INL/RPT-23-75111
Revision 000

Implementation of an ICS Ransomware Testbed

Scenarios, Variants, and Evaluation Methods

September 2023

M3CT-23IN1104033

Idaho National Laboratory

Chris Spirito
Ian King
Parker Naugle
Tra-My Ho
Tori Simon

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory
operated by Batelle Energy Alliance, LLC

INL/
Revision 000

Implementation of an ICS Ransomware Testbed

M3CT-23IN1104033

Idaho National Laboratory
Chris Spirito
Tra-My Ho
Ian King
Parker Naugle

September 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Engineering
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Page intentionally left blank

Page intentionally left blank.

CONTENTS

FIGURES

Page intentionally left blank.

ACRONYMS

ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
BTC	Bitcoin
DNP3	Distributed Network Protocol 3.0
ETH	Ethereum
FBI	Federal Bureau of Investigations
HMI	Human-Machine Interface
ICS	Industrial Control Systems
ICS-CERT	Industrial Control System-Cyber Emergency Response Team
IC3	(FBI) Internet Crime Complaint Center
IDS	Intrusion Detection System
IoC	Indicators of Compromise
IP	Internet Protocol
IPS	Intrusion Prevention System
KB	Kilobyte
MPCI	Malicious Parameter Command Injection
OS	Operating System
OT	Operational Technology
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SE	Secure Energy
SSH	Secure Shell
TCP	Transmission Control Protocol
TNPP	Townsville Nuclear Power Plant (fictitious)
USB	Universal Serial Bus
VLC	Visibility, Likelihood, and Consequence
VM	Virtual Machine
VPN	Virtual Private Network

Page intentionally left blank.

Implementation of an ICS Ransomware Testbed

1. Introduction

Ransomware attacks on Industrial Control Systems (ICS) have emerged as a formidable threat to the United States' critical infrastructure, eliciting grave concerns regarding national security. In March 2023, the FBI Internet Crime Complaint Center (IC3) unveiled its 2022 Internet Crime Report, highlighting a concerning 870 complaints related to ransomware impacting U.S. critical infrastructure. Of the country's 16 critical infrastructure sectors, 14 encountered at least one ransomware attack. Notably, while the Healthcare and Public Health sector suffered the most, reporting 210 attacks, sectors pivotal to ICS networks and governmental organizations were also targeted: the Defense Industrial Base reported 1 attack, Water and Wastewater Systems 3, Chemical 19, Energy 15, Government Facilities 115, and Critical Manufacturing 157. For instance, a ransomware attack on a major chemical company could jeopardize not only its production but also pose environmental risks should systems controlling hazardous materials be compromised. In 2022, three ransomware variants predominantly targeted U.S. critical infrastructure: HIVE, with 87 attacks; ALPHV/BlackCat, with 114; and LOCKBIT, with 149. Several cyber-attacks, such as the MOVEit data breach in May 2023 and the Colonial Pipeline ransomware attack in May 2021, have been so impactful that they commanded national attention.

The DarkSide hacking group's assault on the Colonial Pipeline, initiated on May 6th, 2021, stands as one of the most substantial and publicly acknowledged cyber-attacks against U.S. critical infrastructure. The group exploited an exposed Virtual Private Network (VPN) password, paving the way for initial intrusion and subsequent data theft. A mere day later, DarkSide unleashed a ransomware attack that compromised vital accounting and billing systems, prompting an immediate shutdown of the pipeline to mitigate further ransomware proliferation across its network. This crisis spurred a robust response from the U.S. president and regulators, culminating in a national emergency declaration related to the pipeline shutdown on May 9th, 2021. This incident mirrors the 2017 NotPetya ransomware attack that significantly impacted the shipping giant Maersk, highlighting an urgent need for fortified cybersecurity across various industries. Future incidents, akin to the Colonial Pipeline attack, could potentially be mitigated—or entirely averted—should government agencies and private entities scrutinize system vulnerabilities, exploring various ransomware types and entry points. Proactive measures, such as conducting experiments on VPN accounts or auditing passwords to pinpoint duplicate usage across diverse systems and software, might illuminate feasible entry points and vulnerability zones within an organization's systems.

Leveraging configurable virtual testbeds enables private companies and government organizations to meticulously assess the vulnerabilities of Industrial Control Systems (ICS) against various types of ransomware attacks, such as the notorious Ryuk or WannaCry. For instance, a virtual testbed enabled Georgia Tech researchers to simulate a power grid attack in 2018, offering invaluable insights into potential weaknesses and mitigation strategies. Deploying a virtual testbed allows organizations to finely tailor network environments and experimental scenarios to their unique specifications and needs, irrespective of the network or system infrastructure—like creating a simulated environment mimicking an electric utility's ICS to probe for specific vulnerabilities. Moreover, it provides network security personnel the capability to navigate the continually evolving threat landscape of malware and

ransomware, facilitating both rapid adaptation—such as swiftly altering security protocols in response to new threat intelligence—and the development of nuanced threat predictions, like forecasting the potential impact of emerging ransomware strains based on historical data. Employing a virtual testbed negates the need for constantly updated physical systems and allows for nimble adjustments. This flexible virtual network, which might replicate various real-world ICS configurations (e.g., water treatment plants, manufacturing lines), presents an unimpeded view into the impacts of exploits, solidifying itself as a potent alternative to static hardware testbeds and thereby ensuring organizations can proactively and effectively safeguard against cybersecurity threats.

This report navigates the critical realm of cybersecurity, specifically focusing on conducting simulated ransomware attacks within a modular virtual testbed, with an overarching goal of empowering Industrial Control Systems (ICS) security personnel to pinpoint and implement decisive threat mitigation and defense tactics within their networks. To illustrate, consider the potent ransomware attack on the City of Atlanta in 2018, which compromised numerous critical services, signaling an alarming vulnerability that underscores the necessity for adept threat mitigation. This report charts a practical path, guiding users through the nuances of virtual testbed development and offering several potent test scenarios utilizing this innovative platform. Furthermore, it includes evaluative methodologies considering the visibility, likelihood, and impact of ransomware attacks on ICS systems, thereby aiding the formulation of a framework for the standardization of organizational threat analysis, akin to the structured approach adopted post-WannaCry 2017 attack to buttress cybersecurity resilience across organizations globally. Delving into practical applications, Section 5 elucidates risk mitigations and best practices specifically tailored for the nuclear community, thereby providing pragmatic applications gleaned from scenario testing and pertinent research. Moreover, by employing the methodologies expounded within this report, further recommendations and best practices can be curated and customized, allowing for tactical adaptation to address distinct areas of concern within an organization.

2. Existing Ransomware Testbeds

The underbelly of cybersecurity, particularly in the realm of Industrial Control Systems (ICS) and Industrial Internet of Things (IIoT), has seen a burgeoning challenge arise from ransomware attacks, demanding a meticulous and pragmatic approach to understanding and mitigating these threats. Across the globe, from the academia-industry nexus in Belfast to the technical hubs in Tokyo, numerous institutions have embarked on a journey to decipher the anatomy of ransomware attacks and formulate robust defense mechanisms. For instance, the Centre for Secure Information Technology in Belfast has implemented a network capturing capability within their testbed, efficiently extracting salient features from the Locky ransomware to understand its behavioral propensities, while at the Tokyo Metropolitan University, real ransomware is put under the microscope within a safeguarded environment to deeply inspect sequences of API calls, revealing intrinsic details about ransomware behavior. These endeavors have culminated in the creation of diverse virtual and physical testbeds, each with unique capabilities and focus areas, systematically simulating, analyzing, and mitigating potential ransomware attacks on ICS and related infrastructures. In this section, we embark on an explorative journey through a myriad of testbeds developed by various institutions and researchers, gleaning insights from their capabilities, approaches, and findings, to foster a rich understanding and amalgamate global wisdom on tackling ransomware threats in ICS environments.

Traversing through the echelons of global research and practical experiments in understanding and mitigating ransomware threats, it becomes palpably evident that the cybersecurity community is ardent in its pursuit to safeguard ICS and related networks. From deploying real and virtual machines to examining the minutiae of malware communication via HTTP traffic and developing models that scrutinize API call sequences or analyze kernel-related activities, each testbed and research endeavor proffers unique insights and strategies to fortify cyber-physical systems against malicious ransomware attacks. Thus, as we distill learnings from diverse geographical and technical landscapes—from the systematic extraction and analysis of behavioral features of ransomware like Locky to scrutinizing the malicious communication and components of notorious ransomware like WannaCry—it is pivotal to amalgamate these findings and technologies to devise a holistic, adaptive, and fortified strategy to shield our critical infrastructures from the ever-evolving and escalating threats that loom in the digital shadows.

Sampling of Ransomware Testbeds

Centre for Secure IT, Queen's University Belfast & Computer Center, University of Baghdad
Testbed Capabilities • Components: (3) real and (2) virtual machines • Capable of capturing network traffic and storing it in PCAP files for analysis • Connected to the Internet for Sampling and Information ingest.
Findings 18 features extracted from Locky ransomware, used to extract potential behavioral features, high detection accuracy of multi-classifier network-based ransomware detection method at 97.92% for the packet level and 97.08% for the flow level.
Warsaw University of Technology, Institutes of Computer Science and Telecommunications
Testbed Capabilities • Components: All virtual machines, Cuckoo Sandbox for security, Microsoft Windows 7 target, additional subnetworks for separation • Capable of capturing memory snapshots • Modules for inspecting packets (e.g., HTTP, ...) • Used to analyze and block suspected ransomware
Findings Promising approach to ransomware detection is to detect malicious communication between infected host and attacker using HTTP traffic, detection rates of 97-98%.
Deakin University and UNSW Canberra
Testbed Capabilities • Concentration on IIoT edge gates to enhance connectivity and data transfer. • Architecture grounded in brownfield IIoT, ensuring compatibility and integration ease. • Adoption of the Industrial Internet Reference Architecture, spotlighting a cyber-physical closed-loop system for holistic data collection and analysis. • Utilization of I/O devices to steadfastly monitor critical parameters, including temperature and pressure. • Incorporation of controllers to analytically process data and direct commands to actuators, ensuring responsive system performance.

