




IDENTIFYING CYBER-PHYSICAL VULNERABILITIES OF WATER DISTRIBUTION SYSTEMS USING FINITE STATE PROCESSES

Cade Karrenberg¹, Juan Benavides², Emily Berglund³, Eunsuk Kang⁴, John Baugh⁵

^{1,2,3,5}North Carolina State University, Raleigh, North Carolina, United States of America

⁴Carnegie Mellon University, Pittsburgh, Pennsylvania, United State of America

¹ 0000-0002-6067-1004, ckarren@ncsu.edu, ² jdbenavi@ncsu.edu, ³ 0000-0001-9005-9468,
emily_berglund@ncsu.edu,

⁴eunsuk.kang@gmail.com, ⁵jwb@ncsu.edu

Abstract

Modern water distribution systems (WDS) comprise not only physical infrastructure, but also use smart meters, sensors, automated control systems and wireless communication links to manage hydraulic processes and water quality. Networked devices create new vulnerabilities to cyber-attacks, in which an attacker infiltrates connected devices through internet and network connections with potentially severe consequences, such as creating water supply interruptions and compromising water quality. Attention to cyber-attacks comes at a time of notable intrusions into the computer systems that monitor and control infrastructure, including the recent attack on a water treatment plant in Oldsmar, Florida. This paper describes an approach for probing a system's vulnerabilities and finding attack scenarios that may cause a disruption, for example, by lowering the pressure in water mains and exposing a system to harmful contaminants. Our modelling framework uses a process calculus called Finite State Processes (FSP), which is a formal notation that also includes a supporting tool, Labelled Transition System Analyzer (LTSA), for automatically checking safety and other desirable properties of computer systems. Applying formal methods tools, most-often used in the design and testing of hardware and software systems, to WDSs allows us to augment traditional simulation approaches with the exhaustive model-checking capabilities of tools such as LTSA. This framework couples FSP with epanetCPA, which is an open-source toolbox that can simulate attacks on a system's computer and network components to evaluate the resulting hydraulic response. Within this framework, we treat WDSs as a bounded state control problem to ensure, for instance, that water levels of an elevated storage tank are always within an acceptable range. Attacker capabilities are broadly defined and modelled to allow communication links to be compromised through eavesdropping and packet injection attacks. Feasible attack scenarios are automatically identified and produced by LTSA, which generates counterexamples to a safety property. To incorporate the physics of WDSs into FSP, we discretize and quantize systems and calibrate observable behaviors using Python scripts, including the package Water Network Tool for Resilience (WNTR). Attack scenarios are simulated with epanetCPA for purposes of validation.

Keywords

Cyber-security, formal methods, water distribution system, cyberphysical system, finite state processes

1 INTRODUCTION

The growing popularity and utility of ‘smart’ and connected utilities has given rise to the proliferation of advanced metering infrastructure (AMI) and industrial control systems (ICS) within water utilities [1]. As a result of the integration of AMI and ICS, water distribution systems (WDS) can be treated, operated, and analyzed as cyberphysical systems [2]. Cyberphysical water distribution networks are composed of physical components that store, transport, and deliver water, and cyber components that control the physical components and communication between physical and cyber components. Cyber WDSs include traditional physical network elements such as pipes, pumps, valves, tanks, and reservoirs which ensure consistent and efficient delivery of water. The cyber layer is comprised of elements that enhance data analytics and the automation of network processes. These elements can include supervisory control and data acquisition (SCADA) units that automatically monitor the system and programmable logic controllers (PLCs) that remotely control system elements such as pumps and valves [3]. While the emergence of cyberphysical WDSs has promoted more efficient and reliable service, the technologies incorporated into cyberphysical WDSs introduce vulnerabilities into the networks and the managing utility [1]. SCADA, PLCs and remote telemetry units (RTU) found in cyberphysical WDSs rely on wireless communications and are susceptible to cyber-attacks [4]. Since 2015 the United States Environmental Protection Agency (USEPA) and the National Institute for Standards and Technology (NIST) have provided water utilities tools to help assess cybersecurity risks, but these tools lack network-specific assessments and are intended to identify potential risks, not vulnerabilities of a specific WDS [5].

