DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

SAND2025-12888R

SCEPTRE: A Cyber-Physical Emulation Capability

Meghan Sahakian, Christopher Abate, Christopher Goes, Clifton Mulkey, James Ryan, Zachary Thomas, Jamie Thorpe

October 2025

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

ABSTRACT

Cyber-physical systems form a critical but vulnerable backbone to US critical infrastructure. Recent high-profile cyber-attacks have shown the need for increased assessment and hardening of these systems. However, such assessments and investigations into advanced technologies to harden these systems is difficult due to their operational nature. Instead, modeling of these systems is heavily leveraged. Investigation into these complex systems and their potential cascading failures requires comprehensive modeling of both the cyber and physical components of the system. This paper introduces SCEPTRE, an emulation capability to address this need.

CONTENTS

LIST OF FIGURES

ACRONYMS AND DEFINITIONS

Abbreviation	Definition
API	Application Programming Interface
COTS	Commercial-Off-The-Shelf
GUI	Graphical User Interface
HIL	Hardware-In-the-Loop
HMI	Human-Machine Interface
ICS	Industrial Control System
IT	Information Technology
DCS	Distributed Control System
FEP	Front-End Processor
KVM	Kernel-Based Virtual Machine
OPC	Open-Platform Communication
OT	Operational Technology
PLC	Programmable Logic Controller
QEMU	Quick Emulator
R&D	Research and Development
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SDN	Software Defined Networking
SOAP	SCEPTRE-On-A-Platter
VM	Virtual Machine
VNC	Virtual Network Computing

This page left blank.

1. INTRODUCTION

A cyber-physical system consists of one or more physical processes (power, water distribution, fuel, etc.) integrated with a computer network containing components that monitor and control the physical process [2]. The term cyber-physical system encompasses many types of systems including industrial control systems (ICSs), operational technology (OT) systems, supervisory control and data acquisition (SCADA) systems, distributed control systems (DCSs), among many others. Disruptions to these systems can seriously hinder operations and result in catastrophic consequences such as significant interruption to operation, monetary loss, compromise of sensitive data, and even loss of life.

However, these systems are also often outdated, legacy systems that have been pieced together over the years without cyber-security in mind. This leaves the systems vulnerable to cyber-attack. In 2010, the Stuxnet malware attack first demonstrated the targeting of cyber-physical systems with a cyber-attack during which the centrifuges at a uranium enrichment facility were disabled [44]. Subsequently, similar attacks have been repeatedly demonstrated including attacks on the power grid in Ukraine [28], chemical manufacturing facilities in the Middle East [16], and elsewhere. Recent cyber-attacks demonstrate a willingness and ability to impact critical infrastructure in the United States; a cyber-attack in 2021 to an information technology (IT) system shutdown the Colonial Pipeline and demonstrated the consequences of a cyber-attack on US critical infrastructure systems even if the attack itself didn't reach the cyber-physical portion of the system [7].

As a result, many system owners have concerns about their operational cyber-physical systems and are seeking to: (1) discover, characterize, and quantify their most critical vulnerabilities, (2) examine tradeoffs and quantify risk reduction to avoid negative impacts, (3) prioritize threats, (4) construct and evaluate resilience strategies and mitigations, (5) evaluate the impact of potential new devices, technologies, operational environments and architectures, (6) anticipate future threats [14, 15, 24, 30].

Physical tests to address these types of questions are rarely performed because they are costly and risk interfering with critical operations of the systems under test. Therefore, analysis of cyber-physical systems heavily relies on the use of physical testbeds, emulated testbeds, simulations, or other types of modeling environments.

Physical testbeds offer the greatest realism to the actual system, using both real hardware and software. However, these testbeds are typically costly to operate and maintain, require long lead times to configure, and are at risk for physical damage depending on the type of attack/analysis being performed. Emulated testbeds are virtualized environments which use abstract, virtualized hardware capable of running real software. These testbeds lack the full realism of physical testbeds/real systems but are more flexible to configure and operate. Simulations are typically simplified representations of the system being modeled and focus on the high-level behaviors of the system. These types of testbeds are often quickest to configure and run but offer the least amount of realism. The choice of testbed type is critical because it governs the type of analysis that can be performed.

This paper introduces SCEPTRE, an open-source emulation capability for cyber-physical systems that bridges the gap between the cyber and the physical, through coupling of control process emulation (and/or physical control devices) and physical process simulation. SCEPTRE employs proven technologies and techniques to mesh end device and process simulations, with control hardware, providing an integrated system capable of representing realistic responses in a physical

process as events occur in the control system, and vice versa [40]. This cyber-physical emulation capability provides a less costly, but still comprehensive testbed for cyber-security analyses.

This paper is organized as follows: Section 2 identifies driving needs for cyber-physical testbeds and gives a background on existing testbeds. Section 3 discusses the various applications for the SCEPTRE capability. Section 4 provides an in-depth discussion of the components of SCEPTRE. Section 5 describes a use case to demonstrate one of the many applications of SCEPTRE. Finally, Section 6 discusses conclusions and future work.

2. CYBER-PHYSICAL TESTBEDS

The driving need for the development of SCEPTRE was to create a platform that could model cyber-physical systems to analyze how cyber-initiated events affect the physical world and vice-versa. Based on this need, several requirements for a capability arose:

- Configurable
- Flexible
- Repeatable and reproducible
- High-fidelity SCADA components
- Cyber and physical interdependencies

There exist many survey papers that discuss the numerous testbeds that exist for cyber-physical systems. This section will summarize some of the relevant findings and compare them to the requirements the SCEPTRE capability was addressing [12, 22, 35].

Most SCADA testbeds specialize on either simulating or emulating control system field devices or on understanding the physical processes. While these specialized tools are useful, they lack the ability to identify security issues at the intersection of these two components. The differentiating characteristic of SCEPTRE is that it intertwines the device and process simulations, providing an integrated system capable of representing realistic responses in the physical process as events occur in the control system and vice versa. This allows SCEPTRE to treat control devices and logic the same as if they were installed in a cyber-physical system itself, providing a high level of fidelity.

