CESER
PUBLIC REPORTS

U.S. DEPARTMENT OF ENERGY | Office of Cybersecurity, Energy Security, and Emergency Response

Clean Energy Cybersecurity Accelerator: Cohort 2

runZero Public Report

JULY 2024

CECA CLEAN ENERGY CYBERSECURITY ACCELERATOR

This document was prepared by the National Renewable Energy Laboratory (NREL) under an agreement with and funded by the U.S. Department of Energy.

Clean Energy Cybersecurity Accelerator™ (CECA) Team

Technical Team

Amoresano, Katherine
Balamurugan, Sivasathya Pradha
Blair, Nicholas
Christensen, Dane
Guerra, Jennifer
Hasandka, Adarsh
Howard, Brian
Koul, Neil
Neely, Chelsea
Pailing, Courtney
Urlaub, Nik
Wallace, Anthony
Williams, Gareth

Patria Security LLC

Richardson, Bryan
Schwalm, Keith

Advice and Assistance

Abbondanza, Michael
Castellano, Anthony
Cox, Mariah
Glatter, Casey
Granda, Steve
Henry, Jordan
Lacoste, Jorge
Mujumdar, Monali
Roberts, Cari

Acknowledgments

The authors thank the U.S. Department of Energy Office of Cybersecurity, Energy Security, and Emergency Response and the Office of Energy Efficiency and Renewable Energy for supporting this effort. In addition, we thank utility industry partners Berkshire Hathaway Energy and Duke Energy for sponsoring the technical assessment.

Solution provider:

Sponsors:

Managed by:

Notice

This work was authored by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Cybersecurity, Energy Security, and Emergency Response. The views expressed in the article do not necessarily represent the views of DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

National Renewable Energy Laboratory

15013 Denver West Parkway, Golden, CO 80401

303-275-3000 • www.nrel.gov

NREL/TP-5T00-89105 • June 2024

NREL prints on paper that contains recycled content.

Disclaimer of Endorsement

Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or Alliance. The views and opinions of authors expressed in the available or referenced documents do not necessarily state or reflect those of the United States Government or Alliance.

Acronyms

AaC	assessment as code
AMI	advanced metering infrastructure
API	application programming interface
ARIES	Advanced Research on Integrated Energy Systems
ARP	Address Resolution Protocol
BESS	battery energy storage system
BHE	Berkshire Hathaway Energy
BOE	baseline operating environment
CAASM	cyber asset attack surface management
CECA	Clean Energy Cybersecurity Accelerator™
CESER	Office of Cybersecurity, Energy Security, and Emergency Response
CIDR	classless inter-domain routing
CIFS	Common Internet File System
CLI	command line interface
CPU	central processing unit
DC	direct current
DER	distributed energy resources
DHCP	dynamic host configuration protocol
DMZ	demilitarized zone
DNAT	destination network address translation
DNP3	Distributed Network Protocol, Version 3
DNS	domain name system
DOE	U.S. Department of Energy
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
GUI	graphical user interface
HMI	Human-Machine Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message protocol
ICS	industrial control system
IoT	Internet of Things
IP	Internet Protocol
IPSec	Internet Protocol Security
IPv6	Internet Protocol, Version 6
ISP	internet service provider
IT	information technology
JSON	javascript object notation
KVM	kernel-based virtual machine
LAN	local area network

MAC	media access control
MDNS	Multicast Domain Name System
NDP	Neighbor Discovery Protocol
NetBIOS	Network Basic Input/Output System
NREL	National Renewable Energy Laboratory
NTP	Network Time Protocol
OS	operating system
OSPF	Open Shortest Path First
OT	operational technology
PCAP	packet capture
PV	photovoltaic
QEMU	quick emulator
RDP	Remote Desktop Protocol
REST	representational state transfer
RFC	request for comments
RS232	Recommended Standard 232
RS485	Recommended Standard 485
RTAC	Real-Time Automation Controller
RTU	remote terminal unit
SaaS	software as a service
SCADA	supervisory control and data acquisition
SDN	software defined networking
SEL	Schweitzer Engineering Laboratories
SMA	System, Mess and Anlagentechnik Solar Technology AG
SMB	Server Message Block
SMI	Subscriber Microservices Infrastructure
SNAT	source network address translation
SNMP	Simple Network Management Protocol
SoH	state of health
SSH	Secure Shell Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
URL	Universal Record Locator
VM	virtual machine
VPN	virtual private network
WAN	wide area network
WSMan	Web Services for Management

Executive Summary

National Renewable Energy Laboratory (NREL)'s Clean Energy Cybersecurity Accelerator™ (CECA) program expedites the deployment of emerging operational technology (OT) security technologies to address the most urgent security concerns facing the modern electric grid. The U.S. Department of Energy (DOE) Office of Cybersecurity, Energy Security, and Emergency Response (CESER) and participating utilities sponsor the program. By working directly with utility sponsors to prioritize cybersecurity gaps and to test the ability of solutions under development to address those gaps effectively, CECA helps to both reduce the time to market for developing solutions and assure prospective adopters of the efficacy of new solutions or approaches for solving industry-wide problems.

The second cohort (Cohort 2) of the CECA program sought to address the long-standing challenge of OT asset management. Industrial control system (ICS) networks often "grow organically" and so contain a rich mix of devices developed by multiple vendors over a substantial range of time. This wide variation often impedes asset owners' ability to accurately appraise all devices (known and unknown) connected to their system at any time. This limited visibility inherently prevents system owners from understanding the risks in their system. CECA defined the Cohort 2 prioritized risk as "hidden risks due to incomplete system visibility and device security and configuration." Cohort 2 evaluated solutions designed to identify risks posed by the lack of asset owner visibility into ICS networks and tested the ability of solutions to do this without impact to devices or processes. The latter specifically addresses lingering industry concerns about the potential for active scanning to impact ICS processes and a subsequent reliance on limited passive discovery.

This report presents the outcomes of CECA's evaluation of the runZero product. The runZero product is a highly configurable tool, using deployed agents to discover information about individual assets and a server-based user interface to aggregate and display detailed information about each device in an ICS environment. The runZero product is one of a class of solutions designed to improve an asset owner's visibility into their environment without impeding system operations. This improved visibility allows a better understanding of risks in the system.

The CECA evaluations of the runZero product showed that it identified detailed information about all devices in the environment except for those which were not Internet Protocol (IP)-addressable (i.e. connected to an remote terminal unit (RTU) via serial). The evaluations also verified that runZero's active scanning methods had no impact on the performance of the ICS assets or ongoing supervisory control and data acquisition (SCADA) processes and communications. Although the testing results are not universally generalizable, CECA's conclusions challenge concerns with active scanning in today's energy systems(Dragos 2019; Hanka et al. 2020; Pospisil et al. 2021). In this CECA test environment, active scanning proved "safe," which opens the door for the use of these methods to identify substantially more information than methods that exclusively rely on passive scanning.

Table of Contents

List of Figures

List of Tables

1 Introduction

Many market forces drive the continuous evolution of the electric sector. Energy systems are becoming more diverse, interconnected, distributed, and intelligent, with increasing integration and interconnection of distributed energy resources (DER)s to utility networks. Increased data exchanges between diverse assets introduces new cybersecurity challenges and complicates visibility among interconnected devices.

Cyberattacks that disrupt the critical assets, systems, and networks managed by electric utilities can pose significant, negative impacts on the economy and public health and safety. Mitigating the threats posed by cyberattacks demands increasingly nuanced insight into the technology systems—both information technology (IT) and OT in the case of utilities. Therefore, improving organizations' visibility into their environments is critical to improving the cybersecurity of evolving electric systems. Today, many energy systems' operations manage OT assets using manual processes, which can be time-intensive, making it difficult to respond to cyber incidents quickly and efficiently (NIST 2020). Further, many asset management processes are static and capture only specific points in time or are not repeatable, without real-time visibility into asset status.

Future energy systems will contain an increasing number of clean energy components that are geographically dispersed and have more diverse operators, owners, and stakeholders. The electric sector needs more automated, dynamically responsive tools to improve asset identification and asset management as these diverse, distributed technologies are integrated into existing electric systems.

1.1 CECA Program Overview

The DOE CESER sponsors CECA to expedite the deployment of emerging security technologies that address the most urgent security concerns facing modern and future electric grids. Utility partners provide CECA with strategic direction and cost-sharing. Cohort 2 utility partners include Berkshire Hathaway Energy (BHE) and Duke Energy.

Each CECA cohort focuses on a "prioritized risk," a common risk that is defined by CECA's utility partners. The prioritized risk is then used to select the solutions tested in the defined evaluation scenarios. The cohort participants' solutions are tested on the NREL Advanced Research on Integrated Energy Systems (ARIES) Cyber Range, which provides a platform for controlled emulation in a realistic and scalable cyber-physical environment (NREL 2024). The prioritized risk for CECA Cohort 2 is "hidden risks due to incomplete system visibility and device security and configuration." Cohort 2 includes clean energy components of future energy systems that will help the electric sector assess and gain confidence in adopting new cybersecurity solutions for their evolving electric distribution systems.

1.2 Cohort 2 Theme

The solutions assessed in CECA Cohort 2 focused on identifying risks that might escape detection by asset owners due to incomplete visibility of systems or device configurations. The Cohort 2 solutions aim to improve OT system visibility to shed light on OT networks and assets and to elucidate risks. Capabilities like asset identification, attack surface enumeration, and configuration management can help OT asset owners better understand their risk posture.

Although solutions that monitor and identify assets in IT networks in other domains are widely used, there is much less adoption of monitoring solutions for OT environments. Running active scanning solutions built for IT environments can be unsafe and inappropriate in an OT environment due to bespoke assets, legacy firmware, or proprietary protocols. These factors mean that active scanning in an OT environment may affect expected device functionality if the devices are unable to resolve network packets of modern protocols seen at it's network interface appropriately. The resulting impacts to operations can range from degraded availability to device shutdown or corruption. To increase confidence in the use of active scanning asset identification solutions, the electric sector must ensure that solutions can discover and identify all assets on their systems in real- or near-real-time while not disrupting normal operations. Wider adoption may increase with confidence in the ability of these solutions to enhance visibility while performing within the specific requirements of OT environments.

CECA Cohort 2 evaluated the active and passive asset discovery capabilities of market-ready solutions, documented and analyzed results, and identified gaps in functionality or capabilities. This report describes these results to help accelerate the adoption and improvement of these and similar solutions in the electric sector to mitigate risks. The Cohort 2 evaluations focused on testing the solutions' abilities to illuminate characteristics about the environment. Activities such as red teaming or penetration testing the solution itself were out of scope.

2 Solution Under Test: runZero

runZero Inc.[1] is a proprietary cyber asset attack surface management (CAASM) product that helps organizations identify assets and uncover misconfigurations and risks within an organization's IT and/or OT infrastructure. runZero Inc. designed its product for deployment in IT and OT production environments across a range of industries, including utilities, national laboratories, city and state infrastructure, and federal agencies. The runZero product uses proprietary active scanning and passive sampling, and it can integrate with existing tools to help customers develop an accurate inventory of their assets. It can identify a wide variety of IT and OT assets, including on-site devices, remote devices, and cloud-based resources, such as Internet of Things (IoT).

2.1 Asset Identification

The runZero product's primary function is to identify all assets on a customer's network without disrupting operations. runZero Inc. designed it to avoid common issues of typical security scanners, such as false positives and negatives, as well as adverse impacts on performance and the availability of systems and networks. The runZero product collects hundreds of detailed attributes about devices through unauthenticated scans, passive traffic sampling, and integration with other tools.

The runZero product can identify assets on a network through active scanning and/or passive discovery. runZero's active scanning uses only request for comments (RFC)-compliant IP traffic, does not use security probes, throttles packet rates per host, and uses special provisions for specific fragile devices. When deploying the runZero product, users can enable passive traffic sampling, active scanning, or both methods where the product continuously samples network traffic to identify new assets in between active scans.

2.2 Deployment

2.2.1 Components

Two pieces of software comprise the runZero product: a server that users interact with called the runZero console, and a second piece of software that collects information, a role served either by the runZero command line scanner, the runZero "Explorer" agent,[2] or a combination of the two.

runZero Console

The runZero console is a website/software as a service (SaaS) server that runZero users access online or through a local-hosted version of the server. runZero's console comprises the solution's user interface, with a dashboard that displays information about the organization's sites and Explorers, asset inventory, tasks that are scheduled or have already been completed, reports, and other information. The console also provides account management for the organization. There are several ways to deploy the runZero interface, including both self-hosted and cloud-hosted options. When it is self-hosted, the console is referred to as the platform.

CECA used the self-hosted interface for the tests discussed in this report. The version of the runZero product tested by CECA was reported to identify 180 unique IT and OT device types by "fingerprinting," or correlating attributes and characteristics of the device types into profiles. Types of devices in utility production environments that could be scanned by the runZero product include, but are not limited to, relays, power meters, RTU, programmable logic controllers, and serial-ethernet converters.

Data Collectors

The runZero product collects information in several ways to inform the console. The preferred method is using a runZero Explorer placed behind the firewall in each subnet of interest. Where deploying an Explorer is infeasible, customers can deploy a runZero command line scanner to collect data. If neither method is feasible, customers can collect a packet capture (PCAP) file from their environment with the tools they have access to, and then upload the PCAP to the runZero console.

[1] runZero is both the name of the company and its product. This report uses the term "runZero Inc." to refer to the company and the terms "runZero," "runZero product," or "runZero solution" to refer to the full solutions that were tested by CECA. The terms "runZero product," "runZero console," and runZero platform" are used to distinguish among facets of that solution.

[2] The runZero Explorer is deployed concurrently in multiple locations in a network. Each deployment acts independently. This report frequently references the "runZero Explorers," plural, in recognition of the independent action of the duplicate deployments of the single program.

runZero Explorer
runZero's Explorer is the primary engine for network and asset discovery. Explorers are software-based agents designed to be deployed in each subnet, either to actively scan assets or to passively sample traffic. The interface and Explorer communicate over a web socket and representational state transfer (REST) application programming interface (API). An Explorer connects to the console using the interface's Universal Record Locator (URL) and Transport Layer Security (TLS) certificate, both of which are automatically loaded into the Explorer binary. The TLS certificate is hard-coded for each interface, and there is an option to override the TLS certificate using environment variables. The Explorer binary can be downloaded directly from the interface.

Command Line Scanner
runZero can operate via command line in air-gapped networks, without access to the internet, where it is not feasible to deploy an Explorer with a connection to the console. The scanner has the same options and similar performance characteristics to the Explorer. The scanner output file, named `scan.rumble.gz`, can be uploaded to the runZero console (runZero 2024a).

Manual PCAP Uploads
A final method for collecting data is via PCAP uploads. Customers can collect PCAPs using existing tools and infrastructure and then upload them to the runZero console for analysis.

2.2.2 CECA Integration

CECA deployed runZero with a self-hosted on-premise air-gapped platform and installed Explorer on virtual machines hosted in each subnet of interest.[3] CECA evaluated the Cohort 2 solutions in two separate environments: a smaller-scale generation and distribution system modeled with a PV plant, substation, and control center; and an advanced metering infrastructure (AMI) environment modeled on meters served by a larger substation. The integration of the runZero platform with the baseline operating environment (BOE) is shown in Figure 1.

PV Plant and Substation Environment

Most tests were conducted in an environment that featured a basic utility control center, a clean energy-generating PV plant (i.e., solar plant), and a substation. These are represented in Figure 1. This environment was built to simulate all the complexities that a solution could be expected to deal with when identifying assets in an ICS network containing both clean energy components and legacy OT devices. The environment featured 13 different OT devices communicating over a variety of media, protocols, and firmware versions, as detailed in Appendix A.

The runZero console was integrated into this environment according to the runZero documentation (runZero 2024c). The self-hosted runZero console was installed in a demilitarized zone (DMZ) in the control center, and the only security change to the BOE required for this integration was allowing Transmission Control Protocol (TCP) network traffic destined for port `443` of the runZero platform. runZero Explorer was installed on five hosts, each located in a relevant subnet with devices of interest.

[3] CECA testing used runZero version 20240301. The runZero solution is regularly updated with additional functionality.

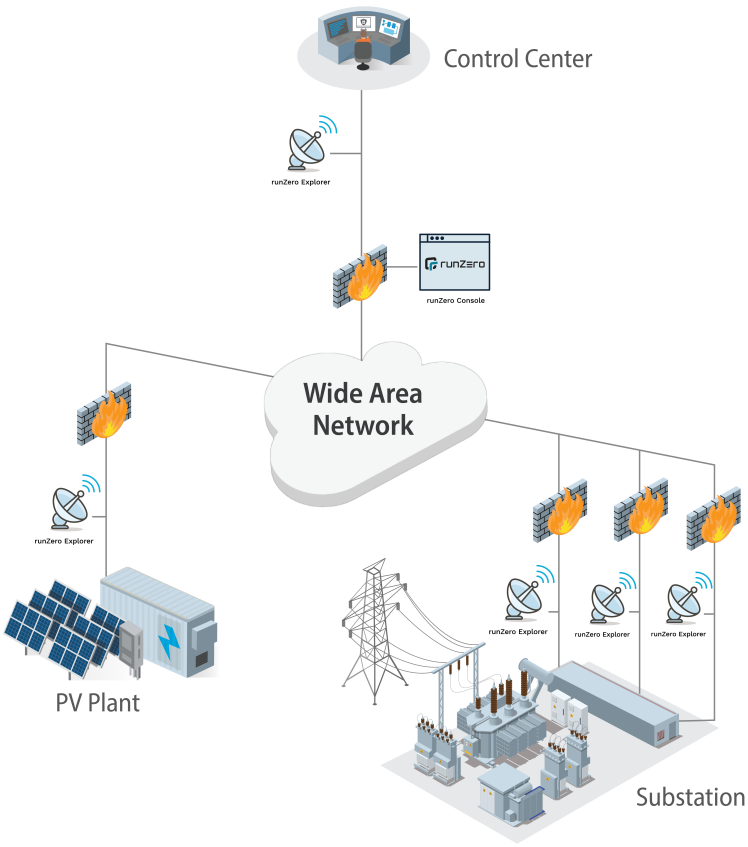


Figure 1. High-level overview of the PV plant and substation environment integrated with runZero

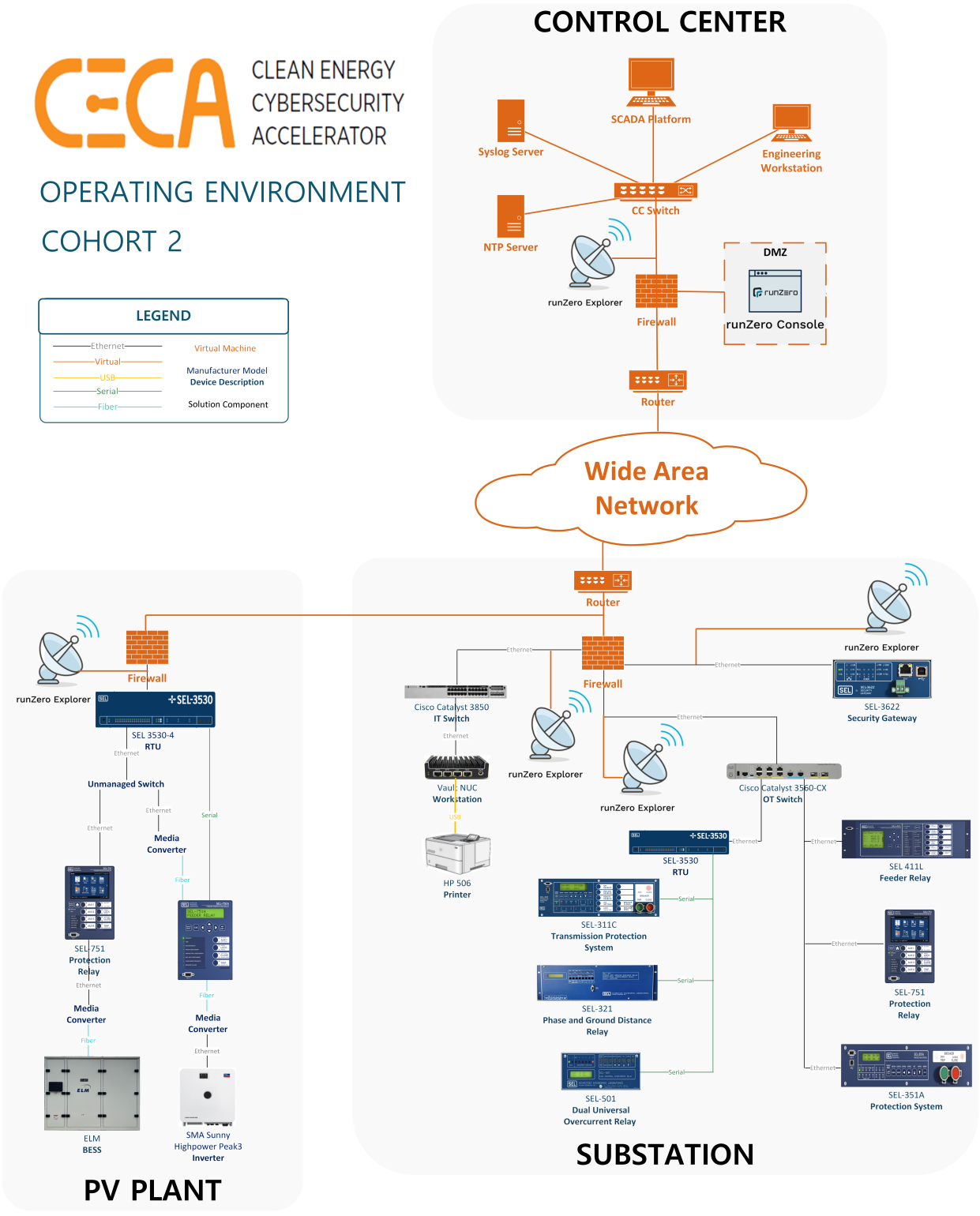


Figure 2. Diagram of the PV plant, substation, and control center integrated with runZero

AMI Environment

To evaluate the solution at scale, CECA also integrated runZero into a separate, larger environment featuring 3,948 AMI devices on a single flat network within the subnet (`10.200.1.0/20`). The size of this environment represents the number of customers that could be served by a larger substation. Figure 3 provides a diagram of this environment, which was used only for evaluation in Scenario 4: Scale Discovery. In the AMI environment, a single virtual machine was added to host a runZero Explorer, and the runZero console were added on a separate subnet.

