LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-740556

Safe Active Scanning for Energy Delivery Systems Final Report

J. Helms, B. Salazar, P. Scheibel, M. Engels, C. Reiger

October 25, 2017

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Lawrence Livermore National Laboratory

Safe Active Scanning for Energy Delivery Systems: Final Report

Unclassified

September 30, 2017

Disclaimer

1. Executive Summary

The Department of Energy's Cybersecurity for Energy Delivery Systems Program has funded Safe(r) Active Scanning for Energy Delivery Systems, led by Lawrence Livermore National Laboratory, to investigate and analyze the impacts of active scanning in the operational environment of energy delivery systems. In collaboration with Pacific Northwest National Laboratory and Idaho National Laboratory, active scans across three testbeds including 38 devices were performed.

This report gives a summary of the initial literature survey performed on the SASEDS project as well as industry partner interview summaries and main findings from Phase 1 of the project. Additionally, the report goes into the details of scanning techniques, methodologies for testing, testbed descriptions, and scanning results, with appendices to elaborate on the specific scans that were performed. As a result of testing, a single device out of 38 exhibited problems when actively scanned, and a reboot was required to fix it. This single failure indicates that active scanning is not likely to have a detrimental effect on the safety and resilience of energy delivery systems. We provide a path forward for future research that could enable wide adoption of active scanning and lead utilities to incorporate active scanning as part of their default network security plans to discover and rectify rogue devices, adversaries, and services that may be on the network. This increased network visibility will allow operational technology cybersecurity practitioners to improve their situational awareness of networks and their vulnerabilities.

Table of Contents

2. Introduction

Cyberattacks on critical infrastructure (CI) have been on the rise in the last several years. In the U.S., attacks on energy infrastructure in particular are increasing in number. In 2015 and 2016, respectively, there were 295 and 290 cyber incidents directed toward critical infrastructure, an increase from 245 incidents reported in 2014.[1] Of the reported incidents in 2015 and 2016, respectively, 46 and 59 targeted the energy sector, demonstrating an almost five percent increase in attacks.[2] In addition, the first confirmed cyberattack to cause a power outage occurred in December of 2015. In a premeditated attack, adversaries used malware and remote access to overwrite firmware in multiple substations in Ukraine, which left 80,000 customers without power for six hours.[3] Another cyberattack on Ukraine's power grid in Kiev occurred in December 2016, when unknown hackers took over computers at an electricity control center,[4] resulting in a 200MW load loss in Kiev. These attacks demonstrated that the cyber threat to the power grid is a reality. It is worth noting that power restoration after such an attack in the U.S. would likely take much longer due to heavy reliance on automated systems.

Cybersecurity of industrial control systems (ICS), specifically energy delivery systems, poses a unique challenge, as traditional ICS networks were not designed to be connected to the Internet. They have historically been air-gapped from the corporate/business network and the Internet. As a result, cybersecurity measures such as authentication and encryption (which are typically employed in information technology (IT) networks) were not perceived as necessary and are not inherent to the design of ICS networks. However, cost-driven trends like the Industrial Internet of Things (which refers to the use of the Internet of Things in industrial control and manufacturing) and an abundant proliferation of cloud-based applications and services resulted in ICS networks that are commonly connected to corporate IT infrastructures and are therefore no longer standing in isolation. Thus, they are exposed to cyber threats. IT and ICS networks are fundamentally different, and because of these differences, it is necessary to gain a full understanding of how deployed cybersecurity measures affect ICS network systems.[5,6]

IT and ICS networks differ in that ICS networks can have a direct impact on the physical world and cause physical reactions. ICS devices control real-time industrial and physical processes in energy delivery, manufacturing, transportation, and chemical infrastructure systems. As such, their efficiency, reliability, integrity, availability, and safety requirements can be much higher

[1] https://ics-cert.us-cert.gov/sites/default/files/Annual_Reports/Year_in_Review_FY2016_Final_S508C.pdf
[2] NCCIC/ ICS-CERT Year in Review", Department of Homeland Security, FY15
[3] https://ics.sans.org/blog/2016/01/09/confirmation-of-a-coordinated-attack-on-the-ukrainian-power-grid
[4] https://www.technologyreview.com/s/603262/ukraines-power-grid-gets-hacked-again-a-worrying-sign-for-infrastructure-attacks/
[5] http://www.computerweekly.com/news/2240232680/Industrial-control-systems-What-are-the-security-challenges
[6] https://www.helpnetsecurity.com/2016/04/13/ics-network-attacks/

than in IT networks, where the primary goals typically include business applications, data transfer, storage, and person-to-person communications. Traditional IT security measures can come into conflict with the safety, efficiency, and availability requirements of ICS networks. Many ICS processes and functions are time critical and cannot tolerate the delays that are typical for IT networks. Rebooting an ICS device or installing a software update must be planned carefully—sometimes weeks in advance—to avoid outages, short-term unavailability, and other negative impacts on the system. While the risk-management focus of IT networks is data integrity and confidentiality, the main concerns for ICS are safety and availability, including preventing injury and loss of life, improving and/or protecting public safety and confidence, equipment damage, etc.[7,8]

When it comes to cybersecurity considerations, it is important to note that different infrastructures will have different priorities with respect to ICS performance requirements. In particular, the highest priorities for energy delivery systems (EDS) are availability and time-criticality requirements, as power systems must always operate to meet the power demands of the customers and cannot tolerate delays. Furthermore, EDS components are limited in computing resources and may not have enough memory, CPU power, or network throughput to support traditional IT cybersecurity measures. Finally, EDS components are distributed over wide geographic areas and can be physically accessed and tampered with. Clearly, a cyberattack can have highly negative impacts on these metrics and cybersecurity is imperative; however, before deploying cybersecurity measures, we must ensure that they themselves will not impact the performance of the EDS.

Network scanning is a commonly used security practice in IT networks. It can be passive, where the scanner only observes the traffic that is generated within the network, or it can be active, where the scanner generates traffic for the purpose of probing the devices on the network and gaining a more thorough understanding of network topology and vulnerabilities. With some exceptions, active scanning is not a standard cybersecurity practice used by EDS operators. It is considered risky, as there are some reports of Supervisor Control and Data Acquisition (SCADA) devices and controllers malfunctioning as a result of its use. Unfortunately, by resisting active scanning as a standard cybersecurity practice, EDS operators lose an important tool for discovering network and system vulnerabilities and rogue devices on their network. Passive scanners can only provide information about devices that are active at the time of the scan, and provide no insight or even enumeration of devices and hosts that are idle. Active scanning provides deeper insight into a network's security posture and provides more thorough coverage by scanning all devices present on the network, regardless of whether they are active or idle at the time of the scan.[9,10,11] Additionally, active scanning mimics how a potential attacker would probe and see the network, offering insight into an attacker's path of least resistance.

---

[7] "Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security," NIST Special Publication 800-82, Keith Stouffer, Joe Falco, Karen Kent, September 2006

[8] https://www.isa.org/standards-and-publications/isa-publications/intech-magazine/2014/may-jun/features/cover-story-top-ten-differences-between-ics-and-it-cybersecurity/

[9] "Blended Security Assessments: Combining Active, Passive and Host Assessment Techniques," Renaud Deraison, Ron Gula, Tenable Network Security, May 2011

[10] http://www.techrepublic.com/article/solutionbase-take-advantage-of-vulnerability-scanning-tools-to-increase-network-security/

[11] "Techno Security's Guide to Securing SCADA: A Comprehensive Handbook on Protecting the Critical Infrastructure," Jack Wiles and Ted Claypoole, Syngress, Jule 2008

Lawrence Livermore National Laboratory (LLNL) was funded by the Department of Energy's Cybersecurity for Energy Delivery Systems under the Safe(r) Active Scanning for Energy Delivery Systems (SASEDS) project to investigate and document any potential negative impacts that active scanning can have on energy delivery systems. The project was also supported under two subcontracts by Pacific Northwest National Laboratory (PNNL) and Idaho National Laboratory (INL). Both PNNL and INL provided EDS testbed environments and subject matter expertise.