The control side (i.e. cyber side) of common testbeds leverage simulation tools such as ns-3 [37], OPNET [10], DETER [8], Emulab [43], CORE [1], and SCADASim [36]. These simulations are based on simplified models that capture the desired behavior of components, but do not have the level of fidelity needed for investigating realistically complex cyber-physical interactions such as the studying of specific vulnerabilities or targeted network attacks. Some of these tools do couple the simulation of cyber networks with a model of the physical process, but the lack of fidelity on the cyber side is the limiting factor for cyber-security applications.

Other testbeds do include high fidelity network emulations and coupling to the physical process but lack fidelity and detail in some of the upstream software beyond the monitoring and control devices (i.e. a power plant's corporate devices) [13, 19]. These testbeds limit the flexibility and realism of experimenting with full cyber-physical systems.

Most cyber-physical system testbeds are also application specific. These testbeds meet current system needs from the designers but are short sighted focusing on specific applications which require extensive rework to be applied outside of their original application space. Current designs focus on hard-coded networking and configuration for specific applications and lack the flexibility and configurability needed for scalable, repeated testability across the large cyber-physical systems space. Examples of existing applications of testbeds range from power-grids [5, 6, 31, 33] to water treatment facilities [29] to manufacturing plants [27] to HVAC systems [21].

A few promising testbeds exist with similar requirements to SCEPTRE [3, 4, 32]. However, government only or commercial licenses limit the accessibility to these tools and the evaluation of the capabilities of these tools.

SCEPTRE addresses the identified requirements for a cyber-physical system testbed through coupling of virtualization of cyber components and physical process simulation. It is worth noting that using virtualization for this capability inherently creates a large computational workload and resulting hardware requirements depending on the size of the model desired. However, for most reasonable sized SCEPTRE models, a singe mid-sized server is sufficient. If there is a larger computational need, the virtualization technologies employed can seamlessly mesh models across multiple servers.

3. APPLICATIONS

The original goal of SCEPTRE was to enable deployment of cyber-physical system environments at scale for testing, evaluation, and analysis (i.e. to support assessments) within a virtual sandbox. Today, SCEPTRE has expanded in scope, scale, fidelity, and usability to support training and mission rehearsal, testing and evaluation, vulnerability and malware analysis, data analysis, and various research and development activities [9, 17, 18, 20, 23, 26, 38, 45, 46, 47].

For training and mission rehearsal exercises, SCEPTRE provides the necessary "look-and-feel" of real cyber-physical system networks by integrating industry and service SCADA tools and faithful protocol traffic for deeper network inspection. Ties to the end process simulation illustrate impacts of control system changes on physical systems. The environment also allows for replication and effects testing as they pertain to mission rehearsal.

SCEPTRE can be used to test and evaluate different hardware, architectures, or technology solutions. The emulation capability of SCEPTRE is much more configurable than a physical testbed, making it easy to create a variety of models depending on the scope of the questions being asked.

The high-fidelity nature of SCEPTRE can be used for vulnerability and malware testing. The emulated environment is a safe sandbox for conducting isolated tests of malware, and the ability to model how this impacts the physical process is an invaluable tool for evaluating mission impact to a system.

SCEPTRE can perform additional data analysis activities which include the identification of critical components on the control network and in the underlying infrastructure, consequence modelling, and real-time SCADA analysis.

There are myriads of other emerging R&D topics that create a necessary use case for SCEPTRE. These include model validation and verification, uncertainty quantification and many more.

This page left blank.

4. COMPONENTS OF SCEPTRE

SCEPTRE is part of an Emulytics™ tool-suite developed over decades at Sandia National Laboratories for government agencies [41]. SCEPTRE provides a means for creating large-scale control system test environments suitable for cyber-physical security experiments. Leveraging modeling, simulation, and test bed techniques, the test environments can be scripted to suit each experiment as necessary, are repeatable, and are much cheaper to construct than real or even lab-scale test environments. In addition, system disruptions and attacks that could damage real-world hardware can safely be run and tested.

The components of SCEPTRE are shown in Figure 1 and are typically referred to collectively as the "SCEPTRE Stack". Starting at the bottom, SCEPTRE runs end process simulations for the physical process of concern. SCEPTRE provides unique communication of data between the physical process simulation and the field devices within the control system. This allows the field devices to receive data from, and write data to, the physical-process simulation as if the devices are connected to and controlling a real-world physical process. SCEPTRE then uses software-defined networking to speak full-fidelity SCADA protocols between the field devices and any desired SCADA software. Since SCEPTRE uses virtualization, any desired industry standard software can be installed and configured within the environment. Real hardware can be integrated with the virtual environment as Hardware-In-the-Loop (HIL) to interact with the virtual control network and physical end process simulation. Finally, phēnix is SCEPTRE's orchestration tool which allows users to configure and quickly deploy environments for experimentation in a consistent, repeatable, and automated fashion, which includes a web user interface for setting up and running experiments. Overall, SCEPTRE provides the look-and-feel of real cyber-physical system networks and software to users of the tool while having the flexibility and scalability of a virtualized environment. The following subsections describe each of the components of SCEPTRE in more detail.

