Engineering Management Methodology for Assessing and Improving the Cyber Resilience of Critical Infrastructures

by Nesrine Nagui Taha

B.Sc. in Mechanical Engineering, June 1993, American University in Cairo
M.Sc. in Mechanical Engineering, February 2004, American University in Cairo
M.Phil. in Nanoscale Science and Engineering, May 2010, University of Cambridge

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

January 2023

Praxis directed by

John Kamp
Professorial Lecturer of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Nesrine Nagui Taha has passed the Final Examination for the degree of Doctor of Engineering as of December 1st, 2022. This is the final and approved form of the Praxis.

Engineering Management Methodology for Assessing and Improving the Cyber Resilience of Critical Infrastructures

Nesrine Nagui Taha

Praxis Research Committee;

John Kamp, Professorial Lecturer of Engineering Management and Systems Engineering, Praxis Director

Amir Etemadi, Associate Professor of Engineering and Applied Science, Committee Member

Michael Jones, Professorial Lecturer of Engineering Management and Systems Engineering, Committee Member

© Copyright 2022 by Nesrine Nagui Taha
All rights reserved

Dedication

Dad, I wish you would have been here to see me complete this work and attend my graduation, *I love* you and hope I make you proud, *I try…*

Mom couldn't have done this without your love and continuous support... I love you

My three amazing kids, who have pushed me along and cheered for me at every hurdle. I look forward to closing this chapter and spending evenings and weekends with you. I couldn't have completed this degree without your patience and support, without you believing in me...

My husband, I am grateful for your love, support and patience. I wouldn't have been able to do this without you.

Acknowledgement

I am very grateful to several entities and individuals for the support they provided throughout the years leading up to this work and throughout the past year when I was working on the research and writing the praxis:

The *US Army* and *US Army Corps of Engineers* and the incredible team of individuals who cheered me on and supported me throughout this journey- Thank you.

Women in Defense and *Society of Women Engineers (BK Krenzer),* thank you for your support and for believing in me. *Here's to strong women, may we know them, may we be them, may we raise them.*

Dolsy Smith, Software Developer and Librarian at the George Washington University, thank you for your time and patience in answering questions and your support in debugging my models.

Prof. John Kamp, my research advisor, thank you for your time, guidance and most of all for believing in me and for your unwavering support and patience as we saw this journey to completion!

Abstract of the Praxis

Engineering Management Methodology for Assessing and Improving the Cyber Resilience of Critical Infrastructures

The praxis investigates the concept of building cyber resilience into the system architecture of critical infrastructures through the lens of an Engineering Management tool developed for this research.

The research demonstrates that embedding well-engineered decoys, which mimic real system components, into the Informational Technology (IT) and Operational Technology (OT) systems architecture of an industrial system builds cyber resilience into the architecture thus reducing the probability of successful attacks, the impact on real system components and increasing the Mean Time to Breach. This is illustrated through the Python model of an infrastructure based on the Purdue System architecture subjected to simulated Advanced Persistent Threat-style attacks. The research demonstrates that the mean-time-to a successful breach and control over the command-and-control signals of a SCADA system is proportionate to the number and complexity of decoy components in an IT/OT environment. Also demonstrated is that the more deception built into the IT system architecture the less the impact of an attack on OT and real system components. The research tool can be used to assess the cyber resilience of an infrastructure through the use of probabilities determined through quantitative risk assessments.

Table of Contents

Acknowledgement ..... **v**
Abstract of the Praxis ..... **vi**
List of Figures ..... **ix**
List of Tables ..... **xi**
List of Symbols / Nomenclature ..... **xii**
CHAPTER I: INTRODUCTION **1**
1.1 Background 1
1.2 Research Motivation 3
1.3 Problem Statement 4
1.4 Thesis Statement 4
1.5 Research Objectives 5
1.6 Research Questions 5
1.7 Research Hypotheses 6
1.8 Scope of Research 6
1.9 Research Limitations 7
1.10 Organization of Praxis 7
2 CHAPTER II LITERATURE REVIEW **9**
2.1 Informational Technology versus Operational Technology System Architecture 10
2.2 Operational Technology Systems 11
2.3 Risk in Critical Infrastructure Systems: 13
2.4 The Purdue Model 15
2.5 The Cyber Security Triad 21
2.6 Attacks on Critical Infrastructures 22
2.7 Advanced Persistent Threat Attacks 26
2.8 Current Critical Infrastructure Defense Mechanisms 30
3 CHAPTER III - Methodology **41**
3.1 Model Description 42
3.2 Modelling attacks 46
3.3 Data Collected from Model: 52
3.4 Model Assumptions and Limitations: 54
3.5 Model Validation and Verification 55

3.6 Hypothesis Testing: 64
4 CHAPTER IV: RESULTS **66**
4.1 Edge Cases and Model Verification: 66
4.2 Base Models 71
4.3 Impact of Honeypot and Decoy Complexity on MTTB: 74
4.4 Impact of Deception on Component Access during an Attack 86
4.5 Improving Deception on the Operational Technology-side of the Architecture- Deception through the use of Identical PLCs and SCADAs: 91
4.6 Hypothesis Testing Summary 94
5 CHAPTER V DISCUSSION and CONCLUSIONS **97**
5.1 Simulation (Base Model) insights 97
5.2 Hypothesis I Testing Insights: 99
5.3 Hypothesis II Testing Insights 100
5.4 CONCLUSION 100
5.5 Future research 101
Bibliography **102**
APPENDIX A **112**
Python Model **112**

List of Figures

Figure 1. Relationship between OT and ICS systems courtesy of Securicon Information Security Solutions 11
Figure 2 SCADA System Network Architecture (Qian & Abdelwahed, 2014)..... 12
Figure 3 The Purdue Reference Architecture for ICS (Zanutto & Follis, 2017) 17
Figure 4 Cyber Security CIA Triad (Preferred IT Group, 2019) 21
Figure 5 Anatomy of Advanced Persistent Threat (Heindal Security, 2022) 27
Figure 6 MITRE Cyber Attack Life Cycle (Bodeau & Graubart, 2017)..... 28
Figure 7 Martti Lehto APT -attack model (Lehto, 2022)..... 29
Figure 8 Schematic illustrating typical location of honeypots within an IT system (Franco, Aris, Canberk, & Uluagac, 2021)..... 33
Figure 9 Whaley Deception Taxonomy (Service, 2022)..... 35
Figure 10 Schematic of Simulated Python Model..... 42
Figure 11. General Layout of SCADA System (Stouffer, Falco, & Scarfone, 2011). 44
Figure 12 Attack Tree of model. 46
Figure 13 Detailed Attack Tree of model (including probabilities) developed using Microsoft Visio..... 48
Figure 14 Code illustrating 32% divide at phishing phase of attack 51
Figure 15 Lines of code which tabulate the data and save it to Main file and Backend file. 53
Figure 16 Data obtained from Backend File versus calculated 68
Figure 17 Comparison between impact of probabilities and firewalls on the bare model..... 69
Figure 18 Classification analysis of model data – 24. Node analysis..... 70
Figure 19 Time Series Plot of Base Model by Attack Phase and overall MTTB 73
Figure 20 Impact of varying honeypot and decoy complexity as well as quantities of deceptive components on MTTB 75
Figure 21 A: One-Way ANOVA of MTTB1: HP 15%, D 50%, MTTB: HP 40%, 80% MTTB2:85%, 100% MTTB3: HP (85% D 100%) x 10 as many real system components 76
Figure 22 Games -Howell analysis..... 78
Figure 23 Interval Plot of MTTB with standard deviations used to calculate intervals 78
Figure 24 Boxplot of results of ANOVA test 79
Figure 25 Classification Decision Tree analysis of Python Model..... 81
Figure 26 Relative Variable Importance as a percentage 83
Figure 27 Regression analysis variable relative importance..... 84
Figure 28 Scatterplot of Response Fits versus Actual Values..... 85
Figure 29 Plot of R-squared versus Terminal Nodes..... 85
Figure 30 Optimal Regression Tree (CART Analysis MINITAB)- 16-Node tree..... 86

Figure 31-A: Number of Real System Components Impacted by Cyber-attacks - No deception 87
Figure 32-C: Honeypots 15% Decoys 50% and 10 times as many as real..... 89
Figure 33 Starting to get a difference at OT end (When decoys are 10 times more and at 100% complexity (0.80 0.85 and 120% real), but can we achieve this) .. 91
Figure 34 Impact of layers of deception complexity on phases of attack 92
Figure 35 Impact of deception using real components for decoys..... 93

List of Tables

Table 1 Major Cyberattacks on critical infrastructures reported between 2003- 2019 adapted from Alladi et al. (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020) **24**
Table 2 Time to detection of APT attacks (Lehto, 2022) **30**
Table 3 Generations of Honeypots (Fan, Du, Fernandez, & Villagr , 2018) **33**
Table 4 model components and probabilities treatment plant layout **45**
Table 5 Data collected from Python model **52**
Table 6 Edge Case used for model verification **56**
Table 7 Impact of Probabilities (Expected) **61**
Table 8 Base Model Components **63**
Table 9 Details of attack scenarios **64**
Table 10 Sample output of Bare model run **66**
Table 11 Counts of components **67**
Table 12 Snapshot of data **70**
Table 13 Descriptive statistics of Base Model **72**
Table 14 Summary of MTTB simulation conditions **74**
Table 15 Results of Welch's Test and Games-Howell. **77**
Table 16 Sample of Python model output used for CART analysis **82**
Table 17 Attack path classification by count and percentage of total attacks **82**
Table 18 Reduction in number of successful cyber-attacks **95**

List of Symbols / Nomenclature

CI: Critical Infrastructure

DCS: Distributed Control Systems

IT: Informational Technology

IIoT: Industrial Internet of Things

ICS: Industrial Control System(s)

OT: Operational Technology

PLC: Programmable Logic Unit

SCADA: Supervisory Control and Data Acquisition

FW: Firewall(s)

CHAPTER I: INTRODUCTION

This Chapter presents the background and motivation for this research as well as research questions and hypotheses.

1.1 Background

The Cybersecurity and Critical Infrastructure Agency of the United States identifies 16 critical infrastructure sectors (Cybersecurity and Infrastructure Security Agency, 2022), each comprised of thousands of critical facilities that provide essential services to sustain daily life operations across the Nation. Whether privately or publicly owned, over the course of the past 10 years, the number of attacks on critical infrastructures, around the world, has continued to rise significantly (Kaspersky, 2022) (Securicon, 2022) (Security, 2022). From compromising the continuity of business of an operation (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020) to taking it hostage in exchange for a ransom (John Riggi, 2022), attackers have inflicted some form of damage on the safety and/ or operations of infrastructures which provide critical services to societies.

The importance of critical infrastructures stems from the instrumental role they play in the continuity of everyday life (providing power, water, etc..) and as such threats to critical infrastructures are considered national security level threats (Reddy, 2022) making the defense of critical infrastructures against cyber threat imperative.

The vulnerability of critical infrastructures to cyber-attacks stems from technological changes within the industry which have rendered complex industrial system of systems accessible through the internet (Cai, Wang, & Yu, SCADA System Security: Complexity,

History and New Developments, 2008). Critical infrastructures are made up of various electrical, mechanical, and electronic components. These components such as pumps, motors, and boilers were designed to be in service for multiple decades, as a form of return on investment. The connection of these pieces of equipment to the internet was brought about by technological advancement in control systems rather advancement in the machinery/equipment. This misalignment between the development of industrial systems and their control systems, the emergence of the Industrial Internet of Things and the race to automate industrial components, in the absence of comprehensive cyber threat modelling, has resulted in the accessibility of industrial systems through the internet, to both operators and owners, and also to malicious actors ((Petrenko, 2019), (Dahad, 2018)).

As attackers continue to acquire more sophisticated skill and nations continue to operate industrial systems which are accessible through the world-wide-web, a need to develop cyber resilience system architectures arises (Bodeau & Graubart, Cyber Resiliency Design Principles, 2017). Cyber resilience is the ability of an organization or infrastructure to maintain business continuity in the face of a cyber-attack (Petrenko, 2019). Cyber resilience is possible through merger confidentiality, integrity, and availability (Coretelligent, 2022), and is at the core of cyber security challenges for critical infrastructures. The reason being that cyber-attacks will continue to happen (Thryft, 2022) and it no longer becomes a matter of preventing them, but rather how an entity responds to a cyber-attack (Bodeau & Graubart, Cyber Resiliency Design Principles, 2017).

Sans (Luallen, 2014), Kaspersky (Kaspersky, 2022) and Purple Secure (Security, 2022) describe phishing-type attacks as being in the top most prevalent forms of attacks on IT systems. This is of concern because most of the notorious forms of attacks are one which start as attacks on the IT-side of the architecture and make their way to the OT side of an infrastructure. The most challenging of attacks which start on the IT end of the infrastructure and result in a cyber breach on the OT-side are Advanced Persistent Threat attacks (Security, 2022) . These are attacks launched by experienced attackers that are often nation actors and often go undetected because the attackers cover their tracks to allow for prolonged dwell time with a system allowing them reconnaissance. There are other attacks which are on the rise such as Denial of Service and Man-in-the-Middle attacks. The focus of this research is on the general concept of attacks which begin in the IT layers as phishing attacks and have significant impact the OT layers. Phishing attacks on IT-systems are reported as being 32% successful (Security, 2022).

1.2 Research Motivation

Engineering managers need to ensure the cyber resilience of a critical infrastructure, and this requires the availability of a tool to help assess the health of the infrastructure and strategies that can be put into place to reduce the impact of attacks on system components and also prolong the time to a successful attack allowing managers and operators the time to act in the face of an attack, thus creating resilience in their infrastructure. Within the field of cyber security for critical infrastructures, this tool does not currently exist, hence the need for this research. The focus within the field has been on development of intrusion detection systems and there is a deficiency in methodologies to address critical

infrastructure defense in the face of the rising number of attacks on critical infrastructures.

1.3 Problem Statement

The emerging remote accessibility of industrial infrastructure control systems, with poorly understood security and system vulnerabilities, has created a pathway for the breach of operating technology systems through cyber-attacks on enterprise Informational Technology (IT) systems.

Advanced Persistent Threats are stealthy cyber-attacks in which the attacker gains unauthorized access to a system through a vulnerability in the IT side of the architecture, typically through a phishing attack, creating a foothold in the system and then remains undetected while collecting system information, such as passwords, before launching a detectible cyber-attack. Because of the sophistication and stealth involved in an APT attack, research is limited in the amount of data collected on these types of attacks. Understanding an APT attack is crucial in designing an infrastructure which is resilient towards these types of attacks. The research will focus on APT-style attacks meaning attacks that begin on the IT-side as phishing attacks and infiltrate the system gaining access to the OT-side of the infrastructure.

1.4 Thesis Statement

An engineering management tool is required to help engineering management identify the vulnerability of critical infrastructures to Advanced Persistent Threat types of attacks.

1.5 Research Objectives

The research aims to develop an engineering management tool to aid in assessing the resilience of an IT/OT system against APT type cyber-attack. Additionally, the research aims to 1) present that cyber deception is a significant tool in the defense of critical infrastructures and 2) demonstrate the need for designing decoys which resemble real system components for successful cyber defense of operational technology systems.

These needs are met through the Python model created for this research. The model simulates APT attacks on IT/OT system architecture and provides a comparison to the system response in defense scenarios with and without decoys of varying complexity.

1.6 Research Questions

The following are the research questions which are addressed:

- R_1: What are the known cyber vulnerabilities in critical infrastructure information technology (IT) and operation technology (OT) systems?
- R_2: What connects enterprise IT infrastructure and OT domain infrastructure systems?
- R_3: What changes have occurred in the OT environment that have led to changes in the interface between the enterprise IT and OT environments?
- R_4: How do critical infrastructure cyber defense systems defend against Advanced Persistent Threat attacks?

1.7 Research Hypotheses

Hypothesis 1 (H1): *Mean Time to successful cyber breach (MTTB) increases with increase in complexity of decoys in a system.*

Complexity, reflected in the model through the probability of selecting a decoy, is analogous to the sophistication with which a decoy is designed to mimic and interact as a real system component.

This hypothesis test is significant because more time which maintaining knowledge of an attacks allows for more time to respond to an attack

- **Hypothesis 2 (H_2):** *Decoys incorporated into layers of a critical infrastructure system architecture reduce the probability of successful cyber intrusions.*
  This hypothesis test is significant because it will test the reduction in the probability of an attack on the real components of the IT/OT system. When the number of components impacted is reduced the architecture is more resilient to cyber-attacks.

1.8 Scope of Research

The research presents a Python model created as a tool for assessing the percentage of successful cyber-attacks on a critical infrastructure. Research demonstrates that as deception complexity increases, the time to a successful breach, starting at the IT levels

and ending at the plant control room or PLCs increases. The model also demonstrates that as complexity increases the probability of a successful system breach decreases.

1.9 Research Limitations

The model is based on Purdue system architecture. It would not be applicable to systems that do not follow the architecture, but provides a methodology for developing such tools.

Limitations of the research are all due to security measures in the field and concerns related to data release. The limitations are that

1) the results obtained are of a Python simulation rather than of a real system;
2) the model presents a simplified model of an IT/ OT system architecture for proof of concept, but can be expanded to simulate a specific system; and
3) an inability to use probabilities from real-world risk assessments, due to security concerns.

The praxis shows the need for well-designed decoys and explains what these would look like, however, does not go into further details relating to decoy design.

1.10 Organization of Praxis

Chapter 2 summarizes the literature related to critical infrastructures, system architectures, the current challenges relating to critical infrastructures cyber protection, and a summary of attacks on critical infrastructures over the past ten years. Models for IT/OT system architecture and for cyber defense are introduced. The concept of deception and decoys is introduced as a means of protecting industrial control systems

and in turn critical infrastructures. Chapter 3 presents the Python model developed to simulate a critical infrastructure and the Advanced Persistent Threat-type attacks on the infrastructure, the metrics used to test the hypotheses and the data collected. Chapter 4 presents a summary of the statistics data analysis comparing the systems before and after the incorporation of the cyber decoys. Finally, Chapter 5 summarizes the findings of the research and highlights directions for future research in critical infrastructure cyber defense.

2 CHAPTER II LITERATURE REVIEW

Critical infrastructures, whether maintained and operated by governments or owned privately, are complex engineering systems which are comprised of smaller subsystems that are made up of electrical and mechanical machinery controlled through control systems. Traditionally these control systems were not accessible through the internet (Stouffer, Falco, & Scarfone, 2011). However, with technological advancements in recent years and the emergence of the Internet of Things (IoT), and the Industrial Internet of Things (IIoT), there has been a race to connect these machines and their control systems to the internet (Rubio, Alcaraz, & Roma, 2019). The drive has been to aid operators and owners in gaining immediate access to data pertaining to the success of their operation, facilitating decision making (Stouffer, Falco, & Scarfone, 2011). These rapid system transformations in the absence of thorough analyses of threats and vulnerability have allowed critical infrastructures to be the subject of invasive cyber-attacks.

The field of cyber security research for critical infrastructures is dominated by the research relating to intrusion detection leaving a significant void pertaining to defense of critical infrastructures against cyber-attacks.

