Cyber Protection of Grid-Connected Devices Through Embedded Online Security

Chris Farnell
Department of Electrical Engineering
University of Arkansas
Fayetteville, Arkansas
cfarnell@uark.edu

Estefano Soria
Department of Electrical Engineering
University of Arkansas
Fayetteville, Arkansas
esoria@uark.edu

Justin Jackson
Department of Electrical Engineering
University of Arkansas
Fayetteville, Arkansas
jej029@uark.edu

H. Alan Mantooth
Department of Electrical Engineering
University of Arkansas
Fayetteville, Arkansas
mantooth@uark.edu

***Abstract*— Cybersecurity research regarding the electric power grid has primarily been focused on protecting the communication layer of grid-connected devices against cyber-attack threats. Although many developed methods have greatly reduced the effects of a cyber-attack on the vulnerabilities of grid-connected devices, discovering new vulnerabilities is inevitable and a constant threat. As a result, the overall reliability and security of network communications with regard to grid-connected devices is a concern. This paper proposes a method that further secures a system by focusing on the control and hardware layer of grid-connected devices. The device's controller firmware will be validated and authenticated using integrated device emulation resources prior to being activated to control the grid-connected device. This verification process is performed while the controller is online and actively controlling power flows related to the device. Therefore, an attack to the system through a malicious firmware patch would be detected by the online security and rejected while safely maintaining continuous and stable control of the device. This method integrates the concepts of firmware hot-patching, digital twins, and active monitoring into an overall cybersecurity protection system.**

Keywords—cybersecurity, hot-patching, digital twin, real-time emulation, embedded systems, grid-connected

I. Introduction

Cybersecurity has become a subject of interest worldwide, especially regarding the potential impact that cyber-attacks have on the electric power grid and other critical infrastructure. As distributed generation from renewable energy sources becomes more prevalent, more advanced controls and communications are required to coordinate optimal power flows. While this increased coordination is both beneficial and necessary to maintain grid reliability and efficiency, engineers and security professionals must also address the increased attack surface that results. When attacks do occur, the attacker typically targets the communications layer as a first step. These types of cyber-attacks include Phishing, Man-in-the-Middle (MitM), Denial of service (DoS), Structured Query Language (SQL) injection, Domain Name System (DNS) Tunneling, and more. If the attacker successfully gains control of the communications layer, the attacker can then pivot to gain control of the remaining layers including the controller and hardware device layers. A graphical representation of these layers is shown in Fig.1., below. One example of this can occur when uploading a malicious firmware to the controller that forces the grid-connected device to power down or cause it to operate in non-optimal modes. This in turn, could cause a portion of the grid to shut down, similar to what occured in the Ukraine attack in 2015 and 2016 [1]. Considering these concerns and challenges, this paper proposes a method of further securing grid-connected devices through the Supervisory, Control, and Hardware layers.