To understand and analyze the impact of active scanning on EDS and execute on SASEDS objectives, the following actions were performed:

1. A detailed survey of reported incidents in literature
2. Interviews with industry partners and energy delivery system operators on their current practices with respect to active scanning
3. Active scanning experiments in three EDS testbeds
4. Analysis of experiment results
5. Collaboration with industry partners to develop a path forward for future research in the area of active scanning for EDS

The rest of the document is organized as follows. In Section 3, we provide an overview of active scanning methodologies and tools. Section 4 focuses on a summary of SASEDS' Phase 1 Task 1 Report. Section 5 describes the experimentation plan. Section 6 gives an overview of the three testbeds used in this project and summarizes experimentation results and analysis. Section 7 defines the path forward for future research in active scanning for energy delivery systems and contains our concluding remarks.

We note that the information in sections 1–4 has been included in SASEDS' Phase 1 Task 1 Report and experimentation plan, but is also included here for context and completeness.

3. Overview of Network Scanning and Network Scanning Tools

Network scanning is an information gathering technique used for assessing the security of a network and the hosts on that network. Network scanning maps internet protocol (IP) addresses to live hosts (this process is known as a ping sweep), discovers services running on open ports (port scanning), and discovers known vulnerabilities of available systems on the network. Gathered information can be compiled for analysis and used to detect noncompliant (e.g. unpatched or poorly configured) or unknown/undesired systems (e.g., if a user plugged a wireless access point into the network).

Network scanning can be categorized as passive or active. Passive scanners learn about the network by observing traffic over a link in the network. They extract information about systems by inspecting the packets normally generated within the network without generating any traffic themselves. Active scanners query systems in the network—they generate traffic to probe the systems so that they can learn about the network.

Passive and active scanners partially intersect in terms of what they can determine: IPs in use, services (e.g. HTTP, FTP, SSH) running on systems, service version, and configuration information. Passive scanners can gather additional data, e.g. which systems are actively communicating with each other and what information they are exchanging. Passive scanning is typically used in a continuous monitoring mode and provides constant insight into the network state.

Passive scanners are constrained in that they only get information from packets that transit the points they are observing. Typically, there is a limited number of observation points because examining network traffic can be resource-intensive and requires hardware investment. Therefore, some communication pathways are not covered. Furthermore, passive scanning will overlook systems that are present on the network but not actively communicating with other devices.

Active scanning is not constrained in the same manner. An active scanner can be deployed anywhere on a network and collect information from any other system in the network. Active scanners generate network packets and probe the network hosts. They analyze the response from the probes and compare the packets that would be generated by specific operating systems and network services. This allows the scanner to identify the host, operating system, ports, and services that are running on the ports as well as their states. Unlike passive scanners, which can only monitor “live” hosts, active scanners can probe any host on the network, thereby offering better coverage. Active scanning offers an attacker’s view of a network, which can help system administrators identify and eliminate the network’s path of least resistance. We note that while passive scanning is typically done in a continuous manner, active scanners offer a snapshot of a

network at a moment in time. Therefore, a combination of continuous passive and periodic active scanning can offer a more comprehensive security assessment than either active or passive scanning alone. [12,13,14]

Deciding whether to use credentials when configuring an active scan is essential. Credentialed scans can gather the same information from systems as uncredentialed scans, but differ in their approach: they log into target systems and run commands on them to learn about their configurations. For example, to learn about open ports on a Linux system, a credentialed scanner can (once logged in) run the `netstat` command.[15] Logging into systems also allows credentialed scans to gather details that uncredentialed scans cannot, including precise service version information, non-networked software, and comprehensive listings of installed patches. Credentials allow for a more accurate scan and require less bandwidth; however, credentialed scanning may not always be available for all devices in the network, as creating scanner-specific login credentials requires upfront administrative effort.

3.1 Network Scanning Tools Overview

The majority of network scanning tools are passive, active, or hybrids of passive and active scanning. Here we give a brief overview of representative tools from all three categories.

3.1.1 Passive Scanners

Passive OS fingerprinter (pof)[16] is a passive network scanner that identifies hosts with incidental TCP/IP communications. pof is a general IT network scanner (not targeted for ICS networks) and is advertised to have the following capabilities:

- "Highly scalable and extremely fast identification of the operating system and software on both endpoints of a vanilla TCP connection—especially in settings where Nmap probes are blocked, too slow, unreliable, or would simply set off alarms.
- Measurement of system uptime and network hookup, distance (including topology behind NAT or packet filters), user language preferences, and so on.
- Automated detection of connection sharing / NAT, load balancing, and application-level proxying setups.
- Detection of clients and servers that forge declarative statements such as X-Mailer or User-Agent."

pof was developed and copyrighted by Michal Zalewski and is free to use.

Sophia[17], developed by Idaho National Laboratory (INL), is a passive ICS network-specific scanner with intrusion detection system (IDS) capabilities. Sophia monitors communication in

[12] "Technical Guide to Information Security Testing and Assessment" NIST Special Publication 800-115, Karen Scarfone, Murugiah Souppaya, Amanda Cody, Angela Orebaugh, September 2008
[13] "Eliminating Cybersecurity Blind Spots: Challenges for Business," Tenable Network Security, April 2015
[14] http://www.techrepublic.com/article/solutionbase-take-advantage-of-vulnerability-scanning-tools-to-increase-network-security/
[15] https://sectorsecuritysolutions.wordpress.com/2012/10/09/vulnerability-scans-vs-credentialed-scans/
[16] http://lcamtuf.coredump.cx/p0f3/
[17] http://web.ornl.gov/info/news/pulse/no373/story3.shtml

static networks (networks whose communication patterns are fairly fixed) and alerts the operators to any anomalous or unusual conversations, allowing them to determine whether the system is being compromised. The Sophia tool is commercially available through NEXDEFENSE (http://www.nexdefense.com).

GRASSMARLIN[18] is a passive ICS network-specific scanner developed by the National Security Agency (NSA) and is provided as an open-source tool on GitHub. As stated in a whitepaper describing GRASSMARLIN, the objective of this tool is to "provide IP network situational awareness of ICS and SCADA networks to support network security[,] [and] passively map, and visually display, an ICS/SCADA network topology while safely conducting device discovery, accounting, and reporting on these critical cyber-physical systems."

3.1.2 Active Scanners

Nmap (Network Mapper)[19] is an uncredentialed active network scanner for IP networks. It determines open network ports (TCP/UDP), running services, and operating systems (and other network configuration details). Nmap also provides a library of scripts for collecting service-specific configuration details (e.g. from NetBIOS, as well as a few ICS protocols like Modbus). It is free to use, popular, and well-documented.

3.1.3 Hybrid Scanners

Nessus is the most popular hybrid network scanner.[20] Nessus was developed by Tenable Network Security and is free of charge for personal use.[21] Nessus scans for known vulnerabilities, misconfigurations in the network, default passwords, denials of service against the TCP/IP stack, and preparation for Payment Card Industry Data Security Standard (PCI DSS) audits.