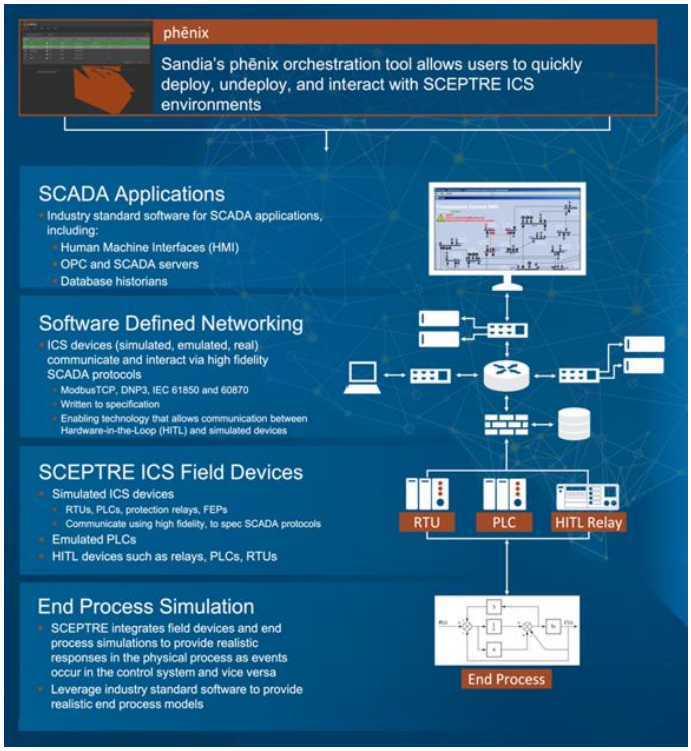

Figure 1: The SCEPTRE stack.

4.1. End process simulation

SCEPTRE is capable of running different physical-process simulations including models of various infrastructures and industry standard modeling software packages. Historically, SCEPTRE has modeled power systems, batch processes (e.g., oil/gas refinery, water treatment, etc.), rail control, pipeline infrastructures, among others using software including PowerWorld, PowerWorld Dynamic Studio, PyPower, MATLAB Simulink, Real-Time Digital Simulator, OpalRT, Synergi and others.

The control-process side of SCEPTRE runs in real time, (i.e., all virtualized devices operate at real-world computing speeds). Therefore, special consideration is needed when integrating the physical-process simulation. Simulations that run in real-time need little adjustments to integrate with SCEPTRE. Simulations that run faster than real-time or simulations such as discrete event simulations can be appropriately controlled to run at the correct real-time rate. Simulations that run slower than real-time are difficult to integrate and alternative solutions should be considered.

Integration of a new type of physical-process infrastructure in SCEPTRE is straightforward and requires the user to define the components of the infrastructure (i.e., a power transmission system has generators, branches, busses, loads, etc.) and the typical data fields associated with those components (i.e., a generator has associated voltage, real power, reactive power, etc.).

New simulation software packages are relatively easy to integrate as long as they have a reliable Application Programming Interface (API). The user must implement a new "provider" in SCEPTRE which leverages the component's API to define how data is both read and written from the underlying simulation. This "provider" supplies physical-process simulation data to all the field devices and is responsible for processing any updates from the field device and updating the physical-process simulation accordingly. On the backend of SCEPTRE, this communication of data between the provider and field devices is done via a Publish/Subscribe model where each device subscribes to the datapoints that it would be physically connected to in the real-world. This occurs on a separate management network that is only used to communicate information between the physical-process simulation and the SCEPTRE field devices and can be thought of as analogous to sensor/actuator connections to a real physical process.

SCEPTRE is also able to bridge multiple infrastructures into the same experiment to show interdependencies not only between a control system and the physical process, but between multiple physical processes themselves. This facilitates the testing of complex infrastructure interdependencies and potential failures including cascading failures among interacting infrastructures and networks.

4.2. Field devices

Cyber-physical system field devices are connected to the "field" or physical process side of a system. Types of field devices include sensors, actuators, remote terminal units (RTUs), programmable logic controllers (PLCs), front-end processors (FEPs), protection relays, smart inverters and more. These devices are responsible for functions such as monitoring the physical process, processing logic, and ultimately controlling the physical process.

There are many different types and vendors of cyber-physical system field devices. For that reason, SCEPTRE generally takes the approach of using vendor and firmware agnostic virtual field devices. These virtual field devices contain all the functionality of real field devices while avoiding the complexity of device, vendor and firmware variability.

Depending on the realism and level of fidelity needed in the emulation, SCEPTRE can also integrate HIL. This is specifically used in scenarios where a device or firmware version is of particular concern and involves wiring a real hardware unit into the SCEPTRE emulation. This allows the user to include high fidelity components where they are needed without sacrificing scalability.

Current research in firmware virtualization (or rehosting) points toward a promising middle ground between the virtualized field devices and HIL [11].

Regardless of the implementation, all field devices in SCEPTRE interact with the physical-process simulation. They subscribe to the current state of the simulation, and when the simulation state updates, all devices receive the new current state, and thus, a common view of the simulation. If the devices are configured with control logic, they also provide updates back to the physical-process simulation. After an update to the physical process simulation, the field devices will see the newly reflected state of the simulation on the next cycle.

4.3. Software defined networking

SCEPTRE leverages software defined networking (SDN) to facilitate full fidelity protocols between the field devices and upstream SCADA software.

The SCEPTRE virtual field devices contain protocol stacks that enable them to speak the desired protocols. SCEPTRE supports both TCP/IP and serial protocols including Modbus, DNP3, BACnet, IEC 61850 and 60870. Integration of a new protocol only requires access to the full protocol stack and implementation of that protocol stack with the underlying code base of the virtual field device. Additionally, by design, HIL field devices are already capable of speaking a set of protocols and integration into SCEPTRE is seamless.

Network devices such a routers and firewalls are also included in the SDN layer, allowing communications to be realistically routed and controlled between sources and destinations.

4.4. SCADA applications

SCADA applications in cyber-physical systems include devices such as Open-Platform Communication (OPC) servers, SCADA servers, human-machine interfaces (HMIs), and data historians. The virtualized nature of SCEPTRE allows commercial-off-the-shelf (COTS) SCADA applications to be seamlessly integrated within the environment. Example SCADA software historically integrated with SCEPTRE includes Kepware TOP Server, mySCADA, Wonderware, and more.

Configuration of SCADA software can be inherently difficult, detailed and time consuming. The automated nature of SCEPTRE allows for users to specify the minimum configuration details necessary, and the underlying code of SCEPTRE can be automated to create configurations compatible with the desired SCADA software. This facilitates scalability and reusability allowing the user to focus on what they are modeling rather than hand configuring all the software.