This chapter will establish the framework for the praxis research by reviewing the literature on the IT/OT system architecture; OT-system risks; cybersecurity vulnerabilities; prominent attacks on critical infrastructures; current state of infrastructure cyber defense; and the measures which need to be taken to ensure cyber resilience of critical infrastructures.

2.1 Informational Technology versus Operational Technology System Architecture

Critical infrastructures were designed to be independent structures not connected to the internet (Cai, Wang, & Yu, 2008). The control systems of these industrial complexes have undergone three episodes of change that have led to their vulnerabilities, as we know them today. What once used to be an isolated industrial system is now comprised of both an enterprise informational technology (IT) system and an operational technology (OT) system. Traditionally, the IT/ OT system architecture had a physical separation between the IT system and the OT system which it managed (Williams T. , 1994). This separation is called an airgap (Colelli, Panzueru, & Pascucci, 2020). With the emergence of the internet, owners and senior managers saw the opportunity to collect data on operations and provide them, through internal and closed computer networks, to operators and decision makers. OT system, which are comprised of control systems which relay information about operational status to remote terminals and receive changes in operational instructions from Programmable Logic Controllers (PLCs), were connected to the internet in an effort to improve business decision making and process efficiency (Cai, Wang, & Yu, 2008). The connection of OT systems to IT system in the enterprise network has eliminated the physical separation which used to exist between the IT and OT systems , replacing the physical separation with a double firewall (Cook, Janicke, Smith, & Maglaras, 2021).This direct connection between the two systems and the elimination of the physical separation which used to isolate the OT system from any cyber-attacks the IT system is subjected to has led to a significant OT system vulnerability (Cai, Wang, & Yu, 2008). This vulnerability is exacerbated by differences

in network security between the two systems. As industrial security is more developed on the IT-side of the architecture, industrial experts continue to promote the same techniques developed for IT system security for the security of OT systems. The differences between the operational needs of both systems make it impossible for these security measures to transfer over from the enterprise side of the architecture to the operational side. Furthermore, there are times when security updates are overlooked because they interfere with the accessibility of the machinery and the command-and-control signals used to control their operations (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020).

2.2 Operational Technology Systems

Operational technology systems are the industrial equivalent of IT systems and mentioned above, they are comprised of both the computing network and the industrial controls of industrial systems, and are shown in Figure 1,

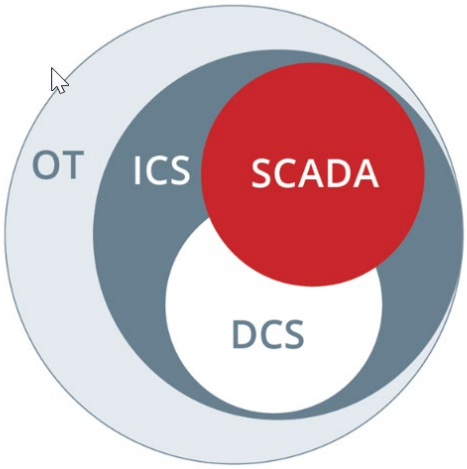


Figure 1. Relationship between OT and ICS systems courtesy of Securicon Information Security Solutions

Industrial Control Systems (ICS) include the components that control and maintain the operation of industrial systems such as power plants and oil and gas plants, electric utility grid and manufacturing plants. Industrial Control systems are comprised of Supervisory Control and Data Acquisition Systems (SCADAs) as well as Distributed Control Systems (DCS). SCADAs provide the systems for management of large complex processes that are found in utilities where the control of system components happens over a large geographic area (Securicon, 2022). The primary function of a SCADA to control widely dispersed assets using centralized data acquisition and supervisory control (Stouffer, Falco, & Scarfone, 2011).**Error! Reference source not found.** Figure 2 illustrates a typical SCADA system architecture.

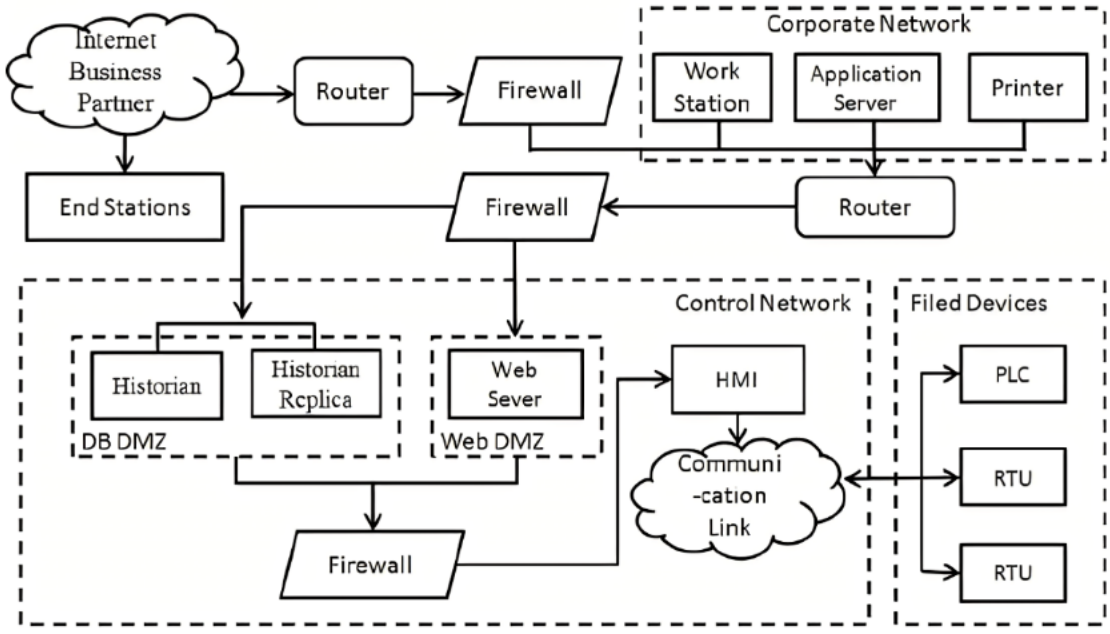


Figure 2 SCADA System Network Architecture (Qian & Abdelwahed, 2014)

SCADAs are comprised of several components that can be classified as control center components or on-site field components. Remote terminal units (RTUs) collect data and

send it to a command center (operating room) where decisions about the processes are made. Distributed Control Systems resemble SCADA systems in that they collect data on controllers, sensors and actuators, but differ in that they are typically situated in the proximity of the instruments from which the data is being collected. Other essential components are Programmable Logic Controllers (PLCs) and Human Machine Interface (HMIs). Within the PLCs is where the control systems operations code resides. An exhaustive review of OT system attack literature (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020) (Allen, 2021) (B. Zhu, 2011) (Dahad, 2018) shows that PLCs are the system components subjected to the highest number of attacks (Biswas, 2022). This is because when an attacker wants to gain access to the command-and-control signals of an industrial control system, it is done through the PLCs ((Greenberg, 2018.) (Kim, 2014)). Moreover, PLCs, being microprocessor-based devices with internal architecture analogous to computers and controllers (Alves, Thiago; Morris, Thomas, 2018) and industry-known proprietary logic ladders and programming codes, are a familiar target for attackers. HMIs are primarily situated in plant control rooms and constitute the displays where the data from various system components are displayed to control room operators (Stouffer, Falco, & Scarfone, 2011).

2.3 Risk in Critical Infrastructure Systems:

In designing a defense for a complex system of systems, such as a critical infrastructure, it is imperative to be aware of all system components and the role they play in the security or vulnerability of the overall system. Although, traditionally, risk has been addressed through the lens of the traditional risk matrix, Hubbard shows the limitations of

assessing risk solely from a qualitative lens, where the actual risk becomes relative and can depend on the assessor (Hubbard, 2009). In the field of cyber security, and with the drive to create cyber resilience infrastructures, ones that can recover from the impact of an attack, there is a growing need amongst engineering managers to quantify risk; (Seda-Sanabria & Morgeson, 2016), (Burns, et al., 2016) (Morgeson, Meeson, Simpson, Dechant, & Hermes, 2016) detail the best practices for assessing risk for industrial control systems and critical infrastructure. These reports describe the need to account for different attack scenarios, knowing all system components, and the probabilistic calculation of risk based on threats vulnerabilities and consequences. Very little information, however, is available in literature when it comes to specifics or examples of cyber risk assessments and the quantification of risk for specific infrastructures (Ryan & Mazzuchi, 2012). This is because many organizations do not publish the quantification of their cyber risk for security reasons. When reviewing procedures and literature pertaining to operational technology systems or critical infrastructures it is typical to find recommendations for cyber hygiene, system vulnerability patching, anti-malware and intrusion detection systems (Verve & Livingston, 2021). Additionally, when assessing cyber risk, the literature illustrates a skew towards the goal of estimating risk so that the financial impact can be quantified. However, assessing risk is also a main factor in devising a defense strategy and ensuring the continuity of operations (Petrenko, 2019).

Many security companies overlook that differences between IT and OT systems require a difference in the management of cyber risk and that one of the main causes of the vulnerabilities of OT-systems is that their vulnerabilities and threats are managed as are

those of IT-systems (B. Galloway and G. P. Hancke, 2013) (Cai, Wang, & Yu, 2008) (Bodeau & Graubart, 2017).

Further, Janicke et al. (Janicke, Cook, Nicholson, & Jones, 2016), indicate that 40% of ICS operators are unaware of the fact that components of their operation are accessible via the internet (Janicke, Cook, Nicholson, & Jones, 2016). Risk is quantified in terms of the product of the cyber threat, vulnerability, and financial consequence. This is often helpful when making decisions about risk management and insurance, but this does not address the non-financial impact of the risk and it does not help with the engineering management and development of defenses for OT systems.

McQueen, et al. (McQueen, Boyer, Flynn, & Beitel, 2006) propose a methodology for risk assessment for SCADAs. The proposed methodology assumes that risk is a function of the elapsed time which is required for a successful attack. Their methodology shows that risk is highly correlated with the estimated time to compromise each device. Proposed outcomes of the assessment are attack paths, time to compromise each device and pathway to risk reduction. Although this is used to calculate the cost of risk, this concept provides guidance as to how to assess risk through a need to assess threats, likelihood of occurrence and the consequence. This not only helps with assessing risk, but also is useful in devising a path for cyber defense.

2.4 The Purdue Model

When designing a methodology for cyber defense, the system architecture becomes a significant element of the equation. This section will look at the system architecture of IT/OT systems and the role the architecture plays in OT system vulnerability.

As IoT and IIoT systems evolved, experts have worked to define their system architectures as a first step to creating defense mechanisms. Challenges have been that the system architectures are not well-defined and based on the source used the architecture can have two, three or five layers (Boral, 2021). Similarly, there is variance in the number and type of components (Boral, 2021). Three common models are the International Telecommunication Union's Telecommunication Standardization Sector model, the NIST Model for Smart Grid (Locke & Gallagher, 2009), the IEEE Three-Tier Architecture Model (Boral, 2021), and the ETSI Architectural Model for M2M Communication (Mohapatra & Bhuyan, 2016). The issue with these models is that they focus primarily on the IT architectural side and the internet protocols of communications. Figure 3 depicts the Purdue Model, developed in the late 1980s by Theodore Williams and his colleagues at Purdue University (Williams T. , 1994). The model was developed with the purpose of providing a general framework for describing interactions between IT and OT systems and is shown in Figure 3.

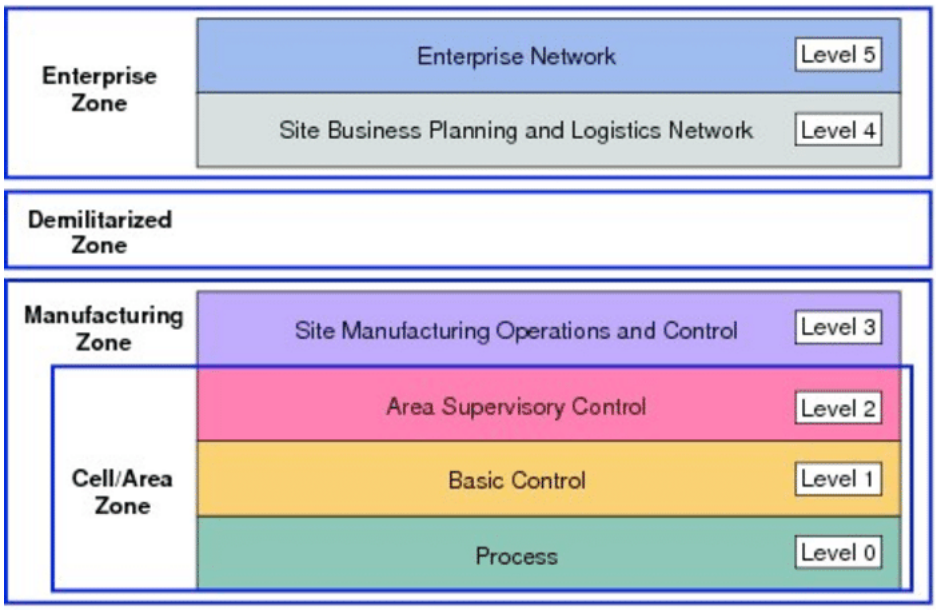


Figure 3 The Purdue Reference Architecture for ICS (Zanutto & Follis, 2017)

In contrast with other models, the Purdue model illustrates the paths through which data travels in between the OT and IT systems as well as the layers of the OT system. The model is divided into distinct layers; at the top is the enterprise IT network and at the very bottom are the various elements (typically gauges and other sensors) that collect data in the field. There is controversy in industry pertaining to if this model still represents the reference architecture for the IT/OT systems. Murphy argues that the original model identifies a segregation between the OT and IT systems which no longer exists in actual systems (Murphy, 2022). However, Patterson holds that the model is still a valid illustration of the layers of the two systems and the interface between them (Patterson, 2018). Although it is true that the physical separation between the IT and OT systems no longer exists, the Purdue Model still provides a valid depiction of the layers of both systems and the possible pathways to a cyber-attack.

2.4.1 Vulnerabilities within Levels of the Purdue Model

Traditionally SCADA systems were deployed in isolation of other IT systems, protected through an airgap (Williams T. , 1994) (Zanutto & Follis, 2017). With the desire to collect real-time data operators have had to allow more connectivity and allow for firewalls to replace airgaps resulting in vulnerabilities (Janicke, Cook, Nicholson, & Jones, 2016). A summary description of each level of the Purdue model and the vulnerability at that level follows

Level 5: The Enterprise network has a well understood architecture making it relatively easy for hackers to navigate once inside (Janicke, Cook, Nicholson, & Jones, 2016; Zanutto & Follis, 2017). The enterprise level is remotely accessible, however lower levels of the Purdue model are also accessible based on need to collect data and send up the hierarchy to management and operators (Janicke, Cook, Nicholson, & Jones, 2016). Not only are the enterprise network vulnerabilities well known to attackers, but security mechanisms and techniques are often not properly deployed leading to vulnerabilities that can be avoided, according to a study by the US Department of Homeland Security US-United States Computer Emergency Readiness Team incident response summary report of 2009–2011 (United States Computer Emergency Readiness Team, 2011).

Level 4: Is very similar to Level 5. Activities are mainly outward facing with 90% of the traffic being data exchanged with other data centers on the internet (B. Galloway and G. P. Hancke, Introduction to industrial control networks, 2013) . Data processed at this level is data coming from Level 3. As a result, this level has critical information regarding the control network (Janicke, Cook, Nicholson, & Jones, 2016).

Airgaps which are sections of the architectural framework that physically separated IT from OT systems have been eliminated to facilitate maintenance (cyber physical reference) these airgaps were replaced with firewalls which are less efficient in terms of separation. At the Level 4 -Demilitarized Zone (DMZ) interface, a pair of firewalls, one at level 3 and the other at level 4 act as the "airgap" separations between both the OT and IT systems (Stouffer, Pillitteri, Lightman, Abrams, & Hahn, 2011).

Research indicates that approximately 40% of the cyber incidents occur during the process of transmission between level 3 and Level 4, as data is cascaded upwards to the enterprise network (Janicke, Cook, Nicholson, & Jones, 2016).

Demilitarized Zone: As previously explained, this area of the network borders Levels 4 and 3 and is in the form of firewalls, which have replaced air gaps (Stouffer, Falco, & Scarfone, 2011). The removal of airgaps has caused a shift in the favor of hackers because the absence of airgaps combined with known computer systems and software codes has allowed attackers an advantage over defenders (B. Galloway and G. P. Hancke, 2013) (Janicke, Cook, Nicholson, & Jones, 2016).

Level 3: At this level changes to operational controls are deployed. Data collected on operations is transmitted to the enterprise network (Levels 4 & 5) through this level passing through the DMZ. Applications at this level are based on operating systems such as Linux, Unix and Microsoft Windows and IP. As a result of the use of standardized computers and software access to this layer can be made easier due to these known codes and computer systems. Access at this stage allows access to both the enterprise network and the OT network. Known internet protocols at this level help facilitate attacks (Janicke, Cook, Nicholson, & Jones, 2016).

Setting up security at this level is a challenge because it historically interferes with the efficiency of the operation of the OT environment and its components. Intrusion detection systems, which are effective at Levels 4 and 5, present a challenge to use at this level due to the latency which can result and consequently impact (Janicke, Cook, Nicholson, & Jones, 2016).

Level 2: At this level the infrastructure control systems network exists and so as a result access to this level of the network are thought to not be connected to the internet or visible from the internet. The control systems communicate with and control operations of instruments at Level 1 and share data with enterprise levels. Reconfiguration of logic controllers for devices at Levels 1 & 0 occur at this level. Distributed Control Systems are used at this level to manage assets and performance of components at Level 1 (Janicke, Cook, Nicholson, & Jones, 2016).

Level 1: Level 1 is a "logically abstracted level" (Janicke, Cook, Nicholson, & Jones, 2016). It is difficult to define the separation of Levels 1 & 0. Programmable Logic Controllers (PLCs) exist primarily at this level and thus an attack at this Level is lethal to the OT system because access allows attackers to inject commands through the command shell allowing attackers to take complete control of the PLC including the devices which it manages (Beresford, 2011).

Level 0: Process occur at this level. Sensors and gauges for the various processes are also at this operating level where data is collected and sent back up to the operators and the enterprise network. To compromise a critical infrastructure, through a man-in-the middle attack, an attacker would need to intercept the signals and spoof them indicating to operators that machinery is operating normally while changing system settings (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020).

The distributed nature of SCADA systems makes physical controls and security a challenge because of the expense associated with the physical security in remote locations. Current system designs allow an attacker who gains access to a single network node access to the full network. Research has shown that attacks at this level are often

insider attacks through the actions of a disgruntled employee or the likes/ industrial sabotage/rogue contractors. Default passwords and usernames that are not changed routinely allow access to this type of attacker who knows system well (Colelli, Panzueru, & Pascucci, 2020) (Janicke, Cook, Nicholson, & Jones, 2016).

2.5 The Cyber Security Triad

Another concept in the equation of critical infrastructure cyber resilience is that of the challenge to balance between integrity, confidentiality and availability. This concept is depicted through the "Cyber Security CIA Triad", illustrated in Figure 4.