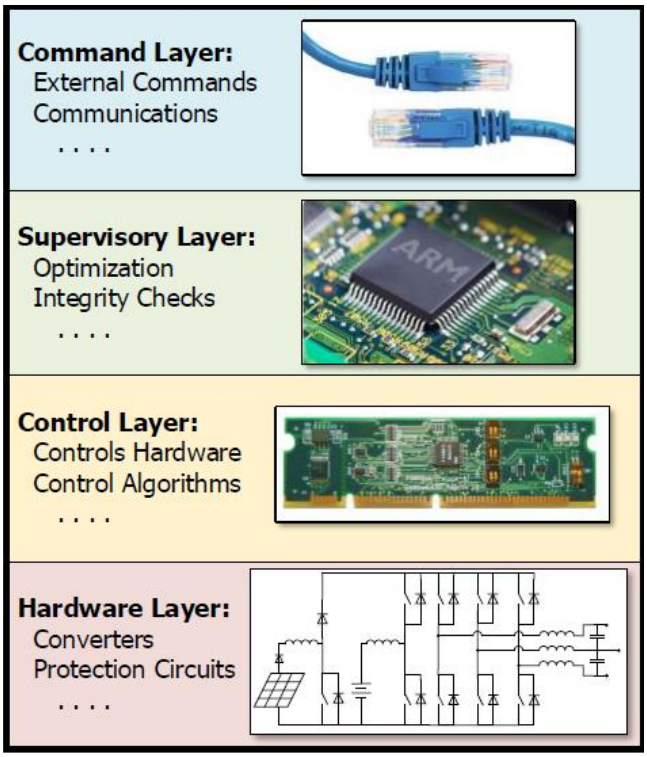

Fig. 1. Cyber-physical layer representation.

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number DE-EE0009026.

II. Embedded Security Architecture

The proposed method in this paper includes an embedded controller architecture consisting of an FPGA (Field Programmable Gate Array) and two Digital Signal Processor (DSP) controllers and various auxiliary components as shown in Fig. 2., below. This architecture was developed in order to allow researchers the ability to easily integrate their cybersecurity detection and mitigation solutions into a common platform used for power electronics control and testing. This common platform also allows for code reuse and rapid prototyping between collaborating researchers such that they will not have to redesign advanced power electronics control algorithms for each cybersecurity module to be tested.

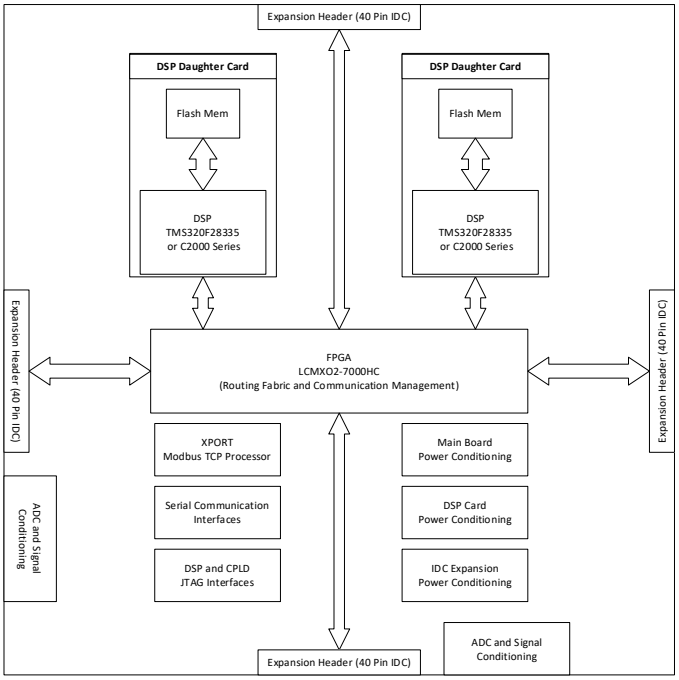

Fig. 2. Block Diagram of UCB architecture showing major components.

This architecture is realized in the Printed Circuit Board (PCB) colloquially called the Unified Controller Board (UCB) and shown in Fig. 3. The physical PCB includes all the required power, communication, and signal conditioning circuitry required for rapid prototyping while also including expansion headers for custom power electronic control interfaces. Peripheral boards are designed to mate with these expansion headers and include: fiber optic transceivers, thermocouple inputs, voltage/current sensor inputs, and other interchangeable boards. This modular approach allows for flexibility and design reuse across multiple different projects and applications. Both the digital twin and hot-patching examples shown in this paper have been successfully implemented using the described architecture and coexist with the subroutines used to control an example inverter to demonstrate the efficacy of the proposed solution.