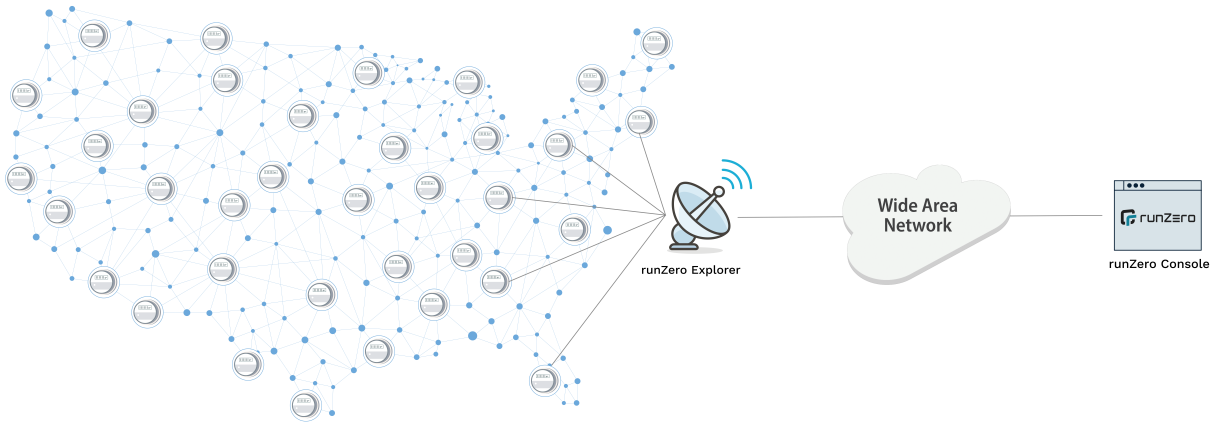


Figure 3. High-level overview of the AMI environment integrated with runZero

3 Evaluations and Results

CECA developed an evaluation plan based on the Cohort 2 prioritized risk—hidden risks due to incomplete system visibility—to test the capabilities of each solution selected for Cohort 2. The evaluation plan detailed four scenarios that each tested several characteristics of the solution. Each individual scenario is a scientific, repeatable set of procedures and data collection methods. Table 1 shows which characteristics were tested in each scenario. Following the table are short descriptions of each characteristic.

Table 1. Testing matrix

	Scenario 1 Initial Discovery	**Scenario 2** Change Discovery	**Scenario 3** Passive Discovery	**Scenario 4** Scale Discovery
Timing	✓			✓
Inventory accuracy	✓		✓	✓
Data richness	✓		✓	
Additional network traffic	✓		✓	✓
Disruption of operations	✓			
Alert		✓		✓
Change detection		✓		

- **Timing**: How long does it take to identify all the assets in the environment?
- **Inventory accuracy**: How many assets in the environment did the solution correctly identify?
- **Data richness**: For each identified asset, how detailed are the data collected by the solution?
- **Additional network traffic**: How much additional network traffic does the solution add to the ICS network? [4]
- **Disruption of operations**: Does the solution affect any normal operations of the ICS system? *The details for how this was evaluated can be found in appendix D.*
- **Alert**: Does the solution notify users of unexpected devices on the network?
- **Change detection**: How does the solution track changes to assets over time?

The following sections describe each test's objective and results. Details about the exact procedures for each test can be found in appendix D. Each test was run five times to ensure that the data were consistent.

3.1 Scenario 1: Initial Discovery

This scenario focused on how a solution performed during the initial discovery of an environment that the solution had not previously identified.

Three different profiles tested the solution across a variety of settings:

- **Scenario 1.A: Conservative**—tuned to be the least likely to affect ongoing operations of the underlying ICS or negatively affect fragile OT devices
- **Scenario 1.B: Default**—the default or recommended settings of the product
- **Scenario 1.C: In Depth**—tuned to identify as much information as possible, specifically about OT assets in the PV plant and substation.

CECA configured the runZero platform to meet these three profiles by using the default settings and applying the runZero playbook for configuring OT scans.[5]

[4] Additional network traffic measures the total amount of data that is added to the network by the solution, not the rate at which it is added. The runZero solution is configurable and the rate can be constrained so that the additional network traffic can be spread out over long periods of time to achieve whatever rate an operator desires.

[5] The runZero console/platform is extremely customizable, and different scan profiles can be applied to each scan. A real-world deployment would likely use one scan profile for the control center subnet and a different profile for the OT subnets. For simplicity of testing, CECA used the same profile for all scans in a test.

Table 2. Scenario 1 scan profiles

Scenario	Scan Profile
1.A: Conservative	runZero "OT Limited"
1.B: Default	runZero default settings
1.C: In Depth	runZero "OT Full" with additional in-depth Modbus and DNP3 settings

3.1.1 Scenario 1.A: Conservative

This scenario focused on a solution's ability to identify assets in a new environment in the safest possible manner, without having a negative impact on OT devices. Details for this configuration can be found in Appendix C.4.1.

Timing

Scenario 1.A took an average of 181 seconds with a standard deviation of 7 seconds.

Inventory Accuracy

The runZero platform successfully identified all assets in the environment except for the assets that were connected via serial behind a RTU and were not IP addressable. These devices were:

- SEL 311C
- SEL 321
- SEL 501.

Data Richness

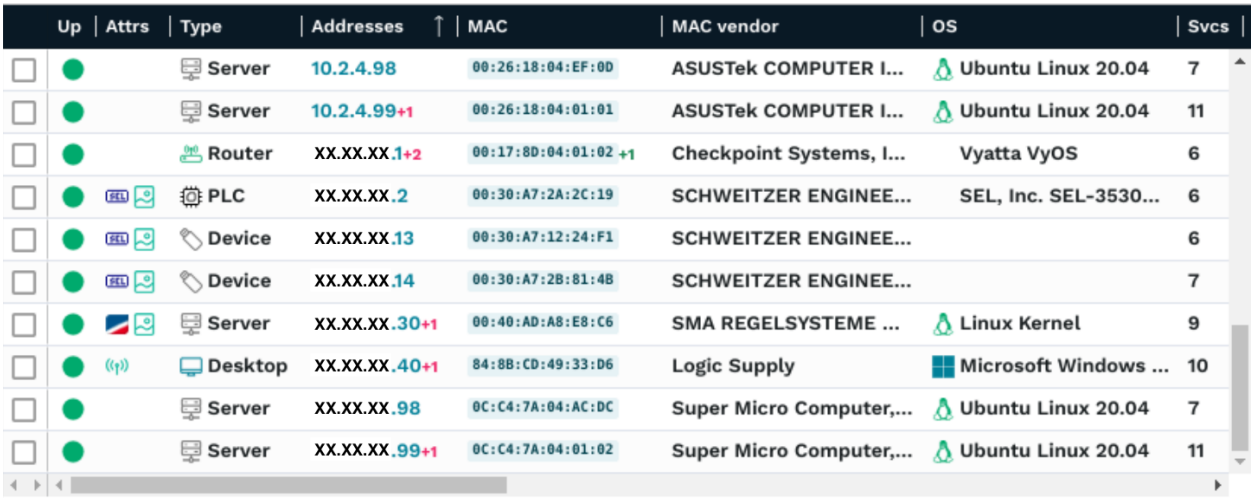
The limited scan identified standard attributes—media access control (MAC) address, secondary MAC addresses, MAC vendor, and IP address(es)—for each device identified in the scan (but not for the devices listed above that were not identified at all). In addition, the scan identified any services available and ports open for probes that were enabled in the scan. Examples include Address Resolution Protocol (ARP), Network Basic Input/Output System (NetBIOS), Simple Network Management Protocol (SNMP), Secure Shell Protocol (SSH), File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP)/TLS, Server Message Block (SMB), Network Time Protocol (NTP), telnet, Remote Desktop Protocol (RDP), and Modbus.

Hostnames were identified for 21 of the 33 devices. Hostnames were not found for the SCADA platform, the NTP and syslog servers, the substation workstation, or any Schweitzer Engineering Laboratories (SEL) 751, 411L, or 351A devices. The operating system (OS) was identified for 24 of the 33 devices. The OS was not found for the substation workstation or any SEL 751, 411L, or 351A devices. The OS versions were identified for 20 of the 33 devices. The OS version was not found for the substation workstation; any of the SEL 3633, 3530 RTAC, 351A, 411L, or 751 devices; or the SMA Solar Sunny Highpower device.

Table 3. Scenario 1.A data richness

Device	Hostname	IP Address	MAC Address	MAC Vendor	OS	OS Version
Control center (7 devices)						
cc firewall	✓	✓	✓	✓	✓	✓
cc admin vm	✓	✓	✓	✓	✓	✓
cc runZero vm	✓	✓	✓	✓	✓	✓
SCADA platform		✓	✓	✓	✓	✓
Engineering workstation	✓	✓	✓	✓	✓	✓
NTP server		✓	✓	✓	✓	✓
Syslog server		✓	✓	✓	✓	✓
Substation OT (11 devices)						
Sub firewall	✓	✓	✓	✓	✓	✓
OT switch	✓	✓	✓	✓	✓	✓
Sub-ot admin vm	✓	✓	✓	✓	✓	✓
Sub-ot runZero vm	✓	✓	✓	✓	✓	✓
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 411L		✓	✓	✓		
SEL 751		✓	✓	✓		
SEL 351A		✓	✓	✓		
*__SEL 311C__						
*__SEL 321__						
*__SEL 501__						
Substation OT gateway (3 devices)						
Sub-ot-gateway admin vm	✓	✓	✓	✓	✓	✓
Sub-ot-gateway runZero vm	✓	✓	✓	✓	✓	✓
SEL 3622	✓	✓	✓	✓	✓	
Substation IT (4 devices)						
IT switch	✓	✓	✓	✓	✓	✓
Sub-it admin vm	✓	✓	✓		✓	✓
Sub-it runZero vm	✓	✓	✓		✓	✓
Workstation		✓	✓	✓		
PV plant (8 devices)						
PV firewall	✓	✓	✓	✓	✓	✓
PV admin vm	✓	✓	✓	✓	✓	✓
PV runZero vm	✓	✓	✓	✓	✓	✓
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 751		✓	✓	✓		
SEL 751		✓	✓	✓		
SMA Sunny Highpower	✓	✓	✓	✓	✓	
ELM BESS	✓	✓	✓	✓	✓	✓
Total (of 33 devices)	21	30	30	28	24	20

* Serial device not identified



Up	Attrs	Type	Addresses	MAC	MAC vendor	OS	Svcs
●		Server	10.2.4.98	00:26:18:04:EF:0D	ASUSTek COMPUTER I...	Ubuntu Linux 20.04	7
●		Server	10.2.4.99+1	00:26:18:04:01:01	ASUSTek COMPUTER I...	Ubuntu Linux 20.04	11
●		Router	XX.XX.XX.1+2	00:17:8D:04:01:02 +1	Checkpoint Systems, I...	Vyatta VyOS	6
●	REL	PLC	XX.XX.XX.2	00:30:A7:2A:2C:19	SCHWEITZER ENGINEE...	SEL, Inc. SEL-3530...	6
●	REL	Device	XX.XX.XX.13	00:30:A7:12:24:F1	SCHWEITZER ENGINEE...		6
●	REL	Device	XX.XX.XX.14	00:30:A7:2B:81:4B	SCHWEITZER ENGINEE...		7
●		Server	XX.XX.XX.30+1	00:40:AD:A8:E8:C6	SMA REGELSYSTEME ...	Linux Kernel	9
●		Desktop	XX.XX.XX.40+1	84:8B:CD:49:33:D6	Logic Supply	Microsoft Windows ...	10
●		Server	XX.XX.XX.98	0C:C4:7A:04:AC:DC	Super Micro Computer,...	Ubuntu Linux 20.04	7
●		Server	XX.XX.XX.99+1	0C:C4:7A:04:01:02	Super Micro Computer,...	Ubuntu Linux 20.04	11

Figure 4. Select view of runZero asset inventory after Scenario 1.A

Additional Network Traffic

The additional network traffic added by each Explorer to its respective subnet is shown in Table 4.

Table 4. Explorer network traffic on subnets in Scenario 1.A

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	27,620 kB	9 kB
Substation IT	4	900 kB	5 kB
Substation OT gateway	3	899 kB	20 kB
Substation OT	11	2,812 kB	16 kB
PV plant	8	6,322 kB	14 kB

The additional network traffic generated by each Explorer to its respective platform is shown in Table 5.

Table 5. Explorer network traffic to platform in Scenario 1.A

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	171 kB	1 kB
Substation IT	4	119 kB	1 kB
Substation OT gateway	3	170 kB	1 kB
Substation OT	11	314 kB	4 kB
PV plant	8	328 kB	3 kB

Disruption of Operations

Active scanning did not affect any of the underlying ICS processes or OT devices. Details of the methods used for monitoring ongoing operations can be found in Appendix A.4.

3.1.2 Scenario 1.B: Default

This scenario focused on a solution's ability to identify assets in a new environment with default or recommended settings. Details for this configuration can be found in Appendix C.4.2.

The main difference between the OT limited scan and the default scan was the richness of information that was identified about each service. Leaving more runZero probes enabled allowed additional information to be collected. A demonstration of the runZero platform's capability to collect and sort rich information about each asset is shown in Figure 5.[6]

[6]These SEL favicon's were observed and collected in all three scan profiles used in Scenario 1.

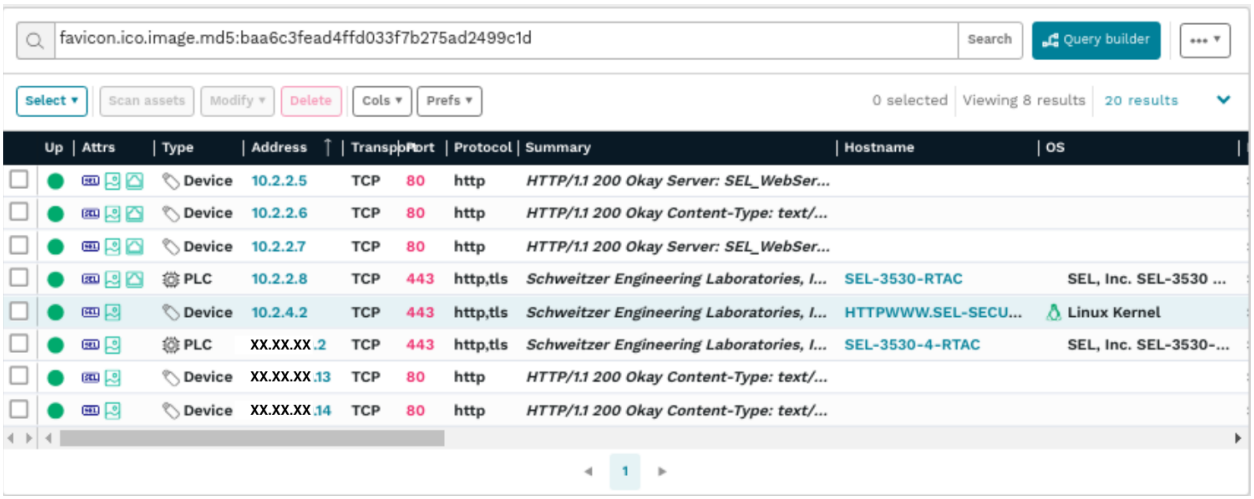


Figure 5. Example of runZero asset view sorted to show only assets with SEL favicons

Timing

Scenario 1.B took an average of 224 seconds with a standard deviation of 3 seconds.

Inventory Accuracy

As in Scenario 1A, the runZero platform successfully identified all assets in the environment except for the assets that were connected via serial behind the RTU and did not have an associated IP address. These devices were:

- SEL 311C
- SEL 321
- SEL 501.

Data Richness

In addition to the data collected in the OT limited scan, the default scan identified three more hostnames (the SCADA platform and the NTP and syslog servers) and several additional services, including PostgreSQL, Internet Control Message protocol (ICMP), Cisco Subscriber Microservices Infrastructure (SMI), and Web Services for Management (WSMan). runZero identified TCP port 20,000 as open on the OT devices communicating via Distributed Network Protocol, Version 3 (DNP3), but it did not identify a DNP3 service.[7]

[7]The inability to profile the DNP3 service might stem from the behavior of DNP3 outstations ignoring new masters if they have an established master. The runZero Explorer defaults to using a DNP3 address of -1 to prevent disrupting ongoing OT processes that are communicating over DNP3.