Figure 4 Cyber Security CIA Triad (Preferred IT Group, 2019)

This figure illustrates a dilemma within IT systems where it is a model used to demonstrate the need to balance system integrity with system confidentiality and its availability. Confidentiality refers to maintaining sensitive information; integrity refers to the consistency of data and reflect the capability of the security system in maintaining data and restricting unapproved changes such as ones resulting from an attack. The third

and last leg of the triad is availability and refers to the accessibility of the system as well as the resolution of software hardware conflict. The CIA Triad can be used to understand the main cause of vulnerabilities with OT systems. OT systems operate on the basis of availability ((Kim, Yannakogeorgos, & Lowther, 2013)), meaning that, unlike IT systems, OT systems depend on their availability/ accessibility for operation. If OT systems operated like Enterprise IT systems, with confidentiality at the core, critical infrastructures would be at risk of operating efficiently, because they are not accessible.

Relating this back to confidentiality, the OT levels of the Purdue Model are dependent on their availability and accessibility. This makes the design of the demilitarized zone (DMZ) essential for security of the OT levels. Traditionally protected by an Airgap, many facilities have removed that airgap for "continuity of business". Airgaps refer to the physical separation between both OT and IT systems (Kim, Yannakogeorgos, & Lowther, 2013). This concept of an air gap was broken down by the need to access OT systems remotely and in real-time to make business decisions. Further, many of the components within an OT system are pieces of machinery that have long lifespans such as pumps and motors. Unlike their IT counterpart, the controls and in turn the programable logics that control their machines when they became connected to the internet, have very little room for security patching and updates (Allen, 2021)**.**

2.6 Attacks on Critical Infrastructures

The Center for Strategic & International Studies (CSIS) reports that business losses from cyber-attacks have risen from $21 million to $105 million (Sectigo, 2022), in the past decade, with the Federal Bureau of Investigation reporting consumer losses being

approximately 4.5 Billion (Sectigo, 2022). These losses have been associated with a rise in phishing attacks up by 1.5 million, and are alarming for critical infrastructure operations because the abundance of statistics pointing to phishing attacks being the starting type of attacks on critical infrastructures (Fire Eye, 2021) (Allen, 2021) (B. Zhu, 2011) (Kim, 2014). Table 1 summarizes significant attacks on critical infrastructures.

Industrial Control System	Attack vector	Purdue Model Layer Impacted	Consequence	Year
Davis-Besse, US (Rowe [. B., 2012.)	Slammer worm	Phishing email Enterprise IT Level	Safety monitoring system was disabled for 5 hours.	2003 (Moore, PAXSON , SAVAGE, & STUART)
DaimlerChrysler automobile plants, US (K. Kimani, 2019)	Zotob worm	Phishing email Enterprise IT Level	Stopped production at several sites.	2005
Natanz nuclear facility, Iran (Kim, 2014)	Stuxnet malware	USB Drive infecting OT system	Destroyed centrifuge tubes at the Uranium enrichment facility.	2010
German steel mill, Germany (R. M. Lee, 2014.)	Unknown	Email phishing/malware Enterprise IT Level	Blast furnaces were shutdown leading to loss of plant control by the plant operators resulting in physical damage to the system and process interruption.	2014
US and European Energy companies (Dragonfly: Western Energy Companies Under Sabotage Threat, 2014)	Havex malware	Enterprise IT Level	Attackers compromised several strategic organizations. This included energy grid operators, major electricity generation companies, petroleum pipeline operators disrupting the energy supplies to impacted countries.	2014
Kemuri Water treatment plant, US (K. Kimani, 2019)	SQL injection and phishing	Enterprise IT Level	Personal information of 2.5 million customers compromised.	2015
Power grid, Ukraine	Spear phishing/ Black Energy 3 malware	Enterprise IT Level	Power outage impacting over 220 thousand users, and access credentials stolen.	2015
Power grid, Ukraine (A. Cherepanov and R. Lipovsky, 2017)	Industroyer malware	Enterprise IT Level	20% of Ukraine's capital, Kiev was disconnected from the grid for 1 hour.	2016
Multiple businesses world-wide - Starting from a Ukrainian software firm (Greenberg, 2018.)	NotPetya ransomware	Enterprise IT Level	The impact to the ransomware spread to pharmaceutical company Merck, to snack company Mondelez and several other big industries worldwide leading to a combined financial loss of over 10 billion dollars across these industries.	2017
Petrochemical plant, Saudi Arabia	Triton malware	Enterprise IT Level	Malware impacts safety systems in the plant	2017
2018 Taiwan chipmaker TSMC (Kumar, 2018)	WannaCry ransomware	Enterprise IT Level	Causes several iPhone factories to shutdown	2018

Table 1 Major Cyberattacks on critical infrastructures reported between 2003- 2019 adapted from Alladi et al. (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020)

These examples (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020), reinforce the concept that attacks will either initiate on the enterprise IT side and make their way down the levels of the Purdue Model to the SCADA level or they would penetrate the

layers at the SCADA/PLC levels through the “insider job” approach. Phishing attacks on the IT-side of the architecture, which are primarily related to poor cyber hygiene and are only part of the problem. There is an equally significant issue on the OT-side. Industrial control systems utilize what is called an Open Protocols for communications (Luallen, 2014) (Real Pars, 2022). Open protocol means that the communication protocols are unified across industry to facilitate communications between control systems and electro-mechanical components and ensure seamless operation. Commonly used protocols include Modbus and TCP/IP. These communication protocols are used for transmission and receipt of instructions between components of the OT system, and include layers of instructions and component identifiers/ labels. These are transmitted as packets and when intercepted by hackers can allow for entry points to system components as these packets are packed and unpacked by the components of the OT network (Real Pars, 2022). Adding to this vulnerability is the fact that Programmable Logic Controllers and their programmable language scripts are known though out industry because these SCADA system components are off the shelf components from a known group of manufacturers. The most widely known, and used, are the Allen-Bradley PLC and code, the Siemens PLC and code and the Schneider Electric PLC and code (B. Galloway and G. P. Hancke, 2013) , (Real Pars, 2022), (Beresford, 2011). In the STUXNET attack, for example, the attackers knew that the PLCs were Siemens ones and as a result the PLC scripts were known to the attackers allowing them the ability to modify and inject a malicious code at the PLCs resulting in damage to the entire infrastructure (Beresford, 2011) (Warner, 2012).

It is important to note that there are other types of attacks on critical infrastructures. The most impactful have been denial-of-service type attacks (Rubio, Alcaraz, & Roma, 2019) and man in the-middle-attack. Denial of service or DoS attacks occur when the IT-side of the architecture is flooded with packets of data at the fire walls which act as gatekeepers to the traffic at the firewall boundaries. This flooding of packets allows for the attacker to breach the perimeter and enter the IT side, once again, and make it to the OT side. The attacker then holds the system hostage pending the payment of a ransom. The man-in-the-middle attack is one in which the attacker intercepts the signals between the elements of the control system and the control room operators. In both cases, the attack commences as a stealthy, undetected attack on the IT-side and ends up causing damage on the OT-side. Because of the commonalities with APT attacks, the remainder of this research will focus on APT-style attacks, ones which commence in the IT layers of the architecture as phishing attack and make their way down the architecture taking control of the OT layers.

2.7 Advanced Persistent Threat Attacks

Cyber security data collected by Kaspersky ((Kaspersky, 2022) and FireEye (Fire Eye, 2021) list APT attacks as being 56% of the types of attacks launched on enterprise and critical infrastructures. An APT is often confused with the other common attacks that are a result of poor cyber hygiene, rather than a stealthy cyber-attack (Mei, Han, Li, & Wu, 2021). APT attack is a term coined by the Air Force (Mei, Han, Li, & Wu, 2021) and they are often conducted by stealth attackers; they are advanced because the groups behind the threat have sophisticated intelligence-gathering techniques at their disposal. They are persistent in that the attackers do not act quickly attacking the system looking

for monetary rewards, but rather dwell in the attacked system, working in a way to cover their tracks, while collecting further information about the system being attacked, allowing them to make further advancements Finally, APTs are defined as threats because of the combination of their capability for a successful attack and the intent behind the attacks (Mei, Han, Li, & Wu, 2021). Because of the sophistication and stealth involved in an APT attack, research is limited in the amount of data collected on these types of attacks. Understanding an APT attack is crucial in designing an infrastructure which is resilient towards these types of attacks. The APT lifecycle is summarized in Figure 5.

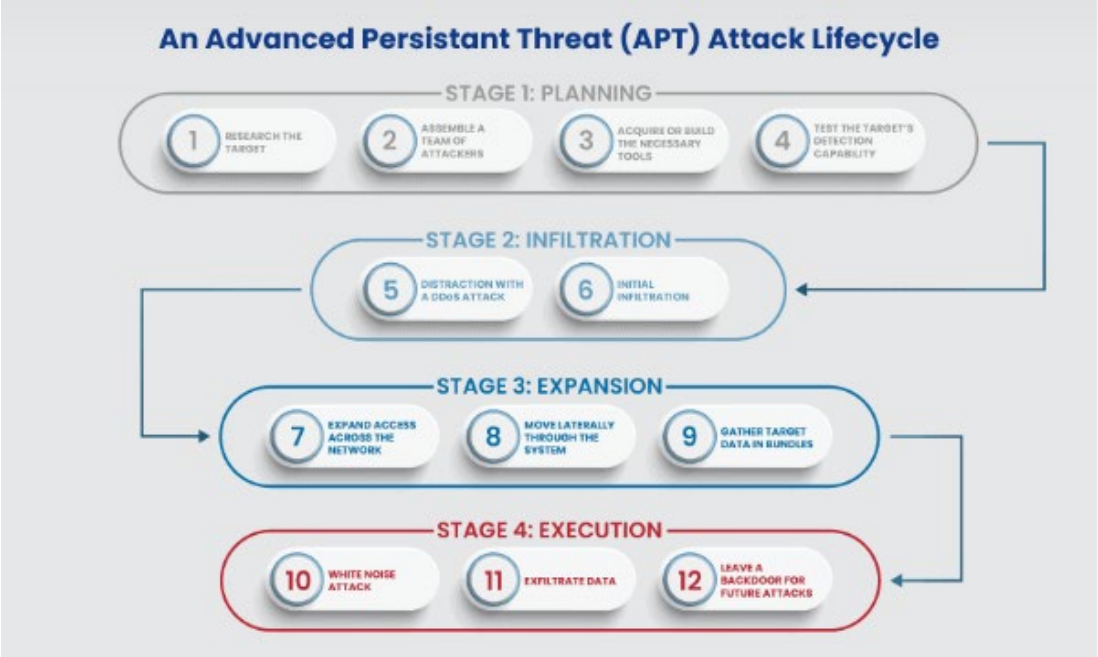


Figure 5 Anatomy of Advanced Persistent Threat (Heindal Security, 2022)

Lockheed Martin defines what is known as a Cyber Kill Chain to aid in the understanding of how cyber-attacks can be disrupted. The Cyber Kill Chain is analogous to kill chains in the military which is a process by which the stages of attacks are described. The Lockheed Martin (LHM) Cyber Kill chains has been adopted by researchers (Heindal

Security, 2022) to model the APT kill chain which is demonstrated in Figure 5. The MITRE Corporation has also developed a model for a cyber-attack (Bodeau & Graubart, 2017), the Cyber Attack Lifecycle, depicted in Figure 6 .

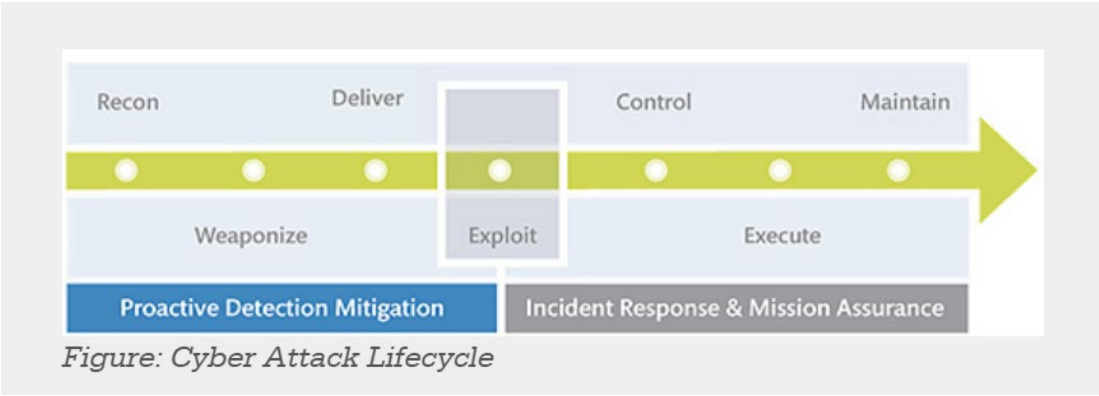


Figure 6 MITRE Cyber Attack Life Cycle (Bodeau & Graubart, 2017)

Like the LHM Cyber Kill Chains, the MITRE Corporation model depicts the steps and attacker takes in collecting information prior to system penetration until the attacker gains control of the command-and-control signals of an OT system. The very first step of the cyber kill chain is to gain an understanding of the environment being targeted. The second step is to weaponize. This often means that the attacker designs a "weapon" to deploy into the targeted system. This will be discussed in further details in future sections. The third step is finding a vulnerability/entry point into the architecture.

The fourth step the attacker takes is to deploy the "weapon, while maintaining stealth. The fifth stage is one in which the attacker gains access to command-and-control environment within the attacked system. This is often the stage at which the operators of the ICS notice that the integrity of the infrastructure has been compromised.

In his paper, APT cyber-attack modelling-building a general model, Dr. Martti Lehto (Lehto, 2022) describes the lack of availability of APT models and the importance of

creating ones to aid in the decomposition of attacks into discrete phases. He presents that models should include an Early- Attack Phase, a Pre-Attack Phase and then an Attack Phase. Lehto's model schematic is presented in Figure 7 below. His model is similar to the LM model and MITRE model even though they are generic of cyber-attacks.

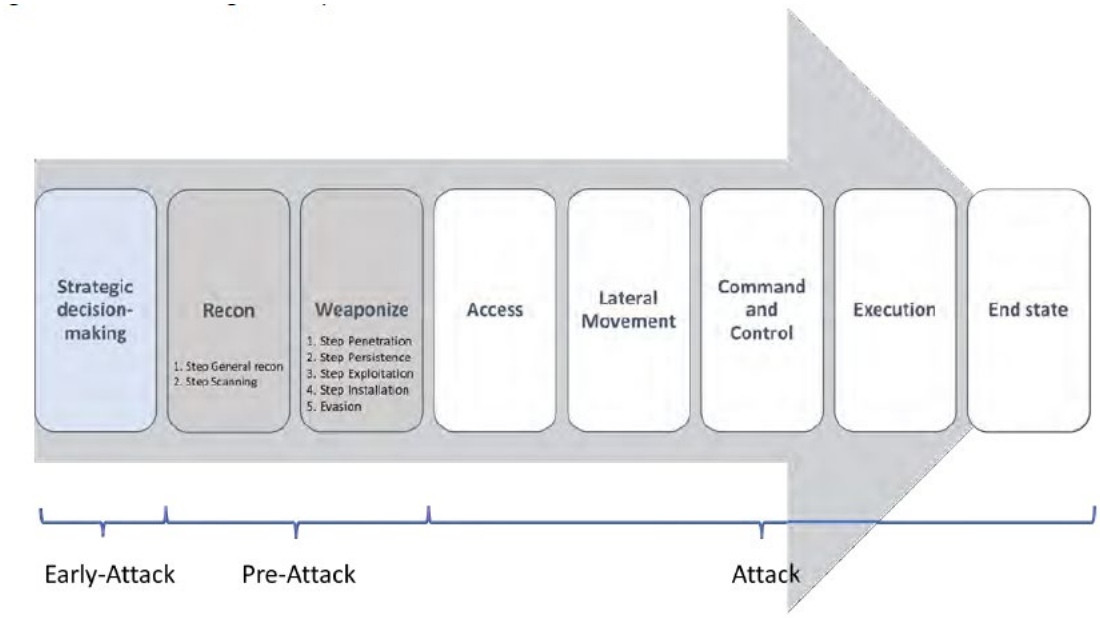


Figure 7 Martti Lehto APT -attack model (Lehto, 2022)

Lehto presents major APT attacks and the timing for their detection. Table 2 indicates that they often go undetected more months if not years and as such presenting the case for the need to understand more about the time and impact of cyber-attacks on system components.

APT operation	Targets	Year of detection	In operational use since
Operation Aura	Dozens of IT organizations (Google among others) specially in US.	2010	2009
Operation Shady RAT	More than 70 companies and government entities around the world.	2011	2007
Duqu	Information from industrial control systems.	2011	2007
sKyWIper/Flame	The Middle East governmental organizations, educational institutions, and individuals.	2012	2008
NetTraveler	More than 350 organizations in 40 countries	2013	2004
Regin	Several organizations at least 14 countries like small businesses and telecom companies.	2014	2008, first sample 2003
Remsec (Project Sauron)	More than 30 organizations in Russia, Iran, China: government, scientific research centers, military, telecommunication providers and finance.	2016	2011

Table 2 Time to detection of APT attacks (Lehto, 2022)

Ryan Wagner of the Carnegie Mellon University Institute for Software Research discusses an "Attack trace" which models an Advanced Persistent Threat, and considerations on both the attacker and defender sides (Wagner, Fredrikson, & Garlan, 2017) . Wagner describes in his articles several attack scenarios however doesn't provide a model for simulation. Through further review of the literature and statistics available through architecture security companies, it becomes evident that this is a deficit in literature pertaining to models of IT/OT architecture and APT modelled attacks on such architecture would aid in understanding and defending against APT attacks.

2.8 Current Critical Infrastructure Defense Mechanisms

Operational Technology system defense has primarily been adopted from Informational Technology Systems. However, the inherent differences between the architectural and operational requirements of both systems, result in the traditional defenses, which are implemented and *may* prove successful on the IT-side, not providing the required level of defense on the OT-side (Fan, Du, Fernandez, & Villagr , 2018). Examples of such IT-

system defenses are Intrusion Detection Systems, Intrusion Prevention Systems, antivirus and anti-malware solutions. In addition to being designed to serve the IT-side of the architecture, these solutions do not offer the cybersecurity professional the ability to analyze a potential attack or disseminate an attacker's strategy, two elements that are crucial in cyber defense. Moreover, from a cyber risk perspective, observing the news pertaining to critical infrastructure attacks it becomes evident that the most common risk management methodology is to transfer the risk, perhaps to an insurance company that pays for the ransom or the damage.

Research in the field of OT systems defense is moving towards cyber deception. (Jajodia, Subrahmanian, & Swarup, 2016). Cyber deception is the "planned actions" which are taken with the intent to "mislead and/or confuse attackers and to thereby cause them to take (or not take) specific actions that aid computer-security defenses." (Jajodia, Subrahmanian, & Swarup, 2016). The following subsections present information on the status of CI cyber defense and the need for more contribution to the fairly new body of knowledge.

