A Cyber-Physical Experimentation Platform for Resilience Analysis

Jamie Thorpe, Raymond Fasano, Meghan Galiardi Sahakian, Amanda Gonzales, Andrew Hahn,
Joshua Morris, Timothy Ortiz, Hannah Reinbolt, Eric D. Vugrin
{jthorpe,refasan,mgaliar,agonza6,ashahn,jmorri2,tortiz,hreinbo,edvugri}@sandia.gov
Sandia National Laboratories
Albuquerque, New Mexico, USA

ABSTRACT

Recent high profile cyber attacks on critical infrastructures have raised awareness about the severe and widespread impacts that these attacks can have on everyday life. This awareness has spurred research into making industrial control systems and other cyber-physical systems more resilient. A plethora of cyber resilience metrics and frameworks have been proposed for cyber resilience assessments, but these approaches typically assume that data required to populate the metrics is readily available, an assumption that is frequently not valid. This paper describes a new cyber experimentation platform that can be used to generate relevant data and to calculate resilience metrics that quantify how resilient specified industrial control systems are to specified threats. Demonstration of the platform and analysis process are illustrated through a use case involving the control system for a pressurized water reactor.

CCS CONCEPTS

• **Security and privacy → Systems security**.

KEYWORDS

resilience analysis, cyber-physical, cyber experimentation

ACM Reference Format:
Jamie Thorpe, Raymond Fasano, Meghan Galiardi Sahakian, Amanda Gonzales, Andrew Hahn, Joshua Morris, Timothy Ortiz, Hannah Reinbolt, Eric D. Vugrin. 2022. A Cyber-Physical Experimentation Platform for Resilience Analysis. In *SaT-CPS 2022: ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, April 26, 2022, Woodstock, Baltimore-Washington DC Area.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The shutdown of the Colonial Pipeline following a cyber attack in 2021 [23] attracted a lot of attention as a recent example of the consequences that cyber attacks could have on critical infrastructure systems and industrial control systems (ICSs). Though this attack received much publicity, several other prominent attacks on ICSs have occurred over the past decade and similarly demonstrated the risks of cyber attacks against ICSs. The disabling of centrifuges at a uranium enrichment facility in Iran by the Stuxnet malware attack in 2010 [25] first demonstrated that ICSs could be disrupted from a cyber attack. Since then, the disruption of critical infrastructure and ICSs has been repeatedly demonstrated by attacks on the power grid in Ukraine [4], chemical manufacturing facilities in the Middle East [5], and elsewhere.

ICS operators are realizing that successfully thwarting all attacks at all facilities at all times is an impossible task, so cyber resilience efforts are being undertaken to complement cyber security practices. Whereas cyber security activities often focus on preventing access by cyber threats and ensuring confidentiality, integrity, and availability of control systems, cyber resilience efforts and technologies complement security features by limiting damage, enabling continued operations, and facilitating a rapid recovery from the attack in the event control systems are compromised.

Though resilience has been increasingly prioritized for physical infrastructure systems over the past 20 years, research into resilience of cyber and cyber-physical systems only started to emerge over the past decade since Goldman's initial paper in 2010 [17]. One growing area of cyber resilience research is the development of cyber resilience metrics. Snyder et al. [26] describe a high-level framework of cyber resilience measures of effectiveness for "informing acquisition decisions during all stages of weapon systems' life cycles". The metrics included in this framework are heavily informed by qualitative subject matter assessments and not intended to be predictive of how cyber-physical systems of interest will perform in the presence of threats. Linkov et al. [21] developed a matrix of cyber resilience metrics that can be useful when performing cyber resilience assessments and when considering features of the system of interest that may need to be improved upon, but the assessment is not intended to predict impacts of threats on the operation of the system. Perhaps the most widely known framework for assessing and developing resilience in cyber systems is the National Institute of Standards and Technology's Special Publication 800-160 (Volume 2) [24]. This standard is based upon the Cyber Resilience Engineering Framework (CREF) that developed a hierarchy of cyber resilience goals, objectives, techniques, and design principles [7]. Bodeau et al. have developed companion documents that describe the Situated Scoring Methodology for Cyber Resiliency (SSM-CR) [10] and a library of possible cyber resilience metrics for use [9]. The SSM-CR blends consequence-based metrics with subjective weights that reflect priority levels for achieving certain resilience objectives. Haque et al. argue that the NIST standard and CREF are primarily geared to information technology (IT) systems, and, thus, Haque et al. [19] propose qualitative and

SaT-CPS 2022, April 26, 2022, Baltimore-Washington DC Area
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

quantitative metrics specifically for cyber-physical and industrial control systems.

The plethora of proposed metrics and frameworks provide many options for quantifying resilience of cyber-physical systems, but the proposed frameworks suffer from a common challenge. They generally either assume that the data to populate metrics is readily available (which is often not the case) or, as Bodeau et al. suggest [8], that "a modeling or emulation environment" should be used. The suggestion to use modeling or emulation is rarely accompanied with specific details on what platform to use or how to implement one. Haque et al. suggest structures for qualitative and quantitative "simulation platforms", but details are sparse, particularly around the necessary technologies to implement and develop the platforms.

This paper describes a new cyber experimentation platform that was developed to conduct resilience assessments for cyber-physical and industrial control systems. The ADvancing Resilience Of Control systems (ADROC) platform specifically addresses the challenge of determining "How does one conduct cyber experimentation and gather data necessary for quantitative cyber resilience analyses?". The ADROC platform leverages virtual cyber testbeds, simulation of physical processes, threat emulation, and cyber resilience metric libraries to create a cyber resilience assessment process that is controlled, repeatable, automated, and configurable. In doing so, the ADROC platform provides a much needed contribution to the suite of resilience assessment capabilities for cyber-physical systems.

The remainder of the paper is structured as follows. Section 2 describes the platform structure and technologies implemented within the ADROC platform. Section 3 introduces a hypothetical resilience study for a pressurized water reactor and describes how the ADROC platform can be leveraged for the study. Section 4 concludes the paper and describes future research opportunities.