- Strategic data management through local databases and selective data transmission to the Azure cloud for optimized storage and access.
- Empowerment of operators with a Web portal for proficient monitoring and control, alongside an alert system for supervisory notifications.

Findings

Kernel-related activity parameters serve as notable indicators of aberrant behavior stemming from ransomware attacks, and the implementation of a logic lock may jeopardize system safety, reliability, and resiliency by potentially introducing vulnerabilities or instabilities into the operational framework.

Department of Electrical Engineering and Computer Science, Tokyo Metropolitan University

Testbed Capabilities

- Components: The host machine utilizes Ubuntu 16.04 LTS and conducts experiments with actual ransomware.
- Deployment of Cuckoo Sandbox for executing ransomware within an isolated virtual environment on the host machine.
- Enables the collection of API logs to scrutinize ransomware activity and impact

Findings

The implementation of an SVM-based scheme to meticulously inspect the sequences of API calls revealed that ransomware could be detected through the analysis of q-gram numbers, providing a methodological approach to identify and understand ransomware behaviors.

School of Computer Science and Engineering, VIT-AP University, Madras Institute of Technology Anna University, and Indian Institute of Technology Roorkee

Testbed Capabilities

- Components: Server equipped with Software Guard Extension and operating on an Ubuntu server platform.
- Engages in dynamic monitoring and behavior analysis throughout the testing phase.
- Conducts runtime monitoring of malware execution to understand its infiltration and operational tactics.
- Employment of a Software-Defined Networking (SDN) controller to verify and manage network interactions and traffic.

Findings

The utilization of a "Honeyfolder" strategy for file system monitoring has proven notably effective in attracting and identifying host ransomware. Moreover, enhancing the infrastructure with Software-Defined Networking (SDN) has substantiated improvements in overall network protection, offering a more secure environment against ransomware attacks.

Information Systems Frontiers

Testbed Capabilities

- Components: The setup comprises a Linux server and two Windows 7 client machines.
- Constructed to offer a realistic environment that facilitates the deployment of malware and allows for the collection of real-time base metal host logs.
- Enables collection of logs both under standard operating conditions and during ransomware activities to discern discrepancies and malicious actions.

Findings
The BiLSTM-FC model has demonstrated its capability to efficiently model the normalcy of an operational enterprise network, providing a baseline against which anomalies from ransomware can be detected. Additionally, the DeepRan method can adeptly categorize ransomware events into existing families, utilizing the BiLSTM-CFR model to enable a structured and informative classification mechanism.

University of York and WCL Department of Electrical and Computer Engineering

Testbed Capabilities

- Components: Utilizes a host machine functioning as a virtual switch, running REMnux, and two VMs with Windows 7 SPI.
- One VM is infected with the WannaCry ransomware while the other serves as a control for comparative analysis.
- Specifically designed for the reverse engineering and thorough analysis of malware, with a particular focus on observing DNS queries made by the WannaCry ransomware.

Findings
The WannaCry ransomware is found to be composed of two distinct components, enabling it to self-propagate in a worm-like fashion alongside executing encryption processes. The ransomware utilizes Tor addresses for Command and Control (C&C) communication, and importantly, it harbors an embedded RSA key utilized to decrypt the malicious DLL, showcasing a complex and multi-faceted threat mechanism.

Traversing through the echelons of global research and practical experiments in understanding and mitigating ransomware threats, it becomes palpably evident that the cybersecurity community is ardent in its pursuit to safeguard ICS and related networks. From deploying real and virtual machines to examining the minutiae of malware communication via HTTP traffic and developing models that scrutinize API call sequences or analyze kernel-related activities, each testbed and research endeavor proffers unique insights and strategies to fortify cyber-physical systems against malicious ransomware attacks. Thus, as we distill learnings from diverse geographical and technical landscapes—from the systematic extraction and analysis of behavioral features of ransomware like Locky to scrutinizing the malicious communication and components of notorious ransomware like WannaCry—it is pivotal to amalgamate these findings and technologies to devise a holistic, adaptive, and fortified strategy to shield our critical infrastructures from the ever-evolving and escalating threats that loom in the digital shadows.

3. How to Establish a Testbed

3.1 Create a Virtual Testbed

This paper articulates a method for devising an easily deployed, modular, and versatile virtual testbed, an imperative tool for the exploration and experimentation of network environments in a secure, controlled setting. The crafted testbed deploys a lightweight physical machine, specifically utilizing a commercially available, mid-grade HP laptop operating on Windows 11, ensuring cost-effectiveness and accessibility for a wide range of researchers or cybersecurity professionals. Utilizing Oracle VM VirtualBox, the virtual environment hosts a majority of simulated machines that run on the Linux Mint distribution, a choice strategically made due to its low resource utilization, high customizability, and user-friendly interface. Moreover, the testbed comprises four Mint machines—representing Industrial Control Systems

(ICS) devices—and one Kali Linux machine, symbolizing an attacker, thereby creating a landscape that echoes real-world ICS network vulnerabilities. The architecture of the testbed involves a bifurcation into two networks, each employing a distinct host-only network adapter to ensure insulation from public networks, and thus, safeguarding against external interference. Within this network dichotomy, the first network (Network 1) envelops the ICS Client system and ICS Server 1, while the second (Network 2) encompasses ICS Server 2 and the Kali attack simulation system. This configuration is highly adaptable, permitting augmentation through the incorporation of new machines, network adapter reconfiguration, and script additions.

In a meticulous endeavor to simulate an ICS network, each machine is imbued with pre-configured capabilities. The Mint Client harbors a Python script, enabling it to probe the Mint Servers for data, exemplifying a typical client-server interaction in ICS environments. Concurrently, each Mint Server is equipped with a script that facilitates information access, thereby enhancing the network's interactive capacity. Additionally, the SCADA router machine is configured with ScadaBR, an open-source SCADA (Supervisory Control and Data Acquisition) simulation program which facilitates data visualization using the Tomcat web host. On the offense side, the Kali machine, renowned for its robust suite of cyber-attack tools, allows for extensive penetration testing without necessitating additional configuration, providing a practical platform to explore and understand potential cyber threats to an ICS network. This detailed network topology, illustratively visualized in Figure 1, is supplemented by a comprehensive guide, ensuring proficient testbed configuration on a Windows machine, thereby providing a holistic tool and guide for cybersecurity research and education:

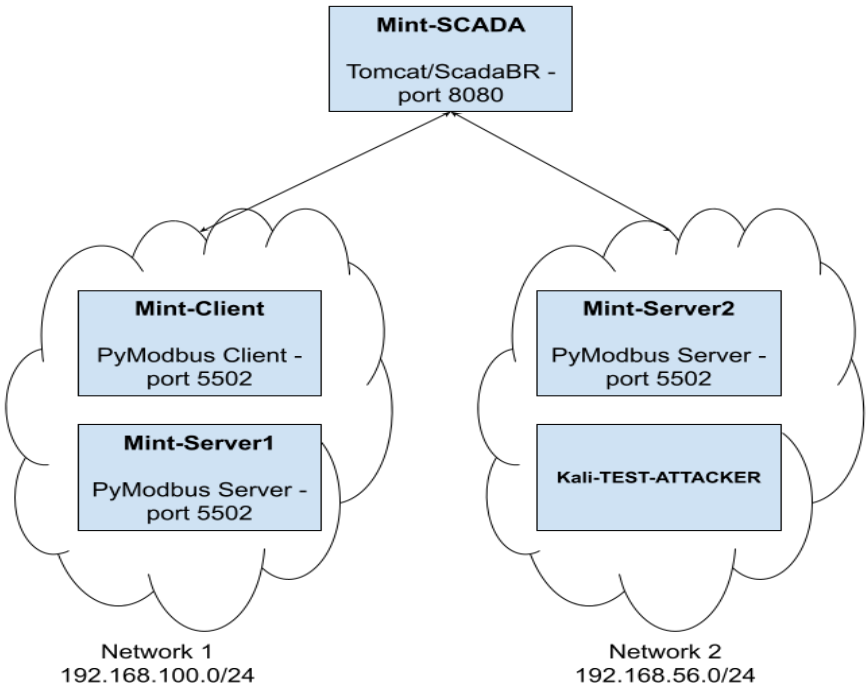

Figure 1: Testbed Network Topology

Guide to Configuring VirtualBox

1. Install the latest version of Oracle VM VirtualBox from the VirtualBox Downloads page (https://www.virtualbox.org/wiki/Downloads).
2. Install the latest versions of Linux Mint Xfce Edition (https://www.linuxmint.com/download.php) and Kali Linux (https://www.kali.org/get-kali/#kali-installer-images) in ISO format.
3. Within VirtualBox, select **Tools > Network** to view your list of host-only networks. Create two new host-only networks. For each network:
    a. Under the **Adapter** tab, select "Configure Adapter Automatically".
    b. Under the **DHCP Server** tab, select "Enable Server".

