Can Open-Source Tools Be Used to Safely Scan a Modern ICS Environment?

Author: Josh Tanski, josh@tanski.net

Advisor: *Michael Long*

Accepted: *November 5, 2023*

Abstract

This research delves into the long-standing belief within the Operational Technology (OT) security community that active scanning in OT and Industrial Control Systems (ICS) environments is exceptionally risky. Historical accounts dating back to the 1990s have warned of system disruptions caused by IT-initiated security scans, emphasizing issues like system crashes and latency problems, which can endanger the safety of critical processes. Consequently, the OT security industry has predominantly relied on passive scanning methods. This paper challenges the prevailing notion that active scanning in OT systems should be avoided, advocating for a reassessment of this stance as both ICS technology and scanning methods mature. Active scanning can bring significant benefits, including streamlined asset and vulnerability management processes, reduced manual efforts, and enhanced security.

Findings revealed that safe scanning practices can mitigate the risks associated with active scanning in OT environments. Ultimately, this research underscores the importance of reevaluating the perception of active scanning in OT and ICS systems, advocating for safer scanning practices, and calling upon vendors to fortify their products against potential disruptions caused by security scans.

1. Introduction

Anecdotal knowledge among the OT (Operational Technology) security community is that running any active scanner in an OT or ICS (industrial control systems) environment is extremely dangerous. There are anecdotes and stories about scanning breaking OT systems going back to at least the 1990s. For example, the book "Hacking Exposed Industrial Control Systems" (Bodungen et al., 2017) starts with a two-page case study of employees in a fictitious company dealing with PLC (Programmable Logic Controllers, a typical component of an ICS) issues that turn out to be caused by IT running basic security scans. "Penetration Testing of Industrial Control Systems" (Duggan et al., 2005), a frequently cited research paper, has several examples of penetration testing causing significant problems. Ping sweeps caused robots to move uncontrollably and dangerously at one factory. At another factory, ping sweeps generated $50K worth of damaged circuits. Penetration testing caused a gas company to have a four-hour outage in supplying its customers' service. (Duggan et al., 2005).

Rather than continue to share these stories from years ago as absolute truths, the goal is to determine whether the active scanner scare is based on science or superstition. The security industry must continuously challenge the concept that running active scans on OT systems is unsafe as ICS and scanning technologies mature.

The official NIST definition of OT is "Programmable systems or devices that interact with the physical environment (or manage devices that interact with the physical environment). These systems/devices detect or cause a direct change through the monitoring and/or control of devices, processes, and events" (NIST). Real-world examples of OT include hoists and conveyors in a mine, energy-generating turbines in a power plant, or refrigeration in a food warehouse. Unlike traditional IT security incidents, the impact of an OT security incident can include physical damage to equipment and facilities, food spoilage, environmental harm, and human injury or death. This paper will use the terms OT and ICS interchangeably.

One common issue caused by OT scanning includes system crashes, which cause the loss of control of the industrial process or machinery. Another problem is latency

issues, or slowness in the response time of the systems. An example at an electric company is a remotely controlled substation needing service work. If a technician needs to enter the substation, the operator must shut off power so the technician can safely enter. The operator expects the power-off button to work precisely when he presses it. If there is latency, the technician could unexpectedly be exposed to energized equipment, causing a significant safety concern.

Active scanners are technologies that directly communicate with the devices they are scanning over the network. Passive scanners are sensors that monitor existing network traffic and do not generate any network traffic themselves (to the monitored devices). Because of the perceived danger of active scanners, ICS security relies heavily on passive scanning technology. The challenge with passive scanners is that they only know what they can monitor – they must infer any vulnerabilities from existing data in the network traffic. The sensors must be appropriately placed and configured throughout a company's ICS environments.

Safe OT active scanning would help enable OT security activities to be automated, quick, and efficient, ultimately enabling OT to be more secure. Manual and time-consuming security activities are only sometimes done for OT and are not always done completely or correctly. Active scanning is also more accurate than passive scanning. Active scanning can discover open ports or exact firmware versions that passive scanning may not find. A benefit of active OT scanning for asset management would be detecting network devices automatically and interrogating them for system or firmware information. An advantage of active OT scanning for vulnerability and patch management would be finding vulnerabilities and missing patches and tracking whether the vulnerabilities are fixed or if the patches are installed.

2. Research Method

2.1. Description of the Lab Environment

The lab for the experiment is composed of hardware devices and virtual machines (VMs), as depicted in Figure 1. The lab utilizes a private, physically separate wired network to simulate a control network, as typical for an ICS. A common item seen in a factory is an electrical panel with a network switch, a PLC, and an HMI. The lab laptop has Internet access for downloading necessary software or security updates. For full details of the lab environment, please see Appendix 1.