There is growing research in the security of cyberphysical WDSs, including the detection of cyber-attacks in real-time, modeling the results of a cyber-attack on a network, and the evaluation of network resilience after a cyber-attack [6]. Cyber-attack detection focuses on utilizing modeling tools, machine learning, and/or statistical models to identify anomalous behavior in the WDS, which is the result of a cyber-attack on the system. Cyber-attack detection relies on water network simulations to identify cyber-attacks as they occur [1]. Impact assessment of cyber-attacks on WDSs, and network resilience evaluation post-cyber-attack also depend on hydraulic simulations of WDSs. Hydraulic simulations can be computationally intensive, depending on the complexity of the water network, and existing hydraulic modeling tools, such as EPANET simulate only the physical components of WDSs and lack the capability to model the cyber components and the communication links of a cyberphysical WDS [7]. Tools such as epanetCPA, developed by Taormina et al. [8], incorporate cyber elements into the network model, which enables modeling of all cyber and physical elements of a WDS to evaluate the hydraulic response to cyber-attacks on a network. EpanetCPA and other impact assessment models require exhaustive hydraulic simulation of the WDS. Additionally, all possible attack scenarios must be known a priori to simulate the network under the attack scenarios and thoroughly identify all unsafe operating conditions of the network during and after an attack. Most cyber-attack detection models rely only on the physical modeling of a WDS, and do not include the cyber elements in the modeling of a network. This simplification of cyberphysical WDSs can obscure the complex relationships between the physical and cyber elements of a network, and the cascading effects of a cyber-attack on the network. The differences in temporal and spatial resolutions of physical models and cyber models can lead to inaccurate representations of cyberphysical WDSs, resulting in cyber-security models that lack application to real WDSs [3].

Using existing tools to manage WDS cybersecurity poses a number of challenges, including computationally expensive hydraulic simulation, identification of all known cyber-attacks a priori and the misrepresentation of cyberphysical WDSs. This research introduces an approach to identify vulnerabilities of WDSs using formal methods. Within computer science and software engineering, formal methods are modeling techniques for rigorous specification of software that assure the correctness of the properties of the software being modeled [9], [10]. Formal methods tools are developed to model the physical and cyber components of a WDS in a process algebra notation, Finite State Process (FSP). A corresponding verification tool, Labelled Transition System

Analysar (LTSA) is applied to verify the properties of a cyberphysical WDS [11], [12]. LTSA automatically performs exhaustive property checks on a model, which allows us to simulate a WDS without extensive hydraulic modeling.

2 METHODOLOGY

2.1 Discretizing Cyberphysical Water Distribution Networks

Modeling in FSP is limited to discrete relationships between components, and any model must be in one of a finite number of states. Given the nature of FSP models, a cyberphysical WDS can be described in discrete terms and simple linear relationships to represent components and their interactions in FSP. Water distribution networks are dynamic and weakly non-linear systems, and hydraulic modeling and simulation tools capture these dynamics using complex algorithms such as a global gradient algorithm [13]. To translate these into more easily managed linear relationships, target components of a water distribution network are identified for inclusion in the model, and behavior of those elements are recorded for the duration of a specified period under normal demand conditions.

This type of modeling can be informed by a hydraulic simulator such as EPANET, or the data can be directly obtained from actual network sensor readings. Complex water distribution networks can be reconfigured into district metered areas (DMAs) that act as isolated, independent water distribution networks to simplify systems, and identify the relationships between physical components in the DMAs [14]. Using time-series behavior of the target elements, a linear regression can be performed to translate network behavior into linear approximations. All network components, physical and cyber, must also be discretized.

For example, while the height of water in a tank can have a continuous value between zero (or a prescribed minimum value) and a maximum value, the height of water in a tank should be modeled as a discrete state of the tank to be represented in FSP. For example, a cylindrical tank with a fixed diameter, a minimum height of water of 1m and a maximum height of water of 8.5m can be represented by a tank with four discrete states: (1) water at a height of 1m, (2) water at a height of 3.5m, (3) water at a height of 6m, (4) water at a height of 8.5m (Figure 1).

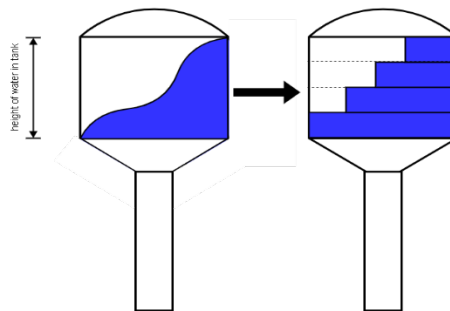


Figure 1. Water tank with continuous height of water (left) and discretized water tank with four states (right)

Demand, like the height of water in a tank, is also a continuous value that must be discretized to be represented in FSP. Similar to tank volume, demand can be represented using a finite number of states to represent the minimum, maximum, and intermediate values. For a system with demands that range from a minimum of 50 LPS to a maximum of 250 LPS, demands can be binned, with each bin representing a discrete state of demand, as illustrated in Table 1.

