Validating an emulation-based cybersecurity model with a physical testbed

Hao Huang*, Patrick Wlazlo*, Abhijeet Sahu*, Adele Walker*, Ana Goulart*,and Kate Davis*
Texas A&M University, College Station, Texas, USA
Laura Swiler†, Thomas Tarman†, and Eric Vugrin†
Sandia National Laboratories, Albuquerque, New Mexico, USA
Email: {hao_huang, pjrwlazlo, abhijeet_ntpc, adelewalker, goulart, katedavis}@tamu.edu,
{lpswile, tdtarma, edvugri}@sandia.gov

Abstract—For researchers studying cyber-physical system security, working with realistic datasets is essential. To produce the datasets, the existing methodology is to emulate the cyber network. A challenge is that the industrial control systems (ICS) network consists of not just computers and communication equipment, but also field devices that collect data and execute controls. These devices play a significant role in the operation and the security of the system. However, in comparison to the cyber network, the research reproducibility and realism of the cyber-physical system emulation and its data has received far less attention. This paper thus develops an approach to answer, "How well can emulated devices replicate the behavior of physical intelligent electronics devices (IEDs) in a realistic cyber attack and defense environment?" To study this, we perform a comparison study based on an emulation experiment using the `minimega` testbed environment that is entirely virtual and a hardware-in-the-loop experiment using the Resilient Energy Systems Lab (*RESLab*) cyber-physical testbed featuring real industrial controllers and communications devices. Results show that under different reconnaissance attack scenarios, *RESLab* generates realistic datasets that validate the emulation-based cybersecurity model in `minimega`. The approach is generalizable toward validating the realism of other types of ICS devices in security studies.

Index Terms—reconnaissance attacks, emulation-based testbeds, cyber experimentation reproducibility, experiment comparison metrics

1 Introduction

[1] The first and foremost step of a successful high-impact cyber attack is the **planning phase**, which can be realized by *reconnaissance*. This type of attack aims to reveal weaknesses and identify information to support adversaries in their goals to target, deliver, and exploit elements of a system. Reconnaissance involves researching industrial control system (ICS) technical vulnerabilities and features as well as gaining an understanding of how each process and operating system may be susceptible to exploitation [1], [2], [3].

Fig. 1 shows Stage 1, cyber intrusion preparation and execution, of the ICS Cyber Kill Chain, from [3]. After *reconnaissance*, the adversary can utilize the information and weaknesses to implement its next steps. Examples can be found with the cyber attacks in the Ukranian power grid [4], the unidentified threats in the European Network of Transmission System Operators for Electricity (ENTSO-E) [5], and the recent Colonial Pipeline attack [6]. In supervisory control and data acquisition (SCADA) systems, intelligent electronic devices (IEDs) play an important role to connect cyber and physical networks. IEDs have both communication and control capability. However, they have less computing capability, making them more vulnerable against cyber attacks. For power grids particularly, IEDs in SCADA systems connect cyber and physical networks, and their functionalities are crucial for system stability and security, such as protecting the system against physical faults, automatically regulating voltage, and automatically adjusting generator output. If those IEDs are compromised, there could be negative outcomes in the power grid. As information technology (IT) networks and operational technology (OT) networks are increasingly interconnected, the reconnaissance attack is an important step to obtain critical *cyber* information that can compromise the security, functionality, and reliability of both IT and OT networks. Thus, for cyber-physical systems, especially the smart grid, the resiliency of the cyber network with IEDs is paramount regarding the monitoring and control for operational technology.

There are four classes of reconnaissance techniques: *Social Engineering*, *Side-Channels*, *Network Information Gathering*, and *Internet Intelligence* [7]. For smart grids, most data is not publicly available. Thus, when adversaries try to obtain the information to plan their activities, *Network Information Gathering*, also known as *Network Scanning*, is a popular choice to map a remote network or identify operating systems and applications. More sophisticated techniques have been proposed to improve the efficiency and effectiveness

1. This paper has been accepted by the IEEE Transactions on Dependable and Secure Computing.

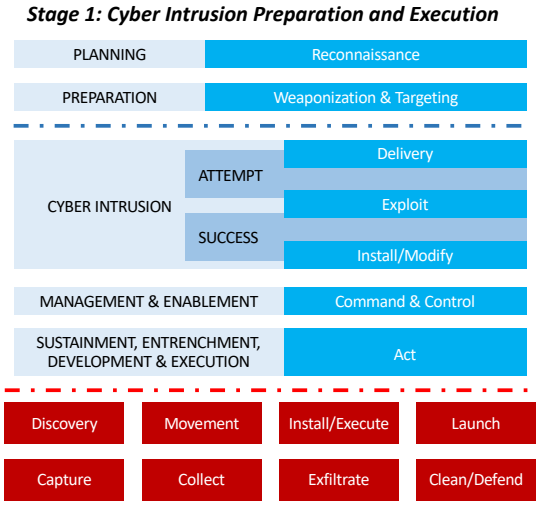


Fig. 1: Stage 1 of Industrial Control System (ICS) Cyber Kill Chain, adapted from [3].

of network scanning. For example, in [8], strategies were proposed to deal with a large number of hosts and to conserve network traffic as well as time of specific tasks during scanning. In [9], Malkawi *et al.* introduced an application program interface (API) based port and vulnerability scanner to collect information for penetration testing. In [10], Li *et al* proposed an adaptive and parallel strategy so that attackers can reach the targets faster with better robustness.

To prevent cyber attacks in an early stage, it is important to effectively detect the reconnaissance attack, especially port scanning. In [11], a rule-based network intrusion detection system (IDS) for port scanning attacks is proposed and implemented with *Snort*. The authors in [12] present an intrusion detection and prevention system using a software-defined network (SDN) to detect and prevent port scanning attacks in real-time. With the data generated from cyber-physical testbeds, machine learning techniques have also been utilized to detect cyber attacks [13], [14].

To defend the reconnaissance attacks, several techniques have been proposed. The study in [15] introduced an SDN-based defense against reconnaissance attacks and showed its effectiveness within an emulated network. In [16], Jafarian *et al.* present the random host address mutation approach to proactively defend zero-day, stealthy reconnaissance, and scanning attacks. In [17], Wang *et al.* present a defense technique against adversarial reconnaissance and scanning by dynamically mutating domain names and internet protocol (IP) addresses. In [18], the authors present a novel architecture that integrates network intrusion detection, mitigation, and prevention systems for wide-area protection against cyber attacks in smart grids.

To investigate the cyber security of ICS, cyber-physical testbeds are necessary. The previously discussed detection and prevention methods [15], [16], [17], [18] have been evaluated in small-scale or emulated communication networks. The question remains open on how repeatable and replicable such results can be especially with respect to the real-world systems that they emulate. In fact, many security research works can offer significant benefits to critical infrastructure defense, if they can be successfully replicated and validated in realistic systems at scale. For example, in [19], a virtual assured network testbed for cyber security, CyberVAN, provided a realistic cyber security tool to evaluate different types of cyber security techniques, such as the detection of botnet command and control mechanisms and malware propagation. An emulation-based testbed [20], `minimega`, has been created for large-scale communication network cyber security studies. For ICS, several hardware-in-the-loop testbeds [21], [22], [23] have also been built with industrial IEDs and communication devices. Compared to the hardware-based testbed, the emulation-based testbed is better at scaling up the network for larger and more complicated systems. However, the industrial IEDs were equated to computer nodes in the emulation environment, which may not capture the IEDs' response under cyber attacks. The science of cybersecurity is an important aspect of its understanding, development, and practice; thus, cybersecurity experiments must be objective, falsifiable, reproducible, predictable, and verifiable [24]. With few works delved into the verification of emulated and physical devices under cyberattacks, the research question identified in this paper is *"How well can emulated devices replicate the behavior of physical IEDs in a realistic cyber attack and defense environment?"*

When developing any type of model (a mathematical, simulation, or emulation model), one must be concerned about the accuracy of the model in representing the behavior or phenomena of interest. Validation addresses the question of adequacy of a model [25], [26]: is the model accurate enough and appropriate to be used for a prediction? Typically, validation involves the comparison of the model with observational or experimental data using statistical metrics called validation metrics [25], [27]. Validation for computational physics models has been a discipline for many years; however, validation for cyber emulation models is starting to be considered [28]. Very few studies have validated the cyber emulation models with their corresponding physical systems for ICS. Hence, this work addresses that gap.

Crussell *et al.* [29] have investigated the question of emulation adequacy, identifying behavior at three levels of abstraction: application, operating system, and network. A unique aspect of their work was the use of Markov models to compare patterns of system call orderings to determine whether the application behavior on the virtual testbed (built in `minimega`) was similar to the physical testbed (the physical computer nodes). They also performed some large scale emulation studies [30], with key lessons learned about the infrastructure required to perform controlled validation experiments, the amount of data generated, and the need for careful statistical analysis when analyzing the data. In terms of cyber-physical validation, Zheng and Julien [31] outlined research challenges associated with validation of cyber-physical systems (CPS), including the lack of knowledge in the CPS development community about verification and validation tools, the insufficiency of formal methods to scale for this domain, the lack of online monitoring capabilities, and the need to address uncertainties.

This paper demonstrates the validation of an emulation experiment with a physical system, addressing the issues of carefully designed experiments, use of metrics and statistical analysis, and incorporation of uncertainties. To the best of our knowledge, there is no other cybersecurity study that

tried to validate the emulation model with the physical IEDs for the communication network in SCADA system. This paper aims to validate results from the `minimega` emulation-based testbed [20] on a *Network Scanning* reconnaissance experiment, more specifically *adversary port discovery*, versus its detection by an IDS, reported in [32]. The physical validation experiments were conducted by the Texas A&M University (TAMU) authors in the Resilient Energy Systems Lab (*RESLab*) cyber-physical testbed environment [23] with collaboration from Sandia National Labs (SNL) to ensure proper experimental setup and network architecture.

