INL/CON-19-54698-Revision-0

Software Defined Cyber-Physical Testbed for Analysis of Automated Cyber Responses for Power System Security

November 2019

Bjorn C Vaagensmith, Craig G Rieger, Justin J Welch, Jacob J Ulrich

Changing the World's Energy Future

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/CON-19-54698-Revision-0

Software Defined Cyber-Physical Testbed for Analysis of Automated Cyber Responses for Power System Security

Bjorn C Vaagensmith, Craig G Rieger, Justin J Welch, Jacob J Ulrich

November 2019

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Software Defined Cyber-Physical Testbed for Analysis of Automated Cyber Responses for Power System Security

Jacob J. Ulrich
Iowa State University
Idaho National Laboratory
Idaho Falls, USA
jacob.ulrich@inl.gov

Bjorn C. Vaagensmith, PhD
Critical Infrastructure
Idaho National Laboratory
Idaho Falls, USA
bjron.vaagensmith@inl.gov

Craig G. Rieger, PhD
Critical Infrastructure
Idaho National Laboratory
Idaho Falls, USA
craig.rieger@inl.gov

Justin J. Welch
Systems Analysis
Idaho National Laboratory
Idaho Falls, USA
justin.welch@inl.gov

***Abstract*—As the power grid becomes more interconnected the attack surface increases and determining the causes of anomalies becomes more complex. Automated responses are a mechanism which can provide resilience in a power system by responding to anomalies. An automated response system can make intelligent decisions when paired with an automated health assessment system which includes a human in the loop for making critical decisions. Effective responses can be determined by developing a matrix which considers the likely impacts on resilience if a response is taken. A testbed assists to analyze these responses and determine their effects on system resilience.**

Keywords— Software Defined Networking, Automated Health Assessment, Automated Response, Microgrid, Smart Grid, Industrial Control Systems, Cyber-Physical Systems, Cyber Security

I. Introduction

Disastrous consequences can transpire when the critical function of a cyber-physical systems (CPS) like the smart grid is interrupted. Civilian, business, and government functions are reliant on the services provided by such CPSs. This makes resilience of the utmost importance in the design and implementation of these systems.

Much of the research conducted on resilience if focused on developing CPSs which can cope with low probability, high-impact events such as natural disasters. This is an important area of resilience research. However, frequent, high probability events must also be considered. One compelling example is the pervasiveness of cyber exploits. The millions of cyber-attacks attempted daily are growing in number. Fully automated exploits are easily accessible and zero-day exploit can be purchased by state actors. A CPS needs to have only a single vulnerability exploited to cease critical function.

Automated health assessment and automated response are critical areas of research for providing resilience in cyber-physical systems (CPSs). An automated health assessment systems (AHAS) monitors the operating status of a CPS. An AHAS detects anomalies in a CPS, determines the cause of the anomaly and provides resilience metrics to inform operators of real-time system functionality. An automated response systems (ARS) attempts to limit degradation and restore functionality to a CPS. ARSs accomplish this by communicating with AHASs and making changes to CPS configurations. Ideally, these changes will restore maximum functionality to a CPS by correcting the actions which caused the anomaly.

The drastic differences between CPS environments make the development of AHASs and ARSs a challenging problem. It is difficult to know if an AHASs designed for one CPS will be effective on another CPS. CPSs consist of multiple interconnected systems and are controlled by algorithms which operate at different timescales. A change in the control software or the physical hardware present in a system could affect the operation of an AHASs. Testbeds can be utilized to assist in developing AHASs which are effective for a CPS.

This paper demonstrates use cases and design of an easily reproducible, software defined testbed, for the analysis of AHASs and ARSs to provide resilience in power systems. The paper is structured as follows. Section II overviews the high-level function of a software defined network (SDN) and reviews previous work. Section III describes the SDN testbed and the microgrid model used for the analysis of this paper. Section IV introduces the concept of a response matrix and tradeoff space for the intelligent selection of automated responses based on a confidence score. Section V presents two automated response use cases and analyzes the results. Section VI details future work and concludes the paper.

II. Background and Related Works

A. Software Defined Networking

SDN is a paradigm which provides programmatic control over the forwarding plane of a network. Traditional networks are managed in a decentralized manner and the forwarding of packets is controlled by logic on the forwarding devices. Conversely, An SDN allows for centralized management of the forwarding devices, such as switches, by locating the forwarding logic on a centralized controller.