4.5. phēnix

Phēnix is the main work horse of SCEPTRE and is an orchestration tool that allows users to seamlessly create experiments using SCEPTRE. The main responsibilities of phēnix are to manage model configuration files and deploy and clean up experiments.

Depending on underlying server infrastructure, SCEPTRE can run numerous parallel experiments. The phēnix web GUI is used to access these various experiments and uses Virtual Network Computing (VNC) to access individual virtual machines (VMs) within experiments. Once on a specific VM within an experiment, the environment appears to the user as if they were on the real device themselves.

Phēnix has other responsibilities such as VM and image management, event analysis, and even data collection capabilities such as packet captures. More details on the operation of phēnix can be found in Section 5.

5. SCEPTRE WORKFLOW

SCEPTRE aims to be highly configurable and user friendly. Creating emulations from scratch can be an involved process employing many tools, however the SCEPTRE workflow, shown in Figure 2, abstracts away much of the complexity into a user-friendly process.

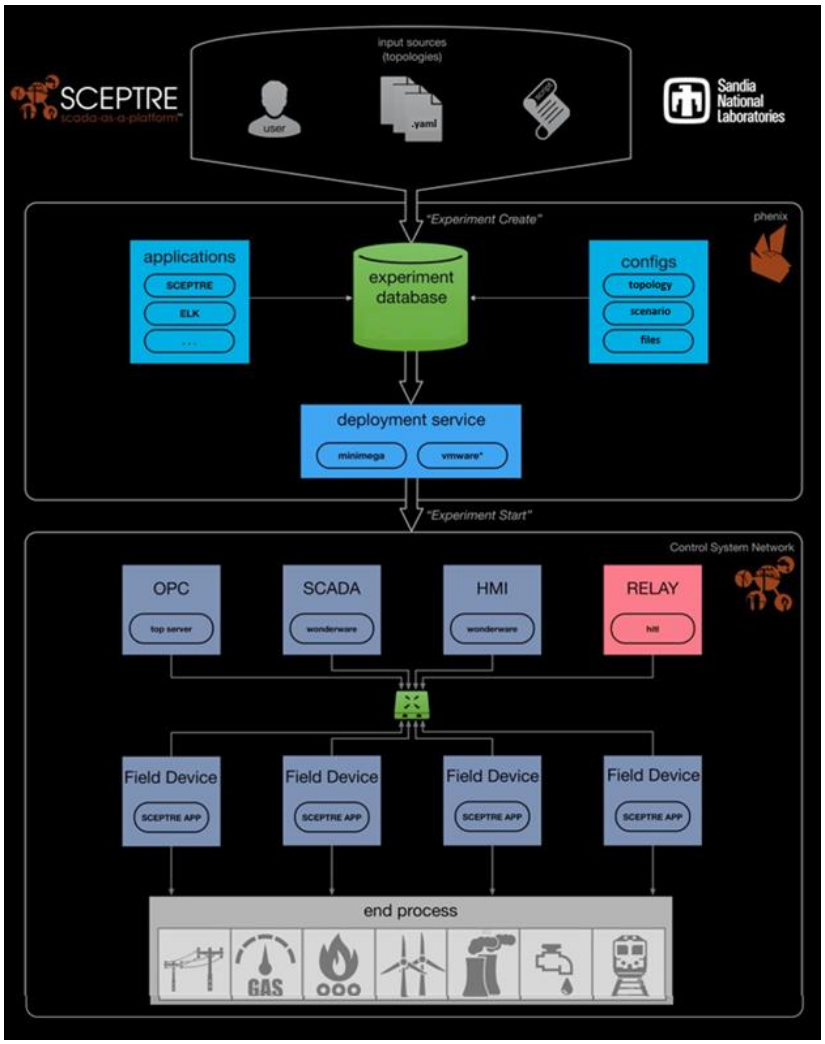

Figure 2: The SCEPTRE workflow.

A model in SCEPTRE is defined by two configuration files, a topology and a scenario. The topology file defines all the devices desired in the model. This includes attributes such as device name, hardware configuration, network interfaces, and more. The scenario file defines any extra details one wants to add on top of the basic topology. This file works by allowing the user to call various "apps" with additional configuration details. The most notable "app" is the SCEPTRE app itself. This app adds the cyber-physical system configuration to the topology. In the SCEPTRE app portion of a scenario file, the user can specify the type of cyber-physical system and supply metadata about the various devices such as the points in the physical-process simulation that an RTU might be monitoring or the set of RTUs that an OPC server is polling. This information is used later in the SCEPTRE workflow to take much of the configuration burden off the user.

Once the topology and scenario files have been defined, the phēnix orchestration tool does the rest. The configuration files must first be added to the phēnix database. Once this is done, the user is ready to run "experiments". The term "experiment" here refers to the instantiation of the model defined by the configuration files into a running emulation of networked VMs. Phēnix is used to create and start an experiment. Apps defined in the scenario file get called at these stages and these apps are typically responsible for appropriately configuring the VMs. This may include injecting files into the specific VMs and setting up processes and services to run on those VMs. The SCEPTRE app, for example, uses the configurations defined in the scenario file and automatically creates files that will:

1. start the desired software and application on startup of the VM,
2. configure the software appropriately.

This creates an experiment, that when it boots, all VMs are operating as desired without the need for intervention from the user. For example, once virtual RTUs in an experiment boot up, they are already receiving the desired data from the physical-process simulation and are speaking to upstream SCADA devices with the specified protocols.

Under the hood of phēnix, a few tools are used for the actual deployment of the VMs within an experiment. After phēnix ingests the configuration files and applies apps for various configurations, it writes a file for the tool Minimega to ingest. Minimega is a tool that launches and manages virtual machines across one or more nodes [42]. That is, phēnix does the high-level thinking and then hands off the configuration to minimega to do the actual deployment of the VMs. Minimega in turn leverages Quick Emulator (QEMU), in conjunction with Kernel-Based Virtual Machine (KVM), to configure and launch the VM components.

