NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IMPLEMENTATION OF SECURITY INFORMATION AND EVENT MANAGEMENT SOFTWARE IN A HONEYPOT

by

Jesse Sciuto

December 2023

Thesis Advisor: Thuy D. Nguyen
Co-Advisor: Neil C. Rowe

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE IMPLEMENTATION OF SECURITY INFORMATION AND EVENT MANAGEMENT SOFTWARE IN A HONEYPOT	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jesse Sciuto	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Many industrial control systems (ICS) were originally designed as standalone systems, unconnected to the Internet, that provided little cybersecurity to maximize reliability and response time. Increasingly, these systems have been exposed to the Internet for better control and management, making them vulnerable to cyberattacks. We explored the use of security information and event management (SIEM) technology to improve an ICS honeypot (decoy) system's ability to detect and respond to attacks against electrical-power grids. We integrated commercial SIEM software into the honeypot architecture and deployed a SIEM-enabled instance in a commercial cloud environment. Our experiments showed that SIEM's real-time alerts, data collection and aggregation, and threat analysis helped speed up the discovery of several living-off-the-land and botnet cyberattacks on the honeypot. This work provides a framework for ICS defenders in the Department of Defense and private sectors to use SIEM and honeypot technology to protect critical assets.

14. SUBJECT TERMS security information and event management, industrial control system, honeypot, SIEM, ICS	**15. NUMBER OF PAGES** 77
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

IMPLEMENTATION OF SECURITY INFORMATION AND EVENT MANAGEMENT SOFTWARE IN A HONEYPOT

Jesse Sciuto
Lieutenant Commander, United States Navy
BS, University of Maine, 2012

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN CYBER SYSTEMS AND OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
December 2023

Approved by: Thuy D. Nguyen
Advisor

Neil C. Rowe
Co-Advisor

Alex Bordetsky
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Many industrial control systems (ICS) were originally designed as standalone systems, unconnected to the Internet, that provided little cybersecurity to maximize reliability and response time. Increasingly, these systems have been exposed to the Internet for better control and management, making them vulnerable to cyberattacks. We explored the use of security information and event management (SIEM) technology to improve an ICS honeypot (decoy) system's ability to detect and respond to attacks against electrical-power grids. We integrated commercial SIEM software into the honeypot architecture and deployed a SIEM-enabled instance in a commercial cloud environment. Our experiments showed that SIEM's real-time alerts, data collection and aggregation, and threat analysis helped speed up the discovery of several living-off-the-land and botnet cyberattacks on the honeypot. This work provides a framework for ICS defenders in the Department of Defense and private sectors to use SIEM and honeypot technology to protect critical assets.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

DMZ	demilitarized zone
GNU	GNU's Not Unix
GPO	Group Policy Object
GUI	graphical user interface
HTTP	Hypertext Transfer Protocol
ICS	industrial control system
IEC-104	International Electrotechnical Commission Protocol 60870-5-104
IPv4	Internet Protocol version 4
LotL	living-off-the-land
MSF	Metasploit Framework
NMAP	Network Mapper
RDP	Remote Desktop Protocol
SIEM	security-information and event management
SSH	Secure Shell Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
XML	Extensible Markup Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, I would like to express my deepest gratitude to my advisors Prof. Nguyen and Dr. Rowe for their guidance and support. I have learned so much from both of you and will use the skills you have taught me, such as asking the right question and giving the most concise answer, in my future endeavors.

A big thank you to Andrew Sill for being a friend, the best project partner I have ever had, and for his help integrating his Web server and the botnet discovery.

I would also like to thank my wife, Stacey Champagne, for her encouragement and support. You are a wonderful partner and a wonderful (involuntary) sounding board.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In factories, industrial plants, and modern infrastructures there are many cyber-physical systems controlled by hybrid networks. They combine information technology with operational technology. The demand for smart power grids and industrial control systems (ICS) has continued to increase, and now involves integration into cloud services. What were once isolated ("air-gapped") systems have connected to the Internet, which enables fewer people to control larger systems from further away, but has increased the attack opportunities against ICSs by malicious actors. Security of ICSs has historically been an afterthought. However, today we see attacks such as Russia's Industroyer software and its variants that targeted the Ukrainian power grid in 2016 and 2022 (Serpanos & Komninos, 2022).

A. MOTIVATION

Operational technology architecture differs from traditional information-technology networks in their interaction with the physical world. While information-technology networks are more virtual and stratified by protocol layers, operational-technology networks using the Purdue Enterprise Reference Architecture model for ICSs have five layers. The layers are external/vendor support/cloud access (layer 5), business-logistics systems and enterprise information-technology (layer 4), manufacturing-operations systems (layer 3), control systems (layer 2), and intelligent devices (layer 1) (Garton, 2019). Layer 5 is optional and connects the ICS to third-party cloud systems for additional security or control. Layer 4 contains information-technology devices with specialized functions such as enterprise resource planning, manufacturing execution, and management information systems; if the system connects to the Internet, the outbound connection is on this layer. Security-information and event management (SIEM) software would be installed on layers 4 or 5. Layer 3 contains specialized information-technology equipment to manage industrial processes such as the data historian, real-time database system, engineering and operating stations, and several kinds of servers. The control systems in layer 2 supervise, monitor, send, and receive data from intelligent devices found

on layer 1. Intelligent devices have sensors that receive input from the environment and can interact with the physical world; they are also called intelligent electronic devices or remote terminal units.

Digital honeypots are decoy systems intended to either lure attackers away from critical infrastructure or to encourage them to stay on the system for gathering attack intelligence. Both kinds of honeypots can collect data and analyze it to determine methods of attack and its targets. Previous work has been done at Naval Postgraduate School (NPS) with ICS honeypots simulating an electrical grid (Dougherty, 2020). Currently, our power-grid honeypot lacks a real-time alert function and rich data-collection capabilities for attacks. Security-information and event management (SIEM) technology can provide a broad view of network activities by consolidating security-relevant information such as firewall logs, Web filtering logs, data-network sensors, event logs, application logs, and so on for detailed analysis. Adding SIEM capability to honeypots enables more detailed analysis of attacks against them.

B. RESEARCH METHODOLOGY

We first compared several commercial SIEM products. We developed high-level designs for scenarios of our honeypot with SIEM. We implemented several stages and validated each to confirm the product worked as advertised to detect and report potential security breaches. We deployed a SIEM-enabled honeypot instance in a cloud environment and analyzed the SIEM reports to assess the technology.

C. THESIS OUTLINE

Chapter II surveys SIEM and its applications, cyber-deception with honeypots, industrial control systems, the DigitalOcean cloud environment that we used, and other prior work. Chapter III describes the process and criteria to select a SIEM application, gives an overview of our honeypot, and discusses possible ways to add SIEM capabilities. Chapter IV discusses the selected design, its implementation and testing, and a live deployment in the DigitalOcean cloud environment. Chapter V shows the results of the experiments and the effectiveness of the SIEM technology. Chapter VI summarizes this report and concludes with ideas for future work.

II. BACKGROUND

A. SECURITY INFORMATION AND EVENT MANAGEMENT

Security-event and information-management (SIEM) technology enables network operators to collect information from multiple sources, detect and analyze the pattern of anomalies, and respond to potential issues. A SIEM deployment has sensors, aggregators, and data synthesizers (Bhatt et al., 2014). When an alert is triggered, network administrators or security-operations center personnel use the data to determine whether the alert is a false positive or actual malicious activity, and possibly protect the network (Bhatt et al., 2014). Typically found in enterprise and critical systems, SIEM software is customizable to best serve the network on which it is deployed.

Although each SIEM configuration can be unique, some components are common. Data collectors collect log data from sources throughout the network (e.g., servers, endpoint devices, databases, firewalls, routers, switches, and applications), and send it to a centralized location for processing. A correlation engine analyzes the data, runs detection rules on it including those tailored to the individual systems, and compares data to signatures of known attacks. A summary with a timeline is presented to network operators through dashboards and generated reports. If an event triggers a rule, the SIEM application can begin mitigation while starting incident response. After the incident is resolved, the data collected by the SIEM system can support forensics analysis by correlating it to other events for improving future responses (González-Granadillo et al., 2021).

B. INDUSTRIAL CONTROL SYSTEMS

Industrial control systems are in every factory and on every ship at sea controlling the flow of traffic, and in many other applications. ICSs contain cyber-physical systems that interact with the physical world by taking inputs such as temperature, voltage, and humidity, and determining an output to keep the ICS within predefined parameters. Previously, the only technology in industrial settings was proprietary operational technology designed to be unconnected to the Internet. With improved networking, many information-technology users are using lower-cost Internet-attached devices using the

Internet Protocol version 4 (IPv4) to improve efficiency and reduce overhead for system management, but at the expense of cybersecurity (Stouffer et al., 2015).

Many ICSs have a hierarchical structure to establish a logical boundary between the information-technology and operational-technology network segments by adopting the Purdue model (Figure 1) for cybersecurity, which places a "demilitarized zone" (DMZ) between the Internet-connected enterprise zone and the operational technology manufacturing zone (Green et al., 2016).

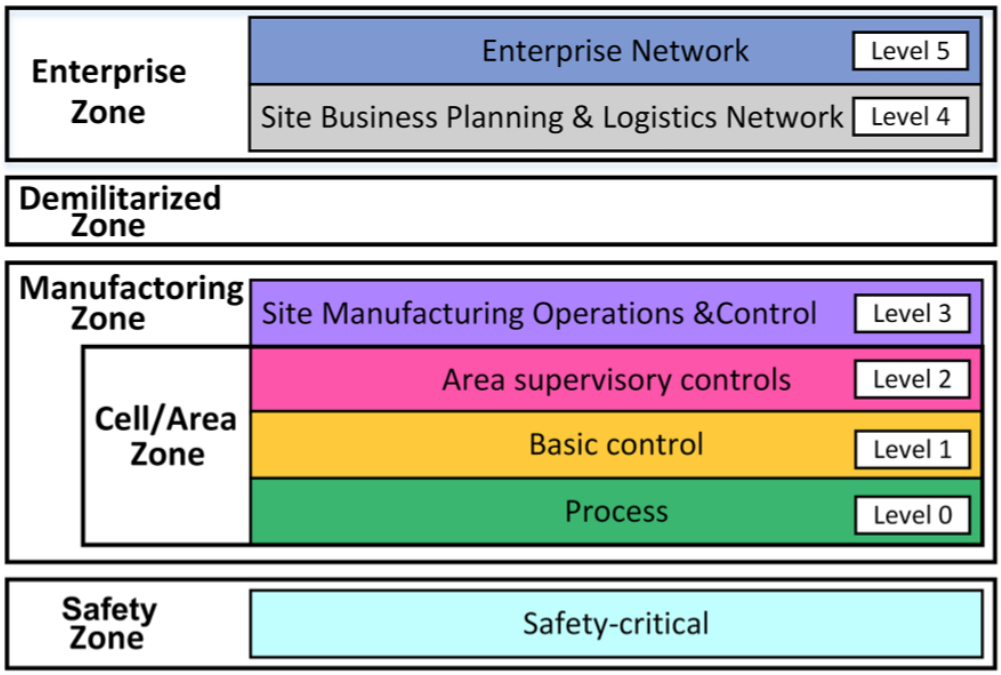


Figure 1. The Purdue Model. Source: Green et al. (2016).

Though the DMZ provides some protection against attack, a modern zero-trust model assumes that intruders can access any part of the network (Zanasi et al., 2022). SIEM technology in an ICS can continuously monitor all architectural layers, from the low-level operational-technology to the upper information-technology layers, which permits thorough scanning for anomalies needed for a zero-trust model.

C. CYBER DECEPTION AND HONEYPOTS

Cyber deception is a defensive technique designed to sow confusion among those with malicious intentions towards a system and misdirect them away from critical network resources. If done correctly, cyber deception can disrupt and delay an intruder enough to cause them to either forgo their assault or provide defenders with enough time and information to analyze the attack and contain damages. The ultimate target of deception is the adversary's mind. Cyber deception techniques offer potential advantages over traditional security controls, and computer networks are good settings for inducing confusion due to their inherent complexity in accessing and understanding them remotely (Ferguson-Walter et al., 2018).