This work was supported by the Office of Naval Research under grant number N000141812395.

The SDN concept can be easily understood from Fig. 1. Typically, an SDN controller provides northbound and southbound interfaces. The northbound interface allows applications in the application layer to interact intelligently with the SDN controller in the control layer. The southbound interface provides the SDN controller with a method to interact with the SDN switches which encompass the infrastructure layer. The SDN controller can control the behavior of switches in the infrastructure layer by sending additional instructions over the southbound interface. The switches in the infrastructure layer can also send traffic statistics to the SDN controller.

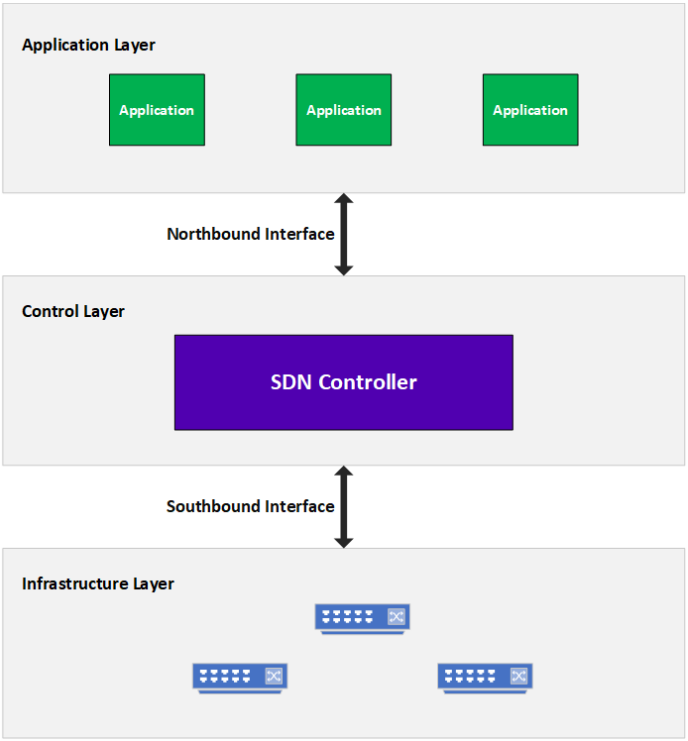

Figure 1. SDN Architecture

Several protocols exist to handle the communication between the controller and the switches. These protocols define the instructions used by the switches for forwarding traffic. OpenFlow is an opensource SDN control protocol used for this paper. OpenFlow switches can view ethernet, Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) headers. An SDN switch is programmed with flows from the SDN controller which inform the switch how to forward traffic. As a packet enters an SDN switch, it matches the header information of the packet with the highest ordered flow rule and completes a combination of matching actions based on its flows. Once a packet has reached a flow rule with a writing action, the packet is forwarded or dropped accordingly.

Certain flows will forward packets to the SDN controller which may modify the application layer data of a packet. The controller may additionally pass packets to an application for further processing. If commanded by an application, the controller can update flows and change the forwarding behavior of the SDN switches. In this manner, applications can be developed to autonomously control the behavior of the network. This ability allows for automated responses which effect the operational network of a power system.

This work was supported by the Office of Naval Research under grant number N000141812395.

B. *Cyber-Physical Security Testbeds*

Anomaly detection is a component of an AHAS. [1] used a physical model for anomaly detection in response to stealthy attacks. Automated responses included sending alarm signals to the physical processes and having a human investigate the anomaly. [2] conducted automated threat response using intelligent agents. The agents were capable of populating and querying a networked database to convey system health and detect anomalies. [3] compared and proposed resilience metrics in conjunction with probabilistic risk assessment which can be used for assessing power system health.

Several studies have looked specifically at automated responses for CPSs. A dynamic security policy based on organizational role-based access control was proposed in [4] to enable automated responses in a cyber system. Responses were taken by changing security contexts. [5] proposed response policies using dynamic access control. This work considered the impact of automated responses and created a networked architecture for implementing the responses. In [6] a network model was implemented to evaluate the effects of intrusion response mechanisms. An evaluation function was provided to classify responses by the amount of negative impact they have on the system.