Prior to this study, two other studies were conducted: a validation study was conducted in [32] in a virtual-machine (VM)-based testbed cluster, and another study [33] in a container-based testbed using common open research emulator (CORE) [34] . Unlike the previous experiment [33], the experiment environment in this paper combines the industrial IEDs and communication devices combined with CORE emulator to build a power grid supervisory control and data acquisition (SCADA) system. Thus, this paper extends those experiments with a hardware-in-the-loop testbed to compare the results between the emulated environment and the physical environment in order to validate cybersecurity emulation experimental results.

These are the main contributions of this paper:

- We present a comparison study of cybersecurity analysis from an emulation experiment and a hardware-in-the-loop experiment to validate the cybersecurity emulation results with physical systems under port scanning attacks. This is an important study for cybersecurity to protect an ICS in an early stage, which has gained less attention from existing literature.
- We introduce the design of the hardware-in-the-loop testbed to optimally utilize the limited number of physical IEDs to represent a scalable and realistic ICS communication network. Specifically, we present the process of automatically collecting data within the physical testbed, and discuss the challenges and lessons from the physical validation experiment, which can be beneficial for other research teams to replicate the scenarios and study the realism and defense of other attacks in different physical testbeds.
- Under two port scanning scenarios and two settings for randomness, we generate realistic datasets from two testbeds to analyze the behavior of cyber attacks in both emulation and physical environments. We have utilized those data to validate a mathematical model that represents the port scanning attacks and detection. These datasets can also be used for future studies on developing data-driven based intrusion detection in ICS networks.
- We perform analyses of the datasets with the Kolmogorov-Smirnov (KS) test and the Bootstrap Method. We interpret the results and what it means for a replication to be considered acceptable. The results show that the `minimega` emulation-based model is close enough to the *RESLab* physical model. The presented experiment, case studies and statistical analysis tools can be generalized with different attack scenarios, such as Distributed Denial of Service, Man-in-the-Middle, and SQL injection, with corresponding parameters.

The rest of the paper is organized as follows. Section 2 reviews the configuration of the emulation testbed, `minimega`, and the reconnaissance scenarios. Section 3 presents the configuration of the physical testbed, *RESLab*, and the implementation of reconnaissance attacks in physical devices. In Section 4, we first we analyze the results from both testbeds with statistical analysis, the Kolmogorov-Smirnov (KS) test and the Bootstrap Method. Then, we validate the mathematical model of port scanning and detection in [32] with the data from both testbeds. Section 5 concludes the paper.

2 Background

This section reviews the port scanning scenarios and emulated topology in `minimega` constructed by SNL team.

2.1 Test Scenarios

The experimental setup, described in detail in [32], includes four types of VMs: a scanning node, a detection node, a router, and Remote Terminal Units (RTUs). The RTUs consist of different IEDs to collect data and execute control actions. The scanning node runs *Nmap*, a popular open source application for discovering hosts on a network and the services those hosts are offering [35]. This port scanning attack is a network-based attack to exploit IEDs' vulnerability in a communication network. It is assumed that the *Nmap* software runs on a compromised computer in an energy utility's control center. To model a reconnaissance scenario, *Nmap* scans Transmission Control Protocol (TCP) applications running on port 20000, which represents the Distributed Network Protocol 3 (DNP3) ICS protocol. *Nmap* scans all IP addresses of hosts in the utility's substation networks to determine which ones had their DNP3 ports *open*, *closed*, or *filtered*.

Nmap includes configuration parameters designed to avoid detection by IDSs. For example, by default, *Nmap* scans hosts in sequential order, but the user can specify random ordering (*randomize-hosts*) to make the scan less obvious to IDSs. The TAMU team found the randomization of port order in the scanning to be a critical issue in reproducibility [32], which is the ability to repeat experiments across different hardware platforms and/or software implementations. In this paper, reproducibility refers to reproducing the experiments in an emulated *vs.* physical environment. Other important IDS avoidance parameters control the rate at which the port scanning takes place, governed by host group size (*max-hostgroup* and *min-hostgroup*). Another parameter is the specified amount of time (*scan-delay*) before *Nmap* will wait to re-scan hosts within the current group or move to the next host group. TABLE 1 shows two port scanning strategies, considering different *host-group* and *scan-delay*. The slow stealthy strategy of *Nmap* has smaller *host-group* and longer *scan-delay*, which can decrease an IDS's ability to detect the scan but sacrifice the efficiency of port scanning attack. The fast and loud strategy of *Nmap* has bigger *host-group* and shorter *scan-delay*, which improves the efficiency of port scanning attack but increase

their chances being detected by IDS. In general, decreasing *host-group* size and increasing *scan-delay* decrease an IDS's ability to detect the scan. However, the delay required to scan all hosts increases.

Strategy	Parameters
Slow, Stealthy	host-group = 4, scan-delay = 10 s
Fast, Loud	host-group = 6, scan-delay = 5 s

TABLE 1: Port scanning attack scenarios using *Nmap* command.

Snort, an open source intrusion detection system (IDS) [36], is used to monitor the traffic going into the substation communication network. For the experiment, *Snort* uses the *sfPortscan* module that is designed to detect port scanning using TCP connection requests [2]. The module comes "pre-installed" in *Snort* and was originally developed by a team at SourceFire. The parameters for *Snort* portscan includes *sense_level*, *proto*, and *scan_type*. In this paper, *Snort* is configured with the same settings *sense_level* {*L*}, *proto* {*all*}, and *scan_type* {*portsweep*} for all experiments. *Snort* keeps track of the number of unsuccessful session attempts that are observed within a 60-second time window (by default). If the number of unsuccessful session attempts exceeds a specified threshold, *Snort* generates an alert containing the source and destination IP addresses of the probe that exceeded the *sfPortscan* threshold.

The experimental topology, *Nmap* and *Snort* software, and emulation environments are being used in this validation study to run four different scenarios, as shown in Table 2. In addition to the *Nmap* and *Snort* configurations, the randomized scenarios also include the option to randomly drop packets at a rate of 10%, to evaluate the effect of packet loss on scanning results and detection probabilities. This study includes a deterministic scenario with no random port ordering in *Nmap* and no packet drops in *Snort*. The variables in all experiments are the port ordering in the address search (random or not), packet loss (no loss or 10% loss), and *Nmap* port scanning strategies (slow or fast), and thus there are eight scenarios of port scanning attack.

2.2 Emulation Experiments

The `minimega` emulation testbed environment [20] is an open-source tool for constructing and orchestrating cyber experiments in a safe, offline fashion on a computing cluster. `minimega` uses the Kernel Virtual Machine (KVM) [37] hypervisor to instantiate network hosts and routers, Open vSwitch [38] to instantiate network switches, and Virtual Local Area Network (VLAN) technology to provide separation between network segments. `minimega` also provides mechanisms for launching applications within virtual machines, transferring files (e.g., for experiment results), and Applications Programming Interfaces (APIs) to automate experimental actions.

The experimental procedure, configurations, and results from this study are described in more detail in [33] and [32]. The topology for the emulation-based experiment is shown in Fig. 2(a). The purpose of the study described here is to validate the original experiments in the fully-virtualized `minimega` environment. Validation is performed through comparison of the `minimega` experimental results with results from a hardware-based *RESLab* testbed.

2. https://www.snort.org/faq/readme-sfportscan

Sources of Randomness	Deterministic Formulation	Stochastic Formulation
***Nmap* Address search order**	fixed sequence same for each trial	random sequence varies between trials
Packet Loss	no loss (none)	random packet drops; 10% per packet sent

TABLE 2: Four experimental designs controlling for the sources of randomness.

3 Physical Experiments

The *RESLab* testbed [23] consists of physical substation equipment as well as emulated network devices. To scale up the communication network in *RESLab* testbed with more IEDs, four protective relays (two SEL-351 and SEL 421, respectively) have been added with a Layer 3 (L3) network switch [39]. The physical substation equipment now includes real-time automation controllers (RTACs), protective relays, and a L3 network switch.

The RTAC and protective relays use the default settings for their communication parameters. The frequency of data collection is based on RTAC's *Integrity Polling Period* and *Class 1,2,3 Polling Period* for normal data and event data from the relay. The *Integrity Polling Period* is configured as *60000 ms* and the *Class 1,2,3 Polling Period* is configured as *5000 ms*, respectively [39].

Even though the control logic of the RTAC and protective relays is not used in this paper, the IEDs' network stacks are affected by the (presumably) limited processing capability of the IEDs. Conducting this test on real industrial IEDs, such as the RTAC, allows us to assess whether the processing limitations on real-world IEDs affect the fidelity of the results, which are based on timing and could be skewed if processing is too limited. Thus, it deserves more attention to investigate IED's vulnerability in ICS through the network-based attack.

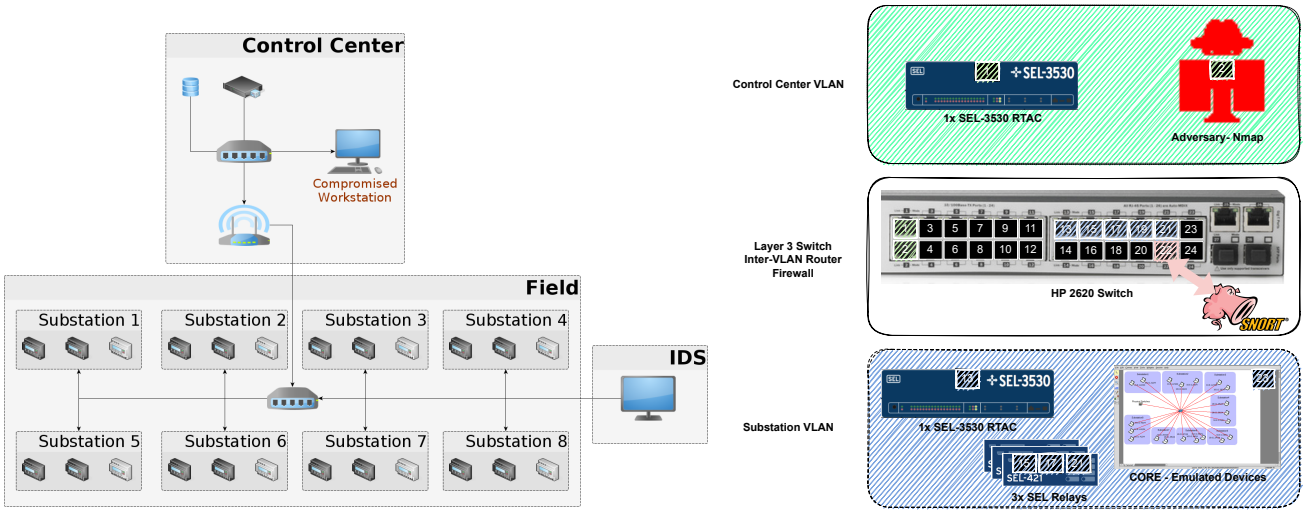
The emulated RTU devices are implemented using CORE emulator, which is integrated with a real-time power system simulation environment using PowerWorld Dynamic Studio (PWDS) for comprehensive cyber-physical power system studies. CORE and PWDS run on separate virtual machines (VMs) hosted in vSphere.

