Interfacing Techniques in Testbed for Cyber-Physical Security Analysis of the Electric Power Grid

Venkatesh Venkataramanan, *Student Member, IEEE,* Pengyuan Wang, *Student Member, IEEE,*
Anurag Srivastava, *Senior Member, IEEE,*, Adam Hahn, *Member, IEEE*, and
Manimaran Govindarasu, *Fellow, IEEE.*

***Abstract*—The electric power grid is an heterogeneous cyber-physical system with various physical, cyber/communication, computation, and control components. Individually, each of these components have established models and well developed tools for modeling, simulation and analysis. However, to analyze the impact of cyber events on the power grid, it is essential to bring all these components together in a coherent simulation environment to study the interdependencies of the cyber and physical system. Additionally, increasing instances of cyber attacks on the electric power grid demands tightly coupled cyber-physical co-simulation for security analysis. Integrated simulation of all the components requires interfacing existing domain-specific modeling and simulation tools for cyber-physical security analysis. This is a challenging task given diversity of domain specific physical and cyber systems simulator/ emulators and interface with hardware in the loop. This paper develops and analyzes number of interfacing techniques for integrated simulation of cyber and power systems for cyber-physical security analysis.**

***Index Terms*—CORE, Cyber-Physical Test Bed, Cyber Security, Interfacing techniques, Microgrid Reconfiguration, Microgrid Resiliency, Real Time Digital Simulator, Smart Grid.**

I. Introduction

To realize the vision of the smart grid, massive amounts of data need to be transferred from the field devices to the control centers. As more monitoring and control algorithms are deployed in the smart grid to produce optimal control at faster rate, the communication infrastructure becomes critical for successful implementation. At the same time, increased number of "smart" devices in the grid leads to increased attack surface for potential cyber-attacks [1]. Given these developments, cyber-physical system based security analysis for the smart grid is very critical.

The smart grid concept has been evolving and gaining prominence over the last decade, especially with more government involvement and investments [2]. The increased possibility of cyber-attacks on the smart grid has been a growing concern, and the recent attacks on the Ukraine power grid [3] has increased these concerns. Simulation of possible cyber-attacks represents a way of studying the impacts of these attacks and ways of mitigating these attacks. However, there are unique challenges [4] for simulating the cyber-attacks on the smart grid. The smart grid is a heterogeneous system and it is very difficult to build a single simulator which can simulate all these systems together while also being granular and detailed. Issues with scalability also arise when trying to simulate larger power systems, and more detailed communication network models [5]. This has led to research efforts to integrate the commercial off-the-shelf (COTS) products to model and simulate the cyber-physical smart grid according to specific requirements.

COTS simulators are already validated for the specific domain, and integrating these involves less effort than building a simulator from scratch. It also offers the flexibility to choose different simulators for various domains as required, and the possibility of interfacing with user developed tools.

However, with using several COTS simulators to model the cyber-physical smart grid also comes with the problem of interfacing these simulators. Interfacing problems include:

1) System modeling across the simulators/emulators/hardware-in-the-loop,
2) Data exchange between the simulators/emulators/hardware-in-the-loop,
3) Timing and synchronization issues between the simulators/emulators/hardware-in-the-loop,
4) Architecture for setting up test cases, and
5) Analyzing results and data correlation.

Another problem with the ad-hoc testbed approach is determining the right set of tools for a specific test case. Especially in case of cyber-physical security testing, there is not a clear way of identifying the test case, the approach and requirements for that test, the products required for the approach, and the interface required between these products. For example, power system simulation tools are focused and classified specific to either transmission or distribution grid, real and non-real time methods, steady state or dynamic simulations. The cyber-physical security analysis does not have such clearly defined sub-domains and techniques yet.

Venkatesh Venkataramanan, Anurag Srivastava, and Adam Hahn are with the Washington State University, Pullman, WA 99163 USA. (E-mail: vvenkata@eecs.wsu.edu)

Pengyuan Wang and Manimaran Govindarasu are with the Iowa State University, Ames, IA 50010 USA.