A few testbeds have been developed for CPS resilience and automated response testing. [7] implemented green city, a cyber-physical testbed consisting of low cost, physical controllers and load generation devices. This method recreates the power and cyber requirements of a DC microgrid. Iowa State University has a hardware in-the-loop cyber-physical testbed presented in [8]. This testbed consists of a real time digital simulator integrated with commercial relays and intelligent electronic devices. These testbeds are both capable of evaluating automated response mechanisms for CPSs.

III. TESTBED ARCHITECTURE

Fig. 2 exhibits the physical construction of the SDN testbed. It contains a mixture of proprietary and opensource products running on commercial off-the-shelf equipment. Apart from the OpalRT server and Schweitzer SEL switches, these components can be easily sourced through multiple vendors. All servers and workstations are standard x86 architecture machines.

A novel approach used by this testbed is the inclusion of a standalone mininet server. Mininet is an opensource network emulator capable of faithfully duplicating a network with thousands of nodes. Mininet can import switch definitions and supports the popular OpenFlow capable, open vSwitch by default. The server contains 26 physical ethernet ports which can be bridged to virtual SDN switches in the mininet

topology. This allows for extremely flexible testing and network design.

This approach provides the maximum accuracy without the high cost of expensive hardware. Real devices, such as programmable logic controllers and relays, can be quickly integrated into the virtual network by connecting them to a physical port on the mininet server. Quality of service rules can be implemented in software to produce different testing conditions such as delay and jitter.

As mentioned, the hardware components of this testbed include two proprietary Schweitzer 2704 SDN switches and an OpalRT server. The SEL switches are not necessary to reproduce the results from this paper. All experiments can be replicated in mininet. The purpose of the SEL switches are for testing the interaction of SDN switches from multiple vendors using a single SDN controller.

The virtualization host runs vSphere on an ESXi hypervisor. The human machine interface (HMI), SDN controller, and snort run on the virtualization host. IFIX from GE is used as an HMI for supervisory control of the microgrid emulated in OpalRT. The opensource OpenDaylight controller was used to control the mininet emulated SDN. Snort was used as an analog AHAS. Snort monitors the HMI and microgrid OpalRT server through a bridged interface provided by mininet. When snort detects an anomaly, it sends a simple DBUS message to the OpenDaylight controller to initiate an automated response.

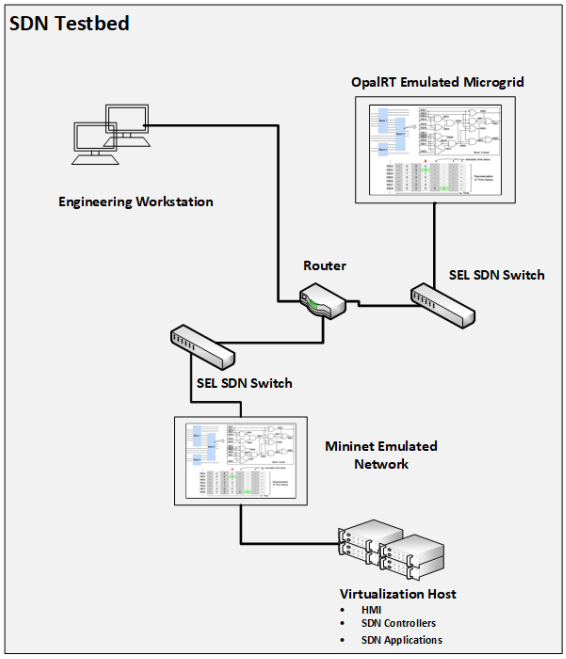

Figure 2. Testbed Architecture

The Open Platform Communications (OPC) protocol is used for supervisory control actions between the HMI and the microgrid model. It should be noted; this testbed is also capable of using DNP3 and MODBUS in place of OPC by modifying modules on both OPC servers. Fig. 3 shows the necessary components to establish OPC communication. The OpalRT server hosts an OPC UA server. This OPC UA server talks to Kepware on the engineering workstation. The engineering workstation works as a router-on-a-stick. It accepts traffic from the OpalRT OPC UA server and forwards the traffic to the OPC UA server hosted by the HMI. This step is necessary due to a bug in the OpalRT OPC UA server. Future updates to OPC UA will eliminate the need for this intermediary step.