NeMS (Network Mapping System) is a hybrid scanner developed by Lawrence Livermore National Laboratory (LLNL) to document network topologies of IP networks. NeMS is designed and configured to minimize disruptions and impacts on the target operational network, and requires minimal intervention by network security staff. Similarly, it is designed to reduce the potential for overburdening the target network and thus can be configured to have no effect on the usability of network services. NeMS includes a sophisticated and demonstrated active scanning control mechanism that is configurable and can be tailored to specific user needs. Users can choose the appropriate active probes that are best for the specific environment, and/or specify the rate and areas of the network to scan to minimize impacts of scanning on operations. The analysis capability in NeMS allows for data fusion from active and passive scanning, which can potentially improve the ability to monitor and secure networks. The NeMS tool is free for government use and a license can be obtained for commercial use.

[18] https://github.com/iadgov/GRASSMARLIN/blob/master/GRASSMARLIN.pdf
[19] https://Nmap.org/
[20] https://en.wikipedia.org/wiki/Nessus_(software)
[21] http://www.tenable.com/products/nessus-vulnerability-scanner

4. Summary of Main Findings from Literature Survey and Industry Partner Interviews

Although active scanning is more comprehensive, it comes with inherent risk: by interacting directly with devices, it may interfere with their behavior. The SASEDS project was primarily motivated to investigate why active scanning for ICS networks has an extremely negative reputation and is perceived as having a detrimental impact on the performance and safety of the control devices being scanned. The first step in this project was to conduct a detailed literature review and collect as much information as possible to understand the situations in which active scanning caused negative impacts. Additionally, industry partners were engaged to gain an understanding of current industry practices and to determine whether active scanning is currently being used in testing or operational environments. This information was used to inform the design of the experiments that were performed in this project. The literature survey and industry interview summaries are covered in detail in the SASEDS Phase 1 Task 1 Report. For completeness, we present a summary of our main findings:

Finding 1: A majority of papers reporting issues in ICS networks caused by active scanning reference "Penetration Testing of Industrial Control Systems," published by Sandia National Laboratories in 2005.[22] This publication describes three incidents of equipment failure and malfunction caused by active scanning. The only other source offering additional information about the three incidents described is the RISI database.[23] According to the database, two of the incidents occurred in 1998, and the third occurred in 2004.

Finding 2: The National Institute of Standards and Technology (NIST), SANS Institute, Department of Homeland Security (DHS), National Security Agency (NSA), and Department of Energy (DOE) have all issued documents warning of the dangers of active scanning and recommend that active scanning be used only in non-production environments. The National Electric Sector Cybersecurity Organization Resource (NESCOR) cautions against active scanning, but does not explicitly prohibit it, and warns that systems predating 2005 have a higher likelihood of experiencing any resulting adverse effects.

Finding 3: Active scanning is routinely performed by both of the EDS industry partners that were interviewed about active scanning practices as part of this effort. Both partners reported some issues resulting from active scanning, but none of a magnitude that would indicate that active scanning is not safe for use in EDS networks.

[22] "Penetration Testing of Industrial Control Systems." D. Duggan, M. Berg, J. Dillinger, J. Stamp. Sandia National Laboratory, SAND2005-2846P, March 7, 2005.
[23] http://www.risidata.com/About/How_It_Works

5. Scanning Experiments Overview

The experiments described in this section were used to evaluate different types of uncredentialed active network scanning approaches to determine whether specific scanning techniques or configurations could adversely affect EDS components or functionality, and if so, to what degree. The specific objectives were:

- Evaluate EDS impact with a range of active scans (from lightweight/minimally invasive to invasive)
- Categorize safe versus unsafe scanning configurations
- Assess and document effects resulting from scanning EDS environments

The experiments were conducted in three independent EDS testbeds, located at Pacific Northwest National Laboratory, Idaho National Laboratory, and Lawrence Livermore National Laboratory. The primary network scanning tool used in these experiments was NeMS, developed at LLNL.[24] A small subset of intensive, per-target scans were performed using Nmap, a widely used open-source utility. All scans were uncredentialed and used Network (ISO Layer 3) and Transport (ISO Layer 4) protocols. In particular, two scanning configuration types were evaluated:

- Industry practice scans (considered benign, used by industry partners on production systems)
- Invasive scans (considered potentially disruptive to EDS equipment and systems)

During and after each scan using NeMS, targeted devices were evaluated for functionality and any overall effects on the operational environment.

5.1 Industry Practice Scans

Industry practice scans are scans identified through our EDS industry partner interviews, and are considered by the partners to be safe for use on EDS networks. There are two types of industry practice scans—one uses uncredentialed port scans to enumerate open Transmission Control Protocol (TCP) ports, and the other uses common TCP service requests to identify network servers.

As an example of an industry practice port scan, the least invasive type is the TCP SYN scan, used for TCP server port mapping. TCP is the fundamental Internet communications protocol, and specific ports are assigned by the Internet Assigned Numbers Authority (IANA) for dedicated uses (e.g., HTTP is assigned to TCP port 80).

[24] https://ipo.llnl.gov/technologies/nems

TCP uses a three-part handshake protocol to establish a connection. The handshake begins when the client chooses a destination port and sends a packet with the SYN flag (a control bit) set. If a service is running on the target port, the server will record the client's intent and send a response with the SYN and ACK flags set. The handshake is completed when the client sends a response with the ACK flag set. In a TCP SYN scan, the scanner (as the client) sends the initial SYN packet in the handshake, but does not send an ACK packet to complete the handshake. This allows the client to learn whether the port is open without establishing a connection with the server. The server must keep track of clients that have started a handshake, but even though the scanner does not complete the handshake, the server will drop this extra state after a timeout period. Because it does not establish a connection, uses few server resources, and follows standard (if not typical) protocol behavior, it was expected that a TCP SYN scan would have negligible impact on device functionality.

Error! Reference source not found. shows which TCP ports were included in the industry practice scans. The ports that were included in this scan were chosen based on the services they run, as they were more likely to be open. The first column identifies the TCP port, the second column identifies the service registered to run on the port, and the third column specifies the type of service. We define four types of services: common, EDS, Operating System (OS), and database.

- **Common** services are routinely used on many ICS systems; they are not specific to EDS devices, but EDS devices may utilize these services
- **EDS** services are specifically intended for use in EDS environments (in cases where the protocol may be used more generally in other ICS applications, the category is "EDS/ICS")
- **OS** services are associated with operating system functions
- **Database** services are associated with database software

Table 5.1.1 Industry Practice Port Scanning Targets

TCP Port	Registered IANA Service	Service Type
21	FTP file transfer protocol	Common
22	SSH secure shell	Common
23	Telnet	Common
25	SMTP simple mail transfer protocol	Common
80	HTTP hypertext transfer protocol	Common
102	Siemens S7/MMS/ICCP/IEC 61850	EDS/ICS
135	Microsoft RPC/OPC	OS (Windows)/EDS
137	NetBIOS (name service)	OS (Windows)
139	NetBIOS (session service)	OS (Windows)
443	HTTPS	Common
445	Microsoft Active Directory	OS (Windows)
502	Modbus TCP	EDS/ICS
1024-8	N/A (Windows Dynamic Ports)	Common

Table 5.1.1 Industry Practice Port Scanning Targets

TCP Port	Registered IANA Service	Service Type
1521	Oracle DB Listener	Database
4712	C37.118	EDS
4840	OPC	EDS
5432	PostgreSQL	Database
20000	DNP3 distributed network protocol	EDS
49152-6	N/A (IANA Dynamic Ports)	Common

IANA has designated ports 49152 and above as dynamic or private.[25] These ports cannot be registered with IANA, may be used by applications for temporary or private use, and are most likely to be open. Earlier Windows versions use ports 1024–5000 as dynamic ports.

The other industry practice scans were port scans that detect and identify active servers. In this network mapping scan, NeMS sends four common service requests to a candidate IP address: an ICMP echo request (ping), an ICMP timestamp request, a TCP SYN packet to port 443, and a TCP ACK to port 80. The first three scans use standard protocol functionality, while the fourth (TCP ACK) is not part of the HTTP protocol but in practice should be disregarded by the server and not cause any problems.