This work is partially supported by the US Department of Energy under Award Number DE-OE0000780. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

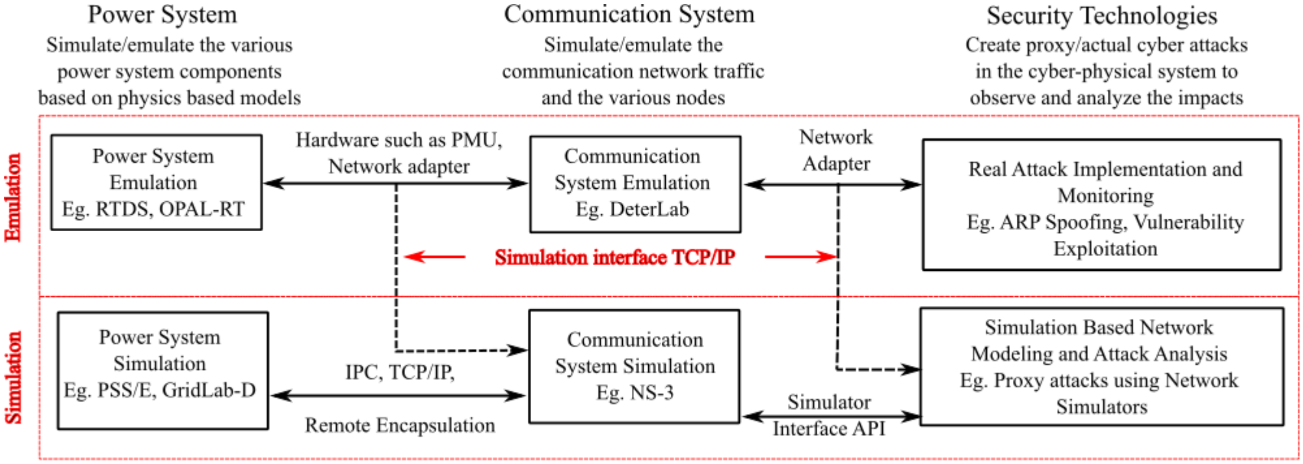

Power System
Simulate/emulate the various power system components based on physics based models
Communication System
Simulate/emulate the communication network traffic and the various nodes
Security Technologies
Create proxy/actual cyber attacks in the cyber-physical system to observe and analyze the impacts
Emulation
Power System Emulation Eg. RTDS, OPAL-RT
Hardware such as PMU, Network adapter
Communication System Emulation Eg. DeterLab
Network Adapter
Real Attack Implementation and Monitoring Eg. ARP Spoofing, Vulnerability Exploitation
Simulation interface TCP/IP
Simulation
Power System Simulation Eg. PSS/E, GridLab-D
IPC, TCP/IP,
Remote Encapsulation
Communication System Simulation Eg. NS-3
Simulator Interface API
Simulation Based Network Modeling and Attack Analysis Eg. Proxy attacks using Network Simulators

Fig. 1: Interface techniques in testbeds for cyber-physical security analysis

This paper aims to explore and analyze the common interfacing techniques in cyber-physical security analysis. Framework for simulating/emulating cyber-attacks and analyzing the results are also discussed. It provides use-cases of current cyber-physical security testbeds, identify the simulators and interfacing techniques used, and then explore how these simulators and interfaces enable the analysis of cyber-security.

II. Modeling and Simulation Tools

The goal of simulating the smart grid is to model the behavior of the real system as close as possible, though this inevitably involves trade-offs and abstractions of certain properties. This simulation typically includes the physical models of the various devices used, the communication protocols used in transferring the data from one place to another, the communication models and architectures, the communication medium, and the various controllers and computing devices which are part of the smart grid. It is important to understand the dependencies between these components, and model them accurately to test different scenarios. Fig. 1 shows an overall block diagram of the various components in a cyber-physical system simulation including power system, communication system, and security technologies. The techniques by which these tools can be integrated can be broadly classified into simulation and emulation technologies. In this section, we will analyze the various tools in these domains, and in the following sections we will study the interfaces between these tools. An overall figure of the different systems considered in this paper is shown in Fig. 1.