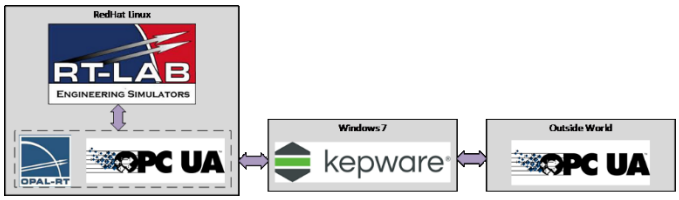

Figure 3. OpalRT OPC Communication

A. *Power System*

Fig. 4 shows the Simulink model running on the OpalRT server. This is a 13-bus model modified from the IEEE standard 13-bus model provided in Simulink. A control point was added to the relay at bus 692. The control point reports the opened or closed status of the relay to the HMI. The HMI can open and close the breaker by sending the appropriate OPC commands. This model can be further edited to allow far greater supervisory and control actions.

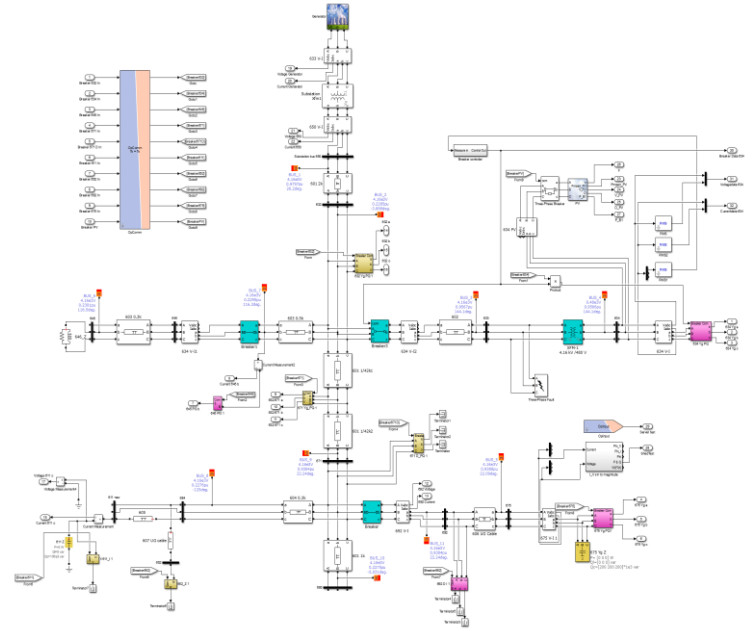

Figure 4. Microgrid Subsystem

B. *Experimental Network Topology*

Fig. 5 displays the network topology used for all experiments in this paper. The OpalRT microgrid is connected directly to the mininet server through a virtual router. The mininet topology consist of three SDN switches running an open vSwitch implementation. The SDN switches are fully meshed and are routed through a physical router to the HMI

and the SDN controller. The SDN controller and HMI are on different subnets to allow for out-of-band management of the SDN switches. The switches contain flows which look at all incoming frames and determine if they are IPv4 packets. If the headers of the frames match as IPv4 frames, they are forwarded without modification. Otherwise, they are dropped.

Snort is an open source intrusion detection system which detects anomalies through signature and rules-based methods. Snort sees all network traffic in mininet through a virtual span port on the mininet server. It is configured to detect the attacks used in our experiments using a custom signature and alert the controller to take an automated response.

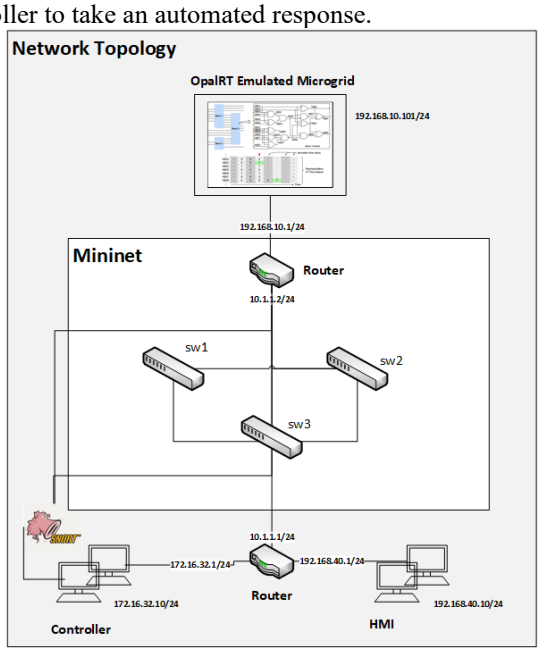

Figure 5. Experimental Topology

IV. Response Matrix

As previously mentioned, power systems operate in several timescales. For example, the protocols used for supervisory control have different sampling rates than protocols used for automatic generation control. Additionally, the complexity of these systems means small changes can lead to unintended consequences. This is more likely if the change is made in a uniformed manner which does not take the design of the system into account. Due to these reasons, automated responses are more effective if they are based on a response matrix.