Honeypots are resources designed to attract attackers by simulating a desirable target (Climek et al., 2016). In a honeypot, deception can be implemented in the operating system, the applications software, or the configuration files. However, to entice attackers to spend effort on attacking a honeypot, it will often have reduced cybersecurity features and use outdated software versions with known vulnerabilities. Honeypots can be either low-interaction or high-interaction. Low-interaction honeypots are easy to set up, have low risk, low cost, and low maintenance; however, they are more easily identified because of their limited user interactions (Franco et al., 2021). High-interaction honeypots are harder to set up and maintain, but they can more closely mimic the actual system and can respond more appropriately when queried. High-interaction honeypots simulate real systems in detail and collect information about the attacker's movement and actions; they provide higher-fidelity data for analysis (Franco et al., 2021).

D. RELATED WORK

In 2012, researchers created a high-interaction honeypot containing four hosts using several Linux distributions, a modified Secure Shell Protocol (SSH) server which allowed remote users to connect to it, and an application named SYNEMA that implemented SIEM (Briffaut et al., 2012). They also added a honeywall (Davis, 2003) to limit the outgoing network traffic and avoid denial-of-service attacks emanating from the honeypot. The SYNEMA software improved forensic capabilities to analyze attacks on the

honeypot by ingesting information from multiple sources, correlating the data using the researcher-defined rulesets, and displaying it graphically on dashboards. The researchers could see the raw data, which was collected over two years for analysis on important event types, such as the sessions during which the attacker downloaded software. SYNEMA allowed the researchers to find an instance of a sophisticated attacker modifying scripts and configuration files to adapt malware programs to the local context (Briffaut et al., 2012)

Placing a SIEM on a honeypot improves cybersecurity practitioner's technical abilities by providing students with hands-on experience (Mohd et al., 2022). Researchers constructed a honeypot and used data from the campus's network in a SIEM. Students analyzed the network traffic using the SIEM and answered a series of questions. Students agreed that the hands-on lab experience had helped them understand and apply cyber defense.

In response to honeypots being deployed on the Internet, malicious actors have developed methods to identify and catalog potential honeypots to avoid or attack them. The Honeypot Hunter program is an example (Krawetz, 2004). It evaluated honeypots to determine their detection mechanisms, and these mechanisms suggested methods to avoid being detected as a honeypot, mainly by increasing the interaction level of a honeypot system to better emulate a real system.

Logging the interactions with a honeypot can create large volumes of data. Researchers implemented an intermediary logging module in their honeypot architecture (Singh & Joshi, 2011). It analyzed received packets and correlated the packets within a flow, where a flow is defined as having the same source IPv4 address, source port, destination IPv4 address, destination port, and flags. Researchers reduced the disk space required to store the packets by 70.5% without losing any data.

Honeypots for programmable logic controllers do not always have enough interaction capabilities to gather useful data (López-Morales, 2020). HoneyPLC was developed to be a high-interaction ICS honeypot with advanced protocol simulations that could deceive honeypot-identifying software. Researchers configured a TCP/IP server,

HTTP server, a Simple Network Management Protocol, and S7Comm (a common ICS protocol) server to produce a realistic ICS network to collect data on potential malicious actors.

ICS honeypots are difficult to design and implement due to the specialized equipment used in an industrial environment. Virtualizing an ICS honeypot improves interaction capability, scalability, and maintainability, and is more cost-effective. Researchers developed MiniCPS, a virtual ICS honeypot-in-a-box written in Python that emulates a small water treatment facility (Antonioli et al., 2016). It was deployed on a commercial cloud service with a minimal configuration and was attacked quickly. To produce the same results as traditional equipment would have been more expensive and would have required more time.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

A. PROBLEM SET

Cyberattacks against ICSs are increasing in complexity, and attacked systems cost an average of $5 million in losses and 50 days of system downtime (Alladi et al., 2020). The United States Department of Defense relies on civilian companies to provide its resources (weapons, hardware, fuel, and so on) and is therefore a key stakeholder in the cyber defense of ICSs. Cyber deception is effective for cyber defense. The aim of our research was to improve cyber-deception capabilities and to prevent attacks against ICSs or lessen their cost and downtime.

Previous honeypot research could not correlate events, provide real-time alerts, or generate statistical reports when malicious activity was detected (Meier, 2022). Attacks were undetected for days after they occurred, logs were deleted, and some events could not be explained without rich log data and parsing and searching through that data (Dougherty, 2020). By adding a SIEM to the honeypot, those requirements should be met.

B. DIGITALOCEAN CLOUD ENVIRONMENT

DigitalOcean is a cloud-based infrastructure-as-a-service platform that provides virtual machines called "droplets." Droplets can be stationed around the world and use the processing, memory, and secondary storage of data centers there (DigitalOcean, n.d.-a). Droplets in the same region may connect with one another by private IP addresses through the DigitalOcean Virtual Private Network or public (routable) IP addresses routed through standard Internet protocols. Droplets are accessible through command-line or graphical user interfaces. DigitalOcean also supports firewall rules outside the droplets to reduce intrusions.

The DigitalOcean cloud environment is well suited for honeypot research. Droplets can be created from a default template or from previously saved images, allowing better scalability. With traditional hardware, new computers or servers must be procured and configured to increase the capacity of the honeypot. The cloud environment permits the deployment of the droplets to appear to be in several countries, which is hard to do with

hardware. Each new droplet created on DigitalOcean receives an IPv4 address that has been recycled from a pool of addresses allocated to DigitalOcean. Some addresses reassigned by DigitalOcean have been associated with questionable or inappropriate websites previously. Addresses that have been suspected of being honeypots are put on lists in anti-honeypot technology and will be avoided by malicious actors (Krawetz, 2004).

C. CRITERIA FOR SIEM SELECTION

SIEM software combines data collection and analysis with threat detection and security-incident-management tools to correlate events and automate threat responses. With much commercial and open-source SIEM software available, we developed criteria for our ideal solution. For each candidate program, we assessed its source, ease of implementation and use, type of license, operating-system compatibility, and scalability. We looked for products with standard features of most SIEMS; for example, that do not require any tailored code and that can ingest many standard data types (like standard logs, user-generated logs, and packet captures). They must filter and search for specific events in real time and alert researchers on specific triggers such as an unauthorized software download. They also should aid digital forensics by summarizing triggers statistically.

Among the many SIEM products available, we first had to identify the major ones. Only a few of roughly two dozen SIEMs are produced by well-known sources and are acceptable for an ICS environment (González-Granadillo et al., 2021). The Gartner company evaluates SIEM products each year (Gartner, 2023) and summarizes their evaluation with quadrants that are labeled as challengers, leaders, niche players, and visionaries. We defined the provenance of a product in one of three ways:

- Highly respected: the product was created by a large company and featured by Gartner in either the leader or challenger quadrant in the previous five years.
- Respected: the product is still well known and used in continuous monitoring, but has a smaller production team, and is featured by Gartner in either the niche player or a visionary in the previous five years.

- Little-known: the product is created as a personal project or by a small team of unknown personnel and has not been featured by Gartner.

Due to the frequent creation and setup of virtual machines in honeypots, it helps for SIEM software to be easy to install and configure, as each experiment requires a fresh IPv4 address. Factors to assess the ease of implementation include the number of components to be installed and the documentation provided. We defined the ease of implementation in one of three ways:

- Simple: the product does not require any specialized knowledge, such as coding or scripting, and has much documentation for users.
- Moderate: the product has some complexity during setup and does not contain all the required modules with initial installation.
- Difficult: the product requires knowledge of operating systems and contains at least some components that must be manually installed without sufficient documentation.

To ensure continuity of use the type of license used by the SIEM product had to be considered. Open-source SIEM software products typically use GNU's Not Unix (GNU) General Public License version which provides copyleft protection, enabling changes to the software (Ganguly, 2007). Several open-source SIEM products that used a GNU General Public License were also free for non-commercial or academic use. Commercial products vary in their license and pricing schemes.

The three primary pricing types for commercial software licenses are based on the amount of data ingested and processed, the number of users or workstations, and a fixed subscription price. Some vendors provide a reduced price or free price for academics and research. Data-use licenses are preferable for us since our project, compared to a full enterprise network, does not ingest much data. SIEM products require that log data be sent to a server or cluster running a correlation engine. Nearly every SIEM program recommends placing their correlation engine on a server due to the high requirements of data ingestion and processing.

For scalability of SIEM software, adding network nodes can give more data collection. A good SIEM product enables adding and removing data-collection nodes. We defined the scalability of the product in one of four ways:

- Very high: the program is cloud-based, and the scalability is limited only by network hardware constraints.
- High: the program can add data sources easily and has a large limit on the number of sources.
- Moderate: the product requires either a specialized setup or has limits on the number of data sources.
- Low: the product requires much time and technical investment when adding sources into the system.

Product Comparisons and Decision

We examined ten SIEM candidate products Suricata (Suricata, n.d.) with ELK Stack (Elastic, n.d.), LogRhythm (LogRhythm, n.d.), Exabeam (Exabeam, n.d.), InsightIDR (Rapid7, n.d.), Splunk Enterprise (Splunk, n.d.-d), QRadar (IBM, n.d.), ArcSight ESM (ArcSight, n.d.), OSSIM (AT&T Business, n.d.), OSSEC (OSSEC, n.d.), and Sentinel (Microsoft, n.d.), using the above criteria for our honeypot systems (see Table 1). We decided to use Splunk Enterprise.

Table 1. Evaluation of SIEM products.

Product	Provenance	Implement-ation	License	OS	Scalability
Suricata / ELK	Respected	Difficult	GNU GPL	Multiple	Low
LogRhythm	Highly Respected	Simple	Varied	Multiple	High
Exabeam	Highly Respected	Simple	Per User	Multiple	Very High
InsightIDR	Highly Respected	Simple	Per Asset	Multiple	Moderate
Splunk	Highly Respected	Moderate	Use-Based / Free for Dev and Academics	Multiple	High
QRadar	Highly Respected	Simple	Based on Users / Assets and Servers	Red Hat	High
ArcSightESM	Respected	Simple	Fixed	Multiple*	High
OSSIM	Respected	Moderate	Fixed	Linux	High
OSSEC	Little Known	Moderate	GNU GPL v2	Linux	High
Sentinel	Highly Respected	Simple	Use-Based	Multiple	Very High

*ArcSight ESM can be used on multiple operating systems but does not support Ubuntu 20.04 LTS.

Splunk is a well-respected product (González-Granadillo et al., 2021). It has a comprehensive installation guide and well-developed library of technical documentation, as well as a Splunk Community message board with many documented issues and their solutions. Some work is required to set up the universal forwarders, small applications installed on machines to send collected data to a remote Splunk server. Splunk provides developers and nonprofit academic institutions with a free license that allows up to 10 gigabytes per day of data ingestion on a six-month and twelve-month recurring basis. Splunk supports both Windows 10 and Ubuntu 20.04 LTS operating systems. No limit is set on the number of remote nodes that can send data to the core application.

D. SPLUNK OVERVIEW

The core application of Splunk is an indexer, which ingests and summarizes raw data and stores it in a database. In response to manual or automatic searches, Splunk can generate reports, display dashboards of data in the form of charts and graphs, check data against predefined rulesets, and trigger alerts (Splunk, n.d.-d). Multiple indexers can be deployed in the same network to concurrently ingest data from different sources. The

indexer can also ingest logs and other data sources from the machine that it is installed on, but it cannot directly view data from remote hosts.

The Splunk Universal Forwarder is a streamlined version of Splunk Enterprise that forwards unparsed data to the indexer (Splunk, n.d.-d). For large volumes of data, an improved forwarder can be installed which has most capabilities of the indexer plus parsing of data before transmission (Splunk, n.d.-d). The indexer can create dashboards using a dashboard tool Splunk Dashboard Studio, which displays charts, tables, and graphs. The dashboard is accessed via a Web browser on the host or server that is running the indexer. Splunk also can generate reports from the data, either ad hoc or at scheduled times, and automatically saves them for future review (Splunk, n.d.-d).