2 PLATFORM DESCRIPTION

Under the ADROC platform, cyber experimentation for resilience analysis includes four primary components:

- Representation of the ICS of interest
- Representation of the threat(s) of interest
- Metrics
- Experimental controls

A cyber experiment consists of the application of the threat representation to the ICS representation, extraction of relevant data and artifacts from the application, and processing of these data and artifacts to produce metrics to quantify the magnitude and duration of the threat effects on ICS operations. Furthermore, the experimental controls ensure that the experiments are performed in a repeatable process and are also customizable to different systems and threats. The following subsections describe how these components are implemented in the ADROC platform.

2.1 System Representation

ICSs generally include supervisory control and data acquisition (SCADA) devices; field devices; networks and protocols; and physical processes such as power generation, water treatment, chemical manufacturing, etc. Though generally not considered a part of ICSs, more traditional corporate networks or information technology systems may be connected to the ICSs, along with appropriate firewalls, public-facing demilitarized zones (DMZs), etc. This observation is relevant when analyzing cyber threats to ICSs because the corporate network may provide the initial pathway for an attacker to access the ICS networks.

Three primary approaches have historically been used to represent ICSs in cyber security and resilience assessments. Mathematical abstractions and simulations can provide a flexible, safe environment for exploring cyber threats to ICSs (e.g., [6]). Mathematical simulations use numerical equations and algorithms to approximate the operations of the control system, the progress of an attack through the network, and the effect of the attack on ICS operations. The significant drawback of these approaches is assessing whether the simulation is a valid representation of the ICS; they do not use actual hardware, software, communication protocols, etc., so more often than not, they generally represent the *effect* of an attack and not the actual attack itself. Whether the attack is actually feasible may not be investigatable with the simulation.

An alternative approach is the use of testbeds that consist of the actual hardware, software, physical equipment, etc. that make up the real system of interest. The benefit of this approach is that the testbed includes the actual components included in the ICS under study and, thus, can be considered a high-fidelity representation. The most significant drawbacks include the time and cost of developing such a testbed; furthermore, components could be irreparably damaged when conducting experiments with cyber threats, potentially increasing the time and cost to conduct assessments.

An increasingly used third approach is virtual testbeds or emulations (e.g., [27]). Emulations use real software and communications protocols, but they often replace actual hosts and devices with virtual machines. The advantage of emulations is that they provide the flexibility and reduced cost and setup time of simulations while also providing a high-fidelity representation of the actual system.

The ADROC platform uses the SCEPTRE emulation capability to represent ICSs [20]. SCEPTRE integrates virtual control system devices such as programmable logic controllers (PLCs) [14] and remote terminal units (RTUs); software defined networking; virtualized supervisory control and data acquisition (SCADA) applications; and physical process simulations. Specification of an ICS in the SCEPTRE environment requires a few key steps. A topology that specifies the hosts included in the ICS and the networking between them is required. Software defined networking is used to connect the virtual machines according to the specified topology. As needed, real hardware (or hardware-in-the-loop) can be used instead of virtual devices. Finally, a unique aspect of SCEPTRE is its ability to integrate the virtual devices with simulations of physical processes included in the ICS. In doing so, SCEPTRE is able to represent both the cyber and physical components and processes one finds in ICSs and facilitate the realization of cyber effects on the system's physical processes, an interdependency that most other ICS emulation platforms do not provide. Consequently, SCEPTRE can provide large-scale, high fidelity control system test environments that can be developed, stood up, and torn down quickly, making it an ideal environment for representing ICSs in the ADROC platform.

2.2 Threat Representation

Testbeds (virtual and hardware-based) enable multiple options for threat representation. Threats can be represented with automated scripts, with human "red teams" directly interacting with the testbed, and even with actual malware. Breach and attack simulations, also called threat emulators, are an emerging technology that are being leveraged to analyze security risks to computer networks and ICSs. Threat emulators are automated programs that mimic actual cyber attacks through the use of known cyber attack tactics, techniques, and tools. Threat emulators can provide greater flexibility and configurability as compared to real malware samples. Additionally, they provide greater consistency and reproducibility for cyber experimentation when compared against human-based red teams. A key limitation of most threat emulators is that they have been developed for IT systems and may not be useful (or usable) in ICS environments.

Many tools exist for threat emulation, and the ADROC platform has integrated MITRE's CALDERA tool [12]. Specification of an attack begins with the definition of an adversary profile. The adversary profile definition includes the tactics and techniques that the attacker has available during the attack and the order in which those techniques are executed. CALDERA uses the MITRE ATT&CK framework [11] to organize tactics (the major stages in an attack) and techniques (the specific means by which the tactics are executed). For example, network service scanning, password policy discovery, and domain trust discovery are techniques for the Discovery tactic. Additionally, CALDERA requires definition of the attack operation. Operation definition requires specifying an adversary profile to use and an initial agent (infected machine) that the CALDERA server can reach. This agent represents the starting point for the attack. Preconditions (requirements for using a particular technique), a priori information, and goals (attack objectives) may also be specified. With the operation specified, CALDERA can be run. CALDERA has a suite of built-in tools that are used to perform the techniques. Thus, when executed, CALDERA will use these tools, the adversary profile, and internal decision logic to automatically step through the attack until the attacker goals are achieved.

One particular contribution of this paper is the extension of CALDERA to emulate an attack on an operational technology (OT) system. CALDERA is generally designed for IT environments, but the tool can be extended with plug-ins and payloads. For the ADROC platform, the authors supplemented the CALDERA tool with payloads specifically designed to target field devices found in ICSs. This extension of CALDERA enabled the emulation of threats that target and affect the physical processes in ICSs. The extension of CALDERA was performed using the ManiPIO tool [18]. ManiPIO is a custom Python-based tool that was developed to emulate attacks on PLCs. CALDERA treats ManiPIO as a payload, and when ManiPIO is deployed, ManiPIO uses the TCP Modbus protocol to overwrite the inputs and outputs of PLCs and other elements of the PLC's control logic. In doing so, ManiPIO can be used to emulate a variety of potential attacks on PLCs.