A. Power System Simulation Tools

Individually, various tools exist to model the power system. The power system simulators work by solving the "network solution", which is a set of characteristic equations that describe the system. These characteristic equations are based on physics based models of the power system components and are considered to be well researched and validated. In this paper, we focus our interfacing for real-time simulators, which becomes important when evaluating the performance of various defense mechanisms. A real-time simulation involves simulating the test system at the same rate as "wall-clock time" in fixed time steps [6]. Real-time simulation requires the model to be simulated and the results calculated within the time step. If the model is not simulated in the specified time, it leads to a condition called overrun, and the simulation becomes erroneous. This places restrictions on the size of the model that can be simulated in real-time, and depending on the computational power pf the simulation hardware. Real-time simulation offers several advantages, such as controller prototyping, verification and validation of smart grid algorithms, and studying the performance of the test system during dynamic scenarios. Various real-time simulators exist including Real Time Digital Simulator (RTDS) [6], and Opal-RT [7].

B. Communication Simulation Tools

The communication network simulators are based on software constructs that model the network stack. Depending on the network simulator, and its purpose, the detail in modeling varies. Broadly, network simulators can be divided into simulation and emulation tools. In simulation, the communication network is modeled using nodes and connections, but the nodes themselves cannot be accessed or used. On the other hand, emulation models the network such that the communication nodes are actual devices which are capable of emulating the actual hardware devices. The difference in emulation is that the user has access to the processes that can be performed in the node [8]. In general, an emulated network can be connected to other hardware devices that are part of a testbed. All the following communication system software can be considered: NS-3, CORE, DeterLab, Riverbed Modeler (formerly OPNET), OMNET++, Mininet, and GloMoSim. The degree to which these software emulate the communication network devices vary.

C. Security Tools

Security tools provide various functions such as defense mechanisms, network monitoring, visualization, packet analysis, and so on. Typically cyber-attacks are implemented on the communication simulator, and their effects on the physical power system are studied. While certain cyber-attacks can also be implemented on devices to study their impacts, this could potentially damage the hardware being tested. It also potentially exposes the hardware to malicious attackers if the hardware is compromised during testing. Hence, it is safer if cyber-attacks are tested in an isolated environment which ensures that there is no actual damage.

These cyber attacks can be tested in an isolated emulation or simulation environment. In an emulation environment, the emphasis is on using real devices to mimic the real system as closely as possible. The goals of the security tools can be considered two-fold-

1) Analysis, detection, and defense mechanisms of attacks at both network and host level
2) Implement cyber attacks by using either proxies or exploiting vulnerabilities to model real attack

Security tools can be either attack tools, or monitoring tools. Intrusion Detection System (IDS) is a popular tool to deploy in cyber-physical security analysis as it offers the user the ability to determine if a cyber-attack is feasible given presence of a defense mechanism. The IDS system can be configured to flag malicious packets based on existing algorithms, or the user can set up specific conditions for testing. Various IDS exists such as Bro [9], and Snort [10]. Tools in Kali Linux [11], and Ettercap [12] are examples of security tools which can be deployed on emulated networks.

III. Power - Communication Interface

In order to accurately study the effects of cyber-attacks on the grid, it is essential to create a communication model that ties in with the power system model. These interfaces can be either emulation or simulation interfaces.

A. Emulation Interfaces

Emulation interfaces tries to mimic the real system by using actual devices wherever possible. This enables the test to be more accurate, while also providing the opportunity to examine various methods of failures and cyber attacks.