A. Tradeoff Space

A response matrix is a collection of cyber responses which can be taken in the event of an anomaly. A response matrix can be developed in conjunction with a system or after careful evaluation of an existing system. The purpose of the matrix is to provide a set of automated actions which system operators may implement in an ARS before an anomaly occurs.

Automated responses must consider the tradeoff space of the system. The tradeoff space illustrates a classic issue of increased security measures reducing functionality and vise versa. Increasing security measures can lead to greater system restrictions and often lower system functionality. This effect could be greatly amplified in power systems if poor automated responses are chosen which break functionality of the system.

Physical devices rely on networked communication with a remote controller. If an anomaly occurs which effects network communications, then one response may be to block communications by automatically adding a firewall rule. This rule may block the attack, but it could also stop the communication of the controller and physical device. This could lead to a physical effect which is dangerous to the physical system locally or globally. The design of the system must be understood to prevent risky actions that break functionality. Automated responses can provide increased resilience, but functionality should not sacrificed.

Fig. 6 shows the health assessment cycle and highlights considerations for choosing automated responses based on their possible system effects. First, an anomaly is detected by the AHAS. Next, an automated cyber response is taken. After this, a cyber effect occurs such as a change in network traffic. Finally, this cyber effect may result in an effect to the physical system. The AHAS continues to monitor the system and determines if the anomaly still exits. If it does, the system may initiate an additional cyber response until the anomaly is absent.

A response matrix analyzes responses based on the health assessment cycle to determine the system effects associated with each automated response. Using analytical and modeling techniques, an AHAS can develop a confidence score regarding the probability a response may break functionality. This would be based off analysis of the distributed power system and cyber measurement collected by the AHAS. A high confidence score indicates a response which is unlikely to affect physical system function. AHAS can be designed to provide additional factors to a confidence score. This would take into consideration the likely cause of an anomaly given all compiled, distributed information available to the AHAS.

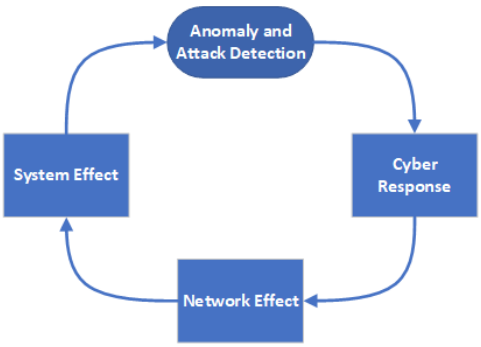

Figure 6. Health Assessment Cycle

B. Response Selection

When an anomaly is detected, there may be several automated responses which will effectively correct the effects of the anomaly and return a CPS to a healthy state. In this case, a tiered approach can be considered when developing an automated response system. Once the AHAS determines an automated response is necessary, it will choose the response which provides the greatest confidence to create minimum disturbance to the functionality of the system. This response

may or may not have the highest probability of correcting the anomaly. If the response is not successful in halting the anomaly, the next response with the second highest confidence can be taken. If an anomaly cannot be halted by the system then a human-in-the-loop should be alerted with information from the AHAS to assist them with correcting the anomaly.

This is an effective approach against anomalies such as those cause by a cyber-attack. Fig. 7 demonstrates the algorithm used by the AHS and ARS when choosing automated responses. The AHAS interprets cyber data and determines if an anomaly has been detected. If an anomaly is detected, then a confidence score is computed. The confidence score is sent to the ARS which then enumerates the possible responses and ranks them by confidence score. The action with the highest confidence score is taken. The AHAS continues to monitor traffic to determine the health of the system.