Next we describe how the physical and emulated devices are setup and configured in the *RESLab* testbed to replicate the experiment topology as in `minimega`.

3.1 Network Topology

To reproduce the experiment in [32] with hardware devices, the *RESLab* testbed is configured as in Fig. 2(b), which shows the physical and emulated devices and their network topology within the testbed as used for the experiments. Each physical device (RTACs and relays) represents one RTU in this paper for simplification.

The physical network has two virtual local area networks (VLANs) or subnetworks:



(a) The `minimega` virtual experiment topology.

(b) The *RESLab* hardware-in-the-loop experiment topology.

Fig. 2: Side-by-side comparison of emulation-based testbed `minimega` and hardware-based testbed *RESLab*.

- A substation VLAN with IP address 10.0.1.0/24 has one Schweitzer Engineering Laboratories 3530 (SEL-3530) RTAC, two SEL 421 relays and one SEL 351 relay. These are connected to the emulated network in CORE (shown at the right side at the bottom of Fig. 2(b)).
- A control center VLAN with IP address 10.0.2.0/24 has another physical SEL-3530 RTAC, configured with IP address 10.0.2.20. The control center VLAN contains the adversary's *Nmap* scanning node.

Although the specific IP addresses used in the physical experiment differ from those used in the `minimega` experiments, the network layouts are identical (i.e., a control center network and a substation network).

To construct the reconnaissance attack scenario, there are two standalone computers to run the *Nmap* port scanning and *Snort* intrusion detection tools. The computer that performs the *Nmap* scanning attacks is located in the control center VLAN, with IP address 10.0.2.11. The computer that performs the *Snort* intrusion detection is in the Substation VLAN, with IP address 10.0.1.1.

The L3 switch that is shown in the middle of Fig. 2(b) routes packets between the substation and control center VLANs. These physical ports of this switch are configured as follows: ports 1-12 is the Control Center VLAN, and Port 13-24 is Substation VLAN.

To capture all traffic flowing in and out of the substation VLAN, Port 22 on the L3 switch is configured as a *monitor* port. The laptop running *Snort* is connected to this port in order to detect any *Nmap* reconnaissance packets. Also in the substation VLAN, Ports 13, 15, 17, and 19 are connected to the physical substation devices, such as relays and RTAC. Port 21 connects with the VM running CORE. In the control center VLAN, Port 1 connects the control center RTAC, and Port 2 connects the laptop running *Nmap*.

3.2 Port Configuration

Now we address the port configuration, where a *port* here means a virtual identifier of the applications or services that are running on the field devices and emulated nodes.

The same TCP port parameters for the RTUs from [32] are configured in our experiments, where we set ports as *open*, *closed*, and *filtered*. An *open* port means an RTU that supports the application layer protocol, which in this case is the Distributed Network Protocol 3 (DNP3) at TCP port 20000. A *closed* port is a node that does not support DNP3 over TCP. And a *filtered* port means that the router, in our case the L3 switch, has an access control list (ACL) that blocks that application protocol and IP address.

Following [32], we need 24 RTUs in the substation network. For the experiments in *RESLab* testbed, a hybrid of real and virtual RTUs was used. Four of these RTUs are IEDs, including one RTAC and three protective relays. The other RTUs are emulated nodes created in the CORE emulator, which are also in the substation network. Not all 24 RTUs were actual hardware devices in the physical experiments because we had a limited number of physical devices available. In general, this reason contributes to the scaling of the physical side of experiments being more prohibitive than scaling in an emulation environment. Understanding the tradeoffs in using real *vs.* virtual devices in studies also better guides how to scale experiments while retaining a desired degree of accuracy. For these reasons, we chose four physical hosts as the ones to be tested as *open* ports, as explained next. This choice allows us to represent a realistic ICS communication network with the combination of IEDs and computer nodes. The rationale is that the *open* ports would exhibit the most significant behavior that we want to study while under attack, so we prioritize making those the real devices.

In the substation network, the RTAC and the three protective relays are RTUs with IP addresses on subnetwork 10.0.1.0/24. They are configured as *open* DNP3 ports. 8 RTUs are modeled as *closed* and 12 RTU's as *filtered* ports.

Port Type	VLAN	IP address	Device
Open	*10.0.1.0*	*.20, .26, .32, .38*	RTAC and Relays
Closed	*10.0.1.0*	*.22, .25, .28, .31, .34, .37, .40, .43*	Emulated in CORE
Filtered	*10.0.1.0*	*.21, .23, .24, .27, .29, .30, .33, .35, .36, .39, .41, .42*	Emulated in CORE

TABLE 3: *RESLab* substation VLAN and IP address configuration.

All *closed* and *filtered* ports are CORE emulated nodes. The IP addresses for all *open*, *closed* and *filtered* ports in the substation network are listed in Table 3.

Using the IP addresses for the *open* and *closed* ports, we configured an access control list (ACL) in the L3 switch, which is shown in Fig. 3. In this ACL, the rules can be explained as follows:

- The first rules, up to Rule 80, allow the control center network to send packets to the IP addresses of the emulated nodes, which do not run DNP3. Therefore, *Nmap* will return these reconnaissance packets as *closed* ports.
- Rules 100 to 130 allow packets to be sent from control center to the RTAC and three relays in the substation network. Since these devices communicate via DNP3 protocol, *Nmap* will return the probe packets as *open* ports.
- Rule 140 allows the RTAC in the substation to communicate with the RTAC in control center on TCP port 20003.
- Rule 150 allows the *Nmap* machine to check if the control center's RTAC is up and running before the experiment begins.
- Rule 160 denies all other traffic. When a packet matches this last rule, the reconnaissance packet is returned as *filtered*.

This ACL is applied as an inbound filter on the physical Port 2 of the L3 switch, where the *Nmap* laptop is connected. Thus, the switch will inspect all reconnaissance packets that are sent into the substation network.

Also, different packet loss probabilities were configured in the *Nmap* machine's *iptable* to emulate different network conditions. As in Table 2, a 10% packet drop was configured for the scenario that considered packet loss.

3.3 Containers vs VMs

In CORE, each node is simply a Linux container known as a FreeBSD jail, whereas in `minimega` each node is a separate KVM. Hence, differences in performance can be attributed to the diversity in resources available in these two types of environment.

There are a few differences between Linux containers and VMs. One key difference is that a Linux container is running on a host operating system (OS) and is able to access tools that are installed on the host OS. For instance, if *Nmap* is installed on the host OS, all of the containers running on that OS now can access and run an *Nmap* through the Linux containers' virtualized network cards.

```
ip access-list extended "100"
        10 permit tcp 10.0.2.0 0.0.0.255 10.0.1.25 0.0.0.0 eq 20000
        20 permit tcp 10.0.2.0 0.0.0.255 10.0.1.28 0.0.0.0 eq 20000
        30 permit tcp 10.0.2.0 0.0.0.255 10.0.1.31 0.0.0.0 eq 20000
        40 permit tcp 10.0.2.0 0.0.0.255 10.0.1.34 0.0.0.0 eq 20000
        50 permit tcp 10.0.2.0 0.0.0.255 10.0.1.37 0.0.0.0 eq 20000
        60 permit tcp 10.0.2.0 0.0.0.255 10.0.1.40 0.0.0.0 eq 20000
        70 permit tcp 10.0.2.0 0.0.0.255 10.0.1.43 0.0.0.0 eq 20000
        80 permit tcp 10.0.2.0 0.0.0.255 10.0.1.22 0.0.0.0 eq 20000
        100 permit tcp 10.0.2.0 0.0.0.255 10.0.1.20 0.0.0.0 eq 20000
        110 permit tcp 10.0.2.0 0.0.0.255 10.0.1.26 0.0.0.0 eq 20000
        120 permit tcp 10.0.2.0 0.0.0.255 10.0.1.32 0.0.0.0 eq 20000
        130 permit tcp 10.0.2.0 0.0.0.255 10.0.1.38 0.0.0.0 eq 20000
        140 permit tcp 10.0.2.20 0.0.0.0 10.0.1.20 0.0.0.0 eq 20003
        150 permit ip 10.0.2.0 0.0.0.255 10.0.2.0 0.0.0.255
        160 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
exit
```

Fig. 3: Access control list on L3 Switch.

However, the drawback is that all the containers now share the same software and hardware resources, and have to request resources from the systems kernel's singular memory and processing unit. The alternative to using a Linux container is to use a VM, where all software and hardware can be dedicated to that VM. A VM might not be as efficient as a Linux container with its hardware resources, but at least it can be relied upon to have those resources available to it at any time. Our previous study has shown the differences between containers-based and VMs-based emulations [33].

3.4 Reconnaissance Scenarios in Physical Environment

The reconnaissance scenario in [32] assumes the adversary sends TCP reconnaissance packets into the substation network to report the "lay of the land," so to speak. Thus, the adversary node is configured with a standalone computer under the same network as the control center RTAC.

To replicate this reconnaissance probing using industrial hardware (network switches, RTACs, and relays), it is assumed that a man-in-the-middle (MiTM) eavesdropping attack is used by the adversary. The MiTM attack can be achieved through Address Resolution Protocol (ARP) cache poisoning, which allows the adversary to receive packets that were sent to another network device [40]. This is important to consider for the attack vector because initially the *Nmap* machine was not able to scan the RTAC or relays in the substation network. Only administrator-approved nodes can establish a DNP3 connection with the RTAC and relays in the substation network. For instance, the control center's RTAC is an administrator approved node.