Splunk can index many kinds of timestamped data (Splunk, n.d.-d). By default, it can ingest standard operating-system and network data and user-defined data structures. The primary sources are usually operating-system logs. These include:

- Windows logs: Firewall, WinEventLog:Application, WinEventLog:Security, and WinEventLog:System, Hypertext Transfer Protocol (HTTP), and Domain Name System.
- Linux logs: authentication, access, and syslog.

Windows Group Policy Objects can be changed to modify the number of events sent to the logs. When a data stream arrives at the indexer, it is first parsed into events (Splunk, n.d.-d). Each event is further broken down and identifiers are assigned to features of the event such as host, source, or sourcetype (Splunk, n.d.-d). Users can search by identifier fields to generate reports, view statistics, or trigger events. Users can also save the data to different indexes to keep feature data separate.

Methods of searching data can be defined in a Search Processing Language (SPL) developed by Splunk, which is based on the Unix pipeline mechanism and Structured Query Language (Splunk, n.d.-b). SPL contains commands for string searches, commands for formatting or counting the result, functions such as computations like sum, average, AND, OR, and NOT, and clauses such as GROUPBY and WHERE. Queries can return

events that match the query, sample-based patterns, graphical statistics, or customizable visualizations. Queries can be automated, and if user-specified conditions are met, Splunk will trigger prespecified events such as email or mobile alerts, outputting the results to dashboards, or running a script.

Splunk uses a NoSQL database for managing the stored data with key-value pairs (Splunk, n.d.-d). Ingested data is compressed, location files are created along with several other metadata files, and the data placed into a bucket organized by age. Data buckets are aged through several "temperature" ratings:

- Hot: new data from the universal forwarders. The hot bucket is where data is parsed and indexed.
- Warm: data that has been moved from hot buckets.
- Cold: Older data that is still searchable, though will soon be frozen depending on the user-defined policy.
- Frozen: If a bucket is not deleted after rolling from cold it is archived in a separate location as defined by the user. This bucket is not searchable.
- Thawed: Frozen buckets that have been unarchived and placed back in the Splunk database. These buckets are searchable.

Splunk requires ingested data to search when the user provides queries. The data can be on the same machine as the indexer or on another machine; to ingest and parse data, Splunk only requires a location of the data. If data is on a remote system, a universal forwarder must be installed on the remote system to send collected data to the indexer. An email server is needed to provide email alerts. For this work, Google mail was used as a Web server to send the email alerts.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DESIGN AND IMPLEMENTATION

A. DESIGN OF GRIDPOT WITH SIEM

1. NPS Honeypot Overview

The NPS honeypot we used contained three virtual machines ("droplets"): the GridPot machine, the user-interface machine, and the logging machine. During previous NPS ICS honeypot research, an attacker accessed the Windows virtual machine and deleted the logs, preventing detailed analysis of the intrusion (Dougherty, 2020). Subsequently we hardened the design by adding a droplet that collected the logs of the other two machines, which was only accessible through the DigitalOcean platform and few ports (Meier, 2022).

The logging machine ran Ubuntu 20.04 LTS operating system. We further improved collecting and parsing of the raw data by installing Splunk Enterprise on the logging machine. The indexer collected data from all the machines, indexed it, and stored the data in a database, which was then accessed and queried through a Web interface. The logging machine contained the Splunk indexer, database, and the Web server to access the graphical user interface.

The GridPot machine ran on Ubuntu 20.04 LTS Linux and ran the GridPot software, with GridLab-D simulating the IEC 60870-5-104 (IEC-104) protocol network traffic of a 13-node residential housing neighborhood (Bieker &Pilkington, 2020). GridPot previously sent the simulated data to a program IndigoSCADA on the user-interface machine (Enscada, 2021). A user interacted with IndigoSCADA to see voltage, amperage, power values, and switch settings in GridLab-D, displays which improved the believability of the honeypot. Though intrusion into the GridPot machine was not expected, we still installed a universal forwarder on the GridPot machine to forward log and configuration data to the indexer to detect possible attacks.

The user-interface machine had a base operating system of Ubuntu 20.04 LTS Linux, which ran a VirtualBox virtual machine with a Windows 10 instance. The Windows 10 virtual machine imitated a control interface for a station of a small electrical company. Attackers were prompted and encouraged to connect remotely into the Windows 10

machine through Remote Desktop Protocol (RDP). We installed a universal forwarder on both the base Linux operating system and the Windows 10 virtual machine, expecting the Windows 10 machine to generate most data for the indexer.

2. Deployment Scenarios

Several architectures can collect data and test the SIEM software. The droplets were accessed and configured by researchers using a password-protected SSH tunnel over port 22.

a. Single Region with Single Indexer

A honeypot contained in a single DigitalOcean region was composed of three virtual machines with only the Windows 10 machine and a Flask-based Web server accessing the Internet. The Windows 10 virtual machine was accessed using an RDP connection over port 3389. The collected data was encrypted and sent from the universal forwarders over TCP port 8000 to the indexer on the logging machine. For Splunk, the indexer provides a search head (Splunk, n.d.-d) where users or scripts run search queries on the database. The indexer also contains a Web server that allows users to access the search head from any machine by entering the IPv4 address of the machine running the indexer and logging in to an authorized user account. The search head and dashboards were accessed by a Web server on port 443 to provide a graphical analysis of the parsed data showing trends and potential malicious activity (Figure 2).

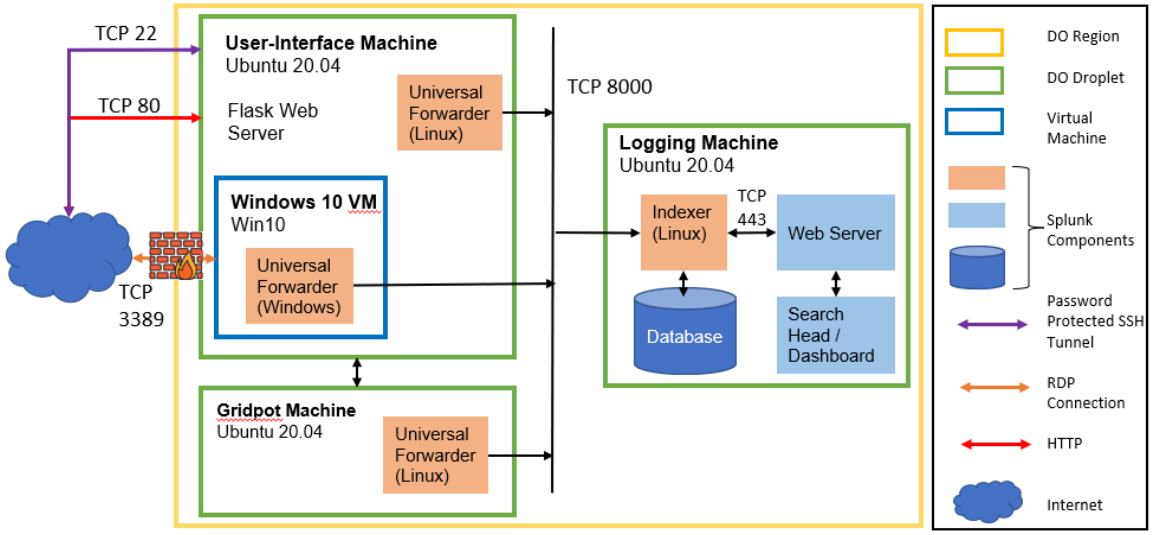


Figure 2. Architecture for a single DigitalOcean region deployment.

b. Multiple Regions with Single Indexer

To increase the rate of malicious traffic, user-interface and GridPot instances can be deployed to geographically dispersed regions such as Germany or Singapore where DigitalOcean provides services. These secondary regions can be more lightweight with only user-interface and GridPot droplets. Data generated by the secondary regions can be encrypted and sent over TCP port 8000 to the logging machine in the primary region. Diversifying regions provides more variety of targets for malicious actors and allows comparisons between regions. A drawback of this design is the reliance on a single indexer, which is a single point of failure and under more network load from ingesting more data (Figure 3).

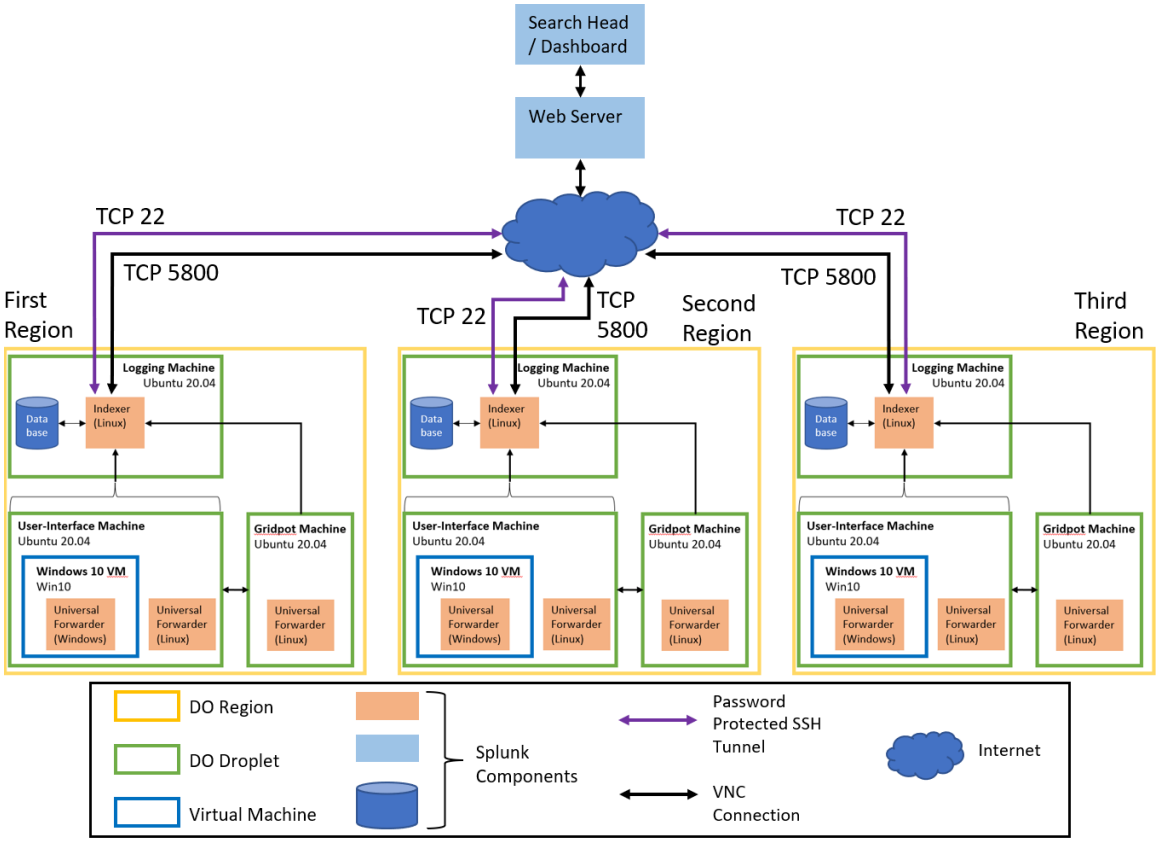


Figure 3. Architecture for multiple DigitalOcean regions sharing a single indexer.

c. Multiple Regions with Multiple Indexers

To prevent a single indexer from being overwhelmed, we can add an indexer to each region and configure the search head to query across all indices. Databases can copy data to prevent loss if failed or compromised. This is the most robust and redundant architecture, but its DigitalOcean implementation is the most expensive to maintain due to DigitalOcean's pricing based on the number of droplets used (Figure 4).

