EVALUATING SOFTWARE DEFINED NETWORKING SOLUTIONS TO REDUCE THE DIGITAL ATTACK SURFACE OF A PHYSICAL PROTECTION SYSTEM

S. L. CLEMENTS
Pacific Northwest National Laboratory (PNNL)
Richland, Washington, USA
Email: samuel.clements@pnnl.gov

C. SMITH
Pacific Northwest National Laboratory (PNNL)
Richland, Washington, USA
Email: cameron.smith@pnnl.gov

B. NICKLESS
Pacific Northwest National Laboratory (PNNL)
Richland, Washington, USA
Email: bill.nickless@pnnl.gov

C. H. NICKERSON
Idaho National Laboratory (INL)
Idaho Falls, Idaho, USA
Email: charles.nickerson@pnnl.gov

Abstract

Most nuclear security systems used today were not designed for today’s threat environment. Systems that were intended to be stand alone are now interconnected. Devices that have a single purpose are built on multi-purpose platforms and communication protocols that, while effective, have no ability to authenticate authorized versus unauthorized commands. These attributes provide an attacker significant ability to affect the system, pivot throughout the interconnected networks, and remain undetected if he/she is able to compromise a single node.

Software defined networking (SDN) is used by information technology (IT) cloud service providers to quickly provision or remove servers or other systems to meet changing demand. The same concept has recently been applied to operational technology (OT) systems to enable very fast failover on critical systems that have stringent and deterministic (<5ms) transmit/receive times. By carefully engineering the communication flows through a network using pre-planned routes and specific pathways, it is possible to achieve deterministic and extremely reliable message delivery even when components fail. This engineering approach to network design has added security benefits including securing the networking control plane, eliminating network scanning and mapping, inhibiting ARP spoofing and host masquerading, eliminating unauthorized network pivoting and enabling greater situational awareness on the network.

SDN in OT environments is new but early testing in electrical power and other critical infrastructure has shown it to be a very powerful tool for building reliable networks and reducing the digital attack surface of the network [1]. The authors tested a software defined network switch on a simple physical protection system with components commonly found in nuclear security systems and found improved mitigations to denial of service attacks, lateral movement and network reconnaissance. The paper details the tests and their results.

1. INTRODUCTION

Nuclear security systems are shifting from analogue systems to digital, from isolated independent systems to networked integrated systems (IAEA, 2017). This shift provides significant cost reductions, increased flexibility, greater efficiency, and more interoperability. While creating benefits of common access across multiple sites, remote monitoring, integrated intrusion detection and response, the move to digital systems also creates new risks to be managed. Cyber-attacks are increasingly targeting operational technology [2] [3]. Physical protection systems consist of many operational technology components or devices [4]. A common integrated physical protection system will include field devices like cameras, badge readers, biometric scanners, magnetic door locks, etc. connected to controllers that are connected to network switches that will connect to headend management systems through network cables, switches and or junction boxes to finally be presented to a security officer in the alarm station. The networks that connect all the devices are often “flat networks” meaning that every node on a given network can communicate uninhibited with every other device on the network. Flat networks are efficient and easy to manage but have intrinsic vulnerabilities. If an attacker is able to get on any node within a flat network, he/she can communicate and potentially attack or pivot to other nodes within the network. For this reason, Defence in Depth computer security measures are recommended to increase costs to attackers [5]. For example, in Figure 1 it would be expected that all cameras can communicate with the network video recorder (NVR). But it is not necessary for camera A to communicate with Camera B. In this network, there is not a mechanism prohibiting camera A from communicating with camera B or camera C. Using software defined networks (SDN), it is possible to create rules within the switch that only allow designed communications to occur and deny all other communications.

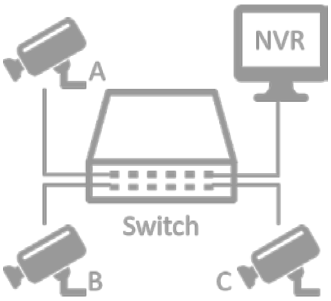

Figure 1: Simple Security Cameras Diagram

The rules and functionality have similarities to firewall rules that separate network segments but do so at the device and or physical port level giving a much more granular control of what network communications are allowed on a network.