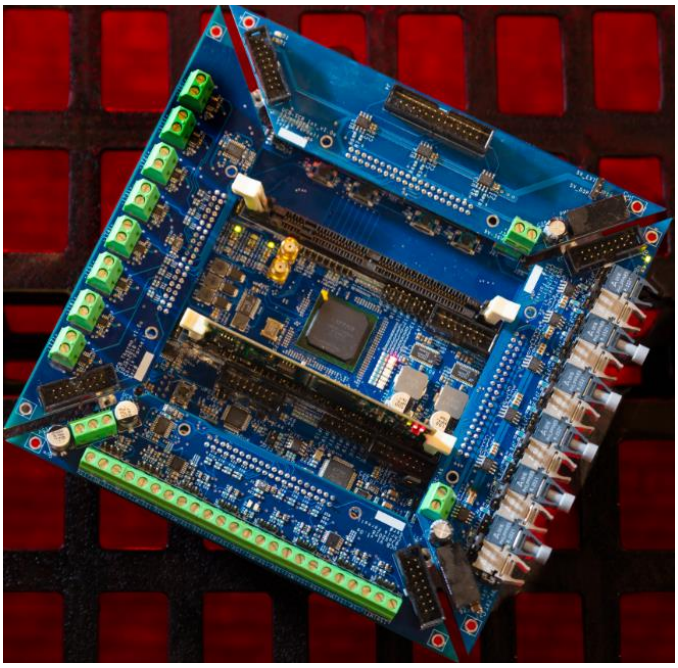

Fig. 3. UCB PCBs with auxillary daughter boards installed.

III. Digital Twin

The concept of a Digital Twin has become a topic of interest for a broad range of applications. As a consequence, the term Digital Twin has multiple definitions depending on each application. Reference [2] describes how the concept of a Digital Twin consists of multiple levels, where each level is an advancement in model accuracy and features. Any implementation or method that can be defined as one of these levels is considered a Digital Twin [2]. In the topic of cybersecurity for grid-connected devices, Digital Twins are Real-Time simulations of these devices that are used to study their performance during various cyber-attack scenarios. Through these simulations, the data collected during the attack scenarios is used to develop a mitigation strategy for the system. This process of development is typically performed offline and/or outside the physical system due to resource constraints. For this application an emulator was designed and embedded into the local FPGA to allow for online verification of firmware at the system level. Fig. 4. shows a block diagram of the proposed digital twin implementation contained within the FPGA subsystem. Within the FPGA, a hardware emulator is instantiated which mimics the physical hardware of the grid-connected device, shown in Fig. 5. This emulator accepts Pulse Width Modulated (PWM) inputs and calculates the associated output voltages. The calculated outputs are then compared to the expected outputs to ensure they are within an acceptable range prior to the new firmware becoming live on the system. Communication interfaces, firmware loading, dual-port memory, and multiplexer modules are also instantiated in the FPGA located on the UCB.

Fig. 4. illustrates the two DSPs in the overall logical system which will be distinguished as DSP1 and DSP2. Initially, DSP2 is referred to as the standby DSP, and DSP1 will be referred as the active DSP controlling the grid-connected device. During the firmware update, the firmware is patched on DSP2, where the DSP will reset similar to the common patching process of any electronic device. Once patched, DSP2 will go through the online validation process that is embedded in the controller board. Simultaneously, DSP1 will be continuously controlling the grid-connected device. The online validation process is programmed in the FPGA, which receives the measurement and control signals from DSP1 and DSP2. This validation feature tests a composition of possible vulnerabilities in the firmware, and emulates the grid-connected device to test the behavior of the firmware in DSP2.

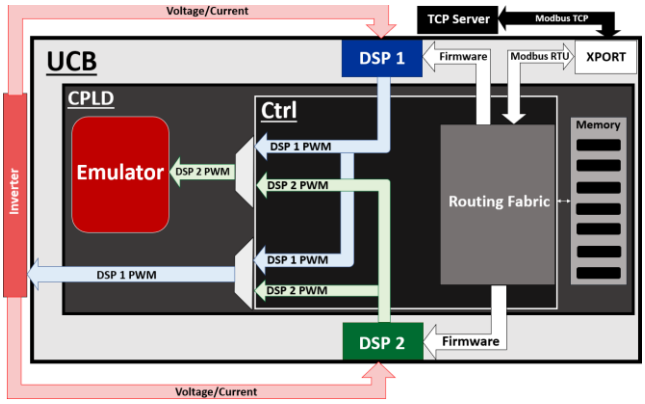

Fig. 4. Controller architecture with active DSP1 and standby DSP2.

