_{6,5}UV-D$; $T_{6,5}UV-D$; $B_6 I_{6,5}UV-D \wedge T_{6,5}UV-D \rightarrow B_6 UV-D$; $w \models V_{UV-D}^{P4\_SD6}(w) = true$; $I_{4,6} \sim UV-D$; $B_4 I_{4,6} \sim UV-D$; $T_{4,6} \sim UV-D$; $B_4 I_{4,6} \sim UV-D \wedge T_{4,6} \sim UV-D \rightarrow B_4 \sim UV-D$; $w \models V_{\sim UV-D}^{P4\_SD4}(w) = true$;	$\exists w \in W \vdash [\,(S_{UV-D} \oplus S_{UV-D})\,]$ $\wedge[\,w \models (\nexists V_{\sim UV-D}^{P4\_SD4}(w))\,]$	$Sensor-failure$
$P403,4=Off$	$w \models V_{P403,4}^{P4\_SD6}(w) = true$ $\sim (P403,4=Off)$ $\Rightarrow P403,4=On$ $\equiv\sim P403,4$		$I_{6,5}P403,4$; $B_6 I_{6,5}P403,4$; $T_{6,5}P403,4$; $B_6 I_{6,5}P403,4 \wedge T_{6,5}P403,4 \rightarrow B_6 P403,4$; $w \models V_{P403,4}^{P4\_SD6}(w) = true$; $I_{4,6} \sim P403,4$; $B_4 I_{4,6} \sim P403,4$; $T_{4,6} \sim P403,4$; $B_4 I_{4,6} \sim P403,4 \wedge T_{4,6} \sim P403,4 \rightarrow B_4 \sim P403,4$; $w \models V_{\sim P403,4}^{P4\_SD4}(w) = true$;	$\exists w \in W \vdash [\,(S_{P403,4} \oplus S_{\sim P403,4})\,]$ $\wedge[\,w \models (\nexists V_{\sim P403,4}^{P4\_SD4}(w))\,]$	$Chemical\ Imbalance$

Table 6.8. MSDND Proof for AIT402 and AIT501 in Process 5 (Figure 6.4)

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$AIT402 = 162$	$w \models V^{P4\_SD6}_{AIT402}(w) = true$ $\sim (AIT402 = 162)$ $\Rightarrow AIT402 = 300$ $\equiv\sim AIT402$		$I_{6,5}AIT402$; $B_6 I_{6,5} AIT402$; $T_{6,5}AIT402$; $B_6 I_{6,5} AIT402 \wedge T_{6,5} AIT402 \rightarrow B_6 AIT402$; $w \models V^{P4\_SD6}_{AIT402}(w) = true$; $I_{4,6} \sim AIT402$; $B_4 I_{4,6} \sim AIT402$; $T_{4,6} \sim AIT402$; $B_4 I_{4,6} \sim AIT402 \wedge T_{4,6} \sim AIT402 \rightarrow B_4 \sim AIT402$; $w \models V^{P4\_SD4}_{\sim AIT402}(w) = true$;	$\exists w \in W \vdash [\,(S_{AIT402} \oplus S_{\sim AIT402})\,]$ $\wedge [\, w \models (\nexists V^{P4\_SD4}_{\sim AIT402}(w))\,]$	Chemical Imbalance
			$PROCESS5$		
$AIT501 = 7.8$	$w \models V^{P5\_SD6}_{AIT501}(w) = true$ $\sim (AIT501 = 7.8)$ $\Rightarrow AIT501 = 6.95$ $\equiv\sim AIT501$		$I_{6,0}AIT501$; $B_6 I_{6,0} AIT501$; $T_{6,0}AIT501$; $B_6 I_{6,0} AIT501 \wedge T_{6,0} AIT501 \rightarrow B_6 AIT501$; $w \models V^{P5\_SD6}_{AIT501}(w) = true$; $I_{4,6} \sim AIT501$; $B_4 I_{4,6} \sim AIT501$; $T_{4,6} \sim AIT501$; $B_4 I_{4,6} \sim AIT501 \wedge T_{4,6} \sim AIT501 \rightarrow B_4 \sim AIT501$; $w \models V^{P5\_SD4}_{\sim AIT501}(w) = true$;	$\exists w \in W \vdash [\,(S_{AIT501} \oplus S_{\sim AIT501})\,]$ $\wedge [\, w \models (\nexists V^{P5\_SD4}_{\sim AIT501}(w))\,]$	Chemical Imbalance

