IoT Security and Safety Testing Toolkits for Water Distribution Systems

Sean O’Toole, Cameron Sewell, Hoda Mehrpouyan

Department of Computer Science
Boise State University
Boise, United States
seanotoole@u.boisestate.edu, cameronsewell@u.boisestate.edu, hodamehrpouyan@boisestate.edu

Abstract—Due to the critical importance of Industrial Control Systems (ICS) to the operations of cities and countries, research into the security of critical infrastructure has become increasingly relevant and necessary. As a component of both the research and application sides of smart city development, accurate and precise modeling, simulation, and verification are key parts of a robust design and development tools that provide critical assistance in the prevention, detection, and recovery from abnormal behavior in the sensors, controllers, and actuators which make up a modern ICS system. However, while these tools have potential, there is currently a need for helper-tools to assist with their setup and configuration, if they are to be utilized widely. Existing state-of-the-art tools are often technically complex and difficult to customize for any given IoT/ICS processes. This is a serious barrier to entry for most technicians, engineers, researchers, and smart city planners, while slowing down the critical aspects of safety and security verification. To remedy this issue, we take a case study of existing simulation toolkits within the field of water management and expand on existing tools and algorithms with simplistic automated retrieval functionality using a much more in-depth and usable customization interface to accelerate simulation scenario design and implementation, allowing for customization of the cyber-physical network infrastructure and cyber attack scenarios. We additionally provide a novel in-tool-assessment of network’s resilience according to graph theory path diversity. Further, we lay out a roadmap for future development and application of the proposed tool, including expansions on resiliency and potential vulnerability model checking, and discuss applications of our work to other fields relevant to the design and operation of smart cities.

Index Terms—smart cities, internet of things, operational technology, industrial control systems, critical infrastructure, simulation, toolkits, software development, safety, resilience

I. INTRODUCTION

As a result of general increases in malicious cyber activity in recent years, and in the wake of STUXNET, the cyber security of industrial control systems is increasingly becoming a massive security concern for cities, companies, and nations [1], [2]. The most prominent types of attacks have usually targeted the Information Technology (IT) components of a system. These attacks are usually conducted for financial gain and usually do not cause long term damage to the systems being affected, especially if there is a significant separation between data management and the actual operational technology (OT). For example, the recent Colonial Pipeline attacks, which were IT-based and targeted billing information. Colonial themselves shut down their own pipeline until the ransomware issue was resolved and there was no structural damage done to the pipeline [3]. If, however, they were targeted with an OT-focused attack that took control of operational devices, sensors, or actuators within the ICS, the attackers could have created serious physical damage in the over 5,500-mile-long pipeline that could have caused millions of dollars damage and left up to 45 percent of the East Coast of the United States without gas for an undetermined amount of time.

An example of the full potential damage of an OT attack could be seen in the 2007 INL Aurora Generator test [4], which created abnormal vibrations in a diesel generator within 13 iterations of the attack loop and eventually caused the engine to dramatically fail. Manipulation of the control logic which controlled the generator led to significant physical damage, a methodology that would be similarly applied in the cyber worm STUXNET several years later in an attack against the country of Iran’s nuclear refinement capabilities to significant effect.

In the years since, an increasing number of attacks have sought to utilize tools like this against city and regional infrastructure. In 2015, several regions of Ukraine’s energy grid were affected by an attack, leaving 225,000 people without power. And again in 2016 a power grid hack affecting a key substation shut down one fifth of power traffic in the city of Kyiv. Following the 2016 attack the President of Ukraine, Petro Poroshenko, stated that state institutions had targeted about 6,500 times in the final months of that year [5]. 2017’s TRITON attacks against Saudi Arabian natural gas processing plants represented a near-miss, where the potential for serious physical damage and threat to human life was only avoided by a single fail-safe missed by attackers [6]. In regards to water distribution systems, the Oldsmar Water Plant Attack in Feb. 2021 is the most recent example [7], following on the heels of the April 2020 attacks against multiple regional water systems in Israel [8], both of which sought to dump toxic levels of chlorine into the water, presenting a serious threat to the health and well being of citizens. However, OT threats to water go back further, all the way to the 2000 Maroochy water plant incident, in which hundred of thousands of gallons of untreated sewage were dumped back into local water supplies by a single attacker with access to OT systems.