Table 6. Scenario 1.B data richness

Device	Hostname	IP Address	MAC Address	MAC Vendor	OS	OS Version
Control center (7 devices)						
cc firewall	✓	✓	✓	✓	✓	✓
cc admin vm	✓	✓	✓	✓	✓	✓
cc runZero vm	✓	✓	✓	✓	✓	✓
SCADA platform	✓	✓	✓	✓	✓	✓
Engineering workstation	✓	✓	✓	✓	✓	✓
NTP server	✓	✓	✓	✓	✓	✓
Syslog server	✓	✓	✓	✓	✓	✓
Substation OT (11 devices)						
Sub firewall	✓	✓	✓	✓	✓	✓
OT switch	✓	✓	✓	✓	✓	✓
Sub-ot admin vm	✓	✓	✓	✓	✓	✓
Sub-ot runZero vm	✓	✓	✓	✓	✓	✓
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 411L		✓	✓	✓		
SEL 751		✓	✓	✓		
SEL 351A		✓	✓	✓		
*__SEL 311C__						
*__SEL 321__						
*__SEL 501__						
Substation OT gateway (3 devices)						
Sub-ot-gateway admin vm	✓	✓	✓	✓	✓	✓
Sub-ot-gateway runZero vm	✓	✓	✓	✓	✓	✓
SEL 3622	✓	✓	✓	✓	✓	
Substation IT (4 devices)						
IT switch	✓	✓	✓	✓	✓	✓
Sub-it admin vm	✓	✓	✓		✓	✓
Sub-it runZero vm	✓	✓	✓		✓	✓
Workstation		✓	✓	✓		
PV plant (8 devices)						
PV firewall	✓	✓	✓	✓	✓	✓
PV admin vm	✓	✓	✓	✓	✓	✓
PV runZero vm	✓	✓	✓	✓	✓	✓
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 751		✓	✓	✓		
SEL 751		✓	✓	✓		
SMA Sunny Highpower	✓	✓	✓	✓	✓	
ELM BESS	✓	✓	✓	✓	✓	✓
Total (of 33 devices)	24	30	30	28	24	20

* Serial device not identified

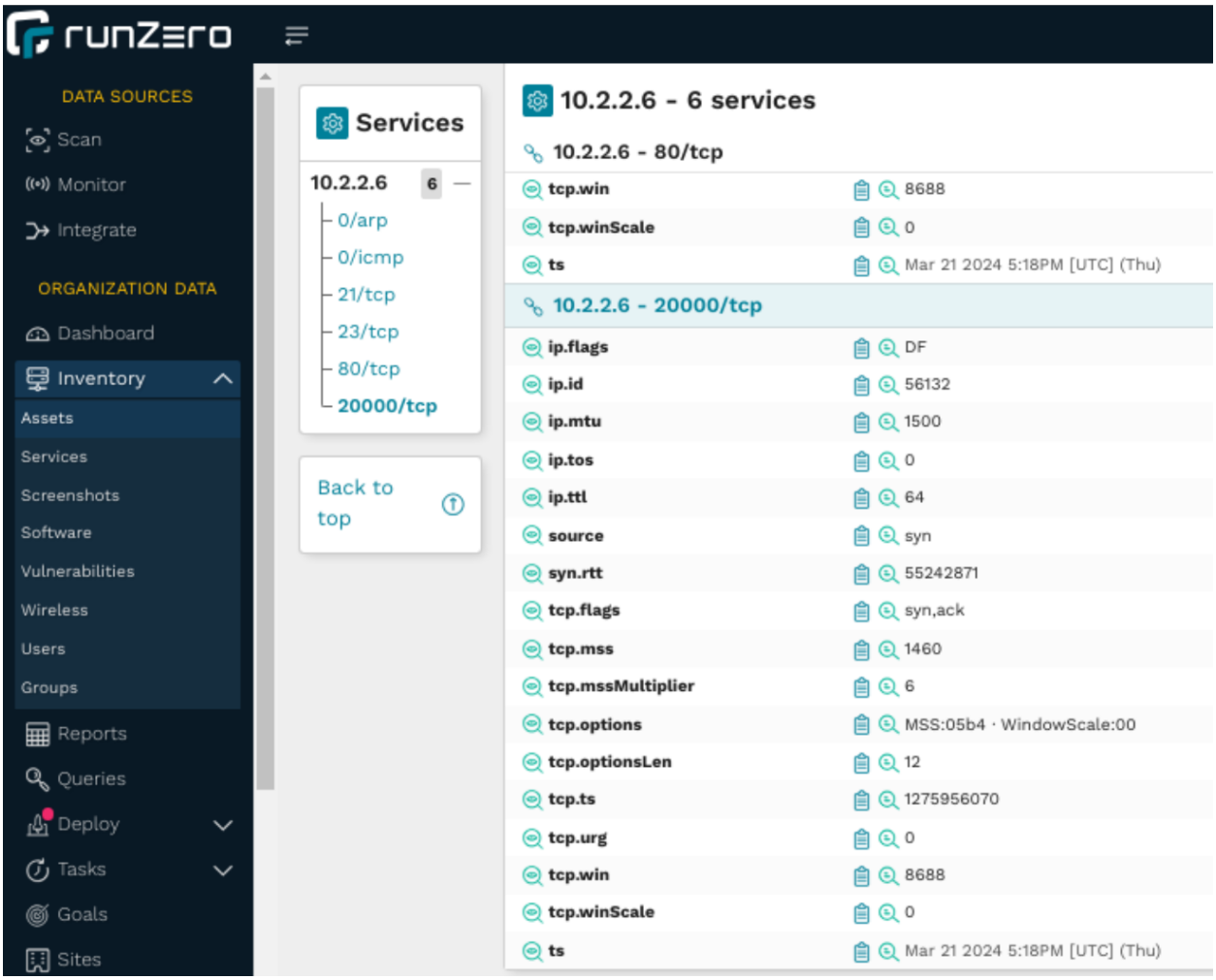


Figure 6. Attributes collected by runZero for port 20000 on a device communicating via DNP3

Additional Network Traffic

The additional network traffic added by each Explorer to its respective subnet is shown in Table 7.

Table 7. Explorer betwork traffic on subnets in Scenario 1.B

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	26,710 kB	37 kB
Substation IT	4	255 kB	1 kB
Substation OT gateway	3	399 kB	1 kB
Substation OT	11	1,730 kB	5 kB
PV plant	8	5,152 kB	73 kB

The additional network traffic generated by each Explorer to its respective platform is shown in Table 8.

Table 8. Explorer network traffic to platform in Scenario 1.B

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	97 kB	1 kB
Substation IT	4	71 kB	2 kB
Substation OT gateway	3	133 kB	1 kB
Substation OT	11	239 kB	2 kB
PV plant	8	240 kB	2 kB

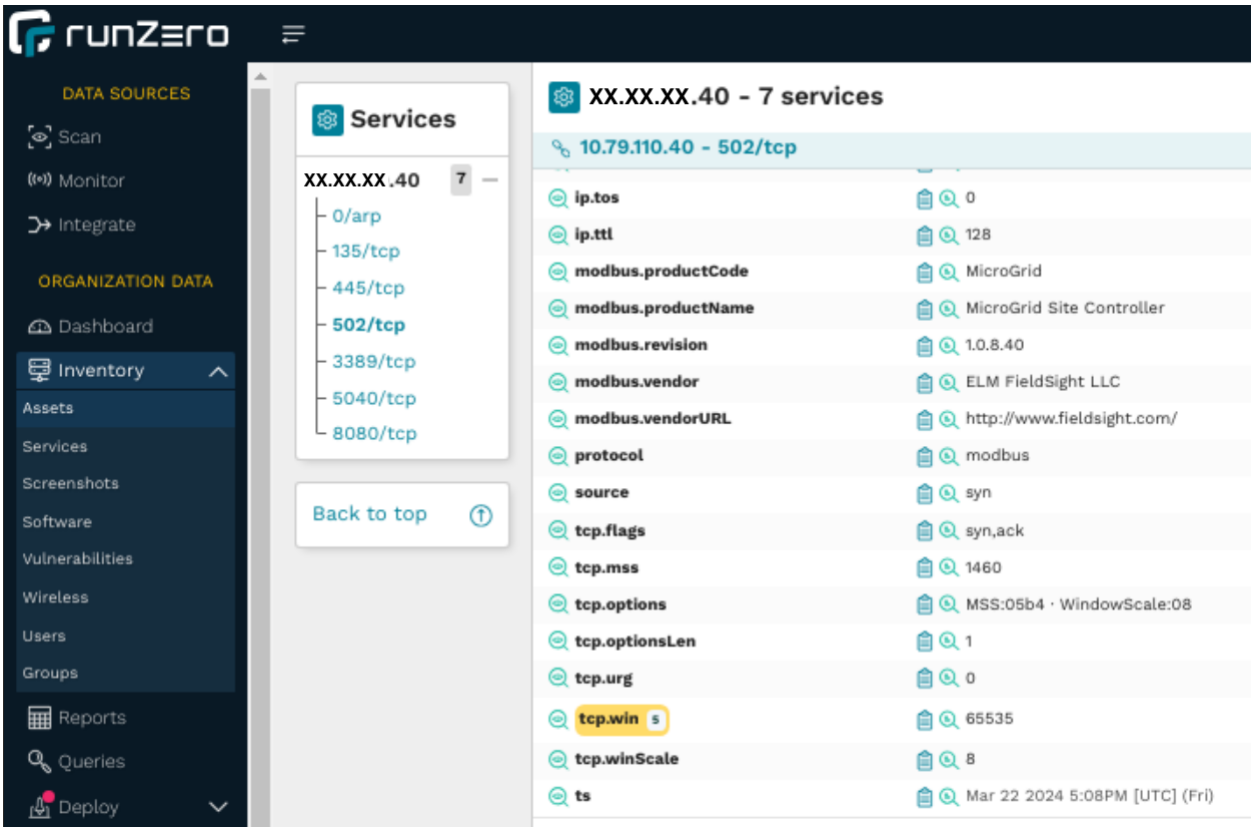


Figure 7. runZero detailed identification of Modbus attributes

Disruption of Operations
Active scanning did not affect any of the underlying ICS processes or OT devices. Details of the methods used for monitoring ongoing operations can be found in Appendix A.4.

3.1.3 Scenario 1.C: In Depth

This scenario focused on a solution's ability to identify assets in a new environment with the goal of identifying as much information as possible about the OT devices. Details for this configuration can be found in Appendix C.4.3.

CECA did not observe a marked difference in the results between the OT full scan and the previously executed default scan. The goal of the OT full scan is to provide detailed identification of OT devices in a safe manner. An example of some of the detailed data that runZero can collect is provided in Figure 7.[8]

Timing
Scenario 1.C took an average of 286 seconds with a standard deviation of 5 seconds.

Inventory Accuracy
As in the previous two scans, the runZero platform successfully identified all assets in the environment except for the assets that were connected via serial behind the RTU and did not have an associated IP address. These devices were:

- SEL 311C
- SEL 321
- SEL 501.

[8]These Modbus values were observed and collected in all three scan profiles used in Scenario 1.

Data Richness

The data richness was identical to the data identified in Scenario 1A across the evaluated fields. Several additional probes are used and fingerprinted (such as ICMP).

Table 9. Scenario 1.C data richness

Device	Hostname	IP Address	MAC Address	MAC Vendor	OS	OS Version
Control center (7 devices)						
cc firewall	✓	✓	✓	✓	✓	✓
cc admin vm	✓	✓	✓	✓	✓	✓
cc runZero vm	✓	✓	✓	✓	✓	✓
SCADA platform		✓	✓	✓	✓	✓
Engineering workstation	✓	✓	✓	✓	✓	✓
NTP server		✓	✓	✓	✓	✓
Syslog server		✓	✓	✓	✓	✓
Substation OT (11 devices)						
Sub firewall	✓	✓	✓	✓	✓	✓
OT switch	✓	✓	✓	✓	✓	✓
Sub-ot admin vm	✓	✓	✓	✓	✓	✓
Sub-ot runZero vm	✓	✓	✓	✓	✓	✓
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 411L		✓	✓	✓		
SEL 751		✓	✓	✓		
SEL 351A		✓	✓	✓		
*__SEL 311C__						
*__SEL 321__						
*__SEL 501__						
Substation OT gateway (3 devices)						
Sub-ot-gateway admin vm	✓	✓	✓	✓	✓	✓
Sub-ot-gateway runZero vm	✓	✓	✓	✓	✓	✓
SEL 3622	✓	✓	✓	✓	✓	
Substation IT (4 devices)						
IT switch	✓	✓	✓	✓	✓	✓
Sub-it admin vm	✓	✓	✓		✓	✓
Sub-it runZero vm	✓	✓	✓		✓	✓
Workstation		✓	✓	✓		
PV plant (8 devices)						
PV firewall	✓	✓	✓	✓	✓	✓
PV admin vm	✓	✓	✓	✓	✓	✓
PV runZero vm	✓	✓	✓	✓	✓	✓
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 751		✓	✓	✓		
SEL 751		✓	✓	✓		
SMA Sunny Highpower	✓	✓	✓	✓	✓	
ELM BESS	✓	✓	✓	✓	✓	✓
Total (of 33 devices)	21	30	30	28	24	20

* Serial device not identified

Additional network traffic
The additional network traffic added by each Explorer to its respective subnet:

Table 10. Explorer network traffic on subnets in Scenario 1.C

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	27,584 kB	5 kB
Substation IT	4	871 kB	1 kB
Substation OT gateway	3	891 kB	2 kB
Substation OT	11	2,753 kB	8 kB
PV plant	8	6,243 kB	12 kB

The additional network traffic generated by each Explorer to its respective platform:

Table 11. Explorer network traffic to platform in Scenario 1.C

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	12 kB	2 kB
Substation IT	4	127 kB	1 kB
Substation OT gateway	3	183 kB	3 kB
Substation OT	11	333 kB	5 kB
PV plant	8	347 kB	6 kB

Disruption of Operations
Active scanning did not affect any of the underlying ICS processes or OT devices. Details of the methods used for monitoring ongoing operations can be found in Appendix A.4.

3.2 Scenario 2: Change Discovery

Scenario 2 focused on how a solution identifies changes to a previously analyzed environment. CECA designed this scenario as a follow-on test from Scenario 1,[9] after the solution had already identified the initial assets integrated into the environment. Scenario 2 sought to understand how the solution adapted to and identified changes in the operating environment. Scenario 2 also explored a solution's ability to detect and produce alerts based on new and/or altered devices in the environment. Such alerts are of interest to system owners to make them aware of new assets that are either intentionally or unintentionally modified or added to the environment.

The changes made to the environment between the Scenario 1 and Scenario 2 tests were:

- **New devices:**
  - **Attacker device:** A new device was added to the environment that simulates an unauthorized attacker connecting to the network. The simulated attacker plugged in a RaspberryPi running Kali Linux OS in the substation OT subnet.
  - **Misconfigured device:** The printer in the substation IT subnet was plugged into the switch via ethernet, which represents a policy violation.
- **Changes to existing devices:**
  - The IP address of the engineering workstation in the control center was changed to `10.1.1.10`, but the MAC address and all other attributes were held constant.
  - The MAC address of the syslog server in the control center was changed to `10:c5:95:ff:04:ff`, but the IP address and all other attributes were held constant.

These changes are visually depicted in Figure 8.

[9] Scenario 2 used the results from Scenario 1C as a starting point to represent an "onboarded" solution.

CLEAN ENERGY CYBERSECURITY ACCELERATOR

OPERATING ENVIRONMENT

COHORT 2 - SCENARIO 2

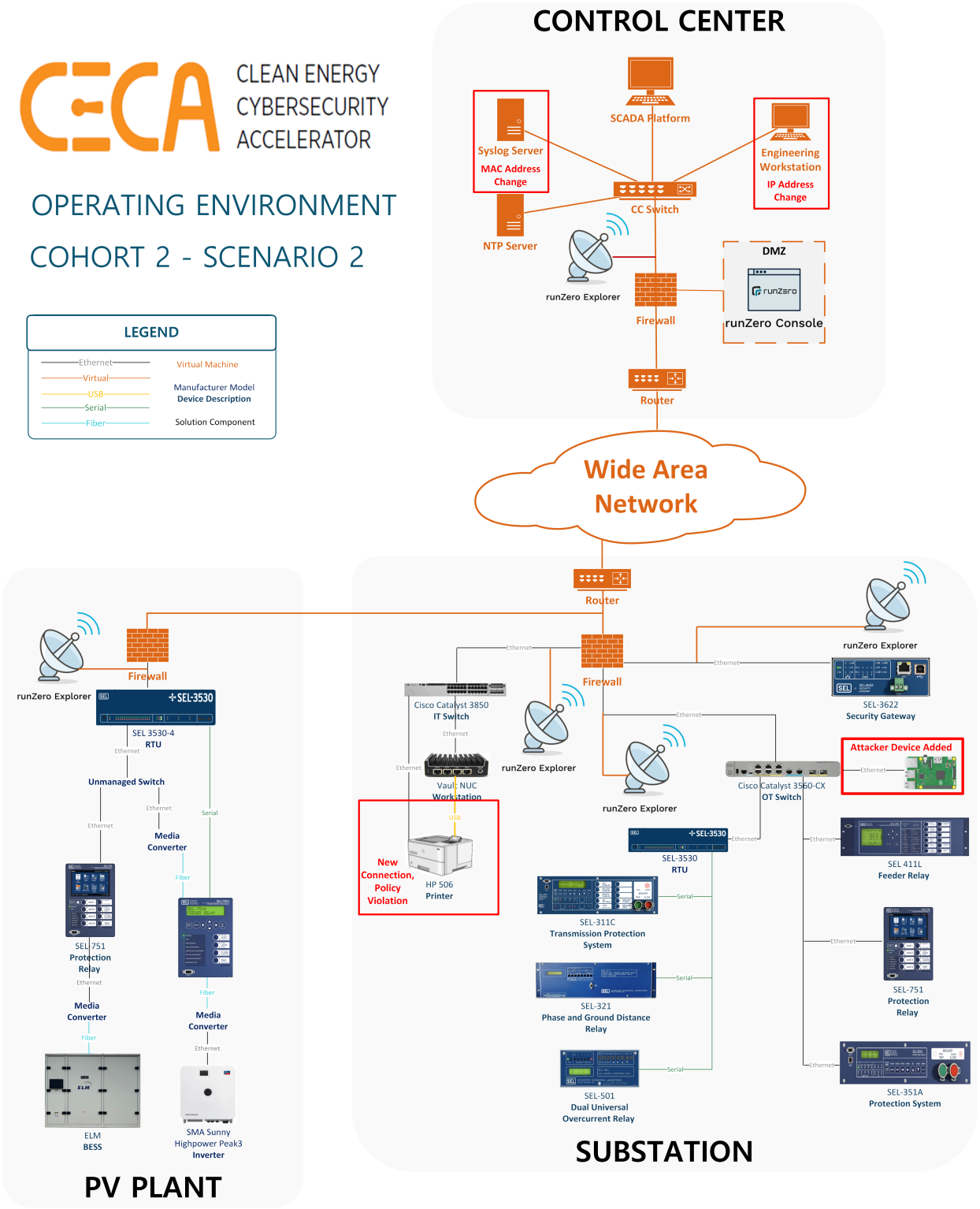


Figure 8. Cohort 2 runZero AOE

Scenario 2 used the same OT full scan profile as was used in Scenario 1C.

Alert

CECA configured the runZero platform to alert when it identified any new assets in the environment. In each test, the runZero platform alerted based on the new devices it found: the printer, the attacker device, and the syslog server with a different MAC address. Figure 9 shows an example alert internal to the runZero platform.[10]

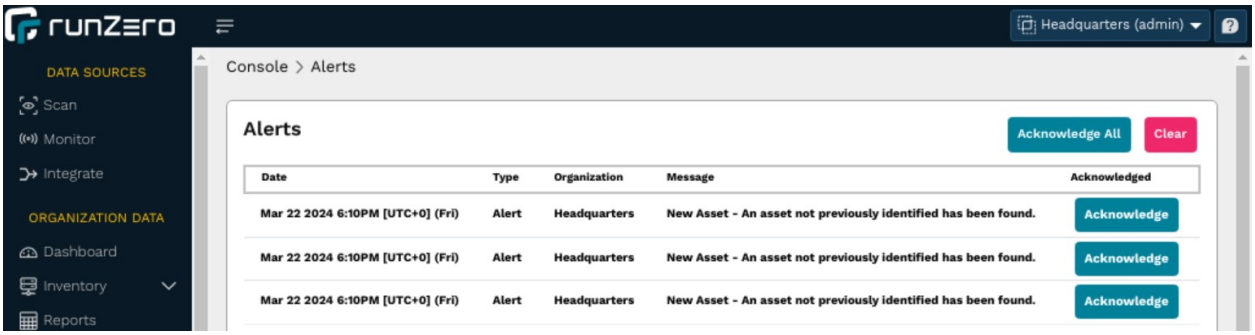


Figure 9. runZero alerts for new devices

Change Detection

The runZero platform successfully identified each of the four changes introduced into the environment. The new attacker device and newly connected printer were identified and profiled—just as in-depth as any other device. The changed IP address was tracked, and the device entry for the engineering workstation was updated with the new IP. The most interesting behavior is how the solution tracked the changed MAC address for the syslog server. The runZero platform created a new device entry in the inventory database, and it marked the "old" syslog server as "offline," as shown in figure 10.

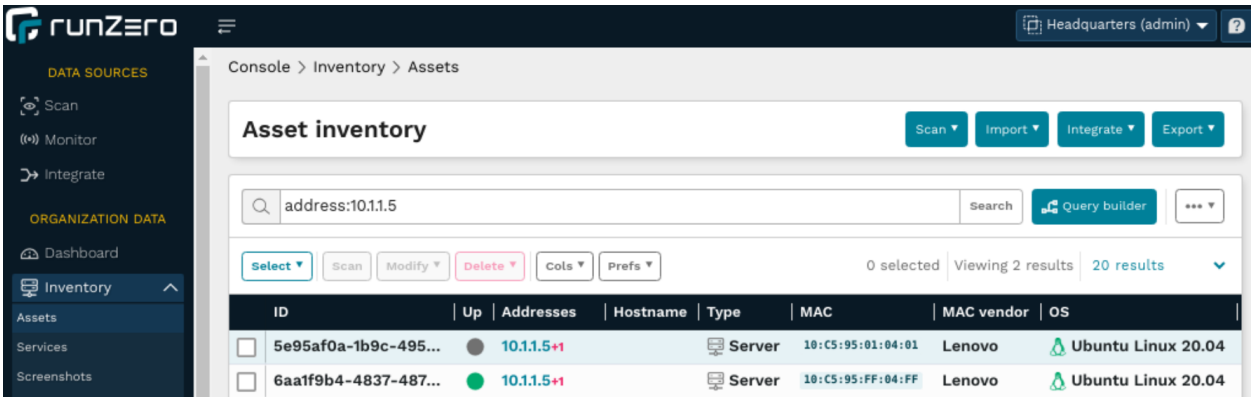


Figure 10. runZero asset inventory sorted to show changed MAC

Measurements Not Used for Evaluation

The following criteria were not part of the objectives for Scenario 2, but they were measured during the tests. They provide additional data points for the OT full scan tested in Scenario 1C.

Timing

Scenario 2 took an average of 294 seconds with a standard deviation of 7 seconds.

[10] The runZero console/platform allows for internal alerts, as shown here, but it can also send alerts via email or webhooks, if configured to do so.

Additional Network Traffic

The additional network traffic added by each Explorer to its respective subnet is shown in Table 12.