2.8.1 Intrusion Detection System Systems

Although often considered a form of defense (Mohamed & Aydinc, 2022), Intrusion Detection Systems (IDS) which are classified into anomaly-based IDS and misuse IDS have several shortcomings and cannot be solely used to prevent attacks. Misuse-IDSs detect intrusions based on known patterns or signatures of attacks, while anomaly detection-IDSs detect attacks based on known patterns of normal (benign) activities and consider all other attack patterns which deviate from the known ones malicious. IDS are

typically based on machine learning algorithms and utilize information fed to them in constructing a detection model. Although, IDSs have proved to be advantageous, they do not perform as expected unless they are constantly updated in order to perform as expected. So, for example, enacting old IDSs on IoT will not provide for a successful intrusion detection system. IDS simply act as a mechanism for detecting variations in the packet exchanges throughout the network; stealthy attacks or attacks that the IDS is not trained to detect will fail to be detected. Lastly, IDSs do not act as a form of defense whether on the IT or OT sides of the architecture.

2.8.2 Honeypots

For several years, researchers have explored the concept of a “honeypot”, a system component which is placed in the network to attract and deceive attackers into thinking it is an actual system component, become compromised and enable the collection of information on the attacker’s tactics and collect information on the attacker’s tactics. (Fan, Du, Fernandez, & Villagr , 2018) (Franco, Aris, Canberk, & Uluagac, 2021). Honeypots can be placed at different locations within the architecture and are designed purely as data collection devices.

A survey of the literature available on honeypots illustrates that honeypots evolved through three generations as summarized in Table 3 below.

Honeypot Generation	Deployment	Success in Defense	Deficiency in Operation
Generation I	Developed in 1999	Comprised of three consecutive levels a) firewall b) IDS c) honeypot	Easily detectible
Generation II	Developed in 2002	Honeypot sensor replaced the firewall and IDS components. The sensor made it more difficult for attackers to detect the honeypot	Easily detectible
Generation III	Developed in 2004	Identical to Generation II but with more advanced deployment and management capabilities (clarify)	Less detectible, but eventually detected.

Table 3 Generations of Honeypots (Fan, Du, Fernandez, & Villagr , 2018)

The typical honeynet architecture is depicted in Figure 8.

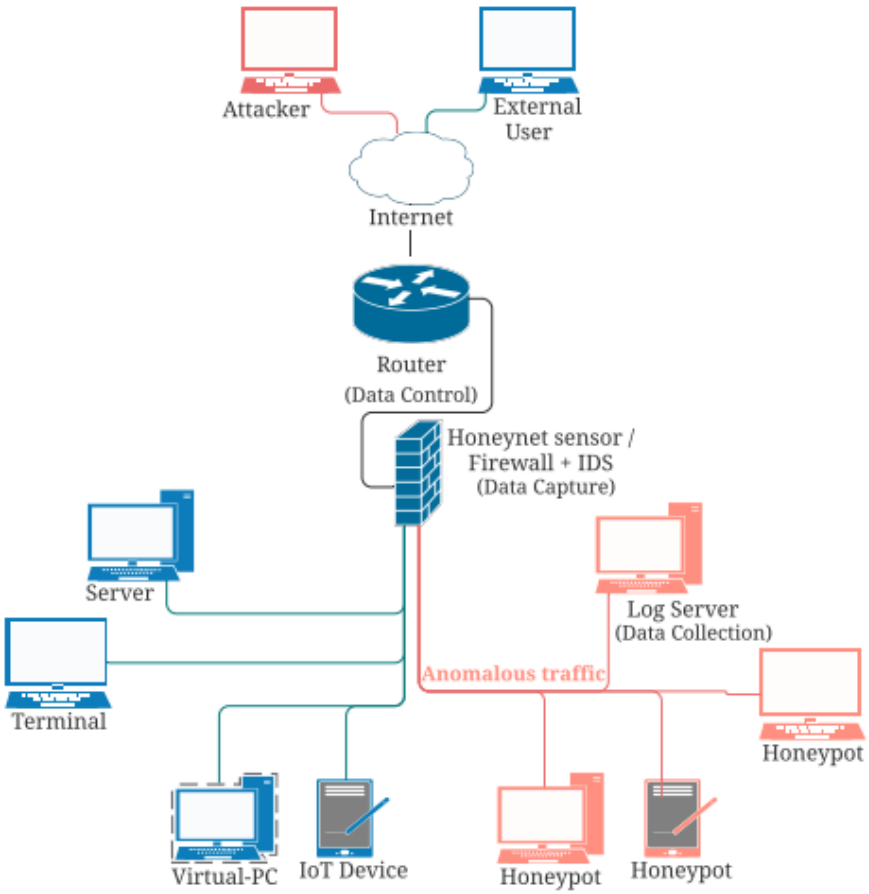


Figure 8 Schematic illustrating typical location of honeypots within an IT system (Franco, Aris, Canberk, & Uluagac, 2021)

The primary requirements for a successfully deployed honeynet are Data Collection, Data Analysis and Transfer (Franco, Aris, Canberk, & Uluagac, 2021). More than one

honeypot deployed in a system constitutes a honeynet. Although honeypots are a great concept, they have not been successful in preventing attacks because of the simplicity of their design which makes them easily discoverable by the attacker (Nguyen, 2022). Research has indicated that the lack of success of a honeypot comes from the lack of resemblance to actual system components which have packets of data traffic affiliated with them. Honeypot research indicates that they are discoverable by 85-100% (Surnin, et al., 2019) placing them lower on the scale for tools to use for cyber defense. As such, honeypots are of more value as components of intrusion detection systems; when honeypots are accessed it is an indication to the defender that there is a potential compromise of the system. Consequently, as their name suggests, honeypots are a great tool to incorporate into the OT system architecture to collect data on potential attacks and attacker and to enable defenders to disseminate attacker tactics.

2.8.3 Cyber Deception

In recent years, with the growing number of attacks on critical infrastructures, there have been many papers which present the idea of cyber deception as a means of defense against cyber warfare, the next generation of wars between nations (Colarik, 2007). Cyber deception refers to the "deliberate and controlled act to conceal networks through the creation of uncertainty and confusion against the adversary's efforts to establish situational awareness, and to influence and misdirect adversary perceptions and decision processes" (Climek, Macera, & Tirenin, 2016). Figure 9 provides an example deception taxonomy.

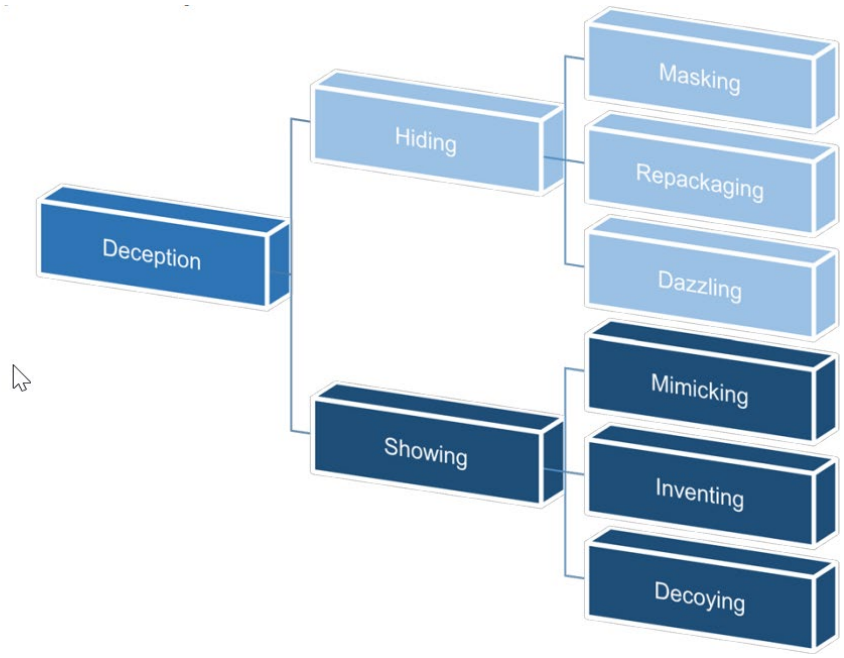


Figure 9 Whaley Deception Taxonomy (Service, 2022)

Cyber warfare has been demonstrated as a concept through the Stuxnet attack on the centrifuges of the Iranian nuclear reactor and attacks on water and power utility companies (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020). Warnings pertaining to potential Advanced Persistent Threat attacks have been high on the list of alerts by the Department of Homeland Security's Critical Infrastructure Agency (CISA.gov, 2022). A survey of the research indicates that OT system cyber defense is still in its infancy (Pawlick & Zhu, 2019; Colarik, 2007).According to MITRE and Bodeau and Graubart (Bodeau & Graubart, 2017), cyber deception as a means of OT system defense is the way forward. Fortinet (FORTINET, 2020) indicates that deception must be built into the OT system architecture.

Cyber defense is often reactive, rather than active (Rubio, Alcaraz, & Roma, 2019); an attacker aims to penetrate a system, the defenders create patches that resolve issue

"patching" areas in the network where vulnerabilities led to network access. Further, often in cyber defense, the human elements of a successful defense are overlooked (Ferguson-Walter, Maxine , Van Bruggen, Furgate, & Gutzwiller, 2019). Papers by Ferguson et al. speak to the human element of an attack and the success of creating deception as a means for cyber defense looking at the psychological warfare between the attacker and defender (Ferguson, 2018). Further, the shift to a system of systems architecture where the IT and OT systems are fully integrated and more accessible through the internet must drive researchers to find solutions to creating an architecture with built-in resilience which stems from cyber decoys that are realistic to the degree that attackers are unable to penetrate the network (Rubio, Alcaraz, & Roma, 2019) (Kakareka, 2020). As it currently stands, there is no practical research available on the implementation of high-fidelity decoys as a form of deceptive defense against attack on critical infrastructure. There are basic upon which one can build, however there are still many deficiencies in the field.

2.8.4 Digital twins

Digital twins are used as a step up from simulation and modeling where a "digital twin" is a very high-fidelity model of a complete machine or product, a process including all the machines and parts involved in the process, or the whole system or plant is replicated digitally and used to simulate a hypothetical scenario. Digital twins are fed with real-time data relating to the physical entity they replicate. They are used to significantly improve industrial operations offering pathways to reduce cost, improve maintenance, reduce downtime, run simulations, monitor assets, as well as predict future conditions using machine learning techniques (Devasia, 2022). It can be hypothesized that the utilization

of the concept of a digital twin would be a successful means for defending OT-system components. When looking at the architecture of an IT system versus an OT-system, it can be determined that OT systems have fewer redundant components in comparison to IT-systems, making the OT-system components more visible to attackers and thus more easily attacked.

2.8.5 Cyber-attack Modeling

Even though for operators and owners have been working to integrate both IT systems and OT systems, the differences between the operational and security needs of both systems, and the design of defensive strategies for operational technology systems that match those of IT systems, has resulted in OT systems being vulnerable to a multitude of cyber-attacks. These attacks can come in different end forms (Denial of Service, Man-in-the-Middle), however at the start they can often be traced back to stealthy attacks which made their way from the IT enterprise layers to the SCADA units on the OT side and then presented as Denial of Service, Man-in-the -Middle or other malicious attacks. The ability to “bounce back from the attack” is a measure of the cyber resilience (Petrenko, 2019) of the industrial control system, however this resilience is a function of the awareness of the operators and owners of the occurrence of an attack. As such, there is a drive that is documented through the multitude of cyber research (both basic and applicational) which is targeted at detecting intrusions or “Intrusion Detection Systems”. With advancements in machine learning, researchers can look at operational data collected from ICS and though an analysis of the packets of data exchange (MODBUS or TCP/IP) between system components and identify anomalies in these packets, thus

identifying an attack. This, however, does not stop attacks, nor does it contribute to the resilience of an infrastructure whereby it is able to bounce back from an attack. Resilience can be created through enabling an engineering manager to "buy time" for decision makers to act on the information that they are receiving from IDS. This time can be provided by deceiving the attacker and prolonging the time the attacker spends in the architecture looking for *real system components*. Because of the potential consequences of real attacks being launched on real systems during red-team blue team cyber exercises, (Bento, 2020), (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013), (Zhenxin Zhan, Maochao Xu, & Xu, 2013) and others point to the use of stochastic modeling and Markov chain models to simulate cyber- attacks. Markov Chain models are stochastic, memoryless processes used to model scenarios based on probabilities. They are referred to as having "Markov properties", named after Andrey Markov the mathematician who defined these processes. Markov processes are characterized as being "memoryless" meaning that a prediction is conditional on the present state and are independent of past and future states (Sharma, 2020), (Scheepers, 2019). Markov chain processes are discrete and there are many different types of Markov chains. In python modeling, Markov Chains can be modeled with or without using a state transition matrix (Scheepers, 2019), (Sharma, 2020).

2.8.6 Chapter Summary

Operational technology systems are vulnerable to cyber-attacks because of the need to maintain the availability and accessibility of their components for the continuity of

operations. The most common attacks on OT systems initiate on the IT side as a phishing attack at the enterprise level of the Purdue Model. The attackers make their way down the layers of the Purdue Model to the SCADA and PLC levels. Once a phishing attack is successful, the attacker attempts to gain access to system passwords and credentials which are stored at the DMZ level and then gains access to the SCADA intercepting Command and Control signals to PLCs and in turn to field equipment. In the past the physical separation between IT and OT systems in the form of an airgap helped prevent attacks on OT systems however with the elimination of the physical separation and the reliance solely on firewalls and intrusion detection systems for protection, OT systems have become more vulnerable to cyber-attacks. Although there are attacks that initiate at the DMZ, the more prevalent ones are ones that initiate at the topmost layers of the Purdue Model. There are several types of attacks on OT environments which have been coined Denial of Service attacks or Man-in-the Middle attacks. When reading and comparing how the attacks start and advance one realizes that they can all be related to the stealthy, persistent attacks in certain ways, the main differences are the end goals of the attacks.

In defending a critical infrastructure, such as a water treatment plant, it is crucial to understand the threat landscape as well as the vulnerabilities within the system and also the likelihood of certain attacks. In the water treatment environment, the targets are the PLCs which control the 6 different steps that make up the water treatment process. Typically, attackers would gain access to the PLCs and would attempt to modify the water quality and because a PLC is a microprocessor-based device the internal structure analogous to computers and controllers (Alves, Thiago; Morris, Thomas, 2018), PLCs,

and the field equipment they control, are easily attacked through the injection of a malicious code into the PLC.

Chapter 3 will present the engineering management tool created for the purpose of this research. The Python model presented is of a critical infrastructure built based on the Purdue system architecture, and the methodology to explore the impact of decoy complexity on the Mean Time to Breach and on the probability of a cyber-attack as well as the impact of various attacks on the breach of real system components.

3 CHAPTER III - Methodology

This Chapter describes the Python model developed for the purpose of this research. The code of the model is presented in Appendix A. The model is designed to emulate the system architecture of an IT / OT system, modeled using the Purdue Reference Architecture (Williams T. , 1994). The model is used to simulate a Markov Chain APT attack on the IT/OT system. Model performance is verified using several edge cases, presented herein, and is then subjected to simulated APT-style cyber-attacks.

Data is collected regarding the number of iterations the attacker takes to breach each layer of the IT/ OT architecture, as a proxy for time. Additionally, the system components accessed during the attempted attacks are tracked.

This section provides a breakdown of the model to the basics (Bare Model), and then different components (probabilities, escalation of privilege, firewalls, etc.) are added to present how the model behaves under different conditions, validating the model behavior. Chapter 4 shows the results of the modeling and compares the results against available findings in the literature.

3.1 Model Description

In this section the structure of the Python model is described. This is separated from the modeling of the attack, on the model, which is described in Section 3.2.

The schematic in Figure 10 depicts the 6-layers of the base model created for the research.

Figure 10 Schematic of Simulated Python Model

This is a tiered model where the type of components at each level, their quantities, and the probability of an attacker successfully accessing components within the given level, are defined for each tier. The Python model code is in Appendix A.

The model layers follow the system architecture of the Purdue Model with the two topmost layers representing the IT side of the architecture and the two lowest layers representing the OT infrastructure and the third and middle layer representing the Demilitarized Zone separating the IT and OT systems. (Alves, Thiago; Morris, Thomas, 2018)and (Cook, Janicke, Smith, & Maglaras, 2021) indicate that SCADAs and PLCs are the OT system components which are targeted the most during an attack.

Although the SCADA of any plant is comprised of several components, as detailed in Chapter 2, the Python model created for this research focuses on the SCADA equipment controlling the ICS at the plant control room , treats the PLC/HMI as one unit (and here labelling them as PLC), as depicted in the plant network schematic of

Figure 2, below and the field equipment such as sensors and actuators.

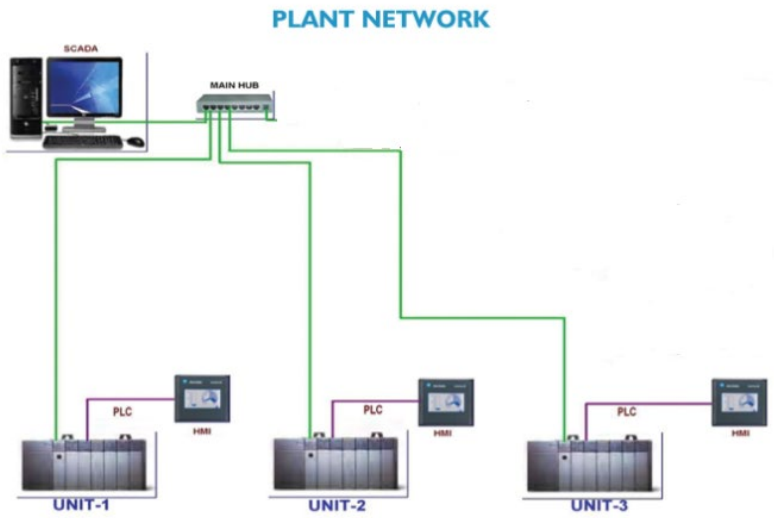


Figure 11. General Layout of SCADA System (Stouffer, Falco, & Scarfone, 2011).

IT/OT level identifier	Python MODEL Component	Purdue Model Level	Number of components at each Level of Purdue	Probability of successful attack at Level
IT section of architecture Administrator	A (AHP, AD)	5	10**	0.32*
IT section of architecture Employee	E (EHP, ED)	4	100**	0.32*
DMZ- Data Historian	H (HHP, HD)	3	2**	0.2*
DMZ- Patching Server	G (GHP, GD)	3	2**	0.2*
DMZ- Application Server	R (RHP, RD)	3	2**	0.2*
DMZ-Remote Access Server	T (THP, TD)	3	2**	0.2*
SCADA controller	C (CHP, CD)	2	1**	0.3*
SCADA architecture PLC	P (PHP, PD)	1	6**	0.5*
SCADA Sensor	S (SHP, SD)	0	240**	undefined

Table 4 provides a summary of the components at each layer of the model, their quantities and the probability of the attacker being able to breach that component/layer.