The web graphical user interface (GUI) of phēnix is used to orchestrate and view experiments and access VMs within an experiment. When users access VMs within an experiment, it drops them into a VNC console that, unless they didn't know they were in an emulation, would appear to be a real version of that machine.

Once a user is done with an experiment, it is as simple as stopping and deleting said experiment through the phēnix web GUI.

6. USE CASE

This section presents a notional use case called SOAP (SCEPTRE-On-A-Platter). SOAP was developed as a simple model to better acquaint users with SCEPTRE [39]. The authors acknowledge that the SOAP model is not the most complex model that can be implemented in SCEPTRE. The purposes of SOAP as a use case for this paper are to introduce readers to a basic model in order to understand the components of SCEPTRE and ensure readers have an example they can reproduce on their own with minimal computation requirements. For more complex/interesting use cases read the papers referenced in Section 3.

SOAP models the notional "Day Valley Power Authority". This power authority monitors a power transmission system, modeled using a portion of the standard IEEE 300 bus architecture [25]. Figure 3 shows the one-line power diagram for the system.

The SCADA portion of the power authority consists of:

- 4 virtual Remote Terminal Units (RTUs)
- 1 hardware-in-the-loop Programmable Logic Controller (HIL PLC)
- 1 Operator HMI
- 1 Attacker.

The network diagram for the SCADA system is shown in Figure 4.

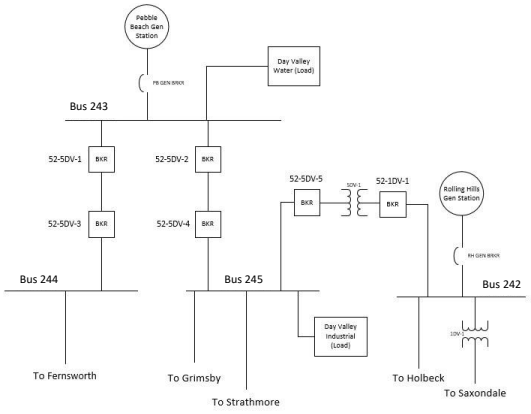

Figure 4: Day Valley Power Authority one-line diagram.

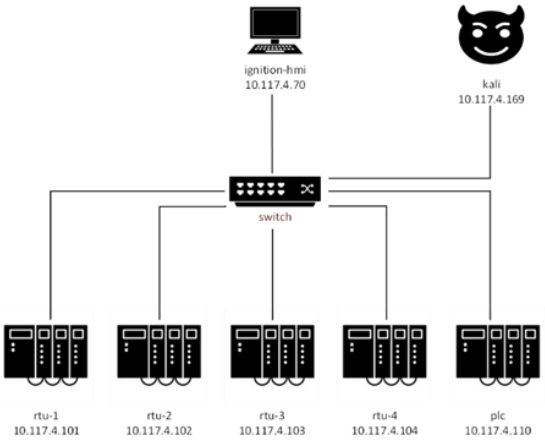

Figure 3: Day Valley network diagram.

Each of the four RTUs monitor and control various components of the power system. These field devices leverage the SCEPTRE virtual field devices described in section 4.2. Each RTU monitors busses and generators in the power system, recording information such as voltage, angle, MW, MVAR, and status. They in turn control the power system by opening/closing associated breakers. Each of the four RTUs are associated with busses 242, 243, 244, and 245 respectively.

SOAP has been intentionally configured in a non-secure state, aimed to provide a training environment for those who wish to learn vulnerability assessment or the impacts of applying cyber-security components to a functioning cyber-physical system. The benefit of such a system modeled

within the SCEPTRE environment is that these activities can be performed without any risk to physical systems or hardware

For reasons explained later in the scenario, the SOAP model was augmented with one additional PLC as HIL. (It is worth noting that the SOAP model can be run without this PLC, however the analysis question for this use case required the use of the PLC). The PLC for this specific use case is a Siemens S7-1200 with an analog addon module, providing 24-volt digital and analog inputs and outputs consistent with what might be found in an industrial environment. The PLC was configured to monitor and control both the generation output and the breaker on bus 243 in the power system. Input/Output signals between the power system simulation and the PLC were facilitated with a LabJack USB Digital Acquisition device Figure 5 shows the complete hardware configuration.

Figure 5: SOAP hardware configuration.

The operator HMI functions as the main operator view into the system. It aggregates the data from the virtual field devices and provides both a means for the operator to view the local system state and to request state changes to the loads, generation, and transmission line breakers. The operator HMI in the SOAP model uses Inductive Automation's Ignition HMI software. This software is commercially available and easy to configure. Figure 6 shows the operator view of the power authority using the Ignition software.