Table 12. Explorer network traffic on subnets in Scenario 2

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	27,583 kB	15 kB
Substation IT	5	6,535 kB	83 kB
Substation OT gateway	3	891 kB	2 kB
Substation OT	12	2,887 kB	4 kB
PV plant	8	6,175 kB	196 kB

The additional network traffic generated by each Explorer to its respective platform is shown in Table 13.

Table 13. Explorer network traffic to platform in Scenario 2

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	178 kB	1 kB
Substation IT	5	522 kB	25 kB
Substation OT gateway	3	185 kB	8 kB
Substation OT	12	355 kB	7 kB
PV plant	8	347 kB	8 kB

Data Richness

Table 14. Scenario 2 data richness

Device	Hostname	IP Address	MAC Address	MAC Vendor	OS	OS Version
Control center (7 devices)						
cc firewall	✓	✓	✓	✓	✓	✓
cc admin vm	✓	✓	✓	✓	✓	✓
cc runZero vm	✓	✓	✓	✓	✓	✓
SCADA platform		✓	✓	✓	✓	✓
Engineering workstation	✓	✓	✓	✓	✓	✓
NTP server		✓	✓	✓	✓	✓
Syslog server		✓	✓	✓	✓	✓
Substation OT (12 devices)						
Sub firewall	✓	✓	✓	✓	✓	✓
OT switch	✓	✓	✓	✓	✓	✓
Sub-ot admin vm	✓	✓	✓	✓	✓	✓
Sub-ot runZero vm	✓	✓	✓	✓	✓	✓
Attacker RPi		✓	✓	✓	✓	
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 411L		✓	✓	✓		
SEL 751		✓	✓	✓		
SEL 351A		✓	✓	✓		
*__SEL 311C__						
*__SEL 321__						
*__SEL 501__						
Substation OT gateway (3 devices)						
Sub-ot-gateway admin vm	✓	✓	✓	✓	✓	✓
Sub-ot-gateway runZero vm	✓	✓	✓	✓	✓	✓
SEL 3622	✓	✓	✓	✓	✓	
Substation IT (5 devices)						
IT switch	✓	✓	✓	✓	✓	✓
Sub-it admin vm	✓	✓	✓		✓	✓
Sub-it runZero vm	✓	✓	✓		✓	✓
Workstation		✓	✓	✓		
Printer	✓	✓	✓	✓	✓	✓
PV plant (8 devices)						
PV firewall	✓	✓	✓	✓	✓	✓
PV admin vm	✓	✓	✓	✓	✓	✓
PV runZero vm	✓	✓	✓	✓	✓	✓
SEL 3530 RTAC	✓	✓	✓	✓	✓	
SEL 751		✓	✓	✓		
SEL 751		✓	✓	✓		
SMA Sunny Highpower	✓	✓	✓	✓	✓	
ELM BESS	✓	✓	✓	✓	✓	✓
Total (of 35 devices)	22	32	32	30	26	21

* Serial device not identified

Disruption of Operations

Active scanning did not affect any of the underlying ICS processes or OT devices. Details of the methods used for monitoring operations can be found in Appendix A.4.

3.3 Scenario 3: Passive Discovery

This scenario focused on how a solution performs using exclusively passive methods to examine network traffic and extract information from that traffic. Each of the previous scenarios tested a solution's ability to identify assets using active scanning, whereas this scenario isolated the solution's passive capabilities.

CECA configured the runZero platform for passive traffic sampling by creating a mirror port on each firewall interface and sending the duplicated traffic to the Explorer in each subnet via a Generic Routing Encapsulation (GRE) tunnel. In addition to the mirrored traffic, each Explorer was also configured to listen to any broadcast traffic on its subnet. Once configured, the runZero platform was allowed to sample traffic for 30 minutes. Each asset in the environment involved in periodic communication did so with a frequency of at least several times per minute, so a period of 30 minutes allowed all active assets to generate network traffic many times over.

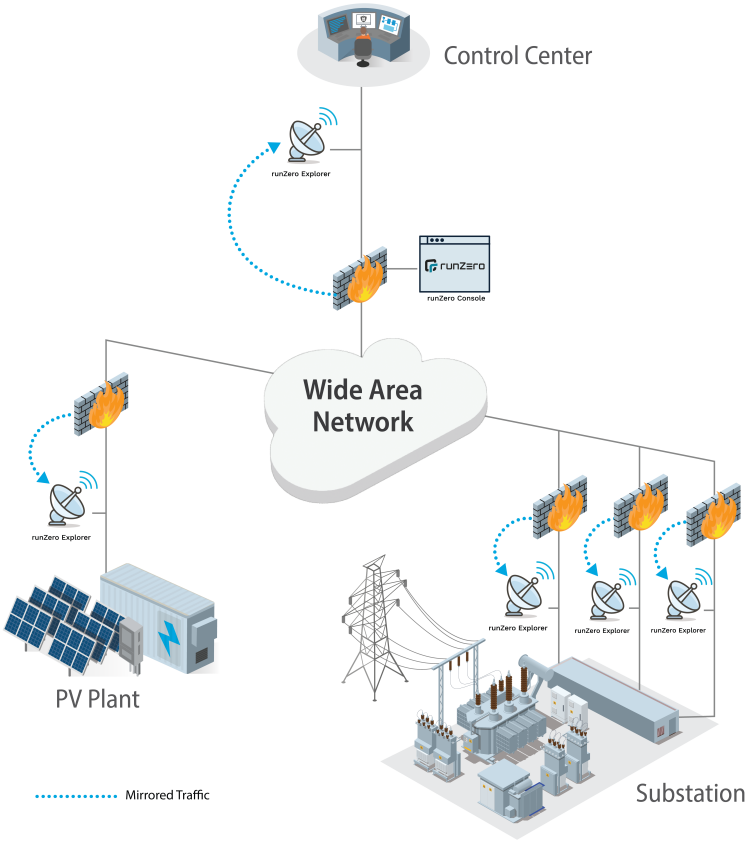


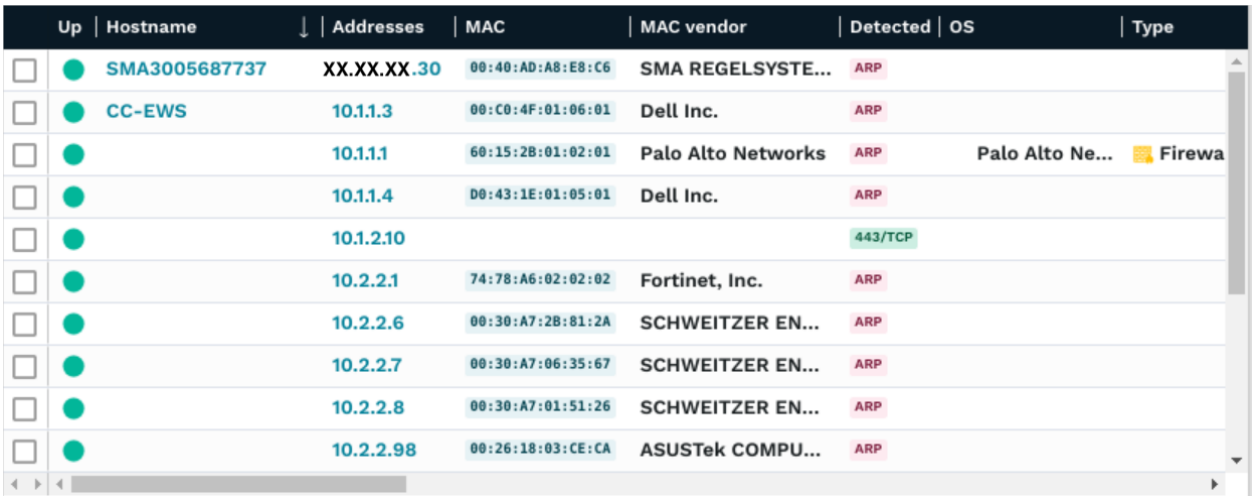
Figure 11. High-level overview showing the mirrored traffic to the runZero Explorers via GRE tunnels on the PV and substation environment

Inventory Accuracy

Compared to the previous active scans, passive sampling was only able to identify a subset of devices and attributes. This is to be expected because active probes can interrogate devices for additional information, whereas passive sampling is subject to only listening to existing network traffic.[11] The limitations of passive sampling can be grouped into several themes:

- **Unable to identify quiet assets**: The solution did not identify assets in the system that did not generate any network traffic during the sampling period. This included the control center syslog and NTP servers, the substation SEL 3622 security gateway, and SEL 411L.

[11] Because passive sampling results depend on the system configuration and which sources of traffic are visible to the solution, additional sampling points and traffic flows could have provided deeper visibility.



Up	Hostname	Addresses	MAC	MAC vendor	Detected	OS	Type
●	SMA3005687737	XX.XX.XX.30	00:40:AD:A8:E8:C6	SMA REGELSYSTE...	ARP		
●	CC-EWS	10.1.1.3	00:C0:4F:01:06:01	Dell Inc.	ARP		
●		10.1.1.1	60:15:2B:01:02:01	Palo Alto Networks	ARP	Palo Alto Ne...	Firewa
●		10.1.1.4	D0:43:1E:01:05:01	Dell Inc.	ARP		
●		10.1.2.10			443/TCP		
●		10.2.2.1	74:78:A6:02:02:02	Fortinet, Inc.	ARP		
●		10.2.2.6	00:30:A7:2B:81:2A	SCHWEITZER EN...	ARP		
●		10.2.2.7	00:30:A7:06:35:67	SCHWEITZER EN...	ARP		
●		10.2.2.8	00:30:A7:01:51:26	SCHWEITZER EN...	ARP		
●		10.2.2.98	00:26:18:03:CE:CA	ASUSTek COMPU...	ARP		

Figure 12. runZero asset inventory from passive collection

- **Unable to identify network infrastructure**: Any switches such as those in the PV plant and substation were not identified because they simply switch traffic and do not have any signatures at any of the points where the passive sampling was conducted. In addition, the runZero platform did not have enough information to combine the observations taken from different perspectives into a single asset. Specifically, the substation firewall, which had three different interfaces—`10.2.2.1`, `10.2.3.1`, and `10.2.4.1`—was listed as three different assets despite being three interfaces on the same firewall. The runZero platform did not have enough information from the passive traffic to correlate these three interfaces to the same device.
- **Only able to identify signatures that traverse the sampling point**: The platform was unable to identify the SEL 751 (IP XX.XX.XX.13) in the PV plant. Although this device was generating network traffic, it was only communicating with the SEL 3530 Real-Time Automation Controller (RTAC) in the PV plant, and therefore passive sampling at the PV plant firewall was unable to detect this local traffic.

Data Richness

Data richness for passive sampling is reduced to reflect only what can be observed at the traffic sampling points. IP and MAC addresses (and MAC vendors) were recorded for every asset identified. The host name for the engineering workstation (running Windows 10) and the System, Mess and Anlagentechnik Solar Technology AG (SMA) inverter were identified. "Noisy" services—such as ARP, Common Internet File System (CIFS), and NetBIOS—were identified for the IT assets; Multicast Domain Name System (MDNS) was identified for the SMA inverter on port 5353; and TLS was identified for the runZero platform on port 443.[12]

[12]The runZero platform at 10.1.2.10 is seen only as the destination address for traffic generated by other hosts, which is why it does not have an associated MAC address.

Table 15. Scenario 3 data richness

Device	Hostname	IP Address	MAC Address	MAC Vendor	OS	OS Version
Control center (7 devices)						
cc firewall		✓	✓	✓	✓	
SCADA platform		✓	✓	✓		
Engineering workstation	✓	✓	✓	✓		
*† **NTP server**						
*† **Syslog server**						
† **cc admin vm**	✓	✓	✓	✓		
cc runZero vm	✓	✓	✓	✓		
Substation OT (11 devices)						
Sub firewall		✓	✓	✓		
* **OT switch**						
Sub-ot admin vm	✓	✓	✓	✓		
Sub-ot runZero vm	✓	✓	✓	✓		
SEL 3530 RTAC		✓	✓	✓		
*† **SEL 411L**						
SEL 751		✓	✓	✓		
SEL 351A		✓	✓	✓		
* **SEL 311C**						
* **SEL 321**						
* **SEL 501**						
Substation OT gateway (3 devices)						
Sub-ot-gateway admin vm	✓	✓	✓	✓		
Sub-ot-gateway runZero vm	✓	✓	✓	✓		
† **SEL 3622**						
Substation IT (4 devices)						
* **IT switch**						
† **Sub-it admin vm**	✓	✓	✓			
Sub-it runZero vm	✓	✓	✓			
† **Workstation**		✓	✓	✓		
PV plant (8 devices)						
PV firewall		✓	✓	✓		
† **PV admin vm**	✓	✓	✓	✓		
PV runZero vm	✓	✓	✓	✓		
SEL 3530 RTAC		✓	✓	✓		
SEL 751						
SEL 751		✓	✓	✓		
SMA Sunny Highpower	✓	✓	✓	✓		
ELM BESS		✓	✓	✓		
Total (of 33 devices)	12	23	23	23	1	0

* Device not identified
† No application traffic

Additional Network Traffic

The additional network traffic displayed in Table 16 only pertains to runZero Explorers communicating with the platform. The Explorer communicates a minuscule amount of information with the platform relative to the quantity of data ingested. It appears that runZero Explorer processes sampled traffic locally and only reports refined data to the console/platform. This minimizes the additional network traffic that the runZero console/platform adds to a system.

Table 16. Explorer network traffic to platform in Scenario 3

Subnet	Number of Hosts	Average	Standard Deviation
Control center	7	217 kB	20 kB
Substation IT	4	237 kB	28 kB
Substation OT gateway	3	234 kB	32 kB
Substation OT	11	215 kB	24 kB
PV plant	8	247 kB	52 kB

3.4 Scenario 4: Scale Discovery

This scenario focused on how a solution performs at scale. The previous three scenarios were all run in the same PV and substation environment with several tens of devices. To stress the solution and test how it performs at scale, CECA created the AMI environment with 3,948 different AMI devices in a single "flat" subnet (`/20` in classless inter-domain routing (CIDR) notation).The AMI devices in this scenario are not configured with any underlying processes that generate baseline traffic. In a real-world scenario, the devices would be communicating and generating network traffic, adding more "noise" to the network and, in turn, adding latency to the network performance. To minimize the variables being tested, no additional network traffic was added.

Scenario 4 consisted of two consecutive scans. First, the solution was activated to identify all of the existing assets in the environment without any previous knowledge. Second, a single additional device was added to the network, and the solution was again activated to identify all assets in the environment, including the new device. In both scans, the solution used the default settings, as in Scenario 1B.

Scenario 4 magnifies both the time that the solution takes to identify a single device and the amount of additional network traffic that the solution adds to the infrastructure when identifying a single asset. In addition, Scenario 4 provides an opportunity to test a solution's ability to identify a new device in a much larger environment. Note that Scenario 4 was not evaluating the accuracy or richness of the data identified.

Alert

The runZero platform generated alerts at the conclusion of each run when it identified the newly added device. The platform found the new device in each iteration.

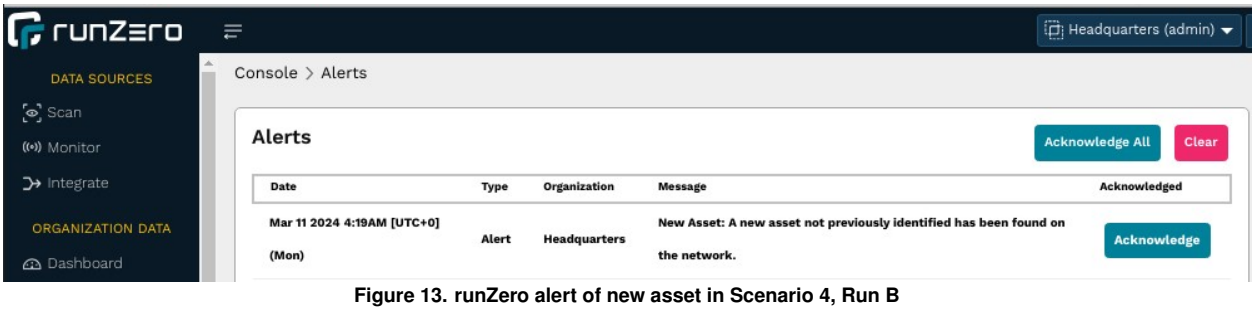


Figure 13. runZero alert of new asset in Scenario 4, Run B

Timing

The average time for Scan A was 8,202 seconds (just under 2 hours, 17 minutes) and 8,152 seconds (just under 2 hours, 16 minutes) for Scan B. runZero's scanning rate is configurable. CECA used the default runZero scan rate (1,000) and the default maximum host rate and groups size (40 and 4,096, respectively). Increasing these settings could have led to faster scan times.

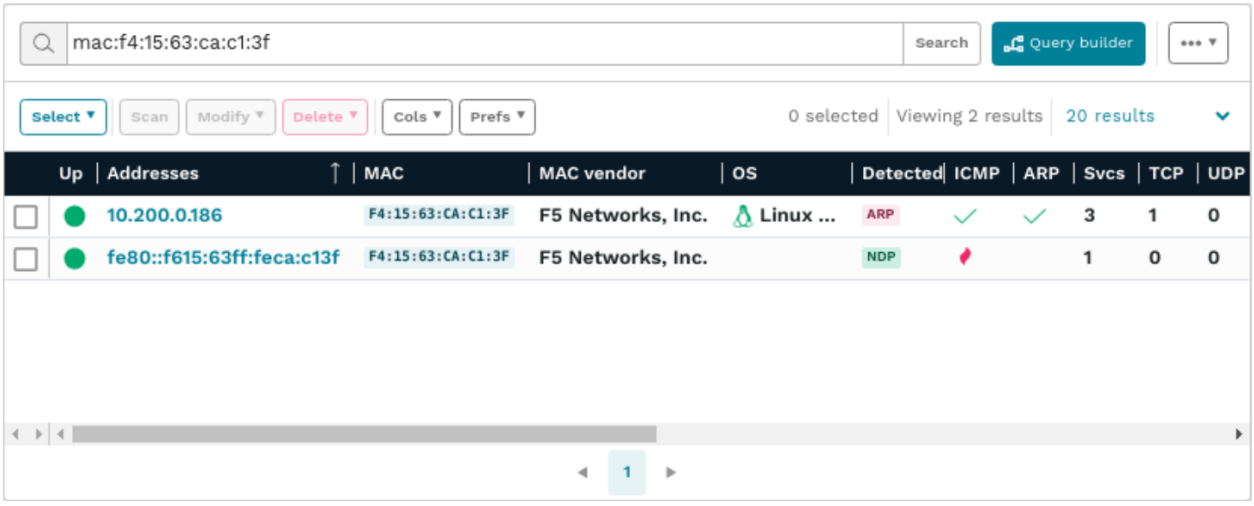


Figure 14. runZero asset inventory displaying the same device discovered via ARP and NDP

Duplicate Devices

Inconsistencies in the total number of devices identified during each run were observed, with each run varying between one to three extra devices. Further investigation found that these extra devices were identified via Neighbor Discovery Protocol (NDP), not ARP, which was the case for all other devices identified during the scan. NDP is an Internet Protocol, Version 6 (IPv6)-specific protocol that serves functions similar to ARP. Each extra entry identified by NDP had a corresponding entry identified via ARP, with a matching MAC address across both entries. It is suspected that these two different discovery methods led to duplicate entries for the same device. In each case, researchers accounted for the deviation and confirmed that the duplicate entities were properly categorized and did not affect the overall results of the test.

Additional Network Traffic

The average network traffic shown in Table 17 and Table 18 amounted to approximately 170 kB for each device identified. This would change based on which probes are enabled in the scan profile.

Table 17. Explorer network traffic on subnets in Scenario 4

Scan	Number of Hosts	Average	Standard Deviation
Scan A	3,950	674,916 kB	3,408 kB
Scan B	3,951	672,409 kB	4,089 kB

Table 18. Explorer network traffic to platform in Scenario 4

Scan	Number of Hosts	Average	Standard deviation
Scan A	3,950	39,284 kB	17,9991 kB
Scan B	3,951	39,359 kB	82,8232 kB

4 Conclusion

Utility ICS networks can be vast, geographically dispersed systems that comprise a heterogeneous set of devices and ICS protocols. These characteristics compound the inability for asset owners to accurately appraise which devices (known and unknown) are connected to their network and to truly understand which risks they face and how those risks emerge and evolve with their environment. runZero represents one product in a class of solutions designed to help asset owners enumerate their assets and understand potential risks while maintaining normal business operations.

CECA evaluations in Cohort 2 tested the runZero product across a range of scenarios. These tests demonstrated that the runZero product was consistently able to identify all IP-addressable assets in the environment. Beyond just identifying devices, the runZero product demonstrated the capability to collect detailed information about each device and all open ports. These capabilities even extend to identifying the presence of some OT protocols, such as Modbus. Across tests, CECA observed the runZero scans adding an additional 170 kB to 1,200 kB of network traffic per active host within the scope of the scans [13] and increasing times by approximately 2 to 9 seconds per active host within the scope of the scans. The lower end of both ranges represents a larger environment with a much higher address-space utilization rate, and the higher end was observed in environments with a lower address-space utilization rate relative to the scope of the scans.