In an actual attack environment, the attacker node would have to impersonate the control center's RTAC because it is the only device that should be able to connect to the RTAC and relays. For simplicity and to ensure anyone who uses our results can differentiate between the adversaries' packet data and the control center RTAC's traffic, we added the *Nmap* machine to the list of hosts that can access the relays and RTAC in the substation network.

Under different *Nmap* scanning strategies, *Snort* generates alerts and logs for cyber threats in the substation's network. For post-processing purposes and to ensure the

network setup does not change over the course of the experiment, a packet sniffer tool called *Wireshark* is used to log packet data in a packet capture (PCAP) format from the L3 switch's monitoring port.

3.5 Automated Data Collection Process

The TAMU team developed Python scripts to automate the dataset generation process. The overall process is shown in Fig. 4 for *Snort* and *Nmap*, respectively. Unlike [33], in this paper, the *Nmap* and *Snort* procedures are performed in different computers, and there are constraints to ensure the time synchronization of data in these different machines.

The automation scripts run on both the *Nmap* and *Snort* machines. They do not communicate directly with one another because the ACL rules programmed on the L3 switch only allow DNP3 traffic to enter the substation VLAN. Instead, they rely upon a schedule. For the fast use cases, the trial duration is three minutes, while for the slow trial the duration time is five minutes.

The *Nmap* script has about a 30-second delay (*Wait Time*) after the start of each trial to ensure that the *Snort* script can properly restart *Snort* and *Wireshark* before the *Nmap* scan begins. Then, the *Nmap* process will start on the *Nmap* machine, taking on average one minute and 55 seconds to run. Next, there will be approximately a one-minute and 30-second gap before the next trial starts. This ensures that there is no packet information contamination between trials.

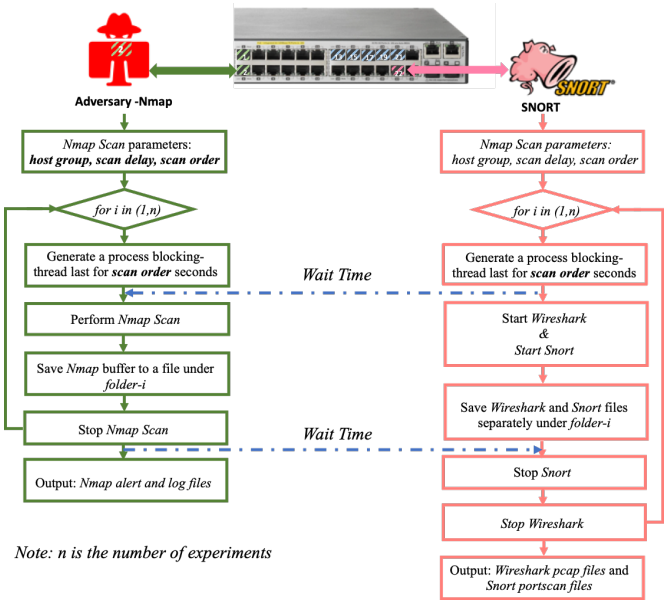


Fig. 4: Automated Process of *RESLab* Testbed Data Collection of *Nmap* and *Snort*.

Because the scripts run using a scheduler, it is critical that both scripts are synchronized. In our testbed, the devices are run on two laptops. If the devices are left to run without synchronizing their internal clocks, *Snort* and *Wireshark* could collect *Nmap* traffic from a past or future trial and incorrectly store it in the wrong trial folder. This could lead to misreporting results. To prevent data contamination, before the trial starts both stand-alone computers are connected to the internet, and their internal clocks synchronize by connecting to a public Network Time Protocol (NTP) server. Global Positioning System (GPS) satellites can also be used to synchronize the time if an NTP server is not available. Then, both computers are disconnected from the internet, and the data collection process for each use case is started.

With the automated process, all port scanning scenarios run 1000 replicates (experimental runs) to generate datasets for the following analyses.

3.6 Lessons Learned from Physical Validation Experiment

The purpose of the physical validation experiment is to validate that the emulated environment that researchers commonly use to generate datasets is accurate. While the TAMU team conducted the physical validation experiments, a few key differences were observed. The following is a list of lessons learned from the physical validation experiment.

- There was an issue when the L3 switch's monitoring port was used to monitor all the substation ports (Port 1-24) on the L3 switch. When *Wireshark* ran on the *Snort* machine, the TAMU team noticed that packets were being monitored as they entered into the L3 switch and as they left the L3 switch. This caused *Snort* to double the amount of alerts that it was generating from the *port_fs_scan* alerting system. To counteract the double counting of packets, only packets entering and leaving the Substation VLAN were assigned to the switch's monitoring port.
- The *Nmap* machine was not able to scan the RTAC or relays in the substation network initially because the RTAC and relays allowed only certain IP addresses, i.e., they had only one DNP3 master. The only IP address allowed was the RTAC in the Control Center. This issue was resolved by "whitelisting" the IP address of the *Nmap* machines to all the devices in the Substation Network. In other words, the RTAC and relays in the substation have two DNP3 masters. However, in an actual attack environment, the adversary would have to impersonate the Control Center RTAC as in Fig. 2(b).
- A timestamp issue occurs between computers used in the data collection process. We observed that the internal clock of the *Nmap* and *Snort* machines had to be synchronized using the network before the experiment starts.
- *Nmap* was not able to be run on a Windows OS machine for these experiments. The TAMU team observed that even when the Windows firewall is turned off, a windows machine is still unable to scan network devices of a VLAN that it is not connected to. For example, if a Windows *Nmap* machine is attached to a 192.168.1.0/24 it can scan network devices within the 192.168.1.0/24, but it cannot scan network devices in a 192.168.2.0/24 network. The error *Nmap* produced was a "128.128.128.128 authorization not allowed" error. Initially, this was thought to be the ACL rules configured on the L3 switch, but after using a Kali Linux machine to scan the substation network, it was noted that the Windows kernel did not allow the scan of an IP address outside of its LAN broadcast domain. To solve this issue, a

Live version of Kali Linux 2020.1 was used to run *Nmap* on the *Nmap* machine.

- The original emulation experiments in [32] had packet losses configured in all the machines. However, we were not able to configure packet losses in the RTACs and relays. In our experiments, packet losses were configured only on the inbound interface of the *Nmap* scanning laptop.

4 Analyses of Datasets

4.1 Brief Review of the Mathematical Model of Discovery and Detection of Port Scanning Attack

In [32], the authors consider a hypothetical reconnaissance attack on an ICS for an electrical power system. They assume the adversary has established a foothold on the SCADA network and is using network scanning tools to identify vulnerable RTUs (e.g., due to firewall misconfigurations) against which payloads can be deployed. They also assume that an IDS has been installed on the SCADA network to detect unauthorized scanning activity. The study in [32] introduces a probabilistic, mathematical model that describes the rate at which the adversary identifies vulnerable RTUs and the probability that the adversary will be detected by an IDS over time. The model defines the state of the system at a discrete timestep to be the number of RTUs that have been categorized as vulnerable, secure, undetermined, or not yet scanned, and the state is updated at each timestep. The order in which RTUs are scanned and the dropping of packets are assumed to be stochastic and modeled with probability distributions. The speed of vulnerability discovery is further determined by parameter settings of the network scanning tool, and detection of network scanning is subject to logic specified in the IDS rules. The authors validate the mathematical model via comparison with a virtual testbed (emulation). The mean estimates for vulnerability discovery and probability of detection produced by the model fall within 95% confidence intervals for mean estimates obtained from the emulation experiment. For details on the mathematical model and the virtual testbed validation, please refer to [32], [33].

Fig. 5 displays the mean port count results for the slow discovery scenario described in Section 2.1. These results have all sources of randomness, i.e., packet drops are included and *Nmap* search address sequences are random and varied. Results are shown for the emulated and physical experiments. Additionally, results of predicted port discovery count are included. The predicted values are calculated using the mathematical model described in [32]. As discussed in [32], the mean quantities estimated by the math model fell within the 95% confidence intervals on the mean estimates from the emulated runs, and Fig. 5 shows the mathematical model results agree well with both the emulated and hardware-based results. However, the analysis included herein does not include further comparison or discussion of the results from [32]. The analysis techniques used to compare experimental results require discrete results from multiple trials at each timestep. Because the mathematical model presented in [32] uses an analytical and not Monte Carlo style approach, the format of the results from the mathematical model are not amenable to the analysis techniques discussed in this paper.

Because the laptop that is connected to the switch's monitor port runs *Snort* and collects packet captures, a common clock reference is used for *Snort* alerts and packets. A post-processing script uses data from the packet capture file to determine when *Nmap* scanning starts, and correlates this with the *Snort* event log data to determine the elapsed time between the start of scanning and the subsequent alert, which is the port discovery activities from *Nmap*. The same post-processing script is used for emulation and physical testing data. Thus, this section presents an analysis of the port discovery results using the Kolmogorov-Smirnov (KS) test and a "bootstrap" method. This statistical analysis concentrates on *open* ports, because those are represented with physical devices in the TAMU experiment. For this reason, much of the analysis focuses on validation comparison of *open* port discovery between SNL's `minimega` emulation and TAMU's *RESLab* testbed. The data sets from two testbeds under port scanning attacks are published at [41].

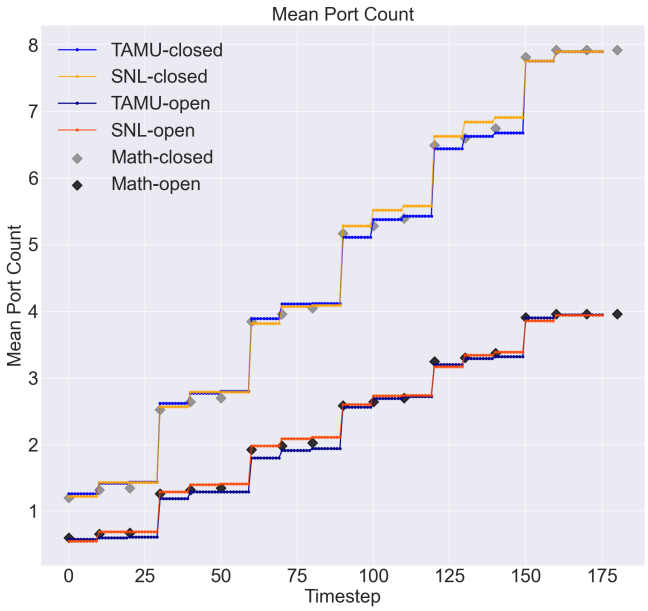


Fig. 5: Mathematical model [32] validation with `Minimega` and *RESLab* under slow discovery of *open* and *closed* ports, random formulation.