Table 6.9. MSDND Proof for PIT1-3, MV501 and MV503

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$PIT1-3=X$	$w \models V^{P5\_SD6}_{P5\_SD1}(w)=true$ $\sim(PIT1-3=X)$ $\Rightarrow PIT1-3=Y$ $\equiv\sim PIT1-3$		$I_{6,PIT1-3}PIT1-3;$ $B_6I_{6,PIT1-3}PIT1-3;$ $T_{6,PIT1-3}PIT1-3;$ $B_6I_{6,PIT1-3}PIT1-3 \wedge T_{6,PIT1-3}PIT1-3 \rightarrow B_6PIT1-3;$ $w \models V^{P5\_SD6}_{PIT1-3}(w)=true;$ $I_{4,6}\sim PIT1-3;$ $B_4I_{4,6}\sim PIT1-3;$ $T_{4,6}\sim PIT1-3;$ $B_4I_{4,6}\sim PIT1-3 \wedge T_{4,6}\sim PIT1-3 \rightarrow B_4\sim PIT1-3;$ $w \models V^{P5\_SD4}_{\sim PIT1-3}(w)=true;$	$\exists w \in W \vdash [\,(S_{PIT1-3} \oplus S_{\sim PIT1-3})\,]$ $\wedge[\,w \models (\nexists V^{P5\_SD4}_{\sim PIT1-3}(w))\,]$	$Infrastructure$ $Damage$
Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$MV501=OPEN$ $MV503=CLOSE$	$w \models V^{P5\_SD4}_{\sim MV501,3}(w)=true$ $\sim(MV501=OPEN)$ $\Rightarrow MV501=CLOSE$ $\equiv\sim MV501$ $\sim(MV503=CLOSE)$ $\Rightarrow MV503=OPEN$ $\equiv\sim MV503$		$I_{6,11}MV501;$ $B_6I_{6,11}MV501;$ $T_{6,11}MV501;$ $B_6I_{6,11}MV501 \wedge T_{6,11}MV501 \rightarrow B_6\sim MV501;$ $w \models V^{P5\_SD6}_{MV501}(w)=true;$ $I_{4,6}\sim MV501;$ $B_4I_{4,6}\sim MV501;$ $T_{4,6}\sim MV501; B_4I_{4,6}MV501 \wedge T_{4,6}MV501 \rightarrow B_4MV501;$ $w \models V^{P5\_SD4}_{\sim MV501}(w)=true;$ $I_{6,12}MV503;$ $B_6I_{6,12}MV503;$ $T_{6,12}MV503;$ $B_6I_{6,12}MV503 \wedge T_{6,12}MV503 \rightarrow B_6MV503;$ $w \models V^{P5\_SD6}_{MV503}(w)=true;$ $I_{4,6}\sim MV503;$ $B_4I_{4,6}\sim MV503;$ $T_{4,6}\sim MV503;$ $B_4I_{4,6}\sim MV503 \wedge T_{4,6}\sim MV503 \rightarrow B_4\sim MV503;$ $w \models V^{P5\_SD4}_{\sim MV503}(w)=true;$	$\exists w \in W \vdash [\,(S_{MV503} \oplus S_{\sim MV503})\,]$ $\wedge[\,w \models (\nexists V^{P5\_SD4}_{\sim MV503}(w)\,]$ $\exists w \in W \vdash [\,(S_{MV501} \oplus S_{\sim MV501})\,]$ $\wedge[\,w \models (\nexists V^{P5\_SD4}_{\sim MV501}(w)\,]$	$Water$ $Wastage$