1) Using Power System Hardware: In this case, power system components such as relays, measurement devices such as Phasor Measurement Units (PMUs), and other SCADA devices are used for interfacing. This approach closely follows the actual approach used in the field. For example, the analog signals from the power simulator is fed to the PMU (using power amplifiers if necessary). The PMU has a CT/PT set up to measure the voltage and current values, and the phasor is estimated by the device. In case of emulation, both the PDC (Phasor Data Concentrator) and the control center can be emulated inside the communication system emulator. Emulated system can directly interface with hardware sensors/controllers, and then the communication system or emulated system can generated sensor data and interface with emulated controllers and communication systems.

2) Using Network Card for Emulation: Digital simulators such as RTDS has several analog and digital I/Os, and a network interface card, called GTNET (Giga Transceiver Network Interface Card) to emulate sensors and actuators. The GTNET card is capable of emulating various protocols, such as DNP3, IEC 61850, C37.118, and GOOSE. The values generated by the simulator can be sent to external devices using any of the protocols supported by GTNET. OPAL-RT also provides a similar set up with its network card and the ability to emulate various power system communication protocols using libraries. For offline simulation tools, the process is similar, and these tools typically use the network card of the PC to export the measurements using specified protocols.

B. Simulation Interfaces

For simulation interfaces, hardware based interfaces are typically not used. These interfaces are usually based on exporting the data using the power system simulator's network interface options and into the communication simulator as discussed below.

1) Interfacing to Emulated Network Through Inter-Process Communication: One of the methods for interfacing the power system simulation and the communication system emulation is to establish direct connection between the two processes. This can be done by creating virtual devices which emulate the network cards. TAP and TUN are virtual network device kernels in Linux.

For example, emulation is possible in NS-3 by using its "Tap NetDevice" model. The Tap NetDevice model can be used to connect the NS-3 simulated nodes to any TAP/TUN model, and hence it is also called TapBridge.

There are three modes of operation for the TapBridge. They are the ConfigureLocal, UseLocal and UseBridge modes. The ConfigureLocal mode is configured by NS-3 itself, and the user has the choice of providing a device name so that they can access the created tap. However, the tap is created only when the simulator is run making it harder to connect the systems in real-time. The UseLocal mode suffers from similar problems, and also is restricted to one bridge per MAC address, hence restricting the number of net devices that can be simulated. Hence, the UseBridge mode is chosen, as this allows many net devices on the non-NS-3 side, i.e., the user interface side. This allows to emulate many devices inside the simulator.

2) Interface to Emulated Network through TCP/IP: Another method of interfacing the simulators is to use local TCP/IP connections. This approach can be demonstrated using the CORE network emulator. CORE provides an environment for running real applications and protocols using the virtualization provided by Linux or FreeBSD operating systems [8]. The interface between the power system simulator and the communications network emulator CORE is through TCP/IP. For example with the RTDS, any external application can connect with the RunTime server by specifying the IP address and the port number. A TCP/IP client has been written in Python, which is capable of communicating with the RunTime server of the RTDS. Once the connections is established, the

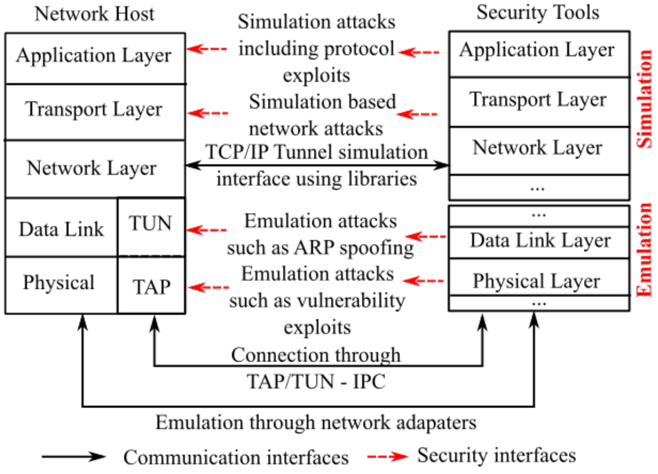

Network Host
Simulation attacks including protocol exploits
Security Tools
Application Layer
Transport Layer
Simulation based network attacks
Network Layer
TCP/IP Tunnel simulation interface using libraries
...
Simulation
Data Link
TUN
Emulation attacks such as ARP spoofing
Data Link Layer
Physical
TAP
Emulation attacks such as vulnerability exploits
Physical Layer
Emulation
Connection through TAP/TUN - IPC
Emulation through network adapaters
Communication interfaces
Security interfaces