2.3 Run Control and Metrics

Within the ADROC platform, experiment control and calculation of resilience metrics are performed by the REsilience VeRification UNit (RevRun). RevRun is a collection of Python scripts that reside external to and interface with the SCEPTRE environment and CALDERA to coordinate the configuration and running of experiments; collection of data; processing of data; calculation of resilience metrics; and analysis of results. RevRun leverages the Elastic Stack [1] (namely Elasticsearch [2] and Kibana [3]) for data collection, storage, and visualization. Metric calculation is performed via a library of Python-based functions.

Setting up RevRun begins with a set of user-defined configuration files. These configuration files specify which SCEPTRE emulation and CALDERA attack to implement, what data to collect from the experiment, how to process that data, and what resilience metrics to calculate. These configuration files are the main components where users interact with RevRun and provide opportunities for analysis customization.

A wide variety of data can be collected from the experiments, including statistics on traffic between devices; up/down status of devices; statistics and logs from cyber security tools; and data about the physical process operations. These data can be processed to calculate metrics about the resilience of the system.

RevRun uses a hierarchical structure to report resilience metrics. The hierarchy consists of four levels: system, group, subgroup, and device. At the base level, RevRun reports device scores based on the data collected from those devices.

When the raw data is extracted from the experiments, RevRun processes the raw data to create normalized time series data. RevRun contains a library of resilience metric functions which are applied to the processed time series. Example metrics include:

- Cumulative difference: this metric calculates cumulative difference between the time series from the experiment and a time series that represents nominal (no attack) conditions
- Time to violate a threshold: this metric calculates the length of time required for the time series to violate a specified threshold
- Count: this metric calculates the total sum for the time series

These and other metrics calculated on the time series are reported at the device level, and each metric function in the library may produce results with different ranges, magnitudes, and interpretations of resilience. Therefore, RevRun considers results from multiple experiments to normalize scores so that they range between 0 and 1. A score of 0 implies the system is maximally degraded, relative to the other experiments; a score of 1 indicates the ICS experienced no degradation of function based on the selected data. When different experiments include different threat scenarios, the scores provide a means for ranking which threats are most impactful.

Subgroups are defined to be collection of devices. The analyst specifies how to collect the devices into subgroups via the the RevRun configuration files. Examples of subgroups could be collections of host devices such as PLCs, collections of critical outputs from the physical process, traffic between devices, etc. Subgroup scores are calculated as weighted averages of the scores for devices that comprise the subgroup.

The “group” level of the RevRun hierarchy has four defined groups: network, host, physical process, and security. Scores for the groups are calculated as weighted averages of the subgroups that comprise each group.

At the topmost level of the hierarchy, RevRun reports an overall system score for the ICS. This score is calculated as the weighted average of the four groups. All weights at the subgroup, group, and system level are specified by the analyst in the RevRun configuration file. See [16] for a more detailed description of RevRun and its components.

With proper configuration, RevRun automates the experiment, data collection, and metric processing. The automation not only makes execution and analysis of the experiment simpler and faster, but automation also makes the experiments repeatable, an important consideration for applying scientific methods to resilience analysis. Furthermore, RevRun is highly configurable, enabling the analyst to swap out SCEPTRE emulations, CALDERA attack profiles, and metrics as needed. This flexibility enables the analyst to more easily investigate a broad range of threats and ICSs.

2.4 Analysis Process

With the ADROC platform, one can use the following process for analyzing the resilience of ICSs to cyber threats:

(1) Select ICS(s) and threat(s) of interest.
(2) Implement the ICS representation(s) in the SCEPTRE environment.
(3) Configure CALDERA to represent the threat(s). Supplement CALDERA with additional plug-ins or payloads to address ICS-specific elements of the attack.
(4) Setup the RevRun configuration files to specify the SCEPTRE emulation, CALDERA profile, data to be collected, and metrics to be calculated.
(5) Execute RevRun. The experiment and data collection will proceed automatically.
(6) Analyze metrics and data produced and collected by RevRun.

The above process is intentionally general to maximize flexibility of the platform and process across different threats and ICSs. The following section presents a use case to demonstrate the more specific details that must be considered when applying the platform and analysis process to a specific system and set of threats.

3 USE CASE

This use case considers a hypothetical system comprised of a corporate network and SCADA network for monitoring and controlling the operation of a pressurized water reactor (PWR) in a nuclear power plant. Readers should note that the use case described is purely hypothetical and not intended to represent an actual system or attacker.

The corporate network is the portion of the system typically associated with IT and could include email servers, file servers, and workstations for engineers or business personnel. This portion of the network is used to perform common business operations that one would expect at most companies.

The SCADA network includes devices such as PLCs, RTUs, Open Platform Communications (OPC) servers, engineering workstations, human machine interfaces, and other devices commonly found in SCADA networks. These devices are used to monitor and control a PWR in a nuclear power plant. The operators attempt to enable both efficient production of electrical power and safe operating conditions. In the event that safe conditions cannot be maintained, reactor protection schemes are automatically enabled, forcing a halt to operations.

There is a demilitarized zone (DMZ) network which is the public-facing portion of the network, typically connected to the Internet. There are also servers here which could host websites or other services that the public can access.

Finally, there is a firewall located at the boundary of all three networks. Beyond the typical security concerns associated with public traffic reaching the internal network, connectivity between the corporate and SCADA networks can also be dangerous. Therefore, rules on the firewall are specially set to block almost all traffic from the corporate network to the SCADA network. However, it is not uncommon to allow a privileged device on the corporate network access to the SCADA network. This practice enables updates and maintenance of SCADA devices without requiring staff to be on-site to perform such maintenance. For the purposes of this use case, we assume that the firewall permits traffic from a single, privileged workstation on the corporate network to the SCADA network.