As with the industry practice port scans, NeMS server scans use simple network mapping and server handshake initiation messages. These simple exchanges on known ports were not expected to cause disruption to a production EDS device. In these experiments, the targeted devices and EDS network were monitored to determine whether there was any degradation in device or overall system functionality or performance.

5.2 Invasive Scans

The invasive scans were designed to assess how the EDS would respond to a set of more aggressive scans and evaluate what impact these scans would have on the EDS devices in a testbed environment. Three types of invasive scans were employed: service detection, OS type detection, and capacity tests.

For port service detection scans, the scanner attempted to establish a connection and query the port to determine what service was running on it. IANA port registrations are used to link commonly used services with known port addresses. However, some ports may support multiple services (e.g., port 50: XNS or RAP), some services may be offered on a variety of ports (e.g., NETRJS on ports 71–74), and non-standard services may utilize any assigned or unassigned port. In addition, IP scans include both TCP and UDP (connectionless User Datagram Protocol). TCP messages (such as SYN) are used to initiate a connection, while UDP services will typically only respond to a known set of expected (application-specific) messages. If the service linked to

[25] These ports may be used by protocols that are not registered with IANA, although registered protocols (e.g. FTP) may use these ports when setting up secondary connections

a port is unknown, the scanner must guess what service is active, typically by sending a message relevant to a service registered to that port.

For both UDP and TCP services, this technique can crash protocol handlers. An unsupported message type or format can trigger an exception within the software, especially when the software is not designed to handle the possibility of unexpected messages from unknown sources. Older applications are particularly vulnerable when they are designed for deployment on private, dedicated networks with known interactions. Applications that do not implement protective measures to detect and reject anomalous events are at risk of failure when subjected to these types of invasive scans. This set of scanning tests included the following UDP ports:

Table 5.2.1 UDP port targets for invasive scanning

UDP Port	Registered IANA Service	Service type
53	DNS	Common
69	TFTP	Common
123	NTP/SNTP	Common
161	SNMP v2/v3	Common
1434	MS SQL Monitor	Database
4713	C37.118	EDS
17185	Wind DeBug (VxWorks)	OS (VxWorks)
49152-6	N/A (IANA Dynamic Ports)	Common

NeMS was also used for OS detection scans. In addition to mapping network hosts and determining open ports and services, NeMS can be used to estimate the operating system of discovered systems. This involves sending a combination of TCP SYN packets, ICMP echo request packets, and UDP packets. These packets vary the header options and attempt to match the response to the distinct TCP/IP implementations provided by different operating systems. These tests are expected to be low risk, but the range of options could potentially disrupt a proprietary or insufficiently-tested application.

As with the TCP scans described previously, device and system performance was monitored during and after invasive UDP scans to determine whether the scans were disruptive.

While NeMS allows for setting multi-target parallelism (where multiple systems are scanned concurrently), it does not support intensive target-specific scanning. In particular, it does not allow configuration of the number of packets per second that are being sent to a single target IP, or of the number of service queries that are attempted for a given open port. It is important to perform capacity tests to reveal vulnerabilities to denial-of-service conditions or attacks. For example, sending an unexpectedly large number of SYN requests can consume server resources to the extent that the system becomes unresponsive to legitimate transactions (this is called a SYN flood denial-of-service attack).

To determine whether intensive per-target scanning could negatively impact an EDS system, invasive scan tests utilized manual Nmap commands to implement aggressive scanning and service detection scenarios. Nmap was configured to perform high-frequency scans across a

much larger range of ports (up to all 65K TCP ports, with a smaller set of UDP ports due to the additional resource overhead from UDP scanning). Nmap can also adjust the level of scan aggressiveness for service detection by increasing the number of unknown service query attempts.[26]

Nmap was also used to execute Nmap scripting engine (NSE) scripts that can be used to explore EDS network devices and provide resource information for a number of relevant ports and services:

- **tftp-enum.nse** (UDP 69): Trivial File Transfer Protocol (TFTP), used to store and retrieve files, does not provide a list of available files; this script guesses a number of common file names to determine which files are available on the server
- **s7-info.nse** (TCP 102): collects device information from Siemens S7 Programmable Logic Controllers (PLCs), including hardware model and serial number
- **modbus-discover.nse** (TCP 502): A Modbus TCP server can act as an intermediary for multiple Modbus slaves; this script can identify Modbus slaves accessible via a Modbus TCP server and collect device model information
- **http-enamines** (TCP80/443): identifies web server implementation by checking for associated resource URLs and limited content information (also known as fingerprints; this is not specific to ICS/EDS devices but does include a fingerprint for the Siemens Simatic S7 PLC
- **wdb-version.nse** (UDP 17185): detects Wind DeBug, a service associated with VxWorks (real-time OS used on some EDS devices)
- **ntp-monlist.nse** (UDP 123): some implementations of NTP support a request for a list of IPs that recently communicated with the service; this script can retrieve IPs along with the nature of the transaction role (server, client, or peer).

5.3 Assessing Impact

A performance baseline for the expected device measurements and system behaviors was first established using the testbed system. After each scanning session, any alarms or exceptions were noted and the scanned devices and overall EDS system were evaluated against the baseline to determine whether the scanning had any adverse impact on devices or overall system performance. The tests were intended to confirm that no unexpected physical activity took place, and to determine whether the system and devices continued to respond to requests, operate within specified time constraints, and report sensible measurements. Any detected impact on the devices was classified into High, Medium, Low, or No Impact, based on the following criteria:

- **High Impact**: As a result of the scan, a device is in a state that requires firmware reload or is unrecoverable. This also includes any case in which a service running on the device fails to respond or function within expected time constraints or in which any undesired/unexpected physical action (such as triggering a relay) takes place.

[26] Although it will not continue sending probes to a given port once it has identified the associated service

- **Medium Impact**: As a result of the scan, a device is temporarily taken out of normal operational mode and assumes an inoperable state. For example, a medium impact would involve a PLC reset such that the controlled process is disrupted.
- **Low Impact**: As a result of the scan, a momentary disruption of communications occurs, causing a loss of situational awareness for the operator or monitoring processes, but not impacting the process operation. Exceptions include situations in which data are shared between PLC/RTU devices for control purposes, and control actions are disrupted.
- **No Impact**: No detectable operational effects.

5.3.1 Assessing Impact—Methods

Scanning effects were evaluated on the device and system level using individual per-device checks and integration testing. Per-device checks are binary yes/no evaluations of all individual device functions, where devices may or may not be attached to an EDS. Integration tests evaluate deployed devices in an EDS from a systems-level approach, where the overall behavior and performance of the system (due to the presence and interactions of multiple devices) can be measured and evaluated. For example, a test might evaluate the propagation of an event through the system, or the sequencing of actions as part of a larger control process.

A functional EDS system can be comprised of a variety of devices, so testing included a variety of representative devices, from IP-enabled intelligent electronic devices (IEDs) to control station systems that support EDS management (e.g., an HMI or data historian). Devices used across the three testbeds varied in their application, functionality, age, vendors, and operating systems.

5.3.1.1 Per-Device Checks

The per-device approach (also referred to as functional testing) for evaluating scanning impact exercises and monitors each of the essential functions of a device before, during, and after scanning. Due to testbed limitations, the device may not be connected and/or configured as it would be in an actual operational deployment. For example, while a device might be connected to the network via an Ethernet interface, other network interfaces (such as serial control lines) might be disconnected. In addition, communications not related to primary device functions might not be enabled (such as device control commands in response to network activity), and the device might not be attached to an EDS if it can be evaluated in stand-alone mode.