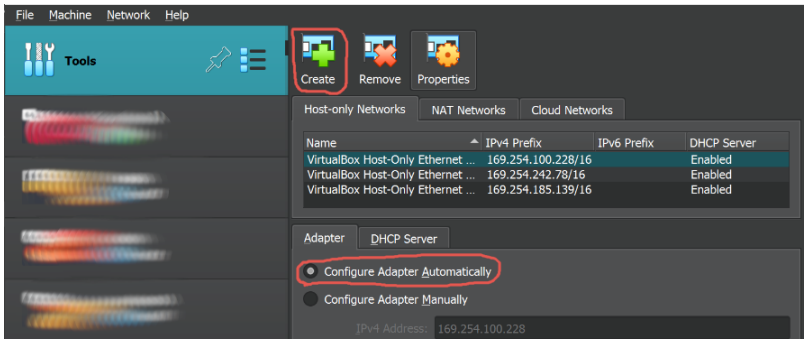

Figure 2: Create Host-Only Adapter

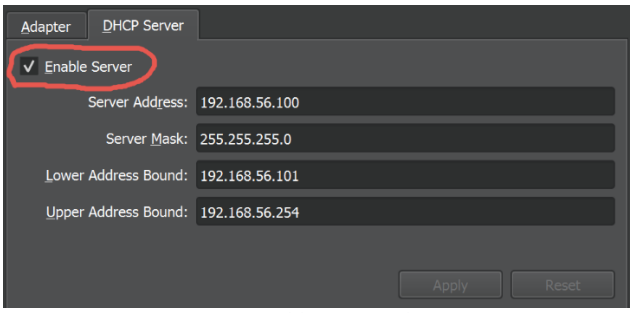

Figure 3: Enable Host-Only Server

Guide to Configuring Virtual Machines

1. Under the **Machine** tab at the top of the VirtualBox window, select "New".
2. Name your virtual machine (VM), then click on the **ISO Image** field and select the appropriate path to your Linux Mint ISO file. Then click "Next" to proceed.
3. On the **Memory** page, allocate at least 2 GB (2000MB) of Base Memory and 2 Processors to your new VM. Then click "Next" to proceed.
4. On the next page, select "Create a Virtual Hard Disk Now" and allocate ~25 GB VM storage.
5. Proceed through the next pages and click "Finish" to finalize the creation of a new VM.
6. In the new Mint machine, follow OS installation steps. Once complete, power off the VM.
7. In VirtualBox, select your new VM and navigate to **Settings > Storage**. Under the **Controller: IDE** heading, remove the optical drive.
8. Navigate to the **Network** page at the left side of the tab and select "Host-only Adapter" under the **Attached to** option. Select the first host-only adapter you created under the **Name** option.
9. Boot up your new Mint VM to ensure proper installation.

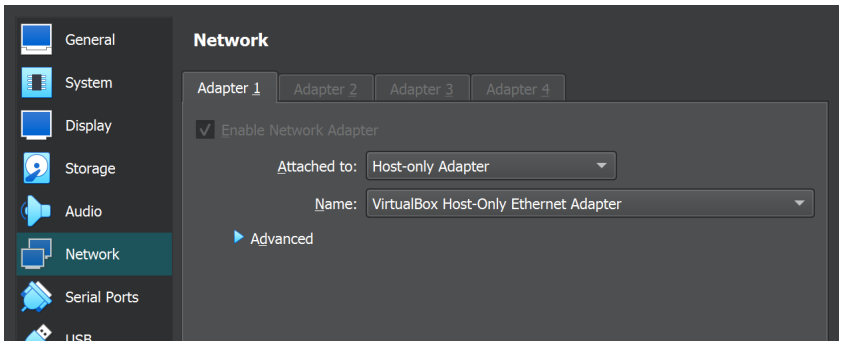

Figure 4: Connect VM to Network Adapter

10. Repeat the above process for devices desired on the same network, such as a client machine.
11. To create a device on a different network, change the Host-only Adapter being used under the **Network** tab in Settings.
12. To create a Kali attack machine, follow the instructions for Linux Mint device creation and allocate 4 processors to the Kali VM.

Establishing Communication Between Networks

1. Follow the steps outlined above for the creation of a Linux Mint machine.
2. Under the new VM's **Network** options, select the first host-only adapter on the **Adapter 1** tab. Then switch to the **Adapter 2** tab and select the second host-only adapter. This will connect the router to both network adapters simultaneously.
3. In the router's Mint terminal, enter the command *sudo echo 1 > /proc/sys/net/ipv4/ip_forward* to enable IP forwarding.
4. When the second network must be accessed from a device on the first network – or vice versa – enter the command *sudo ip route add* NET_IP *via* ROUTER_IP *dev* NET_DEV.
    a. Replace NET_IP with the target network's IPv4 address, e.g., 192.168.100.0/24.
    b. Replace ROUTER_IP with the router device's IP address on the current network, e.g., 192.168.56.105
    c. Replace NET_DEV with the current network adapter's name, e.g., enp0s3.

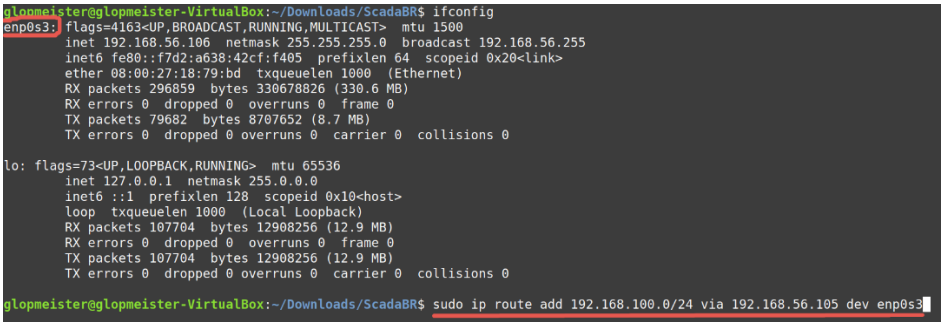

Figure 5: Manual IP Routing

5. It may also be necessary to adjust all involved systems' firewall settings to allow for transmission over a specific port.
6. Communication is now enabled between network adapters and utilizes the router device.

Configuring ScadaBR

1. Select a VM to host the ScadaBR web application. Router device hosts ScadaBR in this testbed.
2. On the chosen VM, install ScadaBR from SourceForge at https://sourceforge.net/projects/scadabr/files/latest/download.
3. Navigate to the **Downloads** folder and run the command *unzip ScadaBR_Setup_Linux.zip* to extract all ScadaBR files.
4. Run the command *sudo ./install_scadabr.sh* to fully install ScadaBR. During installation:
    a. Select a port on the machine for Tomcat to be run on. The default port is 8080.
    b. Create a user account to be used for Tomcat login.
    c. Launch ScadaBR.
5. Access ScadaBR from a web browser at *localhost:8080/ScadaBR*
6. To create a Modbus data source:
    a. Select the **Data Sources** button in the top navigation pane.
    b. Change the data source type to Modbus IP and click the "Add" button.
    c. Enter a Name for your Modbus server device, "Update Period" for more frequent data, and an acceptable IP address and port in the "Host" and "Port" fields.
    d. Under **Modbus node scan**, click "Scan for Nodes" to see available nodes on the server device. The value of these nodes can be investigated in the **Modbus read data** pane.
    e. Enter a Modbus address to gather data from under **Point locator test**, then click "Add point" to save this point for later viewing.
    f. Save the data source in the top right pane.
7. Navigate to the **Watch List** in the top navigation pane to view all data sources and points.
8. Any updates to your Modbus server will be reflected in the **Watch List**.

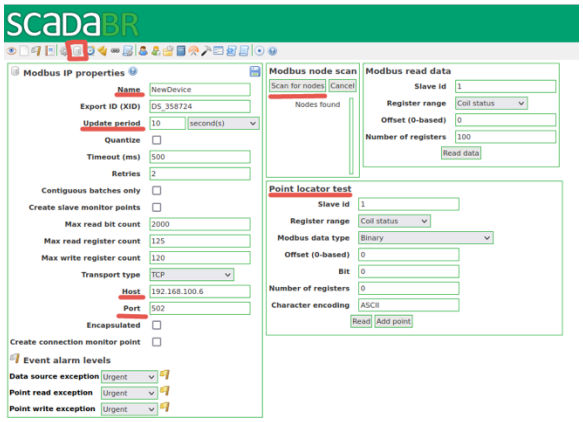

Figure 6: ScadaBR Data Source Page

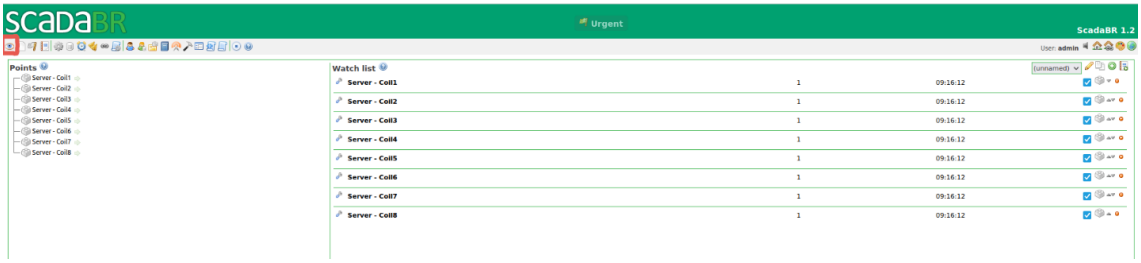

Figure 7: ScadaBR Watch List Page

3.2 Create Test Attack Software

Now that a testbed has been created, it can be adapted and utilized to examine many areas of security concern within a system. An organization may wish to test a full-fledged ransomware infection chain or a single step, and either can be handled by this system. The custom test software for this report was developed in Python and used in conjunction with existing tools from the Kali Linux and OpenPLC suites. Within the Python programs, the open-source library PyModbus was used to implement Modbus TCP/IP communications. Modbus TCP/IP is a common protocol in ICS systems, and its use on this system allows for testing of vulnerabilities in an ICS context. In Section 4, the specifics of each attack scenario are listed along with the software required for each.