Table 1. Discrete state representation of demand

Continuous demand range	Demand State
<50 LPS	1
50 LPS - <100 LPS	2
100 LPS - <150 LPS	3
150 LPS - <200 LPS	4
200 LPS - 250 LPS	5

Pumping stations in a network can be represented by a single element within a finite number of states, depending on the number of pumps in the station and the properties of each pump. A pumping station with three pumps that have identical pump curves can be represented by a single composite element that has four states, as illustrated in Table 2.

Table 2. Discrete state representation of a pumping station

Pump Status	Pump State
1 pump ON, 2 pumps OFF	1
2 pumps ON, 1 pump OFF	2
3 pumps ON	3
3 pumps OFF	4

Another example of a pump station is provided in Table 3. For a pump station with three pumps, with different pump curves a single composite element with eight states is used to represent the pump station (Table 3).

Table 3. Discrete state representation of pumps with different pump curves

Pump Status	Pump State
Pump1 OFF, Pump 2 OFF, Pump3 OFF	1
Pump1 ON, Pump2 OFF, Pump3 OFF	2
Pump1 ON, Pump2 ON, Pump3 OFF	3
Pump1 ON, Pump2 ON, Pump3 ON	4
Pump1 OFF, Pump2 ON, Pump3 OFF	5
Pump1 OFF, Pump2 ON, Pump3 ON	6
Pump1 OFF, Pump2 OFF, Pump3 ON	7

Pump1 ON, Pump2 OFF, Pump3 ON	8
-------------------------------	---

2.2 Water Distribution Network as Finite State Processes

Once cyberphysical WDS elements are identified and discretized into representative states, and relationships between elements are linearized, the system can be represented in FSP. FSP is a textual process algebra that describes models as processes, each with a finite number of states, and actions, which describe how the model transitions from one state to another. Conventionally, processes describe model components and are described in all uppercase letters. A “process ... transforms its state by executing statements [which consist] of one of more atomic actions that make indivisible state changes” [12, p. 32]. Figure 2 illustrates how a door can be modeled in textual FSP.

$$\text{DOOR} = (\text{open} \rightarrow \text{close} \rightarrow \text{DOOR}).$$

Figure 2. FSP for DOOR

DOOR is a process represented by two states, and the actions “open” and “close” that transition DOOR between the two states. This FSP can also be described graphically as a Labelled Transition System (LTS), and diagrams can be automatically generated using LTSA [11]. Figure 3 shows the LTS for DOOR.

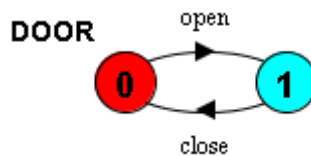


Figure 3. LTS for DOOR

Each physical and cyber component of a water distribution system can be described as an FSP process. A composition of all processes into a single FSP model represents a complete cyberphysical water distribution system. Figure 4 is an LTS of a tank modeled in FSP as the process TANK with four discrete states.

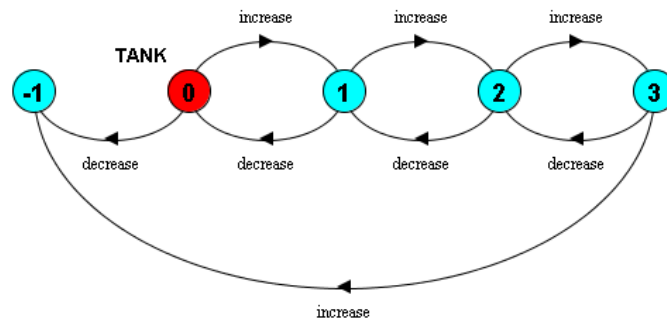


Figure 4. LTS for a tank

The states of the tank, represented in the LTS as nodes 0-3 correspond to the discretized height of water in the tank. When the height of water increases in the tank, the “increase” action is executed, and the tank moves from “State 0” to “State 1”. If the water then decreases, the “decrease” action is executed, and the tank transitions from “State 1” to “State 0.” If the tank is in “State 0” and the water decreases, the LTS demonstrates that the model enters an error state, shown as “State -1.” This error reflects an invariant in a physical tank; the height of water in a tank cannot decrease if it is already at its minimum. The error state is one mechanism that LTSA offers

to check models for safety properties. LTSA also provides an error trace, the path through the system that allows the model to enter an error state. The mechanisms provided by LTSA can be used to identify vulnerabilities in complex systems comprised of physical elements, cyber elements.