IT/OT level identifier	Python MODEL Component	Purdue Model Level	Number of components at each Level of Purdue	Probability of successful attack at Level
IT section of architecture Administrator	A (AHP, AD)	5	10**	0.32*
IT section of architecture Employee	E (EHP, ED)	4	100**	0.32*
DMZ- Data Historian	H (HHP, HD)	3	2**	0.2*
DMZ- Patching Server	G (GHP, GD)	3	2**	0.2*
DMZ- Application Server	R (RHP, RD)	3	2**	0.2*
DMZ-Remote Access Server	T (THP, TD)	3	2**	0.2*
SCADA controller	C (CHP, CD)	2	1**	0.3*
SCADA architecture PLC	P (PHP, PD)	1	6**	0.5*
SCADA Sensor	S (SHP, SD)	0	240**	undefined

Table 4 model components and probabilities treatment plant layout

The listed probabilities (*) were obtained from (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013) for PLC probability used HMI probability. The number of components (**) are notional for a hypothetical water treatment plant. In the model, the "real system components at each level are A, E, H, R, G, T, C, P and S. Honeypots have an HP at the end and Decoys a D. The last two columns of the Table indicate how the complexity of the honeypots (HP) and decoys (D) will be varied from 0 to 1 (100% fidelity/complexity, non-detectible) during the hypothesis testing. (Zhenxin Zhan, Maochao Xu, & Xu, 2013) indicate through their research that honeypot fidelity impacts their detection by attackers and that the probability of detection varies from 15 to 85%.

The complexity of the decoys will be varied from 0 to 100%, with 100 % indicating that the decoy is indistinguishable from a real system component and the smaller the percentage the more likely that a threat will recognize the decoy as a PLC or SCADA component which is not actually used as part of the control system of the CI, but incorporated into the system for deception.

3.2 Modelling attacks

The attack tree depicted in Figure 12 illustrates the progression of the attack from the uppermost layer to the lowest level of the Purdue Model.

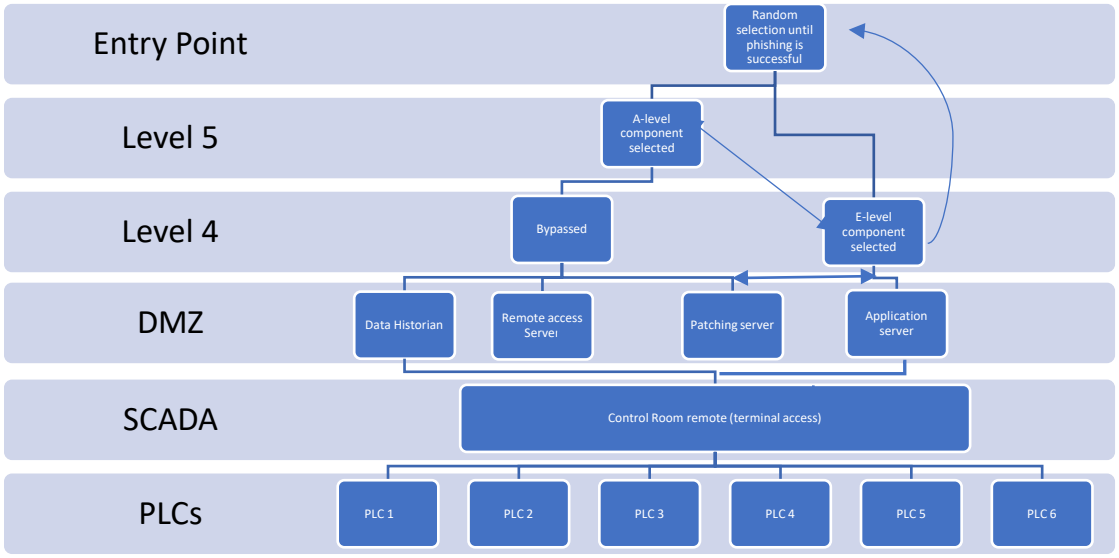


Figure 12 Attack Tree of model.

The attack is modeled as a Markov Chain (Bento, 2020) from the top of the system architecture to the bottom. The advancement of the attacker from one layer to the next is a function of the success of the attack at that level. Success of an attack phase is a function of the stochastic selection of components at that level combined with the

probability of success at that level (probabilities obtained from (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013)). (Zhenxin Zhan, Maochao Xu, & Xu, 2013) propose a stochastic process to model attacks. This is an acceptable methodology because cyber-attacks are “point processes” due to the natural abstraction of the cyber-attack process (Zhenxin Zhan, Maochao Xu, & Xu, 2013).

The number of iterations to a successful breach of a layer is collected and is a function of probability and iterations to successfully access a real system component at that level. The number of iterations is a proxy for the time taken by the attacker to breach each level of the model. The collective time to breach the OT levels from the IT layers, in the model, is considered the Mean Time to Breach. This is not compared to literature because this information is not available except through expert elicitation and often red and blue teams do not disclose this information. Bobbio et al. estimate the total time of an attack to 75 days (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013). This is different to data collected by various IDS and presented in literature (Morris, 2018). In all cases, the data generated by this is simulated and is measured against a baseline model to compensate for the fact that real APT data was not available to publish.

The attack tree depicted in Figure 12 illustrates the progression of the attack from the uppermost layer to the lowest level of the Purdue Model. A similar attack tree and model layout is used by McQueen et al. (McQueen, Boyer, Flynn, & Beitel, 2006). The detailed attack tree depicted in Figure 13 is similar to the model-based attack graph visualization developed by literature reviewed and presented by (Ibrahim, Al-Hindawi, Elhafiz, Alsheikh, & Alquq, 2019).

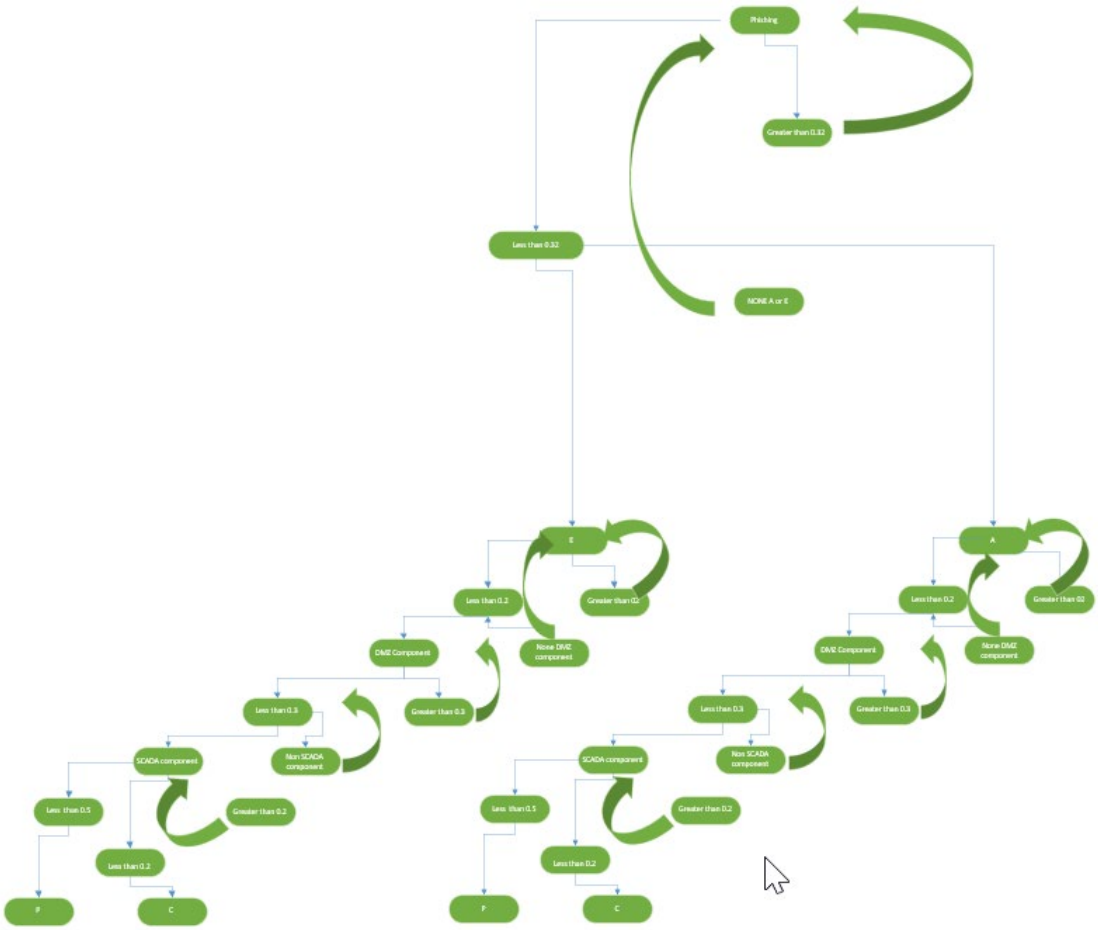


Figure 13 Detailed Attack Tree of model (including probabilities) developed using Microsoft Visio

However, the attack graph in Figure 13 is based on the Purdue System architecture and on the probability of a successful or failed attack as presented by Bobbio et al. (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013). The attack process presented in Ibrahim et al. (Ibrahim, Al-Hindawi, Elhafiz, Alsheikh, & Alquq, 2019) contains several of the system components presented here which comprise an IT OT system architecture. However, Ibrahim et al. do not base their attack on the Purdue System architecture, they use ones for hypothetical nuclear power plant and vehicular network system, and are not used here. The steps below detail the Cyber Kill Chain within the model.

3.2.1 Cyber Kill Chain of Python Model Attacks

Entry Point-Phishing attack: Attacker searches for a phishing entry point (32% chance of success, (Security, 2022))-this is a simulation of the packet exchange at the IT firewalls to enter the IT side of the enterprise. This part of the process is randomized and can be likened to the random selection of marbles from a bag with only two choices considered successful, an A or an E.

STEP 1: Attacker enters the infrastructure at ***Level 5 or 4***. If the successful entry point is at Level 5 component **(A)**, the attacker proceeds to create a foothold in the architecture by making his way to Level 3 and attempting to gain more credentials to enable access to the SCADA. If the entry point is a successful phishing attack on a Level 4 the attacker must make a lateral movement to Level 5 to gain escalated privilege and continue to access deeper layers of the Purdue Model. Assumption: Components at Level 5 have higher privilege compared to Level 4. The model uses the same form of escalation of privilege from levels of lower credentials to that of a higher credential as described by (B. Galloway and G. P. Hancke, 2013).

STEP 2-DMZ Firewall: Attacker penetrates the fire wall at the DMZ with a 20% chance of success (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013).

STEP 3-Foothold: Attacker creates a foothold in DMZ with a 30% chance of success (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013). At the DMZ the attacker searches randomly and can access Remote Server, Patch Management Server, Data Historian and Application Server. Access of any of these components allows the attacker to begin an attack on the SCADA.

STEP 4-SCADA Firewall: Attacker attempts to breach fire wall between DMZ and SCADA with a 30% chance of success (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013).

STEP 5: Attacker launches SCADA attack.

STEP 6: Attacker accesses PLC and HMI and controls signals sent to upper levels and masks field data. Masking is not simulated. Success is capturing the flag at this level or the level above.

3.2.2 Relationship between layer components:

There are three probabilistic relationships that can be used to explain selection of system components, prior to the incorporation of decoys and honeypots into the system architecture.

At the IT level, in the phishing attack phase, the attacker can either select an A component with a 32% chance of success or an E component and then an A component with a 32% chance of success.

Within the Demilitarized zone, the attacker selects from all four of the components with an equal weight (probability) and a 30% chance of succeeding in breaking through that level. The Markov effect comes into play here where the probability of the attacker breaking through an creating a foothold is contingent upon having broken through the IT levels (phishing attack). It is a memoryless process because the probability of the phishing, itself, does not influence the probability of the subsequent level of the attack.

Similarly, component selection at the SCADA levels is based on the mutual exclusive relationship between the components of the levels (PLC and SCADA). The attack

success probability is $p \leq 0.3$ for SCADA and $p \leq 0.5$ for PLC and although the attacker's success in breaching the levels is contingent upon the attacker succeeding at the phishing a foothold attack, the success at the SCADA levels is memoryless of the prior attacks and their probabilities of success.

Within the model, a split occurs at the access points to the different levels of the attack. This is caused by the probability of success or failure in accessing that layer. In Equation 1, for example, this is shown by the less than or equal to signs and the 0.32. This dictates the percentage of the attacks which are successful and the remaining 68% are failed. Figure 14 below, illustrates this split within the code where the conditions which must be satisfied for a successful phishing is that a real system component is selected and only 32% of the attacks can be successful. It is important to note however that as the code is currently written 32% "OR" less of the phishing attacks will be successful in making it through Levels 5 and 4.

```
if ((phishing == 'A') or (phishing == 'E')) and (random.random()>0.32):
    phishing_trials = phishing_trials + [phishing]
    trials['phishing'] += 1
    continue
```

Figure 14 Code illustrating 32% divide at phishing phase of attack

In the Bare Model, the probability of selecting an A component or an E component (P(A) or P(E)) = ½; the probability of selecting an H, T, G or R components (P(H) or P(T) or P(G) or P(R)) = ¼; the probability of selecting a C or P component (P(C) or P(P)) = ½.

When the escalation of privilege between the layers of the IT- side is added, this only impacts the phishing attack making its probability equal to *P(A) or P(A)|P(E).* The

probability of a successful attack/SCADA breach commencing at Levels 5 or 4 and capturing the flag at Level 2 or 1 is P(Breach).

3.3 Data Collected from Model:

Model Level	Python MODEL Data collected at Level	Metric	Backend Metric collected
Level 5 and Level 4	Phishing	Number of iterations until an A or E component were selected	Components attempted to be accessed during phishing attack iterations
Level 4	1st attempted attack	Enterprise level component successfully breached after Enterprise levels are breached (A or E component)	
Level 5	Enterprise level breached	Enterprise level component successfully breached after lateral movement (always an A)	
	Firewall 1 Penetration	Number of attempts until firewall penetrated (based on probability)	N/A
Level 3	Foothold	Number of iterations until a foothold was created and DMZ accessed	
Level 3	DMZ breach	DMZ level component which was breached (H,G,R,T)	Components attempted to be accessed during phishing attack iterations
	Firewall 2 Penetration	Number of attempts until firewall penetrated (based on probability)	N/A
Level 2	Scada Access	Number of iterations until SCADA was successfully breached	
Level 2	SCADA Breach	SCADA component which was breached (C or P)	Components attempted to be accessed during foothold attack iterations
Level 1	Command & Control Accessed	Iterations until signals between PLC and field equipment intercepted	
Level 0	Sensor	Field equipment controlled	Components attempted to be accessed during SCADA attack iterations

Table 5 Data collected from Python model

Table 5 summarizes model outputs as collected in the Main and Backend files. Figure 15 presents the portion of the code where the Main and Backend files are created to collect the model data.

```
table = pd.DataFrame.from_records(many_trials)
table2 = pd.DataFrame({'phishing attempt':pd.Series(many_phishing_trials), 'foothold attempt':pd.Series(many_foothold_trials), 'scada trials':pd.Series(many_scada_trials) })
table.to_csv('PraxisData-BaseRun Final.csv')
table2.to_csv('PraxisData-BACKENDDATA-BaseRun Final.csv')
```

Figure 15 Lines of code which tabulate the data and save it to Main file and Backend file.

The model is designed so that success is only achieved when real system components are accessed. This decision was made so that data could be collected on the variation in Mean Time to Breach (MTTB) with increasing deceptive complexity. This means that when a decoy or honeypot are selected, the model collects this data in the BACKEND Data file, rather than in the Main output file where the MTTB each layer and the attack path data are collected. The code can be modified to collect data on the attack path to include deceptive components. Data is collected on the number of iterations it takes for success (capture the flag at the OT levels) to be achieved. Model data is summarized below:

1) **Mean Time to Breach (MTTB)** is mathematically represented by the following equations:

   MTTB = *Number of iterations to a successful phishing attack + Number of iterations to pass Firewall + Number of iterations to create a foothold + Number of Iterations to a successful access of SCADA or PLC + Number of iterations to access Command & Control Signals (field equipment access)*

   This value is calculated from the Main File through a summation of all the elements of equation 1

2) **Attack Path**: This is the system component accessed in the Phishing attack + Component accessed in the foothold attack + Component access in the SCADA attack + Component accessed in Field equipment access

   Example: A-H-P-C

3) **Percentage Success:** This is calculated by dividing the number of successful attacks (1000) by the total number of attempted attacks. This data is obtained from the BACKEND File where the attempted attacks within each phase of the attack are complied

3.4 Model Assumptions and Limitations:

The following are assumptions made when designing the model and writing the code:

1) Attacker randomly attack/accesses components within each level
2) Movement between levels is targeted, rather than random, and is based on the successful breach of the previous level as in a Markov chain.
3) Success/Capture the flag is at the SCADA or PLC level
4) Movement within layer and selection of a component is randomized

The following are the limitations of the model:

1) The SCADA and PLCs components are the only components represented on the OT side of the architecture because literature indicated that these are the components which are targeted during an attack.
2) Lateral movement within the Enterprise Level allows for an attacker entering at Level 4 to gain access privilege and happens at Level 5 (escalation of privilege)
3) System components A have “higher access credentials” than E components

4) All components within DMZ layer (Remote Server, Patch Management Server, Data Historian and Application Server) allow equal progression privilege to SCADA level

The model allows for the following variations with minor changes in code.

1) Change of capture the flag location is allowed with minimal coding
2) Probability of success or failure at each level can be modified based on calculation of risk
3) Number of components at each level can be modified
4) Type of components at each level can be modified
5) Probability of accessing components at each level can be modified
6) Probability of selection of components can be modified

3.5 Model Validation and Verification

The model was run through various testing scenarios to verify performance and validate that it achieved the set requirements of being a tiered model, based on the Purdue System Architecture. The model performance is verified through a series of edge cases and Classification and Regression Trees (CART) analysis. The edge cases are set up to demonstrate that the model performs as it is designed to. They demonstrate the complexity of the model, as compared to the Bare Model, the reliability of the data it produces and its comparability to research in the field. The CART analysis demonstrates the performance of the model through the decision tree created from the analysis of the data generated through the simulations. The Classification trees show that the model splits at the correct nodes and that the attack progression. The Regression analysis

illustrates that the model splits at the correct probabilities, as explained in Section 3.2.2. All statistical analysis was conducted using MINITAB®. The conditions of the edge cases are summarized in Table 6

Table 6 Edge Case used for model verification

	Purpose	**Components in model**	**Objective**
Edge Case 1	Establish bare model performance,	All components are real A= E= H=R=G=T=C=P=S=1 without probabilities in place	To demonstrate that relationship between model layers matches expected results
Edge Case 2	Establish impact of quantities	All components are real A:10, E:100, H=R=G=T:2, S=1, P=6, S=240	Hypothetical scenario based on hypothetical critical infrastructure. Expected to show how increase in numbers of a component influence model
Edge Case 3	Demonstrate the impact of probabilities	All components are real A:10, E:100, H=R=G=T:2, S=1, P=6, S=240 with probabilities in place	IT phishing probabilities: 0.32 ((Security, 2022) DMZ penetration probabilities :0.3 (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013) SCADA breach probabilities C=P =0.3 (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013)
Edge Case 4	Impact of Firewalls	A:10, E:100, H=R=G=T:2, S=1, P=6, S=240, With probabilities and Firewalls in place	Firewall penetration probabilities (0.2) (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013)
Edge Case 5	Impact of Escalation of Privilege	Movement from Level 4 to Level 5	Lateral movement design

As previously discussed, there is little research that presents on the topic of the probability of breaching each level of the Purdue model, the time it takes for an APT actor to breach each layer of the IT/OT infrastructure and the impact on components. This research strives to contribute to the body of knowledge through the modeling in this

work. Bobbio et al. (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013) estimated, through a risk analysis of the cost, impact and likelihood of an attack on a CI, the probability of breaching various levels within IT and OT systems. These probabilities were used in the Praxis model to make the output of the simulation as realistic as possible to results of a real cyber-attack. In a real-life scenario, the probabilities of an attack and those of component access, estimated through a risk assessment of the infrastructure, can be used to determine the resilience or vulnerabilities within a system. The values of the probabilities to beach the firewalls was obtained from statistics available through (Security, 2022) which presents on numerous cyber security statistics collected from the industry.