Software defined networks can be used to meet multiple requirements [6]. A new use case for SDN is to create known, pre-determined network paths through a given environment enabling deterministic timing for network traffic and has proven valuable in safety protection schemes within electrical power substations. The ability to specify the exact path a given network packet will traverse, based on rules that consider information at OSI Layers 1 through 4 (International Telecommunications Union Recommendation X.200), enables improved control of network flows not found in traditional networked environments. This study evaluates some of these benefits in a simplified but representative physical protection system environment.

2. RELATED WORK

Applying software defined networking to operational technology systems as a part of a computer security system is not unique. Di Lallo et al, has looked at SDN as a way to improve intrusion detection systems [7], Molina and Jacob conducted a survey evaluating SDN and operational technology systems as a way to mitigate bandwidth, latency, redundancy and safety considerations [8]. Ndonda and Sadre show that SDN can improve network performance reducing delay peaks and improving anti-eavesdropping techniques [9]. Multiple efforts have looked at SDN to support incident response in operational technology system [10] [11]. The study conducted by the authors builds upon the work done by Sainz et al [12] but focuses on a specific type of control system, physical protection systems.

3. EXPERIMENTAL ENVIRONMENT DESCRIPTION

The experimental environment consisted of components typically found in networked physical protection systems. The network was intentionally simplified for the study but sufficiently complex for an initial study of the SDN solution. A full-scale test would include multiple network switches, additional end nodes sensors, professional hardware and a variety of software packages not included in this test as well as additional tests to evaluate initial findings. A full-scale test would further validate the effectiveness of this solution to mitigate computer-based attacks. The baseline environment consisted of an IP enabled network camera, a hand geometry access control reader, a magnetic stripe card reader, an access control PLC, and a virtual machine server which hosted five virtual operating systems:

- a “Surveillance Station” used for monitoring the network camera feed;
- a “Workstation” for general purpose use;
- a “SIEM/IDS” for detecting threats;
- a “Hand Geometry Mgmt.” system for configuring the hand geometry unit;
- and an “Access Control Server” which managed the Access Control PLC and stores the access log database.
- The details of the physical protection system architecture are included below in Table 1:

Name	Operation System	Memory	NIC:IP Address	Listening Ports	Additional Details
Virtual Machine Server	Ubuntu 19.04 64-bit	55GB	1:Unused 2:Bridged 3:Bridged 4:Bridged 5:Bridged 6:Bridged 7:Bridged	N/A	VMWare 15.5.0 2 Intel Xeon E5-2650v2 2.60GHz CPUs 1TB HDD
Surveillance Station	VM:Win 7	2 GB	1:192.168.123.50	N/A	None
Workstation	VM:WinXPPro	512 MB	1:192.168.123.2	N/A	None
Access Control Server	VM:Win7	2 GB	1:192.168.123.201	N/A	None
SIEM/IDS	VM:Linux (OSSIM)	4 GB	1:192.168.123.34 2:No/IP	TCP (80, 514, 3128)	None
Hand Geometry Mgmt	VM:Linux (Debian)	8 GB	1:192.168.123.200	N/A	None
Door Camera	Proprietary	Unknown	1:192.168.123.100	TCP (80, 443, 544, 49152)	None
Hand Geo Unit	Proprietary		1:192.168.123.1	TCP (3001)	USB AC PLC
Access Control PLC	Raspberry PI	1 GB	1:192.168.123.3	TCP (22, 80)	USB Hand Geo Unit Hard wire to Card Reader
Card Reader	N/A	N/A	N/A	N/A	Hard wire to AC PLC
Baseline Switch	Netgear ProSAFE JGS524PE		24 Port 12:PoE		
SDN Switch	SEL 2740S		20 Port 4: 1Gbs 16: 100Mbs 1: Mgmt		
SEL-5056 SDN Controller	VM: Win Server 2016	8 GB	1:192.168.123.102		

Table 1: Test Environment Details

3.1. Baseline Architecture

The baseline architecture diagram in Figure 2 depicts the connectivity between devices. The virtual network interface cards (NICs) on the virtual machines on the left were each bridged to a physical NIC on the host machine. From the host, they were connected to a managed network switch without any configurations applied. For testing purposes, the Hand Geometry Management Server is the compromised system. It is from that location in the network that all simulated attacks are conducted. The manner in which it is compromised is outside the scope of this study, but it is assumed that the attackers have root privileges and the ability to install new software and configure or reconfigure existing capabilities on the system.