5.3.1.2 EDS Integration Testing

In an integration test, devices were connected to an EDS testbed and configured to interact as they would in a production environment. While some of the components were simulated and some of the peripheral support systems might have been absent (such as a data historian), actual devices functioned as if part of a real EDS. Running scans against this simulated EDS system allowed for the effects of scanning to be detected in a number of ways:

- monitoring of the system for unexpected behaviors (e.g., EDS bus voltage fluctuations)
- detection of anomalies through the evaluation of trends by human experts reviewing system measurements on HMI displays

- automated detection and reporting of events that exceeded control point limits

By creating a controlled testbed environment, integrated testing introduced the potential for revealing many types of faults, failures, and glitches that otherwise might have evaded detection:

- physical effects on devices (e.g., unintended control actions) can appear as deviations from expected system behavior that may also affect other interconnected devices
- device failures that do not create detectable system performance anomalies may be found by omission: the lack of expected measurements or events on HMI displays or in event logs
- component failures may also be detected by directly observing controls, functions, and device states (such as relays or breakers)

All integration tests began with establishing the baseline behavior of the testbed systems. The trends for system variables (e.g., voltage and frequency) were monitored to understand their nominal operational values prior to the application of network scans. When scans were introduced into the system, any changes to baseline readings and expected system performance metrics under otherwise normal operational conditions were attributed to the application of scans.

All three testbeds utilized in this project were set up and configured differently, and while they adhered to the general process described before, additional checks and setup specific to individual testbed requirement were needed. Additional details on all three testbeds are covered in the next section.

6. Experiment Results and Analysis

6.1 PNNL Testbed: Overview and Test Procedures

Testbed 1 included three categories of devices:

1. **Electrical EDS Infrastructure intelligent electronic devices (IEDs)**
   Three different IED vendors with nine IEDs ranging from two to six years old
2. **Gas/Hydrocarbon EDS Infrastructure IEDs**
   Three different IED vendors represented with nine IEDs all purchased in 2016
3. **Legacy devices**
   Legacy devices do not benefit from advances in hardware and software design of newer devices and are perceived as less tolerant to active scans; three legacy devices were included in Testbed 1, their hardware design dating from as early as the 1980s

6.1.1 Electric IEDs: Per-Device testing

The first category of testing is a per-device, functional testing approach that evaluated impact by exercising and monitoring essential device functions during and after scanning.

The per-device testing scanned nine separate devices from three different vendors with varying aggressiveness. The devices were not connected and configured as they would normally be configured. Specifically, no device functionality was linked to another device via the network. All devices were connected on one small local network, but no specific traffic was generated between the devices. The additional checks included monitoring counters for physical input and output switches, analyzing configurable event recording logs, and checking accumulated energy readings from each device after the scans.

6.1.1.1 Pre-Scan

Prior to any network scan, the following tasks were performed:

1. *Ping each IED IP address*: verifies response from IED and proper network setup.
2. *Reset energy registers*: zeros out the energy registers of applicable IEDs prior to receiving more energy from the source.
3. *Reset counter variables*: zeros out the counters used for confirming proper physical I/O operation of applicable IEDs.
4. *Clear event reports*: clears previous events from IED memory; applies to all IEDs.

5. *Start traffic capture recording*: provides a record of network traffic via Wireshark or tcpdump tools.
6. *Check connections between IEDs and OPC server*: verifies that the protocol connections between applicable IEDs and the OPC Server software on EWS are working correctly (DNP3 and IEC61850 MMS).
7. *Connect PMU IEDs to software*: verifies working PMU data connection between applicable IEDs and EWS.
8. *Energize multi-phase source*: supplies 3-phase power to the measurement circuits of applicable IEDs (115 V and 1.5 A per phase with current lagging 30 degrees).
9. *Verify counting relays are working*: audible clicking can be heard when energy-counting relays actuate.

6.1.1.2 During Scan

During any network scan, the following tasks were performed:

1. *Verify PMU connections*: confirms that each PMU connection displays continually updating data.
2. *Exercise IED DNP3 connections*: confirms that the DNP3 data connection is active and working.
3. *Exercise IED IEC61850 MMS connections*: confirms that the IEC61850 MMS data connections are active and working.
4. *Exercise breaker function*: confirms that applicable IEDs are able to serve protective function while being scanned; includes both remote and local (faceplate) trip.

6.1.1.3 Post-Scan

After any network scan, the following tasks were performed:

1. *Stop recording of network traffic*: turns off traffic capture and saves file.
2. *De-energize source*: turns off the multi-channel source that was feeding the measurement circuits of applicable IEDs; stops further incrementing of the counter variables.
3. *Confirm PMU data*: checks the integrity of PMU data connection by showing proper source state change as the source is de-energized in previous step.
4. *Exercise IED DNP3 connections*: confirms that the DNP3 data connection is active and working.
5. *Exercise IED IEC61850 MMS connections*: confirms that the IEC61850 MMS data connections are active and working.

6. *Exercise breaker function*: confirms that applicable IEDs are able to serve protective function while being scanned; includes both remote and local (faceplate) trip.
7. *Examine event reports*: shows any state changes that were previously configured to trip the event recorder; applies to all IEDs.
8. *Get counter variables*: collects and compares the counter variables for all applicable IEDs that are connected to the energy meter serving as energy counting "clock;" all counters on all applicable IEDs should show the same value.
9. *Get energy accumulated*: reads out the energy registers of applicable IEDs; check that each IED reported the same proportion of power, apparent power, reactive power, and power factor (PF).

6.1.2 Electric IEDs: Integrated Testing

The second category of testing is an integrated functional testing approach of a working system of IEDs. Similar to per-device testing, impact is evaluated by exercising and monitoring essential device functions during and after scanning.

The goal of integrated testing is to evaluate scanning impact on a working, interdependent network of IEDs as it would be found in a real EDS environment. Real IEDs are connected to a Real-Time Digital Simulator (RTDS) capable of combining real and virtual IEDs into a dynamic model. The same nine IEDs used in per-device testing were connected to a larger real-time digital-simulation model (RTDS) with virtual IEDs.

Protocols used during integrated testing included IEC61850 GOOSE, IEC61850 MMS, DNP3, and C37.118 (PMU).

Five IEDs (one automation controller, one meter, one PMU/automation controller/RTU, and two protective relays/PMUs) were functionally interconnected via GOOSE messages. One automation controller communicated with the EWS over DNP3. Two relays were connected to the EWS over MMS. Three PMUs were connected to the EWS over C37.118. The same checks used in per-device scans applied here as well.

6.1.2.1 Pre-Scan

The same pre-scan steps as in per-device tests applied here.

6.1.2.2 During Scan

The same steps as in per-devices tests applied during testing. Exercising the breaker functionality here consisted of remotely tripping one relay, which then sent GOOSE messages to trip the other relays automatically and autonomously.

6.1.2.3 Post-Scan

The same post-scan steps as in per-device tests applied here. The additional IEC61850 GOOSE network traffic generated by integrated testing was examined in the captured network traffic logs.

6.1.3 Legacy Device Testing

Testing of legacy devices is not performed with the same complexity as is testing of modern EDS devices. Device functionality (i.e., accurate measurements) and I/O were not checked. Instead, the following limited functional checks were performed:

1. Monitoring for continuous pings as evidence for an operating and responsive Ethernet port
2. Evaluation of front panel controls and display, as applicable—ensured that front panel control and display remained responsive and did not lock up
3. Device event reporting, as applicable
4. Ethernet traffic capture and inspection

Each of the three legacy RTUs were scanned and monitored individually with no other devices on the network. Both Modbus and DNP3 protocols were in use on these devices. Continued response on the Ethernet port and clear event reports (as applicable) was the primary scanning impact criteria.