For this research, the honeypot complexity /detection probability presented by Surnin et al. were used (Surnin, et al., 2019), who also discussed the use of research and production honeypots and their detection probability. Surnin et al. present that the complexity of the honeypots is proportionate to the likelihood of their detection. Surnin et al. identify the probability of honeypot detection as ranging between 15 and 85%; we used honeypot detection probabilities ranging between 15 and 85%, specifically 15, 40 and 85%. Additionally, Gore et al. (Gore, Padilla, & Diallo, 2017) describe the importance of cyber modeling which is more than simply identifying vulnerabilities and recommends the use of Markov models to simulate cyber-attacks; the use of stochastic modeling to model attacks on critical infrastructures is described by (Zhenxin Zhan, Maochao Xu, & Xu, 2013). The model in this Praxis uses a combination these methods. For the overall progression of the attack from Level 5 to Level 0 of the model, a Markov chain was used where progression from one layer to the next was a function only of

success at the previous layer. The stochastic element of the attack was used within layers for the random attack/selection of component access.

In addition to the Edge cases used primarily to explain the model behavior and compare and contrast it to models that may be available or proposed in literature a series of Defense Scenarios were run, showing the impact of location of the honeypots and decoys on the model behavior. The results of these runs are summarized in Chapter 4 and a summary of the conditions is presented in Table 9, in the following subsection.

3.5.1 Edge Cases

The Edge Cases were used to test the bounds of the model and to demonstrate that it performs as it was designed to perform. These simulations aided in the debugging of the code because they allowed the designer to observe faults in the code when the output was different to that expected. The edge cases are set up in an order to allow the reader to visualize how the model was built progressively from all components being equal in quantities; to the influence of varying the quantities: to adding probabilities of success and failure to the different layers; to adding firewalls and lastly demonstrating the impact of adding an escalation of privilege in the code forcing the attack paths down certain routes.

Edge Case 1 (Impact of proportionality):

This edge case is used to demonstrate how the proportionality of components to one another influences the model output. The probability of selecting a component from each level is based on their proportionality:

$$Phishing\ (ITcomponents): \left(\frac{1}{2}\right), Foothold(DMZ\ components) : \left(\frac{1}{4}\right), \text{SCADA(components)}: \left(\frac{1}{2}\right)$$

When analyzing the results, these values can be calculated by taking the ratio of successful attacks (always 1000) and dividing it by the total number of phishing attacks, foothold attacks or SCADA attacks, obtained from the simulations. It is expected in this run that the output would present that the number of A components is equal to that of the E components; that the number of G, H, R or T components is equal to one another and that the number of C components is equal to the number of P components, which appear in the output.

The exact value of this probability will vary between runs due to the stochastic process associate with the Phishing, Foothold and SCADA attacks and also the variation in the number of successful or failed attacked between runs.

Edge Case 2 (Impact of quantities):

In this Edge Case the impact of quantities of components on the output of the model is assessed. The impact of quantities is observed by counting the number of successful selections of components within a level of the model.

For example, for the foothold attacks, number of components will always be the same because the quantities of components are the same: H=G=R=T.

For phishing on the other hand, A: E = 10:100 so E will always be 10-times as contributing to a phishing attack as A (demonstrated through countif(number of Es in the population of phishing attacks)

As the probability of selecting a real system component is always 100%, the proportionality of real system components to one another is: A: E = 10:100, H: G: T: R =2:2:2:2, and C:P = 1:6.

Edge Case 3 (impact of probabilities of attack phases):

This case demonstrates the impact of the probability of success or failure assigned to the different phases of the attacks, as opposed to Edge Cases 1 and 2 which have no probabilities incorporated. The expected probability at each level of the attack is calculated using Equations 1-3 which incorporate the proportionality of the components in the Level of the Purdue architecture and the probability of success of breaching the Level.

The Calculated probabilities are presented in Table 7.

Attack Phase	Probability of Success of Attack	Average Expected Probability of Success
Phishing	≤ 32%	2.6%
Foothold	≤ 30%	5%

SCADA	≤ 30% for C, ≤ 50% for P	23%

Table 7 Impact of Probabilities (Expected)

These calculated values will be compared to the number of successful attacks as collected from the BACKEND DATA file.

Edge Case 4 (Impact of Firewalls):

This edge case demonstrates the impact of adding firewalls to the system. The firewalls in this model are modelled as dampers which slow the attacker, but do not prevent the attack. This was designed as such so it is representative of real attacks and as presented in (Cook, Janicke, Smith, & Maglaras, 2021) and the probability of success /failure of a firewall is obtained from (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013) and confirmed through (Security, 2022).

Firewall performance is verified through varying the probability of failure of the Firewalls from 0 to 1 and monitoring the results (number of attacks that pass through compared to prior layer). When the probability of failure is set to 1, the simulation runs endlessly after selecting the very first phishing component. When set to zero there are no firewall iterations. The Firewalls are designed to allow less than 20% of the attacks to pass through to the next level of the attack.

Edge Case 5 (Impact of Escalation of Privilege):

Edge Case 5 demonstrates the impact of the escalation of privilege. Within the model, and similar to the set up designed by Cook et al. (Cook, Janicke, Smith, & Maglaras, 2021) and (Caltagirone, Pendergast, & Betz, 2013). The escalation of privilege is to

acquire a higher level of credentials in order to move towards a more successful attack. In the model, this is demonstrated through the movement from Level 4 to Level 5 of the Purdue architecture and model. This impact is demonstrated by commenting out the lines in the code which force the attacker to not be able to move to the DMZ without an escalation of privilege. When these lines are commented out the attacker can move freely from Level 4 to the DMZ. In the output this is demonstrated by the enterprise level attack being only an A which the attempted phishing attack being and A or E. Lateral movement within the model makes the attack path longer and in real life impacts the time creating a foothold as it does within the model.

Within the model results, escalation of privilege is demonstrated through the column of results labelled as Enterprise Level Breached. This column will post an A or E when escalation of privilege is removed. When escalation of privilege is in place the 1st attempted attack column of the results will display an A or E. Escalation of privilege is demonstrated to result in a longer time to breach the uppermost enterprise levels through the phishing attack and the Mean Time to Breach as a result.

3.5.2 Classification and Regression Tree (CART):

The CART analysis of the model was used to verify that the model performed as a Markov Chain as it illustrates that the model progressed from the phishing to the foothold and then SCADA levels. The CART analysis also shows how within each layer of the attack there are splits that relate to the probability of success at that level of the model. In this way, the analysis was used to assess the performance of the model and as a last step validation of the data obtained from the simulations.

3.5.3 Base Model:

This is the final form of the model against which the hypotheses are tested. The conditions are presented in Table 8.

	Real System Components	Honeypots (HP)	Decoys Location
Base run	**IT:** A=10 E=100 **DMZ:** H=G=T=R=2 **SCADA:** C=1 P=6 S= 240	No Honeypots	No decoys

Table 8 Base Model Components

3.5.4 Defense Scenarios

The Defense Scenarios are a series of simulation used to demonstrate the impact of the location of the Decoys and honeypots. Table 9 below summaries the conditions of these simulations.

	Real System Components	Honeypots (HP)	Decoys Location
Defense Scenario 1	Same as Base run	HP on IT side only	On IT side only
Defense Scenario 2	Same as Base run	HP throughout DMZ	Throughout DMZ

Defense Scenario 3	Same as Base run	On OT-side only	On OT side only
Defense Scenario 4	Base run	Distributed throughout architecture	Distributed throughout architecture
Defense Scenario 5	Base Run x 10	No honeypots	No decoys
Defense Scenario 6	Same as Base Run	HP=Decoys =Real x 10	HP=Decoys =Real x 10
Defense Scenario 7	Same as Base Run	HP=Decoys =Real x 100	HP=Decoys =Real x 100
Defense Scenario 8	Same as Base Run		PLC script deception demonstration

Table 9 Details of attack scenarios

The outcome of Defense Scenarios 1 to 3 are not presented, but are summarized in Chapter 4 because although they were used to optimize the location of the honeypots during model testing and validation, they are special conditions and do not all test the main Praxis hypotheses. Defense Scenarios 4-8 are the same as those used for establishing baselines (Defense Scenario 4) and testing the hypotheses (Defense Scenarios 5-8).

3.6 Hypothesis Testing:

- **Hypothesis 1 (H_1):** *Time to successful cyber breach increases with increase in complexity of decoys in a system.*

  Hypothesis 1 is tested through the variation of decoy complexity between 0% and 100% and documenting the model response. To test this hypothesis MINITAB will be used to conduct the statistical and graphical analysis. Two- sample T-tests will be used when comparing two means, ANOVA tests to compare results of

three or more simulations, and a CART analysis to demonstrate that the model runs as expected and to support model behavior validation. Complexity is reflected in the model through the probabilities assigned to the decoys and honeypots, of selecting a decoy. This is analogous to the sophistication with which a decoy, or honeypot, is designed to mimic and interact as a real system component. (Surnin, et al., 2019) presents that honeypots can be detectable with an 85-100% chance of success. The complexity of honeypots was set to 40%, however was varied in some tests between 0 and 1.

- **Hypothesis 2 (H_2):** *Decoys incorporated into layers of a critical infrastructure system architecture will reduce the probability of successful cyber intrusions.*

To test this hypothesis, EXCEL and MINITAB will be used to conduct the statistical and graphical analysis. Comparing the number of successful attacks (1000) to the total number of each of the phishing, foothold and SCADA attempted attacks, as model complexity is increased yield results of H_2 testing. It is important to note once again that the Python code is written to make the model converges to a success with each run and this is how the number of iterations to a success is counted.

4 CHAPTER IV: RESULTS

This Chapter presents the results of the simulated IT/OT model Section 1 presents the results of the Edge Cases used to verify the model and establish its capability. Section 2 presents the results of the Base Case against which the hypotheses are tested. The Base Case is the final case of the Edge Cases Scenarios. Section 3 presents results of Hypothesis 1 and 2 testing.

4.1 Edge Cases and Model Verification:

The results of the edge cases described in Chapter 3 are presented in this section.

Edge Case 1 (Bare Model):

In the first run of the simulation all components were removed from each layer and only one was kept at each level. At Level 5 the only component was a (A). In the DMZ Level the only component was an (H) and at the SCADA Level the only component was a (C). The output is illustrated in the sample output of Table 10.

Trials	Phishing	Enterprise Level Breached	Foothold	DMZ breach	Scada Access	SCADA breach	Command & Control Accessed	Sensor
0	1	A	1	H	1	C	1	S
1	1	A	1	H	1	C	1	S
2	1	A	1	H	1	C	1	S
3	1	A	1	H	1	C	1	S
4	1	A	1	H	1	C	1	S

Table 10 Sample output of Bare model run

The Table presents a sample from the 1000 runs. All runs were identical verifying that there is no error in the model and no probabilities or weights influencing the output of the simulations. This demonstrates that the model connects Level 5 to Level 0. This also

demonstrates that there is no skew in the results; when one component is present only one component is selected and there is only one iteration to get from one level to another.

Edge Case 2 (impact of quantities of components)

The purpose of this edge case is to demonstrate the influence of quantities of components on the simulations and in doing so the results of a simulation with components quantities equal to one is compared with quantity variation.

When the Bare Model is run such that there is one component in each level of the model, Equation 1 resolves to ratio of **Phishing: Foothold: SCADA = 1:1:1** ***(calculated)***.

This is observed in the data collected from the "BACKENDDATA" file where out of a total number of 1000 runs **Phishing: Foothold: SCADA = 1000:1000:1000** ***(collected from model).***

When run with A=10, E=100, H=R=T=G=2, C=1, P=6 and S=240, the results are proportionate to the number of components at each level as shown in Table 10.

	A	E	R	T	G	H	C	P	S
Count in Population (1000)	107	893	260	249	235	256	140	860	1000
Model mean quantities	11.77	98.23	2.08	1.992	1.88	2.048	0.98	6.02	1
Assigned quantities	10	100	2	2	2	2	1	6	1

Table 11 Counts of components

Out of 1000 runs, 107 A components were selected, 893 E components, 260 R components, 249 T components, as presented in the Table ending with 1000 S components, it being the only component in Level 0.

Edge Case 3 (impact of probabilities of attack phases)

The results of different attack defense probabilities are presented in Table 16 below.

	Phishing	**Foothold**	**SCADA**
From model	69115	15164	4983
Successful/total attack (%)	1.45%	6.59%	20%
Calculated	0.026(2.6%)	0.05 (5%)	0.23 (23%)

Figure 16 Data obtained from Backend File versus calculated

The observed values are collected from the model output, in this case the BACKEND file. Here it is important to understand the cause of variation within the model. Observing the results obtained from one of the model runs compared to the calculated probabilities of success presented in Figure 16, the difference between the expected (calculated) and modeled is that using the random.random () statement in Python to select a probability of success between 0 and 32% for phishing success results in a variation in the probability distribution of the output equivalent to that range. Similar variation occurs with Foothold and SCADA by 30% in the case of foothold and 30% in the case of SCADA, for C components and 50% for P components. These variations are acceptable and with

running the model numerous times were always found to be within the same order of magnitude as the calculate probabilities of success.

Edge Case 4 - Impact of firewalls

Figure 17illustrates the impact of adding firewalls.

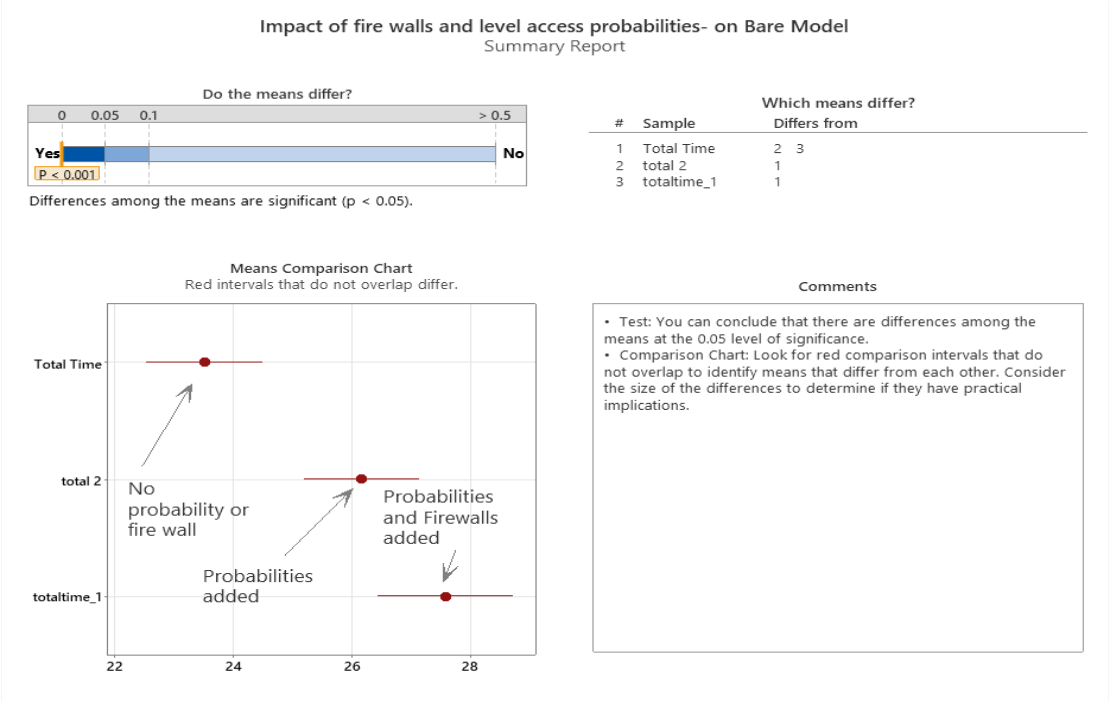


Figure 17 Comparison between impact of probabilities and firewalls on the bare model

The firewalls did not significantly have an impact on the mean number of iterations it took the attacker to breach a level within the architecture. This is because of the way each level of the attack is modeled as Markovian - depending on the parameters for that level of the attack only.

Edge Case 5 privilege escalation

In this case the impact of the escalation of privilege is demonstrated. The sample of results presented in Table 11 demonstrates the impact of escalation of privilege.

Phishing	1st Attack	Enterprise Level Breach	FW1	Foothold	DMZ Breach	FW2	SCADA Access	SCADA Breach	Command & Control Accessed	Sensor
214	A	A	19	112	G	39	48	C	5	S
67	E	A	6	40	T	7	13	C	2	S
27	A	A	4	11	T	3	5	P	1	S
46	E	A	3	25	G	6	9	C	2	S
15	E	A	0	5	T	2	4	C	1	S
123	E	A	14	49	H	15	16	P	2	S
56	E	A	5	17	T	9	10	C	3	S
186	E	A	12	80	R	30	42	P	4	S
20	E	A	1	6	T	4	5	C	1	S
84	E	A	8	35	R	6	10	P	2	S
291	A	A	20	105	G	26	31	C	3	S
233	E	A	12	90	T	31	41	C	6	S
100	E	A	11	48	T	19	28	P	2	S
40	E	A	2	17	T	6	7	C	1	S

Table 12 Snapshot of data

The Table's second and third rows illustrate the escalation of privilege in the cases where the first attempted attack is on a Level 4 component (E) followed by an attack on Level 5 (A). When counting the number of E components to the number of A components they are equivalent to the ratio of A:E in the population.

Classification and Regression Tree (CART) analysis

The CART analysis of the model yields the results depicted in Figure 18

Figure 18 Classification analysis of model data – 24. Node analysis

Through the levels of the trees and the branches the CART analysis demonstrates that the attack path starts at 1st attempted attack, branches off to Layers 5 (A-components) and 4

(E-components) and then continues to branch to DMZ components and then to the SCADA. The Attack path for E-components is shown to be longer as anticipated because users in level 4 have a lower access privilege forcing attacker to need and escalation of privilege to continue an attack started at level 4 of the Purdue system architecture/ Python model.