4.2 Kolmogorov-Smirnov (KS) Test

The Kolmogorov-Smirnov (KS) test statistic is a well-known non-parametric statistical test for equality of distributions [42]. Due to inherent randomness in the scenarios, multiple replicates are run in the emulation and physical experiments to facilitate a statistical comparison between them. The KS statistic has a number of desirable features for performing this comparison. It considers ensemble distributions of results from both testbeds, and produces results that indicate the degree of correspondence and provide a threshold for deciding whether the two distributions match.

The KS distance between two random variables is the maximum vertical distance between their cumulative distribution functions (CDFs), $CDF_1(x)$ and $CDF_2(x)$. The KS test involves testing for equality of these CDFs. That is, the null hypothesis is $H_O : CDF_1 = CDF_2$, while the alternative hypothesis is $H_1 : CDF_1 \neq CDF_2$. Equation (1) is used to test for the equality of these distributions given n samples from distribution 1 and m samples from distribution 2. $D_{n,m}$ converges to a Kolgmorov-Smirnov distribution as n and m become large. Note that $CDF_{i,n}(x)$ refers to the CDF from distribution i based on n sample points. The test statistic ($D_{m,n}$) allows rejection of the null hypothesis at various confidence levels α.

$$D_{n,m} = \sup_x |CDF_{1,n}(x) - CDF_{2,m}(x)|$$
$$\text{Reject } H_0 \text{ if } D_{n,m} > \sqrt{-ln(\alpha/2)\left(\frac{1+m/n}{2m}\right)} \quad (1)$$

The p-value is the probability that one would obtain KS results at least as extreme as what were observed, assuming the null hypothesis is correct. A p-value of 1.0 indicates perfect agreement between the distributions. We use a significance level threshold of 0.05 to determine when results differ: if the p-value is less than 0.05, then the distributions are considered to be statistically significantly different. We chose this threshold as it is most commonly used in statistical test comparisons; it roughly corresponds to rejecting a value that is more than two standard deviations from a mean in a Gaussian case. We do note, however, that there is a revival of statistical interest in the topic of p-values and proposals to lower the threshold to 0.005 [43], [44]. Until this debate is more settled, we will keep the threshold at 0.05. The p-value is the exceedance value for $D_{n,m}$ according to the KS distribution that follows the null hypothesis. The CDF of the KS distribution is difficult to calculate analytically. There are a number of numerical approximations used [45] [46], [47]. We used Matlab's implementation [48], shown as a one-sided version in Equation (2).

$$nt = \frac{nm}{n+m}$$
$$\lambda = max((\sqrt{nt} + 0.12 + 0.11/\sqrt{nt}) * D_{n,m}, 0) \quad (2)$$
$$\text{p-value} = e^{(-2\lambda^2)}$$

The main point to remember about the KS test is that it compares agreement of distributions, not just agreement of means as the t-test does. Also recall that the data in these experiments is recorded at each second (e.g., number of *open* ports found at 7 seconds, number of *closed* ports found at 15 seconds, etc.). Thus, at each second, we perform a full distribution comparison based on the 1000 replicates from SNL and the 1000 replicates from TAMU datasets. That is, $n = m = 1000$ in this study. The 1000 replicates form a distribution, and the SNL distribution and TAMU distribution of *open* ports discovered are the items being compared.

We also show the area metric. Similar to the KS test, the area metric also quantifies the difference between sample CDFs. It accounts for the entire difference (e.g., the area) between the two functions rather than just the maximum vertical distance. The area metric does not have a formal acceptance measure: it does not follow a parametric distribution. However, a value of 0.0 indicates perfect agreement (the two CDFs are identical) and a larger value indicates a higher level of disagreement.

4.3 Results from KS test along with Port Discovery

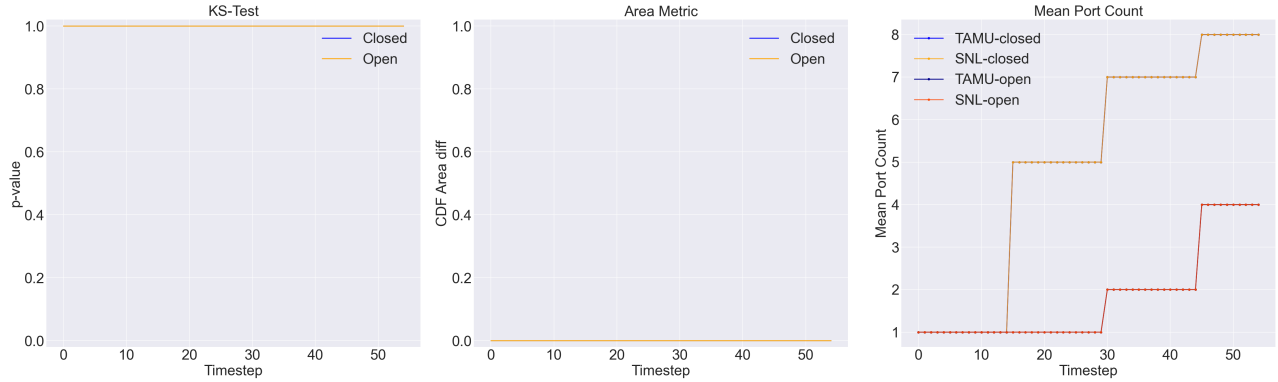
In this section, we show the results of the KS test results and the area metric along with the port discovery for four scenarios: the fast and slow *Nmap* scanning scenarios (determined by the *max-host-group*, *min-host-group* and *scan-delay* settings in *Nmap*, described in Section 2.1) with fixed port ordering and no packet drops (these are "fixed" deterministic scenarios with no randomness), and the fast and slow scenarios with random port ordering and 10% packet drops (these are the "most random" of the combinations that are tested).

The SNL team started with the fixed scenarios as shown in Figures 6(a) and 6(b). As expected, the results in Figures 6(a) and 6(b) show complete agreement, with p-values of 1.0 for the KS-test and area metric values of 0.0 for all time points throughout the scenario, whose plots are overlapped. This occurs for both *open* and *closed* ports. The time traces of the mean number of *open* and *closed* ports found for TAMU and SNL are shown in the graphs on the right. The closed ports are overlapping and thus the test statistics for closed are identical to those of open and also the TAMU results are identical to the SNL results for these deterministic cases.

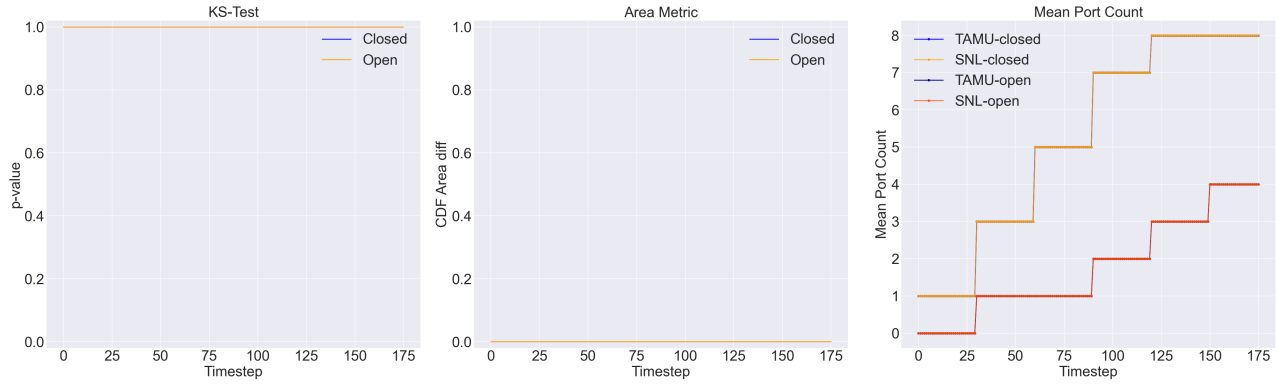
The results for the random scenarios are shown in Figures 6(c) and 6(d). In these figures, the mean values of the *open* and *closed* ports discovered over time do differ slightly. The distribution comparison tests, as shown in the left figures, all have p-values larger than 0.05, indicating that we would not reject the null hypothesis that these two samples come from the same distribution. However, there are some areas in the figures that have lower p-values than others. For example, the *closed* ports show lower p-values between 80 and 130 seconds in Fig. 6(d) and the mean values of *closed* ports found differ in this time interval also. The area metric remains fairly constant: this indicates that the total area between the two CDFs remains relatively unchanged. The KS test is more likely to be affected by a few differences in the discrete distribution values.

The difference between p-values for KS-Test in Fig. 6(c) is not a huge disagreement. Recall that the KS test compares agreement of distributions, not just agreement of means as the t-test does. The distributions would only be considered statistically significantly different if the p-value is less than 0.05: this does not occur in Fig. 6(c). One can say, however, that there is a tighter distribution agreement over time: the KS-test statistic is one by the end of the simulation at 50 seconds, indicating perfect agreement between the two distributions. Finally, recall that at each second of the experiment, we perform a full distribution comparison based on the 1000 replicates from SNL and the 1000 replicates from TAMU datasets. It is a strong statistical result indicating that these two distributions involving 1000 replicates have the same distribution by the end of the experiment.