Funding provided under a grant from the National Science Foundation.
Many fields have incorporated simulation into their education and training to yield well-educated individuals at lower cost and less danger to the individual in-the-field training. The precursor to modern simulation training was first tested in 1978 by the National Aeronautics and Space Administration, who found in their studies of disasters that most aviation accidents involved a lack of leadership coordination, or decision making [12]. Simulations also spread to the maritime world focusing on on the same issues. Most high fidelity maritime simulations reproduce the pilot or captains room and the engine room to inundate the crew with the auditory and visual factors they will need to expect while on the job [13]–[15]. The formalization of simulation-based educational practices has advanced in the medical industry as well due to the development of international and national multidisciplinary societies, usually aimed at a broad membership of healthcare educators, clinicians, researchers and engineers, and supported by industry [16]. These simulation training sessions better prepare individuals for the needs of different scenarios than standard training, reducing accidents and waste. In all cases, simulation could be applied to improve safety outcomes for both those doing the work and those relying on their efforts.

Within those subset of simulators targeting the needs of smart cities and considering the role of IoT devices within those cities, and which are seeing ongoing development and maintenance, several exist which allow for relatively thorough simulation of various networks, power, and IoT simulation. Simulation of Urban Mobility (SUMO) is one of the oldest and best examples of the kind of toolset that can be of great benefit to a city, offering a powerful tool for analyzing road networks and traffic patterns and experimenting on them, before returning to users simulated traffic data, physical effect data, and improved road networks if desired by the user [17].

Bonceur’s CupCarbon [18] is perhaps the best example of a modern smart city simulation tool with the capability of modeling real-world scenarios and their impacts on and interactions with a specific subset of city infrastructure. Bounceur proposed and later presented CupCarbon in 2017 as a flexible simulation tool focused on radio propagation channels and IoT interactions within a user-designated network and set against real-world maps. CupCarbon allows for the simulation of wireless network behavior and communication with a large number of different components and customizable parameters, and most notably allows for designing scenarios in which real-world events such as fire incidents and gas leaks can be simulated, with the event having an effect on the ability of the network to communicate clearly. Custom python scripts can be written for different nodes, allowing for potentially limitless variety and customization of system behavior. However, with few examples and no capability of importing existing files on which to base designs, the tool is currently inhibited by its lack of user accessibility and the lack of training available for it.

It is our goal with this work to help streamline the process of simulation in industrial control system security and infrastructure resilience, as well as bring better physical systems simulation and OT more into focus when discussing smart city infrastructure and security.
III. STATE-OF-THE-ART SAFETY AND SECURITY TESTING TOOLBOXES IN WDS

A variety of open-source tools have been created to address the threat of OT attacks, specifically in water distribution. Among these, the two we find to be most significant in terms of contribution and potential for future development in recent years are Taormina et al.’s epanetCPA [19] and Murillo et al.’s DHALSIM [20]. Taormina et al. created epanetCPA, a MATLAB based program for modeling the response of water systems to cyber-physical attacks that runs on top of EPANET. EPANET is used worldwide by engineers and researchers to design new water infrastructure, update existing water systems, and develop more efficient solutions to solve water quality problems. It is a Windows based software application for simulating and representing water distribution systems. It uses a network configuration or .inp file which represents a physical water system and creates a digital representation of the water network, representing all pipes, junctions, tanks, reservoirs, valves, and pumps. epanetCPA works in tandem with EPANET modifying the input file with information from a user provided cyber-physical attack file, .cpa. The changes in file information create a new data set separate from the ground truth and displayed by the EPANET model analysis. Murillo et al. in “Co-Simulating Physical Processes and Network Data for High-Fidelity Cyber-Security Experiments” introduced Digital Hydraulic Simulator (DHALSIM) [21] to show the physical processes, control logic, and network communication of a cyber-physical system while it is under attack. DHALSIM provides users a full network capture of the PLCs, SCADA systems, and any other network device present on the system.

However, although both toolkits offer considerable utility, creating custom scenarios and models for each requires considerable work on the part of the user, and the prerequisite knowledge of the programming languages and/or file formats in use within each toolkit. As can be seen in Fig. 2 and

Fig. 2. File requirements for epanetCPA configuration. Red indicates user creation required.