6.1.4 Gas/Hydrocarbon Device Testing

The fourth category of testing applies to devices used in gas/hydrocarbon EDS infrastructure. Testing of gas devices is not performed with the same complexity as is testing of modern EDS devices. Distinguishing characteristics (functions that a gas SCADA device is designed to perform in the field, i.e., gas flow calculations) and I/O were not checked. Instead, the following limited functional checks were performed:

1. Monitoring for continuous pings as evidence for an operating and responsive Ethernet port
2. Evaluation of front panel controls and display, as applicable—ensured that front panel control and display remained responsive and did not lock up
3. Device event reporting, as applicable
4. Ethernet traffic capture and inspection

There were eight distinct hardware designs in nine separate IEDs. These eight models were tested individually with no other devices on the test network. The two devices of the same model were tested together. Continued response on the Ethernet port and clear event reports (as applicable) were the primary scanning impact criteria.

6.2 PNNL's Testbed: Results and Analysis

No issues, disturbances, or anomalies were observed during tests of legacy devices and gas/hydrocarbon devices. Detailed scripts of tests performed that resulted in no observed malfunction can be found in Appendix A. For electric EDS tests, one device had performance issues. More details about those tests and results are given below.

6.2.1 Per-device Scans—Electric IEDs

In one of the per-device scans, a protective relay ceased all communication over its Ethernet port until it was power-cycled/rebooted. The device did not respond to pings or connection attempts with the vendor's configuration software. We note that this impact is classified as medium, per the previous section.

The scan that caused this failure was an Nmap scan of the entire network of connected IEDs. The script ran four separate Nmap commands with increasingly aggressive options. The scans performed are given below:

The first scan in the script was:

```
nmap -n -sTU -script modbus-discover.nse -script-args='modbus-
   discover.aggressive=true' -script s7-info.nse -script tftp-enum.nse -script
   http-enum.nse -script wdb-version.nse -script ntp-monlist.nse -top-ports
   3000 192.168.1.0/24
```

The second scan in the script was:

```
nmap -n -sTU -script modbus-discover.nse -script-args='modbus-
   discover.aggressive=true' -script s7-info.nse -script tftp-enum.nse -script
   http-enum.nse -script wdb-version.nse -script ntp-monlist.nse -p
   123,102,502,69,80,443,17185 192.168.1.0/24
```

The third scan in the **`multinmap.sh`** script was:

```
nmap -PE -n -sTU -top-ports 1000 -sV -version-intensity 9 -max-retries 1 -T4
   192.168.1.0/24
```

The fourth and final scan in the **`multinmap.sh`** script was:

```
nmap -n -sTU -p T:21,22,23,25,80,102,135,137,139,443,445,502,1024-1028,1521,
   4712,4840,5432,9000,20000,49152-49156,U:53,69,123,161,1434,4713,17185,
   49152-49156 -sV -version-intensity 9 -T4 192.168.1.0.24
```

To better understand the failure mechanism, the device was re-tested numerous times. The following additional scans were performed:

- Test 1a: TCP port 9000 scanned 50 times, lockup of port after 102 ping sequences.
- Test 1b: A repeat of Test 1a, but for 71 scans. No lockups were observed.
- Test 2a: UDP port 17185 scanned 50 times. Lockup of port after four ping sequences.

- Test 2b: Repeat of 2a. Scanned 16 times. No lockup even after approximately 3,000 ping sequences.

- Test 3a: (Command `Nmap –PE –n –sTU –p –T:502,U:17185 –sV –version-intensity 9 –max-retries 1 –T4 –min-rate 300`) Performed 15 scans, no lockup observed.

- Test 4a: A sequence of two commands:

  ```
  nmap –PE –n –sT –p T:502 –sV –version-intensity 9 –max-retries 1 –T4 –min-rate 300
  ```

  and then

  ```
  nmap –PE –n –sU –p U:17185 –sV –version-intensity 9 –max-retries 1 –T4 –min-rate 300
  ```

  ran for 245 scans without locking up.

- Test 5a: Almost identical to 4a, with an additional TCP Nmap scan added at the end so that the sequence was scan TCP 502 -> scan UDP 17185 -> scan TCP 502 again. Eight scans were performed and no lockup was observed.

- Test 6a: Similar to test 5a. TCP port 502 was replaced with TCP port 9000 so that the scanning sequence was scan TCP 9000 -> scan UDP 17185 -> scan TCP 9000. Lockup was observed after first scan of TCP port 9000. Switched to scanning UDP 17185 after 125 ping sequences, then stopped test.

- Test 6b: Repeat of test 6a. Lockup observed early on, but 1820 scans were performed. Lockup did not resolve itself except with a power cycle.

- Test 7a: Similar to 6a and 6b, but the TCP and UDP ports were reversed in the scan sequence, which was scan UDP 17185 -> scan TCP 9000 -> scan UDP 17185. Lockup observed after 322 ping counts.

Based on the additional focus testing, the following test sequence was found to reliably cause the port lockup:

- First scan:

  ```
  nmap –PE –n –sT –p T:9000 –sV –version-intensity 9 –max-retries 1 –T4 –min-rate 300
  ```

  Then:

  ```
  nmap –PE –n –sU –U:17185 –sV –version-intensity 9 –max-retries 1 –T4 –min-rate 300
  ```

6.2.2 Integrated Testing—Electric IEDs

The same tests as described above were repeated in the integrated environment. Again, the same protective relay ceased all communication over its Ethernet port. The device had to be power-cycled to restore communication. We note that this is the only modern device to use Wind River's VxWorks as its real-time operating system (RTOS). However, two of the tested legacy devices were also running the same RTOS and experienced no issues from the scanning, making it unclear whether any other factors contributed to this failure, such as RTOS version or RTOS/hardware combination.

6.3 INL's Testbed: Overview and Test Procedures

Testbed 2 is modeled after an ICS network as part of an enterprise and includes ten devices from various vendors. The systems and components in this environment are entirely physical. The environment includes both a distribution substation and a process system, which is not specific to EDS environments. The following types of devices were part of the scanning experiments:

- Power meters
- Programmable Logic Controllers (PLCs)
- Human Machine Interfaces (HMIs)
- Data historian
- Inter-Control Center Communications Protocol (ICCP) server
- Engineering workstations

None of the devices used in this testbed can be considered legacy devices. The substation was modeled after an electric distribution substation. It consisted of a PLC and two electric power meters, each having a dedicated IP address. The PLC controlled relays that were arranged in a substation configuration with four feeder outputs. The two power meters connected to branches of the substation to measure voltages and currents. The measured values (voltages, currents, power factor, and reactive values) were then made available in a Modbus table specified by the manufacturer.

Process System A included flow meters, variable position control valves, pressure and level indicators, pumps, and tanks; a PLC provided an interface to these sensors and controllers. A HMI screen was developed to allow an operator to monitor the process and control the valves and pumps necessary to create a product. The HMI has an automatic and manual mode: the manual mode provided the operator with stop/start controls for pump, flow valve position, and open/close for a valve; the automatic mode controlled the transfer of fluids between tanks.

Process System B was responsible for receiving product from Process System A and supplying product to the specialty chemical mixing skid. It can receive up to 30 gallons of product and can control the rate that it is received through a variable position valve. When an adequate amount of product existed in the supply tank and an order for product was received from specialty chemical the chemical supply process operator properly aligned the valves and started the pump that delivered the requested amount of product. A real-time totalizer computed the amount delivered to aid the operator in the transfer process.