CECA tests demonstrated that runZero's active scanning methods had no adverse effects on the deployed ICS assets or ongoing SCADA processes and communications. CECA tested runZero against varied ICS protocols and devices, to validate the conclusions to the greatest degree possible; however, these conclusions cannot be assumed to be generalizable simply due to the sample of devices and protocols being limited by the time and availability to perform the tests. These results indicate that active scanning could be a viable solution for identifying assets without adversely impacting ICS operations. This conclusion should reduce exclusive reliance on passive collection methods by addressing concerns about active scanning disrupting operations.

Challenges still exist for the broader class of asset identification technologies in the ICS space, including visibility of assets connected via legacy media like serial connections, identification of assets that are not IP addressable, and visibility into assets connected behind an RTU that does not forward traffic to subordinate devices. Solving each of these problems likely requires vendor-specific methods for credentialed identification, or manual operator actions.

Cybersecurity is a complex and shifting field full of unique challenges. Threats, risks, architectures, and technologies will continue to evolve as the energy sector undergoes significant transformations. Innovation of solutions should be enabled to evolve as well. There will always be widespread challenges in industry that solution providers are aiming to solve. Using solutions such as those offered by runZero to identify control system assets and to monitor changes in that equipment is expected to improve the security of the industry as a whole.

[13] The runZero product is configurable, and scans can be rate-limited to spread this traffic over time and prevent any large spikes in data rates in sensitive networks.

References

Dragos. 2019. “Key Considerations for Selecting an Industrial Cybersecurity Solution for Asset Identification, Threat Detection, and Response.” https://www.dragos.com/wp-content/uploads/Key-Considerations-Industrial-Cybersecurity-Solution.pdf.

Hanka, T., M. Niedermaier, F. Fischer, S. Kießling, P. Knauer, and D. Merli. 2020. “Impact of Active Scanning Tools for Device Discovery in Industrial Networks.” *Security, Privacy and Anonymity in Computation, Communication and Storage* 12383:557–572. https://doi.org/10.1007/978-3-030-68884-4_46.

Modbus Organization. 2006a. “MODBUS Messaging on TCP/IP Implementation Guidea,” October. https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf.

Modbus Organization. 2006b. “MODBUS over Serial Line Specification and Implementation Guide,” December. https://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf.

Modbus Organization. 2012. “MODBUS Application Protocol Specification,” April. https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

National Institute of Standards and Technology. 2020. “Energy Sector Asset Management for electric utilities, oil ...” https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-23.pdf.

National Renewable Energy Laboratory. 2024. “ARIES Cyber Range.” https://www.nrel.gov/security-resilience/cyber-range.html.

OpenJS Foundation. 2024. “Node-RED.” https://nodered.org/.

Patria Security, LLC. 2023. “Operational Technology (OT) Simulator.” https://ot-sim.patsec.dev/.

Patria Security, LLC. 2024a. “phenix documentation.” https://phenix.sceptre.dev/latest/scorch/.

Patria Security, LLC. 2024b. “phenix documentation.” https://phenix.sceptre.dev/latest/state-of-health/.

Patria Security, LLC. 2024c. “phenix documentation.” https://phenix.sceptre.dev/latest/apps/#vrouter-app.

Pospisil, O., P. Blazek, R. Fujdiak, and J. Misurec. 2021. “Active Scanning in the Industrial Control Systems.” *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC),* 227–232. https://doi.org/10.1109/ISCSIC54682.2021.00049.

runZero. 2024a. “runZero Command Line Scanner.” https://help.runzero.com/docs/using-the-scanner/.

runZero. 2024b. “runZero Scanning OT networks.” https://help.runzero.com/docs/playbooks/scanning-ot-networks/.

runZero. 2024c. “runZero Self Hosting.” https://help.runzero.com/docs/self-hosting/.

Sandia National Laboratories. 2023. “minimega.” https://www.sandia.gov/minimega/.

Sandia National Laboratories. 2024a. “minimega github.” https://github.com/sandia-minimega/minimega.

Sandia National Laboratories. 2024b. “phenix.” https://phenix.sceptre.dev/latest/.

Sandia National Laboratories. 2024c. “sceptre phenix github.” https://github.com/sandialabs/sceptre-phenix.

Wallace, A., A. Liao, D. Rager, A. Hasandka, A. Sahu, N. Ryan, S. Drake, et al. 2024. *Cloud Zero Phase 2 Technical Report.* Technical report. Under submission. NREL. https://www.nrel.gov/docs/fy24osti/xxxx.pdf.

Appendix A. Baseline Operating Environment

A.1 Architecture Overview

The Cohort 2 solutions were tested in solution-specific operating environments built within a common BOE. The BOE describes the environment prior to the inclusion and configuration of the solution under test. The BOE included a control center, substation, and utility-owned PV plant. The Cohort 2 BOE is represented as a combination of virtual machine (VM)s and hardware devices. The control center comprised only VMs, whereas the substation and PV plant consisted of hardware components and virtual firewalls.

The BOE was deployed through the NREL's ARIES Cyber Range, a cyber-physical modeling and simulation platform that supports both virtual and physical deployments of variable-scale environments (NREL 2024). The ARIES Cyber Range leverages multiple open-source software packages to facilitate the design and deployment of experiments, networking, and virtual machines. Minimega is a VM manager that oversees the creation and startup of kernel-based virtual machine (KVM)s and software defined networking (SDN) used within the emulated environment (SNL 2023). Phenix sits above minimega in the software stack and orchestrates the organization and deployment of experiments and scenario executions from structured markup configuration files (SNL 2024b). Details about these tools can be found in Appendix B.

Through the deployment of an experiment on the ARIES Cyber Range, the requisite configurations and networking were set up to allow for repeatable evaluations and analyses of the generated data. The experiment BOE was designed to emulate a simple distribution system topology used by a utility or municipality to deliver power or grid services. To orient the BOE toward the cohort theme of device discovery, different asset types were used, and configurations were diversified to provide a clear understanding of the capabilities of each solution.

Several SEL power hardware assets were deployed, including relays, protection systems, communication devices, and control equipment (e.g., RTAC). The power assets connected were an SMA inverter and an ELM battery energy storage system (BESS). The substation also included IT elements, such as a workstation and printer, to represent such devices that are often present for workers to perform administrative tasks on-site. The OT devices were configured to use various protocols commonly seen in such environments for management and control. Each of the three sites—control center, substation, and PV plant—were connected virtually through the ARIES Cyber Range using a representative wide area network (WAN) built on top of several emulated routers participating in a common Open Shortest Path First (OSPF) area, similar to real-world WANs.

A.1.1 Control Center

The control center was designed with the minimal elements required to represent a basic set of services run by the emulated utility. All systems are VMs and run either Windows or Linux operating systems. The elements included are a SCADA platform running a Human-Machine Interface (HMI), an engineering workstation, and application servers. A DMZ was configured to allow for the isolated deployment of the solution providers' components within that space as needed.

A.1.2 Substation

The substation was designed to represent a geographically separated substation from the PV utility-owned site. In terms of power hardware elements, the substation included several SEL power system devices. There was also a Protectli Vault workstation computer and a printer connected to it via USB to represent on-site IT resources available at the substation. The only virtualized element of this site was the edge firewall that served as the connection point to the experiment environment.

A.1.3 PV Plant

The PV plant was designed to emulate a small utility-owned solar generation plant, and it included SEL power system devices, a BESS, and a commercial-grade inverter. A TerraSAS Module was connected to the inverter to provide an active direct current (DC) power source as well as input power and output demand set according to a predefined curve in the TerraSAS software. This connection enabled testing with the SMA device, but it only provided power values, so this device was not included as an identifiable asset in the testing environment. The only virtualized element of this site was the edge firewall that served as the connection point to the experiment environment.

CLEAN ENERGY CYBERSECURITY ACCELERATOR

BASELINE OPERATING ENVIRONMENT COHORT 2

Application Layer

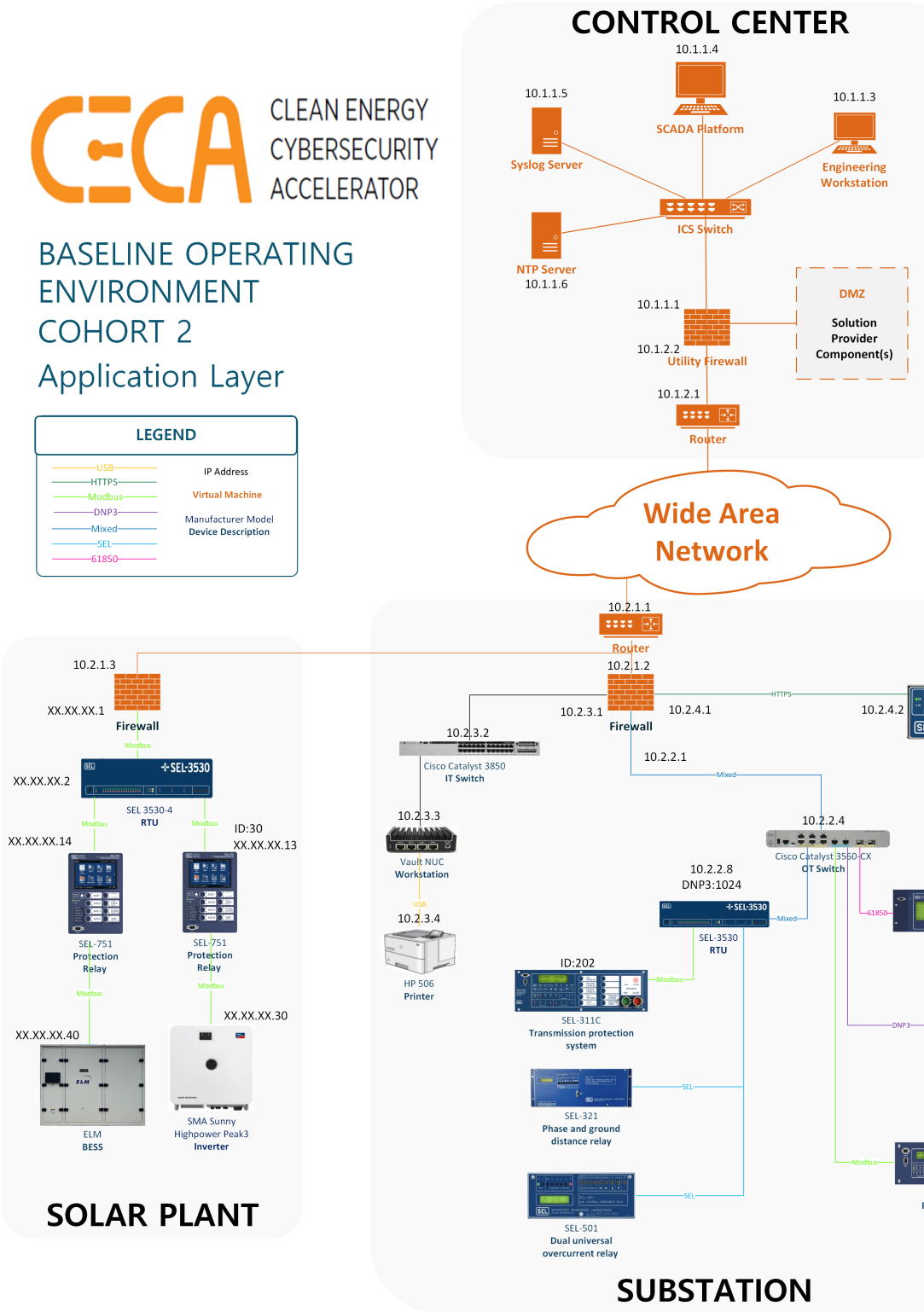


Figure A.1. Cohort 2 application layer BOE

A.2 Network

Seven subnets were used to segment the network. In the virtual environment, all subnets were connected with Internet Protocol Security (IPSec) tunnels over a virtual WAN with redundant internet service provider (ISP) routers.

A.2.1 Subnets

Subnet Name	Subnet	Description
CC-LAN	10.1.1.0/24	Dedicated to the control center network
CC-DMZ	10.1.2.0/24	Dedicated to the solution's components to restrict third-party access to the utility's control center network
SUB-DMZ	10.2.1.0/24	Covers the routers to the PV site and substation
SUB-OT	10.2.2.0/24	Covers the substation site's OT components, such as the relays RTU, and serial devices
SUB-OT-GATEWAY	10.2.4.0/24	Dedicated to the security gateway
SUB-IT	10.2.3.0/24	Covers the IT components in the substation
PV	XX.XX.XX.0/24	Covers the entire PV plant site

Table A.1. Subnets

A.2.2 Communication Protocols

Type	Protocol	Description
OT	DNP3	Commonly used for automation of various industrial systems and components. Examples of devices that are commonly seen to communicate using DNP3 are RTUs, inverters, smart meters, and other such devices related to electrical systems.
OT	Modbus/TCP	IP-based OT protocol that runs on top of the TCP protocol (Modbus 2006a). This protocol is a variant of the Modbus protocol specification (Modbus 2012) maintained by the Modbus Organization.
OT	Modbus/RTU	Serial OT protocol designed to communicate with devices connected over Recommended Standard 232 (RS232) or Recommended Standard 485 (RS485) interfaces (Modbus 2006b). It is a variant of the Modbus protocol specification (Modbus 2012) maintained by the Modbus Organization.
OT	SEL	Proprietary serial OT protocol developed by SEL for communication with devices manufactured by SEL.
IT	HTTP	HTTP and its more secure counterpart, Hypertext Transfer Protocol Secure (HTTPS), are IT protocols commonly used for web applications and general-purpose browsing. HTTPS provides enhanced security through the usage of certificates to encrypt and secure connections.
IT	SMB	Commonly used for sharing files on local or networked data storage.
NET	IPSec	IT security protocol commonly used to set up virtual private network (VPN)s or secure tunnels between two remote networks
NET	SNMPv2	Commonly used network protocol used to send messages related to the management of networked devices. Community public version used.
NET	OSPF	Routing protocol commonly used in local area network (LAN)s or involved in internet routing. It allows packets to traverse multiple interconnected networks to communicate across large areas and geographic regions.

Table A.2. Protocols

A.2.3 Firewall Rules

The firewalls specified in the BOE were configured with a default drop policy, and the following ingress allow rules:

Device	Description	Source	Destination	Port	Protocol
cc-firewall	Allow established				all
sub-firewall					
(to OT LAN)	Allow SCADA platform DNP3 to all OT	10.1.1.4	10.2.2.0/24	20000	TCP
	Allow SCADA platform Modbus to all OT	10.1.1.4	10.2.2.0/24	502	TCP
	Allow SCADA platform SEL protocol to all OT	10.1.1.4	10.2.2.0/24	23	TCP
	Allow established				all
(to IT LAN)	Allow established				all
pv-firewall	Allow SCADA platform Modbus to all OT	10.1.1.4	XX.XX.XX.0/24	502	TCP
	Allow established				all

Table A.3. Firewall rules

A.3 Assets

Table A.4. Asset list

Manufacturer	Model	Function	OS Type
N/A (VM)	Vyatta	Firewalls	VyOS
N/A (VM)	qcow2	SCADA platform	Windows
N/A (VM)	qcow2	Engineering workstation	Windows
N/A (VM)	qcow2	Syslog server	Linux
N/A (VM)	qcow2	NTP server	Linux
Cisco Systems	3850	PV switch	Cisco IOS
Cisco Systems	3560-CX	Substation switch	Cisco IOS
Hewlett Packard	506 Laserjet	Printer	Laserjet Enterprise
Protectli	FW4B	Workstation	Windows
SEL	311C	Transmission protection system	SEL Embedded Linux
SEL	321	Phase and ground distance relay	SEL Embedded Linux
SEL	351A	Protection system	SEL Embedded Linux
SEL	3530	RTAC	SEL Embedded Linux
SEL	3530-4	RTAC	SEL Embedded Linux
SEL	3622	Security gateway	SEL Embedded Linux
SEL	411L	Feeder relay	SEL Embedded Linux
SEL	501	Dual universal overcurrent relay	SEL Embedded Linux
SEL	751	Protection relay	SEL Embedded Linux
ELM	501	BESS	Windows
SMA	SHP 125-US-20	PV inverter	Linux

A.4 Monitoring

A.4.1 Pretesting

Before each test, the phenix state of health (SoH) app[14] was used to validate the environment configuration and to ensure that all assets were available and communicating.

A.4.2 During Testing

To verify that each OT device continued to perform its intended function throughout each test, and that no underlying ICS processes were affected by active scanning, CECA used two different techniques. First, assets that were reachable via ICMP were polled every second, and response latency was monitored.

Second, the data acquisition portion of the environment's SCADA processes was monitored to ensure that no communications were unavailable throughout the duration of each test. The OT devices in the PV plant and substation each had several registers monitored by the SCADA platform in the control center. These values were polled via

[14]For more information on phenix apps, see Appendix B

DNP3, Modbus, or the SEL Protocol (in some instances forwarded or aggregated by an intermediate RTU), at a frequency of once every 5 seconds. During normal environment operations, these values were monitored by the SCADA platform and displayed on the accompanying control center Node-RED HMI. At the end of each test, the logs from the SCADA platform were retrieved and checked for any failures to read from an outstation device.

Appendix B. Evaluation Tools

The assessment was conducted in the NREL ARIES Cyber Range, which uses a collection of open-source and custom tools to emulate complex ICS systems.

B.1 Minimega

Minimega is an open-source tool for starting and managing multiple VMs (SNL 2023) (SNL 2024a). Minimega is based on the quick emulator (QEMU) hypervisor.

B.2 Phenix

Phenix is an open-source application wrapping multiple tools that orchestrates the definition, configuration, deployment, and management of VMs, scenarios, and hardware into various experiments. Phenix flexibly integrates virtualized and hardware components into environments and can be customized using the phenix application framework. Several built-in and custom phenix applications used by CECA are detailed in the following (SNL 2024b) (SNL 2024c).

B.2.1 ScOrch (Core App)

ScOrch (SCenario ORCHestration) is an automated scenario orchestration framework within phenix. ScOrch provides the ability to create customizable attack and data collection pipelines for efficient and repeatable assessment (Patria 2024a). These pipelines can launch Atomic Red Team and other command-based attacks in addition to instrumenting assessment data collection during the scenario to enable subsequent analysis. Using ScOrch allows for entire evaluation scenarios to be documented in a file and creating an assessment as code (AaC) methodology.

B.2.2 State of Health (Core App)

The SoH app continuously collects the state of components in the virtual environment (Patria 2024b). It visually renders the state using a network graph for quick overview. A set of predefined measurements—central processing unit (CPU) load, open ports, running processes, reachability, etc.—can be gathered on the VMs as part of the SoH test. Custom tests specific to the environment can also be configured. SoH simplifies the monitoring of the complex experiment with both virtual and hardware devices connected through the ARIES Cyber Range infrastructure.

B.2.3 Vrouter (Core App)

Virtual Router (vrouter) is a phenix app that enables the automated configuration of routers and firewalls in phenix experiments (Patria 2024c). The vrouter app can configure IPSec tunnels, firewall rules, dynamic host configuration protocol (DHCP) settings, domain name system (DNS) entries, and source network address translation (SNAT) or destination network address translation (DNAT), and it can add custom traffic profile emulations to any router running in a phenix experiment.

B.2.4 AMI (Custom App)

The custom AMI phenix app allows researchers to instantiate thousands of small IoT-like devices in minutes. Each device is a minimega container possessing a (TCP)/(IP) networking stack for switching and routing like any networking device would require to function on ethernet. The application reads a configuration from an underlying power model and determines a set number of containers required to represent the devices within it (Wallace et al. 2024). It creates a number of VMs on which to initiate the containers. For example, in Scenario 4, the app was used to deploy 27 VMs with 150 containers, totaling exactly 3,948 AMI-representative devices in the experiment. Each device functioned as a smart meter with networking capability. This virtualization allowed for many more networked endpoints to be included in the environment than in previous tests.

B.3 OT-sim

The Operational Technology Simulator (OT-sim) is a software tool developed by Patria Security LLC to simulate various components of an OT device using a module-based approach (Patria 2023). OT-sim is deployed as a set of binaries, each for a module that can be configured to run a simulated component of an OT device within VMs or containers. Through deployment of this tool, along with any necessary configurations in an automated manner through

phenix, the ARIES Cyber Range allows researchers to represent a physical system, at scale, in a co-simulation environment. The specific OT-sim modules used in this experiment were: CPU, DNP3, Modbus, and Node-RED. Together, they were deployed in a configuration serving as a HMI for the physical devices in the experiment.