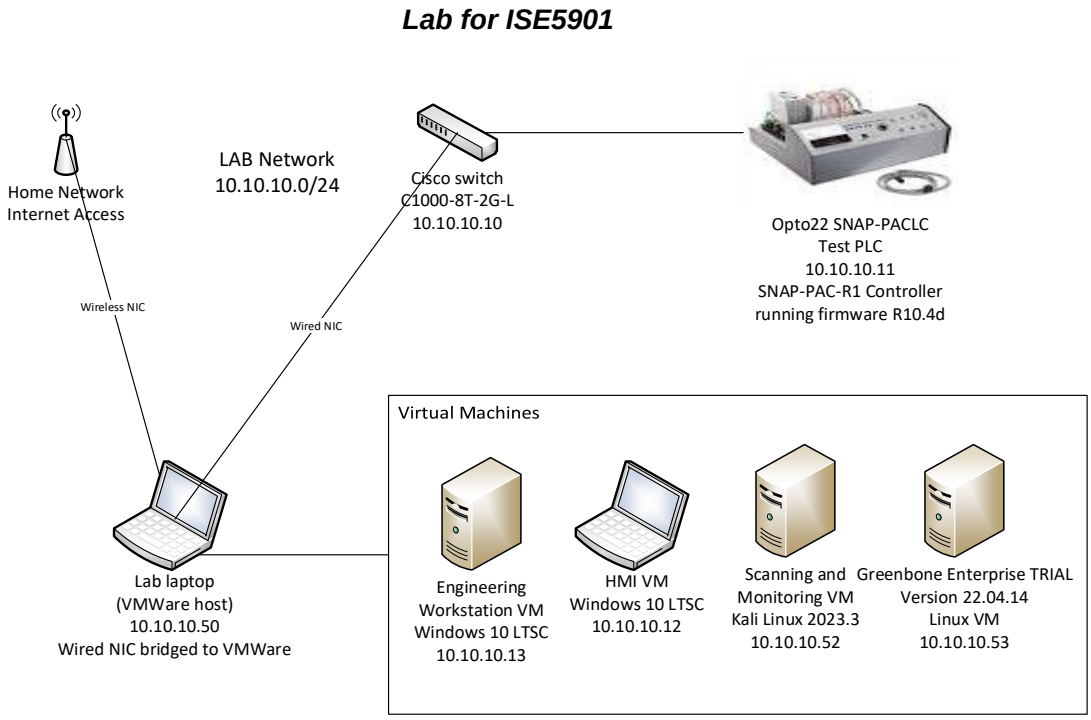

Figure 1 - Lab Diagram

2.2. How Opto22 SNAP-PACLC is representative of a modern ICS environment

A modern ICS environment is one that is still regularly used in the industry, and one that the vendor supports and has available for sale. It does not need to be the latest, cutting-edge technology – as OT favors reliability. The Opto22 SNAP-PACLC Learning Center includes a SNAP-PAC-R1 controller and I/O (Input/Output) modules that are still

available and supported by Opto22. Most importantly, the SNAP-PAC-R1 hardware and software versions for the learning center are the same as what is sold to be used in factories and not explicitly created as a training environment. Opto22 released the latest firmware in 2022. Components to replicate the rest of the Learning Center (e.g., buttons, lights, power supply) are readily available at industrial supply companies.

The food and beverage industry uses Opto22 PAC controllers. Opto22 is typical of the many niche vendors in the OT industry. For example, on CISA.gov's "Cybersecurity Alerts & Advisories" page, after expanding Vendors under ICS Advisories, there are hundreds of vendors in the list, many of which are smaller or niche component providers to various industries.

2.3. How to measure system reliability while running scans

To determine how the scans affect the test equipment, one must monitor it while the scans are happening to see any adverse effects. Unfortunately, Opto22 PAC controllers are very limited in what one can watch for low-level system performance or health. There is no interface to view CPU or memory usage like a Windows or Linux system. However, one can still manually monitor the hardware and software for deviations from normal operating conditions, and the software can monitor latency (called loop time) and communication errors from the "View Status" tool in the included PAC Terminal utility application. Opto22 product support was contacted to inquire about additional monitoring options available in the controller's API or elsewhere. They determined this was the only indicator available (N. Freeman, personal communication, March 17, 2023).

2.3.1. Hardware

The Opto22 PAC Learning Center (Figure 2) includes a physical control panel with switches, lights, and an alarm speaker. To monitor normal conditions, the alarms function as configured in the Convenience Store simulation, developed for the training kit, and freely downloaded from Opto22's website. The low fuel alarm is a short beep every three seconds, which can be triggered by turning the fuel level to zero (all the way

left). The emergency alarm is a continuous beep, which can be triggered by holding the emergency switch on for at least two seconds and staying on until the switch is released.

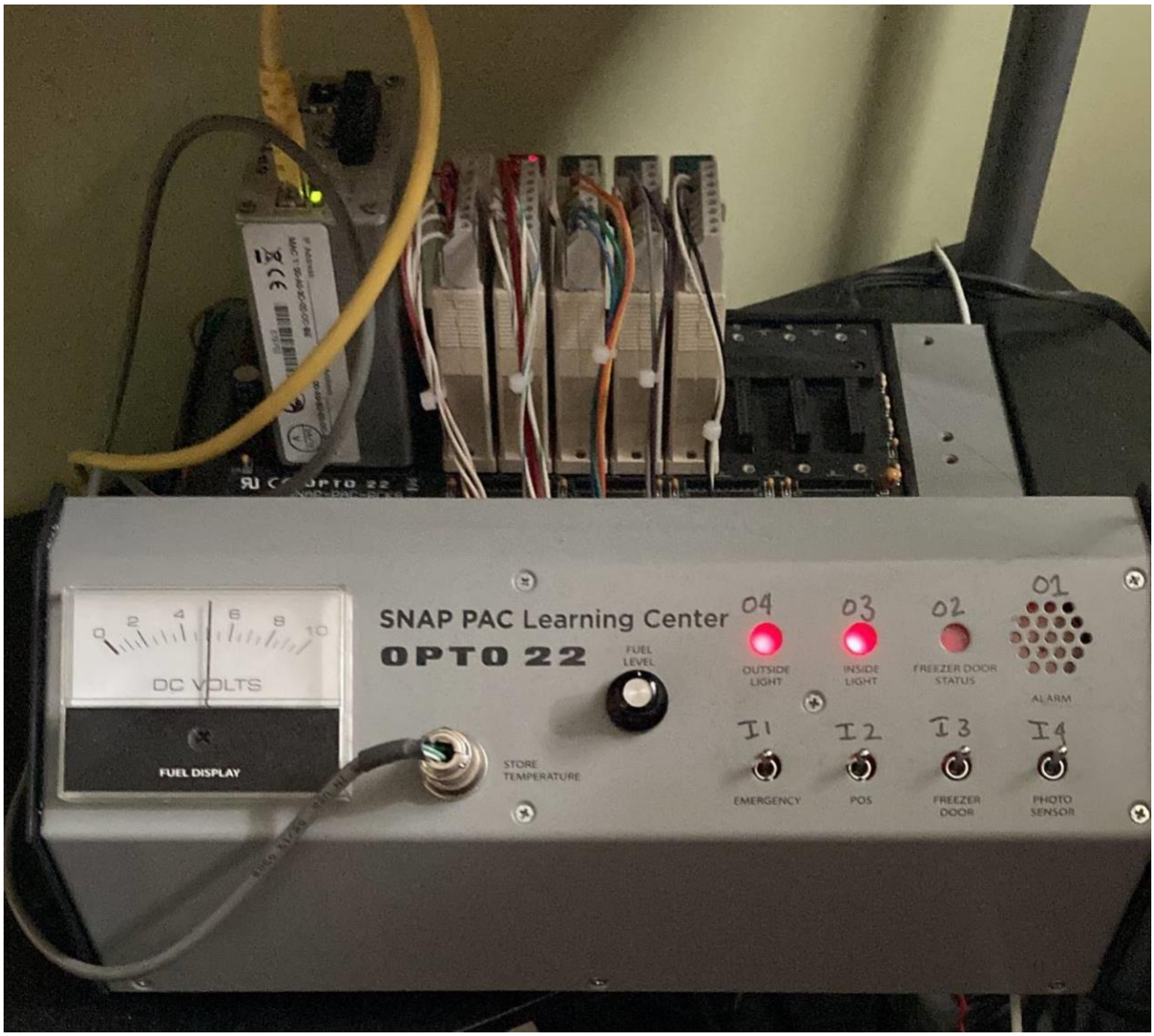

Figure 2 - Picture of Opto22 SNAP PAC Learning Center

2.3.2. Software

The Engineering Workstation and the HMI can use the PAC Terminal application to monitor normal control system operation by inspecting controller communication loop time and errors. PAC Terminal is shown in Figure 3.