2.3 Case Study: Mini-Town

The FSP methodology and modeling approach was applied to an illustrative water distribution network, Mini-Town. Mini-Town is a looped water distribution network comprised of one reservoir, one cylindrical tank with a fixed diameter, one pumping station with two pumps in parallel, two loops of five pipes total, and four demand nodes, (Figure 5). The demand follows a diurnal pattern. The status of the pumping station is governed by the level of water in the tank by the following rules:

PUMP1 is turned ON if the level of water in TANK is below 4 m

PUMP1 is turned OFF if the level of water in TANK is above 6.3 m

PUMP2 is turned ON if the level of water in TANK is below 1 m

PUMP2 is turned OFF if the level of water in TANK is above 4.5 m

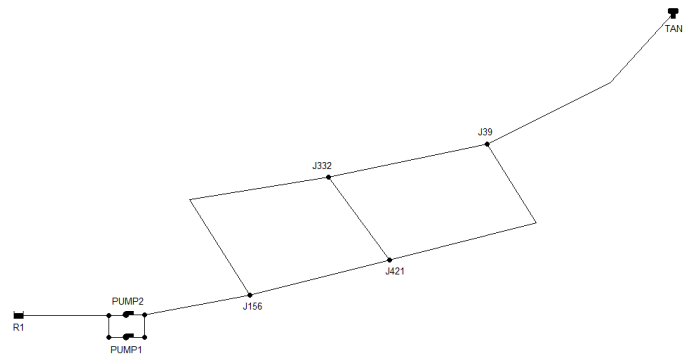


Figure 5. Mini-Town

Figure 6 shows the height of water in the tank, the total flow through the pumping station, and the total demand for the duration of a 168-hour hydraulic simulation of Mini-Town. The analysis shown in Figure 6 was performed using EPANET.

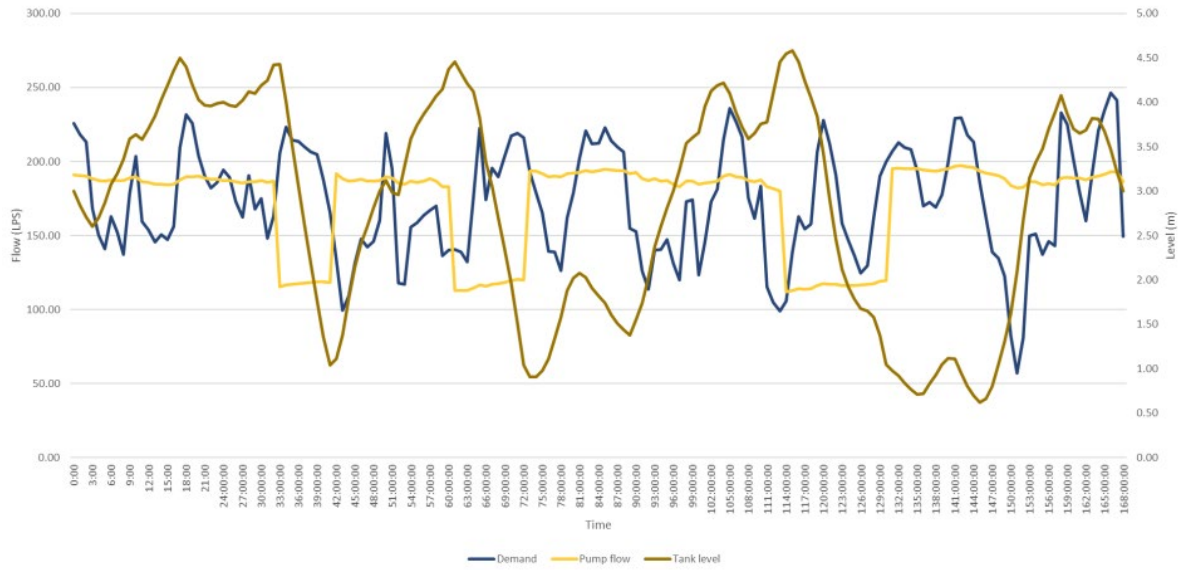


Figure 6. Demands, pump flow, and tank level in Mini-Town, report by EPANET model.

3 RESULTS

3.1 Finite State Processes model of Mini-Town

The physical components of Mini-Town that are included in the FSP model are the reservoir, tank, pumps, and demand nodes (Table 4).