B.4 Node-RED

Node-RED is a flow-based programming tool developed by the OpenJS Foundation (OpenJS 2024) that provides a browser-based editor for developing applications that have a user-interfacing dashboard. The tool works with hardware devices, application programming interfaces, and other peripheral interfaces using a software plug-in framework. Node-RED provides both a graphical user interface (GUI) for development of applications and dashboards and simple user access control for deployed applications using passwords and configurable user credentials. Configurations can also be exported and imported as javascript object notation (JSON) files. Doing so allows for easy modification of templates because the modified JSON file can be injected into the VM running Node-RED each time an experiment is started. For the CECA Cohort 2 experiments, Node-RED was deployed through an OT-sim module and configured using the JSON import method; however, the software is capable of dynamic modification of deployed flows and provides a platform for users to interact with the deployed web applications in real time.

Appendix C. Configuration of Technology

C.1 Version

The CECA Cohort 2 testing used the self-hosted runZero platform version 4.0.240301.0 and the accompanying Explorers version 4.0.240301.0.

C.2 Installation

CECA followed the runZero documentation to install the platform according the directions on the runZero website (runZero 2024c) and create an initial user (ceca@nrel.gov). Once the platform was installed, an organizational API token was created, and an Explorer binary was downloaded for injection into each Explorer VM during testing. All tests started from this initial state.

C.3 API

The runZero API was heavily leveraged to enable CECA's methodology to allow repeatable runs of each test. For details about all API calls, see Appendix D - Procedures.

C.4 Task Profiles

The following task profiles were used for the tests.[15] runZero groups both active identification (referred to as a scan in runZero nomenclature) and passive identification (referred to as a sample) as tasks. Each task described here is a template that was copied and updated at run time to customize the scan profile to the Explorer and subnet that it was destined to discover. Updates were applied to the following fields:

- `scan-start` to match the current epoch time so that the scan would start
- `agent` and `explorer` to be specific to one of the five Explorers that were checked in to the platform (one in each subnet of interest)
- `targets` to match the appropriate `/24` subnet of interest for the relevant Explorer
- `name` and `tags` to add the specific Explorer/subnet name.

Each task profile was used universally across all subnets.[16]

C.4.1 OT Limited Scan

The OT Limited scan is taken from the runZero playbook for scanning OT networks and is designed to "determine what is alive on your OT network while minimizing the volume of traffic generated and avoiding proprietary or ICS ports during initial scanning"(runZero 2024b). [15]

```
{
  "scan-name": "CECA-CH2-Sc1B",
  "scan-tags": "scenario=1B",
  "scan-description": "CECA Cohort 2, Scenario 1B - A limited OT Scan",
  "excludes": "defaults",
  "host-ping": "true",
  "host-ping-probes":
   ↪  "arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns",
  "layer2-add-targets": "true",
  "layer2-force": "false",
```

[15] The CECA testing was conducted under the scheme Sc1A = Default, Sc1B = OT Limited, Sc1C = OT Full. These were reordered in this report to Sc1A = OT Limited, Sc1B = Default, Sc1C = OT Full to match a scheme of progressively more in-depth scans.

[16] runZero is extremely customizable, and different scan profiles can be applied to each scan. In a real deployment, it would be more realistic to use one scan profile for the control center subnet and a different profile for OT subnets. For simplicity of testing, CECA used the same profile for all scans in a test.

```
"layer2-max-retries": "2",
"layer2-tcp-ports": "22,80,135,179,443,3389,5040,7547,62078",
"layer2-udp-trace-port": "9",
"max-attempts": "3",
"max-group-size": "2048",
"max-host-rate": "20",
"max-sockets": "2048",
"max-tos": "0",
"max-ttl": "64",
"nameservers": "",
"netbios-port": "137",
"ntp-port": "123",
"passes": "1",
"probes": "arp,layer2,netbios,ntp,snmp,ssh,syn,connect,tftp",
"rate": "500",
"screenshots": "true",
"server-time": "0",
"snmp-comms": "private,public",
"snmp-disable-bulk": "false",
"snmp-max-repetitions": "16",
"snmp-max-retries": "1",
"snmp-poll-interval": "300",
"snmp-port": "161",
"snmp-timeout": "5",
"snmp-v3-auth-passphrase": "",
"snmp-v3-auth-protocol": "",
"snmp-v3-context": "",
"snmp-v3-privacy-passphrase": "",
"snmp-v3-privacy-protocol": "",
"snmp-v3-username": "",
"snmp-walk-timeout": "60",
"ssh-fingerprint": "true",
"ssh-fingerprint-username": "_STATUS_",
"subnet-ping": "false",
"subnet-ping-net-size": "256",
"subnet-ping-probes":
 ↪  "arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns",
"subnet-ping-sample-rate": "3",
"syn-disable-bogus-filter": "false",
"syn-forwarding-check": "false",
"syn-forwarding-check-target": "13.248.161.247",
"syn-max-retries": "2",
"syn-max-sockets": "2048",
"syn-ports": "21,22,23,69,80,123,135,137,161,179,443,445,3389,5040,590
 ↪  0,7547,8080,8443,62078,65535",
"syn-report-resets": "true",
"syn-reset-sessions": "true",
"syn-reset-sessions-delay": "0",
"syn-reset-sessions-limit": "50",
"syn-traceroute": "true",
"syn-udp-trace-port": "9",
"targets": "10.0.0.0/8",
"tcp-excludes": "",
```

```
  "tcp-ports": "21,22,23,69,80,123,135,137,161,179,443,445,3389,5040,590
   ↪  0,7547,8080,8443,62078,65535",
  "tftp-ports": "69",
  "tos": "0"
}
```

Listing 1. OT Limited runZero scan profile, tuned to be as safe as possible in OT environments

C.4.2 Default Scan

The default scan uses runZero's default probes and values.[15]

```
{
  "scan-name": "CECA-CH2-Sc1A",
  "scan-tags": "scenario=1A",
  "scan-description": "CECA Cohort 2, Scenario 1A - A basic scan based
   ↪  on default values",
  "arp-fast": "false",
  "aws-instances-access-key": "",
  "aws-instances-assume-role-name": "",
  "aws-instances-delete-stale": "false",
  "aws-instances-exclude-unknown": "false",
  "aws-instances-include-stopped": "false",
  "aws-instances-regions": "",
  "aws-instances-secret-access-key": "",
  "aws-instances-service-options": "defaults",
  "aws-instances-site-per-account": "false",
  "aws-instances-site-per-vpc": "false",
  "aws-instances-token": "",
  "azure-client-id": "",
  "azure-client-secret": "",
  "azure-exclude-unknown": "false",
  "azure-multi-subscription": "",
  "azure-password": "",
  "azure-service-options": "defaults",
  "azure-site-per-subscription": "false",
  "azure-subscription-id": "",
  "azure-tenant-id": "",
  "azure-username": "",
  "azuread-client-id": "",
  "azuread-client-secret": "",
  "azuread-exclude-unknown": "false",
  "azuread-include-inactive": "false",
  "azuread-password": "",
  "azuread-service-options": "defaults",
  "azuread-tenant-id": "",
  "azuread-username": "",
  "bacnet-ports": "46808,47808,48808",
  "bedrock-ports": "19132",
  "censys-api-url": "",
  "censys-client-id": "",
  "censys-client-secret": "",
```

```
"censys-exclude-unknown": "false",
"censys-mode": "assets",
"censys-query": "",
"clock-offset": "0",
"coap-port": "5683",
"crestron-port": "41794",
"crowdstrike-api-url": "",
"crowdstrike-client-id": "",
"crowdstrike-client-secret": "",
"crowdstrike-exclude-unknown": "false",
"crowdstrike-filter": "",
"crowdstrike-fingerprint-only": "false",
"crowdstrike-risks": "None,Low,Medium,High,Critical",
"crowdstrike-severities": "Info,Low,Medium,High,Critical",
"dahua-dhip-ports": "37810",
"defender365-client-id": "",
"defender365-client-secret": "",
"defender365-exclude-unknown": "false",
"defender365-include-inactive": "false",
"defender365-tenant-id": "",
"dnp3-address-probe-timeout": "30",
"dnp3-banner-address-discovery": "ignore",
"dnp3-destination-address-discovery-range": "0-32",
"dnp3-explorer-address": "-1",
"dns-disable-google-myaddr": "false",
"dns-disable-meraki-detection": "true",
"dns-port": "53",
"dns-resolve-name": "off",
"dns-trace-domain": "off",
"dtls-ports": "443,3391,4433,5246,5349,5684",
"echo-report-errors": "false",
"ethernetip-use-tagged-context": "false",
"excludes": "",
"fins-port": "9600",
"gcp-exclude-unknown": "false",
"gcp-service-options": "defaults",
"gcp-site-per-project": "false",
"genudp-payload-base64": "",
"genudp-payload-hex": "",
"genudp-payload-text": "",
"genudp-ports": "",
"googleworkspace-client-email": "",
"googleworkspace-client-id": "",
"googleworkspace-customer-id": "",
"googleworkspace-delegate": "",
"googleworkspace-exclude-unknown": "false",
"googleworkspace-private-key": "",
"googleworkspace-private-key-id": "",
"googleworkspace-project-id": "",
"googleworkspace-service-options": "defaults",
"hiddiscoveryd-port": "4070",
"host-ping": "false",
```

```
"host-ping-probes":
 ↪  "arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns",
"igel-discovery-ports": "30005",
"ike-port": "500",
"insightvm-api-url": "",
"insightvm-exclude-unknown": "false",
"insightvm-fingerprint-only": "false",
"insightvm-insecure": "",
"insightvm-password": "",
"insightvm-risks": "None,Low,Medium,High,Critical",
"insightvm-severities": "Info,Low,Medium,High,Critical",
"insightvm-thumbprints": "",
"insightvm-username": "",
"intune-client-id": "",
"intune-client-secret": "",
"intune-exclude-unknown": "false",
"intune-password": "",
"intune-tenant-id": "",
"intune-username": "",
"ipmi-port": "623",
"kerberos-port": "88",
"knxnet-ports": "3671",
"l2t-port": "2228",
"l2tp-ports": "1701",
"lantronix-port": "30718",
"layer2-add-targets": "true",
"layer2-force": "false",
"layer2-max-retries": "2",
"layer2-tcp-ports": "22,80,135,179,443,3389,5040,7547,62078",
"layer2-udp-trace-port": "9",
"ldap-base-dn": "",
"ldap-exclude-unknown": "false",
"ldap-insecure": "",
"ldap-legacy-tls": "",
"ldap-password": "",
"ldap-service-options": "defaults",
"ldap-thumbprints": "",
"ldap-url": "",
"ldap-username": "",
"max-attempts": "3",
"max-group-size": "4096",
"max-host-rate": "40",
"max-sockets": "2048",
"max-tos": "0",
"max-ttl": "255",
"mdns-port": "5353",
"memcache-port": "11211",
"miradore-api-key": "",
"miradore-exclude-unknown": "false",
"miradore-hostname": "",
"modbus-identification-level": "regular",
"mssql-port": "1434",
"nameservers": "",
```

```
"natpmp-port": "5351",
"nessus-access-key": "",
"nessus-api-url": "",
"nessus-exclude-unknown": "false",
"nessus-fingerprint-only": "false",
"nessus-insecure": "",
"nessus-risks": "None,Low,Medium,High,Critical",
"nessus-secret-key": "",
"nessus-severities": "Info,Low,Medium,High,Critical",
"nessus-thumbprints": "",
"netbios-port": "137",
"nopcap": "false",
"ntp-port": "123",
"openvpn-ports": "1194",
"passes": "1",
"pca-port": "5632",
"ping-only": "false",
"probes": "arp,arp,aws-instances,azure,azuread,bacnet,bedrock,censys,c
 ↪  oap,connect,connect,crestron,crowdstrike,dahua-dhip,defender365,dn
 ↪  p3,dns,dtls,echo,ethernetip,fins,gcp,genudp,googleworkspace,hiddis
 ↪  coveryd,igel-discovery,ike,insightvm,intune,ipmi,kerberos,knxnet,l
 ↪  2t,l2tp,lantronix,layer2,ldap,mdns,memcache,miradore,modbus,mssql,
 ↪  natpmp,nessus,netbios,ntp,openvpn,pca,psdisco,qualys,rdns,rpcbind,
 ↪  s7comm,sample,sentinelone,shodan,sip,snmp,ssdp,ssh,steam,syn,tenab
 ↪  le,tenablesecuritycenter,tftp,ubnt,vmware,webmin,wlan-list,wsd",
"psdisco-ports": "987,9302",
"qualys-api-url": "",
"qualys-exclude-unknown": "false",
"qualys-fingerprint-only": "false",
"qualys-include-unscanned": "false",
"qualys-password": "",
"qualys-risks": "None,Low,Medium,High,Critical",
"qualys-severities": "Info,Low,Medium,High,Critical",
"qualys-username": "",
"rate": "1000",
"rdns-max-concurrent": "64",
"rdns-timeout": "3",
"rpcbind-port": "111",
"rpcbind-port-nfs": "2049",
"s7comm-request-extended-information": "false",
"sample-duration": "300",
"sample-excludes": "",
"sample-interfaces": "",
"sample-targets": "10.0.0.0/8 169.254.0.0/16 172.16.0.0/12
 ↪  192.168.0.0/16",
"scan-mode": "host",
"scanner-name": "main",
"screenshots": "true",
"sentinelone-api-url": "",
"sentinelone-client-id": "",
"sentinelone-client-secret": "",
"sentinelone-exclude-unknown": "false",
"server-time": "1709225884520333457",
```

```
"shodan-api-key": "",
"shodan-exclude-unknown": "false",
"shodan-mode": "assets",
"shodan-query": "",
"sip-port": "5060",
"site_id": "aa70a2c4-4d08-41ab-acbd-03979e8c5bfb",
"skip-broadcast": "true",
"snmp-comms": "private,public",
"snmp-disable-bulk": "false",
"snmp-max-repetitions": "16",
"snmp-max-retries": "1",
"snmp-poll-interval": "300",
"snmp-port": "161",
"snmp-timeout": "5",
"snmp-v3-auth-passphrase": "",
"snmp-v3-auth-protocol": "",
"snmp-v3-context": "",
"snmp-v3-privacy-passphrase": "",
"snmp-v3-privacy-protocol": "",
"snmp-v3-username": "",
"snmp-walk-timeout": "60",
"ssdp-port": "1900",
"ssh-fingerprint": "true",
"ssh-fingerprint-username": "_STATUS_",
"steam-ports": "27036",
"subnet-ping": "false",
"subnet-ping-net-size": "256",
"subnet-ping-probes":
 ↪  "arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns",
"subnet-ping-sample-rate": "3",
"syn-disable-bogus-filter": "false",
"syn-forwarding-check": "false",
"syn-forwarding-check-target": "13.248.161.247",
"syn-max-retries": "2",
"syn-report-resets": "true",
"syn-reset-sessions": "true",
"syn-reset-sessions-delay": "0",
"syn-reset-sessions-limit": "50",
"syn-traceroute": "true",
"syn-udp-trace-port": "9",
"targets": "10.0.0.0/8",
"tcp-excludes": "",
```

```
"tcp-ports": "1514,5392,4848,12345,5400,8850,16102,113,5984,55580,6905
 ↪  ,9524,10050,10080,1030,5061,5355,5433,32844,37718,34964,4679,6080,
 ↪  6101,8161,25,53,69,993,8901,17775,48899,5683,7443,9060,54923,25672
 ↪  ,445,5222,5908,7070,1583,9200,55553,617,3299,6503,17185,2222,5347,
 ↪  5900,664,1102,1443,2083,2100,6000,6002,8303,8890,32913,50051,21,50
 ↪  93,6106,7800,50070,52302,444,4433,27888,51443,5250,6405,7787,18881
 ↪  ,500,995,1811,3269,40317,1582,2199,8012,8081,1494,7080,8014,34962,
 ↪  10628,12397,28017,38010,407,3220,4730,5051,11211,5351,5353,8172,81
 ↪  81,5907,9099,9600,12174,38080,41524,3037,5903,8089,8545,9530,23472
 ↪  ,903,3260,3351,5988,54321,135,384,2181,25565,921,2074,12203,25025,
 ↪  26000,139,6660,8205,20000,42,5247,8686,33060,8086,8902,20101,1089,
 ↪  2362,7100,7474,1270,5901,8000,8883,8180,47001,5554,5555,5906,8531,
 ↪  8471,9418,13500,1024,3628,5022,7902,1883,1900,4368,5902,9091,17798
 ↪  ,3268,3632,6542,8889,8090,11000,70,442,1440,3200,5275,6514,9152,40
 ↪  2,9080,11333,81,548,3306,6262,2049,3460,27017,22222,9809,37892,23,
 ↪  1723,5601,8001,44818,8333,8444,8983,10162,222,3128,5498,23943,8123
 ↪  ,705,2380,32764,2224,9081,11099,16993,79,2021,5168,4949,9092,47002
 ↪  ,50021,137,1129,1530,2002,17777,2638,7071,8222,9390,7879,34205,655
 ↪  35,8023,20171,57772,37,3003,5000,7579,9471,7,161,5001,5938,3217,80
 ↪  87,5432,8099,8903,17200,264,3690,4366,4786,4444,5007,82,2598,3502,
 ↪  4365,8006,8445,37777,65002,513,1100,1611,5632,7676,8083,14330,3829
 ↪  2,17,554,3050,6379,8098,28222,49152,4369,7777,8880,18264,3790,9090
 ↪  ,9495,61616,13,85,888,8008,998,2379,9443,771,3311,44343,50013,743,
 ↪  3057,3817,5910,8899,12221,27080,119,5038,8182,8443,1090,5814,9160,
 ↪  1101,4353,19300,5520,6443,10001,41025,1091,10000,50000,54921,19810
 ↪  ,512,2082,8009,8028,2121,7510,8488,52311,1,631,1211,1581,62078,770
 ↪  0,88,465,717,910,9,5671,9391,50090,9042,17472,44334,523,5985,8080,
 ↪  9000,20031,27019,5986,8649,11234,16992,912,5631,8100,17783,6504,75
 ↪  80,8030,8787,1352,4950,5989,6082,9084,587,5580,8812,13778,7144,506
 ↪  0,17776,37891,2604,4445,10443,17778,9593,540,1311,5920,8834,22,607
 ↪  0,9100,9300,9595,2323,3780,4659,9594,17791,1158,1533,3872,17784,17
 ↪  781,1755,2947,2967,15672,8500,31099,1080,1604,3312,8020,9401,9440,
 ↪  10098,1099,4000,7547,8503,6050,6112,6661,6667,7002,8088,8871,9855,
 ↪  443,4343,5560,7000,3389,41080,524,1610,2533,3083,8095,20293,23791,
 ↪  28784,111,1000,1241,5521,61614,3900,17782,20111,4092,7373,7770,900
 ↪  2,1128,2381,3000,3871,19888,20222,19,109,3001,4840,10203,17790,102
 ↪  ,1103,6161,7021,5037,9001,13364,110,515,1220,1234,9380,123,15671,2
 ↪  0010,1521,45230,689,1199,5800,10008,8800,990,2375,2376,6988,5909,6
 ↪  001,10616,12401,1035,1083,1801,3500,31001,54922,7778,9527,38008,63
 ↪  6,5040,5672,7210,502,10051,47290,2105,4567,20034,30000,40007,2000,
 ↪  6556,8300,27000,2809,179,3033,2103,2207,6502,41523,43,783,873,2023
 ↪  ,9999,389,2068,3273,4443,27018,5666,5905,8530,9111,8082,280,3071,4
 ↪  322,6060,143,1433,5904,15200,34963,7181,105,3300,7001,7077,8003,88
 ↪  88,37890,50121,623,1468,1830,5911,1098,5405,13838,34443,1300,2443,
 ↪  8010,26122,10202,46823,83,84,7801,8127,902,2601,25000,80,1260,2525
 ↪  ,46824,4987,8401,38102,62514,49",
"tcp-skip-protocol": "false",
"tenable-access-key": "",
"tenable-api-url": "",
"tenable-exclude-unknown": "false",
"tenable-fingerprint-only": "false",
"tenable-include-unscanned": "false",
"tenable-risks": "None,Low,Medium,High,Critical",
```

```
  "tenable-secret-key": "",
  "tenable-severities": "Info,Low,Medium,High,Critical",
  "tenablesecuritycenter-access-key": "",
  "tenablesecuritycenter-api-url": "",
  "tenablesecuritycenter-batch-size": "2000",
  "tenablesecuritycenter-exclude-unknown": "false",
  "tenablesecuritycenter-fingerprint-only": "false",
  "tenablesecuritycenter-insecure": "",
  "tenablesecuritycenter-query-id": "",
  "tenablesecuritycenter-query-mode": "filters",
  "tenablesecuritycenter-risks": "None,Low,Medium,High,Critical",
  "tenablesecuritycenter-secret-key": "",
  "tenablesecuritycenter-severities": "Info,Low,Medium,High,Critical",
  "tenablesecuritycenter-sync-since": "1706631764",
  "tenablesecuritycenter-thumbprints": "",
  "tftp-ports": "69",
  "tos": "0",
  "ubnt-port": "10001",
  "verbose": "true",
  "very-verbose": "false",
  "vmware-insecure": "",
  "vmware-password": "",
  "vmware-thumbprints": "",
  "vmware-username": "",
  "webmin-ports": "10000",
  "wlan-list-poll-interval": "300",
  "wsd-port": "3702"
}
```


Listing 2. Default runZero scan profile

C.4.3 OT Full scan

The OT Full scan was taken from the runZero playbook for scanning OT networks and is designed to "perform comprehensive discovery and fingerprinting on the OT network while still taking a conservative approach to the volume of traffic generated by scanning" (runZero 2024b).[15] After starting with this playbook, CECA also tuned several additional parameters to increase the level of detail and aggressiveness of the OT probes in an attempt to collect as much OT information as possible. These included:

- setting `modbus-identification-level` to `extended`
- setting `dnp3-banner-address-discovery` to `prefer`
- setting `dnp3-explorer-address` to `1` to match the SCADA platform address in the environment.