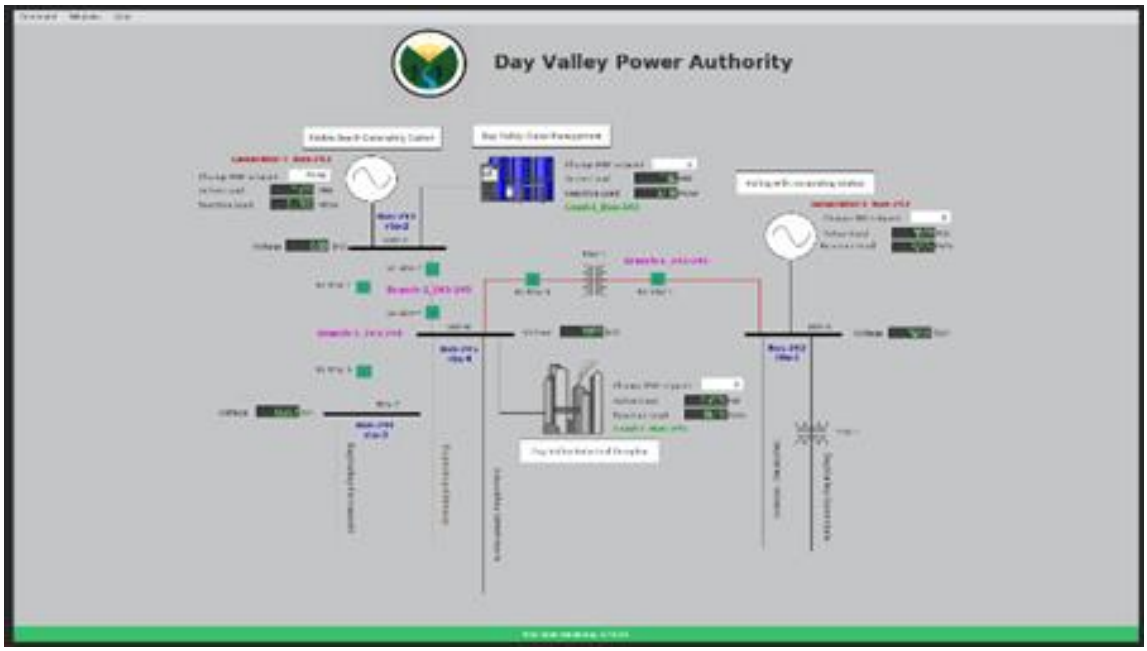

Figure 6: Ignition operator HMI.

The scenario of interest for this use case is a specific attack against the Siemens PLC. The system owners were concerned about a recent (notional) vulnerability in their PLC configuration and wanted to investigate if their system was susceptible to this vulnerability and what the ultimate effects of this vulnerability would be if exploited. To facilitate testing of this concern, the attacker component was added to the model by adding a machine to the SCADA network containing the digital forensic and penetration testing tool, Kali Linux.

Once all the model components were in place, an experiment was stood up using the phēnix orchestration tool, and the investigation began. In the attack, the attacker leveraged the fact that the default PLC configuration accepts unauthenticated S7 protocol connections. The PLC was unaware that these commands were malicious, so it acted as designed and made the appropriate changes in the power system, and resulting dynamics occurred. The attacker

1. opened breaker on Branch-1_243-244
2. changed the MW setpoint for Generator-1_Bus243 to 50
3. closed the breaker on Branch-1_243-244
4. repeated the same with a Generator-1_Bus-243 output of 100MW.

Since SCEPTRE couples the cyber components with the physical-process simulation, the effects of the attack on the power system can be observed. This test highlighted 4 key characteristics of this power system and vulnerability:

1. The PLC can be leveraged to control power system operation outside of the authorized HMI.
2. Output MW could be set by the attacker outside the parameters allowed in the HMI (100MW).
3. Cycling the breaker with a lower generator output (50MW) does not cause a system blackout.
4. Increasing the generator output (100MW) and cycling the breaker does cause a blackout. Figure 7 shows this blackout.

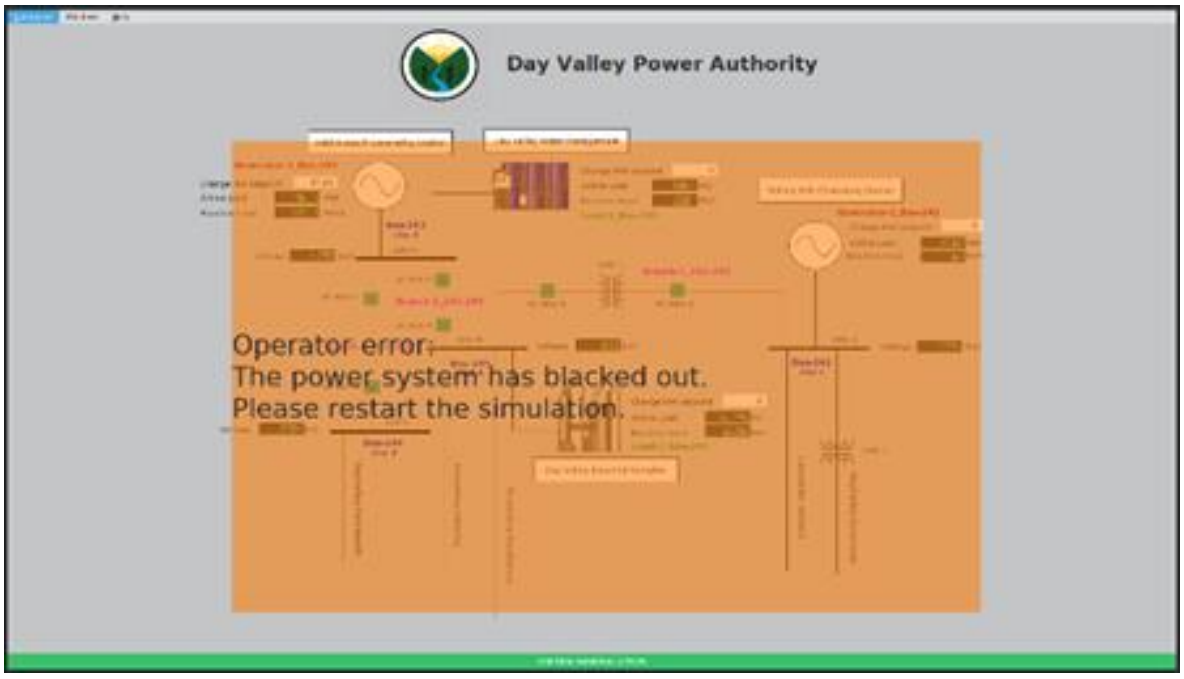

Figure 7: Ignition operator HMI attack effect.

The impacts of having this use case implemented in SCEPTRE are invaluable. First and foremost is that the cyber and the physical components are interconnected so that the impact of the vulnerability to the power system can be accurately assessed. Also, the experiment is easy to stand up and tear down so that multiple variations of the experiment can be performed with minimal configuration from the user. The power system can go from a complete blackout to normal operation with a few button clicks with no damage to physical hardware or the real system. The experiment can also easily be expanded to include different security measures or intrusion detection tools to explore how this scenario could potentially be detected and/or mitigated before the dire consequences to the power system.