Table 6.10. MSDND Proof for Level Switch and Pump in Process 6

Actual	Attack	Invariant	BIT Path	MSDND Secure?	Impact
$LS60X = H$	$w \models V^{P6\_SD6}_{LS60X}(w) = true$ $\sim (LS60X = H)$ $LS60X = L$ $\equiv\sim LS60X$		$I_{6,0}LS60X$; $B_6 I_{6,0}LS60X$; $T_{6,0}LS60X$; $B_6 I_{6,0}LS60X \wedge T_{6,0} \sim LS60X \rightarrow B_6 LS60X$; $w \models V^{P6\_SD6}_{LS60X}(w) = true$; $I_{4,6} \sim LS60X$; $B_4 I_{4,6} \sim LS60X$; $T_{4,6} \sim LS60X$; $B_4 I_{4,6} \sim LS60X \wedge T_{4,6} \sim LS60X \rightarrow B_4 \sim LS60X$; $w \models V^{P6\_SD4}_{\sim LS60X}(w) = true$;	$\exists w \in W \vdash [\,(\,S_{LS60X} \oplus S_{\sim LS60X}\,)\,]$ $\wedge[\,w \models (\,\nexists V^{P6\_SD4}_{\sim LS60X}(\,w)\,)\,]$	$Overflow$
$P603 = ON$	$w \models V^{P6\_SD6}_{P6\_SD1}(w) = true$ $\sim (P603 = ON)$ $\Rightarrow P603 = OFF$ $\equiv\sim P603$		$I_{6,1}P603$; $B_6 I_{6,1}P603$; $T_{6,1}P603$; $B_6 I_{6,1}P603 \wedge T_{6,1}P603 \rightarrow B_6 P603$; $w \models V^{P6\_SD6}_{P603}(w) = true$; $I_{4,6} \sim P603$; $B_4 I_{4,6} \sim P603$; $T_{4,6} \sim P603$; $B_4 I_{4,6} \sim P603 \wedge T_{4,6} \sim P603 \rightarrow B_4 \sim P603$; $w \models V^{P6\_SD4}_{\sim P603}(w) = true$;	$\exists w \in W \vdash [\,(\,S_{P603} \oplus S_{\sim P603}\,)\,]$ $\wedge[\,w \models (\,\nexists V^{P6\_SD4}_{\sim P603}(\,w)\,)\,]$	$Infrastructure$ $Damage$

7. CONCLUSIONS

The MSDND-based approach was found useful in modeling attacks on a water treatment plant where the goal of an attacker is to hide critical information from an operator rather than to steal it. Using this model, vulnerabilities across each stage of the water system are found and tabulated in this Section. This table lists the number of components in each process, invariants developed and vulnerabilities remaining. For each process, design decision is suggested which helps in making that stage more secure and help in generating invariants. Though the MSDN-based approach was used in this work in the context of a specific infrastructure, the approach, in its design, is generic and also applicable to other infrastructures such as power and oil&gas.

However, there remain several security domains that need to have invariants, or additional sensors, for knowing the true plant status. Vulnerabilities were discovered in later processes of the testbed such as in Processes 5 and 6 which requires the development of additional invariants. It is found that components that are vulnerable across all the stages of the testbed are sensors and actuators that are related to maintaining chemical balance of the water, hence, these components demand additional work for developing invariants in the future.