Fig. 3. File requirements for DHALSIM configuration. Red indicates user creation required.
multiple files of different formats and languages are required for each of epanetCPA and DHALSIM, although the added complexity of DHALSIM in attempting to co-simulate network behavior results in requiring significantly more custom files and configurations. This is a non-trivial issue for research and application of these toolkits going forward, if they are to be utilized for safety and security testing and training. While some understanding of programming is not an unreasonable expectation for users, the current toolkits require an in-depth understanding of the toolkits themselves, along with the formats and languages in use, in order to customize to any significant degree. Cutting down on this requirement would make them vastly more approachable and usable. We believe that inp2cpa and expansion of its current functionality, and the creation of tools like it for other simulation toolkits in WDS and general ICS simulation, is a critical part of lowering that barrier to entry.

IV. inp2cpa: A TOOL FOR RAPID INTEGRATION

A. inp2cpa Basic Functionalities

As mentioned, the epanetCPA toolkit requires an engineer to have knowledge on how to extract relevant data from .inp files and use that information to create .cpa files. Learning these file types and their syntax can be difficult and time consuming to correctly implement for those without experience with relevant field-specific programs, as well as the toolkits themselves. To make these toolkits more accessible we have taken and extended the functionalities of the tool inp2cpa, created by Nikolopoulos [11], [22] and the EU STOP-IT team, to drastically reduce the time it takes to run bulk test simulations as well as create custom cyber-physical attacks to simulate on a water network. This tool is a python-based program that utilizes the WNTR toolkit [23]. WNTR is a python package created by the EPA and Sandia Labs to evaluate drinking water systems for water quality and resilience against damage cause by demand, natural disasters, and cyber attacks. WNTR is capable of generating water network models, modifying network structure, evaluating disruptive events, simulating pressure driven and demand driven hydraulics. It is used by engineers in the infrastructure industry as well as other cybersecurity researchers around the world. inp2cpa uses WNTRs file parsing capabilities in its network.controls script to build a python dictionary using the control section of .inp files. With this dictionary inp2cpa automatically generates a basic .cpa file. We have expanded inp2cpa to allow users to modify the automatically generated cyber-attack file by adding Cyberlinks and Cyberattacks. A Cyberlink is a way that a user can show how their hardware is connected. For example, if a series of actuators where connected to one plc, a user could add cyberlinks to the .cpa show the link between each actuator and its plc and then the plc to its SCADA system. Adding these links will allow the user to simulate attacks that take advantage of or disrupt those connections. The user can now add four types of cyber-attacks:

- Communication - intercepting and replacing communications between hardware
- Control - Changing the control logic of how physical components operate
- Sensor - manipulates the values sent out by sensors
- Actuator - take control of a physical component (e.g. valves, pumps) and manipulate it

We have simplified the process of creating these attacks by getting key information like the names of the sensor to be attacked and what values the want to interject into that sensor and then using string manipulation and concatenation we add it to a .cpa file with the correct syntax (as specified in Munteanu et al. [24]) and create an attack file ready for use. inp2cpa is meant to be used specifically with epanetCPA, and our contribution has expanded on the original automated processing of controls by completing several of the originally-proposed features, including customization and control menus, as well as numerous fixes to the text-parsing and error-checking features of the original code. We intend, however to make it compatible with DHALSIM as well by adding YAML, “Yet Another MarkUp language”, used for DHALSIM configuration files, as an output, along with potentially other languages for file conversion and creation.

B. inp2cpa: Resiliency Checking On-the-Fly

In order to allow in-application review of the user’s networks against common network resiliency metrics, inp2cpa creates a $G(V,E)$ graph theory model of the network as defined by the user, using the general form of:

- $\text{Nd} \{N_{d}, L_{n}\}$
- $\text{Nd} \{S, A, Con\}$
- $\text{Ln} \{S_{o}, De, S\}$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4.png}
\caption{inp2cpa Base Menu and Attack Selection Window}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig5.png}
\caption{inp2cpa Sensor Attack Specification Window}
\end{figure}
Fig. 6. Control section of an .inp file

```
[CONTROLS]
LINK WWT0FCV0001 120.000000 If Node WWT0TEK0001 Below 5.500000
LINK WWT0FCV0001 Closed If Node WWT0TEK0001 Above 8.500000
LINK WWT0MPF0101 Open If Node WWT0TEK0002 Below 5.500000
LINK WWT0MPF0101 Closed If Node WWT0TEK0002 Above 8.500000
```