Table 4. Key physical components of Mini-Town

Component	Name	Elevation (m)	Min./Max. level (m)	Diameter (m)	Curve equation	Base Demand (LPS)
Reservoir	R1	59.0	--	--	--	--
Tank	TANK	71.5	0.0 / 6.5	31.3	--	--
Pump	PUMP1	--	--	--	$h_G = 70.0 - 0.07731q^{1.36}$	--
Pump	PUMP2	--	--	--	$h_G = 70.0 - 0.07731q^{1.36}$	--
Node	J156	56.2	--	--	--	21.8
Node	J332	44.2	--	--	--	8.61
Node	J421	37.1	--	--	--	15.04
Node	J39	45.9	--	--	--	62.41

To analyze the system and develop linear equations and discrete representations, the simulation is split into two regimes. In the first regime, one pump is on, and in the second regime two pumps are on. A multivariate linear regression is performed for each regime using the height of water in

the tank, h , and the total demand of the nodes, Q_d as predictors for the total flow through the pumps, Q_p . Equations (1) and (2) show the results of the linear regression for the first and second regime, respectively.

$$Q_p = 0.036Q_d - 1.34h_t + 113.9 \quad (1)$$

$$Q_p = 0.067Q_d - 2.20h_t + 183.5 \quad (2)$$

Equation (1) reports an $R^2=0.99$, and equation (2) an $R^2=0.84$. Equations (1) and (2) are used to simulate flows through the pumps based on the height of water in the tank and the total demand. There are only two water sources in Mini-Town, the reservoir, and the tank. Therefore, the total demand, Q_d must be equal to the flow from the reservoir, Q_r plus the flow from the tank, Q_T , as shown in equation (3). The pumping station is directly downstream of the reservoir with no other junctions between the reservoir and the pumping station, and the total flow of the pumps is equal to the flow from the reservoir, as shown in equation (4).

$$Q_d = Q_r + Q_T \quad (3)$$

$$Q_r = Q_p \quad (4)$$

Combining equations (1) or (2) with equations (3) and (4) and rearranging, the flow from the tank under regime one and two, is shown in equations (5) and (6), respectively.

$$Q_T = Q_d - 0.036Q_d - 1.34h_t + 113.9 \quad (5)$$

$$Q_T = Q_d - 0.067Q_d - 2.20h_t + 183.5 \quad (6)$$

Given the physical characteristics of the tank, the flowrate from the tank is given by equation (7). For each time step, Δt , given demand, Q_d and the height of water in the tank at the beginning of the time step, h_t equation (7) can be rearranged to solve for the change in height of water in tank, Δh , as shown in equation (8).

$$Q_T = \frac{\pi r^2 h}{t} \quad (7)$$

$$\Delta h = \Delta t \frac{Q_d - Q_p}{\pi r^2} \quad (8)$$

From these equations, dynamic hydraulic relationships of the physical components of Mini-Town are represented as linear state transitions in FSP. Figure 7 illustrates the performance of the linear representation of Mini-Town compared to a hydraulic simulation of Mini-Town using EPANET. The linear model has a root mean square error (RMSE) of 0.235, indicating that the linearization of Mini-Town is an accurate representation of the dynamic relationships between its physical components.

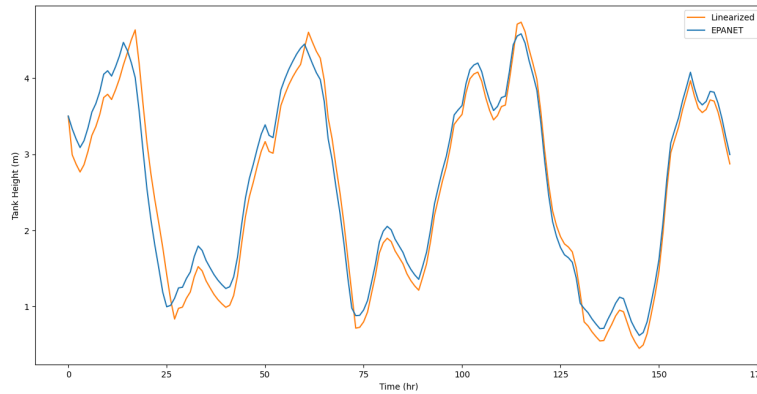


Figure 7. Linear simulation of Mini-Town vs. EPANET simulation

Figure 8 is a process diagram that outlines how the relationships between the physical and cyber components of Mini-Town will be structured in the FSP model.

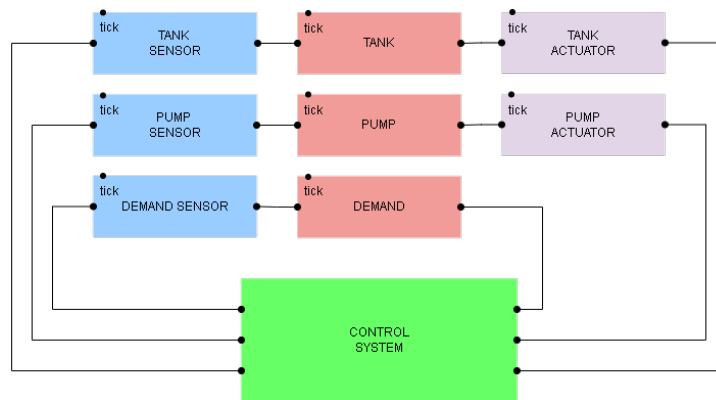


Figure 8. Process diagram of FSP model of Mini-Town

Based on the process diagram, the key physical components of Mini-Town are then discretized and represented as *states* in FSP. Figure 9 illustrates the performance of the linearized and discretized model of Mini-Town compared to a hydraulic simulation in EPANET. The total demand has 13 discrete states, the tank has 17 discrete states, and the pumps have two discrete states. The linearized, discretized model has an RMSE of 0.423. The results of the linear regression and the discretization indicate that an FSP model of Mini-Town would retain a high level of fidelity when compared to a hydraulic simulation.