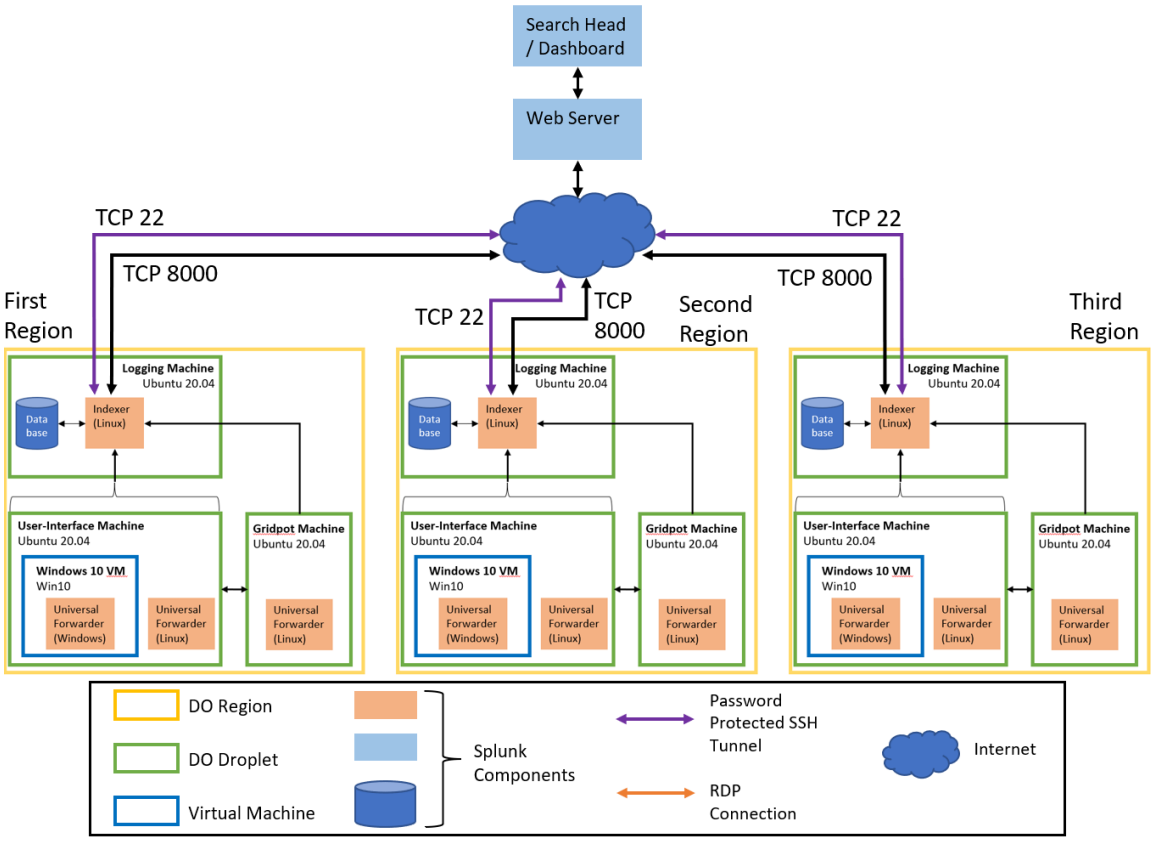


Figure 4. Architecture for multiple DigitalOcean regions with multiple indexers.

d. Multiple Regions with Shared Indexer and Shared GridPot Droplet

To minimize the number of droplets used, several regions can share a single GridPot droplet using DigitalOcean's Netmaker connection. Netmaker provides secure network access between globally distributed droplets (DigitalOcean, n.d.-b). This architecture suffers the same drawback as the multiple architecture with a single indexer, in that the indexer could be a single point of failure (Figure 5).

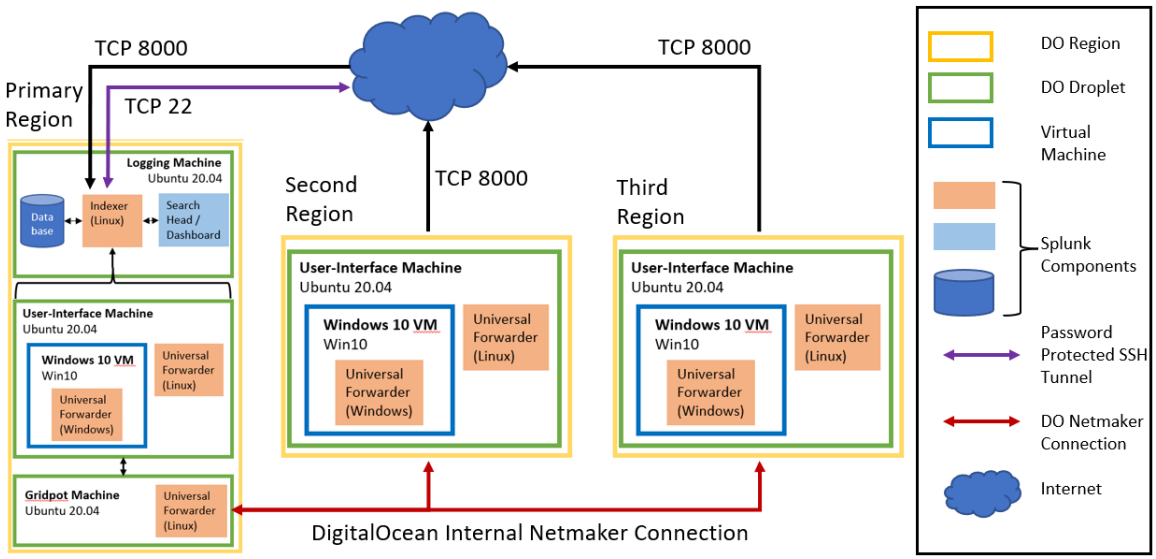


Figure 5. Architecture for multiple regions with shared indexer and shared GridPot droplet.

For our research we selected a single region with a single-indexer configuration (Figure 2). This architecture is the simplest to set up and provides all the necessary conditions to test the implementation of a SIEM in a honeypot.

B. DESIGN OF SYSTEM UNDER TEST

Splunk provides several licensing options based on application use and data ingest quantities: Enterprise, Enterprise Cloud, and Developer. We used a Developer license. This license provides access to all the functions and features of Splunk Enterprise.

1. Indexer Configuration

The indexer is the core part of the Splunk architecture where data is sent, and queries are done. We installed the indexer on the logging machine, and the default installation automatically constructed the database and automatically configured a Web server providing a graphical user interface (GUI). From the Web GUI, we configured the indexer for a single region deployment (Figure 2). To receive data, we chose a high-numbered port that was unlikely to receive any other traffic, to avoid interference with the indexer's data collection. Splunk can ingest two types of logs, the default operating system logs and user-defined logs. When Splunk reads a log, it breaks down the information and

assigns fields to each component. For example, the Windows security event log's first field is the event source, followed by the provider's name, event identification, version and so on. Since this is a standard log, Splunk knows to assign the identifier "event source" to the first field of a log record. Refer to Appendix B for an example of two standard Windows logs. When a user searches by event source, Splunk only searches for the "event source" identifier, which improves search efficiency. User-defined logs must be configured in Splunk with defined fields. We configured Splunk to ingest the following Linux operating system logs on all Linux machines: kernel, authentication, access, and system logs. On Windows systems, it ingested the following event logs on the Windows virtual machine: application, security, and system logs. We did not use user-defined logs.

2. Forwarders Configuration

Universal forwarders were installed on the Linux host of the user-interface machine and on the Windows virtual machine. They required some manual configuration. By default, Windows collects only a portion of logging data, but gives the option to enable additional data to be logged in the Group Policy Object (GPO) editor. On the Windows virtual machine, we used several PowerShell commands to activate firewall logging, set a maximum log size, and to collect data about allowed and dropped connections. An "add monitor" command was used in the Splunk universal forwarder on both the Linux host and the Windows virtual machine to send the firewall data to the indexer.

3. Add-on Applications

We installed two additional Splunk-provided applications, Splunk Security Essentials (Splunk, n.d.-e) and Splunk Stream (Splunk, n.d.-c). Splunk Security Essentials provides more advanced intrusion detection and response. The application contains modules developed by Splunk from the MITRE ATT&CK Framework (MITRE, n.d.) and the Cyber Kill Chain (Lockheed Martin, n.d.). The modules provide hundreds of search queries for the user to implement that display any results detected. We implemented the Basic Brute-Force Detection and Basic Scanning modules as email alerts. The Basic Brute-Force Detection module searches data from Windows security logs and Linux authentication logs and looked for a single source IPv4 address trying to connect to a

machine more than a predetermined number of times within a short duration. The Basic Scanning module ingests firewall data and looks for a single source IPv4 address trying to connect to a predetermined number of ports on the Windows virtual machine. Other than email alerts, if an automated Splunk query detects a result, it can send the log event to another machine, run a user-defined script, add the result to a dashboard, and so on.

Splunk Stream enables limited packet capturing and network monitoring. With it, we can see on the dashboard the bandwidth use, number of active "netflows," total packets captured, and dropped packets in real time. Splunk implements a proprietary form of the NetFlow protocol (Cisco, n.d.) which captures information such as the source and destination IP address, source and destination ports, and application data. Splunk's "netflow" implementation groups packets being used by the same machines for the same applications together and combines the data into a single data stream. Splunk Stream only examines header information which includes source and destination addresses, port, bytes in, and bytes out. By default, Splunk Stream only estimates the amount of data streaming per protocol but does not log that data. We configured the indexer to collect packet data from HTTP, the Internet Control Message Protocol, the Internet Protocol (IP), the Transmission Control Protocol (TCP), and the User Datagram Protocol. Data from the Remote Desktop Protocol (RDP) is included in the TCP capture. Collecting IEC-104 traffic requires custom configuration, which due to time constraints, we did not do.

4. Windows Event Codes and Group Policy Objects

Several Windows Group Policy Objects (GPO) were modified enabling additional Windows event codes to be generated and logged (Microsoft, 2018). Windows event codes are generated by specific actions of users, applications, and the operating system, and provide a history of the activity on the machine. A GPO can have multiple event codes describing different audit events. The modified GPOs and the event codes they generate that Splunk looked for are listed in Table 2. The event codes are stored in different Windows log files. The universal forwarder must be configured to monitor these log files.

Table 2. Monitored event codes and GPOs.

Event Code	GPO	Log
4776	Audit Credential Validation	Performs a validation test on account logon credentials
4720	Audit User Account Management	User account created
4632	Audit User Account Management	Administrator account created
4688	Audit Process Creation	Process started
4689	Audit Process Termination	Process stopped
5712	Audit Process RPC Events	Audit inbound remote procedure calls
4622	Audit Directory Service Access	Active directory domain service accessed
5136	Audit Directory Service Changes	Active directory domain service changed
4634	Audit Logoff	Logon session terminated
4624	Audit Logon	Logon success
4625	Audit Logon	Logon failure
1102	Enabled by default	Audit log was cleared

5. Alerts

Splunk alerts are queries that generate notifications if the conditions specified in the queries are met. To create an alert, the user enters a query into the search bar, saves that query as an alert, and specifies when the query should run (real-time or scheduled) and the actions to be taken if the query returns a result. To detect malicious actors in real time during the live experiments, we configured several alerts that emailed researchers when

their conditions trigger. High-priority alerts were generated every 5 seconds for observed activities considered as potentially malicious on the user-interface machine, on data that had been parsed by the indexer since the previous search. They were set for a successful logon, a downloaded file, creation of a new local user or administrator account, or clearing of the Windows event-log file. To conserve system resources, other alerts were scheduled once an hour including basic scanning detection and brute-force-logon attempt detection.

When an alert condition is triggered, Splunk can run a script, send an alert to a mobile device, and output the alert and conditions to a file. We configured Splunk to display the alerts on a dashboard and email the research team. The alerts contained which condition was met, the search string used, which machine detected the condition, and date and time.

C. DESIGN OF EXPERIMENTS

1. Control Testing

Before exposing the user-interface machine to the Internet, we ran three sets of tests to verify that the configured alerts worked as intended. The high-priority alerts were based on Windows event codes and should be triggered by the following actions on the user-interface machine:

- Successful logon with RDP.
- Creation of a local-user or administrator account by the command line.
- Deletion of old event-log data.