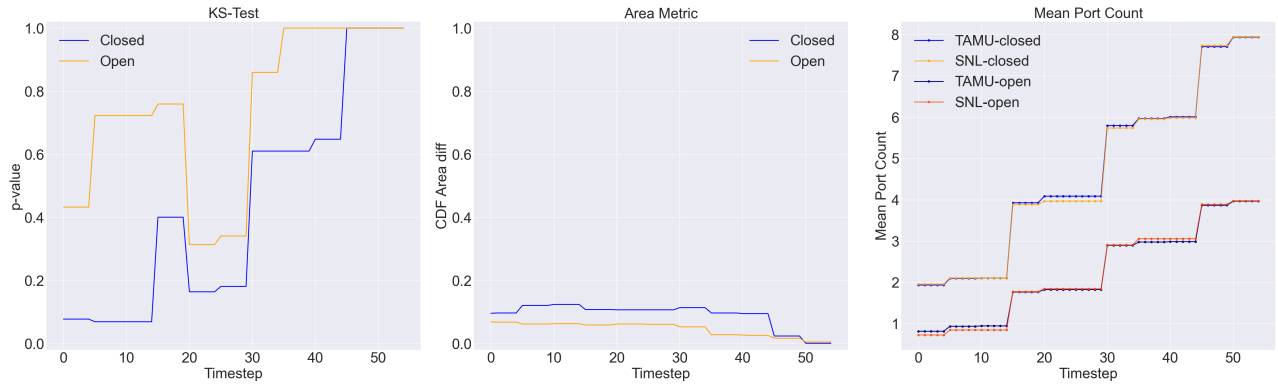
As described in [33], the KS statistic represents one of a number of possibilities for comparing results, and more research is needed to determine whether, and under what



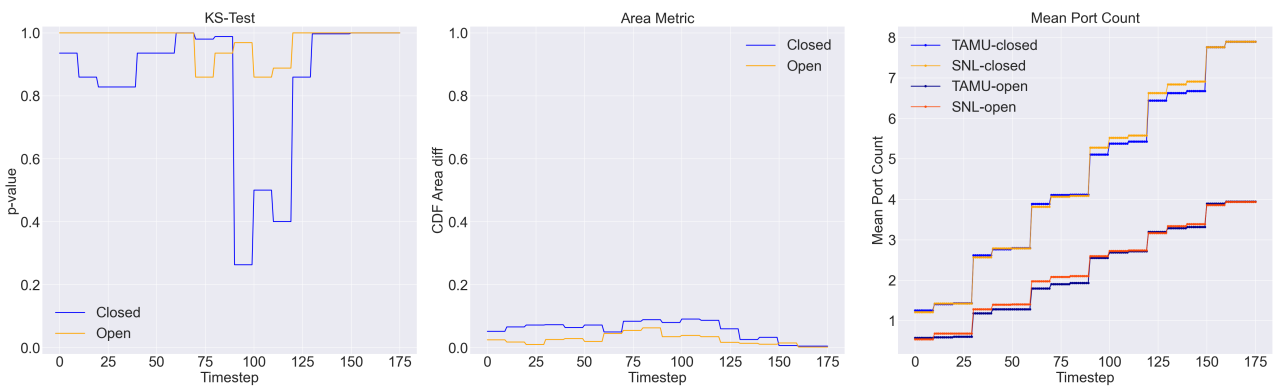
(a) Fast discovery of *open* and *closed* ports, deterministic formulation.



(b) Slow discovery of *open* and *closed* ports, deterministic formulation.



(c) Fast discovery of *open* and *closed* ports, random formulation.



(d) Slow discovery of *open* and *closed* ports, random formulation.

Fig. 6: Analyses of data from `Minimega` and *RESLab* under four port scanning scenarios.

conditions, other metrics should be used to perform similar comparisons.

4.4 Bootstrap and Additional Statistical Analyses

We found that the results of the KS test can be quite noisy, especially for the *open* ports. The reason is that there are only five values for the *open* ports: 0, 1, 2, 3, or 4 *open* ports may be found at a particular timestep. Slight differences in the replicates may result in fairly large differences in the distributions and CDFs. For example, of 1000 replicates at twenty seconds, the TAMU results may show 200 replicates with zero *open* ports and 600 replicates with one *open* port found (with the remaining replicates having more than one port found). The SNL results may show 250 replicates with zero *open* ports and 630 replicates with one *open* port found with the rest having more than one port found (for example). These discrete values for the distribution make the histogram "choppier" than if it were a continuous distribution, especially because the support of the distribution is limited to the five values.

To further investigate this, we use a method similar to a "bootstrap" approach in statistics [49], where we take many random draws from the full datasets with 1000 replicates each. The overall bootstrap method we used is outlined in Algorithm 1, with B=500 and n=1000.

Algorithm 1 Bootstrap Method

1: Input: a sample from population with sample size n
2: **for** i in (1, B) **do**
3: Draw a sample of size n from the original sample with replacement: this is a bootstrap sample
4: Evaluate the statistic of interest Q (e.g., mean, median, percentile) on the bootstrap sample
5: **end for**
6: Use the empirical distribution of the B values of Q to identify the mean, median, 5^{th} and 95^{th} percentile of Q
7: Output = Statistics on Q from the set of B boostrap samples

We compare each of these "bootstrapped" distributions of TAMU results and SNL results and generate the p-value plots. The bootstrap method allows us to generate an ensemble of possible realizations of the p-value traces. We can then examine the entire ensemble to understand how much uncertainty is associated with the p-values.

Since the deterministic results had complete agreement, there was no need to do this further statistical investigation. To keep the number of figures reasonably concise, we only performed this analysis on the random results. Random port ordering and random packet drops for the open ports tend to have more variable results than closed ports, because there are more open ports to search. Thus, Fig. 7 shows the statistical quantities indicating the range of p-values based on the 500 bootstrap samples, where each bootstrap sample compares a distribution generated by 1000 randomly chosen replicates from the TAMU and SNL datasets. Repeating this process for 500 bootstrap samples provides a sense of the range and variability in p-values.

One can see that the 95^{th} percentile of p-values in the bootstrap set (shown in blue) is 1.0, indicating that the

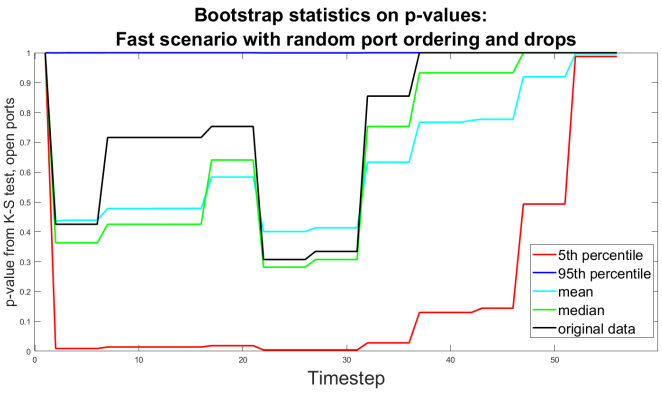


Fig. 7: Bootstrap statistics on the p-values for Fast discovery of *open* ports.

top 5% of the bootstrap samples have perfect distribution agreement. The mean and median (in cyan and green, respectively) show p-values ranging from about 0.3 to 1.0 near the end time of the fast scenario. The mean and median p-values are reasonably close to the p-values from the comparison of the original TAMU and SNL datasets which is shown in black. Finally, the bottom 5^{th} percentile of p-values is very low, below 0.05 and sometimes near zero, indicating rejection of the null hypothesis. Note, however, that after 35 seconds, the p-values of the 5^{th} percentile increase. When looking at the entire ensemble statistics for the p-values including the mean, median, and the fifth and ninety-fifth percentiles, we find the evidence strongly supports the acceptance of the null hypothesis. That is, most of the statistical measures indicate that the distribution of p-values would be acceptable and we would not tend to reject the null hypothesis. There is high variability in the p-values based on the discrete nature of the *open* port distribution. In aggregate, Fig. 7 quantifies that variability using a bootstrap sample.

Fig. 8 shows a comparable set of 500 bootstrap samples for the random slow case. Again, each of the 500 bootstrap samples compares a distribution generated by 1000 replicates randomly chosen from the TAMU data and 1000 replicates randomly chosen from SNL data. While this figure still shows high variability in p-values, it is not as large as the variability in the fast random results.

The overall p-values for the original set of 1000 TAMU and 1000 SNL samples are shown in the black line in Fig. 8. The statistics from the bootstrapping are shown in the colored lines: the red line is the 5^{th} percentile, the blue line is the 95^{th} percentile, and the cyan and green lines are the mean and median p-values, respectively. Again, these statistics of the bootstrap ensemble provide a bound on what we can expect when repeating this experiment.

Fig. 8 indicates a much tighter distribution of results than Fig. 7. The black line comparing the original results indicates that the results between TAMU and SNL are in strong agreement and only differ between about 75 and 120 seconds. The 95^{th}-percentile, the mean, and the median have high p-values. The 5^{th} percentile only goes below 0.05 in the 75 to 120 second window and otherwise is above 0.05. The differences between the slow random and fast random

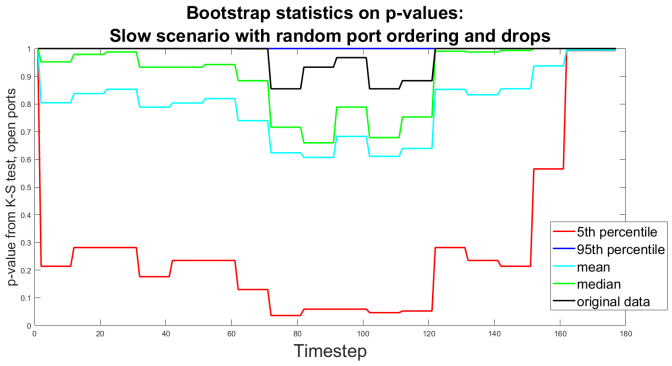


Fig. 8: Bootstrap statistics on the p-values for Slow discovery of *open* ports.