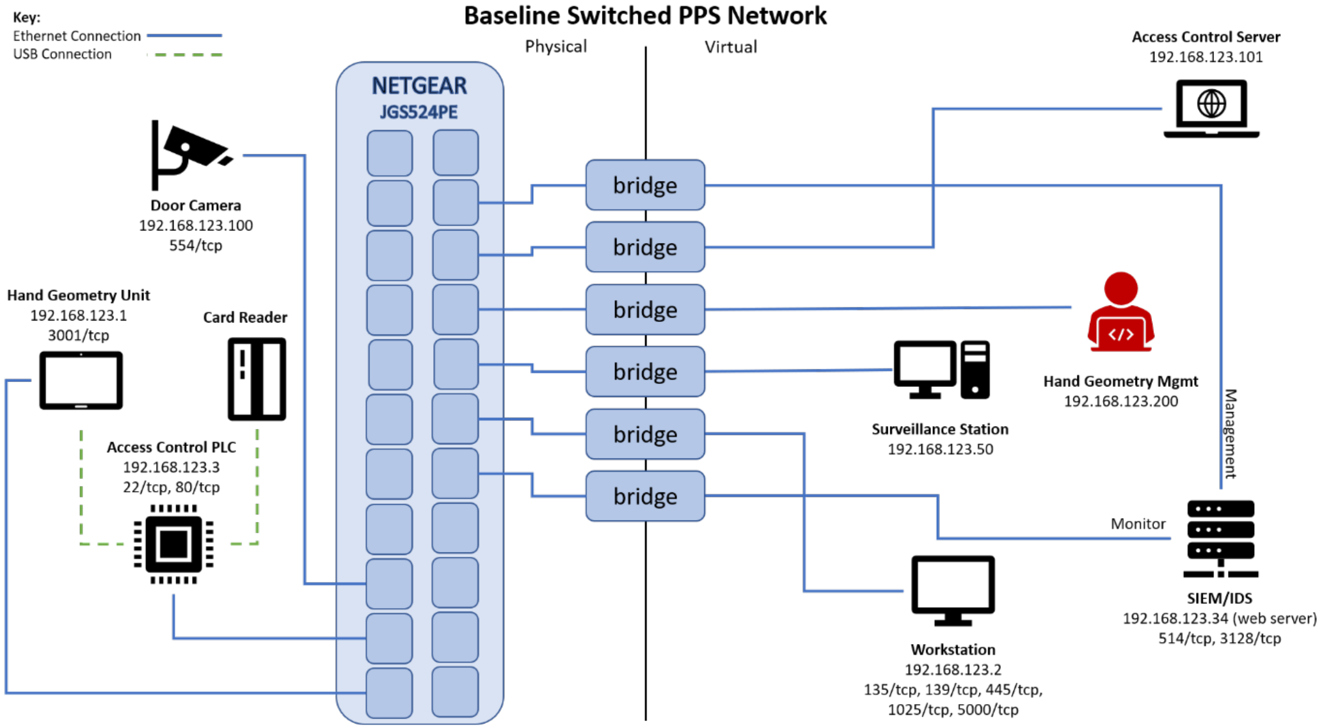

Figure 2: Baseline Architecture Diagram

3.2. SDN Based Architecture

The modified network replaces the baseline switch with a software defined network capable switch and a management host with configuration software. The physically connected port on an SDN switch is essential for properly writing flow rules. The SEL-2740S has 20 data ports and one management port. Only the ports used in the study are included in the illustration of the SEL-2740S in Figure 3. All changes from Figure 2 are highlighted in orange in Figure 3.