Before deploying this configuration for evaluations, CECA tested these aggressive settings as recommended in the runZero playbook. Despite the advanced settings and duplicated DNP3 master address, no adverse affects were observed.


```
{
  "scan-name": "CECA-CH2-Sc1C",
  "scan-tags": "scenario=1C",
  "scan-description": "CECA Cohort 2, Scenario 1C - A full OT Scan",
  "arp-fast": "false",
  "bacnet-ports": "46808,47808,48808",
```

```
"clock-offset": "0",
"dahua-dhip-ports": "37810",
"dnp3-address-probe-timeout": "30",
"dnp3-banner-address-discovery": "prefer",
"dnp3-destination-address-discovery-range": "0-1024",
"dnp3-explorer-address": "1",
"dns-disable-google-myaddr": "false",
"dns-disable-meraki-detection": "true",
"dns-port": "53",
"dns-resolve-name": "off",
"dns-trace-domain": "off",
"dtls-ports": "443,3391,4433,5246,5349,5684",
"ethernetip-use-tagged-context": "false",
"excludes": "",
"fins-port": "9600",
"host-ping": "false",
"host-ping-probes":
 ↪  "arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns",
"ike-port": "500",
"ipmi-port": "623",
"kerberos-port": "88",
"knxnet-ports": "3671",
"l2t-port": "2228",
"l2tp-ports": "1701",
"lantronix-port": "30718",
"layer2-add-targets": "true",
"layer2-force": "false",
"layer2-max-retries": "2",
"layer2-tcp-ports": "22,80,135,179,443,3389,5040,7547,62078",
"layer2-udp-trace-port": "9",
"ldap-base-dn": "",
"ldap-exclude-unknown": "false",
"ldap-insecure": "",
"ldap-legacy-tls": "",
"ldap-password": "",
"ldap-service-options": "defaults",
"ldap-thumbprints": "",
"ldap-url": "",
"ldap-username": "",
"max-attempts": "3",
"max-group-size": "2048",
"max-host-rate": "20",
"max-sockets": "2048",
"max-tos": "0",
"max-ttl": "64",
"modbus-identification-level": "extended",
"mssql-port": "1434",
"nameservers": "",
"netbios-port": "137",
"nopcap": "false",
"ntp-port": "123",
"openvpn-ports": "1194",
"passes": "1",
```

```
"pca-port": "5632",
"ping-only": "false",
"probes": "arp,bacnet,dahua-dhip,dnp3,dns,dtls,ethernetip,fins,ike,ipm
 ↪  i,kerberos,knxnet,l2t,l2tp,lantronix,layer2,ldap,modbus,mssql,netb
 ↪  ios,ntp,openvpn,pca,s7comm,sip,snmp,ssdp,ssh,syn,connect,tftp,ubnt
 ↪  ,vmware,webmin",
"rate": "500",
"s7comm-request-extended-information": "true",
"scan-mode": "host",
"scanner-name": "main",
"screenshots": "true",
"sip-port": "5060",
"skip-broadcast": "true",
"snmp-comms": "private,public",
"snmp-disable-bulk": "false",
"snmp-max-repetitions": "16",
"snmp-max-retries": "1",
"snmp-poll-interval": "300",
"snmp-port": "161",
"snmp-timeout": "5",
"snmp-v3-auth-passphrase": "",
"snmp-v3-auth-protocol": "",
"snmp-v3-context": "",
"snmp-v3-privacy-passphrase": "",
"snmp-v3-privacy-protocol": "",
"snmp-v3-username": "",
"snmp-walk-timeout": "60",
"ssdp-port": "1900",
"ssh-fingerprint": "true",
"ssh-fingerprint-username": "_STATUS_",
"subnet-ping": "false",
"subnet-ping-net-size": "256",
"subnet-ping-probes":
 ↪  "arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns",
"subnet-ping-sample-rate": "3",
"syn-disable-bogus-filter": "false",
"syn-forwarding-check": "false",
"syn-forwarding-check-target": "13.248.161.247",
"syn-max-retries": "2",
"syn-report-resets": "true",
"syn-reset-sessions": "true",
"syn-reset-sessions-delay": "0",
"syn-reset-sessions-limit": "50",
"syn-traceroute": "true",
"syn-udp-trace-port": "9",
"targets": "10.0.0.0/8",
"tcp-excludes": "",
```

```
"tcp-ports": "9593,81,4092,9091,28222,17791,19300,20171,25000,3690,443
↪ 3,9092,34962,1220,1241,7770,84,50021,5560,5632,2181,5521,8400,921,
↪ 2376,8086,7579,9152,19810,8006,40317,65535,37,6660,22222,55580,908
↪ 4,54321,65002,1103,2967,6661,9471,9594,33060,179,2021,6542,1091,25
↪ 565,5988,54923,7,3000,5683,47001,1234,5908,7001,17777,617,3460,558
↪ 0,7778,8303,8649,20034,37718,70,4950,6443,222,8471,1583,27019,1567
↪ 2,50000,51443,7080,9099,10050,3817,8899,9530,28017,42,43,3003,2222
↪ ,2533,8161,54922,1300,5405,7700,5555,11211,1443,2121,5022,705,8834
↪ ,1098,2601,5985,1199,1440,1801,4444,30000,6502,8098,23472,1,3057,3
↪ 083,7547,9001,32764,3299,4567,5938,8488,8903,137,1260,4322,9080,15
↪ 30,6405,7676,16992,111,912,12174,4786,7801,8787,17185,37892,80,82,
↪ 3351,46823,8014,8503,4368,5520,5986,7002,9200,13,1090,4848,7800,17
↪ 783,62078,88,5900,5911,69,407,8300,13778,4000,10202,11333,1533,608
↪ 2,18881,15671,38010,79,1128,6112,9000,17784,3071,6101,6667,44334,1
↪ 830,8333,26000,7510,8090,8182,8445,10162,4987,5984,6080,12203,8180
↪ ,8205,23943,4369,5250,7902,11099,13364,52311,5814,8099,8901,3220,9
↪ 418,9495,139,873,1129,18264,55553,623,3200,8800,62514,8003,20101,6
↪ 1616,21,5040,12397,34443,38292,41523,515,8089,9524,9391,17781,3496
↪ 3,1101,1468,4343,38008,5905,6161,6556,5907,2380,3273,4949,9160,202
↪ 93,25,998,1158,25025,41025,6514,8000,8012,8100,8530,902,995,6000,5
↪ 2302,10443,12345,46824,910,17778,54921,3312,6106,8883,2100,17200,2
↪ 0031,6262,6379,8127,9060,17472,123,2207,3632,8890,6002,8081,8812,1
↪ 0080,27080,402,2323,5902,5433,9401,10001,26122,57772,389,2199,5000
↪ ,5989,13838,631,3900,5672,500,6001,19,4443,41524,8983,49,1521,2083
↪ ,5601,9111,1099,2375,4365,7181,7443,8010,9380,161,1270,5093,5901,7
↪ 580,10203,3269,5051,5247,5222,9809,783,888,1000,502,5554,7021,4366
↪ ,5007,6905,143,512,1604,27000,31001,32844,2105,8087,9443,9300,8850
↪ ,20222,49152,50121,1433,1514,8686,7070,8181,44818,689,4445,5910,38
↪ 4,10628,12221,20111,6504,8008,8082,2598,17798,34205,109,445,1211,1
↪ 900,3389,7373,7879,105,465,1030,8030,10616,25672,23,8001,2947,3260
↪ ,4659,5800,8095,554,1352,1582,20010,50013,3217,5168,8889,27017,410
↪ 80,22,1089,2082,5392,8444,16102,17776,993,3872,5351,34964,37891,43
↪ 53,5347,5906,10000,17,1883,3050,15200,37777,110,1083,11234,8545,95
↪ 95,540,5353,8172,3780,44343,717,2381,8871,903,1581,38080,113,444,6
↪ 36,10051,40007,61614,1494,8023,8123,3268,17775,17790,27888,119,135
↪ ,2000,5920,11000,28784,31099,50090,264,1755,3300,4679,5275,27018,5
↪ 87,3001,3790,8531,9999,50070,53,102,1610,8500,9081,20000,1811,2103
↪ ,2362,7000,7077,8020,8028,8888,2809,3311,3502,16993,19888,7071,800
↪ 9,9600,771,2074,3500,9440,1611,6060,6503,524,1100,1102,83,5631,605
↪ 0,4840,9042,442,1723,3128,1080,2002,8222,5909,9855,32913,2224,5400
↪ ,5671,7777,8080,548,743,990,12401,23791,47002,7100,8902,48899,513,
↪ 664,1024,280,7210,37890,7787,9002,9390,9,3306,5498,9090,38102,85,5
↪ 666,5903,3871,45230,443,2049,3033,3628,6070,8443,10008,47290,523,2
↪ 525,2604,14330,1035,2443,7474,5355,6988,2023,2379,3037,10098,2068,
↪ 4730,7144,5060,5061,8088,5001,5037,9100,5904,8083,8880,1311,5038,5
↪ 432,17782,50051,2638,9527,13500",
"tcp-skip-protocol": "false",
"tftp-ports": "69",
"tos": "0",
"ubnt-port": "10001",
"verbose": "true",
"very-verbose": "false",
"vmware-insecure": "",
```

```
  "vmware-password": "",
  "vmware-thumbprints": "",
  "vmware-username": "",
  "webmin-ports": "10000"
}
```

Listing 3. OT Full runZero scan profile, tuned to identify as much information about OT systems as possible

C.4.4 Passive Sampling

The passive sampling task simply enables passive sampling for the Explorer on `eth0` (the interface that each Explorer has on the relevant subnet in the environment) and `gre_sys` (an interface that is the termination of a Generic Routing Encapsulation (GRE) tunnel mirroring all traffic passing through the relevant interface on the local firewall). This means that the Explorer can "see" all local broadcast traffic on `eth0` and all "north-south" traffic transiting into or out of its local subnet through the firewall.

```
    {
      "explorer": "00000000-0000-0000-0000-000000000000",
      "agent": "00000000-0000-0000-0000-000000000000",
      "targets": "10.0.0.0/8 169.254.0.0/16",
      "tags": "method=passive",
      "interfaces": "eth0,gre_sys"
    }
```

Listing 4. Passive runZero sample profile

Appendix D. Evaluation Procedures

The specific steps for each evaluation are described in this section. Using the phenix ScOrch app and custom bash scripts, CECA translated the testing plan into a suite of tests that are scientific and repeatable. Each scenario can be thought of as a "program" that comprises several function calls, which are called "components" in ScOrch. These components are listed in a separate section because many of them are repeated across several scenarios.

D.1 Scenarios

D.1.1 Scenario 1: Initial Discovery

Each sub-scenario of Scenario 1 completes the same steps; the only difference is the scan profile that is injected into the virtual machine (VM) in the fourth component.

- Scenario 1A: Conservative—injects the OT limited scan profile
- Scenario 1B: Default—injects the default scan profile
- Scenario 1C: In Depth—injects the OT full scan profile.

Steps

1. `soh`: Run the phenix SoH app to ensure that all assets in the environment are powered and working as expected
2. `checkin-explorers-Sc123`: In each VM hosted in the five different subnets, execute the Explorer binary to have it register with the platform.
3. `get-rz-info`: Perform a series of API requests and command line interface (CLI) commands to record basic information about the runZero solution before starting the test.
4. `put-scan-Sc1*`: Inject the scan profile that will be used for this test (see above for mapping of profile to specific sub-scenarios).
5. `create-scans-Sc12`: Dynamically update the scan profile injected in the previous component to create five different scan profiles specific to each Explorer in each subnet of interest.
6. start `measure-disruption`: Start the ICMP polling apparatus to detect any unresponsive hosts.
7. start `tcpdump-solution-Sc123`: Start tcpdump on each runZero Explorer and platform.
8. start `tcpdump-firewalls`: Start tcpdump on each firewall.
9. `start-solution-Sc12`: Use the scan profiles created in Step 5 to start scans in each subnet.
10. `watch-solution-Sc12`: Query the runZero platform for each scan's status, and continue once all scans are processed.
11. stop `tcpdump-firewalls`: Stop the tcpdumps started in Step 8.
12. stop `tcpdump-solution-Sc123`: Stop the tcpdumps started in Step 7.
13. stop `measure-disruption`: Stop the polling apparatus started in Step 6.
14. `get-rz-results`: Perform a series of API requests to record information from the runZero platform at the conclusion of the test.
15. `recv-files`: Download artifacts created by this run for offline analysis.

D.1.2 Scenario 2: Change Discovery

Scenario 2 is similar to Scenario 1 except for steps 4 and 7, which upload a previous scan to put the solution into an "onboarded" state, and steps 8 and 19, which create and check an alert.

Steps

1. `soh`: Run the phenix SoH app to ensure that all assets in the environment are alive and working as expected.
2. `checkin-explorers-Sc123`: In each VM hosted in the five different subnets, execute the Explorer binary to have it register with the platform.
3. `get-rz-info`: Perform a series of API requests and CLI commands to record basic information about the runZero solution before starting the test.
4. `upload-previous-scan`: Upload the scan results from a previous run (Scenario 1A) to put the solution into an "onboarded" state.
5. `put-scan-Sc1C`: Inject the scan profile that will be used for this test.
6. `create-scans-Sc12`: Dynamically update the scan profile injected in the previous component to create five different scan profiles specific to each Explorer in each subnet of interest.

7. `get-rz-info-updated`: Perform more API requests to record information about the runZero solution after uploading the previous scan but before running the test.
8. `create-alert`: Manually create a rule and channel to alert on new devices found during this test.
9. start `measure-disruption`: Start the ICMP polling apparatus to detect any unresponsive hosts.
10. start `tcpdump-solution-Sc123`: Start tcpdump on each runZero Explorer and platform.
11. start `tcpdump-firewalls`: Start tcpdump on firewall.
12. `start-solution-Sc12`: Use the scan profiles created in Step 5 to start scans in each subnet.
13. `watch-solution-Sc12`: Query the runZero platform for each scan's status, and continue once all scans are processed.
14. stop `tcpdump-firewalls`: Stop the tcpdumps started in Step 10.
15. stop `tcpdump-solution-Sc123`: Stop the tcpdumps started in Step 9.
16. stop `measure-disruption`: Stop the polling apparatus started in Step 8.
17. `get-rz-results`: Perform a series of API requests to record information from the runZero platform at the conclusion of the test.
18. `recv-files`: Download artifacts created by this run for offline analysis.
19. `check-alert`: Manually check for alerts for the new devices found during this test.

D.1.3 Scenario 3: Passive Discovery

Scenario 3 follows the same structure as the previous two scenarios but has fewer steps and waits for a predetermined amount of time before exiting.

Steps

1. `soh`: Run the phenix SoH app to ensure that all assets in the environment are alive and working as expected.
2. `checkin-explorers-Sc123`: In each VM hosted in the five different subnets, execute the Explorer binary to have it register with the platform.
3. `get-rz-info`: Perform a series of API requests and CLI commands to record basic information about the runZero solution before starting the test.
4. `put-sample-Sc3`: Inject the sample profile that will be used for this test.
5. `create-samples-Sc3`: Dynamically update the sample profile injected in the previous component to create five different sample profiles specific to each Explorer in each subnet of interest.
6. start `tcpdump-solution-Sc123`: Start tcpdump on each runZero Explorer and platform.
7. start `tcpdump-firewalls`: Start tcpdump on each firewall.
8. `start-solution-Sc3`: Use the sample profiles created in Step 5 to start the passive sampling in each subnet.
9. `pause30m`: Wait for 30 minutes.
10. stop `tcpdump-firewalls`: Stop the tcpdumps started in Step 8.
11. stop `tcpdump-solution-Sc123`: Stop the tcpdumps started in Step 7.
12. `get-rz-results`: Perform a series of API requests to record information from the runZero platform at the conclusion of the test.
13. `recv-files`: Download artifacts created by this run for offline analysis.

D.1.4 Scenario 4A: Scale Discovery—Initial

Scenario 4A follows a similar template as the previous scans, but it is tailored to use just one Explorer.

1. `checkin-rz-explorer-Sc4`: Execute the Explorer binary to have it register with the platform.
2. `get-rz-info`: Perform a series of API requests and CLI commands to record basic information about the runZero solution before starting the test.
3. `put-scan-Sc1B`: Inject the scan profile that will be used for this test.
4. `create-scan-Sc4`: Dynamically update the scan profile injected in the previous component.
5. start `tcpdump-solution-Sc4`: Start tcpdump on the runZero Explorer and platform.
6. `start-solution-Sc4`: Use the scan profiles created in Step 4 to start the scan.
7. `watch-solution-Sc4`: Query the runZero platform for the scan's status, and continue once it is processed.
8. stop `tcpdump-solution-Sc4`: Stop the tcpdumps started in Step 5.

9. `get-rz-results`: Perform a series of API requests to record information from the runZero platform at the conclusion of the test.
10. `recv-files`: Download the artifacts created by this run for offline analysis.

D.1.5 Scenario 4B: Scale Discovery—Second

Scenario 4B builds on Scenario 4A and differs only in the first four components.

1. `turn-on-attacker`: Start the attacker VM.
2. `get-rz-info-updated`: Perform more API requests to record information about the runZero solution after uploading the previous scan but before running the test.
3. start `tcpdump-solution-Sc4`: Start tcpdump on the runZero Explorer and platform.
4. `start-solution-Sc4`: Use the scan profiles created in Step 4 to start the scan.
5. `watch-solution-Sc4`: Query the runZero platform for the scan's status, and continue once it is processed.
6. stop `tcpdump-solution-Sc4`: Stop the tcpdump started in Step 5.
7. `get-rz-results`: Perform a series of API requests to record information from the runZero platform at the conclusion of the test.
8. `recv-files`: Download the artifacts created by this run for offline analysis.

D.2 Components

D.2.1 soh

Run the phenix SoH app to ensure that all assets in the environment are alive and working as expected.

D.2.2 checkin-explorers-Sc123

In each VM hosted in the five different subnets, execute the Explorer binary to have it register with the platform.

Code

(In each Explorer VM)

```
/root/runzero-explorer.bin
```

D.2.3 get-rz-info

At the very beginning of all tests, the following API calls were made to verify the runZero platform status before a test was started.

- GET `/org/sites` to get the unique site identifier
- GET. `/org/explorers` to verify that all five Explorers were checked into the platform and get their unique IDs.
- `GET /org` to record basic information about the organization.
- `GET /org/assets` and `GET /org/services` to verify that no assets exist in the database.
- `GET /org/tasks` to verify that no tasks have been started.

Code

(On the helper VM that interacts with the runZero platform)

```
# oid that is specific to this organization, should be the only id
 ↪  that is manually changed
echo "00000000-0000-0000-0000-000000000000" |
tee /root/oid.txt
oid=$(cat /root/oid.txt)
# site_id that defines the Primary site
curl -s -k -X GET -H @/root/get-api-header.txt \
```

```
  https://10.1.2.10/api/v1.0/org/sites?_oid=$oid |
jq -r '.[] | select(.name="Primary") | .id' |
tee /root/site_id.txt
# explorer_id for each explorer after they check in
sleep 5
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/explorers?_oid=$oid |
jq |
tee /root/runzero-explorers.json
# Start collecting information about the RZ platform
echo "GET /org" > /root/runzero-pre-run-info.txt
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org?_oid=$oid |
jq |
tee -a /root/runzero-pre-run-info.txt
echo "GET /org/assets" >> /root/runzero-pre-run-info.txt
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/assets?_oid=$oid |
jq |
tee -a /root/runzero-pre-run-info.txt
echo -e "\nGET /org/tasks" >> /root/runzero-pre-run-info.txt
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/tasks?_oid=$oid |
jq |
tee -a /root/runzero-pre-run-info.txt
echo -e "\nGET /org/services" >> /root/runzero-pre-run-info.txt
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/services?_oid=$oid |
jq |
tee -a /root/runzero-pre-run-info.txt
```

(On the runZero platform)

```
runzeroctl --version > /root/runzero-platform-version.txt
```

(On the runZero Explorer in the control center)

```
/root/runzero-explorer.bin --version >
 ↪  /root/runzero-explorer-version.txt 2>&1
```

D.2.4 put-scan-Sc1*

This component uses the minimega inject command to upload a scan profile to the helper VM's `/root/task-profiles/` directory.