Table 7.1. Summary of Invariants, Vulnerabilities, and Components in each Stage of the Water Treatment System

Process	Comp	Summary	Design Recommendations
Process 1	4	Invariants Developed : 4 Vulnerabilities Remaining : 0	Invariants for FIT and LIT should be modified to better capture multipoint attacks
Process 2	11	Invariants Developed : 7 Vulnerabilities Remaining : 6	Chemical processes should be further analyzed for getting more reliable invariants. Chemical dosing pumps and level indicators should be modified.
Process 3	9	Invariants Developed : 4 Vulnerabilities Remaining : 2	Several attacks can be performed on motorized valves for damaging pumps and draining water. Install PIT near UF Unit to generate invariant for DPIT
Process 4	7	Invariants Developed : 3 Vulnerabilities Remaining : 1	Dechlorination Unit and $NaHSO_3$ dosing's effects chemical properties of water, using this, better invariants should be made as it effects RO Unit
Process 5	16	Invariants Developed : 7 Vulnerabilities Remaining : 9	Many MSDND Secure paths are identified, invariants should be developed to break the MSDND
Process 6	7	Invariants Developed : 2 Vulnerabilities Remaining : 5	Level switches should be replaced with level indicators, and additional FITs should be installed for generating invariants

Table 7.2. Testing Results

PROCESS 1

Comp	Expected Result	Approach	Observed Result	Comments
LIT101	$V^{P1\_SD4}_{LIT101}(w) = true$	$Normal$	$V^{P1\_SD4}_{\sim LIT101}(w) = true$	Using LIT_Est the change can be found
LIT101 FIT101	$V^{P1\_SD4}_{LIT,FIT}(w) = true$	$Normal$	$V^{P1\_SD4}_{\sim LIT,FIT}(w) = true$	As the FIT101 is made equal to FIT201 and LIT101 value is kept as constant (500), the level of the water in the tank increases without violating any invariants
FIT101	$V^{P1\_SD4}_{FIT101}(w) = true$	$Normal$	$V^{P1\_SD4}_{\sim FIT101}(w) = true$	Using $P1\_INV1$ the change can be found. This invariant is violated only if the changed value is lesser than 0.5 units when it is supposed to be greater than 0.5 and vice versa.
MV101	$V^{P1\_SD4}_{MV101}(w) = true$	$Normal$	$V^{P1\_SD4}_{\sim MV101}(w) = true$	Using $P1\_INV2$ the change can be found.
P101	$V^{P1\_SD4}_{P101}(w) = true$	$Normal$	$V^{P1\_SD4}_{\sim P101}(w) = true$	Using $P1\_INV3$ the change can be found.

PROCESS 2

Comp	Expected Result	Approach	Observed Result	Comments

Table 7.2 Testing Results (cont.)

AIT201	$V_{AIT201}^{P2\_SD4}(w) = true$	$Normal$	$V_{\sim AIT201}^{P2\_SD4}(w) = true$	Using $P2\_INV1$ the change can be found. This invariant does not tell anything about the change in the values of conductivity, it only tells that the P201,2 should be off when AIT503 is High. This might not be correct when there is a change in inlet water.
AIT202	$V_{AIT202}^{P2\_SD4}(w) = true$		$V_{\sim AIT202}^{P2\_SD4}(w) = false$	There is no invariant to know the status of AIT202
AIT203	$V_{AIT203}^{P2\_SD4}(w) = true$	$Normal$	$V_{\sim AIT203}^{P2\_SD4}(w) = true$	Using $P2\_INV2$ the change can be found. This invariant does not tell anything about the change in the values of conductivity, it only tells that the P201,2 should be off when AIT503 is High. This might not be correct when there is a change in inlet water.

Table 7.2 Testing Results (cont.)

AIT201 AIT202 AIT203	$V^{P2\_SD4}_{AIT201-3}(w) = true$		$V^{P2\_SD4}_{\sim AIT201-3}(w) = true$	Using $P2\_INV3$ the change can be found. In this we changed the value of three AIT but only one invariant is raised which tells us about AIT203 and hence the invariant does not capture the change.
MV201	$V^{P2\_SD4}_{MV201}(w) = true$	$Normal$	$V^{P2\_SD4}_{\sim MV201}(w) = true$	Using $P2\_INV4$ the change can be found.
P201-8	$V^{P2\_SD4}_{P201-8}(w) = true$		$V^{P2\_SD4}_{\sim P201-8}(w) = false$	There is no invariant to know the status of AIT202.