This use case also considers a hypothetical malicious attacker that aims to disrupt operation of the PWR. The attacker is assumed to have compromised and established a presence on a single workstation on the corporate network. This could happen, for instance, if someone in the corporate network downloaded infected software (e.g., [22]) or was the victim of a phishing attack. The attacker aims to pivot from this first machine to others in the corporate network, eventually compromising the privileged device. Once on this device, the attacker aims to pivot through the firewall onto an engineering workstation in the SCADA network. From there, the attacker has network visibility to much if not all of the SCADA network. The attacker is able to deploy a payload that will affect the control logic in one of the PLCs controlling the PWR. If successful, the attacker will ultimately cause unsafe operating conditions (e.g., extreme temperature or pressure), forcing a shutdown.

3.1 Analysis Goals and Questions

Given the large potential for variation of this hypothetical attack, several specific variants are specified for analysis. The goal of the analysis is to produce a ranked list of these attack variants according to the resilience of the PWR system to each attack. This type of ranking helps inform prioritization of concerns when system defense resources are limited.

In addition to ranking the attack variants, the analysis can address some additional questions:

(1) Which attack variants cause the greatest impact to the system? This question follows directly from the analysis goal, and the answer will help to direct system resources to address the most critical concerns.
(2) Can any of the attack variants cause unsafe operating conditions, which would activate reactor protection schemes?
(3) What can be learned about how to mitigate this attack type? Although it is not the primary goal of this analysis, the

data collected could help to inform design of an effective mitigation.

Specification of the analysis goals and questions informs how to configure the RevRun tool, what data to collect from the experiments, and how to define metrics which measure the system's resilience to the attack and ability to continue to operate safely.

3.2 Emulated Environment

3.2.1 System Emulation. The system for this use case was emulated using SCEPTRE. The emulated model consists of four primary components: the corporate network, the ICS, the DMZ, and the firewall. Note that many of the machines included are not modeled to run representative services (i.e. Microsoft Office, email, connection to the internet). Because the hypothetical attack being analyzed did not require this level of modeling detail, these details are left out for simplicity.

- *Corporate Network*: This portion of the network is intended to represent a typical IT network within a company. The model includes two Windows servers (Windows Server 2012) to represent an Exchange server and a Domain Controller, two Linux machines (Ubuntu 16.04) to represent a DHCP server and an FTP server, three Windows 7 workstations, one Kali Linux workstation, and one privileged Kali Linux workstation which is able to access the SCADA network. In addition, a Kali Linux machine was added to the corporate network to act as the attacker. Besides the details mentioned here, the workstations are all configured to be general purpose and to emulate only some prior regular use.

  All of the machines listed above were configured to be part of a single subnet. They connect to a router in the corporate network. This router in turn communicates to the firewall via an additional virtual local area network (VLAN).
- *ICS*: The ICS is comprised of the SCADA network and the PWR physical process simulator. The PWR is modeled using the Asherah simulator [13]. The simulator models a "two-loop 2,772 MWt pressurized water reactor including primary, secondary and tertiary loops as well as the control system" [13]. The key portion of the model is the Simulink simulator that models the physics of the reactor and computes pressure, temperature, and other states and properties of the system. The simulator also includes a simplified reactor protection system. For example, reactor protection schemes will be triggered if pressure in the PWR exceeds 8.97 MPa in the primary system or the reactor core temperature exceeds 580 degrees Kelvin. A departure from nucleate boiling ratio (DNBR) below 1.3 would also have negative effects of the PWR.

  The SCADA network monitoring and controlling the PWR model includes a variety of devices typically found in SCADA networks. Two PLCs in particular are worth noting: the reactor coolant pump PLC and steam generator pressure PLC. The reactor coolant pump helps maintain convective cooling in the primary system, so failure of that pump or PLC could result in extreme temperatures. The steam generator valve PLC opens and closes a valve to directly control pressure in the steam generator. Failures with the valve could cause extreme pressures to be realized or wet steam to destroy the generator's turbine. These PLCs were implemented with virtual machines running OpenPLC.

  Inclusion of a Kali Linux machine that serves as an engineering workstation is also worth noting. SCADA networks commonly include such a workstation to make updates to the system or to install patches on various components. For this use case, the SCADA workstation will also be used by the attack vector as a place from which to deploy the payload.

  Components in the SCADA network communicate over a variety of contained VLANs. Open communication is allowed between VLANs within the ICS network. One particular VLAN connected the Supervisory control machine, the SCADA workstation, and one of the firewall interfaces. This path is the only means for communication between the ICS network and any of the other subnets.

  The SCADA network implemented in the ADROC platform is simplified, relative to Silva's initial implementation. These simplifications were made because the level of redundancy and fidelity of the network is not necessary for the threat analysis and had no impact on simulated consequences analysis.
- *Demilitarized Zone (DMZ)*: Two machines were included in the DMZ for this use case. One was a Windows Server 2008 machine, intended to represent a backup data historian. The second was an Ubuntu Linux 14.04 machine meant to model a VPN. The DMZ was also on its own subnet which also included one of the firewall interfaces.
- *Firewall*: There is a firewall located at the boundary of all three subnets listed above. The firewall is configured to route any allowable traffic between subnets and to block all other traffic. Firewall configurations were based on points made in a 2005 survey paper by the British Columbia Institute of Technology [15]. In addition to these more standard firewall rules, a rule is added to allow SSH/SCP traffic between the privileged Linux machine on the corporate subnet and the Linux workstation on the ICS subnet. This represents an administrator setting up a shortcut that would allow them to configure machines on the ICS subnet remotely without physically touching the hardware.

3.2.2 Threat Emulation. This use case considers eight attack variants. All variants will be compared to a single baseline experiment in which no attack is run. A summary of the attack scenarios for each experiment can be seen in Table 1. All of the attacks are assumed to start on the same Windows workstation on the corporate network.

The two key dimensions along which the attack was varied were the amount of network topology knowledge available to the attacker and the nature of the attacker's final ICS-targeted payload.

(1) **Network Knowledge**: One variation assumes that the attacker has insider knowledge of which machine on the corporate network has privileges to connect through the firewall. This knowledge allows the attacker to take the minimal path through the corporate network to the privileged Linux machine. The other variation assumes that the attacker has no such knowledge. The attacker must move blindly through the

corporate network, potentially exploring the full network, before eventually pivoting to the privileged machine.