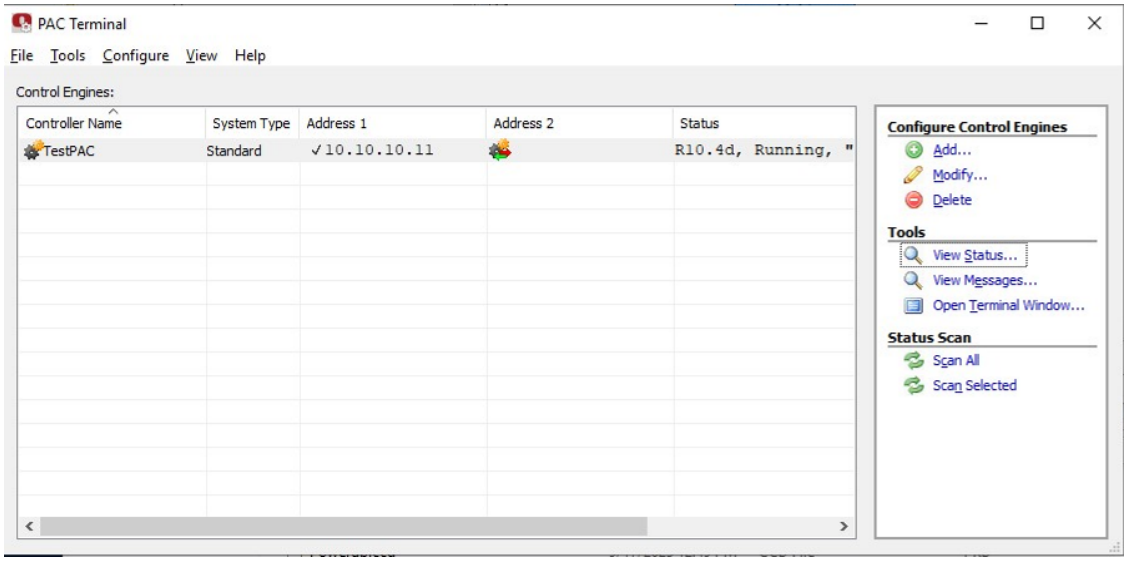

Figure 3 - PAC Terminal utility application

Loop time is latency and is typically in the eight to twelve-millisecond range for the lab configuration. There are no communication errors during regular operations, as depicted in Figure 4.

Figure 4 - Inspecting controller, normal loop time

The Engineering Workstation uses the PAC Control application for the engineer to program, monitor, and debug the input-outpoints on the controller. The application has screens to show the physical variables read by the controller. In normal operation, this looks like Figure 5, and the user can responsively navigate the various screens in the application.

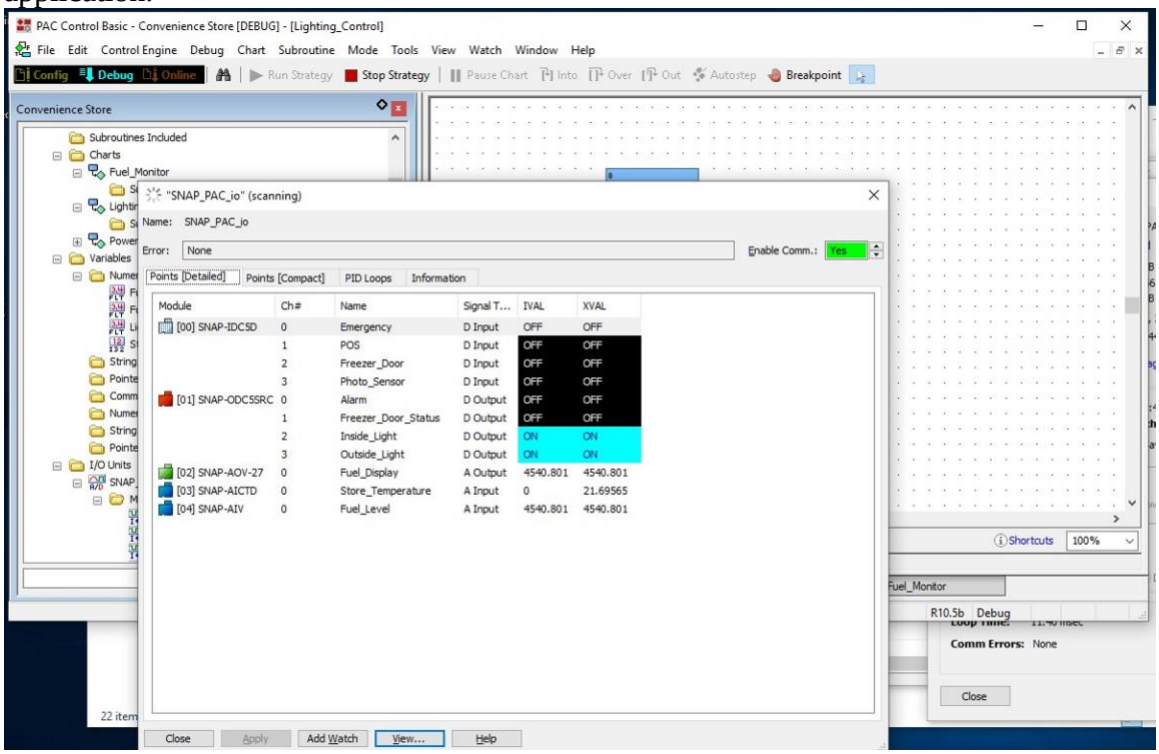

Figure 5 - Normal operation of PAC Control on the Engineering Workstation

If communication breaks on the Engineering Workstation, an error screen appears as shown in Figure 6.

Figure 6 - PAC Control error popup on the Engineering Workstation

The HMI uses the PAC Display application in runtime mode for the control system operator to view and control the system. The main screen of the HMI is shown in Figure 7.