This software is accessible via GitHub at https://github.com/im-jking/TestbedExploits.git and can be adapted quickly to most test scenarios. Each program is also thoroughly commented for easy understanding of functionality.

```
TestbedExploits
├── README.md
└── Scripts
    ├── BasicRansom
    │   ├── BadStuff
    │   │   ├── comboClient.py
    │   │   └── encryptor.py
    │   └── comboServer.py
    └── pyModbus
        ├── Basics
        │   ├── testClient.py
        │   └── testServer.py
        ├── actuatorState
        │   ├── actuatorClient.py
        │   ├── actuatorServer.py
        │   └── inject.filter
        └── setpointMod
            ├── bandMonitor.py
            ├── setpointAttack.py
            ├── setpointClient.py
            └── setpointServer.py
```

```
TestbedExploits
└── VBoxShare
    ├── build
    │   └── comboClient
    │       ├── Analysis-00.toc
    │       ├── EXE-00.toc
    │       ├── PKG-00.toc
    │       ├── PYZ-00.pyz
    │       ├── PYZ-00.toc
    │       ├── base_library.zip
    │       ├── comboClient.exe.manifest
    │       ├── comboClient.pkg
    │       ├── localpycs
    │       │   ├── pyimod01_archive.pyc
    │       │   ├── pyimod02_importers.pyc
    │       │   ├── pyimod03_ctypes.pyc
    │       │   ├── pyimod04_pywin32.pyc
    │       │   └── struct.pyc
    │       ├── warn-comboClient.txt
    │       └── xref-comboClient.html
    └── dist
        └── comboClient.exe

13 directories, 29 files
```

4. Ransomware Evaluation Methods

Seeking to elucidate and strategically counteract ransomware and various cyber-attacks, it is imperative to establish a robust and coherent framework for attack analysis. Over the years, cybersecurity research has intensively explored this domain, evolving significantly with notable advancements concentrated on risk analysis anchored in multifaceted social, technological, and situational models (Pipyros et al., 2018). To illustrate:

- **Social analysis models** prioritize minimizing risk by proactively identifying and mitigating social factors that pose potential threats. For example, insider threats, which are often implicated in significant security breaches, can be mitigated by leveraging social analysis to identify non-trustworthy individuals, thereby alleviating the risk of behavioral challenges translating into security incidents.
- **Situational analysis models** embody a comprehensive, multilevel approach to threat analysis. These models, by scrutinizing the entirety of an organization, integrate diverse factors into a singular model—incorporating legal, organizational, and technological aspects, amongst others. For instance, in situations where legal compliances such as GDPR are vital, the model considers potential threats to data privacy and regulatory adherence.
- **Technological analysis models** delve into the meticulous examination of the technical mechanics of threats and malware. Understanding an attack's modus operandi empowers security experts to deploy specific preventative measures. For example, by dissecting the technical operation of a ransomware like WannaCry, experts can discern its propagation vectors and formulate countermeasures to impede its spread.

In the context of our testbed's development, no experimental social facet has been incorporated. The evaluation methods employed are inclined predominantly towards either technological or situational approaches. Given that our employed model does not amalgamate sufficient organizational data to qualify as a situational model, it more precisely aligns with being a technological model. Consequently, our evaluation model zeroes in on scrutinizing the technical ramifications exerted by each ransomware variant on the target system. The construction of a technical evaluation model within our framework has accentuated three pivotal attack attributes for systematic enumeration: visibility, likelihood, and consequence, each playing a critical role in deciphering the potential impact and mitigation of a cyber-attack.

Attack Visibility: Anchored in the discernibility of attack indicators during various attack phases, attack visibility is pivotal for comprehensive cybersecurity analysis. During each step of the attack chain, analysts should meticulously evaluate network transmissions as a starting point. For instance, are there discernible attempts to access blocked ports or systems within the network? Can anomalous communications between systems be detected? Utilizing a network monitoring tool, such as Wireshark within the testbed, can shed light on network interactions and potential vulnerabilities. Subsequently, it's crucial to scrutinize visible impacts on availability or integrity. Are there interruptions or alterations in data access by users or applications? Has any system downtime been observed? By evaluating whether

each observable effect might be indicative of a security event and segmenting the attack stepwise, security administrators can pinpoint anomalies, decipher adversarial actions on targets, and methodically analyze them. Organizing effects by attack phase can also enable analysts to determinatively correlate one effect to another, viewing each interrelated effect as a fragment of the overarching security event.

Attack Likelihood: Quantifying the probability of an attack's occurrence, attack likelihood can be gauged through a plethora of methodologies, ranging from subjective analyst evaluations to comprehensive likelihood assessments tailored to specific scenarios. An invaluable practice in contemplating likelihood involves critically assessing the assumptions embedded within a test attack. For instance, if a test attack presupposes that an attacker has executed several preventable steps, its likelihood might be deemed lower compared to a test attack devoid of prerequisite assumptions. It's crucial to reconcile the hypothetical assumptions with realistic threat vectors and attacker capabilities to generate an authentic representation of potential risk.

Consequence Assessment: Tailing these considerations, the concept of 'consequence' emerges as the final critical criteria for testing and assessing attack vectors. Consequence, an expansive consideration for cybersecurity experts, can be dissected from myriad perspectives. For the objectives of this study, the MITRE ICS ATT&CK Matrix Impact column furnishes an effective framework, enumerating potential attack repercussions on ICS systems. By exploring which system attributes are influenced by an attack, a nuanced consequence assessment can be orchestrated. This should encapsulate the possible impacts of each test attack, scrutinizing aspects such as the connectivity of the network to critical or sensitive systems and positing the tangible operational ramifications should such an attack transpire against the system in real-world contexts. Understanding the tangible and intangible impacts of an attack vector is crucial in categorizing and prioritizing threat mitigation strategies.

The triad of Visibility, Likelihood, and Consequence (VLC), while inherently subjective and qualifiable, can be systematically assessed using a quantifiable approach, particularly by employing a 1-5 scale, enhancing both cohesiveness and comparability amidst diverse attack scenarios. A structured template, denoted as the VLC Table and illustrated in Figure 8, echoes a methodology akin to the one utilized for character profiles delineated in the ensuing section. This model doesn't endeavor to render a definitive evaluation of the attack format under scrutiny. Instead, it aims to showcase the utility of the testbed in amalgamating a comprehensible perspective on specified attack attributes, offering a pragmatic lens through which varying cyber assault paradigms may be comparatively analyzed.

Visibility					
Likelihood					
Consequence					

Figure 8: VLC Table Template

5. Scenarios

In this section, three ransomware attack scenarios with differing Tactics, Techniques, and Procedures (TTPs) will be executed on the virtual testbed and evaluated to track observable effects and potential Indicators of Compromise. Each scenario begins with a brief fictional overview alongside character profiles that allow for a comparison in TTPs between the scenarios. These character profiles will contain several attributes, each graded on a subjective 1-5 scale for quick visual comparison. Attacker profiles visualize 3 attributes:

- **Access –** How close to accessing the target system is the attacker, physically or digitally? An attacker who has no relation to the victim network may be rated a 1 on this scale, while an insider with regular unimpeded access to the victim network may be rated a 5.
- **Resources –** How high is the attacker's funding, staffing, and technology access? A solo attacker using commercially available machinery may be rated a 1 on this scale, while a state-funded team of attackers using proprietary machinery may be rated a 5.
- **Experience –** How familiar is the attacker with the victim's system and its services? A hobbyist hacker with little awareness of ICS network layouts may be rated a 1 on this scale, while an employee with specialized training and firsthand experience with the system may be rated a 5.

Addressing these attributes allows for a more complete understanding of an attacker's capabilities on a social, organizational, and technical level. The template used for the Attacker Profile Table is shown in Figure 9: Attacker Profile Table Template.

Access					
Resources					
Experience					

Figure 9: Attacker Profile Table Template

Victim profiles also visualize 3 attributes:

- **Sensitivity –** How severe could the repercussions to outsider access to the victim's network be? Does the network hold classified information? Could a system breach cause hardware/ICS damage? A network that contains only publicly published scientific papers may be rated a 1 on this scale, while an ICS network that contains classified data from and directly controls reactor operations may be rated a 5.
- **Exposure –** How easily accessible is this network to the public? An air-gapped and isolated private network may be rated a 1 on this scale, while a completely public network may be rated a 5.
- **MITRE ICS Impact Codes** – Allow for the direct analysis and comparison of impacts based upon attack TTPs. Impact codes will be written rather than graded on the 1-5 scale.

Addressing these attributes allows for quick understanding of the given scenario and assumptions made in the deployment of the test exploit. The template used for the Victim Profile Table is shown in Figure 10.

Sensitivity					
Exposure					
Impact Codes					

Figure 10: Victim Profile Table Template

Each attack scenario is described as a sequence of attack steps. The procedure for simulating these steps on the virtual testbed is defined in detail, as well as the evaluation steps that will be executed during and following each attack procedure. The Visibility, Likelihood, and Consequence of each attack scenario is analyzed step by step to build a VLC Table for each scenario, then defense and mitigation recommendations based upon the attack scenario are given for several organizational levels.

5.1 HMI Encryption and Exfiltration

Overview

This scenario follows the deployment of a remote-access ransomware used for encryption and exfiltration on a Linux machine, which represents an HMI in an ICS context.

Character profiles

Attacker – Ransomware gang with limited ICS experience.