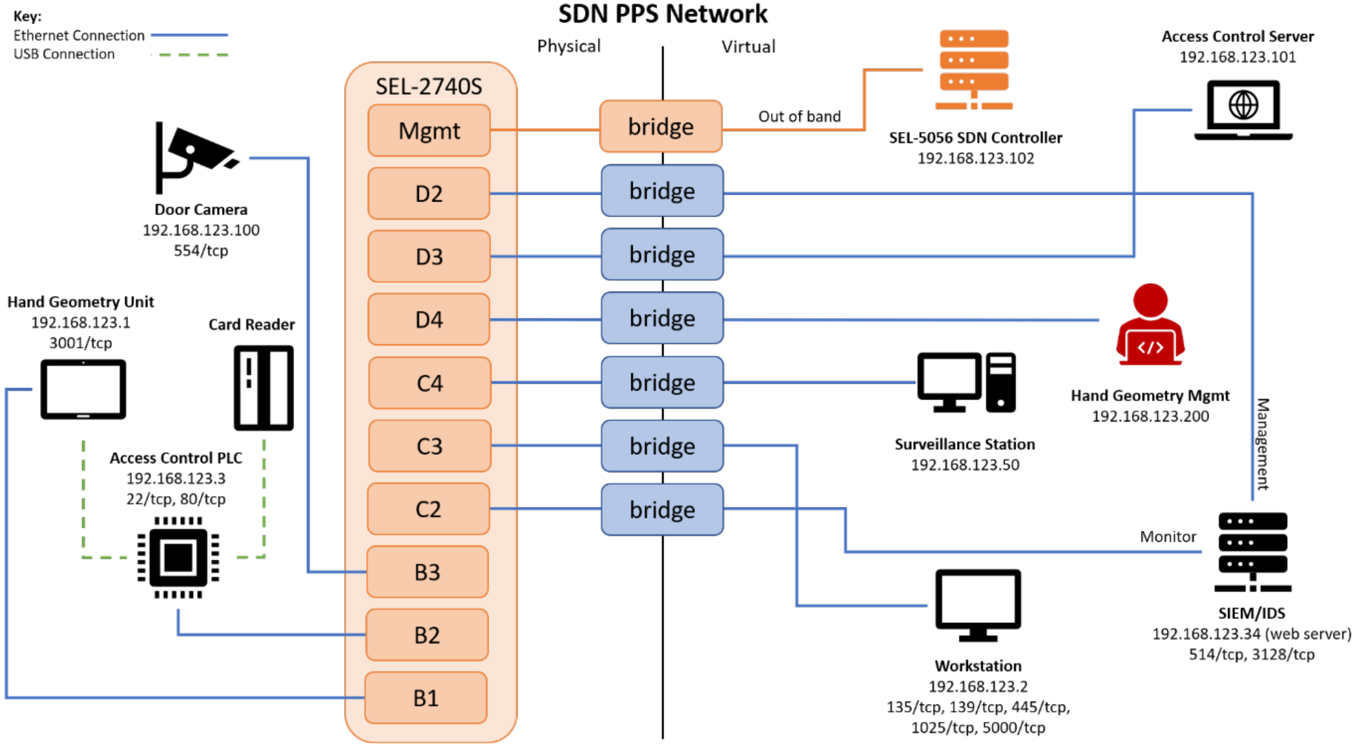

Figure 3: SDN Network Architecture

Table 2 contains the flow rules defined in the SDN environment to enable the basic functionality desired as defined in the test plan described below. For clarity, irrelevant rules have been removed including the mandatory rules for the switch to function.

Name	SRC IP	DST IP	DST Port	Protocol	ARP TPA	SRC Physical Port	DST Physical Port	Comments
Access Ctrl PLC				ARP	192.168.123.3		B2	Enable ARP to PLC
Access Ctrl PLC WEB	192.168.123.101	192.168.123.3	80	TCP			B2	Enable Web to From ACS to PLC
Access Ctrl PLC WEB Return	192.168.123.3	192.168.123.101	80	TCP			D3	Enable Web response from PLC to ACS
Access Ctrl Server ARP				ARP	192.168.123.101		D3	Enable ARP to ACS
Camera ARP				ARP	192.168.123.100		B3	Enable Camera ARP
CameraRTSP	192.168.123.50	192.168.123.100	554	TCP		C4	B3	Enable RTSP from SurvStation to Cam
CameraRTSPReturn	192.168.123.100	192.168.123.50	554	TCP		B3	C4	Enable RSTP from Cam to SurvStation
HandGeoMgmt ARP				ARP	192.168.123.200		D4	Enable ARP on Hand Geo Mgmt.
HandGeoMgmt3001	192.168.123.200	192.168.123.1	3001	TCP			B1	Enable 3001 to Hand Geo Unit from Hand Geo Mgmt.
HandGeoMgmt3001Return	192.168.123.1	192.168.123.200	3001	TCP			D4	Enable 3001 response to Hand Geo Mgmt. from Hand Geo Unit
SIEM ARP				ARP	192.168.123.102		D2	Enable ARP on SIEM
SIEM HTTPS	192.168.123.101	192.168.123.34	443	TCP			D2	Enable HTTPS from ACS to SIEM
SIEM HTTPS Return	192.168.123.34	192.168.123.101	443	TCP			D3	Enable HTTPS Response to ACS from SIEM
SurvStation ARP				ARP	192.168.123.50		C4	Enable ARP on SurvStation