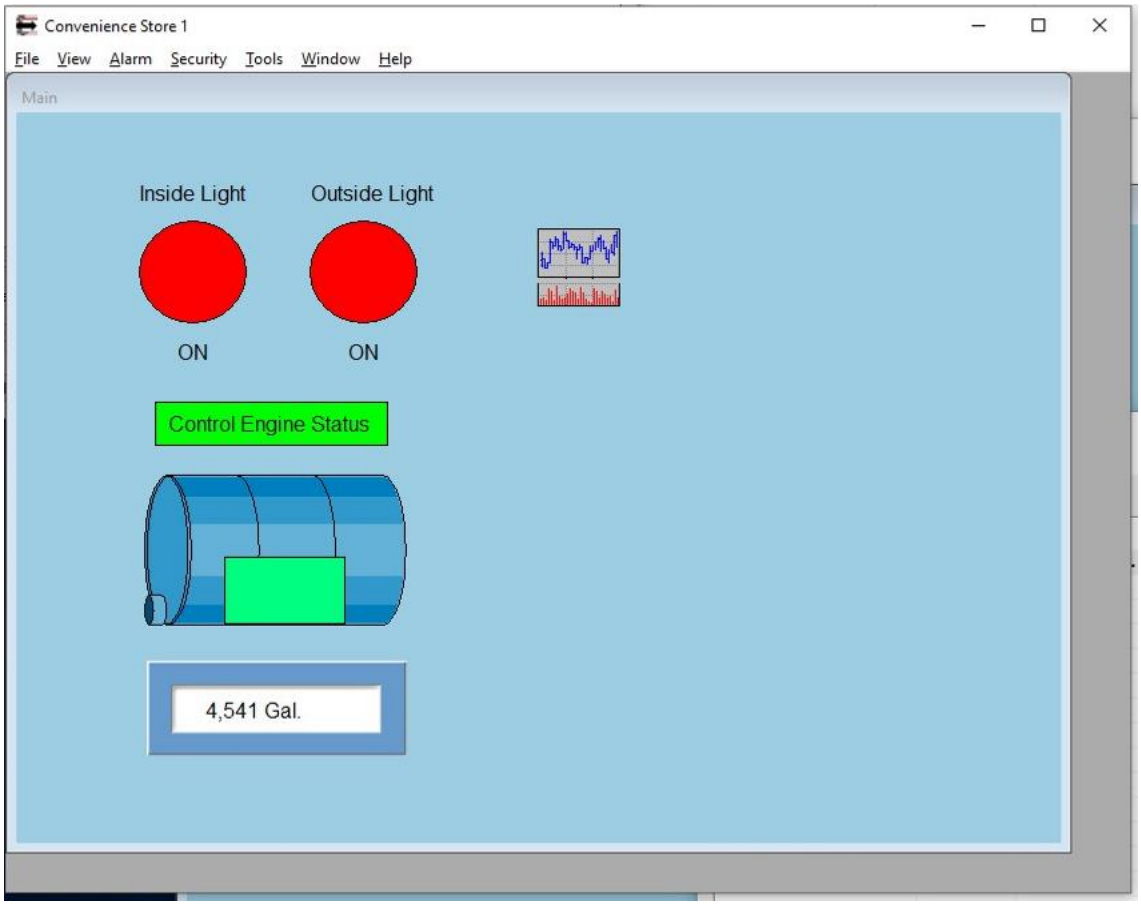

Figure 7 - Normal operation of PAC Display on the HMI

If something abnormal happens to the HMI, the application pops up an event log viewer to display the errors, as depicted in Figure 8.

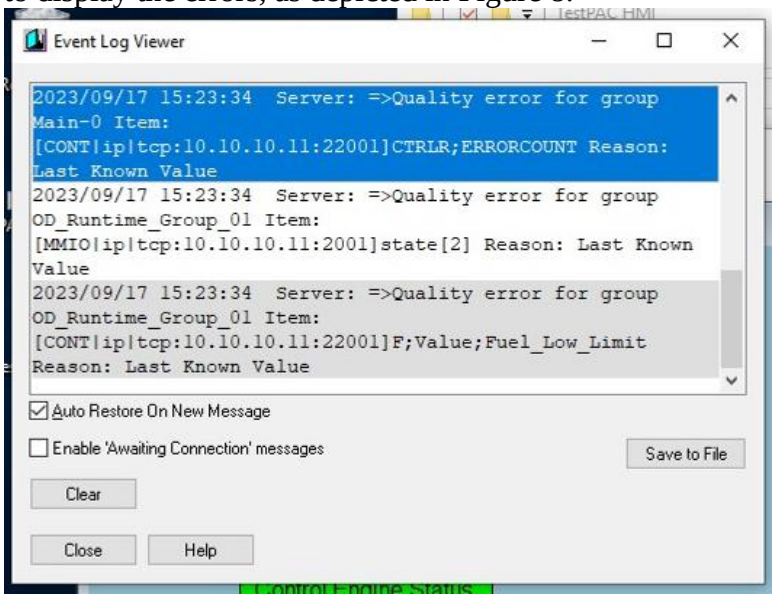

Figure 6 - PAC Display error popup on the HMI

2.4. Scanners Used

ARP-scan, Nmap, and Greenbone OpenVAS were run against the target controller to determine if open-source scanners affect the PAC controller. If there are no immediate adverse effects, run the scan 100 times and see if or how latency is impacted.

1. **ARP-scan** – ARP-scan is a tool that uses ARP (Address Resolution Protocol) to discover devices on the local network. ARP is a protocol for discovering a device's MAC address (or network card address), given its IP address. ARP is part of the standard Ethernet communication to allow IP layer 3 addresses to communicate with each other at layer 2. ARP-scan is a simple way of discovering assets but only works on the local network.
2. **Nmap** – Nmap is the most well-known and widely used open-source security scanner. Nmap scans at layer 3, so it works across any routable IP address space. Nmap can discover devices and identify the ports and services the devices are running.
3. **Greenbone OpenVAS** – Greenbone is a full-featured vulnerability assessment scanner. Greenbone can discover devices and ports like Nmap. However, it also has an extensive database to detect vulnerabilities and misconfigurations in the

devices it scans, which a malicious attacker could use to exploit. The goal of doing a vulnerability scan is to detect vulnerabilities so they can be fixed before they can be exploited.

3. Findings and Discussion

3.1. ARP-scan

Running an ARP scan against the control network had no noticeable effect on the systems. Employing a simple bash script to run the scan 100 times to see if there was any effect, and there was none. Therefore, simple ARP scans are not expected to be dangerous to control systems and could be a handy tool for asset discovery. However, the downside is that the ARP scan needs to be run locally per subnet scanned.