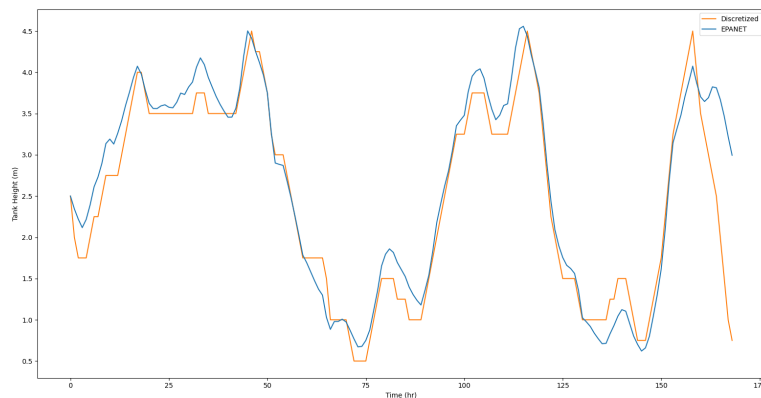


Figure 9. Linearized and discretized simulation of Mini-Town vs. EPANET simulation

The tank, pumping station, and demand are the physical components of Mini-Town represented in the FSP model as TANK, PUMP, and DEMAND (the modeling of the reservoir is implicit based on equation (4)). The cyber components included in the FSP model are sensors, actuators, the control system (PLCs and SCADA), as TANK SENSOR, PUMP SENSOR, DEMAND SENSOR, TANK ACTUATOR, PUMP ACTUATOR, and CONTROL SYSTEM, and the communication links among all components. TANK, PUMP, and DEMAND are synchronized with their respective sensors by an FSP *action*, “tick”, which denotes a time-step. The sensors send their state readings of each of the physical components to CONTROL SYSTEM via cyber communication links. Based on the input from the sensors, CONTROL SYSTEM sends commands to TANK ACTUATOR and PUMP ACUATOR via cyber communication links, which send transition commands to TANK and PUMP, respectively. Demand is modeled stochastically, which occurs in CONTROL SYSTEM, which is then sent directly to DEMAND via a cyber communication link. Figure 10 shows a fragment of the Mini-Town FSP model, whose parameters are generated automatically by a custom Python script. The pump controller is represented by the FSP *process* PC, which is a part of CONTROL SYSTEM. The status of the pumping station is governed by the height of water in the tank, modeled as the state in which the *process* TANK is. Demand is controlled by the *subprocess* DC, which changes the state DEMAND is in nondeterministically.