The basic scanning alert was also configured to trigger when a user tried to bind with 500 or more ports on the user-interface machine, or if more than 500 unique IPv4 addresses tried to connect to the user-interface machine within an hour.

For these tests we used a Kali Linux virtual machine. Kali Linux is a Linux distribution used for penetration testing and ethical hacking, and it comes preloaded with tools such as the Network Mapper (NMAP) and Metasploit Framework. NMAP is an open-source tool to map and scan networks to audit security practices (Lyon, 2023). It sends

packets to a target machine or network to solicit simple responses. We used NMAP on the IPv4 address of the user-interface machine with the -p (port) option to scan for open ports from 21–10000. The NMAP scan triggered a single basic scanning alert that sent an email to the researcher and Splunk displayed the IPv4 address of the machine which ran the NMAP scan.

We installed Nessus on the Kali Linux virtual machine. Nessus is a scanning tool to detect security vulnerabilities on a host machine or network (Tenable, 2023). Like NMAP, Nessus scans machines in a network and looks for vulnerabilities recorded in the Common Vulnerabilities and Exposures database maintained by the MITRE Corporation (MITRE, n.d.). The default Nessus scan did not trigger the basic scanning alert in Splunk, but it did identify six vulnerabilities and twenty-nine dangerous Windows settings listed in Appendix A. We used the information in the Nessus report to select several Metasploit Framework (MSF) modules for penetration testing on the user-interface machine. The top five vulnerabilities identified by Nessus related to Secure Socket Layer (SSL) and Transport Security Layer protocols. We tried to use the following SSL MSF modules in Metasploit to exploit and establish a remote connection with the user-interface machine:

- Payload/python/shell_reverse_tcp_ssl
- Payload/cmd/unix/python/shell_reverse_tcp_ssl
- Payload/cmd/windows/powershell/powershell_reverse_tcp_ssl
- Payload/cmd/unix/python/shell_reverse_tcp_ssl
- Payload/cmd/unix/reverse_openssl

Due to the patches already installed on the Linux host and Windows virtual machine, no modules succeeded. Splunk did not trigger a logon alert when these modules were run as they were unsuccessful.

Several damaging ransomware attacks are based on the EternalBlue exploit (Liu et al., 2022). EternalBlue exploits a vulnerability in the Server Message Block protocol used by Windows on port 445, and allows attackers to remotely execute code and potentially

access a system. We ran the "ms17_010_eternalblue" module in MSF against the Windows virtual machine, but the attack never reached our victim machine.

2. Live Testing

With the user-interface machine connected to the Internet, we did four experiments. All experiments used the "single region with single indexer" deployment configuration shown in Figure 2 and allowed traffic on ports 22 (SSH), 3389 (RDP), and 5800 (VNC) to make inbound connections. Each experiment used a new droplet with a different IPv4 address. Every experiment used the same set of default alert conditions, The alert conditions were the detection of a successful logon, detection of a downloaded file, creation of a new local user or administrator account, or clearing of the Windows event-log file.

a. Experiment 1

The objective was to determine if Splunk's network monitoring, logging, searching, and event correlation capabilities could detect attacks that were unlikely to be identified with only packet captures. The Flask Web server was unused. We expected to see the most traffic on port 3389 and to be told if alert trigger conditions were met. We configured Splunk to generate alerts for multiple login attempts on the Windows user-interface on port 3389. The experiment ran for 96 hours.

b. Experiment 2

Experiment 2 looked for executables that are commonly used in living-off-the-land (LotL) attacks. These attacks use tools which are already part of the operating system, such as PowerShell or operating-system management applications (Ongun et al., 2021). Only ports 22, 3389, and 5800 were open to inbound traffic. The experiment ran for 83 hours.

c. Experiment 3

Experiment 3 was the same as Experiment 2 with the addition of the Flask Web server. Its objective was to gather HTTP traffic data on port 80 and have Splunk to sort and parse the data. For this experiment we opened ports 22, 3389, 5800, and 80 (HTTP) to inbound traffic. After 48 hours, the experiment was showing far less network traffic than

the previous two experiments. We checked the IPv4 address that we had been assigned by DigitalOcean with the Shodan search engine and were told that the address had been previously tagged as a honeypot. We shut down the droplet after 60 hours.

d. Experiment 4

Experiment 4 was configured the same as Experiment 3 with a new droplet address. We verified the assigned IPv4 address as not being identified as a honeypot by the Shodan search engine. The experiment ran for 96 hours.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS

For each experiment, Splunk queries created before starting the experiment ran in real-time with the experiment, and alerts were generated as the conditions specified in the queries were met. During post-experiment analysis, additional queries were issued for testing. We started each post-experiment analysis by running queries to count unique IPv4 addresses and their countries of origin, count the number of connections made on each port, and determine what data transferred between our droplet and the source address. Next, we ran additional post-experiment Splunk queries on each experiment's collected data to detect local-administrator account creations, masquerade attacks, tampering with Windows Defender, port scanning, unusually long commands, system-elevation attempts, clearing of the PowerShell history, and unusual command-line entries. The droplet used in each experiment was created from the same snapshot of a baseline droplet and contained two months of data gathered before the live experiments. The baseline data had only normal operating system functions and researcher interactions with the baseline droplet logged.

Dropped packets were a problem in some experiments involving attacks when a SYN packet left the attacking machine but never arrived at the victim machine. We troubleshot the EternalBlue MSF module by capturing packets on both the attacker and victim machine with Wireshark. We verified that the firewalls on DigitalOcean, the attacking machine, the Linux side of the victim machine, and the Windows side of the victim machine were all open on port 445. We asked DigitalOcean support to check whether their system automatically flagged the traffic as malicious and blocked it, but they said that while their system does have some security protections, it would not have automatically stopped this attack (DigitalOcean Support, personal communication, November 6, 2023). We could not determine why the SYN packet did not reach the victim machine.

A. EXPERIMENT 1 RESULTS

Experiment 1 ran from November 1, 2023, to November 5, 2023. The user-interface machine's IPv4 address was not advertised anywhere. 516,309 connections were made on

port 3389 (RDP), 5,636 connections on port 22 (SSH), and 17 connections on port 5800 (VNC). 801 unique IPv4 addresses allegedly came from 54 different countries. The top three countries in unique IPv4 addresses connecting to the user-interface machine were the United States, South Korea, and China. The top three countries for the number of bytes of data exchanged were Panama, The Netherlands, and Ukraine. 868,325 login attempts were made with 23 successful logins on the standard user account without administrator privileges. Splunk found anomalies in the form of unusual command-line entries and masquerade attacks.

1. Living-off-the-Land Attacks

We saw some evidence of living-off-the-land attacks. These require a login to the target machine either physically or through remote access such as the RDP protocol. Then the attacks look for vulnerable binaries associated with Windows accessibility features such as:

- Sticky Keys: C:\Windows\System32\sethc.exe
- Accessibility Menu: C:\Windows\System32\utilman.exe
- On-Screen Keyboard: C:\Windows\System32\osk.exe
- Magnifier: C:\Windows\System32\Magnify.exe
- Narrator: C:\Windows\System32\Narrator.exe
- Display Switcher: C:\Windows\System32\DisplaySwitch.exe
- App Switcher: C:\Windows\System32\AtBroker.exe

The vulnerable binary is then replaced with the cmd.exe binary so that, when the accessibility feature is launched, a command prompt with system credentials will open instead. During normal system use, Windows will automatically run AtBroker.exe and sethc.exe; however, the other five features require manual input to run. On 4 November 2023, five of the seven accessibility features were detected by Splunk, as shown in Figure 6.

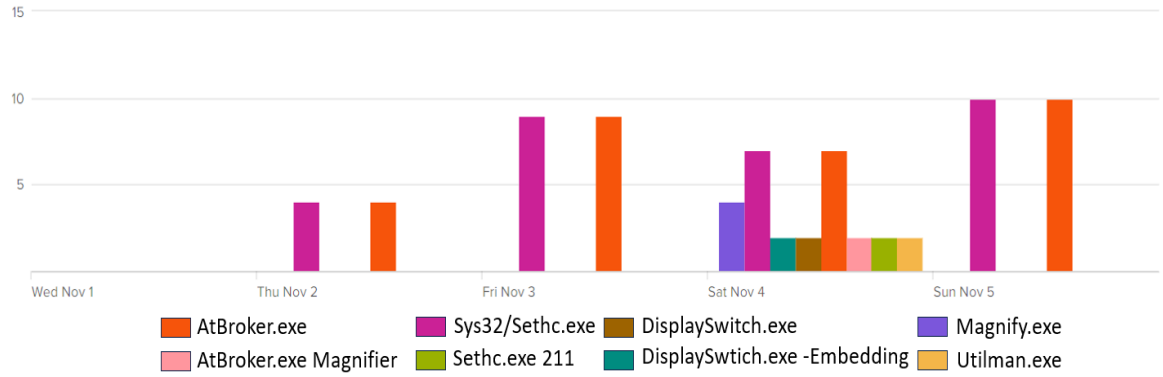


Figure 6. Experiment 1 – Frequency of Windows accessibility features executed per day.

In the two months of data preceding the live experiment, there were no occurrences of more than three of the seven Windows accessibility binaries running within three seconds of one another. AtBroker.exe and Sys32/Sethc.exe were occasionally run by the operating system and did not indicate a LotL attack. A key indicator in identifying potential LotL attacks is the "sethc.exe 211" command used to invoke the "sticky keys" Windows accessibility feature and open a command prompt. These attacks were tried twice on November 4, 2023, once at 3:41 am and again at 7:30 pm, each attack resulted in seven binaries running in a short period. Binary-execution sequences happening within a two-second window do not appear anywhere else in the baseline data we collected before the experiments (Figure 7).

2023-11-04 03:41:38	atbroker.exe /start magnifierpane
2023-11-04 03:41:38	C:\Windows\System32\DisplaySwitch.exe -Embedding
2023-11-04 03:41:38	sethc.exe 211
2023-11-04 03:41:38	utilman.exe /debug
2023-11-04 03:41:38	DisplaySwitch.exe
2023-11-04 03:41:39	"C:\Windows\System32\Magnify.exe"
2023-11-04 03:41:39	"C:\Windows\System32\Magnify.exe"
2023-11-04 19:30:15	DisplaySwitch.exe
2023-11-04 19:30:16	atbroker.exe /start magnifierpane
2023-11-04 19:30:16	sethc.exe 211
2023-11-04 19:30:16	C:\Windows\System32\DisplaySwitch.exe -Embedding
2023-11-04 19:30:16	utilman.exe /debug
2023-11-04 19:30:17	"C:\Windows\System32\Magnify.exe"
2023-11-04 19:30:18	"C:\Windows\System32\Magnify.exe"

Figure 7. Experiment 1 – Binary-execution sequences of potential LotL attacks.

During each attack only a single IPv4 address connected to the user-interface machine. The addresses used in the two attacks were different and were from different countries. Although this type of cyberattack elevates user privileges and provides access to an administrator account, the Splunk query to detect administrator access indicated no logins to the user-interface machine as an administrator. Similarly, no significant increases in bandwidth or bytes of outgoing data followed indicating no exfiltration. We concluded that this attack was unsuccessful because our version of Windows 10 was patched against this cyberattack.

2. Masquerading Attacks

A masquerading attack is one in which an operating-system process appears to be running normally but has been altered. Our query to detect this type of cyberattack checked whether specific processes were using the default process path assigned by the operating system. For instance, the svchost.exe program in Windows allows services to share resources. We observed that svchost.exe normally ran about 100 times per hour. Between

8 PM and 9 PM on 4 November 2023, 20,986 instances of svchost.exe occurred, and for the next 16 hours, between 1,500 and 19,000 instances of svchost.exe occurred per hour, well above the baseline (Figure 8). Splunk detected that the log path for svchost.exe was not the one defined by the operating system. Executions of svchost.exe are logged in WinEventLog:Security by default.