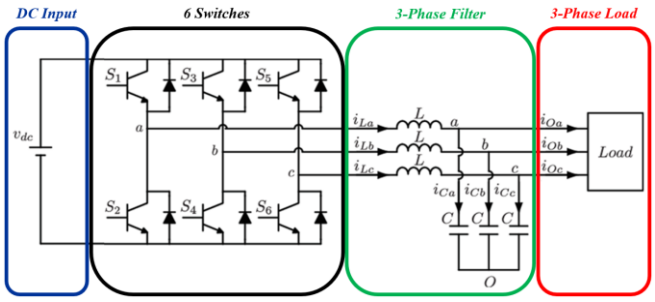

Fig. 5. Typical representation of 2-level inverter hardware.

The firmware validation subroutines implemented in this project include: shoot-through protection, voltage verification, and frequency verification. Additional firmware verifications can easily be integrated in the future due to the modularity of the overall design. As a first step with regard to the firmware verification, the PWM input signals are checked during the operation of the standby DSP to ensure there are no shoot-through conditions or dead-time issues present. A shoot-though condition occurs when the upper and lower switches of the device are active at the same time which would cause a large current that is typically damaging to the physical components of the inverter. An illustration of a shoot-through event is shown in Fig.6. If a shoot-through condition is detected than a software flag is raised and the firmware in deemed as invalid in the standby DSP.

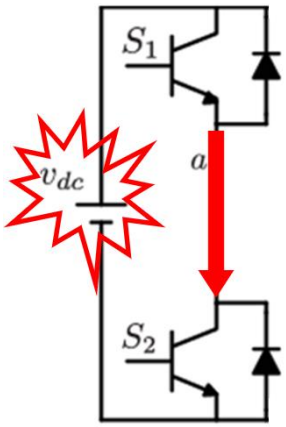

Fig. 6. Representation of shoot-through event.

An additional check is then completed using the aforementioned hardware emulator to calculate the voltage and frequency of the output signals based on the input PWM signals. The PWM input signals are captured analyzed using the emulator module. The period and duty cycles for the PWM signals are then used along with the systems input voltage to calculate the instantaneous output voltage of the system. These values are stored in a dual-port RAM module, also located in the FPGA, for later analysis. A block diagram of the online emulation process is shown in Fig. 7., below.

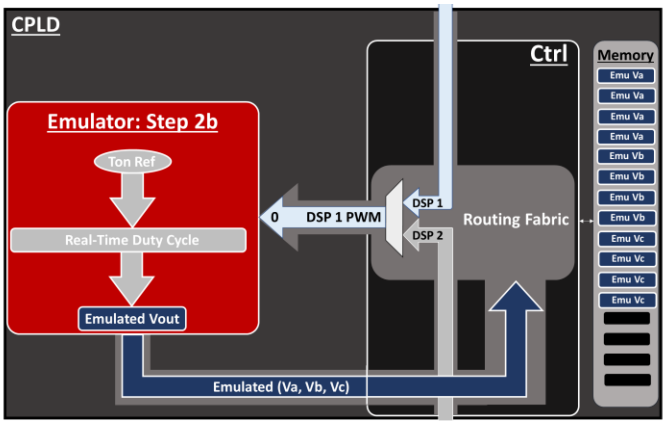

Fig. 7. Block diagram of online emulation process.

Both the Digital Twin and Hot-Patching modules are controlled externally with a custom LabVIEW interface, shown in Fig. 8. This interface allows for the selection of firmware patches, initiation of Digital Twin emulation, and the reporting of the status and results for the system overall. Figs. 9 and 10 show the results of the emulation for both a nominal firmware patch, Fig. 9., as well as a comprimised firmware patch which resulted in an unbalanced phase error, Fig. 10.