There are three different versions of this component (`put-scan-Sc1A`, `put-scan-Sc1B`, and `put-scan-Sc1B`). The only difference is the scan profile that is injected into the VM in the fourth component.

- Scenario 1A: Conservative—injects the OT Limited scan profile.
- Scenario 1B: Default—injects the Default scan profile.
- Scenario 1C: Deep—injects the OT Full scan profile.

D.2.5 create-scans-Sc12

Dynamically update the scan profile injected in the previous component to create five different scan profiles specific to each Explorer in each subnet of interest.

Code

(On the helper VM that interacts with the runZero platform)

```
oid=$(cat /root/oid.txt)
cc_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "CC-RUNZERO-ADMIN-VM") | .id')
sub_ot_gateway_explorer_id=$(cat /root/runzero-explorers.json | jq -r
 ↪  '.[] | select(.name == "SUB-OT-GATEWAY-RUNZERO-ADMIN-VM") | .id')
sub_ot_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "SUB-OT-RUNZERO-ADMIN-VM") | .id')
sub_it_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "SUB-IT-RUNZERO-ADMIN-VM") | .id')
pv_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "PV-RUNZERO-ADMIN-VM") | .id')
epoch=$(date +%s)
cat /root/task-profiles/scan-profile.json |
jq --arg jq_epoch $epoch --arg jq_explorer $cc_explorer_id --arg
 ↪  jq_add_name "-CC" --arg jq_add_tag ", location=CC" --arg
 ↪  jq_target "10.1.1.0/24" '."scan-start" = $jq_epoch | .explorer =
 ↪  $jq_explorer | .agent = $jq_explorer | ."scan-name" |= . +
 ↪  $jq_add_name | ."scan-tags" |= . + $jq_add_tag | .targets =
 ↪  $jq_target' |
tee /root/task-profiles/scan-cc.json
cat /root/task-profiles/scan-profile.json |
jq --arg jq_epoch $epoch --arg jq_explorer
 ↪  $sub_ot_gateway_explorer_id --arg jq_add_name "-SUB-OT-GATEWAY"
 ↪  --arg jq_add_tag ", location=SUB-OT-GATEWAY" --arg jq_target
 ↪  "10.2.4.0/24" '."scan-start" = $jq_epoch | .explorer =
 ↪  $jq_explorer | .agent = $jq_explorer | ."scan-name" |= . +
 ↪  $jq_add_name | ."scan-tags" |= . + $jq_add_tag | .targets =
 ↪  $jq_target' |
tee /root/task-profiles/scan-sub-ot-gateway.json
cat /root/task-profiles/scan-profile.json |
jq --arg jq_epoch $epoch --arg jq_explorer $sub_ot_explorer_id --arg
 ↪  jq_add_name "-SUB-OT" --arg jq_add_tag ", location=SUB-OT" --arg
 ↪  jq_target "10.2.2.0/24" '."scan-start" = $jq_epoch | .explorer =
 ↪  $jq_explorer | .agent = $jq_explorer | ."scan-name" |= . +
 ↪  $jq_add_name | ."scan-tags" |= . + $jq_add_tag | .targets =
 ↪  $jq_target' |
tee /root/task-profiles/scan-sub-ot.json
cat /root/task-profiles/scan-profile.json |
jq --arg jq_epoch $epoch --arg jq_explorer $sub_it_explorer_id --arg
 ↪  jq_add_name "-SUB-IT" --arg jq_add_tag ", location=SUB-IT" --arg
 ↪  jq_target "10.2.3.0/24" '."scan-start" = $jq_epoch | .explorer =
 ↪  $jq_explorer | .agent = $jq_explorer | ."scan-name" |= . +
 ↪  $jq_add_name | ."scan-tags" |= . + $jq_add_tag | .targets =
 ↪  $jq_target' |
tee /root/task-profiles/scan-sub-it.json
cat /root/task-profiles/scan-profile.json |
```

```
jq --arg jq_epoch $epoch --arg jq_explorer $pv_explorer_id --arg
 ↪  jq_add_name "-PV" --arg jq_add_tag ", location=PV" --arg
 ↪  jq_target "XX.XX.XX.0/24" '."scan-start" = $jq_epoch | .explorer
 ↪  = $jq_explorer | .agent = $jq_explorer | ."scan-name" |= . +
 ↪  $jq_add_name | ."scan-tags" |= . + $jq_add_tag | .targets =
 ↪  $jq_target' |
tee /root/task-profiles/scan-pv.json
```

D.2.6 measure-disruption

Starts and stops the ICMP polling apparatus to detect any unresponsive hosts.

Start

Code

(In each monitoring VM)

```
tmux new-session -s md -d \
  "fping -l -p 1000 -t 200 -r 3 -D -e -f /root/cc-monitor-ips.txt >
   ↪  /root/cc-disruption-results.txt 2>&1" &&
  true
```

Stop

Code

(in each monitoring VM)

```
tmux send-keys -t md "C-c"
```

(In the SCADA platform)

```
journalctl -u ot-sim -g ERROR > /root/ot-sim-errors.log
```

After which the file `/root/cc-disruption-results.txt` is extracted from each monitoring VM, and `/root/ot-sim-errors.log` is extracted from the SCADA platform, for analysis.

D.2.7 tcpdump-solution-Sc123

Start and stop tcpdump on each runZero Explorer and platform.

D.2.8 tcpdump-firewalls

Start and stop tcpdump on each firewall interface in the environment.

D.2.9 start-solution-Sc12

The same API call was made five times in a row to start scans on each of the five Explorers in each subnet:

- `PUT /org/sites/{site_id}/scan` with a formatted `scan-{subnet}.json` scan profile from below.

Code

(On the helper VM that interacts with the runZero platform)

```
# variables
oid=$(cat /root/oid.txt)
site_id=$(cat /root/site_id.txt)
# record start
echo "Start:" > /root/timing.txt
date +%s.%N >> /root/timing.txt
echo "Stop:" >> /root/timing.txt
# start scans
# CC
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/scan-cc.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/scan?_oid=$oid |
jq |
tee /root/scan-cc-response.json
# SUB-OT-GATEWAY
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/scan-sub-ot-gateway.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/scan?_oid=$oid |
jq |
tee /root/scan-sub-ot-gateway-response.json
# SUB-OT
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/scan-sub-ot.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/scan?_oid=$oid |
jq |
tee /root/scan-sub-ot-response.json
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/scan-sub-it.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/scan?_oid=$oid |
jq |
tee /root/scan-sub-it-response.json
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/scan-pv.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/scan?_oid=$oid |
jq |
tee /root/scan-pv-response.json
```

D.2.10 watch-solution-Sc12

While the scans were running, the API call was made to check on the status of all running scans using:

- `GET /org/tasks` to get all of the tasks, including the scans started above.

This call was repeated every second until the return showed that all of the scans started in the previous component had a `status` of `processed`. This check allowed CECA to measure the time between starting a test's scans and their completion with sub-two-second precision.

Code

(In each Explorer VM)

```
# variables
oid=$(cat /root/oid.txt)
cc_task_id=$(cat /root/scan-cc-response.json | jq -r '.id')
```

```
sub_ot_gateway_task_id=$(cat /root/scan-sub-ot-gateway-response.json
 ↪  | jq -r '.id')
sub_ot_task_id=$(cat /root/scan-sub-ot-response.json | jq -r '.id')
sub_it_task_id=$(cat /root/scan-sub-it-response.json | jq -r '.id')
pv_task_id=$(cat /root/scan-pv-response.json | jq -r '.id')
# check the tasks api to see each of the statuses are processed
while [[  $(curl -s -k -X GET -H @/root/get-api-header.txt \
              https://10.1.2.10/api/v1.0/org/tasks?_oid=$oid |
            jq --arg cc_task_id $cc_task_id --arg
             ↪  sub_ot_gateway_task_id $sub_ot_gateway_task_id --arg
             ↪  sub_ot_task_id $sub_ot_task_id --arg sub_it_task_id
             ↪  $sub_it_task_id --arg pv_task_id $pv_task_id '[.[] |
             ↪  select((.id == $cc_task_id or .id ==
             ↪  $sub_ot_gateway_task_id or .id == $sub_ot_task_id or
             ↪  .id == $sub_it_task_id or .id == $pv_task_id) and
             ↪  .type == "scan") | .status == "processed"] | all') ==
             ↪  false ]]
do
  sleep 1
done
# record the time when done
date +%s.%N >> /root/timing.txt
```

D.2.11 get-rz-results

After each test, the following API calls were used to extract information from the platform:

- `GET /export/org/assets.csv`, `GET /org/assets`, and `GET /org/services` to get information about the identified assets in two different formats and services
- `GET /org/tasks` to record the tasks that were run during the test.

Code

(In each Explorer VM)

```
oid=$(cat /root/oid.txt)
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/export/org/assets.csv?_oid=$oid |
tee /root/inventory.csv
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/assets?_oid=$oid |
jq |
tee /root/runzero-post-run-assets.json
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/tasks?_oid=$oid |
jq |
tee /root/runzero-post-run-tasks.json
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/services?_oid=$oid |
jq |
tee /root/runzero-post-run-services.json
```

(On the runZero platform)

```
tar czfv /root/runzero-raw-storage.tar.gz /opt/runzero/storage/
```

D.2.12 recv-files

Extract the following files from the following hosts for offline analysis:

- Helper VM
  - `/root/timing.txt`
  - `/root/runzero-explorers.json`
  - `/root/runzero-pre-run-info.txt`
  - `/root/inventory.csv`
  - `/root/runzero-post-run-assets.json`
  - `/root/runzero-post-run-tasks.json`
  - `/root/runzero-post-run-services.json`.
- runZero platform:
  - `/root/runzero-platform-version.txt`
  - `/root/runzero-raw-storage.tar.gz`.
- runZero Explorer VM in the control center: `/root/runzero-explorer-version.txt`

D.2.13 upload-previous-scan

In Scenario 2: Change Detection, the API was used to upload the previous scan results for each subnet to put the platform into an "onboarded state" as if it had already identified the assets in the environment.
First, the previous scan results are injected into the helper VM at `/root/previous-scans/previous-scan-{subnet}.json.gz`.
Then, the API calls are made five times to upload these to the platform:

- `PUT /org/sites/{site_id}/import` with an appropriate `previous-scan-{subnet}.json.gz`.

Code

(In each Explorer VM)

```
# variables
oid=$(cat /root/oid.txt)
site_id=$(cat /root/site_id.txt)
# uploads
# CC
curl -s -k -X PUT -H @/root/put-import-api-header.txt \
  --data-binary @/root/previous-scans/previous-scan-cc.json.gz \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/import?_oid=$oid |
jq |
tee /root/upload-previous-scan-cc-response.json
# SUB-OT-GATEWAY
curl -s -k -X PUT -H @/root/put-import-api-header.txt \
  --data-binary
   ↪  @/root/previous-scans/previous-scan-sub-ot-gateway.json.gz \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/import?_oid=$oid |
jq |
tee /root/upload-previous-scan-sub-ot-gateway-response.json
# SUB-OT
curl -s -k -X PUT -H @/root/put-import-api-header.txt \
  --data-binary @/root/previous-scans/previous-scan-sub-ot.json.gz \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/import?_oid=$oid |
jq |
```

```
tee /root/upload-previous-scan-sub-ot-response.json
# CC
curl -s -k -X PUT -H @/root/put-import-api-header.txt \
  --data-binary @/root/previous-scans/previous-scan-sub-it.json.gz \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/import?_oid=$oid |
jq |
tee /root/upload-previous-scan-sub-it-response.json
# CC
curl -s -k -X PUT -H @/root/put-import-api-header.txt \
  --data-binary @/root/previous-scans/previous-scan-pv.json.gz \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/import?_oid=$oid |
jq |
tee /root/upload-previous-scan-pv-response.json
```

D.2.14 create-alert

A "break" component that pauses execution until manually exited. During this time, CECA created an internal alert using the runZero playbook for creating alerts for any new device being identified.

D.2.15 get-rz-info-updated

Last, before Scenario 2: Change Detection and the second scan of Scenario 4: Scale Detection began, some additional checks were remade to capture the platforms' "onboarded" state:

- `GET /org/assets` and `GET /org/services` to verify that all the assets from the previous run exist
- `GET /org/tasks` to get the previous tasks that were run.

Code

(In each Explorer VM)

```
# setup variables
oid=$(cat /root/oid.txt)
echo -e "\n\n========== AFTER UPDATING THE SYSTEM ========== \n\n" >>
 ↪  /root/runzero-pre-run-info.txt
echo "GET /org/assets" >> /root/runzero-pre-run-info.txt
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/assets?_oid=$oid |
jq |
tee -a /root/runzero-pre-run-info.txt
echo -e "\nGET /org/tasks" >> /root/runzero-pre-run-info.txt
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/tasks?_oid=$oid |
jq |
tee -a /root/runzero-pre-run-info.txt
echo -e "\nGET /org/services" >> /root/runzero-pre-run-info.txt
curl -s -k -X GET -H @/root/get-api-header.txt \
  https://10.1.2.10/api/v1.0/org/services?_oid=$oid |
jq |
tee -a /root/runzero-pre-run-info.txt
```

D.2.16 check-alert

A "break" component that pauses execution until it is manually exited. During this time, CECA checked the internal alerts on the runZero platform and recorded a screenshot.

D.2.17 put-sample-Sc3

This component uses the minimega inject command to upload a template sample profile to the helper VM's `/root/task-profiles/` directory.

D.2.18 create-samples-Sc3

Dynamically update the sample profile injected in the previous component to create five different sample profiles specific to each Explorer in each subnet of interest.

Code

(On the helper VM that interacts with the runZero platform)

```
oid=$(cat /root/oid.txt)
cc_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "CC-RUNZERO-ADMIN-VM") | .id')
sub_ot_gateway_explorer_id=$(cat /root/runzero-explorers.json | jq
 ↪  -r '.[] | select(.name == "SUB-OT-GATEWAY-RUNZERO-ADMIN-VM") |
 ↪  .id')
sub_ot_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "SUB-OT-RUNZERO-ADMIN-VM") | .id')
sub_it_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "SUB-IT-RUNZERO-ADMIN-VM") | .id')
pv_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "PV-RUNZERO-ADMIN-VM") | .id')
# epoch time to get the current time
epoch=$(date +%s)
# for each subnet / explorer, create a sample profile
# CC
cat /root/task-profiles/sample-profile.json |
jq --arg jq_explorer $cc_explorer_id --arg jq_add_tag ",
 ↪  location=CC" '.explorer = $jq_explorer | .agent = $jq_explorer |
 ↪  .tags |= . + $jq_add_tag' |
tee /root/task-profiles/sample-cc.json
# SUB-OT-GATEWAY
cat /root/task-profiles/sample-profile.json |
jq --arg jq_explorer $sub_ot_gateway_explorer_id --arg jq_add_tag ",
 ↪  location=SUB-OT-GATEWAY" '.explorer = $jq_explorer | .agent =
 ↪  $jq_explorer | .tags |= . + $jq_add_tag' |
tee /root/task-profiles/sample-sub-ot-gateway.json
# SUB-OT
cat /root/task-profiles/sample-profile.json |
jq --arg jq_explorer $sub_ot_explorer_id --arg jq_add_tag ",
 ↪  location=SUB-OT" '.explorer = $jq_explorer | .agent =
 ↪  $jq_explorer | .tags |= . + $jq_add_tag' |
tee /root/task-profiles/sample-sub-ot.json
# SUB-IT
cat /root/task-profiles/sample-profile.json |
jq --arg jq_explorer $sub_it_explorer_id --arg jq_add_tag ",
 ↪  location=SUB-IT" '.explorer = $jq_explorer | .agent =
 ↪  $jq_explorer | .tags |= . + $jq_add_tag' |
tee /root/task-profiles/sample-sub-it.json
# PV
cat /root/task-profiles/sample-profile.json |
```

```
jq --arg jq_explorer $pv_explorer_id --arg jq_add_tag ",
 ↪  location=PV" '.explorer = $jq_explorer | .agent = $jq_explorer |
 ↪  .tags |= . + $jq_add_tag' |
tee /root/task-profiles/sample-pv.json
```

D.2.19 start-solution-Sc3

Use the sample profiles created in Step 5 to start samples in each subnet.

Code

(On the helper VM that interacts with the runZero platform)

```
# variables
oid=$(cat /root/oid.txt)
site_id=$(cat /root/site_id.txt)
# record start
echo "Start:" > /root/timing.txt
date +%s.%N >> /root/timing.txt
echo "Stop: 30 minutes later" >> /root/timing.txt
# start samples
# CC
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/sample-cc.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/sample?_oid=$oid |
jq |
tee /root/sample-cc-response.json
# SUB-OT-GATEWAY
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/sample-sub-ot-gateway.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/sample?_oid=$oid |
jq |
tee /root/sample-sub-ot-gateway-response.json
# SUB-OT
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/sample-sub-ot.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/sample?_oid=$oid |
jq |
tee /root/sample-sub-ot-response.json
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/sample-sub-it.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/sample?_oid=$oid |
jq |
tee /root/sample-sub-it-response.json
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/sample-pv.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/sample?_oid=$oid |
jq |
tee /root/sample-pv-response.json
```

D.2.20 pause30m

Wait for 30 minutes.

D.2.21 checkin-rz-explorer-Sc4

Execute the Explorer binary to have it register with the platform.

Code

(In the Explorer VM)

```
/root/runzero-explorer.bin
```

D.2.22 create-scan-Sc4

Dynamically update the scan profile injected in the previous component.

Code

(In the Explorer VM)

```
oid=$(cat /root/oid.txt)
cc_explorer_id=$(cat /root/runzero-explorers.json | jq -r '.[] |
 ↪  select(.name == "CC-RUNZERO-ADMIN-VM") | .id')
# epoch time to get the current time
epoch=$(date +%s)
# CC
cat /root/task-profiles/scan-profile.json |
jq --arg jq_epoch $epoch --arg jq_explorer $cc_explorer_id --arg
 ↪  jq_add_name "-CC" --arg jq_add_tag ", location=CC" --arg
 ↪  jq_target "10.200.15.0/20" '."scan-start" = $jq_epoch | .explorer
 ↪  = $jq_explorer | .agent = $jq_explorer | ."scan-name" |= . +
 ↪  $jq_add_name | ."scan-tags" |= . + $jq_add_tag | .targets =
 ↪  $jq_target' |
tee /root/task-profiles/scan-cc.json
```

D.2.23 tcpdump-solution-Sc4

Start and stop tcpdump on the runZero Explorer and platform.

D.2.24 start-solution-Sc4

Use the scan profiles created in Step 4 to start a scan.

Code

(In the Explorer VM)

```
# variables
oid=$(cat /root/oid.txt)
site_id=$(cat /root/site_id.txt)
# record start
echo "Start:" > /root/timing.txt
date +%s.%N >> /root/timing.txt
echo "Stop:" >> /root/timing.txt
# start scans
curl -s -k -X PUT -H @/root/put-api-header.txt \
  -d @/root/task-profiles/scan-cc.json \
  https://10.1.2.10/api/v1.0/org/sites/$site_id/scan?_oid=$oid |
```

```
    jq |
    tee /root/scan-cc-response.json
```

D.2.25 watch-solution-Sc4

Query the runZero platform for the scan's status, and continue once it is processed.

Code

(In the Explorer VM)

```
    # variables
    oid=$(cat /root/oid.txt)
    task_id=$(cat /root/scan-cc-response.json | jq -r '.id')
    # check the tasks api to see each of the statuses are processed
    while [[  $(curl -s -k -X GET -H @/root/get-api-header.txt \
                  https://10.1.2.10/api/v1.0/org/tasks?_oid=$oid |
                jq --arg task_id $task_id '[.[] | select((.id ==
                 ↪  $task_id) and .type == "scan") | .status ==
                 ↪  "processed"] | all') == false ]]
    do
      sleep 1
    done
    # record the time when done
    date +%s.%N >> /root/timing.txt
```

D.2.26 turn-on-attacker

Use minimega to start the attacker VM, which adds it to the environment.

CESER
PUBLIC REPORTS

U.S. DEPARTMENT OF ENERGY
Office of Cybersecurity, Energy Security, and Emergency Response