3.2 Vulnerability Identification using LTSA Safety Checks

The LTSA model checker was applied to the discrete FSP model that was developed for Mini-Town to perform a property check and a progress check, which identifies safety violations and unreachable states, respectively. The initial check was performed with no threats present to ensure that the control system receives and sends all communications as intended, and that the physical processes, TANK, PUMP, and DEMAND, entered all states as designed. Output from the LTSA safety check and progress check are shown in Figures 11 and 12. These figures demonstrate that the model was verified without generating progress violations or deadlocks.

```

SYSCONTROL = (start -> SYSCONTROLRUN[A][B][C][Tank]),
SYSCONTROLRUN[pi:PumpState][di:DemandState][hi:TankState][ni:N] =
(stop -> SYSCONTROL
|pumpchan.receive[p:PumpState] -> tankchan.receive[h:TankState] -> demandchan.receive[d:DemandState] -> HC[p][d][h][Tank]),
HC[p:PumpState][d:DemandState][h:TankState][ni:N] =
(when(p==0 && d==1 && ni==Tank) tankcontrollerchan.send[0] -> PC[p][d][h][Pump]
|when(p==0 && d==2 && ni==Tank) tankcontrollerchan.send[-1] -> PC[p][d][h][Pump]
|when(p==0 && d==3 && ni==Tank) tankcontrollerchan.send[-1] -> PC[p][d][h][Pump]
|when(p==1 && d==1 && ni==Tank) tankcontrollerchan.send[1] -> PC[p][d][h][Pump]
|when(p==1 && d==2 && ni==Tank) tankcontrollerchan.send[0] -> PC[p][d][h][Pump]
|when(p==1 && d==3 && ni==Tank) tankcontrollerchan.send[-1] -> PC[p][d][h][Pump]
|when(p==2 && d==1 && ni==Tank) tankcontrollerchan.send[1] -> PC[p][d][h][Pump]
|when(p==2 && d==2 && ni==Tank) tankcontrollerchan.send[0] -> PC[p][d][h][Pump]
|when(p==2 && d==3 && h<=2 && ni==Tank) tankcontrollerchan.send[0] -> PC[p][d][h][Pump]
|when(p==2 && d==3 && h>=3 && ni==Tank) tankcontrollerchan.send[-1] -> PC[p][d][h][Pump]),

PC[p:PumpState][d:DemandState][h:TankState][ni:N] =
//0 pumps on when tank height is 6,7
//1 pump on when tank height is 3,4,5
//2 pumps on when tank height is 1,2
(when(h==1 && p==2 && ni==Pump) pumpcontrollerchan.send[3] -> DC[p][d][h][Demand]
|when(h==1 && p!=2 && ni==Pump) pumpcontrollerchan.send[2] -> DC[p][d][h][Demand]
|when(h==2 && p==2 && ni==Pump) pumpcontrollerchan.send[3] -> DC[p][d][h][Demand]
|when(h==2 && p!=2 && ni==Pump) pumpcontrollerchan.send[2] -> DC[p][d][h][Demand]
|when(h==3 && p==1 && ni==Pump) pumpcontrollerchan.send[3] -> DC[p][d][h][Demand]
|when(h==3 && p!=1 && ni==Pump) pumpcontrollerchan.send[1] -> DC[p][d][h][Demand]
|when(h==4 && p==1 && ni==Pump) pumpcontrollerchan.send[3] -> DC[p][d][h][Demand]
|when(h==4 && p!=1 && ni==Pump) pumpcontrollerchan.send[1] -> DC[p][d][h][Demand]
|when(h==5 && p==1 && ni==Pump) pumpcontrollerchan.send[3] -> DC[p][d][h][Demand]
|when(h==5 && p!=1 && ni==Pump) pumpcontrollerchan.send[1] -> DC[p][d][h][Demand]
|when(h==6 && p==0 && ni==Pump) pumpcontrollerchan.send[3] -> DC[p][d][h][Demand]
|when(h==6 && p!=0 && ni==Pump) pumpcontrollerchan.send[0] -> DC[p][d][h][Demand]
|when(h==7 && p==0 && ni==Pump) pumpcontrollerchan.send[3] -> DC[p][d][h][Demand]
|when(h==7 && p!=0 && ni==Pump) pumpcontrollerchan.send[0] -> DC[p][d][h][Demand]),

DC[p:PumpState][d:DemandState][h:TankState][ni:N] =
(demand[1] -> SYSCONTROLRUN[p][d][h][Tank]
|demand[2] -> SYSCONTROLRUN[p][d][h][Tank]
|demand[3] -> SYSCONTROLRUN[p][d][h][Tank]).
    
```

Figure 10. FSP snippet of Mini-Town model

```

Edit Output Draw
Progress Check...
-- States: 2509 Transitions: 5248 Memory used: 19291K
No progress violations detected.
Progress Check in: 22ms
    
```

Figure 11. Progress check of Mini-Town using LTSA

```

Edit Output Draw
Composition:
SYS = TANK || TANKSENSOR || TANKACTUATOR || DEMAND || DEMANDSENSOR || PUMP || PUMPSENSOR ||
PUMFACTUATOR || TIME || SYSCONTROL
>> tick
State Space:
15 * 16 * 6 * 7 * 8 * 7 * 8 * 7 * 2 * 1765 = 2 ** 38
Analysing using Supertrace (Depth bound 100000 Hashtable size 8000K)...
-- Depth: 0 States: 2509 Transitions: 5248 Memory used: 16872K
No deadlocks/errors
Analysed using Supertrace in: 7ms
    
```

Figure 12. Safety check of Mini-Town using LTSA

A communication threat was simulated in the FSP model of Mini-Town and the safety checking tool in LTSA was used to perform an exhaustive check for any system vulnerabilities. The communication threat is modeled as a *process* that sends faulty sensor readings to the system control. This model asserts a safety condition that reports the level of water in the tank never



drops below state one, even when this is an inaccurate report. Output from the LTSA safety check and progress check indicate that when a communication threat is present, the safety condition of TANK is violated. This communication threat represents a vulnerability in the system. Performing a trace through the safety violation characterizes the vulnerability: the communication threat intercepts the tank's sensor reading through the *process* TANKSENSOR and send a reading of "1" regardless of the actual level of water in the tank. This reading is sent to SYSCONTROL, which sends the command to PUMP to turn both pumps on and remain on, filling the tank above its threshold, causing it to overflow, represented as an ERROR STATE in FSP.