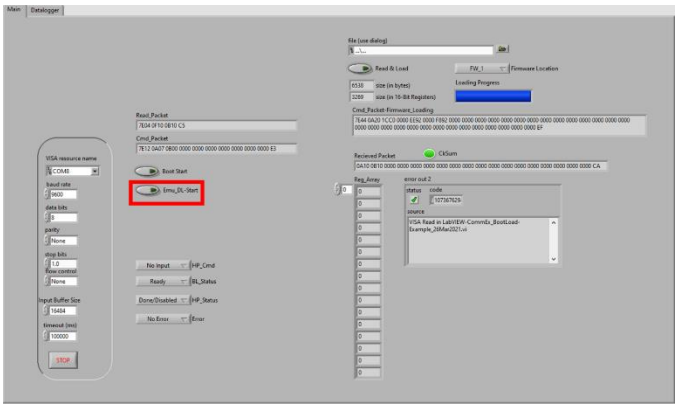

Fig. 8. LabVIEW interface to start emulation operation.

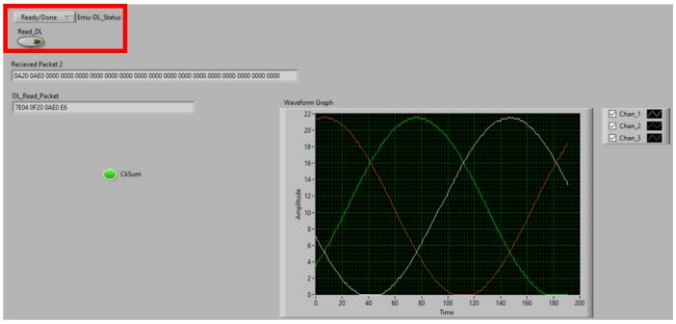

Fig. 9. LabVIEW interface displaying collected emulation data.

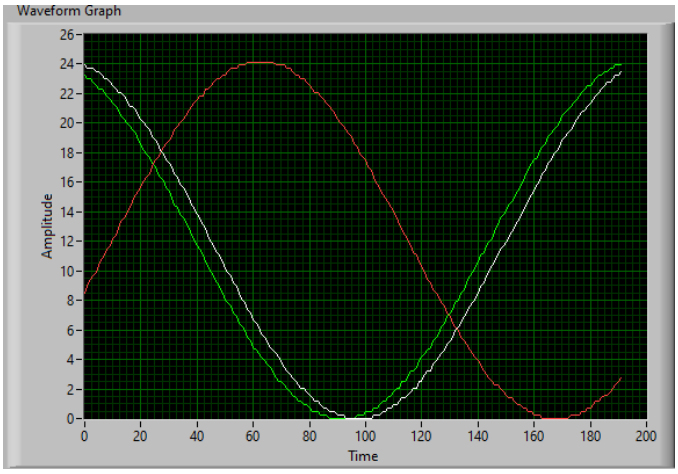

Fig. 10. Emulation result with phase unbalance error.

IV. Hot-Patching

Another concept to consider is Hot-Patching, which is the ability of patching firmware to a device without causing downtime or disruption to the system. Some Hot-Patching attempts include [3-6], where the implementation methods contain delays, restrictions to code in the firmware, and other assumptions that limit their applicability. One previous approach taken is [6] which succeeded under fixed conditions. To succeed, the firmware needed to have a specific structure, which limits the feasibility of this method. Although current methods of Hot-Patching contain limitations, the concept would be a benefit with regard to patching vulnerabilities of grid connected devices, if correctly implemented. As described in [7], there are efforts focusing on reducing the delay of patching vulnerabilities by creating specific patching schedules. Once the patch is developed, the device would traditionally require downtime to upload the firmware. When finalizing the patch a reset operation is also typically required. Due to the desire to keep downtimes for grid-connected devices at a minimum, updates may not be completed for a relatively long time. During this interim time, the grid connected devices contain known vulnerabilities, which means that an attacker could exploit these vulnerabilities and attack the system prior to the patch being implemented.