PROCESS 3

Component	Expected Result	Approach	Observed Result	Comments
LIT301	$V^{P3\_SD4}_{LIT301}(w) = true$	$Normal$	$V^{P3\_SD4}_{\sim LIT301}(w) = true$	Using LIT_Est the change can be found.
P301	$V^{P3\_SD4}_{P301}(w) = true$	$Normal$	$V^{P3\_SD4}_{\sim P301}(w) = true$	Using $P3\_INV1$ the change can be found.
FIT301	$V^{P3\_SD4}_{FIT301}(w) = true$	$Normal$	$V^{P3\_SD4}_{\sim FIT301}(w) = true$	Using $P3\_INV2$ the change can be found. This invariant is violated only if the changed value is lesser than 0.5 units when it is supposed to be greater than 0.5 and vice versa.

Table 7.2 Testing Results (cont.)

MV301	$V^{P3\_SD4}_{MV301}(w) = true$	Emp	$V^{P3\_SD4}_{\sim MV301}(w) = true$	Using $P3\_INV3$ the change can be found.
MV302	$V^{P3\_SD4}_{MV302}(w) = true$	$Normal$	$V^{P3\_SD4}_{\sim MV302}(w) = true$	Using $P3\_INV1$ the change can be found.
MV303	$V^{P3\_SD4}_{MV303}(w) = true$	Emp	$V^{P3\_SD4}_{\sim MV303}(w) = true$	Using Emp_P3_INV3 the change can be found. In the absence of an invariant, when MV303 is open when it is supposed to be close the water is sent to the drain without being detected
MV304	$V^{P3\_SD4}_{MV304}(w) = true$	Emp	$V^{P3\_SD4}_{\sim MV304}(w) = true$	Using Emp_P3_INV3 the change can be found. In the absence of an invariant, when MV304 is open when it is supposed to be close the water is sent to the drain without being detected
DPIT	$V^{P3\_SD4}_{DPIT}(w) = true$		$V^{P3\_SD4}_{\sim DPIT}(w) = false$	There is no invariant to know the status of DPIT

PROCESS 4

LIT401	$V^{P4\_SD4}_{LIT401}(w) = true$	$Normal$	$V^{P4\_SD4}_{\sim LIT401}(w) = true$	Using LIT_Est the change can be found.
P401	$V^{P4\_SD4}_{P401}(w) = true$	$Normal$	$V^{P4\_SD4}_{\sim P401}(w) = true$	Using $P4\_INV1$ the change can be found.

Table 7.2 Testing Results (cont.)

FIT401	$V^{P4\_SD4}_{FIT401}(w) = true$	$Normal$	$V^{P4\_SD4}_{\sim FIT401}(w) = true$	Using $P4\_INV2$ the change can be found. This invariant is violated only if the changed value is lesser than 0.5 units when it is supposed to be greater than 0.5 and vice versa.
UV-D	$V^{P4\_SD4}_{UV-D}(w) = true$		$V^{P4\_SD4}_{\sim UV-D}(w) = false$	There is no invariant to know the status of UV-D
AIT402	$V^{P4\_SD4}_{AIT402}(w) = true$		$V^{P4\_SD4}_{\sim AIT402}(w) = false$	There is no invariant to know the status of AIT402
P403,4	$V^{P4\_SD4}_{P403,4}(w) = true$		$V^{P4\_SD4}_{\sim P403,4}(w) = false$	There is no invariant to know the status of P403,4

PROCESS 5

AIT501	$V^{P5\_SD4}_{AIT501}(w) = true$		$V^{P5\_SD4}_{\sim AIT501}(w) = false$	There is no invariant to know the status of AIT501
AIT502	$V^{P5\_SD4}_{AIT502}(w) = true$		$V^{P5\_SD4}_{\sim AIT502}(w) = false$	There is no invariant to know the status of AIT502
AIT503	$V^{P5\_SD4}_{AIT503}(w) = true$		$V^{P5\_SD4}_{\sim AIT503}(w) = false$	There is no invariant to know the status of AIT503
P501	$V^{P5\_SD4}_{P501}(w) = true$		$V^{P5\_SD4}_{\sim P501}(w) = true$	Using $P5\_INV2$ the change can be found.
PIT1-3	$V^{P5\_SD4}_{PIT1-3}(w) = true$		$V^{P5\_SD4}_{\sim PIT1-3}(w) = false$	There is no invariant to know the status of PIT1-3

Table 7.2 Testing Results (cont.)