4 CONCLUSIONS

Research in cybersecurity attacks on WDSs focuses on attack detection and network resilience. The existing literature describes research directions that rely exclusively on hydraulic simulation, and may therefore miss difficult-to-find corner cases. A formal methods approach to identifying cybersecurity vulnerabilities in water distribution systems, such as finite state processes, augments traditional simulation approaches with safety and progress checks that can be automatically performed with model checking tools such as LTSA. When a threat is present in the model, LTSA can identify the vulnerabilities in the system by finding traces that violate safety and progress conditions. The identification of vulnerabilities through finite state processes can be used to inform utility managers of potential security system needs or upgrades, and as a reference for attack scenario detection and network resilience analysis.

5 REFERENCES

- [1] N. Tuptuk, P. Hazell, J. Watson, and S. Hailes, "A Systematic Review of the State of Cyber-Security in Water Systems," *Water (Switzerland)*, vol. 13, no. 1. MDPI AG, Jan. 01, 2021. doi: 10.3390/w13010081.
- [2] A. Rasekh, A. Hassanzadeh, S. Mulchandani, S. Modi, and M. K. Banks, "Smart Water Networks and Cyber Security," *Journal of Water Resources Planning and Management*, vol. 142, no. 7, p. 01816004, Jul. 2016, doi: 10.1061/(ASCE)WR.1943-5452.0000646.
- [3] J. Lin, S. Sedigh, and A. Miller, "Towards Integrated Simulation of Cyber-Physical Systems: A Case Study on Intelligent Water Distribution," 8th IEEE International Symposium on Dependable, Autonomic and Secure Computing, DASC 2009, pp. 690–695, 2009, doi: 10.1109/DASC.2009.140.
- [4] K. B. Adedeji and Y. Hamam, "Cyber-Physical Systems for Water Supply Network Management: Basics, Challenges, and Roadmap," *Sustainability (Switzerland)*, vol. 12, no. 22, pp. 1–30, Nov. 2020, doi: 10.3390/SU12229555.
- [5] R. M. Clark, S. Panguluri, T. D. Nelson, and R. P. Wyman, "Protecting Drinking Water Utilities from Cyberthreats," *Journal AWWA*, vol. 109, no. 2, 2017.
- [6] E. Z. Berglund et al., "State-of-the-Art Review Smart Infrastructure: A Vision for The Role of the Civil Engineering Profession In Smart Cities," 2020, doi: 10.1061/(ASCE)IS.1943-555X.0000549.
- [7] "EPANET | US EPA." <https://www.epa.gov/water-research/epanet> (accessed Apr. 03, 2022).
- [8] R. Taormina, S. Galelli, H. C. Douglas, N. O. Tippenhauer, E. Salomons, and A. Ostfeld, "A Toolbox for Assessing The Impacts Of Cyber-Physical Attacks on Water Distribution Systems," *Environmental Modelling and Software*, vol. 112, pp. 46–51, Feb. 2019, doi: 10.1016/j.envsoft.2018.11.008.
- [9] M. Sirjani, E. A. Lee, and E. Khamespanah, "Verification of Cyberphysical Systems," *Mathematics*, vol. 8, no. 7, Jul. 2020, doi: 10.3390/MATH8071068.
- [10] J. Voas and K. Schaffer, "Insights on Formal Methods In Cybersecurity," *Computer (Long Beach Calif)*, vol. 49, no. 5, pp. 102–105, May 2016, doi: 10.1109/MC.2016.131.
- [11] "LTSA - Labelled Transition System Analyser." <https://www.doc.ic.ac.uk/ltsa/> (accessed Apr. 04, 2022).
- [12] J. Magee and Jeff. Kramer, *Concurrency: State models & Java programs*. Wiley, 2006.

- [13] “12. Analysis Algorithms — EPANET 2.2 documentation.”
https://epanet22.readthedocs.io/en/latest/12_analysis_algorithms.html (accessed Apr. 05, 2022).
- [14] J. E. Pesantez, E. Z. Berglund, and G. Mahinthakumar, “Multiphase Procedure to Design District Metered Areas for Water Distribution Networks,” *Journal of Water Resources Planning and Management*, vol. 145, no. 8, p. 04019031, Aug. 2019, doi: 10.1061/(ASCE)WR.1943-5452.0001095.