After the emulation and verification, from the Digital Twin process described earlier in this paper, if the firmware is deemed valid then the Hot-Patching process will begin, otherwise, the backup firmware process will initialize. The FPGA is utilized as a "routing fabric" for the Hot-Patch and can switch the DSP output control signals on the order of nanoseconds. As a result of the Hot-Patch, DSP2 will be activated to control the grid-connected device, and DSP1 will be on standby mode for backup, shown in Fig. 4. Conversely, during the backup process in the case of an invalid firmware, DSP2 remains on standby and DSP1 remains active. For this process, a firmware backup stored in the FPGA memory is loaded and patched back onto DSP2. In case DSP1 fails for any reason, DSP2 containing the same firmware will be assigned to control the grid-connected device, thus incorporating redundancy and failsafe functionality to this method. Fig. 11 shows the LabVIEW control interface during the Hot-Patching operation while Fig. 12. shows the VHDL simulation of the Hot-Patching module used to verify DSP functionality.

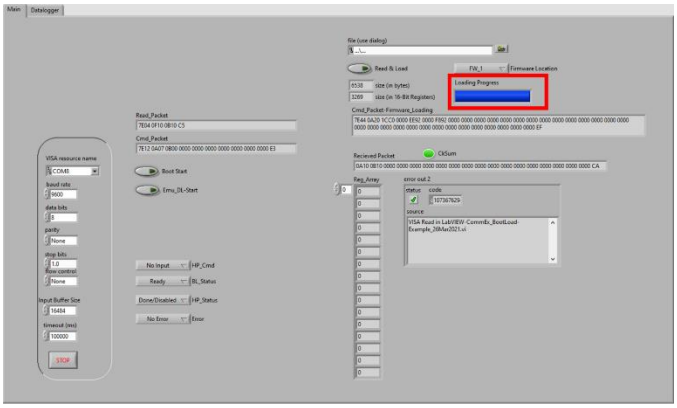

Fig. 11. LabVIEW interface showing Hot-Patching operation in progress.

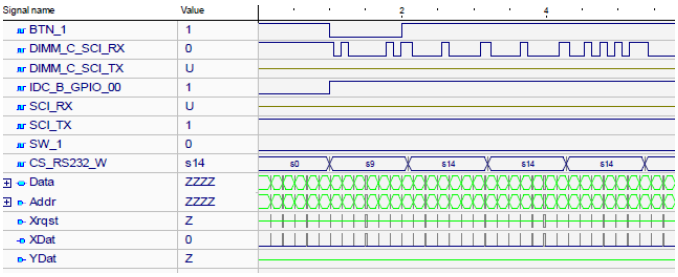

Fig. 12. FPGA-Based DSP firmware loading.

Because the two DSPs are running in parallel and are receiving all the same measurement data from the physical hardare their operations can be synchronized during Hot-Patching. This synchronization is important to ensure a seamless transition from one control DSP to the other while not causing any interuption in service. Fig. 13. shows the result of a DSP transition after firmware patching and verification without any synchronization. As can be seen in the waveform the transition is apparent and in a real-world senario could cause an outage and/or damage to equipment. Fig. 14 shows the transiton event with the DSPs properly synchronized which would result in no disruption in service or damage to equipment.

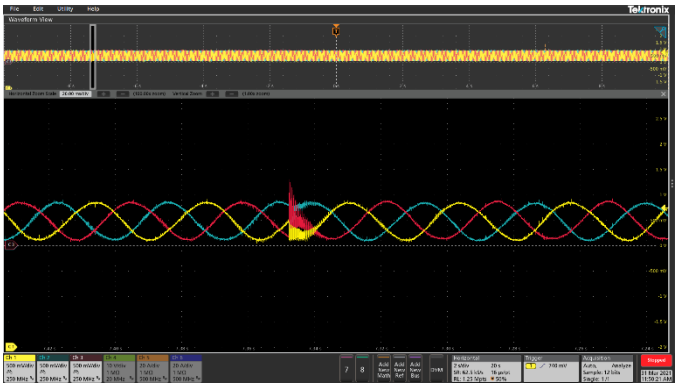

Fig. 13. Unsynchronized 3-phase inverter output waveform during hot-patch.