Fig. 2: Various interfaces for communication and security tools

client can be used to send commands to the RTDS through this interface.

3) Interface to Remote Testbeds through TCP/IP: Remote encapsulation involves wrapping the measurement packet into another packet and sent over a TCP/IP tunnel, which can be unwrapped to get the original packet. For many Emulab based testbeds, SSH provides a way for the user to connect to the remote testbed. SSH offers the advantage of being secure, and also the flexibility of connecting to the testbed from anywhere. DeterLab is an example for such a testbed.

When connecting DeterLab to the RTDS to run real time simulations, we need to use the Deter Federation Architecture (DFA). DFA provides a way for the researcher to connect local testbed resources to DeterLab. In this case, the user can connect power system components and data to DeterLab and use DeterLab as a platform for implementing cyber-attacks and security mechanisms. Once the network architecture is created (through the GUI/Tcl scripting interface), the cyber-attack can be created using the SEER (Security Experimentation EnviRonment) interface, and the results can be visualized and analyzed through the same interface. When the control signal needs to be sent back to the physical power system layer, the same gateway node is used. The advantage of having a hardware testbed with actual PCs is that it offers the option of implementing cyber-attacks in a more realistic way.

IV. Communication - Security Interface

Depending on the type and requirements of the use case, the security tools can be interfaced in various ways and at various locations of the network stack. A figure showing the various interfaces is shown in Fig. 2.

A. Emulation Environment Attack Implementations and Analysis

(i) Modeling: In an emulation environment, the process closely follows the actual implementation of the attack in the field. Since the networks hosts, traffic, and communication medium can all be considered as attack surfaces, a wide variety of tools can be used in this method. The attack surface in an emulation environment can be considered to be broader, as the setup allows various points of entry for a cyber attack.

The emulation environment offers the opportunity to send and receive TCP/IP packets, and Ethernet frames to implement real cyber attacks. This usually involves using security tools to exploit vulnerabilities present in the hosts and the network to gain elevated privileges, and using this privilege to disrupt the performance of the power system. Usually, the attacker tries to initially monitor the network using tools such as Nmap, and port scanning to scan the network for potential entry points. Once the attacker(s) gains access to the network, they can observe the network to obtain more information, and then perform a cyber attack. DeterLab can be considered as an example of the emulation environment. Since Deter nodes also consider the memory limitations of the emulated nodes, it is possible to study the effects of attacks such as TCP SYN flood, which cannot be studied in a simulation with only a Linux kernel. DeterLab provides excellent resources which detail the types of studies possible, and the methods here [13], [14].

(ii) Monitoring: For network monitoring, various tools such as Wireshark can be used to capture the packets. The attacker can thus gain knowledge about the communication and the power system network for potential attacks. The emulation environment also offers the opportunity to deploy and test various defense mechanisms such as Bro, Snort, which offer IDS functionalities. Bro also uses data from each individual nodes to monitor the performance of individual nodes of the network.

B. Simulation Environment Attack Implementations and Analysis

(i) Modeling: In the simulation environment, the emphasis is usually on analyzing the impact on the power system. In the simulation environment, real devices are rarely used. In case of the power system layer, the measurements are extracted directly from the simulator and sent to the communication layer using various technologies. In the communication layer, the communication network hosts, and its associated links are simulated. The network hosts do not usually model the complete network stack such as the physical layer, or the device kernel, or the memory management for these hosts. Hence, it is not feasible to perform attacks such as buffer overflow which needs the model of the memory. Hence unlike the emulation environment, device level vulnerabilities, and associated cyber attacks cannot be performed in a simulation environment. However, network level attacks can be performed in the simulation environment. The simulators offer the capability to vary the latency, drop packets, drop links, disable certain hosts, and more.