Table 2: SDN Added Ruleset

4. EXPERIMENT TEST PLAN

The authors created a series of tests to evaluate the baseline environment and then contrast it with the SDN environment. The initial tests ensure the system functions as designed. The remaining tests conducted evaluate the environment against activities an attacker may do in various stages of an attack. The desired system behaviour under normal conditions consists of the following activities. A user should be able to:

- (a) Program the Hand Geometry Unit from the device itself
- (b) Administer the Hand Geometry Unit remotely from the Hand Geometry Management system.
- (c) Unlock the door using both the Hand Geometry Unit and the card reader together
    - (1) If only one authentication method has been used, the red light will stay illuminated for ten seconds. If another method has not been used by then, the red light goes out and the yellow light flashes three times.
    - (2) If both authentication methods are used (in either order), the green light flashes once and the door opens.
    - (3) The red-light flashes if the door is left open for more than thirty seconds. It stops flashing after the door has been closed.
- (d) View the Access Control on the Access Control Server
    - (1) Verify that the door log updates correctly
- (e) View the live feed from the door camera on the Surveillance Station
- (f) Access the IDS web interface from the Access Control Server

Once the baseline system was configured and successful verification of normal system behaviour was functional, the following tests were conducted. Table 3 identifies the name of the test, a brief description of what the test entails, the expected results for both the baseline and SDN environments and the specific commands issued.

Name	Description	Expected results
Nmap Scan	Perform a Nmap scan on the local network from the Hand Geometry Management device using the -A switch. -A Enables OS detection, version detection, script scanning, and traceroute	Baseline: Most, if not all, hosts and information on open ports should be visible to Nmap on the baseline network. Host operating system identification may be successful. SDN: Hosts are not expected to be identified via this scan. Host identification and port information should fail.
	Command(s) nmap -v -A 192.168.123.0/24	
MAC Table Flood	Use macof.py to perform a mac table flood attack against the switch. On the baseline network, the target is the physical switch in the INS kit. On the SDN, the target is the SEL-2740S. Evaluate the effects of the attack by checking if the camera, door and door log continue to function normally and whether the Hand Geometry Unit can still be connected to remotely. Command(s) macof.py -w 1 -dip 192.168.123.12	Baseline: Traffic will be briefly interrupted as the switch rebuilds the MAC address table. SDN: The SEL-2740S does not have a MAC table, so it will not be affected by the attack.
ARP Scan	Use arp-scan to attempt to find hosts on the local network.	Baseline: Most, if not all, of the devices will be found. SDN: At least one device should be found, the Hand Geometry Unit.

	Command(s) arp-scan -interface=eth0 192.168.123.0/24	
ARP Fingerprint	Use arp-fingerprint to attempt to identify hosts on the local network Command(s) arp-fingerprint -l	Baseline: Hosts will be found and an attempt to identify them will be made. SDN: At least one device should be found, the Hand Geometry Unit, with an attempt at identification
Camera Feed DoS	Attempt to deny the Surveillance Station access to the live camera feed by targeting it with an hping3 packet flood. Run this test with three different types of packets: first TCP, then ICMP, then UDP, evaluating the functionality of the camera (in both VLC and iSpy on the Surveillance Station), the door and access log during the attack. After stopping the attack, evaluate whether the camera feed is able to recover and if so, how long it takes to recover. Command(s) hping3 –flood 192.168.123.100 hping3 -flood -icmp 192.168.123.100 hping3 -flood -udp 192.168.123.100	Baseline: All three variants should be successful with the same results. The camera feeds will be interrupted, but the door and door log will continue to function normally SDN: The DoS attack will not be successful because the packets will be dropped.
Reverse TCP Shell	Use msfvenom to create a reverse shell executable. Copy it to the Access Control Server. Set up a handler on the Hand Geometry Management system. Execute the executable on the Access Control Server Command(s) msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.123.200 lport=4444 -f exe -o ./payload.exe msfconsole msf > use exploit/multi/handler msf exploit(handler) > set payload windows/meterpreter/reverse_tcp msf exploit(handler) > set lhost 192.168.123.200 msf exploit(handler) > set lport 4444 msf exploit(handler) > run	Baseline: The reverse TCP connection will succeed, and the Hand Geometry Management system will be able to open a shell on the Access Control Server. This action should be flagged by the IDS. SDN: The reverse shell connection will not succeed, and there will be no action for the IDS to flag.