Fig. 7. Post-processing of Fig. 4 within inp2cpa, resulting .cpa file, including user-specified attack against communication link.

```
[CYBERNODES]
; Name, Sensors, Actuators
PLCL1, P_WWT0TEK0001,
PLCL2, P_WWT0FCV0001 S_WWT0FCV0001, WWT0FCV0001
PLCL3, P_WWT0MPF0101 S_WWT0MPF0101, WWT0MPF0101
PLCL2, P_WWT0TEK0002
; SCADA

[CYBERLINKS]
; Source, Destination, Sensors
PLCL1, SCADA, P_WWT0TEK0001
PLCL1, PLCL2, P_WWT0FCV0001
PLCL2, SCADA, P_WWT0TEK0002
PLCL1, PLCL2, P_WWT0FCV0001
PLCL1, PLCL2, P_WWT0MPF0101
PLCL2, SCADA, P_WWT0FCV0001 S_WWT0FCV0001
PLCL3, SCADA, P_WWT0MPF0101 S_WWT0MPF0101

[CYBERATTACKS]
; Type, Target, Init_coord, End_coord, Arguments
; Attack on communication link between PLC1 and PLC2, represent the control nodes PLC1 and PLC2, responsible for respectively: monitoring pump flow and junction pressure on the first three pumps in the network; monitoring water reserve tank level for the corresponding pumped reserve tank. This subnet of the overall network represents just under half of the overall network capacity and complexity, and can be seen in in Fig.8 However, while it represents a high level of physical complexity, the logical complexity is often significantly lower. As shown in Taormina et al. [19] in their cybernode and link layouts for the ctown attack scenarios, a common way this might be handled is simply by assigning one PLC to monitor tank levels and separately, one to control the pumps feeding the network and monitor pressure levels on either side of the pumps for potential issues. However, as demonstrated by their own attack simulations against this set of control components, such a simple design has effectively no resiliency against any attack which interrupts or replaces valid communication between components, nor did they attempt to quantify or otherwise assess the resilience of cyber components of the network.

The results of this attack as simulated in Fig. 9 show that even interrupting one communication link between the sensor-monitoring PLC and the pump-controlling PLC will result in the system being entirely unable to identify and respond to conditions in the tank. While pumps should have been disconnected at 58 hours, the pumps continue to run and create overflow. Either duplicate tracking on the relevant sensor by the pump PLC or additional nodes (for example, a SCADA unit) tracking and sending data to nodes waiting on information would have resolved this, but the total lack of either results in likely damage to the system. While total duplication and redundancy across all nodes and controls in a network is impractical financially, and introduces both considerable extra complexity and potential additional surface area for attackers to target, we believe that even a small amount of additional connectivity and built-in resiliency in design would contribute significant value here, if only by ensuring that multiple attacks or interruptions would need to occur simultaneously to guarantee significant impact on the system.

In addressing this gap, we utilized the aforementioned graph theory model [25]–[27] alongside the findings of Alenazi et al. [28], in which Total Graph Diversity (TGD) [1] was identified across multiple structured and random network designs to be the most effective predictor of network resiliency against outages and directed attacks, to evaluate different configurations of the cyber-physical network. This metric, which is first described in Rohrer et al. [29], is measured as the average of Effective Path Diversity (EPD) [2] representing lowest path
diversity over a set of paths through the network, in this case over the directed edge links from node to node. Path diversity is determined for any given path among a set of possible paths between two nodes by a simple equation evaluating path length against shortest possible path length. EPD additionally factors in a $\lambda$ value which is decided upon for any given network based on experimentation and desired outcomes. As a result of the exponential nature of the EPD formula, greater $\lambda$ values correlate with a reduced impact of $K_{sd}$ (minimum path diversity between two points in a graph) values on the outcome, indicating reduced gains on resiliency for additional path diversity.

Pseudocode for a simple implementation of each can be found in 1 and 2.

Algorithm 1 TGD(Graph):