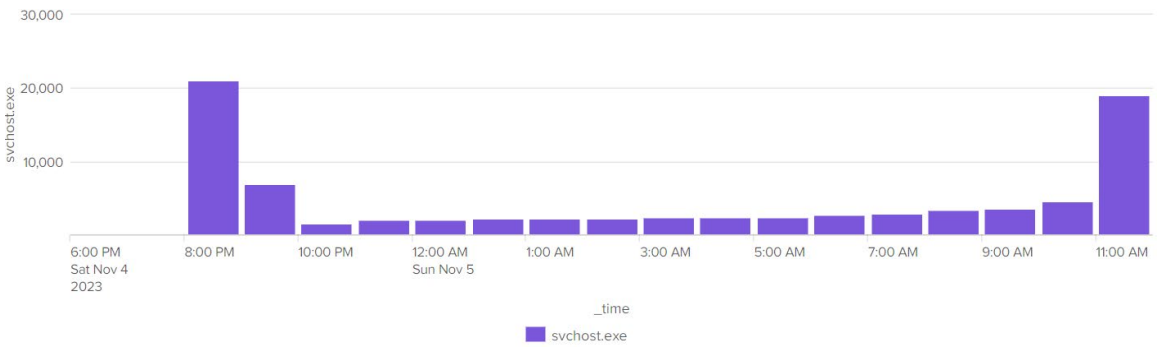


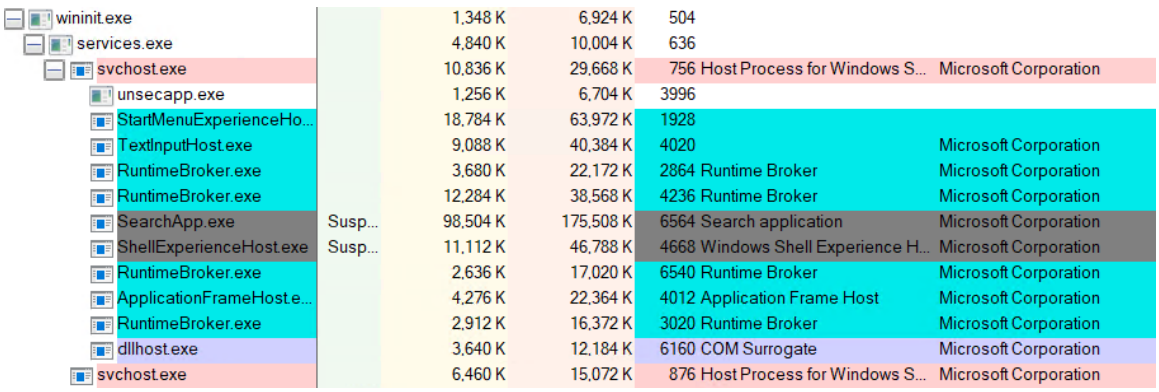
Figure 8. Experiment 1 – Abnormal patterns with svchost.exe displayed by Splunk query.

At 8:07 pm on November 4, 2023, an execution of svchost.exe was recorded in the XmlWinEventLog:Security log, and an event code 4689 was logged in WinEventLog:Security. At 8:11 pm on November 4, 2023, svchost.exe was logged in XmlWinEventLog:Microsoft-Windows-Sysmon/Operational but no event code was recorded in the Windows security log. A possible explanation for this auditing difference is the installation or modification to sysmon.exe (the Windows System Monitor). This was ruled out as sysmon.exe was already installed and was unmodified during the experiment. Only two other executables ran in the hour after svchost.exe was changed: MoUsoCoreWorker.exe and taskhostw.exe, both typical system processes related to Windows updates and the dynamic link library, respectively.

We compared the log data for a suspicious svhost.exe executions from the XML (extensible markup language) WinEventLog:Microsoft-Windows-Sysmon/Operational logs ingested by Splunk on November 4 to the log data for a standard svchost.exe execution from the WinEventLog:Security log (Appendix B). The Microsoft-Windows-Sysmon/ Operational log specifies the protocols with which svchost.exe is run. The log data

(Appendix B) shows that svchost.exe was using the RDP and TCP protocols. When the first svchost.exe log entry was created in the WinEventLog:Microsoft-Windows-Sysmon/ Operational log, three RDP connections were established with the user-interface machine. All three IPv4 addresses scanned and did not interact with svchost.exe. So they were not suspicious.

On the Windows virtual machine used in Experiment 1, we installed Sysinternals Suite, utility programs that give additional technical information. We compared Experiment 1's Windows machine to the Windows machine on a clean user-interface machine that had not connected to the Internet. Comparing processes running at startup showed no difference between the two machines. However, we noticed two differences between the machines. Figure 9 shows the processes running on the clean Windows machine and Figure 10 shows the processes running on Experiment 1's Windows machine. MoUsoCoreWorker.exe and a second dllhost.exe were running under the primary svchost.exe binary; neither had a digital signature to verify its legitimacy, and that is suspicious.



wininit.exe		1,348 K	6,924 K	504		
services.exe		4,840 K	10,004 K	636		
svchost.exe		10,836 K	29,668 K	756	Host Process for Windows S...	Microsoft Corporation
unsecapp.exe		1,256 K	6,704 K	3996		
StartMenuExperienceHo...		18,784 K	63,972 K	1928		
TextInputHost.exe		9,088 K	40,384 K	4020		Microsoft Corporation
RuntimeBroker.exe		3,680 K	22,172 K	2864	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		12,284 K	38,568 K	4236	Runtime Broker	Microsoft Corporation
SearchApp.exe	Susp...	98,504 K	175,508 K	6564	Search application	Microsoft Corporation
ShellExperienceHost.exe	Susp...	11,112 K	46,788 K	4668	Windows Shell Experience H...	Microsoft Corporation
RuntimeBroker.exe		2,636 K	17,020 K	6540	Runtime Broker	Microsoft Corporation
ApplicationFrameHost.e...		4,276 K	22,364 K	4012	Application Frame Host	Microsoft Corporation
RuntimeBroker.exe		2,912 K	16,372 K	3020	Runtime Broker	Microsoft Corporation
dllhost.exe		3,640 K	12,184 K	6160	COM Surrogate	Microsoft Corporation
svchost.exe		6,460 K	15,072 K	876	Host Process for Windows S...	Microsoft Corporation

Figure 9. Experiment 1 – Process Explorer display on clean Windows machine.

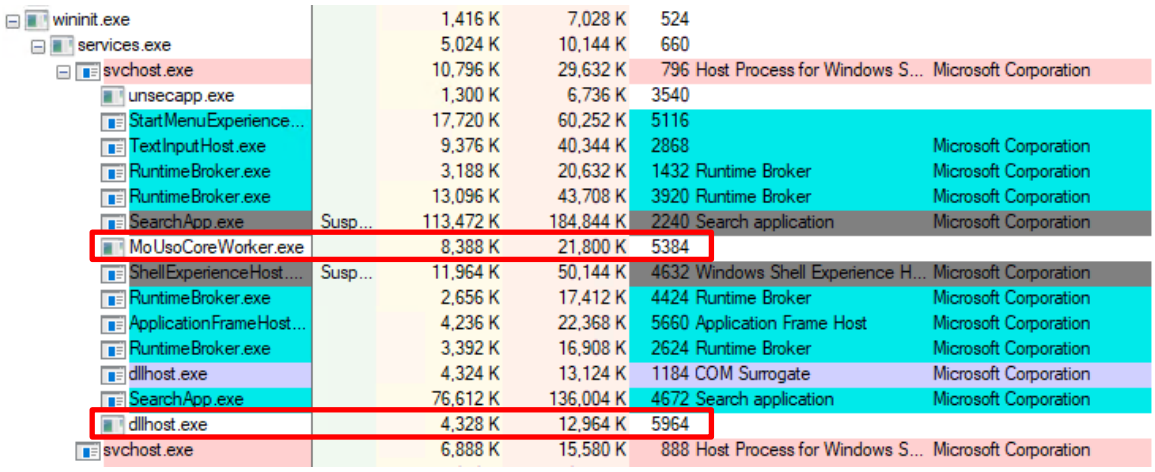


Figure 10. Experiment 1– Process Explorer display on Experiment 1's Windows machine.

B. EXPERIMENT 2 RESULTS

Experiment 2 ran from November 10, 2023, to November 14, 2023. The user-interface machine's address was not advertised anywhere. 701,992 connections were made on port 3389 (RDP), 20877 connections on port 22 (SSH), and 18 connections on port 5,800 (VNC). 1,801 unique IPv4 addresses came from 70 different countries. The top three countries with unique addresses were Turkey, the United States, and China. The top three countries for the number of bytes of data sent were Panama, Monaco, and The Netherlands. 1,366,657 login attempts were made with 10 successful logins on the standard user account without administrator privileges.

1. Dictionary Attack

Data collected for Experiment 2 had 57.4% more login attempts and 63.7% more unique usernames compared to Experiment 1. These tried many usernames and are called "dictionary" attacks. 28,645 unique usernames were attempted during Experiment 1 and 46,898 unique usernames were attempted during Experiment 2. The usernames in Experiment 2 were more advanced than those in Experiment 1 and contained not just common first and last names, but combinations of first and last names. "WHATUPTIME.COM," one of the top five usernames seen in Meier, was attempted 87 times, indicating the attacker was aware of our association with DigitalOcean (Meier,

2022). “UtilityAdmin,” the administrator account on the Windows user-interface machine, was the second most attempted login name. Since that administrator account was not advertised when a user connected to the machine on port 3389, it is unclear how attackers found that username.

We used Splunk to identify addresses guessing usernames with RDP. Guessing was logged in the WinEventLog:Security and XmlWinEventLog:Security logs. The latter security log records the protocol and address making the request; however, due to the configuration of the user-interface machine on which Windows ran as a guest operating system and used the address of the Linux host, the logs recorded all addresses as 10.0.0.0, which was not helpful. Similarly, Splunk can determine which addresses sent the most bytes to the user-interface machine, and when spikes in login attempts occurred; but RDP data is encrypted and does not show which addresses tried to login with which username.

2. Living-off-the-Land Attacks

Fewer Windows accessibility binaries launched in Experiment 2 compared to Experiment 1; however, we detected the same suspicious pattern (Figure 11). We used Sysinternals Suite Process Explorer to view all active binaries running on the post-experiment Experiment 2 droplet and compared them to binaries running on a clean droplet that had not connected to the Internet. We identified some unsigned binaries running on the Experiment 2 droplet; however, they matched the unsigned binaries running on the clean droplet indicating that the Experiment 2 binaries were unmodified. The first alert condition set to detect living-off-the-land attacks checked if one of the seven binaries associated with Windows accessibility ran (listed in Section V.A.1). This produced many false positives. The conditions were changed to require more than five of the binaries (including multiples) running to trigger the alert. This reduced false positive results to zero for the rest of Experiment 2. When the modified query was run, the only result produced was the event with seven binary executions shown in Figure 11.

```
2023-11-08 17:22:18    atbroker.exe /start magnifierpane
2023-11-08 17:22:18    sethc.exe 211
2023-11-08 17:22:18    utilman.exe /debug
2023-11-08 17:22:18    C:\Windows\System32\DisplaySwitch.exe -Embedding
2023-11-08 17:22:18    DisplaySwitch.exe
2023-11-08 17:22:19    "C:\Windows\System32\Magnify.exe"
2023-11-08 17:22:20    "C:\Windows\System32\Magnify.exe"
```

Figure 11. Experiment 2 – Binary-execution sequence of potential LotL attacks.

C. EXPERIMENT 3 RESULTS

Experiment 3 ran from November 13, 2023, to November 16, 2023. The user-interface machine's IPv4 address was not advertised anywhere. 23,239 connections were made on port 3389 (RDP), 4,673 connections on port 80 (HTTP), 3,936 connections on port 22 (SSH), and 9 connections on port 5,800 (VNC). 712 unique IPv4 addresses occurred from 49 different countries. 44,865 login attempts were made with 5 successful logins on the standard user account without administrator privileges. We observed the Splunk Stream's network monitoring for 48 hours and noticed less network traffic than in the previous two experiments. Table 3 shows the difference in port connections after 48 hours for the first three experiments.