(2) **Nature of Payload**: The use case assumes that once the attacker arrives at the engineering workstation on the SCADA network, the attacker executes a data injection attack on one or both of the reactor coolant pump and steam generator pressure PLCs. The PLCs are targeted in one of four ways:
   (a) *Payload 1*: The pump speed is set to zero for the reactor coolant pump PLC. The expected effect is for temperatures to increase.
   (b) *Payload 2*: Close the steam generator valve entirely with the steam generator valve PLC. The expected effect is for pressures to increase.
   (c) *Payload 3*: Repeat the scenario modeled in [18] ; that is, change the flow sensor input on the reactor coolant pump PLC to a constant 11,000 kg/s and hold the steam turbine speed constant on the other PLC. This causes the PLC to slow the coolant pump down until the speed reaches 0.
   (d) *Payload 4*: Toggle the flow rate input value for the reactor coolant pump PLC between 0 and 100. This attack is intended to simulate a broken sensor and appear less like a malicious attack.

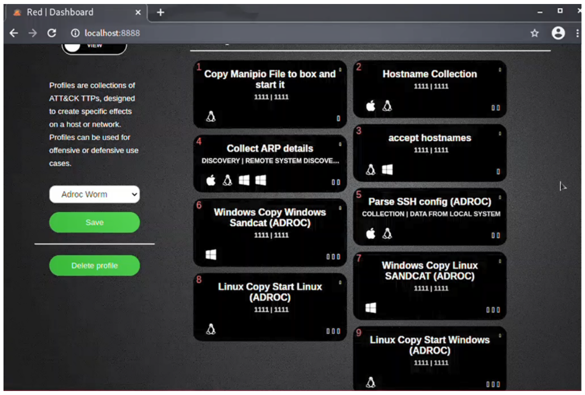
The use case considers a total of eight attack scenarios (two knowledge variants times four payload variants).

Scenario ID	Network Path	Payload
Baseline/ Attack 0	N/A	N/A
Attack 1	Full Attack Path	Payload 1
Attack 2	Full Attack Path	Payload 2
Attack 3	Full Attack Path	Payload 3
Attack 4	Full Attack Path	Payload 4
Attack 5	Minimal Attack Path	Payload 1
Attack 6	Minimal Attack Path	Payload 2
Attack 7	Minimal Attack Path	Payload 3
Attack 8	Minimal Attack Path	Payload 4

Table 1: Summary of Attack Variants

The attacks described above are implemented using CALDERA and ManiPIO. CALDERA includes several, pre-installed and configured attacker profiles. Several of those profiles perform worm-like series of techniques in order to pivot through a network. For this use case, some modifications were required to the worm profile in order to run as intended on the use case system. However, the core functionality is still present, and most of the techniques included in the profile can be traced back to MITRE's ATT&CK framework. The techniques used by the adversary developed for this use case are summarized in Table 2. Figure 1 provides a screenshot of the specific adversary profile specified for this use case, and Figure 2 describes the agent's decision logic and control flow.

The attack experiments are initiated by installing a CALDERA agent on one of the corporate workstations. This agent communicates via command and control (C2) beacons with the CALDERA server, which acts as a remote C2 server. As the attack progresses and the worm propagates through the system, more agents will be created on other machines and will also beacon back to the CALDERA server. All agents receive instructions from CALDERA about commands to run and then report the resulting output back to the CALDERA server.



Profiles are collections of ATT&CK TTPs, designed to create specific effects on a host or network. Profiles can be used for offensive or defensive use cases.

Adroc Worm

Save

Delete profile

1 Copy Manipio File to box and start it
1111 | 1111

2 Hostname Collection
1111 | 1111

3 accept hostnames
1111 | 1111

4 Collect ARP details
DISCOVERY | REMOTE SYSTEM DISCOVE...

5 Parse SSH config (ADROC)
COLLECTION | DATA FROM LOCAL SYSTEM

6 Windows Copy Windows Sandcat (ADROC)
1111 | 1111

7 Windows Copy Linux SANDCAT (ADROC)
1111 | 1111

8 Linux Copy Start Linux (ADROC)
1111 | 1111

9 Linux Copy Start Windows (ADROC)
1111 | 1111

Figure 1: Screenshot of CALDERA Adversary Profile.

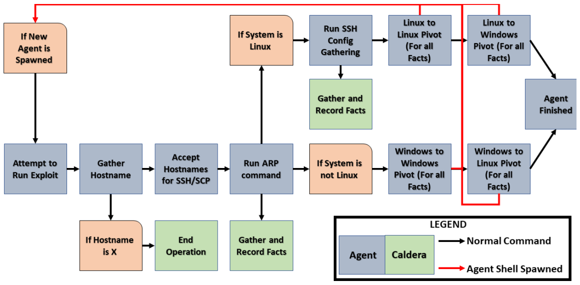


Figure 2: CALDERA Agent Logic and Control Flow.

The manner in which the worm pivots through the network depends on the information initially provided to it. In the attack scenarios with insider knowledge, the added information available to the worm allows it to pivot through the corporate network in a more direct path.

Once the worm has traversed the corporate network and pivoted through the firewall, it reaches the SCADA workstation. CALDERA was configured to recognize the hostname on this machine and to stop any further worm propagation. From here, the phase of the attack shifts from a worm to payload execution. CALDERA and ManiPIO were integrated in this use case such that CALDERA treats the ManiPIO script as a malware payload, executing it on a target machine. In this case, the target machine was the SCADA workstation because it has network visibility to the PLCs on the ICS network. ManiPIO is configured in each experiment according to the payload variations described above.

3.3 RevRun Configuration

The configuration of this use case is described in more detail below.