Access					
Resources					
Experience					

Figure 11: Scenario 1 Attacker Profile Table

Victim – Windows HMI connected to ICS system.

Sensitivity					
Exposure					
Impact Codes	Loss of Productivity and Revenue (T0828), Theft of Operational Information (T0882)				

Figure 12: Scenario 1 Victim Profile Table

Background/Event

Jordan is a systems operator who works at the Townsville Nuclear Power Plant (TNPP). Jordan recently received an email from a similar but unaffiliated nuclear site across the country warning of CVE-2023-3392, a vulnerability affecting the RTU used at TNPP. The email directs Jordan to the company's website, where he downloads and runs an updater on the HMI system. Unfortunately, the update website has been compromised by an attacker. Upon opening the file, Jordan's HMI is infected with a piece of ransomware that has encrypted all the system's data. The attackers demand 30 BTC in exchange for system decryption and assurances against data leaks.

Attack Sequence

1. The attacker compromises an organization's firmware update page and replaces authentic update files with malicious reverse shell files.
2. The attacker turns on their reverse shell Client and waits for a victim to connect.
3. Jordan downloads and runs "Firmware Updater", which is the attacker's reverse shell program.
4. The attacker exfiltrates and encrypts classified data and poses a ransom to decrypt the system and prevent data sharing.

Experimental Format

- Goal: Determine potential warning signs for remote access ransomware in order to better prepare system defenses and security experts.
- Software Configuration:
    - This scenario utilizes two machines: Mint-Server2 and Kali-TEST-ATTACKER. The former acts as a victim machine, with the assumption that a phishing attempt has successfully led to the installation of test software on the system. The latter acts as an attacker, which is listening for the victim to connect to its server program.
    - A malicious payload was constructed containing a Python-based client program and encryptor. These programs are compiled to the target OS using PyInstaller and placed on the victim machine.
    - The customer runs the client program, which automatically connects to the server program and runs in the background. The attacker now has remote access, exfiltration capabilities, and an encryptor program to use as they please.
- Procedure:
    - On the attacker machine, the program *basicServer.py* was installed and run to begin listening for connections.
    - On the victim machine, the "payload" folder was installed, containing programs *basicClient* and *encryptor*. *basicClient* was run to connect to the attacker device.
    - From the attacker, the command *transfer encryptor* was used to observe the effects.
    - From the attacker, *encryptor* was moved to the "~/home" folder and executed to encrypt the victim's user folders.
        - A broader test attack may be performed using *sudo* permissions if analysts assume some level of privilege escalation on the attacker's part.
    - A mock ransom note, *ransom.txt* was created on the victim's machine.
    - Each encrypted folder is decrypted using the *encryptor* program.
    - The *exit* command is entered to shut down the reverse shell.
- Evaluation Steps for Each Phase:
    - Investigate Wireshark pane for any anomalous network activity.
    - Investigate user folders for any visible changes to file information.
    - Attempt data access and editing on several files throughout the user folders.
    - Note any assumptions made in reaching this phase.
    - Identify newly accessed or exposed systems.

Experimental Findings*:*

In order to best evaluate the Visibility, Likelihood, and Consequence (VLC) of an attack of this nature, the execution procedure is followed in order and segmented by attack phase. Following the Cyber Attack Lifecycle, the first phase explored in this experiment is the Delivery Phase. This first phase involves connection of the victim device to the remote server that acts as a reverse shell. Immediately following the Delivery Phase are the Exploit and Control Phases, during which an attacker may navigate the victim's network freely in order to escalate privileges and potentially gain access to new machines in anticipation for the final Execution Phase. At this point, the ransomware is fully deployed for effect on the victim machine. In this scenario, no Maintenance Phase is pursued following the Execution Phase.

The Delivery Phase is of relatively low visibility in that all communication happens over the network with no apparent effects on the physical machine or any data therein. A network monitoring tool such as Wireshark or an IDS/IPS will likely be able to help identify the three-way handshake that allows for TCP connection between devices, but lapses in security policy or failure to watch for warning signs may allow for unmonitored connections. Likelihood of this phase's occurrence is high, as successful phishing attacks represent the most common initial access vector for cyber-attacks. Additionally, navigation to more sensitive systems on the victim is possible once initial access has been gained. Consequence during this phase is present only in the potential of a future attack as the initial payload offers no inherent damage.

Moving into the Exploit and Control Phases visibility remains nearly constant; the only change in this area is the potential detection of encryptor program relocation from the initial payload folder. Consequence during this phase depends entirely upon the attacker's goals. If the attacker is able to escalate user privileges and encrypt large swaths of data, Consequence increases accordingly. If sufficient controls are placed on lateral movement methods, system Consequences can be effectively contained to the initially infected machine.

Finally, the Execution Phase leads to a very sharp increase in visibility and consequence. The encryption of user data will quickly become apparent either through a ransom note or the hampered functionality of local programs and connected systems. Exfiltration efforts can be noticed through immense TCP interactions on network monitoring systems. Additionally, any systems to which the attacker has gained access may be completely shut down for impact using the encryptor program. Consequence is not adjusted during this phase, as any affected systems that will be accessed likely already have been accessed. This being said, there exists a high possibility of unnoticed infection on machines that the attacker decided not to directly impact.

This scenario shows a very basic form of ransomware, but execution of the full test procedure can help security professionals better understand the common signs of ransomware infection. If left unchecked an infection of this variety may lead to immense operational, strategic, and regulatory damages to any organization. The overall visibility is low, and experience attackers may be able to affect an expansive network of systems. Figure 13 shows the VLC rating for this attack scenario.

Visibility					
Likelihood					
Consequence					

Figure 13: Scenario 1 VLC Table

Knowledge, Skills, and Attitudes*:*

To best address information security concerns, it is essential to involve personnel from every layer of the security process. At the most minute level, cyber defense teams and security professionals must directly address the technology used in the attack and its immediate operational impacts. OT personnel must be kept aware of necessary updates and adjustments to prevent damage to their devices. Regulators must moderate cyber response patterns at the highest level to maintain orderly and just incident response. Each of these positions offers an essential service to the security community and can help to mitigate the effects of cyber incidents. The mitigation techniques below are drawn from the MITRE ICS ATT&CK Mitigations table, each accompanied by the correlating Mitigation code.

In their role as system administrators and monitors, cyber defense teams have the most granular control over potential mitigations within an organization. In this scenario controls such as Access Management (M0801), Multi-Factor Authentication (M0932), or Account Use Policies (M0936) may be implemented to prevent later movement via unauthorized access to valid accounts. Exfiltration efforts may be hampered using Limit Access to Resource Over Network (M0935), while data itself can be protected using Encrypt Sensitive Information (M0941) and Data Backup (M0953). The latter method is key in mitigating the impacts of an encryption attack; backups should be kept separate from the corporate network and configured for quick recovery.

OT personnel should focus on preventing the physical and operational impacts associated with this kind of threat. Watchdog Timers (M0815) may be instated on field devices and HMIs in order to immediately detect system interruptions caused by program encryption. Install Safety Instrumented Systems (M0812) in the field to automatically detect disruptions and adjust to the condition. Such systems should always be segmented from operational networks in order to protect them from additional targeting. When installing new machinery, always ensure Boot Integrity (M0946) by following secure, recommended boot methods.

Regulatory activity may take place at the organizational or government level and is vital in holding personnel accountable for their maintenance of and activity on an ICS system. Organizations may hold a regular Audit (M0947) of major systems to ensure that no potential threat actors have gained access. Network Segmentation (M0930) is key in preventing the spread of remote access between machines, and can be aided with User Account Management (M0918) that reduces the number of traversable, out-of-date accounts on the network. On a broad level, Password Policies (M0927) should be implemented to prevent brute-forced account access and to limit potential for lateral movement within a network.

5.2 Altered Actuator State (MSCI)

Overview

This exploit utilizes Modbus vulnerabilities to inject a command that changes the state of an actuator in a system, such as turning a controller on or off. This is an implementation of Malicious State Command Injection, or MSCI.

Character Profiles

Attacker – A state actor made up of cybersecurity experts.

Access					
Resources					
Experience					

Figure 14: Scenario 2 Attacker Profile Table

Victim – A private, local government-operated ICS network.

Sensitivity					
Exposure					
Impact Codes	Manipulation of Control (T0831), Loss of Safety (T0880), Loss of Protection (T0837), Denial of Control (T0813)				

Figure 15: Scenario 2 Victim Profile Table

Background/Event

The aggressive but unidentified state actor TheftMen have gained access to a water control system in Philadelphia through advanced social engineering and malware tactics. An attack device, functioning as a man-in-the-middle, has been connected to the system and is used to deactivate vital devices in the ICS network. The attackers refuse to release their hold on the water system unless the local government declassifies top secret information related to a recent visit from a top foreign government official.

Attack Sequence

1. The TheftMen gained access to a water control network through unknown and unmonitored social engineering and malware attacks.
2. The attackers initiate a man-in-the-middle attack on a network that houses control systems for the operation of water deployment across the city.
3. The attack deactivates vital machinery and denies data access to operators at the site.
4. The attackers demand the release of top secret information from the city to stop the attack.

Experiment Format:

- Goal: Determine potential warning signs for man-in-the-middle attacks in order to better prepare system defenses and security experts.
- Software Configuration:

- This scenario utilizes three machines: Mint-SCADA, Mint-Server2, and Kali-TEST-ATTACKER. The former two act as the client and server in an ICS network, the client prompting the server for coil data every 10 seconds. The latter acts as a man-in-the-middle (MitM) attacker which alters network packets as they flow between the client and the server.
- On the Mint-SCADA (client) machine, the open-source web app ScadaBR is being used to simulate a SCADA system. This application is configured as follows:
  - One data source is created to query the server program (192.168.56.106:5502) every ten seconds.
  - Eight setpoints are created that display the status of eight coils on the server.
- All eight coils on the Mint-Server2 (server) machine are initially set to True, representing eight functioning ICS devices.
- The MitM attack modifies data flowing between the client and server to set all coils on the server to False, representing the deactivation of ICS devices. Any query from the client after this point will erroneously show that the coils are all still set to True.