6.3.1 General Evaluation of Devices Based on Commodity Operating Systems

Many of the EDS devices are implemented as custom applications hosted on commodity operating systems. For those that are, this subsection describes a set of checks that were performed to inspect for anomalies based upon performance parameters embedded in the operating system. This check was performed as a complement to loop checks that were performed per the next subsection.

6.3.1.1 Before Scanning

1. Bring up the performance monitor for Windows Devices (or comparable for non-Windows)
2. Establish trends of device performance, which should include processor and memory utilization, but can include other aspects
3. Start EDS-specific applications
4. Baseline behavior and process

6.3.1.2 During Scanning

1. Monitor performance for variations
2. Document when performance violates performance limits, including the scan command, device affected, and device type

6.3.1.3 After Scanning

1. Restore proper operation (and baseline device configuration) as necessary for follow-on testing
2. If performance issues occur, evaluate a scanning regimen to identify the root cause issue

6.3.2 Loop Checks of EDS Devices

The purpose of an EDS system is to monitor and control physical processes. Therefore, an end-to-end evaluation of anomalies includes monitoring of the human machine interface (HMI) for characteristics that indicate the programmable logic controller (PLC) and other intermediate devices do not show uncharacteristic variations. To quantify evidence of whether devices were affected by network scanning, process operations were in effect that provided a baseline to normal operation. During these operations, traffic between the devices was continuously occurring in real time, with the exception of devices such as the engineering workstation (EWS). Monitoring during the scanning included checking sensor devices (which were expected to have mildly varying values) and ensuring that the overall system maintained desired control. More detailed evaluations can be performed on individual devices, dependent on the manufacturer for proprietary EDS devices specifically, and allow processor utilization and other device characteristics to be evaluated before, during, and after scanning.

As the environment in Testbed 2 has physical processes, these data sources provided an understanding of baseline behavior. However, similar tests could be performed with emulated data. General testing procedure is given below.

6.3.2.1 Before Scanning

1. Start physical process and confirm EDS-specific applications are also started
2. For remote terminal units (RTU), PLCs, and power meters, ensure data is normal and displaying on the HMI, historian, EWS and ICCP channel
3. Select physical data points that establish trends of device performance, which should include several values monitored within each RTU, PLC, and meter
4. Establish observable but statistically significant limits on deviation of monitored data points to quantify an anomaly
5. Baseline behavior and process

6.3.2.2 During Scanning

1. Monitor data points for variations
2. Document when performance violates performance limits, including the scan command and device affected and device type

6.3.2.3 After Scanning

1. Restore proper operation (and baseline device configuration) as necessary for follow-on testing
2. If performance issues occur, evaluate a scanning regimen to identify the root cause issue

6.3.3 INL's Testbed: Results and Analysis

Per section 5, there were two sets of scans in INL's testbed: less invasive scans (using NeMS) and more aggressive scans (using Nmap). Previously described checks were performed to confirm continued operation without impact. All ten scanned devices continued normal operation after both sets of scans. No issues or anomalous behavior were observed. Additionally, both physical processes performed flawlessly with no observable impact.

We note that in this testbed, the device resources were not operating near full utilization, and the bandwidth utilization was low. This might not always be the case in the operational environment, so it is possible that at full utilization active scanning could have a negative impact. This gap was addressed in Testbed 3.

6.4 LLNL's Testbed: Overview and Test Procedures

Testbed 3 is representative of a common configuration of a power transmission substation. The tests included eight devices from the same vendor, with none that would be considered legacy devices. The types of devices included protection relays, current differential relays, real time automation controller, and transformer monitor. The experiments in this testbed had three objectives:

1. Perform individual device scans
2. Perform integrated environment scans with devices near full utilization that would be representative of a high activity production environment

3. If negative impacts were observed in integrated environment scans, repeat the scans with hardware in the loop simulation to assess large-scale impacts of failures caused by active scanning

6.4.1 Individual Device Scans

The individual device scans consisted of two parts. The first step was a full port scan of all devices in the testbed. The second step executed scans described in section 4 on devices that were communicating with each other, but were not configured or set up in the way they would be if they were in an EDS environment. The test procedures followed are described below.

6.4.1.1 Pre-Scan

1. Ping each IP-enabled device
2. Confirm that the SCADA Master is receiving data from all devices

6.4.1.2 During Scan

1. Monitor all device values and event reports

6.4.1.3 Post-Scan

1. Examine event reports
2. Re-perform all pre-scan checks

6.4.2 Integrated Environment Scans

The devices in the integrated environment were communicating with each other through a vendor proprietary communication protocol. They were supplied with the voltages and currents they would see in an EDS environment. The intensity of communication was designed to replicate traffic one would observe in a substation at a time of high activity. The Nmap scans were slightly modified to include a more comprehensive port scan and accelerate the scanning process. The testing procedures are provided below.

6.4.2.1 Pre-Scan

1. Start physical process and confirm EDS-specific applications are also started
2. For remote terminal units (RTUs) and power meters, ensure data is normal and present on the HMI, engineering workstation, and RTAC
3. Select physical data points that establish trends of device performance, which include several values monitored within each RTU and meter
4. Establish observable, but statistically significant limits on deviation of monitored data points to quantify an anomaly
5. Baseline behavior and process

6.4.2.2 During-Scan

1. Monitor data points for variations
2. Document when performance violates performance limits, including the scan command and device affected and device type
3. Observe reportable data within RTAC from other devices, ensuring network connections are still working as expected

6.4.2.3 Post-Scan

1. Restore proper operation (and baseline device configuration) as necessary for follow-on testing
2. If performance issues occur, evaluate a scanning regimen to identify the root cause issue
3. Compare measured values to previously recorded values

6.4.3 Hardware in the Loop (HIL) Scans

Hardware in the loop scans leverage LLNL's power transmission modeling capability, GridDyn[27], to assess large-scale impacts of failures and anomalous behavior resulting from active scanning. For example, in Testbed 1, a protective relay became unresponsive and had to be power-cycled to resume normal operation, but it is unclear what impacts, if any, that would have had on the EDS performance. LLNL's hardware in the loop capability allows for simulating a power grid model in which one of the substations in the model is replaced with the physical testbed.

Any load loss caused by active scanning device failures or malfunctions would be considered a large-scale impact. The checks for this test included measuring any load loss that resulted from the failure as well as monitoring voltage values and capturing whether they were within the normal operational bounds. We note that these tests are valuable and performed only in case a failure in individual device scans or integrated environment scans is observed.

6.5 Testbed 3: Results and Analysis

Per section 5, the scans in LLNL's testbed included two sets of scans: less invasive scans (using NeMS) and more aggressive scans (using Nmap). Previously described checks were performed to confirm continued operation without impact. All eight scanned devices continued normal operation after both sets of scans, even with high activity traffic between the devices. No issues or anomalous behavior were observed. As a result, HIL scans were not performed.

6.6 Experiment Results Summary

Across three different testbeds, 38 devices of varying application, functionality, vendors, age, and operating systems were tested. The scans ranged from basic, non-invasive scans using NeMS to highly aggressive scans using Nmap commands. Only one failure of medium impact was observed, wherein a protective relay became unresponsive and needed to be power-cycled to resume normal operation. No permanent damage has been inflicted on the device. It is unclear what caused this failure, and while it is likely that the RTOS is a factor, other factors could be at play as well. A follow-up with a vendor of this device is recommended to determine a root cause of the device failure.

[27] https://github.com/LLNL/GridDyn

7. Research Path Forward and Conclusion

Based on the findings of both our literature survey and our industry partner interviews, as well as extensive testing across three independent testbeds, we conclude that while there are some documented incidents in which active scanning has caused dramatic negative impacts, it does not deserve the extremely negative reputation it has acquired in literature, standards, and guidance documents. While active scanning is used by industry, cybersecurity experts are fighting an uphill battle with EDS operators who perceive active scanning as something that will negatively impact their operations, or worse, cause permanent damage to equipment.