The command run for this scan:

```
$  sudo arp-scan -I eth1 --ouifile /usr/share/arp-scan/ieee-oui.txt --
macfile /etc/arp-scan/mac-vendor.txt 10.10.10.0/24
```

Option -I is to set to the correct network interface on the Kali VM. Options –outfile and –macfile are to specify the correct locations of the respective database in the Kali OS.

Results:

```
Interface: eth1, type: EN10MB, MAC: 00:0c:29:04:37:e2, IPv4:
10.10.10.52
Starting arp-scan 1.10.0 with 256 hosts
(https://github.com/royhills/arp-scan)
10.10.10.10     00:df:1d:eb:dd:c0       Cisco Systems, Inc
10.10.10.12     00:0c:29:d0:01:7a       VMware, Inc.
10.10.10.13     00:0c:29:9f:db:06       VMware, Inc.
10.10.10.11     00:a0:3d:02:cc:be       OPTO-22
10.10.10.50     e8:6a:64:fc:94:15       LCFC(HeFei) Electronics
Technology co., ltd
```

Script:

```
#!/bin/bash
for i in {1..100}
do
        echo $i
        arp-scan -I eth1 --ouifile /usr/share/arp-scan/ieee-oui.txt --
macfile /etc/arp-scan/mac-vendor.txt 10.10.10.0/24
Done
```

Please note that when running Nmap via a similar script, modify the above script and substitute the ARP-scan command line with the applicable Nmap command line. Control-C stops the script if it runs longer than is needed to return/receive results.

3.2. Nmap Scans

Several Nmap scans were employed against the controller to see how it was affected. The raw results of the Nmap scan are in Appendix 2. All Nmap commands were run as root (using sudo), as running as root gives Nmap raw socket access necessary for some of the scans, and Nmap uses different default settings if run as a non-privileged user. One of the different defaults is whether to do SYN scans or full CONNECT scans, which would affect the experiment results.

3.2.1. Nmap default scan against controller

The default Nmap settings were tried as a starting point for the initial scan. When run 100 times, this scan had minimal effect on the control system. The loop time increased from a normal of 8 to 12 milliseconds to about 35 milliseconds and had no noticeable effect on the functioning of the hardware or software.

The command run for this scan:

```
sudo nmap 10.10.10.11
```

According to the Nmap Project (n.d.), Nmap scans the most common 1,000 ports by default. However, this scan only found two ports open on the controller – TCP 21, the controller's embedded FTP server, and TCP 2001, a proprietary Opto22 communication protocol (OptoMMP – memory map protocol). The scan did not find Modbus or other common ports for ICS protocols expected to be open on the controller.

Also, the default scan type performed is a SYN scan, where Nmap sends a TCP SYN packet and uses the response packet to determine whether the port is open or closed but never completes the expected TCP three-way handshake, speeding up the operation of Nmap but leaving the target system's connection in a half-open state. This scanning is considered unsafe in an ICS environment as the specialized hardware may run out of resources to maintain the half-open connections.

3.2.2. Nmap all-ports against controller

The all-ports scan finds all open ports, which was necessary to identify typical ICS protocols the controller runs – including Modbus on TCP 502 and EtherNet/IP on TCP 44818. The full scan also found TCP 22001 open for PAC Control, another Opto22 proprietary communication protocol.

The command run for this scan:

```
sudo nmap -p1-65535 10.10.10.11
```

This scan immediately caused issues with the control system. The engineering workstation immediately had a communications error; the loop time is indeterminable without communications. When ran 100 times, the HMI had inconsistent results. Sometimes, the HMI lost communications completely; other times, the HMI still functioned, but the loop time increased to between 800 and 1500 milliseconds.

The hardware itself experienced a minor loss of control. It still functioned with some noticeable, yet less than a second, delays when attempting to trigger the alarm beeper.

A significant result is that the control system recovered after any scans that caused problems. The hardware recovered on its own and functioned normally after the problem scans stopped, and any software errors on the Windows systems were cleared by clicking the "retry" button on the error message. Nothing caused a permanent denial of service or complete hardware failure.

3.2.3. Nmap only open ports against the controller

Since the previous scan caused issues with the control system responsiveness, the issue was isolated to find a way to limit the scans to reduce the probability of issues. The first attempt was to limit the scan to only the open ports to see if scanning the open ports caused the issues. However, this causes no issues whatsoever. Scanners could use this technique to further probe the ports for more information. However, not all ICS systems thoroughly document what ports they have to open, so active scanning may still be necessary to determine the open ports.

The command run for this scan:

```
sudo nmap -p21,502,2001,22001,22500-22531,44818 10.10.10.11
```

3.2.4. Nmap all ports, full CONNECT scan, against the controller

One option considered safer for ICS environments is a CONNECT scan, which completes a full TCP handshake instead of leaving connections half open like a SYN scan. For example, the section on ICS/SCADA security of the Nmap tutorial by Gnebbia (n.d.) discusses the importance of securing industrial control systems: “It is advised to run -sT scans to open and close each connection, since these devices may be very susceptible.”

The command run for this scan:

```
sudo nmap -sT -p1-65535 10.10.10.11
```

However, this scan produced the worst results of all the tests. Both the HMI and engineering workstation immediately throw errors when the scans happen. There are several-second delays on the hardware between triggered alarms and when the alarm beeper starts or stops.

3.2.5. Nmap all ports, full CONNECT scan, adding delays, against the controller

The next change to the scans was to see if a safer method could be found by adding a delay between port scans. It was initially thought a one-second delay would be very safe but aborted the scan once it was realized it would make scanning all 65535 TCP take over 18 hours. A 50-millisecond delay was then tried, which ran for about 55

minutes, and that caused no effect on the control system. A ten-millisecond delay was finally tried, which took about 11 minutes to run and caused no issues, which was a reasonable tradeoff between time and safety.

The command run for this scan:

```
sudo nmap -sT --scan-delay 10ms -p1-65535 10.10.10.11
```

3.2.6. Nmap all ports, SYN scan with 10ms delay, against the controller

The SYN scan was run again against the controller, with the 10-millisecond delay, as the final scan against the controller. Again, this caused no issues, so safely scanning these systems requires a 10-millisecond delay.

The command run for this scan:

```
sudo nmap -sS --scan-delay 10ms -p1-65535 10.10.10.11
```

3.2.7. Nmap scans against the Windows devices

Several of the same Nmap scans against the Windows devices were ran during the experiment. However, they had no open ports in the configurations used for the lab. The scans also had no noticeable effect on the Windows systems, reinforcing that active scanning on traditional IT systems is safe and that the OT systems need focus.