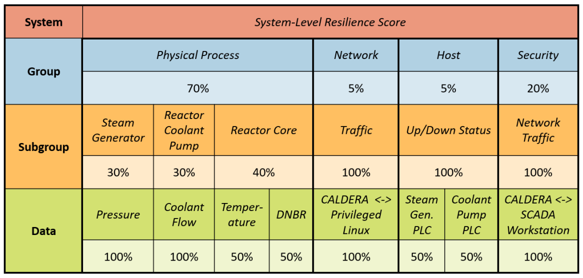
- *Experiment Control*: When setting up the emulation platform, a period of time is required to initialize the VMs, firewall, Asherah simulator, etc. Hence, the CALDERA attacks do not begin until 10 minutes after the initialization process begins. This helps ensure that scenario impacts in the data

Technique ID	Technique Name	Use Case Application
T1005	Collection - Data from Local System	Collect Hostname to Check for Target Machine
T1059	Code Execution - Command and Scripting on Unix Shell	Copy ManiPIO to Target and Run
T1570	Lateral Movement - Lateral Tool Transfer	Copy Agent Code to Target and Run
T1780	Privilege Escalation - Valid Local Account	Move through firewall via Admin privileges

Table 2: ATT&CK Framework Techniques in this Use Case

are separable from initialization artifacts. Experiments were each run for 50 minutes total. This time was selected allow ample time for the attack to initialize, run fully, and for potential impacts on the physical process to be realized.

- *Data Collection*: RevRun can collect data from a variety of sources, including data from the physical process simulation, from hosts, and from network traffic. For this use case, collected data included reactor pressures, temperatures, and DNBR; beacon traffic between the CALDERA server and infected machines; and network reachability of the Reactor Coolant Pump and Steam Generator PLC hosts.
- *Data Processing*: Collected data is processed in a variety of ways in order to best reflect the impact of the given attack scenarios on the system data. Below, the processing functions applied to each data stream are described in Table 3. In addition, data processing also aligns the timelines of all data in order to make the results of each experiment more easily comparable.
- *System Resilience Metrics*: The processed data as described above is weighted and aggregated in a hierarchical manner in order to determine the system-level resilience score for each experiment. Recall that these scores are on a scale from zero to one, with one being most resilient. System resilience scores are calculated in a hierarchical manner according to groupings defined in the RevRun configuration. For this use case, groupings are broken down first by data source, then by the type of information stored in the data, and finally by device. Figure 3 below summarizes how this hierarchy was defined and weighted for this use case.



System	System-Level Resilience Score							
Group	Physical Process				Network	Host		Security
	70%				5%	5%		20%
Subgroup	Steam Generator	Reactor Coolant Pump	Reactor Core		Traffic	Up/Down Status		Network Traffic
	30%	30%	40%		100%	100%		100%
Data	Pressure	Coolant Flow	Temper-ature	DNBR	CALDERA <-> Privileged Linux	Steam Gen. PLC	Coolant Pump PLC	CALDERA <-> SCADA Workstation
	100%	100%	50%	50%	100%	50%	50%	100%

Figure 3: Configured Score Hierarchy for this Use Case. Percentages at Each Level Show how Components were Weighted in the Score at the Next Level Up.

3.4 Experiment Results

Figure 4 shows the system-level resilience scores for each scenario. The most striking conclusion to be drawn here is that Attacks 1 and 5 have, by far, the lowest resilience scores. These two scenarios both used Payload 1, which set the coolant pump speed to zero. No other attack scenarios seemed to cause the same magnitude of impact to the resilience of the system.

An additional point of interest in Figure 4 is that the Attacks which used the minimum path through the corporate network caused uniformly lower system resilience compared to their counterpart scenarios which took more time in the corporate network. The speed of the worm does impact the resilience of the system to attack.

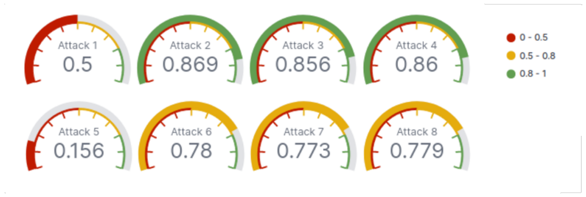


Figure 4: System-level Resilience Scores for Each Attack Scenario.

Given this high-level view of the experiment results, the next step is to analyze the experimental data that is used to calculate the system level metrics. Figures 5 and 6 show the pressure and temperature readings respectively from the system under each scenario. The black horizontal line in each image denotes the established safety condition threshold for each reading. Discounting system initialization in the first 10 minutes, the only scenarios that ever cross the given thresholds are Attacks 1 and 5, meaning these scenarios would lead to unsafe operating conditions and subsequent shutdown of the PWR. This observation is consistent with what was observed in the resilience scores. Since the resilience metrics are based so heavily on whether safety thresholds were crossed, it makes sense that these attacks led to the greatest decrease in system resilience overall.

Figures 7 and 8 show the reachability, or up/down status, of the two PLCs. These figures shows that both PLCs remained online for the duration of every attack scenario. This observation can be used to confirm that the impacts on the PWR did not occur because the PLCs were offline; instead, a malfunction or data injection could be the source of the issue. Furthermore, from an experimental monitoring perspective, one can verify that the attacks described, which were not intended to take either PLC offline, did not have that unintended consequence.

Figure 9 shows traffic between the targeted SCADA workstation in the ICS network and the CALDERA server. This data is an effective way to see when the attacker first reached the target machine.

Data Source	Metric
Pressure	1st time pressure exceeds 8.974 MPa
Reactor Core Temp	1st time temp exceeds 580K
DNBR	1st time DNBR drops below 1.3
PWR Coolant Flow	Cumulative diff between nominal and attack values
Traffic between C2 server and privileged device	Cumulative packet count
Traffic between C2 server and SCADA Workstation	time 1st packet is sent
Steam Generator PLC Status	up/down status
Coolant Pump PLC Status	up/down status

Table 3: Summary of Use Case Data Processing

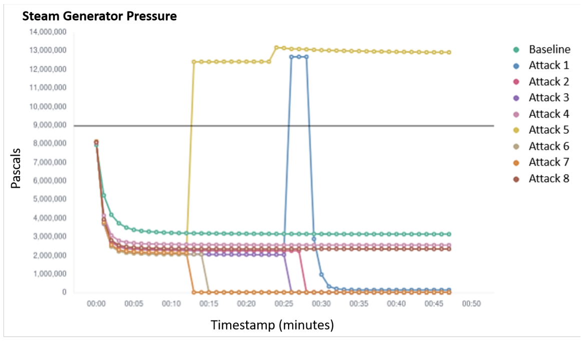


Figure 5: Pressure Data for Each Attack Scenario. Horizontal black lines represent safety threshold of 8.97MPa.