(ii) Monitoring: Most simulator provide options to monitor the performance of the network. In case of NS-3 for example, NetAnim can be used to visualize the network and monitor its performance. In addition, tools such as Snort which operate at a network level can also be deployed for defense mechanisms. Attack implementations using simulation platforms are usually

TABLE I: Lessons Learned from Interfacing Studies

Security Experiment	Requirements	Use-case	Possible Implementation
Man in the middle attack	Emulation of communication network	Microgrid reconfiguration	Emulation with network card based communication and TCP/IP sockets with third party libraries can be used [15]
Latency effects	Simulation of communication network	Transmission system algorithms (such as voltage stability, transient stability, RAS) testing	If transmission system is very large (greater than computational limit of real time simulators), offline simulation with network simulation tools such as NS-3 can be used [16]
Real time cyber attack implementation	Emulation and use of real devices	Transmission system closed loop testing	Transmission system needs to be smaller, and vulnerabilities need to be exploited to study the effects on the power system [17]
Implementing defense mechanisms	Emulation testbed with use of security tools	Evaluating defense mechanisms	It is essential to use emulated networks with granular models of network devices, and hardware based testbeds are best suited [18]

through a proxy interface. In this case, the attacker is assumed to have gain unauthorized access, and the cyber attack is performed using scripts written by the user. This might involve physical impacts such as opening and closing of breakers, manipulating various control signals and such. The effect of these actions can be observed in the power systems simulator.

In the next sub-section, two commonly performed cyber attacks are analyzed for both simulation and emulation environments.

V. Cyber Attack Implementations and Analysis

Depending on the type of communication simulator used, various cyber-attacks can be executed with differing levels of accuracy. Testbeds at Washington State University [15], [17] show two different testbeds that are used for different purposes. Both testbeds use the RTDS, but the way the communication emulation is achieved is different. In [15], the communication emulation is achieved through TCP/IP, as described in section III. In [17], emulation is achieved through IPC, using the TAP/TUN models. Similarly for Iowa State, the choice of the tools used for simulation depends on the use case to be evaluated. In [18], since defense mechanisms need to be evaluated, hardware devices have been used, while in [16], a simulation interface has been used. Table I shows various cyber-physical security use cases demonstrated at Iowa State and Washington State University.