3.3. Greenbone/OpenVAS Scan

A Greenbone/OpenVAS default scan was executed using the default “Full and fast” config against the controller. The scan took 13 minutes to complete and caused the same problems as the Nmap full CONNECT scan, as OpenVAS is doing an equivalent scan. Therefore, it can be concluded that OpenVAS and similar open-source vulnerability scanners cannot safely be used in an ICS environment without determining safe options for each type of device scanned.

3.4. Summary of Findings

The various scans run for the experiment are summarized in Table 1. The Nmap scan raw results are contained in Appendix 2. Normal loop time was in the eight to twelve millisecond range, and delays are noted, and "error" indicates loop time could not be measured due to communication error.

Table 1 - Summary of findings

Target	Scan	Loop Time effect (Normal 8-12ms)	Software Loss of Control	Hardware Loss of Control
Network	ARP-scan	Normal	No	No
Controller	1 Nmap default	35ms	No	No
Controller	2 Nmap all ports	HMI – inconsistent, some scans 800-1500ms, some Error Engineering Workstation - Error	Yes	Minimal, small delays, less than one second
Controller	3 Nmap open ports only	Normal	No	No
Controller	4 Nmap TCP Connect scan	Error	Yes	Yes – several second delays in Fuel Alarm starting or Emergency Alarm Stopping
Controller	5 Nmap TCP Connect scan with delay	Normal	No	No
Controller	6 Nmap SYN Scan with delay	Normal	No	No
Controller	Greenbone default scan	Error	Yes	Yes – several second delays in Fuel Alarm starting or Emergency Alarm Stopping
Windows	7 Nmap default	Normal	No	No
Windows	8 Nmap all ports	Normal	No	No
Windows	9 Nmap TCP Connect scan	Normal	No	No

4. Recommendations and Implications for Future Research

The most crucial recommendation for the ICS industry is for vendors and manufacturers to recognize that their systems have issues when scanned and build resiliency and protection into their products to prevent problems. Nmap and similar tools have been around for a very long time now. Vendors such as Opto22 should be notified of these issues so they can fix them.

The recommendations for future research would be to experiment with other options and tools freely available that have ICS-specific modules. Nmap has nse-scripts for Modbus and Scada. Also, tools such as the Metasploit framework should be tested against ICS environments in a lab setting so the ICS industry can understand the implications of their usage. Other research could test other vendors and products besides Opto22 PAC Controllers.

5. Conclusion

In conclusion, this research sheds light on the prevailing cautious approach towards active scanning in Operational Technology (OT) environments, emphasizing these practices' potential risks and consequences. Through a comprehensive series of experiments conducted in a controlled lab environment, the impact of various scanning techniques on the stability and reliability of an Opto22 SNAP-PAC-R1 controller was explored, serving as a representative of modern Industrial Control Systems (ICS).

The results confirmed the historical concerns, showing that indiscriminate active scanning can disrupt OT systems, leading to communication errors, increased latency, and occasional loss of control. However, the findings also highlight a critical distinction: while active scanning can pose challenges, it is not inherently perilous. Instead, with careful consideration and the application of safe scanning practices, the risks associated with security scans can be significantly mitigated.

One pivotal observation is the adaptability of OT systems. Despite disruptions caused by active scans, these systems exhibited a remarkable ability to recover, resuming normal operations once the scanning ceased. Given the suitable precautions and methodologies, it is possible to conduct security scans in OT environments without causing permanent harm or compromising the reliability of critical processes.

Looking forward, the implications of this research extend beyond the lab environment. It underscores the urgency for vendors and manufacturers to recognize the vulnerabilities in their systems and work towards building resilience and security measures into their products. Furthermore, it encourages the security community to explore and develop safer scanning techniques tailored to OT and ICS environments' unique requirements and challenges.

This study challenges the status quo, advocating for a nuanced perspective on active scanning in OT security. While exercising caution and employing best practices remains crucial, it is equally important to acknowledge that active scanning can be a valuable tool for enhancing OT security. By advancing security practitioners' understanding of safe scanning practices and fostering collaboration between security experts and OT vendors, security practitioners can work towards a more secure and resilient future for critical infrastructure and industrial processes.

References

Bodungen, C. E., Singer, B. L., Shbeeb, A., Hilt, S., Wilhoit, K. (2017). *Hacking Exposed Industrial Control Systems: ICS and SCADA Security Secrets & Solutions.* McGraw-Hill Education.

CISA.gov. (n.d.). Cybersecurity Alerts & Advisories | CISA.gov. Retrieved 9/24/2023 from https://www.cisa.gov/news-events/cybersecurity-advisories?f%5B0%5D=advisory_type%3A95

Duggan, D., Berg, M., Dillenger, J., Stamp, J. (2005). *Penetration Testing of Industrial Control Systems.* Sandia National Laboratory, SAND2005-2846P.

Gnebbia. (n.d.). ICS/SCADA Security. In Nmap Tutorial. Retrieved 9/29/2023 from https://github.com/gnebbia/nmap_tutorial/blob/master/sections/ics_scada.md

Nation Institute of Standards and Technology (NIST). (2018). *Special Publication 800-37 Revision 2: Risk Management Framework for Information System and Organizations.* Retrieved 10/15/2023 from https://doi.org/10.6028/NIST.SP.800-37r2

Nmap Project. (n.d.). Port Specification and Scan Order. Retrieved 9/29/2023 from https://nmap.org/book/man-port-specification.html

OpenAI Team. (2022). OpenAI GPT-3.5 Model. Retrieved 9/29/2023, from OpenAI GPT-3.5 model. (Used ChatGPT/OpenAI to format APA citations correctly and to help generate abstract and conclusion of paper)

Opto 22. (n.d.). Firmware for all SNAP PAC products | Opto 22. Retrieved 9/24/2023 from https://www.opto22.com/support/resources-tools/downloads/snap_pac_firmware-zip

Opto 22. (n.d.). SNAP PAC Learning Center Tutorial – version 1.0B | Opto 22. Retrieved 9/24/2023 from https://www.opto22.com/support/resources-tools/downloads/1638_snap_pac_learning_center-zip

Opto 22. (n.d.). SNAP PAC R1 | Opto 22. Retrieved 9/24/2023 from https://www.opto22.com/products/snap-pac-r1

Orebaugh, A., & Pinkard, B. (2011). Nmap in the enterprise: your guide to network scanning. Elsevier.