Table 3: Test Descriptions

Once the tests were run and data collected on the baseline environment, the Netgear switch was replaced with the SEL-2740S switch and the SEL-5056 management software connected via the front management port. The SEL2740S was configured following the SEL Flow Programming Methodology provided in the SEL 2740S manual Section 5 [13]. The switch was configured to ensure the six basic functionality tests described at the beginning of this section were successful. Then the same battery of test conducted on the baseline system were conducted on the SDN environment. The results of the tests on both the baseline and SDN environments are detailed in the following section.

5. EXPERIMENTAL RESULTS

Results of Tests on the Baseline Network

Test Name	Baseline Network	SDN Network
Nmap Scan	Nmap was able to find and correctly identify the Workstation, the Access Control PLC, the IDS webserver, Surveillance Station, the Door Camera, the Access Control Server, and the Hand Geometry Management device. Nmap was able to find but not identify the Hand Geometry Unit.	Nmap found the Hand Geometry Unit, the Access Control PLC, the IDS web server, the Surveillance Station, the Door Camera, the Access Control Server, and the Hand Geometry Management device (from which the scan was run); however, device information was limited. Nmap was only able to attempt device identification by reporting the company to which each MAC address was registered. All ports on all devices were reported as filtered except port 3001 on the Hand Geometry Unit, which is used for remote administration of that device, and port 111, which was on the local host. OS detection failed on all hosts except the local host. The SDN controller was not found.
MAC Table Flood	The Mac Table Flood against the physical switch had no effect on the functionality of the system.	As expected, MAC table flooding, which in this case targeted the IP address of the port used to administer the SEL-2740S, was completely unsuccessful. The system continued to function normally.
ARP Scan	ARP scanning was able to find all the devices on the network. It did not, however, able to detect the connection used by the IDS to sniff traffic.	The ARP scan was able to find the Hand Geometry Unit, the Access Control PLC, the ISD web server, the Surveillance Station, the Door Camera, and the Access Control Server. The SDN Controller was not found. The Hand Geometry Management device was not found because it was the local host.
ARP Fingerprint	ARP fingerprinting was able to identify all devices except the Hand Geometry Unit and the Workstation as running desktop operating systems. Additionally, the Hand Geometry Management device was not scanned because it performed the scan.	ARP fingerprinting completely failed. The Hand Geometry device was labelled “UNKNOWN,” and all other devices found (the same list as ARP Scan) were incorrectly identified as running some version of Cisco IOS.

Camera Feed DoS	The TCP packet flood did not affect the functionality of the system. The ICMP flood caused instant interruption of the video stream. The VLC stream died after several seconds and the iSpy stream froze on the last frame. The video stats in iSpy also froze. The rest of the system continued to function normally. Resuming the video feed after the flood was stopped took about a minute. The UDP flood caused the video feed to be choppy and delayed. The image froze several times and the connection was intermittent. VLC disconnected and stayed disconnected while iSpy was able to automatically reconnect. The rest of the system continued to function normally. The video feed quickly recovered after the flood was stopped.	The same packet flood DoS attack against the Door Camera was run three times - once with TCP packets, once with ICMP packets, and once with UDP packets. Each flood was allowed to run for a full minute. During each attack, the video feed remained uninterrupted, and the system continued to function normally.
Reverse TCP Shell	Executing payload.exe caused the remote shell to open immediately on the Hand Geometry Management device. The IDS generated multiple messages that correctly identified a shellcode attack.	Executing payload.exe did not open a reverse shell on the Hand Geometry Management device. Wireshark revealed that the Access Control Server was sending SYN packets to the Hand Geometry Management device, but they were not delivered. The IDS did not flag any suspicious activity.