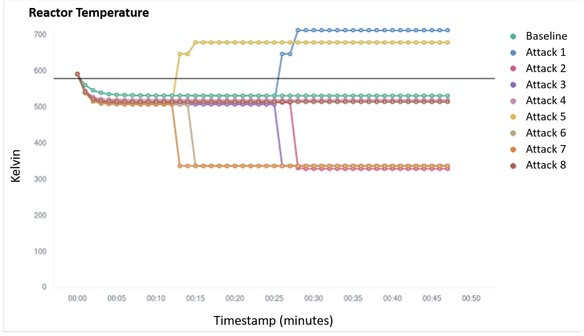


Figure 6: Temperature Data for Each Attack Scenario. Horizontal black lines represent safety threshold of 580K.

There are two clusters of connections amongst the scenarios. One set occurs around 12 minutes into the experiments, when the data for Attacks 5, 6, 7, and 8 first start to show traffic to the CALDERA server. The next is around 26 minutes into the experiments, when the data for Attacks 1, 2, 3, and 4 follow suit. The difference between these two clusters of attacks is whether the worm used the known minimum path to the privileged machine on the corporate network or not. Recall that the worm is not started until 10 minutes into the experiment. When the worm used this direct path, the attacker took around 2 minutes to reach the target machine. Without this added information, the attacker took around 16 minutes to reach the same goal.

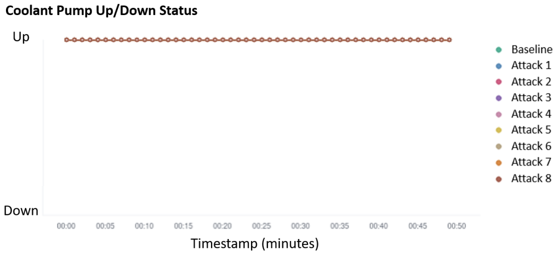


Figure 7: Up/Down Status for the Reactor Coolant Pump PLC During Each Attack Scenario.

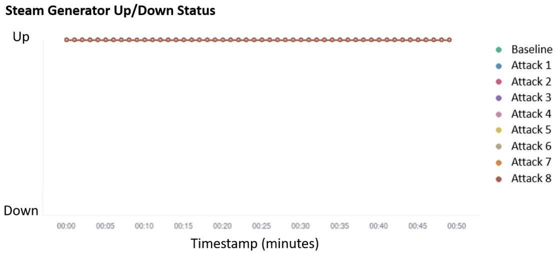


Figure 8: Up/Down Status for the Steam Generator PLC During Each Attack Scenario.

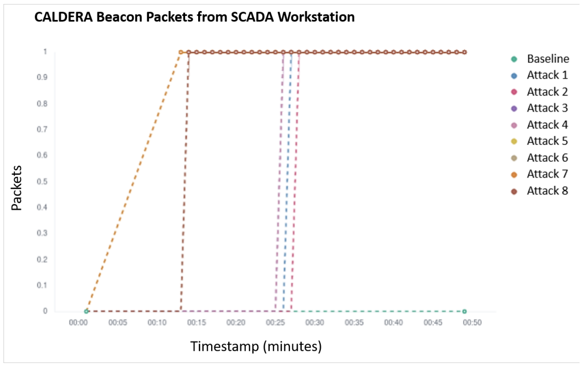


Figure 9: CALDERA Command and Control Beacon Traffic from the SCADA Workstation During Each Attack Scenario.

It is clear from these results that the greatest determinant of attacker impact to the system is the nature of the final payload. Of the payloads tested, the most effective one only targeted the Reactor Coolant Pump PLC. The direct change to the pump speed caused by this payload would have a cascading effect on other properties of the system, most notably coolant flow. In addition, the network data reflects the fact that level of network knowledge significantly impacts the attacker's speed through the system. The key to detecting this type of attack might be in identifying malicious C2 beacons. The fact that the speed of the attack changes so drastically (by about 14 minutes) depending on level of network knowledge could indicate two potential paths forward for the use case system. Either all malicious C2 traffic needs to be caught and mitigated within 2 minutes to stop even a well-informed attacker, or network configuration should be better obfuscated to give system administrators more time to detect such an attacker before they pivot to more critical portions of the system.

4 CONCLUSIONS

Recent cyber attacks to critical infrastructure have brought the security of these systems into the spotlight. A variety of resilience metrics have been proposed for cyber systems, but populating the metrics with data is frequently a challenge. Without a means for gathering such data, resilience assessment methods for critical infrastructure and other cyber-physical systems will be limited. Reliable and quantitative methods for evaluating the resilience of industrial control systems to various attacks is crucial for understanding the real threat these attacks pose to critical systems.

The ADROC platform addresses this need in a way that is highly automated, repeatable, and flexible. The output provides quantitative evidence to help understand a system and direct resources to improve that system's resilience to attack. The ADROC platform not only automatically calculates resilience metrics according to user-specified configurations, but it does so using data generated from cyber experiments. By generating data needed in cyber resilience assessment, one can start to understand why resilience scores are higher/lower for an ICS threat and how to start mitigating that threat. This understanding is as important, if not more so, than the actual score itself.

This paper includes a use case to demonstrate how the ADROC platform can be used to analyze the resilience of a PWR in a nuclear power plant to a set of hypothetical attacks. Though the rankings of the attacks are specific to this PWR system and the specific attacks studied, the ADROC platform can be applied to a much more general set of studies. When using the ADROC platform for resilience analyses, the analyst has the flexibility to choose the ICS of interest, the threats of interest, and the metrics with which to quantify resilience of the ICS. The primary requirements for ADROC are that the ICS is represented within the SCEPTRE emulation platform. Though the authors used CALDERA and ManiPIO to emulate the threats in the use case, doing so is not a requirement for using the ADROC platform. Furthermore, if the analyst wants to use resilience metrics that are not currently implemented in ADROC, the analyst can easily supplement the metrics library by adding some additional Python functions. The flexibility to swap out ICSs, threats, and metrics gives ADROC broad applicability to a large set of potential studies.