Josh Tanski, josh@tanski.net

6. Appendix 1 – Details of the Lab Environment

Lab laptop: Windows 10 laptop running VMWare Workstation to host the virtual machines (VMs) used for the experiment, with the wireless NIC connected for Internet access and the wired NIC connected to a Cisco hardware the lab network. A virtual bridge is created in VMWare to bridge the virtual NICs to the wired network.

Cisco switch: Cisco C1000-8T-2G-L compact switch, managed, IOS Lite, used to connect lab network devices.

Opto22 SNAP-PACLC: Test PLC – Opto22 Learning Center including a SNAP-PAC-R1 controller, parts (switches, knobs, lights, buzzer), software, and example files to simulate operating a convenience store (freezer, fuel tank, security system lights, and alarm). The Opto22 SNAP PAC Learning Center Tutorial contains a completed control example used for the experiment. The SNAP-PAC-R1 controller in the lab environment is running the latest firmware, version R10.4D.

HMI: Windows 10 LTSC VM to simulate the human-machine interface (HMI) the operator would use to control the process. HMI ran the Opto22 PACDisplay software to monitor and control the process.

Engineering Workstation VM: Windows VM to simulate the computer a controls engineer or electrician would use to program, configure, monitor, and troubleshoot the controller. The workstation ran the Opto22 PACControl software to program the controller and debug the process and the PACManager software and other utilities to maintain, monitor, or troubleshoot the controller itself.

For both the HMI and Engineering Workstation, Windows LTSC (Long-Term Servicing Channel) was selected for the OS as its functionality and features do not change over time, which is ideal for an OT environment, and does not include features such as

Cortana that are unnecessary for the functionality of an OT environment. All Windows security updates available during this experiment were installed on both systems.

Also, both Windows systems had PAC Project Basic version R10.5002 installed, released on 5/22/2023. PAC Project is Opto22's software suite that includes the software to program the PAC controller, program and run the HMI, and many other utilities.

Scanning and Monitoring VM: A Kali Linux VM was used to run most of the security tools needed for the experiment. Kali Linux is a prevalent and powerful penetration testing platform commonly used in the cybersecurity industry. Kali contains an extensive collection of open-source security tools, including ARP-scan and Nmap.

Greenbone Enterprise TRIAL VM: There were compatibility problems in the latest rolling release of Kali between the open-source OpenVAS vulnerability scanner and the version of the PostgreSQL database server installed. Greenbone is the commercial version of OpenVAS, and their free trial VM download is an analogous replacement to OpenVAS for the experiment.

7. Appendix 2 – Raw Nmap Scan Results

7.1. Nmap Scan 1 Results: Nmap default against controller

```
$ sudo nmap 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 09:34 EDT
Nmap scan report for 10.10.10.11
Host is up (0.0073s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
2001/tcp open  dc
MAC Address: 00:A0:3D:02:CC:BE (Opto-22)

Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
```

7.2. Nmap Scan 2 Results: Nmap all ports against controller

```
$ sudo nmap -p1-65535 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 10:01 EDT
Nmap scan report for 10.10.10.11
Host is up (0.0048s latency).
Not shown: 65498 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
502/tcp   open  mbap
2001/tcp  open  dc
22001/tcp open  optocontrol
22500/tcp open  unknown
22501/tcp open  unknown
22502/tcp open  unknown
22503/tcp open  unknown
22504/tcp open  unknown
22505/tcp open  unknown
22506/tcp open  unknown
22507/tcp open  unknown
22508/tcp open  unknown
22509/tcp open  unknown
```

```
22510/tcp open  unknown
22511/tcp open  unknown
22512/tcp open  unknown
22513/tcp open  unknown
22514/tcp open  unknown
22515/tcp open  unknown
22516/tcp open  unknown
22517/tcp open  unknown
22518/tcp open  unknown
22519/tcp open  unknown
22520/tcp open  unknown
22521/tcp open  unknown
22522/tcp open  unknown
22523/tcp open  unknown
22524/tcp open  unknown
22525/tcp open  unknown
22526/tcp open  unknown
22527/tcp open  unknown
22528/tcp open  unknown
22529/tcp open  unknown
22530/tcp open  unknown
22531/tcp open  unknown
44818/tcp open  EtherNetIP-2
MAC Address: 00:A0:3D:02:CC:BE (Opto-22)

Nmap done: 1 IP address (1 host up) scanned in 20.47 seconds
```

7.3. Nmap Scan 3 Results: Only open ports against controller

```
$ sudo nmap -p21,502,2001,22001,22500-22531,44818 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 10:15 EDT
Nmap scan report for 10.10.10.11
Host is up (0.0077s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
502/tcp   open  mbap
2001/tcp  open  dc
```

```
22001/tcp open  optocontrol
22500/tcp open  unknown
22501/tcp open  unknown
22502/tcp open  unknown
22503/tcp open  unknown
22504/tcp open  unknown
22505/tcp open  unknown
22506/tcp open  unknown
22507/tcp open  unknown
22508/tcp open  unknown
22509/tcp open  unknown
22510/tcp open  unknown
22511/tcp open  unknown
22512/tcp open  unknown
22513/tcp open  unknown
22514/tcp open  unknown
22515/tcp open  unknown
22516/tcp open  unknown
22517/tcp open  unknown
22518/tcp open  unknown
22519/tcp open  unknown
22520/tcp open  unknown
22521/tcp open  unknown
22522/tcp open  unknown
22523/tcp open  unknown
22524/tcp open  unknown
22525/tcp open  unknown
22526/tcp open  unknown
22527/tcp open  unknown
22528/tcp open  unknown
22529/tcp open  unknown
22530/tcp open  unknown
22531/tcp open  unknown
44818/tcp open  EtherNetIP-2
MAC Address: 00:A0:3D:02:CC:BE (Opto-22)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

7.4. Nmap Scan 4 Results: All ports, full CONNECT scan, against controller

```
$ sudo nmap -sT -p1-65535 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 10:20 EDT
Nmap scan report for 10.10.10.11
Host is up (0.021s latency).
Not shown: 65498 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
502/tcp   open  mbap
2001/tcp  open  dc
22001/tcp open  optocontrol
22500/tcp open  unknown
22501/tcp open  unknown
22502/tcp open  unknown
22503/tcp open  unknown
22504/tcp open  unknown
22505/tcp open  unknown
22506/tcp open  unknown
22507/tcp open  unknown
22508/tcp open  unknown
22509/tcp open  unknown
22510/tcp open  unknown
22511/tcp open  unknown
22512/tcp open  unknown
22513/tcp open  unknown
22514/tcp open  unknown
22515/tcp open  unknown
22516/tcp open  unknown
22517/tcp open  unknown
22518/tcp open  unknown
22519/tcp open  unknown
22520/tcp open  unknown
22521/tcp open  unknown
22522/tcp open  unknown
22523/tcp open  unknown
```