A. *Denial-of-service (DOS) Attack*

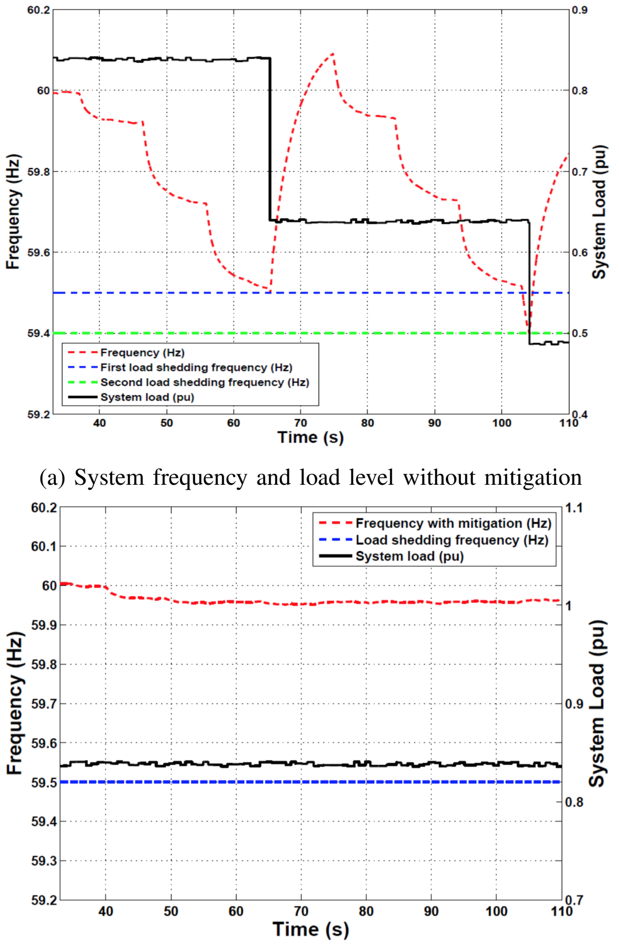
The DOS attack can cause a severe impact on the smart grid. DOS attack affects the availability requirement. The theory behind the DOS attack is simple - the objective is to disable a device in the network. However, implementing the DOS attack can be tricky based on the simulator used. Consider the TCP SYN flood attack. The attack involves the attacker sending multiple SYN packets to the compromised node until it becomes non-responsive. However, if only the network stack is simulated, it is not possible to simulate the node running out of computation space. In these cases, the DOS attack has to be implemented by using the bandwidth restriction on the link. The attacker has to send enough traffic that the node is unable to send any information through that particular channel. In cases where the actual implementation of the attack itself is not important, rather the point is to study the power system impact, it may be sufficient to simply disable the node in question and then monitor the power system impact.

B. *Man-in-the-middle (MiTM) Attack*

MiTM is also a popular attack to study the performance of the smart grid. The MiTM attack is typically used to the study the response of the smart grid to integrity based attacks. The theory behind the attack is that the attacker gains access to the compromised node, and then proceeds to modify the system in a malicious manner. This may involve changing a measurement from the node, changing the control signal sent by the node, manipulating the data stored in a node, or triggering false control actions. In order to simulate the process of gaining access to the node, the user has to go through multiple steps such as modeling the ARP traffic, modeling vulnerabilities which allows elevation of privilege, and modeling intelligent attack agents capable of manipulating the node without detection. If the end goal is to study the power system impact and not the defense mechanisms, the process is simpler.

C. *Testbed Implementation*

Various penetration tools and mitigation methodologies have been used, tested or developed based on PowerCyber testbed at Iowa State. For instance, with LOIC (Low Orbit Ion Cannon) one can effectively launch DoS attack on Siemens relays, Nmap is often utilized for host and service scanning, OpenVAS and Nessus are tools for vulnerability scanning, and Scapy in python is frequently chosen to carry out ARP spoofing and MiTM attack. Simple mitigation such as firewall configuration, port blocking and vulnerability patching can be easily done on the testbed, and more complex defensive strategies with IDS tools, moving target defense and advanced model based mitigation strategy have also been developed. In paper [18], a resilient model-based AGC algorithm evaluation is carried out on ISU PowerCyber testbed under data integrity attack. IEEE 9-bus model is divided into 3 balancing authorities with AGC, and MiTM attack is launched on the AGC function in area 1 to modify the frequency and tie-line flow measurements. The system frequency responses and load levels without and with mitigation under this attack are shown in Fig. 3a and Fig. 3b,



(a) System frequency and load level without mitigation

(b) System frequency and load level with mitigation

Fig. 3: Simulation Results from ISU Testbed [18]

reproduced from [18]. It can be observed that the system frequency will be continuously driven down by manipulating AGC without any mitigation and the under frequency load shedding will be triggered when the frequency goes below certain thresholds. On the other hand, the performance of AGC with proposed mitigation strategy is comparatively more resilient. More details about the experiment implementation can be found in [18].