Table 4: Test Results

6. CONCLUSIONS

The results were much as expected. The SDN environment improves security by increasing the Defence in Depth of the network, limiting connectivity between devices that are not required to interact under the design basis, and by increasing the cost to the adversary to conduct an attack by limiting reconnaissance and lateral movement capabilities. The ability of the Nmap scan to identify all the hosts on the network was unexpected. Upon further investigation the NMAP, with the flags used, includes an ARP scan for enumeration. Discovering this insight emphasizes the need to have a sufficient understanding of the environment to establish accurate models from which to derive rulesets. It also highlights the need to understand tools and how the operate.

The experimental environment was intentionally simple as a starting point. All indications show that SDN technology can successfully be used in a nuclear physical protection system without impacting performance and improving the computer security of the overall system. The next steps to further this research would include testing situational awareness capabilities by redirecting all denied traffic to an analysis engine, conducting the same tests in a more complex, multi-switch, environment, and conducting a pilot deployment of SDN technologies on a functioning full-scale physical protection system. It would also be worth studying the operational costs of establishing, configuring and maintaining an SDN environment versus currently deployed environments.

ACKNOWLEDGEMENTS

This research was supported in part by the International Nuclear Security Program, National Nuclear Security Administration of the U.S. Department of Energy, under contracts DE-AC05-76RL01830 and DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

REFERENCES

[1] W. J. Hutton, M. D. Hadley and A. D. McKinnon, Software-Defined Networking Impacts on the OT Security OODA Loop, 2019.

[2] N. Falliere, L. O. Murchu and E. Chen, “W32.Stuxnet Dossier,” Symantec, Mountain View, 2010.

[3] Dragos Inc, “TRISIS Malware,” Dragos Inc, Hanover MD, 2017.

[4] National Institute of Standards and Technology, “NISTIR 8183,” 05 20 2019. [Online]. Available: https:/nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8183.pdf. [Accessed 04 11 2019].

[5] IAEA, “Computer Security for Nuclear Security (Draft),” 12 2016. [Online]. Available: https:/www-ns.iaea.org/downloads/security/security-series-drafts/implem-guides/nst045.pdf. [Accessed 4 11 2019].

[6] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolomolky and S. Uhlig, “Software-Defined Networking: A Comprehensive Study,” IEEE, 2014.

[7] R. di Lallo, F. Griscioli, G. Lospoto, H. Mostafaei, M. Pizzonia and M. Rimondini, “Leveraging SDN to monitror Critical Infrastructure Networks in a Smarter Way,” in IFIP/IEEE International Symposium on Integrated Network and Service Management, Lisbon, 2017.

[8] E. Molina and E. Jacob, “Software-defined networking in cyber-physical systems: A survey,” Computers & Electrical Engineering, vol. 66, pp. 407-419, 2018.

[9] G. K. Ndonda and R. Sadre, “A low-delay SDN-based countermeasure to eavesdropping attacks in industrial control systems,” in 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin, Germany, 2017.

[10] A. F. M. Piedrahita, V. Gaur, J. Giraldo, A. A. Cardenas and S. J. Rueda, “Leveraging Software-Defined Networking for Incident Response in Industrial Control Systems,” IEEE Software, vol. 35, no. 1, pp. 44-50, 2018.

[11] F. Patzer, A. Meshram and M. Heß , “Automated Incident Response for Industrial Control Systems Leveraging Software-Defined Networking,” in International Conference on Information Systems Security and Privacy, Prague, Czech Republic, 2019.

[12] M. Sainz, I. Garitan and U. Zurutuza, “Software Defined Networking Opportunities for Intelligent Security Enhancement of Industrial Control Systems,” in International Joint Conference SOCO’17-CISIS’17-ICEUTE’17, León, Spain, 2017.

[13] Schweitzer Engineering Laboratory, “SEL-2740S Software Defined Network (SDN) Switch SEL-5056 SDN Flow Controller Instruction Manual,” 14 06 2019. [Online]. Available: https:/selinc.com/products/5056/. [Accessed 04 11 2019].