```
22524/tcp open  unknown
22525/tcp open  unknown
22526/tcp open  unknown
22527/tcp open  unknown
22528/tcp open  unknown
22529/tcp open  unknown
22530/tcp open  unknown
22531/tcp open  unknown
44818/tcp open  EtherNetIP-2
MAC Address: 00:A0:3D:02:CC:BE (Opto-22)

Nmap done: 1 IP address (1 host up) scanned in 22.16 seconds
```

7.5. Nmap Scan 5 Results: All Ports, full CONNECT scan with delays, against controller

```
$ sudo nmap -sT --scan-delay 1s -p1-65535 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 10:47 EDT

$ sudo nmap -sT --scan-delay 50ms -p1-65535 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 10:54 EDT

$ sudo nmap -sT --scan-delay 10ms -p1-65535 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 10:57 EDT
Nmap scan report for 10.10.10.11
Host is up (0.0015s latency).
Not shown: 65498 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
21/tcp    open     ftp
502/tcp   open     mbap
2001/tcp  open     dc
22001/tcp open     optocontrol
22500/tcp open     unknown
22501/tcp open     unknown
22502/tcp open     unknown
22503/tcp open     unknown
22504/tcp open     unknown
22505/tcp open     unknown
```

```
22506/tcp open     unknown
22507/tcp open     unknown
22508/tcp open     unknown
22509/tcp open     unknown
22510/tcp open     unknown
22511/tcp open     unknown
22512/tcp open     unknown
22513/tcp open     unknown
22514/tcp open     unknown
22515/tcp open     unknown
22516/tcp open     unknown
22517/tcp filtered unknown
22518/tcp open     unknown
22519/tcp open     unknown
22520/tcp open     unknown
22521/tcp open     unknown
22522/tcp open     unknown
22523/tcp open     unknown
22524/tcp filtered unknown
22525/tcp open     unknown
22526/tcp open     unknown
22527/tcp open     unknown
22528/tcp open     unknown
22529/tcp open     unknown
22530/tcp open     unknown
22531/tcp open     unknown
44818/tcp open     EtherNetIP-2
MAC Address: 00:A0:3D:02:CC:BE (Opto-22)

Nmap done: 1 IP address (1 host up) scanned in 748.22 seconds
```

7.6. Nmap Scan 6 Results: All Ports, SYN scan with 10ms delay, against controller

```
$ sudo nmap -sS --scan-delay 10ms -p1-65535 10.10.10.11
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 11:15 EDT
Nmap scan report for 10.10.10.11
Host is up (0.0015s latency).
```

```
Not shown: 65498 closed tcp ports (reset)
PORT      STATE    SERVICE
21/tcp    open     ftp
502/tcp   open     mbap
2001/tcp  open     dc
22001/tcp open     optocontrol
22500/tcp open     unknown
22501/tcp open     unknown
22502/tcp open     unknown
22503/tcp open     unknown
22504/tcp open     unknown
22505/tcp open     unknown
22506/tcp open     unknown
22507/tcp open     unknown
22508/tcp open     unknown
22509/tcp open     unknown
22510/tcp open     unknown
22511/tcp open     unknown
22512/tcp open     unknown
22513/tcp open     unknown
22514/tcp open     unknown
22515/tcp open     unknown
22516/tcp open     unknown
22517/tcp filtered unknown
22518/tcp open     unknown
22519/tcp open     unknown
22520/tcp open     unknown
22521/tcp open     unknown
22522/tcp open     unknown
22523/tcp open     unknown
22524/tcp filtered unknown
22525/tcp open     unknown
22526/tcp open     unknown
22527/tcp open     unknown
22528/tcp open     unknown
22529/tcp open     unknown
22530/tcp open     unknown
22531/tcp open     unknown
```

Josh Tanski, josh@tanski.net

```
44818/tcp open     EtherNetIP-2
MAC Address: 00:A0:3D:02:CC:BE (Opto-22)

Nmap done: 1 IP address (1 host up) scanned in 756.81 seconds
```

7.7. Nmap Scan 7 Results: Nmap default against Windows

```
$ sudo nmap 10.10.10.12-13
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 12:27 EDT
Nmap scan report for 10.10.10.12
Host is up (0.00025s latency).
All 1000 scanned ports on 10.10.10.12 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:D0:01:7A (VMware)

Nmap scan report for 10.10.10.13
Host is up (0.00042s latency).
All 1000 scanned ports on 10.10.10.13 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:9F:DB:06 (VMware)

Nmap done: 2 IP addresses (2 hosts up) scanned in 42.11 seconds
```

7.8. Nmap Scan 8 Results: Nmap all ports against Windows

```
$ sudo nmap -p1-65535 10.10.10.12-13
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 12:33 EDT
Nmap scan report for 10.10.10.12
Host is up (0.00016s latency).
All 65535 scanned ports on 10.10.10.12 are in ignored states.
Not shown: 65535 filtered tcp ports (no-response)
MAC Address: 00:0C:29:D0:01:7A (VMware)

Nmap scan report for 10.10.10.13
Host is up (0.00028s latency).
All 65535 scanned ports on 10.10.10.13 are in ignored states.
Not shown: 65535 filtered tcp ports (no-response)
MAC Address: 00:0C:29:9F:DB:06 (VMware)
```

```
Nmap done: 2 IP addresses (2 hosts up) scanned in 2671.76 seconds
```

7.9. Nmap Scan 9 Results: All ports, full CONNECT scan, against Windows

```
$ sudo nmap -sT -p1-65535 10.10.10.12-13
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 13:21 EDT
Nmap scan report for 10.10.10.12
Host is up (0.00015s latency).
All 65535 scanned ports on 10.10.10.12 are in ignored states.
Not shown: 65535 filtered tcp ports (no-response)
MAC Address: 00:0C:29:D0:01:7A (VMware)

Nmap scan report for 10.10.10.13
Host is up (0.00022s latency).
All 65535 scanned ports on 10.10.10.13 are in ignored states.
Not shown: 65535 filtered tcp ports (no-response)
MAC Address: 00:0C:29:9F:DB:06 (VMware)

Nmap done: 2 IP addresses (2 hosts up) scanned in 2673.28 seconds
```