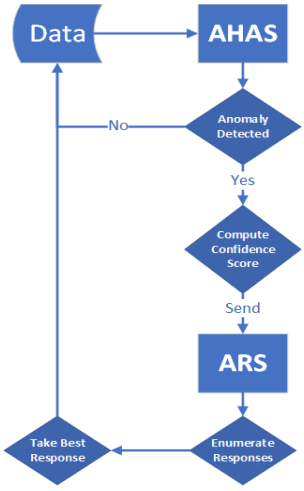

Figure 7. Health Assessment Algorithm

V. Use Case Analysis

The following use cases demonstrate how automated responses are analyzed in the testbed. Fig. 8 shows the position of the attacker in both use cases. The attacker is in the mininet topology and can see all traffic between the HMI and the microgrid. Fig 9. Shows the buses controlled and measured by the HMI. The breaker for bus 692 was directly attacked in both use cases. While bus 692 is the direct target, physical measurements from bus 672 are analyzed to show the system effects caused by the automated responses and the actions of the attacker.

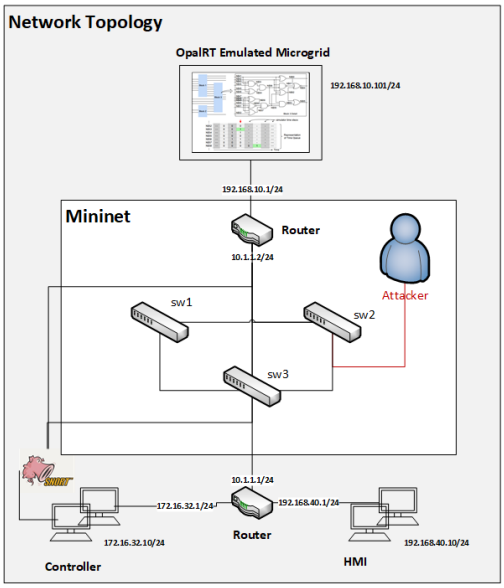

Figure 8. Attacker Location

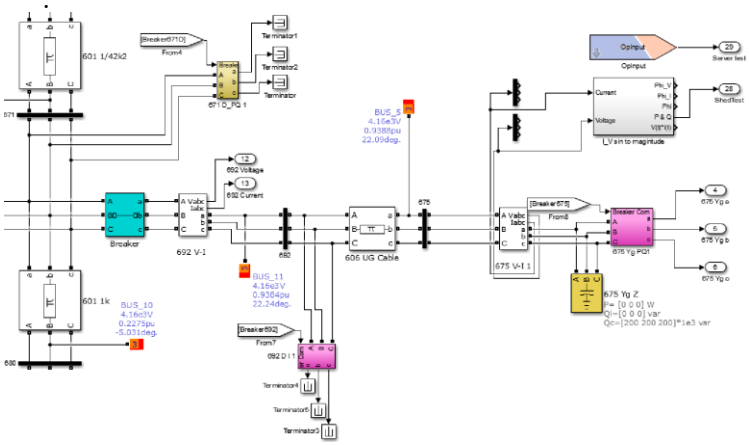

Figure 9. Bus 571 and 692

Fig. 10 shows the normal voltage profile for the three phases of bus 692. Fig 11. Shows the normal voltage profile for the three phases of bus 671. These voltages do not change because the load profile for the microgrid repeats every 24 hours. The average round trip time for OPC traffic as measure from the HMI is 80 milliseconds.

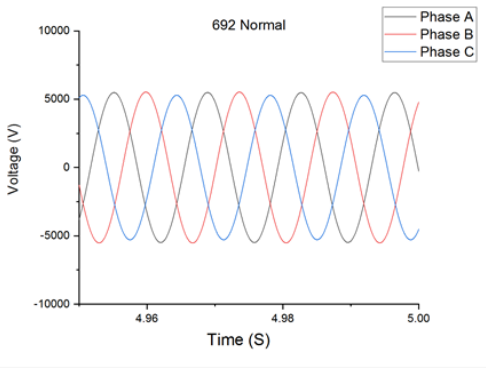

Figure 10. Bus 692 Normal Voltage

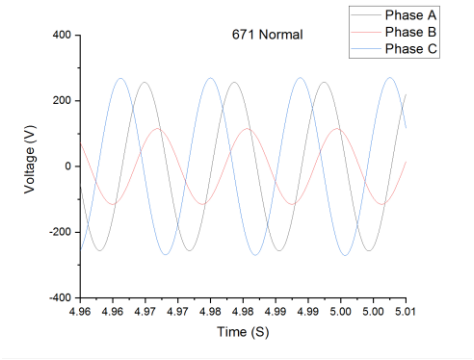

Figure 11. Bus 671 Normal Voltage

A. *Use Case One*

In the first use case, the attacker successfully injects false OPC commands by conducting ARP spoofing. This allows the attacker to capture OPC commands and modify packets. In this manner the attacker defeats standard replay protections provided by OPC. In this use case, the attacker sends a single packet to open the breaker on bus 692. Snort is configured to detect this packet and initiate an automated response.