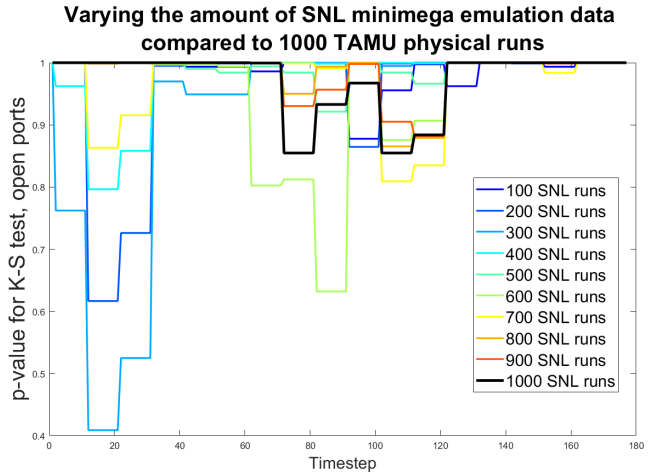


Fig. 9: Convergence study of the p-values for Slow discovery of *open* ports.

scenario are to be expected: the fast scan does not allow much time for port discovery and so the differences are compressed into a 55 second window, in contrast to the slow scanning scenario. It was also noted that the statistical test results shown in Fig 7 and 8 both show the results becoming more similar near the end of the experiments. This is to be expected because, eventually, all four *open* ports are found and the distributions become very similar or identical.

As a final statistical study, the SNL team analyzed the convergence of the p-values as we increase the amount of data available. The 1000 TAMU results are the baseline physical test results against which we measure the agreement of the SNL `minimega` emulation. We analyzed the slow, random case just discussed. Fig. 9 shows the use of 100 SNL `minimega` runs, followed by 200, then 300, etc. The idea is to increase the amount of emulation data from Sandia's emulation testbed in the comparison to the 1000 TAMU physical test results. Fig. 9 does not show a monotonic convergence. That is, the darkest blue lines do not have the lowest p-values. During the period between 60 and 120 seconds, for example, the datasets with the lowest p-values are those in green, yellow, or orange hues, indicating a larger number of samples being used in the Kolmogorov-Smirnov test comparison. Again, this discrepancy is believed to be caused by the small number of *open* ports and the discrete nature of the distributions that we are comparing. Although Fig. 9 does not indicate monotonic convergence, all of the p-values are well above 0.05 (the y-axis has a lower bound of 0.4 in Fig. 9) and thus the null hypothesis that these distributions are similar cannot be rejected.

The results discussed in Figures 6-9 all involve comparison of emulations with the physical experiment. As mentioned, we previously compared the performance of the `minimega` emulation with the CORE emulation for reproducibility [33]. While the emulation-to-emulation comparison had a different focus on reproducibility of the study across emulation platforms, it is instructive to consider the relative differences in that study compared to this one. As an example, Fig. 10 shows that the slow discovery case with random port ordering and packet drops has differences in mean number of *open* or *closed* ports found that may be as large as 0.2, depending on the timestep. Fig. 10 shows these differences especially significant between 30 and 150 seconds for *open* ports found, as demonstrated by the differences in the mean traces and the p-values.

Fig. 10 can be compared with Fig. 6(d). We do not want to over-interpret this comparison as the emulation-to-emulation results in Fig. 10 are based on 100 samples from each emulation whereas the results in Fig. 6(d) are based on 1000 samples from the emulation and the physical experiment. The p-values can be influenced by the variance of the number of ports found, which does depend on the number of samples. However, the relative differences in the mean number of *closed* and *open* ports found in the subfigures on the right side is the data we wish to compare. We simply observe that the `minimega` comparison to the physical experiments is not noticeably different from its comparison to the CORE emulation. If Fig. 10 were drastically different from Fig. 6(d), we would be concerned that there was a fundamental issue with one of the studies. The fact that they both give similar results adds credibility to this particular experiment: the emulation-emulation and emulation-physical testbed serve as a cross-validation exercise to a certain extent because all of the experiments (emulation or physical testbed) were running the same port discovery scenarios with *Nmap* and *Snort*.

5 Conclusions and Future Work

The KS statistics and bootstrapping results indicate that one would not reject the hypothesis that there is agreement between the `minimega` emulation results and the *RESLab* testbed results. Although there were some differences in the *RESLab* testbed setup, due to differences in testbeds and the fact that the *RESLab* setup used physical hardware, these differences mapped well to the original `minimega` experiments and did not result in substantial differences in the port discovery results. Therefore, we consider the original `minimega` emulation-based model to be validated with respect to physical experiments, for both the deterministic and stochastic scenarios. Nevertheless, we found that substantial effort by subject matter experts was required to configure and verify/debug the physical experiment in order to reproduce the emulation-based experiment in hardware, and we presented these findings.

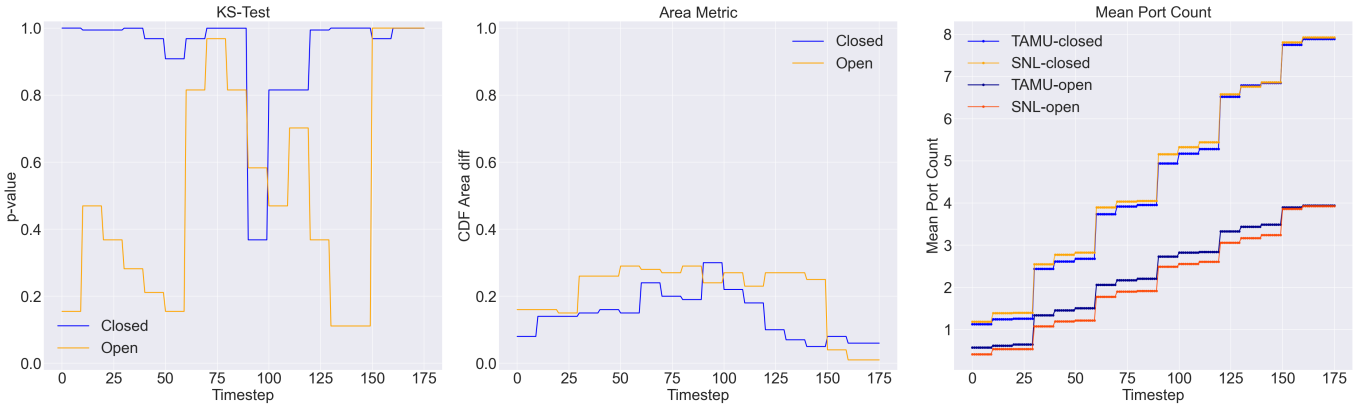


Fig. 10: Analysis of data from `Minmega` and *CORE* under slow discovery of *open* and *closed* ports, random formulation [33].

In addition, this work presented a process for portability and comparisons of scenarios between emulation-based and physical testbeds. It opens the door to further studies and research on reproducibility for improving the science of security in validation. The presented experiment, case studies and statistical analysis tools can be generalized with different attack scenarios, such as Distributed Denial of Service, Man-in-the-Middle, and SQL injection, with corresponding parameters. It is also of great interest to expand the *RESLab* testbed with different manufacturers' devices and consider different types of communication protocols, such as IEC 61850, ICCP/TASE.2, and Modbus, to further replicate the real substation automation system and SCADA systems. However, there are challenges to run control studies on these more complex attack scenarios due to their variability. Future work to enhance the generalization of the capabilities, including to reduce the amount of expert input needed during the process, would be worthwhile. As the by-product, the data sets (1000 samples of attack in each scenario) generated from the physical testbed can be used for the cybersecurity studies with data-driven approaches and validation for other testbeds.

Acknowledgment

The authors thank Ali Pinar for his leadership of this research program, and Jerry Cruz, Christian Reedy, and Seth Hanson for orchestrating and running the Sandia `minimega` emulation experiments. We are grateful for the support they provided. This work has been supported by the LDRD Program at the Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

[1] R. Bejtlich, *The Tao of network security monitoring: beyond intrusion detection*. Pearson Education, 2004.

[2] L. Martin, "Cyber kill chain," *URL: http://cyber. lockheedmartin. com/hubfs/Gaining the Advantage Cyber Kill Chain. pdf*, 2014.

[3] M. J. Assante and R. M. Lee, "The industrial control system cyber kill chain," *SANS Institute InfoSec Reading Room*, vol. 1, 2015.

[4] Defense Use Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, vol. 388, 2016.

[5] E. Targett. (2020, March) High Voltage Attack: EU's Power Grid Organisation Hit by Hackers. [Online]. Available: https://www.cbronline.com/news/eu-power-grid-organisation-hacked

[6] A. Hobbs, *The colonial pipeline hack: Exposing vulnerabilities in us cybersecurity*. SAGE Publications: SAGE Business Cases Originals, 2021.

[7] W. Mazurczyk and L. Caviglione, "Cyber reconnaissance techniques," *Communications of the ACM*, vol. 64, no. 3, pp. 86–95, 2021.

[8] M. Shah, S. Ahmed, K. Saeed, M. Junaid, H. Khan *et al.*, "Penetration testing active reconnaissance phase–optimized port scanning with nmap tool," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2019, pp. 1–6.

[9] M. Malkawi, T. Özyer, and R. Alhajj, "Automation of active reconnaissance phase: an automated api-based port and vulnerability scanner," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2021, pp. 622–629.

[10] X. Li, J. D. Smith, and M. T. Thai, "Adaptive reconnaissance attacks with near-optimal parallel batching," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 699–709.

[11] S. K. Patel and A. Sonker, "Rule-based network intrusion detection system for port scanning with efficient port scan detection rules using snort," *International Journal of Future Generation Communication and Networking*, vol. 9, no. 6, pp. 339–350, 2016.

[12] C. Birkinshaw, E. Rouka, and V. G. Vassilakis, "Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks," *Journal of Network and Computer Applications*, vol. 136, pp. 71–85, 2019.

[13] A. Sahu, Z. Mao, P. Wlazlo, H. Huang, K. Davis, A. Goulart, and S. Zonouz, "Multi-source multi-domain data fusion for cyberattack detection in power systems," *IEEE Access*, vol. 9, pp. 119 118–119 138, 2021.

[14] S. Mubarak, M. H. Habaebi, M. R. Islam, A. Balla, M. Tahir, E. A. Elsheikh, and F. Suliman, "Industrial datasets with icss testbed and attack detection using machine learning techniques," *Intelligent Automation & Soft Computing*, vol. 31, pp. 1345–1360, 2022.

[15] S. Achleitner, T. F. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha, "Deceiving network reconnaissance using sdn-based virtual topologies," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1098–1112, 2017.