MV501-4	$V^{P5\_SD4}_{MV501-4}(w) = true$		$V^{P5\_SD4}_{\sim MV501-4}(w) = true$	There is no invariant to know the status of MV501-4. If the status of MV501 is On and MV503 is Off in the normal operation, if these are reversed then there is no way for one to evaluate this. This is same with MV502 and MV504
P501	$V^{P5\_SD4}_{P501}(w) = true$		$V^{P5\_SD4}_{\sim P501}(w) = true$	Using $P5\_INV2$ the change can be found.
FIT501-4	$V^{P5\_SD4}_{FIT501-4}(w) = true$	$Normal$	$V^{P5\_SD4}_{\sim FIT501-4}(w) = true$	Using FIT_Est the change can be found.

PROCESS 6

LS601-3	$V^{P6\_SD4}_{LS601-3}(w) = true$		$V^{P6\_SD4}_{\sim LS601-3}(w) = false$	There is no invariant to know the status of LS601-3
P601,3	$V^{P6\_SD4}_{P601,3}(w) = true$		$V^{P6\_SD4}_{\sim P601,3}(w) = false$	There is no invariant to know the status of P601,3
P602	$V^{P6\_SD4}_{P602}(w) = true$	$Normal$	$V^{P6\_SD4}_{\sim P602}(w) = true$	Using $P6\_INV2$ the change can be found.
FIT601	$V^{P6\_SD4}_{FIT601}(w) = true$		$V^{P6\_SD4}_{\sim FIT601}(w) = true$	Using $P6\_INV1$ the change can be found.

REFERENCES

'Ics-cert,' https://ics-cert.us-cert.gov/, 2016.

Adepu, S. and Mathur, A., 'Generalized attacker and attack models for cyber physical systems,' in 'Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual,' volume 1, IEEE, 2016a pp. 283–292.

Adepu, S. and Mathur, A., 'An investigation into the response of a water treatment system to cyber attacks,' in 'High Assurance Systems Engineering (HASE), 2016 IEEE 17th International Symposium on,' IEEE, 2016b pp. 141–148.

Blanchet, B., 'Vérification automatique de protocoles cryptographiques: modele formel et modele calculatoire,' Mémoire dâĂŹhabilitationa diriger des recherches, Université Paris-Dauphine, 2008.

Cárdenas, A. A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., and Sastry, S., 'Attacks against process control systems: risk assessment, detection, and response,' in 'Proceedings of the 6th ACM symposium on information, computer and communications security,' ACM, 2011 pp. 355–366.

Cardenas, A. A., Amin, S., and Sastry, S., 'Secure control: Towards survivable cyber-physical systems,' in 'Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on,' IEEE, 2008 pp. 495–500.

Chen, T., 'Stuxnet, the real start of cyber warfare?[editor's note],' IEEE Network, 2010, **24**(6), pp. 2–3.

Cobb, P., 'German steel mill meltdown: Rising stakes in the internet of things,' 2016 .

Cruz, T., Barrigas, J., Proença, J., Graziano, A., Panzieri, S., Lev, L., and Simões, P., 'Improving network security monitoring for industrial control systems,' in 'Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on,' IEEE, 2015 pp. 878–881.

Dunaka, P. R. and McMillin, B., 'Cyber-physical security of a chemical plant,' in 'High Assurance Systems Engineering (HASE), 2017 IEEE 18th International Symposium on,' IEEE, 2017 pp. 33–40.