4.2 Base Models

The purpose of the base model simulation is to establish a baseline against which the hypotheses can be tested. The structure of the Base Model is as presented in Chapter 3. The Base Model is the Bare Model used in verification and validation with the all the probabilities, the firewalls and the real system components in place and the only variable are the numbers of honeypots and decoys and their complexities. The Base Model is used as a baseline to test the hypotheses and to demonstrate that with no high-fidelity decoys incorporated into the system architecture, an attacker will eventually, given time, be successful in hacking into a system and fully accessing the SCADA/PLC control systems. The model showed 1000 in 69115 phishing attacks will result in a 20% chance of successfully breaching the OT side of the architecture from the IT-side resulting. The descriptive statistics obtained through the MINITAB statistical analysis tool and the time series illustrating the variation in number of iterations (MTTB) for each phase of the attack are presented in Table 13.

Statistics

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median
phishing_1	1001	0	69.12	2.08	65.94	2.00	21.50	51.00
Firewall1Penetration_1	1001	0	35.39	1.13	35.63	1.00	9.00	26.00
foothold_1	1001	0	15.165	0.473	14.977	1.000	4.000	11.000
Firewall2Penetration_1	1001	0	10.562	0.344	10.896	1.000	3.000	7.000
scada_access_1	1001	0	4.9840	0.0996	3.1521	2.0000	3.0000	4.0000
Command & Control accessed_1	1001	0	1.0000	0.000000	0.000000	1.0000	1.0000	1.0000
MTTB1	1001	0	136.21	3.91	123.70	9.00	45.50	101.00

Variable	Q3	Maximum
phishing_1	96.00	442.00
Firewall1Penetration_1	50.00	271.00
foothold_1	21.000	105.000
Firewall2Penetration_1	15.000	73.000
scada_access_1	6.0000	22.0000
Command & Control accessed_1	1.0000	1.0000
MTTB1	185.00	812.00

Table 13 Descriptive statistics of Base Model

A significant finding presented in this table is that Phishing and FireWall1 make up 76.6% of the MTTB. This means that the attacker spends almost 77% of the time attempting to break through these layers.

The base model presents 1000 out of 69453 phishing attacks as having been successful, 1000/15403 foothold creation attacks and 1000/ 4978 attacks on OT components. This is a 1.4 % success for phishing, 6.5% for footholds and a 20% for SCADA/PLC attacks.

Figure 19 shows base model behavior during an attack. The MTTB for the Base model is 136 iterations.

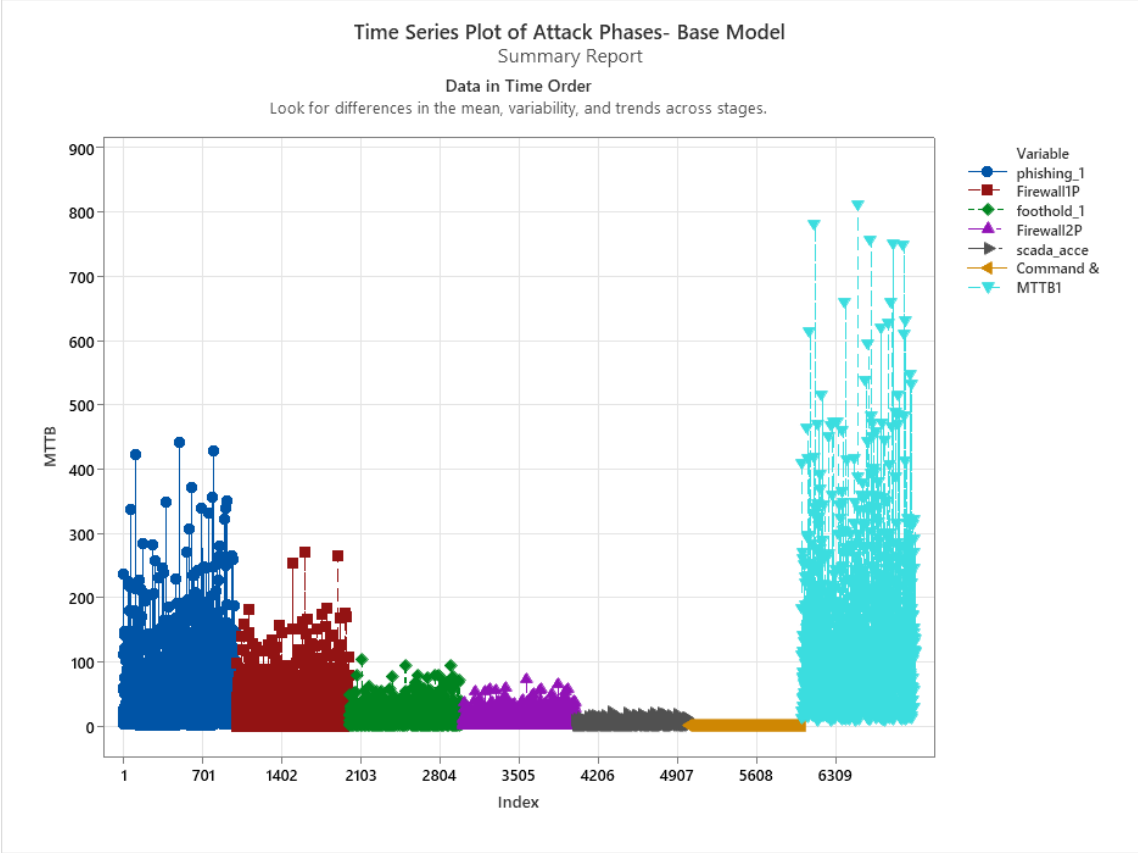


Figure 19 Time Series Plot of Base Model by Attack Phase and overall MTTB

The most important observation from the time series of Figure 19 is that attacker spends most of the time within the phishing phase of the attack and proportionately less time on the other two phases of the attack. Once successful in accessing the IT- layers of the Purdue Model it becomes easier for the attacker to gain access to the DMZ and in turn the OT layers of the Purdue architecture. This is consistent with results described by (Alladi, Tejasvi, Chamola, Vinay & Zeadally, Sherali., 2020) (B. Zhu, 2011). Although the Firewalls reduce the number of attacks, in the grand scheme of the attack scenarios, 20% of attacks in the 10 or 100 of thousands are still a significant number of attacks impacting the critical infrastructure.

4.3 Impact of Honeypot and Decoy Complexity on MTTB:

The results of this section present data in support of the Hypothesis 1 and Hypothesis 2. The results of MINITAB® statistical, graphical and machine learning modules are in Table 13 below.

Mean Time to Breach	MTTB1	MTTB	MTTB2	MTTB3
Honeypot Complexity	15%	40%	85%	85% +10 x Real System Components
Decoy Complexity	50%	80%	100%	100% +10 x Real System Components

Table 14 Summary of MTTB simulation conditions

4.3.1 Statistical and Graphical Analysis:

The time series of Figure 20 illustrates the increase in Mean Time to Breach between MTTB1, MTTB, MTTB2 and MTTB3 where there are 10 as many decoys and honeypots are real system components, in comparison to prior simulations.

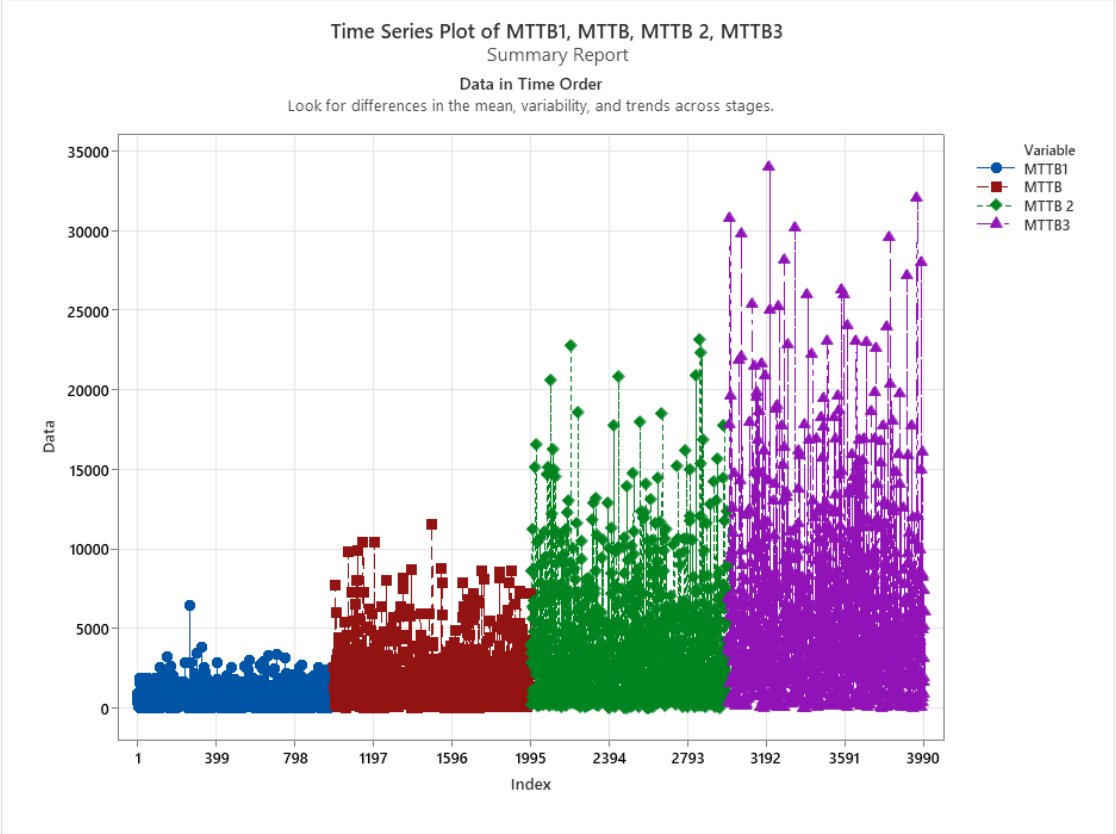


Figure 20 Impact of varying honeypot and decoy complexity as well as quantities of deceptive components on MTTB

MTTB1: HP 15%, D 50% MTTB: HP 40%, 80% MTTB2:85%, 100% MTTB3: HP (85% D 100%) x 10 as many decoys and honeypots are real system components

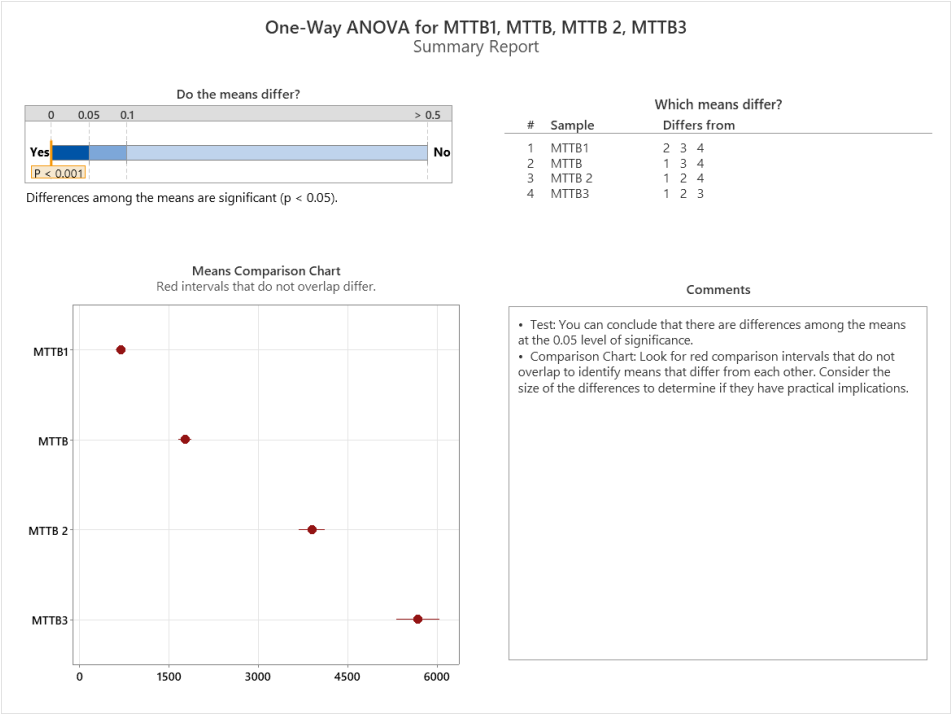


Figure 21 A: One-Way ANOVA of MTTB1: HP 15%, D 50%, MTTB: HP 40%, 80% MTTB2:85%, 100% MTTB3: HP (85% D 100%) x 10 as many real system components

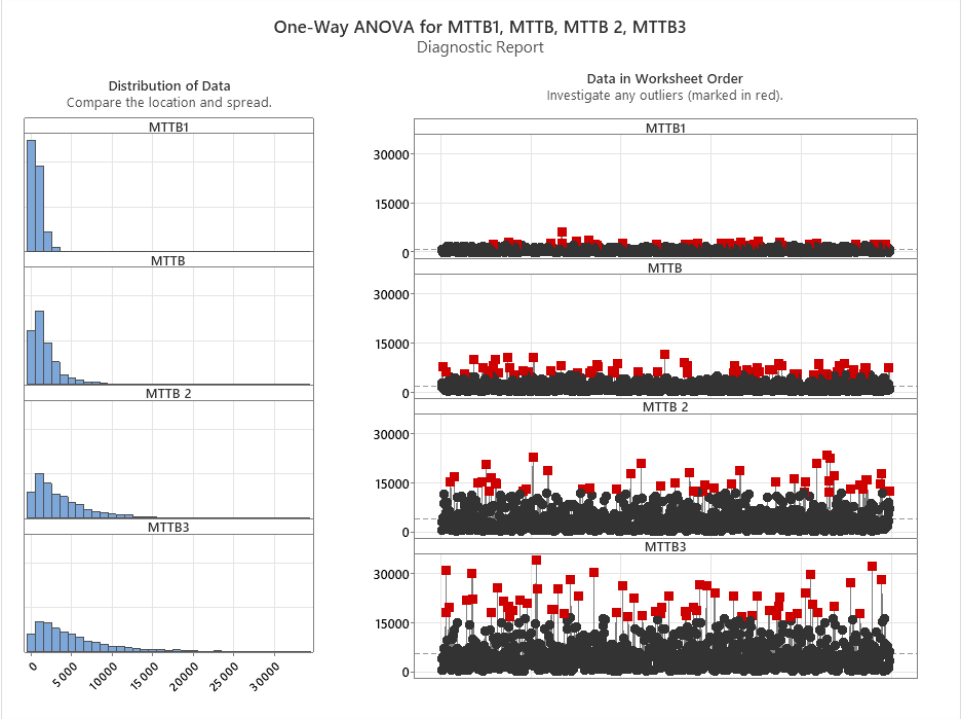
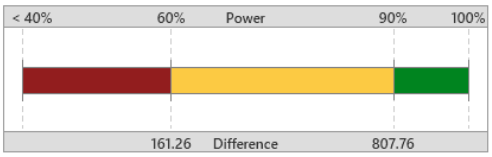


Figure 21 B ANOVA results MTTB1: HP 15%, D 50% MTTB: HP 40%, 80% MTTB2:85%, 100% MTTB3: HP (85% D 100%) x 10 as many real system components

One-Way ANOVA for MTTB1, MTTB, MTTB 2, MTTB3
Power Report

What is the chance of detecting a difference?



Based on your samples and α level (0.05), you have at least a 90% chance of detecting a difference of 807.76, and at most a 60% chance of detecting a difference of 161.26.

What difference can you detect with your sample sizes?

Difference	Power
161.26	8.5 - 60.0%
572.90	60.0 - 100.0%
634.90	70.0 - 100.0%
707.25	80.0 - 100.0%
807.76	90.0 - 100.0%

Power is a function of the sample sizes and the standard deviations. To detect differences smaller than 707.25, consider increasing the sample sizes.

Statistics

Sample	Sample Size	Mean	Standard Deviation	Individual 95% CI for Mean
MTTB1	998	698.29	658.45	(657.38, 739.19)
MTTB	998	1777.6	1791.6	(1666.3, 1888.9)
MTTB 2	998	3900.1	3817.4	(3663.0, 4137.2)
MTTB3	998	5680.0	5606.6	(5331.7, 6028.2)

Figure 21 C ANOVA results continued MTTB1: HP 15%, D 50% MTTB: HP 40%, 80% MTTB2:85%, 100% MTTB3: HP (85% D 100%) x 10 as many real system components

One-Way ANOVA for MTTB1, MTTB, MTTB 2, MTTB3

Report Card

Check	Status	Description
Unusual Data	⚠	Some of the data points are unusual compared to the others in the same sample. Because unusual data can have a strong influence on the results, you should try to identify the cause of their unusual nature. These points are marked in red on the Diagnostic Report. You can hover over a point or use Minitab's brushing feature to identify the worksheet row. Correct any data entry or measurement errors. Consider removing data that are associated with special causes and repeating the analysis.
Sample Size	✓	The sample is sufficient to detect differences among the means.
Normality	✓	Because all your sample sizes are at least 15, normality is not an issue. The test is accurate with nonnormal data when the sample sizes are large enough.
Equal Variance	ⓘ	Minitab's Assistant uses Welch's method, which does not assume or require that the samples have equal variances. Research shows that the test performs well with unequal variances, even when the sample sizes are not equal.

Figure 21 D ANOVA results continued MTTB1: HP 15%, D 50% MTTB: HP 40%, 80% MTTB2:85%, 100% MTTB3: HP (85% D 100%) x 10 as many real system components

Method

Null hypothesis	All means are equal
Alternative hypothesis	Not all means are equal
Significance level	α = 0.05

Equal variances were not assumed for the analysis.

Welch's Test

Source	DF Num	DF Den	F-Value	P-Value
Factor	3	1823.48	554.54	0.000

Model Summary

R-sq	R-sq(adj)	R-sq(pred)
22.96%	22.90%	22.81%

Means

Factor	N	Mean	StDev	95% CI
MTTB1	998	698.3	658.4	(657.4, 739.2)
MTTB	998	1777.6	1791.6	(1666.3, 1888.9)
MTTB 2	998	3900	3817	(3663, 4137)
MTTB3	998	5680	5607	(5332, 6028)

Grouping Information Using the Games-Howell Method and 95% Confidence

Factor	N	Mean	Grouping
MTTB3	998	5680	A
MTTB 2	998	3900	B
MTTB	998	1777.6	C
MTTB1	998	698.3	D

Means that do not share a letter are significantly different.

Table 15 Results of Welch's Test and Games-Howell.