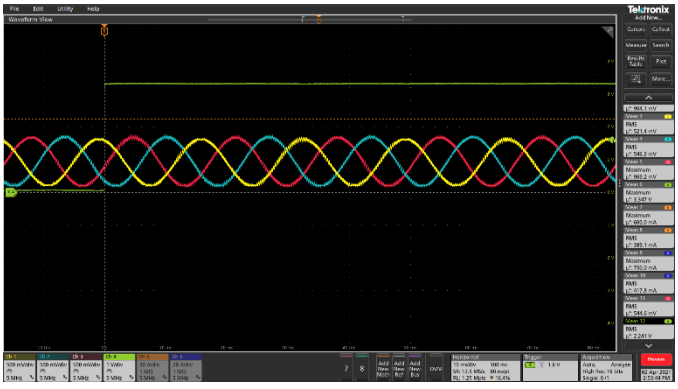

Fig. 14. Synchronized 3-phase inverter output waveform during hot-patch.

V. Conclusions and Future Work

Cyber securing grid-connected devices have become a requirement, and many methods to mitigate vulnerabilities are continuously being developed. This proposal demonstrates a different approach where the focus of security is embedded in the control and hardware layers instead of only focusing on the network communication layer. An overview of the architecture and functionality of the proposed method and device have been described. Hardware has been developed to test the efficacy of the proposed solution and results have been presented. The results presented demonstrate the capability and applicability of this method and where it can be leveraged for further cybersecurity research related to grid-connected devices.

The proposed approach consists of embedding both an online Digital Twin and Hot-Patching methodology into the controls of a grid-connected device. This approach will allow for firmware to be patched and validated from the control layer before it is activated to control the overall system which adds an additional layer of protection. In the future, additional verification and mitigation methods will be developed and integrated into to the proposed cybersecurity framework and architecture.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number DE-EE0009026. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Development and testing for this project were conducted at the National Center for Reliable Electric Power Transmission (NCREPT), the University of Arkansas' High-Power Test Facility.

References

[1] B. Huang, M. Majidi and R. Baldick, "Case Study of Power System Cyber Attack Using Cascading Outage Analysis Model," 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, OR, 2018, pp. 1-5, doi: 10.1109/PESGM.2018.8585921.

[2] V. V. Makarov, Y. B. Frolov, I. S. Parshina and M. V. Ushakova, "The Design Concept of Digital Twin," 2019 Twelfth International Conference "Management of large-scale system development" (MLSD), Moscow, Russia, 2019, pp. 1-4, doi: 10.1109/MLSD.2019.8911091..

[3] Z. Xu, "Source Code and Binary Level Vulnerability Detection and Hot Patching," 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, VIC, Australia, 2020, pp. 1397-1399.

[4] H. Jeong, J. Baik and K. Kang, "Functional level hot-patching platform for executable and linkable format binaries," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017, pp. 489-494, doi: 10.1109/SMC.2017.8122653..

[5] F. Pozo, G. Rodriguez-Navas and H. Hansson, "Work-in-Progress: A Hot-Patching Protocol for Repairing Time-Triggered Network

Schedules," 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Porto, Portugal, 2018, pp. 89-92, doi: 10.1109/RTAS.2018.00015..

[6] A. Ramaswamy, S. Bratus, S. W. Smith and M. E. Locasto, "Katana: A Hot Patching Framework for ELF Executables," 2010 International Conference on Availability, Reliability and Security, Krakow, Poland, 2010, pp. 507-512, doi: 10.1109/ARES.2010.112..

[7] F. Zhang and Q. Li, "Dynamic Risk-Aware Patch Scheduling," 2020 IEEE Conference on Communications and Network Security (CNS), Avignon, France, 2020, pp. 1-9, doi: 10.1109/CNS48642.2020.9162225