Table 3. Experiment 3 – Connection differences among experiments in first 48 hours.

Port	Exp 1	Exp 2	Exp 3	Exp 1 Δ Exp 3	Exp 2 Δ Exp 3
22	1,978	11,584	2,718	+37.41%	-76.53%
3389	343,276	219,766	18,548	-94.59%	-91.56%
5800	9	11	8	-11.11%	-27.27%

The user-interface machine used in Experiment 3 was tagged as a honeypot by Shodan before Experiment 3 started. It is inconclusive whether this affected port 22 (SSH) connections, though it appeared to reduce port 3389 (RDP) connections. The standard analysis queries did not return any results.

D. EXPERIMENT 4 RESULTS

Experiment 4 ran from November 17, 2023, to November 21, 2023. The user-interface machine IPv4 address was not advertised anywhere. 152,065 connections were made on port 3389 (RDP), 24,476 connections on port 22 (SSH), 15,754 connections on port 80 (HTTP) from 141 unique IPv4 addresses, and 19 connections on port 5,800 (VNC). 4,458 unique IPv4 addresses occurred from 89 different countries. The top three countries connecting to the user-interface machine were the United States, United Kingdom, and China. The top three countries for the number of bytes of data sent were the United States, South Korea, and Luxembourg. 302,026 login attempts were made with 6 successful logins on the standard user account without administrator privileges.

The standard analysis queries did not return any results. Experiment 4 ran the Flask Web server and was open on port 80. We ran a query to list the Web pages that external addresses tried to access on the Web server. Multiple source addresses tried to connect to sitemap.xml and robots.txt to acquire metadata about the website. Others tried to access user and administrator login pages.

One address made 82 URL requests in under 70 seconds, most with .env extensions. An .env file contains lists of environmental variables used in development and can hold information such as applications-programming interface tokens, passwords, or database logins (Microsoft, 2021). Some botnets have scanned the Internet looking for insecure .env files to exploit in the past few years (Cimpanu, 2020).

Most Web-page attempts contained the address of our droplet and the site they were trying to reach. We identified two which tried to connect to 127.0.0.1:80/shell. 127.0.0.1 is a non-routable IPv4 address reserved for loopback connections on the local host. We isolated the source address in the Wireshark data and found that it sent an HTTP GET method requesting our droplet download data from a website (Figure 12). The attack sent

a “GET /shell?cd+/tmp;rm+-rf+*;wget+http://%s/jaws;sh+/tmp/jaws” request where the “%s” was replaced by a malicious URL containing the botnet code. The Flask webserver responded with a “404 Not Found” error. We used a client URL command and tried to download the contents of the malicious URL’s Web page in the packet into a text file to prevent execution; however, the website had already been moved. Within the malicious URL was the word “jaws” which suggested that this was part of the “JAWS Webserver unauthenticated shell command execution” botnet, a variant of the Mirai botnet (Nigam, 2018). JAWS targets Internet-of-things devices. The GET method tells a Linux victim machine to open a shell, download a malicious file from a website, and run the program. A Metasploit Framework module was developed by Paul Davies, Andrew Tierney, and B. Coles to test devices for this vulnerability (Metasploit, 2017). The linux/http/mvpower_dvr_shell_exec module in MSF tries to make the same connection as the JAWS botnet and then tries to open a shell; if the module detects that a shell was opened, it warns the user (Metasploit, 2017). Running the MSF module on Experiment 4 droplet returned a “target is not vulnerable” result.

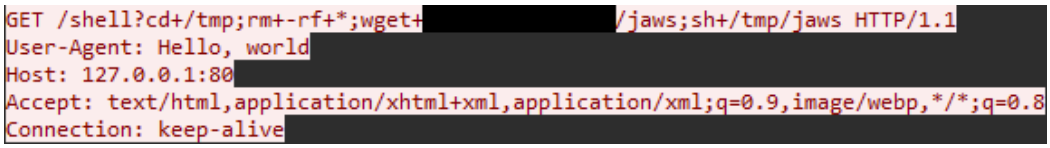


Figure 12. Experiment 4 – HTTP GET method used by a botnet.

E. SPLUNK EVALUATION

Splunk enables monitoring and searching log and network data across a network to produce visualizations showing patterns and potential anomalies. It can also search in real-time, which can be automated as alerts, can notify users by email, and act immediately by launching a script if the alerts are triggered. With advanced configuration, Splunk can act as an advanced intrusion-detection and prevention mechanism. However, detecting intrusions in real time requires advanced knowledge of network engineering and many changes to the default Splunk configuration. Splunk only searches for user-defined conditions in the ingested data; it will not suggest or look for patterns without being told.

For each experiment, we expected Splunk could process the collected data and provide threat intelligence to identify and respond to potentially malicious activity in real time. However, Splunk is only effective when the indexer can ingest correct data and when the correct queries are run. There is a steep learning curve for those who wish to use the Search Processing Language; however, once the syntactical knowledge is gained, it can produce correlated tables and visualizations allowing researchers to see changes in baselines and identify potentially malicious activity.

Using several queries, we could detect the potential living-off-the-land and masquerade attacks that likely would not have been identified with previous logging methods. The attempts to gather .env data could have been detected with the logging methods used in previous GridPot experiments, such as NXLogs, though Splunk can automate searches and send alerts if it detects certain types of attacks. Between Splunk Inc. and the active user community, new queries are being generated frequently to identify the latest threats.

VI. CONCLUSION AND FUTURE WORK

A. SUMMARY OF FINDINGS

Splunk improved the detection and logging methods of our honeypot over the logging methods used in previous GridPot experiments, and we could detect potential threats that would not have been noticed.

In Experiment 1 we used Splunk to quickly inspect and sort the command prompts logged by Windows. To manually check the thousands of event logs in Windows would have taken considerable time. We analyzed Windows accessibility binaries found in the command-prompt logs indicating that a LotL attack likely occurred. With Splunk, we could see that the svchost.exe binary changed its log path and found which log recorded the event. It is unlikely that we would have detected these changes without Splunk queries.

Experiment 2 logged the largest unique number of username login attempts, indicating an advanced dictionary attack. In Experiment 1 we saw brute-force tactics that included common last names and a first initial. Experiment 2 saw common and uncommon names and first-last name combinations. We also revised the living-off-the-land attack detection to reduce false positives.

Although we did not gather much data during Experiment 3, we saw the effect of the random address assigned to our droplets by DigitalOcean. We observed that an address tagged as a honeypot in Shodan received over 90% less traffic than addresses that are not tagged.

In Experiment 4, we did not detect any new attacks against Windows, but we did observe two botnets. One botnet tried to collect data by exploiting insecure .env files, and one tried remote code execution. This information will help us develop additional queries for future HTTP Web-server deployments.

B. FUTURE WORK

The Splunk Security Essentials application within the Splunk Enterprise software suggests six steps to provide advanced network protection, and intrusion detection and

prevention (Splunk, n.d.-a). With implementation and initial configuration, we have only met the prerequisites for stage one, data collection. The subsequent stages are normalization, expansion, enrichment, automation and orchestration, and advanced detection. Each stage adds more applications and data sources and refines the queries on that data. More research should be done to implement the later steps (Splunk, n.d.-a) which can enable future researchers to better detect and respond to threats.

With the data gathered in our experiments, new queries and alerts should be implemented to detect additional malicious conditions. Due to the limitations of the logging droplet, the indexer can only process 6–8 searches at a time. Deploying Splunk Enterprise on a droplet with more resources will permit more real-time alerts. Additional Splunk alerts should be written to detect connections from countries with known malicious actors and when .env or shell URL requests occur.

When Windows runs as a guest operating system on the Linux host, it uses a virtual ethernet 0 connection to access the Internet. This configuration causes Windows to log all source IPv4 addresses as a 10.0.0.* private address, which prevents Splunk from correlating malicious events to source addresses. This limitation could be corrected by either giving the Windows virtual machine its own IPv4 address, or by running Windows natively on the droplet.

The benefit of using SIEM technology on honeypots can be further explored by implementing the other deployment models described in Chapter IV. Capturing data from different regions could reveal different attack methods and patterns such as the time zone of the attacks and the use of IPv4 addresses associated with different geographic locations.

APPENDIX A. NESSUS SCAN OF OUR HONEYPOT

Nessus Scanner looks for vulnerabilities on the targeted system and produces a report of the vulnerabilities found. We used Kali Linux for a Nessus basic host scan with default setting on our user-interface machine. Below are the results of that scan.

Vulnerabilities Total: 35

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
HIGH	7.5	6.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
MEDIUM	6.5	-	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	-	57582	SSL Self-Signed Certificate
MEDIUM	6.5	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	-	157288	TLS Version 1.1 Protocol Deprecated
LOW	2.6*	-	10407	X Server Detection
INFO	N/A	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	11002	DNS Server Detection
INFO	N/A	-	54615	Device Type
INFO	N/A	-	11219	Nessus SYN scanner
INFO	N/A	-	19506	Nessus Scan Information
INFO	N/A	-	11936	OS Identification
INFO	N/A	-	117886	OS Security Patch Assessment Not Available
INFO	N/A	-	181418	OpenSSH Detection
INFO	N/A	-	10940	Remote Desktop Protocol Service Detection
INFO	N/A	-	70657	SSH Algorithms and Languages Supported
INFO	N/A	-	149334	SSH Password Authentication Accepted
INFO	N/A	-	10881	SSH Protocol Versions Supported

INFO	N/A	-	153588	SSH SHA-1 HMAC Algorithms Enabled
INFO	N/A	-	10267	SSH Server Type and Version Information
INFO	N/A	-	56984	SSL / TLS Versions Supported
INFO	N/A	-	10863	SSL Certificate Information
INFO	N/A	-	70544	SSL Cipher Block Chaining Cipher Suites Supported
INFO	N/A	-	21643	SSL Cipher Suites Supported
INFO	N/A	-	156899	SSL/TLS Recommended Cipher Suites
INFO	N/A	-	22964	Service Detection
INFO	N/A	-	121010	TLS Version 1.1 Protocol Detection
INFO	N/A	-	136318	TLS Version 1.2 Protocol Detection
INFO	N/A	-	110723	Target Credential Status by Authentication Protocol - No Credentials Provided
INFO	N/A	-	64814	Terminal Services Use SSL/TLS
INFO	N/A	-	10287	Traceroute Information
INFO	N/A	-	19288	VNC Server Security Type Detection
INFO	N/A	-	65792	VNC Server Unencrypted Communication Detection
INFO	N/A	-	10342	VNC Software Detection

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. WINDOWS XML LOG COMPARISON

The table below shows two log entries of an svchost.exe execution. The left column shows the log data for a WinEventLog:Security log event, and the right column shows the log data for a WinEventLog:Microsoft-Windows-Sysmon/Operational log event. These log entries contain slightly different data; rows with data in both columns indicate similar data were logged, and rows with data in only one column indicate the specific data was not logged in the other log. Only the data in the right column has protocol and network data. When Splunk detected that logging was switched from the Security log to the Sysmon/ Operational log, it alerted researchers to the issue.