- Procedure:
  - On the server machine, the program *actuatorServer.py* is installed and run to begin listening for connections.
  - On the client machine, the web application ScadaBR is configured and run to display data gathered from the server.
  - On the attacker machine, the program *Ettercap* is used to begin an ARP poisoning attack against the client and server devices. All traffic between the two devices now runs through the attacker machine, and a custom filter begins falsifying data in any Modbus packets transferred between the victim machines.
  - After observing effects at this stage, the attack program is killed and the ICS devices return to direct communication.
- Evaluation Steps for Each Phase:
  - Investigate Wireshark panes on the server and client machines for any anomalous network activity.
  - Observe ICS activity via ScadaBR on the client machine.
  - Note any assumptions made in reaching this phase.
  - Identify newly accessed or exposed systems.

Experiment Findings:

Since the attacker's presence on the victim network is assumed in this scenario, no Delivery phase can be assessed from the exploit. The Exploitation and Control phases are represented through any tactics used by the attackers to gain and solidify this network presence. The Execution phase is represented by the activation of the MitM attack, which is not supplemented with any other attack procedures. As a result, this scenario directly represents only one phase of the Cyber Attack Lifecycle. The flexibility of this testbed design allows for the evaluation of exploit concerns at any level and allows for a better understanding of individual attack steps alongside complete infection chains.

This scenario makes some assumptions leading up to the Execution phase. First, the attackers must first gain initial access to the victim's ICS network. This may be accomplished through exploitation of network security flaws, social engineering and credential theft, or direct hardware access/infiltration. The visibility and likelihood of any one of these access vectors is unique to each organization and should be specifically evaluated. In a legitimate attack scenario, numerous controls can be placed on initial access and social engineering efforts that may significantly reduce the likelihood of a MitM attack occurring.

The Execution phase of this attack brings only a minor change in visibility and likelihood over any preceding steps in the Attack Lifecycle. ARP routing of traffic through the attacker device is visible on Wireshark alongside a renewed TCP three-way handshake to reestablish client/server connections. A TCP disconnect exchange is visible when the attack ceases. No file changes are visible on either victim machine, and the client program continues to receive safe coil data. While no change is visible on the server machine in the current configuration, loss of functionality would be clear in a physical ICS context. Though visibility is low in this phase, consequence is extremely high. This attack may shut off valves, alarm systems, or entire industrial machines that are vital to safe functionality. Deactivation without discretion may result in severe damage to the ICS network as well as significant danger to any personnel near unmonitored ICS devices. Restoration of these systems may be time-consuming and costly, pressuring the organization to accept the ransomware terms as soon as possible.

Though this attack phase is predicated on a long line of assumptions, it is important to analyze even the most unlikely of attacks. The visibility of this phase is relatively low, with only minor network impacts being visible. The likelihood is also low due to the assumptions made. Consequences of an attack of this nature are immense, particularly in an ICS context. Figure 16 shows the VLC rating for this attack scenario.

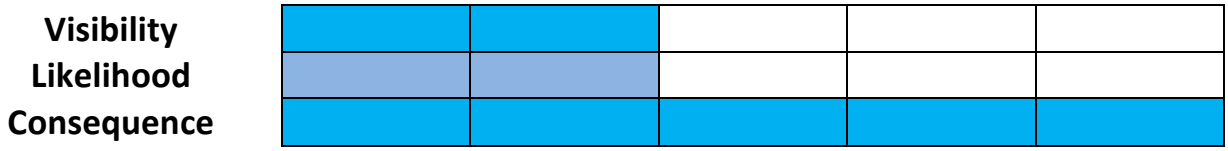

Figure 16: Scenario 2 VLC Table

Knowledge, Skills, and Attitudes*:*

It is often easy to dismiss scenarios of such immense consequence due to their perceived unlikelihood when compared to other attack cases. Though it is important to consider the likelihood of every attack scenario, security personnel of all levels should be fully aware of warning signs related to high-consequence attacks. Simulating portions of these attacks allows for a better understanding of each step and therefore a greater preparedness against the attack as a whole.

Cyber defense teams can effectively reduce the risk of man-in-the-middle attacks via several mitigation efforts. Communication Authenticity (M0802) is the clearest method for weakening data interception, but the inherent weaknesses of Modbus as an unencrypted protocol make this difficult. Exploit Protection (M0950) may be effective when used to prevent ARP poisoning, though this is a broad mitigation. Defenders may be able to Filter Network Traffic (M0937) to the same end, but this may be difficult to implement while still allowing for valid ARP queries and Modbus TCP/IP communications. Network Allowlists (M0807) should be implemented to only permit known MAC addresses on the network, in effect preventing ARP poisoning from an external device.

OT personnel may implement Mechanical Protection Layers (M0805) to prevent damage to physical systems even during dramatic shutdowns. It may be useful to use an Out-of-Bound Communications Channel (M0810) to ensure access to server data even during an MitM attack or other denial-based scenarios. Redundancy of Service (M0811) will have a similar effect, keeping systems running or stable in emergency situations. This may be difficult when the servers cannot distinguish the malicious signal they are receiving from a legitimate request. Perhaps the most direct preventative measure against MitM attacks is the use of Static Network Configuration (M0814), which restricts the use of ARP and other dynamic network protocols that are used in network poisoning attacks.

Regulatory actions should focus largely on the prevention of MitM attacks through network configuration and defense requirements. Regular Audits (M0947) can help ensure that no insecure communication protocols are in use among ICS devices, thus preventing data spoofing and state injections. The reduction of weak protocols can be supplemented through Operational Information Confidentiality (M0809) in order to prevent external personnel from gaining an understanding of network communications within the ICS network. Regulators can Limit Hardware Installation (M0934) to prevent the introduction of unauthorized devices to the ICS network, removing at least one initial access vector. This mitigation is aided by Supply Chain Management (M0817), which can prevent compromised devices from being used on a network by ensuring that all new devices are trusted and tested before implementation.

The most effective mitigations against an advanced MitM attack are Network Allowlists, Static Network Configuration, and Auditing. The implementation of these mitigations at every level of an organizational structure can significantly reduce the viability of ARP poisoning and packet injection onto ICS networks. To better understand the utility of each mitigation, this scenario may be expanded to include Delivery, Exploit, and Control phases that lead up to the eventual MitM attack execution. It is possible to combine this attack with *Scenario 4.1* in order to better simulate a full attack chain.

5.3 Altered Control Setpoint (MPCI)

Overview

This exploit changes one or more setpoints in a system to change system behavior. In this scenario, the setpoint for triggering a system alarm is modified to allow unauthorized persistence on an ICS system. This is an implementation of Malicious Parameter Command Injection, or MPCI.

Character profiles

Attacker – A low-level insider threat who is acting alone.

Access					
Resources					
Experience					

Figure 17: Scenario 3 Attacker Profile Table

Victim

Sensitivity					
Exposure					
Impact Codes	Loss of Safety (T0880), Theft of Operational Information (T0882), Loss of Protection (T0837), Loss of Productivity and Revenue (T0828), Damage to Property (T0879)				

Figure 18: Scenario 3 Victim Profile Table

Background/Event

Sarah, who is a low-level employee at the Simulacrum National Laboratory, decided to infect her company system with a script that alters the control setpoint of the digital alarm system, allowing her to move outstation data without notifying anyone. The malware moves laterally and stays on the network, observing normal operations while quietly saving data to a hidden container on her machine.
After 6 months of observation and data collection, Sarah plugs in a thumb drive that infects the network and wipes the data from the impacted machines and saves the data to her removable thumb drive. She then flees to Cuba – where she knows she will not be extradited to the United States – and holds the stolen data for ransom while threatening to leak it to the highest bidder if the ransom is not paid.

Attack Sequence

1. Sarah creates a script to alter the digital setpoint of the device's alarm system.
2. Sarah accesses the work site through authorized means with a hidden thumb drive.
3. Sarah inserts the drive into an HMI, installing her malware on the machine.
4. The malware observes and saves regular operational data in a hidden container.
5. When observation is finished, Sarah plugs the thumb drive in to download the collected data.
6. Sarah flees to Cuba in order to avoid extradition and offers to sell the data back for a ransom or leak the data to the highest bidder if no ransom is paid.