ACKNOWLEDGMENTS

The authors appreciate Chris Mairs's thoughtful comments and review. This work has been supported by the LDRD Program at the Sandia National Laboratories. Specifically, the Resilient Energy Systems Mission Campaign sponsored this research. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

REFERENCES

[1] [n.d.]. Elastic Stack. http://www.elastic.co/.
[2] [n.d.]. Elasticsearch. https://www.elastic.co/products/elasticsearch.
[3] [n.d.]. Kibana. http://www.elastic.co/products/kibana.
[4] 2017. *CRASHOVERRIDE Analysis of the Threat to Electric Grid Operations.* Technical Report version 2.20170613. Dragos Inc., Hanover, MD.
[5] 2017. *TRISIS Malware Analysis of Safety System Targeted Malware.* Technical Report version 1.20171213. Dragos Inc., Hanover, MD.
[6] T. Alpcan and T. Basar. 2004. A game theoretic analysis of intrusion detection in access control systems. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, Vol. 2. 1568–1573 Vol.2.
[7] D. Bodeau, R. Graubart, W. Heinbockel, and E. Laderman. 2015. *Cyber Resiliency Engineering Aid –The Updated Cyber Resiliency Engineering Framework and Guidance on Applying Cyber Resiliency Techniques.* Technical Report. MITRE Corporation, Bedford, MA.
[8] D. Bodeau, R. Graubart, R. McQuaid, and J. Woodill. 2018. *Cyber Resiliency Metrics and Scorind in Practice– Use Case Methodlogy and Examples.* Technical Report. MITRE Corporation, Bedford, MA.
[9] D. Bodeau, R. Graubart, R. McQuaid, and J. Woodill. 2018. *Cyber Resiliency Metrics Catalog.* Technical Report. MITRE Corporation, Bedford, MA.
[10] D. Bodeau, R. Graubart, R. McQuaid, and J. Woodill. 2018. *Cyber Resiliency Metrics, Measures of Effectiveness, and Scoring.* Technical Report. MITRE Corporation, Bedford, MA.
[11] MITRE Corporation. 2021. ATT&CK. Retrieved December 8, 2021 from https://attack.mitre.org/
[12] MITRE Corporation. 2021. CALDERA. Retrieved December 8, 2021 from https://www.mitre.org/research/technology-transfer/open-source-software/caldera%E2%84%A2
[13] R. Busquim e Silva. 2019. Development of the Asherah Nuclear Power Plant Simulator for Cyber Security Assessment. In *Proceedings of the International Conference on Nuclear Security 2020.*
[14] R. Fasano, C. Lamb, M. El Genk, T. Schreiner, and A. Hahn. 2020. Emulation methodology of programmable logic controllers for cybersecurity applications. In *ASME International Conference on Nuclear Engineering.*
[15] Group for Advanced Information Technology at British Columbia Institute of Technology. 2005. *Firewall Deployment for SCADA and Process Control Networks: Good Practice Guide.* Technical Report. Centre for the Protection of National Infrastructure, Burnaby, B.C, Canada.
[16] M. Galiardi, A. Gonzales, J. Thorpe, E. Vugrin, R. Fasano, and C. Lamb. 2020. Cyber Resilience Analysis of Scada Systems in Nuclear Power Plants. In *Proceedings of the 2020 28th Conference on Nuclear Engineering & Joint With the ASME 2020 Power Conference.*
[17] H. Goldman. 2010. Building Secure, Resilient Architectures for Cyber Mission Assurance. In *Proceedings of the Secure and Resilient Cyber Architectures Conference.* McLean, Virginia, 422–431. https://doi.org/99.9999/woot07-S422
[18] A. Hahn, D. Sandoval, R. Fasano, and C. Lamb. 2021. Automated Cyber Security Testing Platform for Industrial Control Systems. In *Proceedings of the 12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies Conference.*
[19] M. Haque, S. Shetty, and B. Krishnappa. 2019. *Cyber-Physical System Resilience.* John Wiley and Sons, Ltd, Chapter 12, 301–337.

[20] Sandia National Laboratories. 2016. SCEPTRE. Retrieved December 8, 2021 from https://www.osti.gov/servlets/purl/1376989

[21] I. Linkov, D. Eisenberg, K. Plourde, T. Seager, J. Allen, and A. Kott. 2013. Resilience metrics for cyber systems. *Environ Sys Decis* 33 (2013), 471–476.

[22] Pierluigi Paganini. 2014. Malware based attack hit Japanese Monju Nuclear Power Plant. Retrieved December 16, 2021 from https://securityaffairs.co/wordpress/21109/malware/malware-based-attack-hit-japanese-monju-nuclear-power-plant.html

[23] Colonial Pipeline. 2021. Media Statement Update: Colonial Pipeline System Disruption. Retrieved November 18, 2021 from https://www.colpipe.com/news/press-releases/media-statement-colonial-pipeline-system-disruption

[24] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid. 2019. *Developing Cyber Resilient Systems: A Systems Security Engineering Approach.* Technical Report version 1. National Institute of Standards and Technology, Gaithersburg, MD.

[25] J. Slowik. 2019. *Stuxnet to CRASHOVERRIDE to TRISIS: Evaluating the History and Future of Integrity-Based Attacks on Industrial Environments.* Technical Report. Dragos Inc., Hanover, MD.

[26] D. Snyder, L. Mayer, G. Weichenberg, D. Tarraf, B. Fox, M. Hura, S. Genc, and J. Welburn. 2020. *Measuring Cybersecurity and Cyber Resiliency.* Technical Report. RAND Corporation, Santa Monica, CA.

[27] E. D. Vugrin, J. Cruz, C. Reedy, T. Tarman, and A. Pinar. 2020. Cyber threat modeling and validation: port scanning and detection. In *Proceedings of the 7th Symposium on Hot Topics in the Science of Security.* ACM.