Standard Svchost.exe Log Source: XmlWinEventLog:Security	**Modified Svchost.exe Log Source: XmlWinEventLog:Microsoft-Windows-Sysmon/Operational**
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'>	<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'>
<System>	<System>
<Provider Name='Microsoft-Windows-Security-Auditing' Guid='{54849625-5478-4994-a5ba-3e3b0328c30d}'/>	<Provider Name='Microsoft-Windows-Sysmon' Guid='{5770385f-c22a-43e0-bf4c-06f5698ffbd9}'/>
<EventID>4689</EventID>	<EventID>3</EventID>
<Version>0</Version>	<Version>5</Version>
<Level>0</Level>	<Level>4</Level>
<Task>13313</Task>	<Task>3</Task>
<Opcode>0</Opcode>	<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>	<Keywords>0x8000000000000000</Keywords>

Standard Svchost.exe Log Source: XmlWinEventLog:Security	**Modified Svchost.exe Log Source: XmlWinEventLog:Microsoft-Windows-Sysmon/Operational**
<TimeCreated SystemTime='2023-11-05T03:07:24.7762176Z'/>	<TimeCreated SystemTime='2023-11-05T03:11:01.7476040Z'/>
<EventRecordID>7625832</EventRecordID>	<EventRecordID>1249307</EventRecordID>
<Correlation/>	<Correlation/>
<Execution ProcessID='4' ThreadID='49408'/>	<Execution ProcessID='3432' ThreadID='3900'/>
<Channel>Security</Channel>	<Channel>Microsoft-Windows-Sysmon/Operational</Channel>
<Computer>SGEUtilities</Computer>	<Computer>SGEUtilities</Computer>
<Security/>	<Security UserID='S-1-5-18'/></System><EventData>
	<Data Name='RuleName'>RDP</Data>
	<Data Name='UtcTime'>2023-11-05 03:10:48.801</Data>
	<Data Name='ProcessGuid'>{b217f8a7-8f24-6542-1500-000000002900}</Data>
<Data Name='ProcessId'>0x73fc</Data>	<Data Name='ProcessId'>420</Data>
<Data Name='ProcessName'>C:\Windows\System32\svchost.exe</Data>	<Data Name='Image'>C:\Windows\System32\svchost.exe</Data>
<Data Name='SubjectUserSid'>NT AUTHORITY\SYSTEM</Data>	<Data Name='User'>NT AUTHORITY\NETWORK SERVICE</Data>
	<Data Name='Protocol'>tcp</Data>
	<Data Name='Initiated'>false</Data>

Standard Svchost.exe Log Source: XmlWinEventLog:Security	**Modified Svchost.exe Log Source: XmlWinEventLog:Microsoft-Windows-Sysmon/Operational**
	<Data Name='SourceIsIpv6'>false</Data>
	<Data Name='SourceIp'>10.0.2.2</Data>
	<Data Name='SourceHostname'>-</Data>
	<Data Name='SourcePort'>12511</Data>
	<Data Name='SourcePortName'>-</Data>
	<Data Name='DestinationIsIpv6'>false</Data>
	<Data Name='DestinationIp'>10.0.2.15</Data>
	<Data Name='DestinationHostname'>SGEUtilities</Data>
	<Data Name='DestinationPort'>3389</Data>
	<Data Name='DestinationPortName'>ms-wbt-server</Data>
	</EventData></Event>
</System>	
<EventData>	
<Data Name='SubjectUserName'>SGEUTILITIES$</Data>	
<Data Name='SubjectDomainName'>WORKGROUP</Data>	

Standard Svchost.exe Log Source: XmlWinEventLog:Security	**Modified Svchost.exe Log Source: XmlWinEventLog:Microsoft-Windows-Sysmon/Operational**
<Data Name='SubjectLogonId'>0x3e7</Data>	
<Data Name='Status'>0x0</Data>	
</EventData></Event>	
host = SGEUTILITIES	host = SGEUTILITIES
source = XmlWinEventLog:Security	source = XmlWinEventLog:Microsoft-Windows-Sysmon/Operational

LIST OF REFERENCES

Alladi, T., Chamola, V., & Zeadally, S. (2020). Industrial control systems: Cyberattack trends and countermeasures. *Computer Communications*, *155*, 1–8. https://doi.org/10.1016/j.comcom.2020.03.007

Antonioli, D., Agrawal, A., & Tippenhauer, N. O. (2016). Towards high-interaction virtual ICS honeypots-in-a-box. *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. https://doi.org/10.1145/2994487.2994493

ArcSight (n.d.). ArcSight Enterprise Security Manager (ESM). https://www.microfocus.com/en-us/cyberres/secops/arcsight-esm

Bhatt, S., Manadhata, P. K., & Zomlot, L. (2014). The operational role of security information and event management systems. *IEEE Security & Privacy*, *12*(5), 35–41. https://doi.org/10.1109/msp.2014.103

Bieker, M. & Pilkington, D. (2020). Deploying an ICS honeypot in a cloud computing environment and comparatively analyzing results against physical network deployment. [Master's thesis, Naval Postgraduate School]. NPS Archive: Calhoun. http://hdl.handle.net/10945/66586

Briffaut, J., Clemente, P., Lalande, J. F., & Rouzaud-Cornabas, J. (2012). Honeypot forensics for system and network SIEM design. *Advances in Security Information Management: Perceptions and Outcomes*. https://www.researchgate.net/profile/Clemente-Patrice/publication/236216116_Honeypot_forensics_for_system_and_network_SIEM_design/links/5e4fd2be458515072dad8907/Honeypot-forensics-for-system-and-network-SIEM-design.pdf

Cimpanu, Catalin. (2020). Botnets have been silently mass-scanning the internet for unsecure ENV files. *ZDnet.* https://www.zdnet.com/article/botnets-have-been-silently-mass-scanning-the-internet-for-unsecured-env-files/

Cisco (n.d.). Cisco IOS NetFlow. https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

Climek, D., Macera, A., & Tirenin, W. (2016, March 8). Cyber deception. *CSIAC*. https://csiac.org/articles/cyber-deception/

Davis, B. (2003). Second generation honeynet honeywall. *Global Information Assurance Certification Paper – SANS Institute*. https://www.giac.org/paper/gsec/3173/second-generation-honeynet-honeywall/102801

DigitalOcean (n.d.-a). Droplets. https://www.digitalocean.com/products/droplets

DigitalOcean (n.d.-b). Netmaker. https://docs.digitalocean.com/products/marketplace/catalog/netmaker/

Dougherty, J. (2020). Evasion of honeypot detection mechanisms through improved interactivity of ICS-based systems [Master's thesis, Naval Postgraduate School]. NPS Archive: Calhoun. http://hdl.handle.net/10945/66065

Elastic (n.d.). Elastic Stack. https://www.elastic.co/elastic-stack

Enscada (2021). IndigoSCADA [Computer software]. https://github.com/enscada/IndigoSCADA

Exabeam. (n.d.). Exabeam SIEM. https://www.exabeam.com/

Ferguson-Walter, Kimberly, Shade, Temmie, Rogers, Andrew, Combs, Angela, Divis, Kristin Marie, Nauer, Kevin S., Jones, Aaron, Trumbo, Michael Christopher Stefan, & Abbott, Robert G. (2018, May 5). Science of cyber deception: Experimental design and implementation. *U.S. Department of Energy Office of Scientific and Technical Information Conference*. https://www.osti.gov/biblio/1648653

Franco, J., Aris, A., Canberk, B., & Uluagac, A. S. (2021). A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *ArXiv:2108.02287* [Cs]. http://arxiv.org/abs/2108.02287

Ganguly, N. (2007). Copyleft: An alternative to copyright in computer software and beyond. *NIScPR Online Periodical Repository: Home.* http://nopr.niscpr.res.in/handle/123456789/248

Gartner (2023). Security information and event management (SIEM) reviews and ratings. https://www.gartner.com/reviews/market/security-information-event-management

Garton, David. (2019). Purdue Model Framework for industrial control systems & cybersecurity segmentation. *Working Document of the NPC Study Dynamic Delivery – America's Evolving Oil and Natural Gas Transportation* Infrastructure. https://www.energy.gov/sites/default/files/2022-10/Infra_Topic_Paper_4-14_FINAL.pdf

González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures. *Sensors*, *21*(14), 4759. https://doi.org/10.3390/s21144759

Green, B., Krotofil, M., & Hutchison, D. (2016). Achieving ICS resilience and security through granular data flow management. *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy.* https://doi.org/10.1145/2994487.2994498

IBM (n.d.). IBM Security QRadar Suite. https://www.ibm.com/qradar

Krawetz, N. (2004). Anti-honeypot technology. *IEEE Security & Privacy Magazine*, *2*(1), 76–79. https://doi.org/10.1109/msecp.2004.1264861

Liu, Z., Chen, C., Zhang, L. Y., & Gao, S. (2022). Working mechanism of Eternalblue and its application in Ransomworm. *Cyberspace Safety and Security*, 178–191. https://doi.org/10.1007/978-3-031-18067-5_13

Lockheed Martin (n.d.). The cyber kill chain.https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

LogRhythm (n.d.). LogRhythm security information and event management (SIEM) Solutions. https://logrhythm.com/

López-Morales, E., Rubio-Medrano, C., Doupé, A., Shoshitaishvili, Y., Wang, R., Bao, T., & Ahn, G.-J. (2020). HoneyPLC: A next-generation honeypot for industrial control systems. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. https://doi.org/10.1145/3372297.3423356

Lyon, Gordon. (2023). NMAP. *NMAP Software LLC.* https://www.nmap.org

Meier, Joseph T. (2022). Hardening Windows-based honeypots to protect collected data. [Master's thesis, Naval Postgraduate School]. NPS Archive: Calhoun. http://hdl.handle.net/10945/70752

Metasploit (2017). MVPower DVR TV-7104HE 1.8.4 115215B9 – Shell command execution (Metasploit). https://www.exploit-db.com/exploits/41471

Microsoft (2018). Group policy objects. https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-objects

Microsoft (2021). Environmental variables. https://learn.microsoft.com/en-us/windows/win32/procthread/environment-variables

Microsoft (n.d.). Microsoft Sentinel. https://www.microsoft.com/en-us/security/business/siem-and-xdr/microsoft-sentinel

MITRE (n.d.). MITRE ATT&CK. https://www.attack.mitre.org

Mohd Ariffin, M. A., Darus, M. Y., Haron, H., Kurniawan, A., Muliono, Y., & Pardomuan, C. R. (2022). Deployment of honeypot and SIEM tools for cyber security education model in UiTM. *International Journal of Emerging Technologies in Learning (iJET)*, *17*(20), 149–172. https://doi.org/10.3991/ijet.v17i20.32901

Nigam, Ruchna. (2018). Unit 42 finds new Mirai and Gafgyt IoT/Linux botnet campaigns. *Unit 42, Palo Alto Networks.* https://unit42.paloaltonetworks.com/unit42-finds-new-mirai-gafgyt-iotlinux-botnet-campaigns/

Ongun, T., Stokes, J. W., Or, J. B., Tian, K., Tajaddodianfar, F., Neil, J., Seifert, C., Oprea, A., & Platt, J. C. (2021). Living-off-the-land command detection using active learning. *24th International Symposium on Research in Attacks, Intrusions and Defenses*. https://doi.org/10.1145/3471621.3471858

OSSEC (n.d.). Open source HIDS SECurity. https://www.ossec.net/

Rapid7 (n.d.) Rapid7 SIEM solution InsightIDR. https://www.rapid7.com/products/insightidr/

Serpanos, D., Komninos, T. (2022). The cyberwarfare in Ukraine. *Computer*, 55(7), 88–91. https://doi.org/10.1109/mc.2022.3170644

Singh, A. N., & Joshi, R. C. (2011). A honeypot system for efficient capture and analysis of network attack traffic. *2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies*. https://doi.org/10.1109/icsccn.2011.6024606

Splunk (n.d.-a). Security monitoring. https://lantern.splunk.com/Security/UCE/Foundational_Visibility/Security_monitoring

Splunk (n.d.-b). Splexicon. https://docs.splunk.com/Splexicon

Splunk (n.d.-c). Splunk App for Stream. https://splunkbase.splunk.com/app/1809

Splunk (n.d.-d). Splunk Enterprise. https://www.splunk.com/

Splunk (n.d.-e). Splunk Security Essentials. https://splunkbase.splunk.com/app/3435

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., & Hahn, A. (2015). Guide to industrial control systems (ICS) security. *National Institute of Standards and Technology*. https://doi.org/10.6028/nist.sp.800-82r2

Suricata (n.d.). Suricata. https://suricata.io/

Tenable (2023) Tenable Nessus. https://www.tenable.com/products/nessus

Zanasi, C., Magnanini, F., Russo, S., & Colajanni, M. (2022). A zero trust approach for the cybersecurity of industrial control systems. *2022 IEEE 21st International Symposium on Network Computing and Applications (NCA)*. https://doi.org/10.1109/nca57778.2022.10013559

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California