```
sum ← 0
cnt ← 0
for node_s in Graph do
    for node_d in (Graph − node_s) do
        cnt ← cnt + 1
        sum ← sum + EPD(Graph, node_s, node_d)
    end for
end for
return sum/cnt
```

Algorithm 2 EPD(Graph, node_s, node_d):

```
if Graph contains node_s, node_d then
    k_{sd} ← 0
    best ← []
    p0 = ShortestPath(Graph, node_s, node_d) 1
    for path in AllPaths(Graph, node_s, node_d) do
        if path ≠ p0 and 1 − \frac{size(p0 \oplus \text{path})}{size(path)} > t_k_{sd} then
            best = path
            if size(best) == 0 then \triangleright Path diversity is 0
                k_{sd} ← 0
            else
                k_{sd} ← 1 − \frac{size(p0 \oplus \text{best})}{size(best)} > t_k_{sd}
            end if
        end if
    end for
    return 1 − \exp^{−\lambda\times K_{sd}}
else
    return 0 \triangleright One or more nodes DNE in network.
end if
```

As both ShortestPath and AllPaths are well-known graph theory algorithms which have numerous possible implementations, we note that any of the commonly-used approaches can be applied to this problem.

scores as nodes are added and connectivity increased. The base network scenario with limited connectivity, a fully-connected variation, and two variations on the network with a duplicate tracker on the tank level sensor were run through TGD checking, with the results shown in Table 1 below.
You can see that TGD weights heavily on interconnectivity, encouraging two-way communication and checking between cyber nodes. At all $\lambda$ values, it rewards additional connectivity, and penalizes where the number and directness of paths are limited, although additional connectivity is much more heavily rewarded for higher levels of $\lambda$. It can be seen that simply adding a handful of connections to the base logical model, predicted resiliency (correlated with TGD) can be greatly improved. However, we must also note the severe penalization for the additional duplicate node when said node is not fully connected—this is not reflective of the likely effect on resiliency, since resiliency should not be negatively impacted by the addition of backup/duplicate sources and checks. This indicates that while TGD can be tested for networks in development and used to estimate relative resilience, additional metrics should be tested and integrated in future releases of inp2cpa and any future publications. Nevertheless, resiliency metrics like this are relevant to both standard communication and critical infrastructure control systems networks, and this additionally allows users an initial assessment of the likely resiliency of their network before lengthy simulations and tests are applied.

V. Conclusion and Future Work

The extended functionalities of the tool inp2cpa drastically reduces the time it takes to create and run bulk test simulations, as well as enabling technicians and engineers to create custom cyber-physical attacks to simulate on a water network and investigate the impact of the attack. This will be beneficial to engineers and technicians who work with critical infrastructure but do not have the necessary programming experience to efficiently use EPANET and epanetCPA. Future additions will include features and functionality to expand this to include DHALSIM configuration and attack scenario creation, along with a deeper dive into relevant existing resiliency metrics, and potential development of custom resiliency metrics for cyber-physical and smart-device systems based on their unique characteristics. We plan to continue to add to inp2cpa to make it as useful as possible to technicians and engineers in the industry. We are currently laying the ground work to add YAML functionality, and are also considering adding python configuration and device file generation, and will consult with members of industry and technical experts to identify what they would see as the most potentially useful additions to any such tool aiming to provide utility and create ease-of-access to simulation and verification toolkits. We believe one of the best ways to move forward with this tool is to eventually migrate it from a guided file creation tool to a fully automated file converter. While some components, such as cyberlinks, cannot be inferred by us and require user input we believe most other functions can be at least in-part, if not fully, automated. Eventually we would also like to see this tool, or core automated components of it, integrated into a simulation toolkit like epanetCPA or DHALSIM. If we were able to integrate this tool with an existing simulation toolkit that would significantly improve the accessibility of that toolkit to technicians in industry and researchers looking to advance knowledge in the field. inp2cpa, and tools like it, will make securing critical infrastructure from cyber-physical threats quicker and easier.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation Computer and Information Science and Engineering (CISE) division, award number 1846493 of the Secure and Trustworthy Cyberspace (SaTC) program: Formal TOols foR SaEty aNd. Security of Industrial Control Systems (FORENSICS).