The response in this scenario is to quickly generate an OPC command to close the breaker. The HMI is configured to keep the breaker closed by measuring the breaker status. If the HMI receives a measurement showing the breaker is open, it will send a close response.

The following occurs as the attack is initiated:

- Snort detects the attack and prepares a DBUS message to the SDN controller.
- The SDN controller takes no action as it is still waiting on the DBUS message.
- Snort withholds this message to allow the HMI an opportunity to close the breaker.
- Snort captures the OPC command to close the breaker and contacts the SDN controller telling it to take no action.
- The HMI sends a close command to the relay.

No automated response was taken except for the normal action of the HMI. Fig. 12 and Fig. 13 show the effects of the attack. The voltage on bus 692 drops to 0V during the duration of the attack. This is expected because bus 692 is downstream from a generator and supplies no power. Fig. 13 shows bus 671 also drops to 0V. This is unexpected because bus 671 is upstream from bus 692 and still receives a load from the generator. This effect is due to the attack on bus 692 and is an example of an unintended physical reaction in the system.

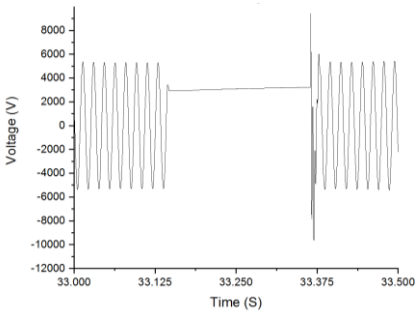

Figure 12. Bus 692 Command Injection

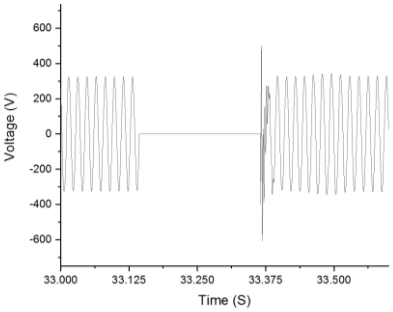

Figure 13. Bus 671 Command Injection

B. *Use Case Two*

In the second use case, the attacker uses the same method to inject OPC commands over the SDN network. In addition, the attacker conducts a denial of service attack on the HMI subnet. This prevents the HMI from sending OPC close commands to address the anomaly. Instead of keeping the breaker closed, the attacker alternates between sending open and close commands to the relay at regular intervals. This causes flapping causing the breaker on bus 692 to constantly open and close.

This behavior can be seen in Fig. 14. The voltage on bus 692 is alternating between 0V and 5000V. Fig. 15 shows bus 671 is affected by the behaviors as well. The voltage level on bus 671 alternates between 0V and 120V at nearly the same time the breaker opens and closes.

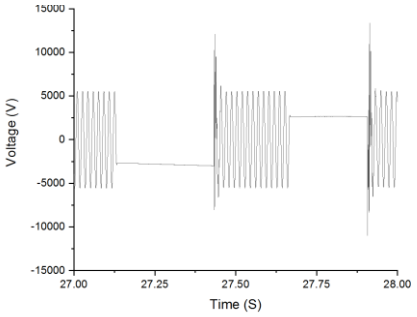

Figure 14. Bus 692 Flap Attack

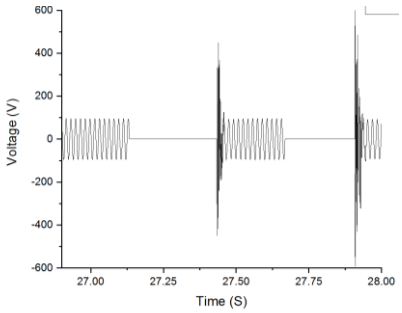

Figure 15. Bus 671 Flap Attack

Snort detects this attack using a custom signature. It instructs the SDN controller to push flows to the SDN switches to halt the denial of service attack. Since there are three links, the controller sends flows to disable one link at a time. This forces traffic to take a new route. For example, looking at Fig. 8, if SW1 and SW3 stop accepting traffic from SW2, the attacker will not be able to communicate on the network. HMI traffic will flow between SW3 and SW1.