In our previous work [15], we have detailed another testbed in which proxy attacks are demonstrated. The WSU microgrid testbed uses the RTDS for power system simulation, CORE for communication system emulation and uses proxy based attacks to determine the impact on the microgrid. A reconfiguration algorithm based on resiliency is used to study the microgrid performance. When the simulation is started, data from the power system simulator is obtained, and is routed through this network model to the control center, which runs the control algorithm for reconfiguration. This algorithm takes in the data, analyzes it, and sends the new switch status as necessary. This is again communicated back to the power system simulator thorough the same interface. For more details and results, please refer to [15].

VI. Conclusions

This paper provides the various interfacing techniques for integrated simulation of power system, communication system, and security tools for cyber-physical security analysis. Methods of network emulation, and connecting to a real time simulator have been discussed. Techniques to interface cyber-power testbed with cyber-attacks modeling, and implementing security tools have also been discussed. Specific testbed architectures focused on cyber-physical security analysis have been described as examples, and various case studies have been presented to identify lessons learned in interfacing domain specific simulators/emulators with hardware-in-the-loop.

References

[1] P. Eder-Neuhauser, T. Zseby, and J. Fabini, “Resilience and security: A qualitative survey of urban smart grid architectures,” *IEEE Access*, 2016.

[2] “Smart grid,” http://energy.gov/oe/services/technology-development/smart-grid, accessed: 2016-09-30.

[3] “Inside the Cunning, Unprecedented Hack of Ukraines Power Grid,” https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/, accessed: 2016-09-30.

[4] S. C. M. et. al, “Interfacing power system and ict simulators: Challenges, state-of-the-art, and case studies,” *Smart Grid, IEEE Trans. on*, 2016.

[5] J. Sztipanovits, G. Hemingway, A. Bose, and A. Srivastava, “Model-based integration technology for next generation electric grid simulations,” in *IEEE Power and Energy Society General Meeting*, July 2012.

[6] R. Kuffel, J. Giesbrecht, T. Maguire, R. Wierckx, and P. McLaren, “RTDS-a fully digital power system simulator operating in real time,” in *WESCANEX Communications, Power, and Computing. Conference Proceedings., IEEE*, May 1995.

[7] D. Bian, M. Kuzlu, M. Pipattanasomporn, S. Rahman, and Y. Wu, “Real-time co-simulation platform using OPAL-RT and OPNET for analyzing smart grid performance,” in *IEEE Power Energy Society General Meeting*, July 2015.

[8] “CORE Manual [Online],” http://downloads.pf.itd.nrl.navy.mil/docs /core/coremanual.pdf, Tech. Rep.

[9] I. Bro, “Homepage: http://www. bro-ids. org,” 2008.

[10] R. Alder, A. Baker, E. Carter, J. Esler, J. Foster, M. Jonkman, C. Keefer, R. Marty, and E. Seagren, “Snort: Ids and ips toolkit,” *Syngress Publishing*, 2007.

[11] K. Linux, “Kali linux— penetration testing and ethical hacking linux distribution,” 2015.

[12] D. Norton, “An ettercap primer,” *SANS Institute InfoSec Reading Room*, vol. 5, 2004.

[13] “Deter Federation,” http://seer.deterlab.net/trac, accessed: 2016-09-30.

[14] “Deter SEER Wiki,” http://fedd.deterlab.net/, accessed: 2016-09-30.

[15] V. Venkataramanan, A. Srivastava, and A. Hahn, “Real-time co-simulation testbed for microgrid cyber-physical analysis,” in *Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, April 2016.

[16] V. K. Singh, A. Ozen, and M. Govindarasu, “Stealthy cyber attacks and impact analysis on wide-area protection of smart grid,” in *2016 North American Power Symposium (NAPS)*, Sept 2016, pp. 1–6.

[17] C. B. Vellaithurai, S. S. Biswas, and A. K. Srivastava, “Development and application of a real-time test bed for cyber-physical system,” *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2015.

[18] A. Ashok, S. Sridhar, A. D. McKinnon, P. Wang, and M. Govindarasu, “Testbed-based performance evaluation of attack resilient control for agc,” in *Resilience Week (RWS)*, Aug 2016, pp. 125–129.