Experiment Format

- Goal: Determine potential warning signs for physical device access and Malicious Parameter Command Injection (MPCI) attacks to better prepare system defenses and security experts.
- Software Configuration:
  - This scenario uses two machines – Mint-Client and Mint-Server1 – as well as a thumb drive containing malicious code.
  - The Mint machines use Python scripts (and the PyModbus module) to simulate a regularly communicating ICS system.
    - The client reaches out to the server every 5 seconds for register data, which is printed to the client machine's console. If the server's alarm coil is True, a warning message is also printed to the console.
    - The server maintains a static set of register data; holding register 10 contains the maximum KB transmission over the course of 10 transmissions, while coil 0 contains a Boolean stating whether the alarm condition has been triggered.
    - The server device runs a bandwidth tracker, which runs every 10 seconds and checks the amount of data sent from the server during that cycle. If this exceeds the bandwidth limit (register 10) the alarm condition is set to True.
  - The thumb drive is used to upload a malicious program which modifies the alarm setpoint and tracks communications.
    - Setpoint is changed to 50, higher than expected during constant tracking.
    - Every second, the server register contents are written to a fake system log file.
    - Once the attack is complete, the log can be exfiltrated onto the thumb drive.
- Procedure:
  - On the server machine, the programs *setpointServer.py* and *bandMonitor.py* were installed and run to begin listening for connections and monitoring network activity.
  - On the client machine, the *setpointClient.py* program was installed and run, connecting to the server machine.
  - Regular network operations were observed.
  - A USB thumb drive was mounted to the client machine and the *setpointAttack* program was transferred onto the machine.
  - *setpointAttack* was run from File Explorer on the client machine, beginning the exploit.
  - The *bandMonitor* window was observed for network activity.
  - After running the attack for 10-20 seconds, the *testSyslog* file created by the attack program was transferred to the USB drive.
  - The USB drive was unplugged, and all programs shut down.
- Evaluation Steps for Each Phase:
  - Run *iostat* command to check for changes in data written to/from devices.
  - Investigate Wireshark pane for any anomalous network activity.
  - Investigate user folders for any visible changes to file information.
  - Note any assumptions made in reaching this phase.
  - Identify newly accessed or exposed systems.

Experiment Findings:

This experiment displays the Delivery, Exploitation, Control, and Execution phases of the scenario's Attack Lifecycle. The Delivery phase of this experiment follows the connection of the USB thumb drive and transferal of the malicious program onto the server machine. The Exploitation phase involves the execution of the *setpointAttack* program on the server, starting the collection of data from the server machine. This phase is closely tied to the Control phase, which in this scenario meant waiting for a large amount of register data to be written into the *testSyslog* file. During this phase, network and disk transmissions may also be read to better understand the communications occurring on the system. The final phase displayed in this experiment is the Execution phase, during which the *testSyslog* file is transferred from the server machine onto the USB thumb drive. The thumb drive is then removed, and once effects have been monitored the network is shut down.

The Delivery phase is the most visible of any phase in this attack scenario. During this phase the attacker must have physical access to the client machine and connect a USB thumb drive; this increases the likelihood of their activities being monitored by a physical security system such as a camera or card reader, as well as the chance of detection by digital systems on the client machine designed to prevent or log any physical device connections. Any users connected to the machine during the Delivery phase may see the USB mounted, and the appearance of a new executable on the client machine would be another warning sign. Few assumptions are involved in reaching this attack phase – primarily the attacker must have physical access to the client machine and this machine must have an accessible USB drive. The direct consequences of this phase are low, as no inherent damage is done to the system by addition of a USB drive and executable.

During the Exploit and Control phases physical and local visibility is reduced, while network visibility increases. The USB drive is removed and no longer shown on the device, though the executable transferred from it is still visible and running. The *testSyslog* file is created and visible, though it can be disguised as a system log and hidden to reduce visibility to personnel. Additionally, the execution of the attack program produces two Malformed TCP Packets that are visible over Wireshark. The likelihood of executing this phase is not substantially different from the likelihood of executing the previous phase; the most obvious assumptions involved here are that the attacker knows the internal layout of the server machine and that the attacker can identify an unmonitored area on the system in which to store the log file. Similar to the Delivery step, no inherent consequence comes from the execution of the attack program. Data is stored in the log file and some extra bandwidth is taken up on the network, but these effects are largely negligible to most organizations.

The Execution phase introduces the stored consequence from earlier steps. Physical visibility is reintroduced when the USB drive is mounted to the client machine again, and the transfer of the log file onto the USB drive produces a trace of data written to the USB. When the attack program is killed, a TCP disconnection frame is visible over Wireshark. Network transmissions then continue as normal between the client and server. This attack phase assumes that the attacker is able to access the system physically a second time but involves no further system access or exposure.

Altogether, this experiment showed high physical and network visibility for the given attack scenario. The attacker must undertake an immense amount of risk in that they must gain physical access to the system, and the execution of an unknown executable on the client system can raise numerous alerts. The

transmission of Malformed Packets when the attack begins is a clear warning sign of command injection, but the intention of modifying alarm setpoints is to prevent any automated detection of high network activity. The influx of data written to disk may also act as a warning sign, but on large ICS networks this increase may go unnoticed. Figure 19shows the VLC rating for this attack scenario.

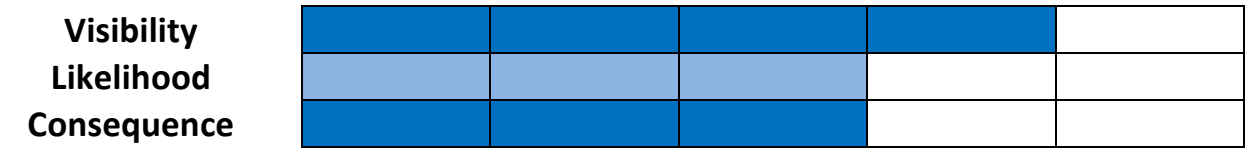

Figure 19: Scenario 3 VLC Table

Knowledge, Skills, and Attitudes*:*

In planning networks in an ICS context, it is all too common for security personnel to focus on perimeter defenses without considering the possibility of an attacker gaining physical access. In the 2010 Stuxnet attack, attackers were able to disrupt centrifuges by physically injecting a worm onto Iranian ICS networks. The use of an infected USB drive as the initial access vector allowed the attack to cross air gaps and directly infect the target system. As such, personnel from every organizational level should consider the physical aspect of network security.

Cyber defense teams must ensure that physical attacks are hampered by protecting networks from the inside. Execution Prevention (M0938) should be implemented to prevent attack binaries from being executed on the network, and systems should Filter Network Traffic (M0937) to detect anomalous behavior such as the Malformed TCP Packets that were observed in this experiment. It is also advisable to Restrict File and Directory Permissions (M0922) so that no unauthorized applications may be moved onto physical systems and running applications must be specifically allowed to create files.

OT personnel should Disable or Remove Features or Programs (M0942) from newly installed devices in order to prevent potential exploitation of physical vulnerabilities. They may also Limit Hardware Installation (M0934) in order to prevent the introduction of unknown drives and physical devices on the network, and prevent unauthorized device access through Mechanical Protection Layers (M0805). Implementing such protections will largely preclude physical threat actors from covert theft tactics.

Organization regulators and administrators can encourage awareness of physical threats through mandated User Training (M0917) and the creation of a Threat Intelligence Program (M0919). Potential insider threats can be mitigated by implementing Operational Information Confidentiality (M0809), which will prevent single employees from accessing specific information on ICS servers by excluding operators from specific register information. On a broader level, administrators should ensure that employees are held accountable for monitoring each other's work and ensuring no unauthorized activities are being performed.

The clearest mitigations for this exploit are Restrict File and Directory Permissions, Limit Hardware Installation, and User Training. By employing these mitigation tactics, organizations can significantly reduce the likelihood of an insider attack using physical media. Though such an attack can often be overlooked due to its relative rarity, security professionals must consider every attack family to truly protect their organization and its information.

6. Nuclear Community Best Practices

Though it is often easiest to summarize best practices in a few short phrases, the variety of effects and indicators present in each of the above scenarios shows that security practices must be evaluated based on individual attack sequences. Companies should focus on identifying areas of security concern within their organization and prepare accordingly. By establishing a virtual testbed, network security teams may individually investigate attack chains and better understand effective mitigation tactics. These mitigations may then be summarized and shared with management for deployment on an organizational level. Safe practices can be perpetuated by regulating a culture of security, educating employees on risks and mitigations, and continually building an understanding of existing and potential cyber threats through further testing.

In each Section 4 scenario evaluation, the roles of several organizational sectors in mitigating specific threats have been examined. These three sectors – cyber defense, OT, and regulation – are key to the maintenance and security of ICS networks. To promote security throughout every sector, it is vital that organizations recognize network security as a two-way street. Cyber defense teams evaluate network security to predict potential security flaws, which may be patched at the network level or may require company regulation to implement. Once regulators have been made aware of security flaws it is their responsibility to ensure that these vulnerabilities are minimized by whatever means are practical. Tactics such as Password Policies (M0927) and User Training (M0917) can be used to share safe practices with an entire organization, and compliance can be encouraged from the top down by superiors who enthusiastically participate in safe practices. Just as it is the network security team's responsibility to identify potential vulnerabilities, it is the management's responsibility to ensure that these vulnerabilities are protected against within every echelon of the organization.

Along the lines of employee training, there are many opportunities for security education on the company level. The Cybersecurity and Infrastructure Security Agency (CISA) offers ICS training and certification both via web-based and instructor-led classes. These courses range from basic introductions to ICS networks and controls to ICS cybersecurity and security evaluations, and all are presented free of charge. More ICS security trainings are available from organizations such as the SANS Institute and Dragos, and MITRE offers training on the effective use of the ATT&CK framework. The latter training is available free of charge and can help organizations standardize communications around cyber vulnerabilities and threat mitigation. In addition to public training, it is vital to standardize a set of training modules to prepare personnel for work on a specific organization's systems. In the creation of such a course, instructors should maintain Operation Information Confidentiality (M0809) by only sharing information that is pertinent to the trainee's position.

While it is key to build a company-wide understanding of known security vulnerabilities and mitigations, it is equally important to continue building an understanding of emerging threats. Security departments should keep up to date on recent attacks, recorded vulnerabilities, and security testing. Information surrounding these topics can be disseminated throughout an organization in the form of regular newsletters that highlight key info for the ICS industry. Many regular cybersecurity newsletters are available online, and security personnel may choose to receive any that they prefer or aggregate news from several sources. In order to predict future vulnerabilities, security teams should work with management personnel to perform regular penetration tests on their company's network. Potential

vulnerabilities can be evaluated my deploying custom exploits on a testbed; this also offers the advantage of revealing indicators of compromise and unexpected attack effects. To fully utilize vulnerability awareness, security teams should strive for adaptability to new information. Major security adjustments should be carefully considered, but it is important for organizational security to remain active and adjust to novel risks and mitigation tactics.