[16] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "An effective address mutation approach for disrupting reconnaissance attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2562–2577, 2015.

[17] K. Wang, X. Chen, and Y. Zhu, "Random domain name and address mutation (rdam) for thwarting reconnaissance attacks," *PLOS ONE*, vol. 12, no. 5, pp. 1–22, 2017.

[18] V. K. Singh and M. Govindarasu, "A novel architecture for attack-resilient wide-area protection and control system in smart grid," in *2020 Resilience Week (RWS)*. IEEE, 2020, pp. 41–47.

[19] R. Chadha, T. Bowen, C.-Y. J. Chiang, Y. M. Gottlieb, A. Poylisher, A. Sapello, C. Serban, S. Sugrim, G. Walther, L. M. Marvel *et al.*, "Cybervan: A cyber security virtual assured network testbed," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 1125–1130.

[20] minimega developers, "minimega: a distributed vm management tool," 2019. [Online]. Available: https://minimega.org/

[21] U. Adhikari, T. Morris, and S. Pan, "Wams cyber-physical test bed for power system, cybersecurity study, and data mining," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2744–2753, 2016.

[22] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "Scada system testbed for cybersecurity research using machine learning approach," *Future Internet*, vol. 10, no. 8, p. 76, 2018.

[23] A. Sahu, P. Wlazlo, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Design and evaluation of a cyber-physical testbed for improving attack resilience of power systems," *IET Cyber-Physical Systems: Theory & Applications*, vol. 6, no. 4, pp. 208–227, 2021.

[24] J. Dykstra, *Essential cybersecurity science: build, test, and evaluate secure systems*. " O'Reilly Media, Inc.", 2015.

[25] W. L. Oberkampf and C. J. Roy, *Verification and validation in scientific computing*. Cambridge University Press, 2010.

[26] N. R. Council *et al.*, *Assessing the reliability of complex models: mathematical and statistical foundations of verification, validation, and uncertainty quantification*. National Academies Press, 2012.

[27] K. A. Maupin, L. P. Swiler, and N. W. Porter, "Validation metrics for deterministic and probabilistic data," *Journal of Verification, Validation and Uncertainty Quantification*, vol. 3, no. 3, 2018.

[28] S. T. Jones, K. G. Gabert, and T. D. Tarman, "Evaluating emulation-based models of distributed computing systems," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2017.

[29] J. Crussell, T. M. Kroeger, A. Brown, and C. Phillips, "Virtually the same: Comparing physical and virtual testbeds," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 847–853.

[30] J. Crussell, T. M. Kroeger, D. Kavaler, A. Brown, and C. Phillips, "Lessons learned from 10k experiments to compare virtual and physical testbeds," in *12th {USENIX} Workshop on Cyber Security Experimentation and Test ({CSET} 19)*, 2019.

[31] X. Zheng and C. Julien, "Verification and validation in cyber physical systems: Research challenges and a way forward," in *2015 IEEE/ACM 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems*. IEEE, 2015, pp. 15–18.

[32] E. Vugrin, J. Cruz, C. Reedy, T. Tarman, and A. Pinar, "Cyber threat modeling and validation: port scanning and detection," in *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*. Association for Computing Machinery, 2020.

[33] T. Tarman, T. Rollins, L. Swiler, J. Cruz, E. Vugrin, H. Huang, A. Sahu, P. Wlazlo, A. Goulart, and K. Davis, "Comparing reproduced cyber experimentation studies across different emulation testbeds," *Proceedings, 14th Workshop on Cyber Security Experimentation and Test (CSET21), ACM*, 2021.

[34] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "Core: A real-time network emulator," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, 2008, pp. 1–7.

[35] G. Lyon, "Nmap: the network mapper," 2019, http://nmap.org/.

[36] Cisco, "Snort intrusion detection and prevention system," 2019, https://www.snort.org/.

[37] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: the linux virtual machine monitor," in *Proceedings of the Linux symposium*, vol. 1. Dttawa, Dntorio, Canada, 2007, pp. 225–230.

[38] L. Foundation, "Open vswitch," 2019. [Online]. Available: http://www.openvswitch.org/

[39] H. Huang, P. Wlazlo, Z. Mao, A. Sahu, K. Davis, A. Goulart, S. Zonouz, and C. M. Davis, "Cyberattack defense with cyber-physical alert and control logic in industrial controllers," *IEEE Transactions on Industry Applications*, vol. 58, no. 5, pp. 5921–5934, 2022.

[40] P. Wlazlo, A. Sahu, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Man-in-the-middle attacks and defence in a power system cyber-physical testbed," *IET Cyber-Physical Systems: Theory & Applications*, vol. 6, no. 3, pp. 164–177, 2021.

[41] H. Huang, P. Wlazlo, A. Sahu, A. Walker, A. Goulart, K. Davis, L. Swiler, T. Tarman, and E. Vugrin, "Dataset of port scanning attacks on emulation testbed and hardware-in-the-loop testbed," 2022. [Online]. Available: https://dx.doi.org/10.21227/cva5-nd75

[42] "NIST/SEMATECH e-handbook of statistical methods," National Institute of Standards, Tech. Rep., 2012. [Online]. Available: http://www.itl.nist.gov/div898/handbook/

[43] D. J. Benjamin, J. O. Berger, M. Johannesson, B. A. Nosek, E.-J. Wagenmakers, R. Berk, K. A. Bollen, B. Brembs, L. Brown, C. Camerer *et al.*, "Redefine statistical significance," *Nature Human Behaviour*, vol. 2, no. 1, pp. 6–10, 2018.

[44] G. Di Leo and F. Sardanelli, "Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach," *European Radiology Experimental*, vol. 4, no. 1, pp. 1–8, 2020.

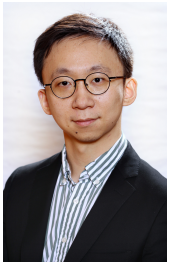
[45] G. Marsaglia, W. W. Tsang, and J. Wang, "Evaluating kolmogorov's distribution," *Journal of Statistical Software*, vol. 8, pp. 1–4, 2003.

[46] D. E. Knuth, *Art of computer programming, volume 2: Seminumerical algorithms*. Addison-Wesley Professional, 1998.

[47] R. Simard and P. L'Ecuyer, "Computing the two-sided kolmogorov-smirnov distribution," *Journal of Statistical Software*, vol. 39, pp. 1–18, 2011.

[48] "kstest,kstest2," 2021, [Online; accessed Feb. 20,2022]. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/

[49] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.



Hao Huang (Member, IEEE) : received the B.S. degree in Electrical Engineering (Power System and Its Automation) from Harbin Institute of Technology, Harbin, Heilongjiang Province, China, in 2014; the M.S. degree in Electrical Engineering (Electric Power) from University of Southern California, Los Angeles, CA, USA, in 2016; the Ph.D. degree in Electrical Engineering at Texas A&M University in 2022. He is a postdoc research associate at Princeton University. His research focuses on power system resilience, data-driven approaches for power systems, cyber-physical security, etc.

Patrick Wlazlo received his B.S. Degree in electrical engineering systems technology and his M.S. Degree in engineering technology from Texas A&M University in 2020 and 2022 respectively. He is working at Vistra Corp as Development Security Operations Engineer. His research interests involve studying cybersecurity as it relates to electrical transmission and generation systems.

Abhijeet Sahu (Graduate Student Member, IEEE) received his B.S. degree in Electronics and Communications from National Institute of Technology, Rourkela, India, in 2011 and his M.S. and Ph.D degree in Electrical and Computer Engineering from Texas A&M University, TX, USA in 2018 and 2022, respectively. He is a researcher at National Renewable Energy Laboratory. His research interests include network security, cyber-physical modeling for intrusion detection and response, and Artificial Intelligence for cyber-physical security in power systems.

Adele Walker received her B.B.A in Management Information Systems and a minor in Cybersecurity from Texas A&M University in 2022. She is pursuing a career in federal cybersecurity with focuses in network security and SCADA systems. Her research interests include network and API security, as well as precision in the prediction and response to attacks on critical infrastructure.

Ana E. Goulart (Member, IEEE) received a bachelor's degree in electrical engineering from the Federal School of Engineering of Itajuba (EFEI), Brazil; a M. Sc. degree in Information Systems Management from the Pontificial Catholic University of Campinas, Brazil; a M. Sc. in Computer Engineering at North Carolina State University; a Ph.D. in Electrical and Computer Engineering at Georgia Tech, Atlanta, GA. She is a Professor in the Electronics Systems Engineering Technology program at Texas A&M University, in College Station, TX. Her research interests include protocols for real-time voice and video communications and their performance, IP-based emergency communications, last-mile communication links and cybersecurity for the smart grid, wireless network systems, and rural telecommunications.

Katherine R. Davis (Senior Member, IEEE) received the B.S. degree from The University of Texas at Austin, Austin, TX, USA, in 2007, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2009 and 2011, respectively, in electrical engineering. She is an Associate Professor of electrical and computer engineering with Texas A&M University. Her research interests include operation and control of power systems, interactions between computer networks and power networks, security-oriented cyber-physical analysis techniques, data-driven and model-based coupled infrastructure analysis and simulation.

Laura Swiler is a computational scientist at Sandia National Laboratories whose research focuses on quantifying the uncertainty associated with predictions from models. Her research interests include experimental design, sampling algorithms, Bayesian inference, and surrogate models. Laura has worked on many applications, including nuclear waste repository assessment, circuit model calibration, additive manufacturing, and cyber emulation.

Thomas D. Tarman is a distinguished member of the technical staff at Sandia National Laboratories, in Albuquerque New Mexico, where he leads research in virtualization and rigorous cyber experimentation methodologies, with application to high-consequence cyber systems. His research interests are in network modeling and simulation, hybrid simulation-emulation-physical testbeds for cyber-security research, and network security protocols.

Eric Vugrin received BS and MS degrees in Mathematics from Texas Tech University in 1998 and 2000, respectively, and a PhD in Mathematics from Virginia Tech in 2004. He is currently a Distinguished Member of Technical Staff with Sandia National Laboratories in Albuquerque, New Mexico, USA. His research interests include cyber resilience, mathematical modeling, and emulation.