7. CONCLUSIONS AND FUTURE WORK

SCEPTRE is a tool with over a decade of development originally built for critical infrastructure experimentation, most pertinent to government agencies and defense applications. However, as the proliferation of SCADA systems grows, so did the need for access to this capability. The modeling capability itself is mature and now fully accessible to the public [40].

This paper describes the general features of SCEPTRE and advantages of this capability over other types of testbeds for modeling cyber-physical systems when the interdependency between the cyber and physical components of the system are of interest. The use case showed a simple example of how SCEPTRE models can be configured and how SCEPTRE models can be used to answer a specific analysis question. More advanced models and use cases can be found in a variety of other SCEPTRE papers cited in Section 3.

The future of SCEPTRE has several strategic directions. Most of the directions pivot from spending time on manually building models to using models for activities such as cyber-physical threat quantification and reduction, scalable testing, and demonstrating the potential of proposed technologies before costly investments are made. A first step to this goal is automated model generation using data-driven sources to build models from potentially sparse data sets. Additionally, an automated experiment data gathering process will expand opportunities for more rigorous cyber-physical experimentation by facilitating more post-experiment scenario analyses. This will enable much needed statistical approaches to improve confidence in model results such as verification, validation, sensitivity analysis, and uncertainty quantification. Some work in these areas has begun, but much remains [34].

This page left blank.

REFERENCES

[1] Ahrenholz, J., Danilov, C., Henderson, T. R., & Kim, J. H. (2008, November). CORE: A real-time network emulator. In MILCOM 2008-2008 IEEE Military Communications Conference (pp. 1-7). IEEE.

[2] Aravinthan, V., Balachandran, T., Ben-Idris, M., Fei, W., Heidari-Kapourchali, M., Hettiarachchige-Don, A., ... & Tindemans, S. (2018, June). Reliability modeling considerations for emerging cyber-physical power systems. In 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS) (pp. 1-7). IEEE.

[3] Argonne National Laboratory. (2024). Secure Cyber Testbed (SECuRE). https://www.anl.gov/sss/secure-cyber-testbed

[4] Army's Program Executive Office Simulation, Training and Instrumentation. (n.d.). Persistent Cyber Training Environment (PCTE). https://www.peostri.army.mil/persistent-cyber-training-environment-pcte

[5] Ashok, A., Hahn, A., & Govindarasu, M. (2011, October). A cyber-physical security testbed for smart grid: System architecture and studies. In Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research (pp. 1-1).

[6] Ashok, A., Krishnaswamy, S., & Govindarasu, M. (2016, September). PowerCyber: A remotely accessible testbed for Cyber Physical security of the Smart Grid. In 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT) (pp. 1-5). IEEE.

[7] Beerman, J., Berent, D., Falter, Z., & Bhunia, S. (2023, May). A review of colonial pipeline ransomware attack. In 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW) (pp. 8-15). IEEE.

[8] Benzel, T., Braden, R., Kim, D., Joseph, A. D., Neuman, B. C., Ostrenga, R., ... & Sklower, K. (2007, August). Design, Deployment, and Use of the DETER Testbed. In DETER.

[9] Castillo, A., Arguello, B., Cruz, G., & Swiler, L. (2019, November). Cyber-physical emulation and optimization of worst-case cyber attacks on the power grid. In 2019 Resilience Week (RWS) (Vol. 1, pp. 14-18). IEEE.

[10] Chang, X. (1999, December). Network simulations with OPNET. In Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future-Volume 1 (pp. 307-314).

[11]Clements, A. A., Gustafson, E., Scharnowski, T., Grosen, P., Fritz, D., Kruegel, C., ... & Payer, M. (2020). HALucinator: Firmware re-hosting through abstraction layer emulation. In 29th USENIX Security Symposium (USENIX Security 20) (pp. 1201-1218).

[12] Conti, M., Donadel, D., & Turrin, F. (2021). A survey on industrial control system testbeds and datasets for security research. IEEE Communications Surveys & Tutorials, 23(4), 2248-2294.

[13] Dehlaghi-Ghadim, A., Balador, A., Moghadam, M. H., Hansson, H., & Conti, M. (2023). ICSSIM—a framework for building industrial control systems security testbeds. Computers in Industry, 148, 103906.

[14] Department of Defense. (2018). Department of Defense Cyber Strategy.

[15] Department of Homeland Security. (2023). The Third Quadrennial Homeland Security Review.

[16]Dragos. (2017). TRISIS Malware Analysis of Safety System Targeted Malware. Technical Report version 1.20171213. Dragos Inc.

[17] El Genk, M. S., Schriener, T., Altamimi, R., Hahn, A., Lamb, C., & Fasano, R. (2020, August). NICSIM: nuclear instrumentation and control simulation for evaluating response to cyber-attacks. In International Conference on Nuclear Engineering (Vol. 83778, p. V002T08A055). American Society of Mechanical Engineers.

[18] El-Genk, M., & Schriener, T. (2022). A cybersecurity platform for simulating transient responses of emulated programmable logic controllers in instrumentation and control systems for a pwr plant. Journal of Cyber Security Technology, 6(1-2), 65-90.

[19] Formby, D., Rad, M., & Beyah, R. (2018). Lowering the barriers to industrial control system security with {GRFICS}. In 2018 USENIX Workshop on Advances in Security Education (ASE 18).

[20] Galiardi, M., Gonzales, A., Thorpe, J., Vugrin, E., Fasano, R., & Lamb, C. (2020, August). Cyber resilience analysis of scada systems in nuclear power plants. In International Conference on Nuclear Engineering (Vol. 83778, p. V002T08A003). American Society of Mechanical Engineers.

[21] Gillen, R. E., Anderson, L. A., Craig, C., Johnson, J., Anderson, R., Craig, A., & Scott, S. L. (2020, August). Design and implementation of full-scale industrial control system test bed for assessing cyber-security defenses. In 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM) (pp. 341-346). IEEE.