By cultivating a community of security-minded personnel it is possible to greatly reduce the digital security risk to an organization. Regulation, education, and forward-thinking research all play a direct role in maintaining network security and preventing ransomware attacks. This report outlines an effective method of uncovering and investigating threats in an ICS landscape, as well as providing a framework to allow for increased cohesion between threat profiles. Profiles developed through experimentation must work in conjunction with known threat data such as Common Vulnerabilities and Exploits (CVEs) and governmental recommendations such as those found in the CISA #StopRansomware Guide. CVEs can provide granular insight into past and current security flaws in ICS and related devices. The #StopRansomware Guide (CISA), updated in May 2023, lists and describes best practices for ransomware prevention and response. Implementing public, government, and organizational information provides a solid base of knowledge for the deployment and maintenance of a more secure network environment.

7. Conclusion

The pervasive threat of ransomware attacks on Industrial Control Systems (ICS), especially concerning the United States' critical infrastructure, has been thoroughly explored within this research. The core objective has been to scrutinize the viability of deploying virtual testbeds as a means for organizations to craft network environments and experimental scenarios in alignment with their distinct specifications and needs, regardless of the network or system infrastructure. Evidently, a virtual testbed affords network security personnel the flexibility to navigate the perpetually shifting threat landscape, facilitating both swift adaptation and the cultivation of in-depth threat predictions. It enables both private enterprises and government entities to rigorously test the vulnerabilities of ICS systems against a spectrum of ransomware attack modalities, utilizing highly configurable virtual testbeds.

As ransomware threats perpetually evolve, it is imperative that both government and private entities adroitly evolve in tandem. To this end, the deployment of a modular, customizable testbed emerges as a crucial initiative in countering threats across various expertise and resource levels. The testbed, meticulously developed within this research, was employed in experiments exploring diverse network entry points, ransomware architectures, and Indicators of Compromise (IoC), consistently updated to keenly observe each scenario's effects. Employing diverse tactics, such as the creation of two discrete networks and the use of a physical thumb drive, alongside leveraging open-source features like ScadaBR, showcased the testbed’s capacity to adeptly adapt to distinct testing objectives.

A pivotal component of testing networks and systems against threats lies in grasping that the identification of ransomware and additional unwarranted software on a network or system can often be intricate. Therefore, understanding and categorizing warning signs and IoCs becomes paramount to safeguarding network and system security. IoCs, which might encompass specific files, file extensions, or anomalous network traffic, can vary between ransomware variants and may morph as attackers modify their Tactics, Techniques, and Procedures (TTPs). A pertinent example of IoCs for the LOCKBIT ransomware variant is elaborated upon in a March 2023 CISA report (pg. 5).

Promising domains for subsequent research emerge from this project. Firstly, the testbed can be enhanced to incorporate additional hardware components. Although virtual testbeds offer ease of deployment and expandability for imminent threats, organizations might seek to explore the hardware impacts of certain exploits or simulate their actual ICS network more robustly. Sustaining both hardware and virtual testbeds might be beneficial—hardware testbeds provide a more error-resistant environment for standard exploit testing, while virtual testbeds, with their adaptability, could be harnessed for exploits targeting specific configurations. This adaptability feeds into another significant expansion domain for this project: configuring testing for nascent threats. For instance, after reading a BleepingComputer article on VMware ESXi encryption attacks by the Monti ransomware group, an organization could extend the research herein by testing subtle ESXi locking methods within a virtual machine to complement publicly available indicators of compromise related to the Monti group.

While this project has shed light on threat testing via virtual testbeds, its principal objective resides in facilitating further research and testing amongst cybersecurity professionals. Comprehensive instructions for virtual testbed configuration are provided and can be promptly adapted to meet the majority of exploit analysis network requirements. The development and testing of distinct exploit scenarios to accumulate observable exploit effects show promising potential to assist in targeted defense and mitigation efforts.

Nonetheless, this capability must be meticulously adapted to each analyst's objectives. Frameworks furnished for the analysis and summary of threat scenarios and exploit effects might serve to enhance the distribution of threat intelligence throughout an organization. Virtual testbeds stand out as a pragmatic option for modular exploit research, and the work presented in this report can feasibly be integrated into forthcoming development by ICS security personnel and researchers alike.

The virtual realm as a vast, tumultuous ocean, where the serene waters are network connections, and the tempestuous storms symbolize the pervasive threat of ransomware and cyber-attacks. The sturdy ships signify ICS and critical infrastructure, navigating through these perilous waters, seeking secure harbors (security protocols) amidst the cyber storms.

8. References

#StopRansomware guide: CISA. CISA. (2023, May). https://www.cisa.gov/resources-tools/resources/stopransomware-guide

Akbanov, M., Vassilakis, V. G., & Logothetis, M. D. (2019). WannaCry ransomware: Analysis of infection, persistence, recovery prevention and Propagation Mechanisms. *Journal of Telecommunications and Information Technology*, *1*(2019), 113–124. https://doi.org/10.26636/jtit.2019.130218

A. O. Almashhadani, M. Kaiiali, S. Sezer and P. O'Kane, "A Multi-Classifier Network-Based Crypto Ransomware Detection System: A Case Study of Locky Ransomware," in IEEE Access, vol. 7, pp. 47053-47067, 2019, doi: 10.1109/ACCESS.2019.2907485.

Evangelia, E. I. (2018, February). *Vulnerabilities of the Modbus Protocol*. Dione. https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/11394/Evangeliou_1508.pdf?sequence=1&isAllowed=y#:~:text=Modbus%20Protocol%2C%20as%20it%20is,operation%20of%20the%20control%20system

Federal Bureau of Investigation. (2023, March). Internet Crime Report 2022.

Krzysztof Cabaj, Marcin Gregorczyk, Wojciech Mazurczyk, Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics, Computers & Electrical Engineering, Volume 66, 2018, Pages 353-368, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2017.10.012.

M. Al-Hawawreh, F. d. Hartog and E. Sitnikova, "Targeted Ransomware: A New Cyber Threat to Edge System of Brownfield Industrial Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 4, pp. 7137-7151, Aug. 2019, doi: 10.1109/JIOT.2019.2914390.

Making Linux VM a Router. (2021). *Youtube*. Retrieved August 15, 2023, from https://www.youtube.com/watch?v=u1d5Jss3a3g.

Roy, K.C., Chen, Q. DeepRan: Attention-based BiLSTM and CRF for Ransomware Early Detection and Classification. *Inf Syst Front* **23**, 299–315 (2021). https://doi.org/10.1007/s10796-020-10017-4

S. Sibi Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi and B. Raman, "Design of Intrusion Detection Honeypot Using Social Leopard Algorithm to Detect IoT Ransomware Attacks," in IEEE Access, vol. 8, pp. 169944-169956, 2020, doi: 10.1109/ACCESS.2020.3023764.

The MITRE Corporation. (2015). *Overview of How Cyber Resiliency Affects the Cyber Attack Lifecycle*. MITRE. http://www2.mitre.org/public/industry-perspective/documents/lifecycle-ex.pdf

The MITRE Corporation. (n.d.). *ICS matrix*. Matrix | MITRE ATT&CK®. https://attack.mitre.org/matrices/ics/

The MITRE Corporation. (n.d.). *ICS mitigations*. Mitigations - ICS | MITRE ATT&CK®. https://attack.mitre.org/mitigations/ics/

The MITRE Corporation. (2023). *CVE-2023-33921*. CVE. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-33921

The MITRE Corporation. (n.d.). *ICS Tactics*. Mitre ATT&CK®. https://attack.mitre.org/tactics/

Ornaghi, A., & Valleri, M. (n.d.). *ettercap - Linux man page*. ETTERCAP(8) - linux man page. https://linux.die.net/man/8/ettercap

Pipyros, K., Thraskias, C., Mitrou, L., Gritzalis, D., & Apostolopoulos, T. (2018). A new strategy for improving cyber-attacks evaluation in the context of Tallinn Manual. *Computers & Security*, *74*, 371–383. https://doi.org/10.1016/j.cose.2017.04.007

Toulas, B. (2023, August 14). *Monti ransomware targets VMware ESXi servers with new Linux Locker*. BleepingComputer. https://www.bleepingcomputer.com/news/security/monti-ransomware-targets-vmware-esxi-servers-with-new-linux-locker/

Trend Micro. (n.d.). *Industrial Control System*. Definition. https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system#Communication_within_ICS_Systems

Yuki Takeuchi, Kazuya Sakai, and Satoshi Fukumoto. 2018. Detecting Ransomware using Support Vector Machines. In Workshop Proceedings of the 47th International Conference on Parallel Processing (ICPP Workshops '18). Association for Computing Machinery, New York, NY, USA, Article 1, 1–6. https://doi.org/10.1145/3229710.3229726

#StopRansomware: LockBit 3.0 (2023). Retrieved from https://www.cisa.gov/sites/default/files/2023-03/aa23-075a-stop-ransomware-lockbit.pdf.

Annex I
Essential Reading

Book or Article Name
Mitigations - ICS | MITRE ATT&CK https://attack.mitre.org/mitigations/ics/
Federal Bureau of Investigation | Internet Crime Report 2022 https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf
#StopRansomware Guide https://www.cisa.gov/sites/default/files/2023-06/stopransomware_guide_508c_1.pdf

Page intentionally left blank.