As part of the last phase of the project, we engaged with industry partners to share our findings and determine the next research steps and objectives that will enable wider adoption of active scanning. One of the main ideas that emerged was that failures resulting from active scanning should be perceived as a vulnerability of the system, rather than as a shortcoming of active scanning. An adversary targeting an EDS system is likely to employ active scanning as part of their reconnaissance strategy. By uncovering and mitigating those vulnerabilities upfront in a non-production testing environment, a more resilient EDS will exist. We do acknowledge that while worthy, this goal is long term and will require a paradigm shift in how operators perceive active scanning.

More tangible steps were also identified:

- To increase the confidence of the community in the safety of active scanning, it would be beneficial to build a database with documentation about specific devices, vendors, configurations, and scans performed and their performance results. This can serve as a template informing which scans are "safe" and under what circumstances. Any failures or anomalous behavior identified should be documented as well, and reported to vendors so that patches and mitigations can be developed and deployed. The specific scan scripts using NeMS and Nmap that should be used to conduct these tests are provided in the appendices.
- Developing or modifying existing active scanning tools to further tailor them for OT devices that recognize OT device signatures, services, and operating systems could also help wider adoption of active scanning in OT environments, as the tools would provide a more comprehensive understanding of the assets present on the network and their potential unintended exposure to the open Internet.
- Characterizing OT environments more effectively, in terms of identifying network times and conditions when active scanning is less likely to cause issues, is another area of research that should be pursued. For example, identifying times of high activity; characterizing processes that push devices to nearly full utilization, when active scanning

could cause too much additional burden; and understanding bandwidth bottlenecks can be used to define the environments in which active scanning is safe (or not safe).

In conclusion, we believe that active scanning is a valuable cybersecurity technique that can offer unique insights into the cybersecurity posture of the EDS operational environment. This project has laid out a foundation that will help debunk some of the negative perceptions of using active scanning in OT environments and enable adoption of active scanning as a recommended standard practice.

Appendix A – PNNL's Testbed Commands

Test 1 utilized Nmap's command line interface with the following options:

```
nmap –PE –n –sTU –top-ports 1000 –sV –version-intensity 9 –T3 <target IP>
```

Test 2 utilized Nmap's command line interface with the following options and scripts:

```
nmap –n –sTU –script Modbus-discover.nse –script-args='modbus-
   discover.agressive=true' –script http-enum.nse –script wdb-version.nse –
   script ntp-monlist.nse –p 123, 102, 502, 69, 80, 443, 17185 <target IP>
```

Test 3 utilized Nmap's command line interface with the following options and scripts:

```
nmap –n –sTU –script Modbus-discover.nse –script-args='modbus-
   discover.agressive=true' script s7-info.nse –script tftp-enum.nse –script
   http-enum.nse –script wdb-version.nse –script ntp-monlist.nse –top-ports
   3000 <target IP>
```

Test 4 utilized NeMS with the **`autorun_mapper.py`** program to scan all devices on the network.

Test 5 is a scan of the entire network of connected IEDs. The script **`multiNmap.sh`** runs four separate Nmap commands with increasingly aggressive options. The script also records network traffic via tcpdump from the virtual machine scanning the network. Wireshark is not used in this test.

The first scan in the **`multinmap.sh`** script is:

```
nmap –n –sTU –script modbus-discover.nse –script-args='modbus-
   discover.aggressive=true' –script s7-info.nse –script tftp-enum.nse –script
   http-enum.nse –script wdb-version.nse –script ntp-monlist.nse –top-ports
   3000 192.168.1.0/24
```

The second scan in the **`multinmap.sh`** script is:

```
nmap –n –sTU –script modbus-discover.nse –script-args='modbus-
   discover.aggressive=true' –script s7-info.nse –script tftp-enum.nse –script
   http-enum.nse –script wdb-version.nse –script ntp-monlist.nse –p
   123,102,502,69,80,443,17185 192.168.1.0/24
```

The third scan in the **`multinmap.sh`** script is:

```
nmap –PE –n –sTU –top-ports 1000 –sV –version-intensity 9 –max-retries 1 –T4
   192.168.1.0/24
```

The fourth and final scan in the **`multinmap.sh`** script is:

```
nmap –n –sTU –p T:21,22,23,25,80,102,135,137,139,443,445,502,1024-1028,1521,
   4712,4840,5432,9000,20000,49152-49156,U:53,69,123,161,1434,4713,17185,
   49152-49156 –sV –version-intensity 9 –T4 192.168.1.0.24
```

Appendix B – INL's Testbed Commands

Nmap defines a number of categories for scripts including "safe" and "discovery." The following NSE command can be used to run scripts that are anticipated to provide useful information in an EDS network:

```
nmap -n -sTU --script modbus-discover.nse --script-args='modbus-
discover.aggressive=true' --script s7-info.nse --script tftp-
enum.nse --script http-enum.nse --script wdb-version --script ntp-
monlist -p 123,502,102,69,80,443,17185 <target ip>
```

- tftp-enum.nse (UDP 69): The trivial file transfer protocol (TFTP) is used to store and retrieve files; TFTP doesn't provide a means to list available files on a server, so this script guesses a number of common file names and can determine which are present based on the server response.
- s7-info.nse (TCP 102): Collects device information from Siemens S7 PLCs, including hardware model and serial number.
- modbus-discover.nse (TCP 502): A Modbus TCP server serves as an intermediary for some number of Modbus slaves; this script can enumerate Modbus slaves accessible via a Modbus TCP server and gathers their device model information.
- http-enum.nse (TCP80/443): Identifies web server implementation by checking for associated resource URLs (and to a small extent their contents), also known as fingerprints. This is not specifically oriented towards ICS/EDS devices but does include a fingerprint for the Siemens Simatic S7 PLC.
- wdb-version.nse (UDP 17185): Wind DeBug is a service associated with VxWorks: a real-time OS used on some EDS devices.
- ntp-monlist.nse (UDP 123): Some implementations of NTP support a request for a list of IPs that recently communicated with the service; this script can retrieve each IP along with the nature of its association (server, client, or peer).

Appendix C – LLNL's Testbed Commands

For all IP-enabled devices, the following Nmap scans were performed:

```
nmap -PE -n -sT -p "*" -sV --version-intensity 9 --max-retries 1 -T5 -O --script
modbus-discover.nse --script-args='modbus-discover.aggressive=true' --script s7-
info.nse --script tftp-enum.nse --script http-enum.nse --script wdb-version.nse -
-script ntp-monlist.nse
```

```
nmap -PE -n -sU -p "*" -sV --version-intensity 9 --max-retries 1 -T5 --defeat-
icmp-ratelimit --script modbus-discover.nse --script-args='modbus-
discover.aggressive=true' --script s7-info.nse --script tftp-enum.nse --script
http-enum.nse --script wdb-version.nse --script ntp-monlist.nse
```

On the serial port server and the RTAC devices, the team also performed this additional scan:

```
nmap -PE -n -sTU -p T:21,22,23,25,80,102,135,137,139,443,445,502,1024-
1028,1521,4712,4840,5432,9000,20000,32000-32002,49152-
49156,U:53,69,123,161,1434,4713,17185,49152-49156 -sV --version-intensity 9 --
max-retries 2 --defeat-icmp-ratelimit --script modbus-discover.nse --script-
args='modbus-discover.aggressive=true' --script s7-info.nse --script tftp-
enum.nse --script http-enum.nse --script wdb-version.nse --script ntp-
monlist.nse "$1" > scan.out 2> scan.err
```