Gao, W., Morris, T., Reaves, B., and Richey, D., 'On scada control system command and response injection and intrusion detection,' in 'eCrime Researchers Summit (eCrime), 2010,' IEEE, 2010 pp. 1–9.

Howser, G. and McMillin, B., 'A multiple security domain model of a drive-by-wire system,' in 'Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual,' IEEE, 2013a pp. 369–374.

Howser, G. and McMillin, B., 'A multiple security domain model of a drive-by-wire system,' in 'Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual,' IEEE, 2013b pp. 369–374.

Howser, G. and McMillin, B., 'A modal model of stuxnet attacks on cyber-physical systems: A matter of trust,' in 'Software Security and Reliability (SERE), 2014 Eighth International Conference on,' IEEE, 2014 pp. 225–234.

Humayed, A., Lin, J., Li, F., and Luo, B., 'Cyber-physical systems securityâĂŤa survey,' IEEE Internet of Things Journal, 2017, **4**(6), pp. 1802–1831.

Krotofil, M., Larsen, J., and Gollmann, D., 'The process matters: Ensuring data veracity in cyber-physical systems,' in 'Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security,' ACM, 2015 pp. 133–144.

Kwon, C., Liu, W., and Hwang, I., 'Security analysis for cyber-physical systems against stealthy deception attacks,' in 'American Control Conference (ACC), 2013,' IEEE, 2013 pp. 3344–3349.

Lee, E. A., 'Cyber physical systems: Design challenges,' Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, 2008.

Liau, C.-J., 'Belief, information acquisition, and trust in multi-agent systemsâĂŤa modal logic formulation,' Artificial Intelligence, 2003, **149**(1), pp. 31–60.

Liau, C.-J., 'A modal logic framework for multi-agent belief fusion,' ACM Transactions on Computational Logic (TOCL), 2005, **6**(1), pp. 124–174.

LIPOVSKY, R., 'New wave of cyber attacks against ukrainian power industry,' https://www.welivesecurity.com/2016/01/20/new-wave-attacks-ukrainian-power-industry/, 2016.

Mathur, A. P. and Tippenhauer, N. O., 'Swat: A water treatment testbed for research and training on ics security,' in 'Cyber-physical Systems for Smart Water Networks (CySWater), 2016 International Workshop on,' IEEE, 2016 pp. 31–36.

Owicki, S. and Gries, D., 'An axiomatic proof technique for parallel programs i,' Acta informatica, 1976, **6**(4), pp. 319–340.

securelist.com, 'threatlandscape,' https://securelist.com/threat-landscape-for-industrial-automation-systems-in-h1-2017/82660/, 2017.

Stouffer, K., Falco, J., and Scarfone, K., 'Guide to industrial control systems (ics) security,' NIST special publication, 2011, **800**(82), pp. 16–16.

SUTD, i., 'Secure water treatment testbed, singapore,' 2016.

Sutherland, D., 'A model of information, in proceedings of the 9th national computer security conference,' 1986 pp. 175–183.

Urbina, D. I., Giraldo, J. A., Tippenhauer, N. O., and Cárdenas, A. A., 'Attacking fieldbus communications in ics: Applications to the swat testbed.' in 'SG-CRC,' 2016 pp. 75–89.

Weinberger, S., 'Computer security: Is this the start of cyberwarfare?' Nature News, 2011, **474**(7350), pp. 142–145.

VITA

Sai Sidharth Patlolla was born in Hyderabad, India. He earned a Bachelor of Technology from Jawaharlal Nehru Technological University in May 2016, majoring in Computer Science.

He received his Master of Science degree in Computer Science from Missouri University of Science and Technology in May 2018. While there, he greatly enjoyed his work as a research assistant to Dr. Bruce McMillin for two years. Sai Sidharth presented his research work at scientific meetings and participated in conferences discussing challenges in high performance computing. This was possible as a result of his securing competitive funding from the National Science Foundation.