SDN flows work by matching ethernet headers. The flows pushed by the SDN controller match ethernet headers and create a match action to drop all ethernet frames on a port. This effectively blocks traffic on a switchport. The controller pushes the flows in phases and takes down one link at a time. If snort still detects the anomaly, the controller reestablishes the link it took down and removes a different link. If each link has been taken down once, the controller begins removing links in pairs until the anomaly is resolved.

The results from Fig.16 and Fig.17 show the successful recovery of the system. Fig. 16 shows the HMI reestablished connectivity and closed the breaker because the voltage returns to normal levels. Fig. 17 shows the anomalies in bus 671 were resolved as a consequence. Fig. 14 and Fig.15 show the attack beginning at 27.125 seconds into the experiment. Fig. 16 and Fig.17 show the anomaly was corrected at 34.825 seconds into the experiment. This means the automated response successfully resolved the anomaly within 7.7 seconds.

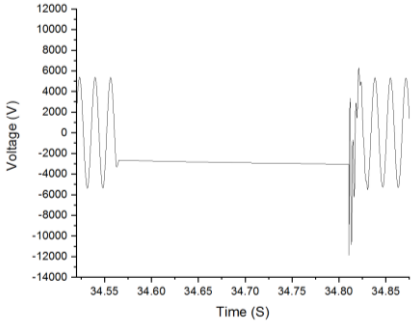

Figure 16. Bus 692 Recovery

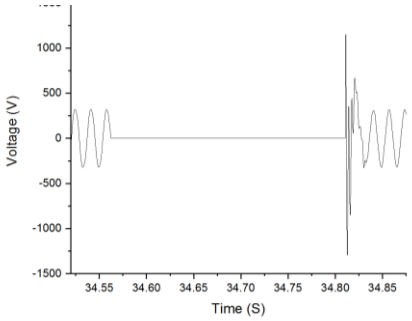

Figure 17. Bus 671 Recovery

VI. CONCLUSION

This paper presented a cyber-physical testbed relying on SDN for implementing and analyzing automated responses in a microgrid environment. Software defined switches were shown to be an effective platform for mitigating cyber-attacks. Additionally, the idea of a tradeoff space and a response matrix were introduced.

Future work includes developing a robust framework for automated Reponses. The production of a robust response matrix relative to the SDN testbed and the 13-bus model is required. These responses will be built into an ARS which includes a SIEM interface for human in the loop interaction. An AHAS will be developed utilizing unsupervised machine learning techniques which capture physical and cyber measurements through deep packet inspection.

ACKNOWLEDGEMENT

We thank our colleague Professor Masood Parvania, Phd from the University of Utah who provided insight and expertise that assisted this research.

REFERENCES

[1] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*, 2011, pp. 355-366: ACM.

[2] A. Quan *et al.*, "Automated threat response using intelligent agents (atria)," in *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*, 2001, vol. 6, pp. 2721-2730: IEEE.

[3] B. Vaagensmith *et al.*, "An Integrated Approach to Improving Power Grid Reliability: Merging of Probabilistic Risk Assessment with Resilience Metrics," in *2018 Resilience Week (RWS)*, 2018, pp. 139-146.

[4] H. Debar, Y. Thomas, F. Cuppens, and N. Cuppens-Boulahia, "Enabling automated threat response through the use of a dynamic security policy," *Journal in Computer Virology,* vol. 3, no. 3, pp. 195-210, 2007.

[5] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, and S. Dubus, "Risk-aware framework for activating and deactivating policy-based response," in *2010 Fourth International Conference on Network and System Security*, 2010, pp. 207-215: IEEE.

[6] T. Toth and C. Kruegel, "Evaluating the impact of automated intrusion response mechanisms," in *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, 2002, pp. 301-310: IEEE.

[7] A. C. Becerra, W. Zeng, M.-Y. Chow, and J. J. Rodríguez-Andina, "Green city: A low-cost testbed for distributed control algorithms in smart grid,"

in *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 001948-001953: IEEE.

[8] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid," *IEEE Transactions on Smart Grid,* vol. 4, no. 2, pp. 847-855, 2013.