[22] Holm, H., Karresand, M., Vidström, A., & Westring, E. (2015). A survey of industrial control system testbeds. In Secure IT Systems: 20th Nordic Conference, NordSec 2015, Stockholm, Sweden, October 19–21, 2015, Proceedings (pp. 11-26). Springer International Publishing.

[23] Hossain-McKenzie, S., Jacobs, N., Summers, A., Kolaczkowski, B., Goes, C., Fasano, R., ... & Overbye, T. (2022). Harmonized automatic relay mitigation of nefarious intentional events (harmonie)-special protection scheme (sps) (No. SAND2022-13427). Sandia National Lab. (SNL-NM), Albuquerque, NM (United States).

[24] The White House. (2018). National cyber strategy of the United States of America. WH, Washington, DC.

[25] Illinois Center for a Smarter Electric Grid. (n.d.). IEEE 300-Bus System. https://cyber-physical systemeg.iti.illinois.edu/ieee-300-bus-system/

[26] Johnson, J., Onunkwo, I., Cordeiro, P., Wright, B. J., Jacobs, N., & Lai, C. (2020). Assessing DER network cybersecurity defences in a power-communication co-simulation environment. IET Cyber-Physical Systems: Theory & Applications, 5(3), 274-282.

[27] Kovalenko, I., Saez, M., Barton, K., & Tilbury, D. (2017). SMART: A system-level manufacturing and automation research testbed. Smart and Sustainable Manufacturing Systems, 1(1).

[28] Lee, R. M., Assante, M. J., & Conway, T. (2017). Crashoverride: Analysis of the threat to electric grid operations. Dragos Inc.

[29] Mathur, A. P., & Tippenhauer, N. O. (2016, April). SWaT: A water treatment testbed for research and training on cyber-physical system security. In 2016 international workshop on cyber-physical systems for smart water networks (CySWater) (pp. 31-36). IEEE.

[30] NIST, C. (2013). Strategic R&D Opportunities for 21st Century Cyber-Physical Systems. In Workshop Report: Foundations for Innovation in Cyber-Physical Systems.

[31] Pacific Northwest National Laboratory. (n.d a). CyberNET Testbed. https://www.pnnl.gov/projects/center-collaborative-cyber-physical-research/cybernet-testbed

[32] Pacific Northwest National Laboratory. (n.d b). powerNET Testbed. https://www.pnnl.gov/projects/center-collaborative-cyber-physical-research/powernet-testbed

[33] Parvania, M. (2024). Cyber-Physical System Resilience (CPSR) Testbed. https://usmart.ece.utah.edu/cyber-physical-resilience-testbed/

[34] Pinar, A., Tarman, T., Swiler, L. P., Gearhart, J., Hart, D., Vugrin, E., ... & Punla-Green, S. I. (2021). Science and Engineering of Cybersecurity by Uncertainty quantification and Rigorous Experimentation (SECURE) (No. SAND-2021-11719). Sandia National Lab. (SNL-NM), Albuquerque, NM (United States); Sandia National Lab.(SNL-CA), Livermore, CA (United States).

[35] Qassim, Q., Jamil, N., Abidin, I. Z., Rusli, M. E., Yussof, S., Ismail, R., ... & Daud, M. (2017). A survey of scada testbed implementation approaches. Indian Journal of Science and Technology.

[36] Queiroz, C., Mahmood, A., & Tari, Z. (2011). SCADASim—A framework for building SCADA simulations. IEEE Transactions on Smart Grid, 2(4), 589-597.

[37] Riley, G. F., & Henderson, T. R. (2010). The ns-3 network simulator. In Modeling and tools for network simulation (pp. 15-34). Berlin, Heidelberg: Springer Berlin Heidelberg.

[38] Sahu, A., Wlazlo, P., Mao, Z., Huang, H., Goulart, A., Davis, K., & Zonouz, S. (2021). Design and evaluation of a cyber-physical testbed for improving attack resilience of power systems. IET Cyber-Physical Systems: Theory & Applications, 6(4), 208-227.

[39] Sandia National Laboratories. (2020). SOAP: SCEPTRE on a Platter. SAND2020-8439O.

[40] Sandia National Laboratories. (2023), SCEPTRE | An emulation capability for Industrial Control Systems. https://sandialabs.github.io/sceptre-docs/

[41] Sandia National Laboratories. (n.d a). Emulytics™. https://www.sandia.gov/emulytcyber-physical system/

[42] Sandia National Laboratories. (n.d. b). minimega. https://www.sandia.gov/minimega/

[43] Siaterlis, C., Garcia, A. P., & Genge, B. (2012). On the use of Emulab testbeds for scientifically rigorous experiments. IEEE Communications Surveys & Tutorials, 15(2), 929-942.

[44] Slowik, J. (2019). Stuxnet to CRASHOVERRIDE to TRISIS: Evaluating the history and future of integrity-based attacks on industrial environments. Dragos Inc.

[45] Summers, A., Goes, C., Calzada, D., Jacobs, N., Hossain-McKenzie, S., & Mao, Z. (2022, March). Towards cyber-physical special protection schemes: Design and development of a co-simulation testbed leveraging SCEPTRE™. In 2022 IEEE Power and Energy Conference at Illinois (PECI) (pp. 1-7). IEEE.

[46] Thorpe, J., Fasano, R., Galiardi Sahakian, M., Gonzales, A., Hahn, A., Morris, J., ... & Vugrin, E. D. (2022a). A cyber-physical experimentation platform for resilience analysis. In Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (pp. 3-12).

[47] Thorpe, J., Swiler, L. P., Hanson, S., Cruz, G., Tarman, T., Rollins, T., & Debusschere, B. J. (2022b). Verification of Cyber Emulation Experiments Through Virtual Machine and Host

Metrics. In Proceedings of the 15th Workshop on Cyber Security Experimentation and Test (pp. 71-80).