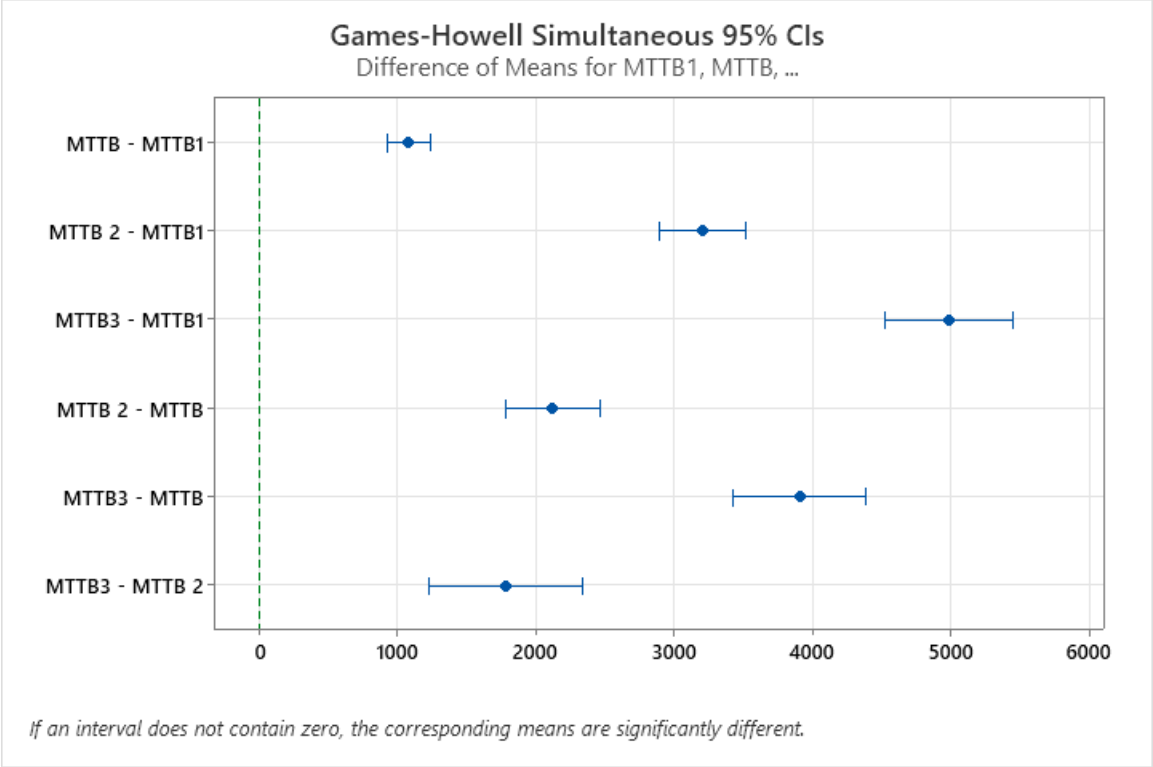


Figure 22 Games -Howell analysis

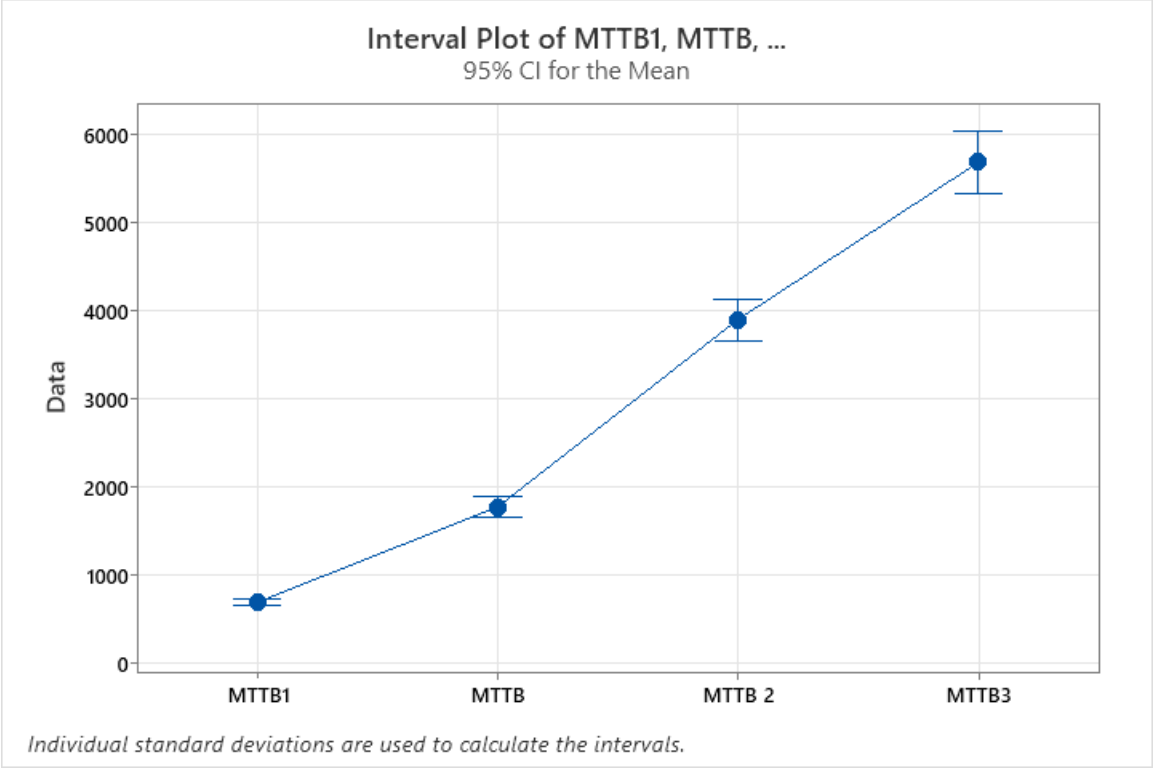


Figure 23 Interval Plot of MTTB with standard deviations used to calculate intervals

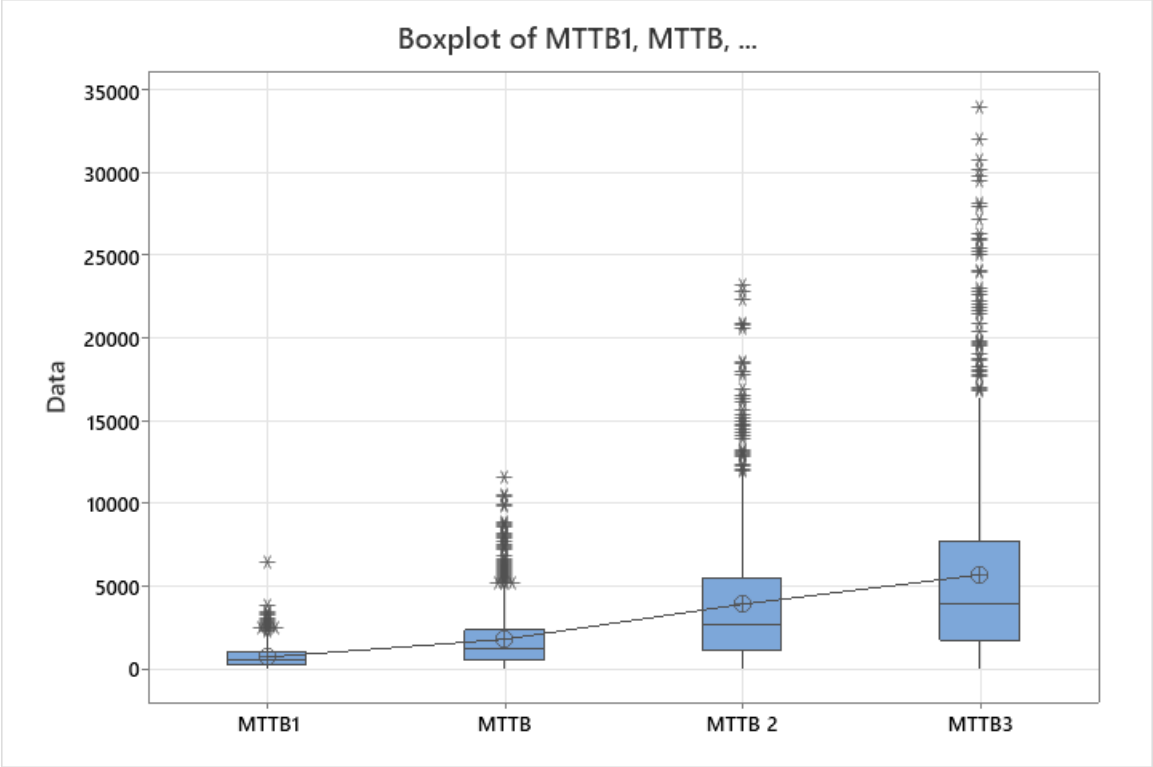


Figure 24 Boxplot of results of ANOVA test

Figure 21A-D, Table 15and Figure 22, Figure 23and Figure 24 present the results of the ANOVA MINITAB analysis used to demonstrate the impact of honeypot and decoy complexity is demonstrated on the Mean Time to Breach (MTTB). The analysis also demonstrates the impact of the increase in quantities of decoys and honeypots throughout the system architecture on the MTTB.

Complexity is represented in the model through the probabilities assigned to the deceptive components, the honeypots and the decoys. The complexity varies from zero to one for decoys, and from 0.15 to 0.85 for honeypots. The lower the probability assigned to a component, the more it is detectible by the attacker as not being a real system component.

The results illustrate that complexity increases the MTTB per attack phase and also the MTTB of the overall system increases proportionately from 689 to 3900 iterations. Results presented by Bobbio et al. estimate the average MTTB via Multi-Terminal Binary Decision Diagrams as 1800 hours and state that this is equivalent to 75 days (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013). Other differences and similarities with the Bobbio et al. model is that both models are based on decision trees and weights. The Praxis model incorporates the two IT-layers while the Bobbio et al. model models the SCADA and DMZ layers only. The One-Way ANOVA test yielded similar results to the Welsh and Game-Howell test allowing the acceptance of the Null hypothesis that there is a difference between the MTTB of the four conditions used to test the hypothesis. Figure 23 presents how the MTTB vary compared to one another and the boxplot results depict the expected variation in results which looks like outliers but is typical of the depiction of cyber-attacks in literature.

4.3.2 Classification Analysis: First Case with HP 15% complexity and decoys 50% complexity

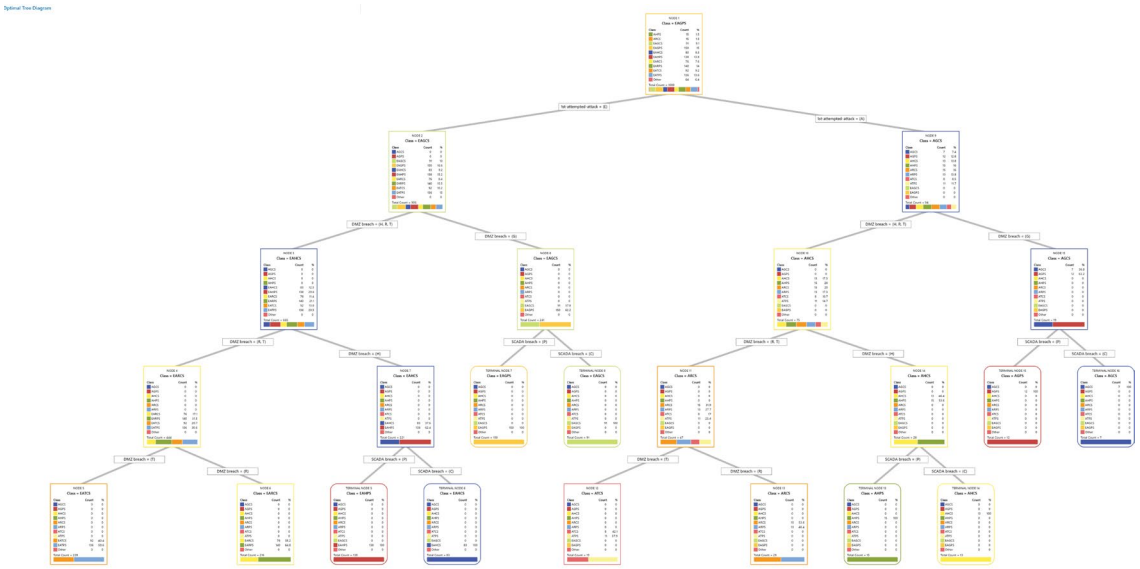


Figure 25 Classification Decision Tree analysis of Python Model

CART classification was used to analyze the categorical data and CART regression was used to analyze the numerical data. Table 16 Sample of Python model output used for CART analysis**Error! Reference source not found.Error! Reference source not found.**presents a sample of the data output from the model. The CART analysis used a K-fold cross-validation because the number of data points was less than 5000. Rows were randomly assigned to each fold to observe if the classification would read the model correctly. The output indicated that the model was classified correctly, starting with the 1st-attempted-attack, and ending with the SCADA breach. The Sensors were not included in the classification because all 240 sensors are named the same, 'S', and as such there is no split in the tree at that level.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Trials	phishing	1st-attempted-attack	Enterprise level breached	Firewall1Penetration	foothold	DMZ breach	Firewall2Penetration	scada_access	SCADA breach	Command & Control accessed	sensor
2	0	863	E	A	383	145	H	40	11	C	1	S
3	1	1040	E	A	337	141	G	39	18	P	1	S
4	2	10859	E	A	4031	1707	G	348	176	C	3	S
5	3	3934	E	A	1366	614	R	111	55	P	1	S
6	4	149	E	A	45	25	G	2	4	P	1	S
7	5	3094	E	A	1002	481	R	69	40	P	3	S
8	6	15713	E	A	5581	2564	H	640	278	P	9	S
9	7	8378	E	A	3137	1431	R	348	131	P	3	S
10	8	621	A		194	96	G	44	14	P	1	S
11	9	3994	A		1506	640	G	214	87	P	2	S
12	10	1194	E	A	456	204	H	32	16	P	1	S
13	11	5606	E	A	1894	900	H	208	89	P	3	S

Table 16 Sample of Python model output used for CART analysis

Method

Prior probabilities	Same for all classes
Node splitting	Gini
Optimal tree	Within 1 standard error of minimum misclassification cost
Model validation	10-fold cross-validation
Rows used	1000

Multinomial Response Information

Variable	Class	Count	%
Attack path1	AGCS	7	0.70
	AGPS	12	1.20
	AHCS	13	1.30
	AHPS	15	1.50
	ARCS	15	1.50
	ARPS	13	1.30
	ATCS	8	0.80
	ATPS	11	1.10
	EAGCS	91	9.10
	EAGPS	150	15.00
	EAHCS	83	8.30
	EAHPS	138	13.80
	EARCS	76	7.60
	EARPS	140	14.00
	EATCS	92	9.20
	EATPS	136	13.60
	All	1000	100.00

Table 17 Attack path classification by count and percentage of total attacks

Table 17 Also presents the results of the CART classification analysis. A summary of attack path, in addition to the count and percentage out of the totals attacks is presented. As can be seen attack paths beginning at the Enterprise Level 4 are more frequent that attack paths commencing at Enterprise Level 5. The lateral attack paths are also shorter

because they do not include an escalation of privilege between the lower levels of the enterprise and the upper ones. The Enterprise level 5 attacks are more frequent because there are more real system components at that Level of the Purdue system architecture.

This is of benefit to an engineering manager because this aid in devising defense scenarios allowing one to determine the best allocation of decoy resources.

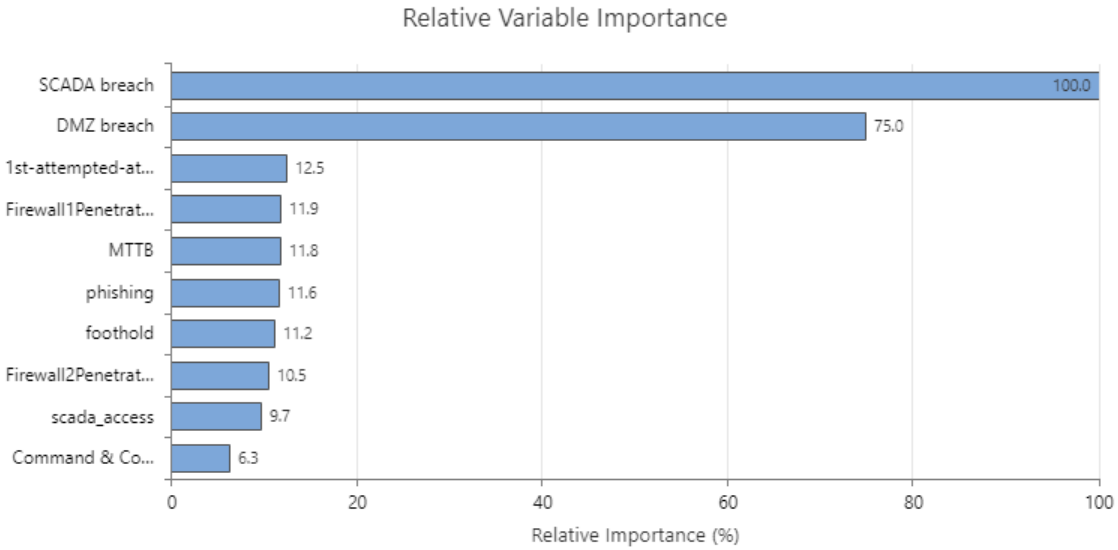


Figure 26 Relative Variable Importance as a percentage

Figure 26 presents the variables in order of their importance to the classification with SCADA breach being the variable of highest importance and 1-st attempted attack being the lowest amongst the categorical variables. The weight for the numeric variables was almost equal for the classification.

The results of the confusion matrix for both the training and testing data for a 10-fold K-cross validation indicate zero error in classification.

The area under the Receiver Operating Classification (ROC) graph for all attack path was 1 and the misclassification cost for both testing and training was determined to be zero.

4.3.3 Regression analysis First Case with HP 15% complexity and decoys 50% complexity:

This section presents the results of the CART Regression analysis. The regression analysis helped reinforce that the model was behaving as expected splitting and the correct nodes. In the case of the regression analysis the numeric output from the simulations was most impactful. Figure 27 through Figure 29 illustrate the results of the regression analysis which just like the classification analysis demonstrates that the model behaves as expected and that there is no abnormal or unexplained behavior.

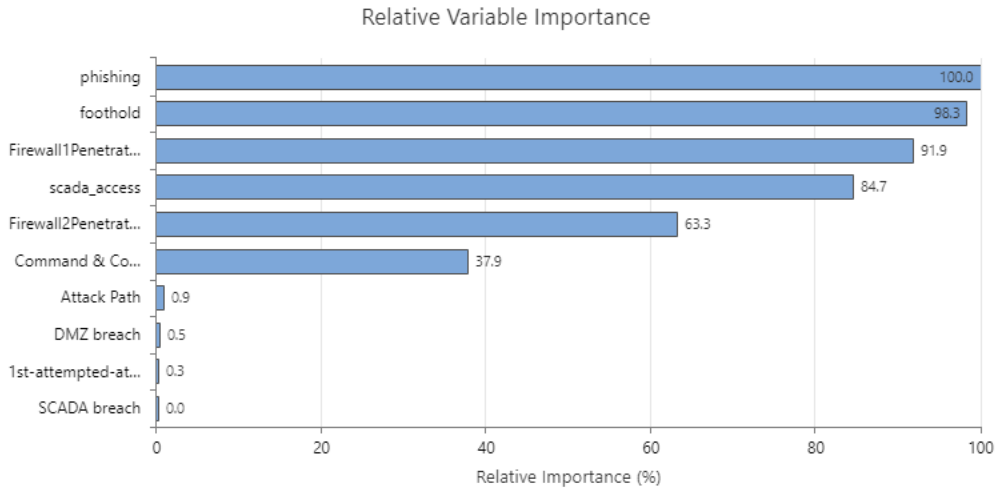


Figure 27 Regression analysis variable relative importance

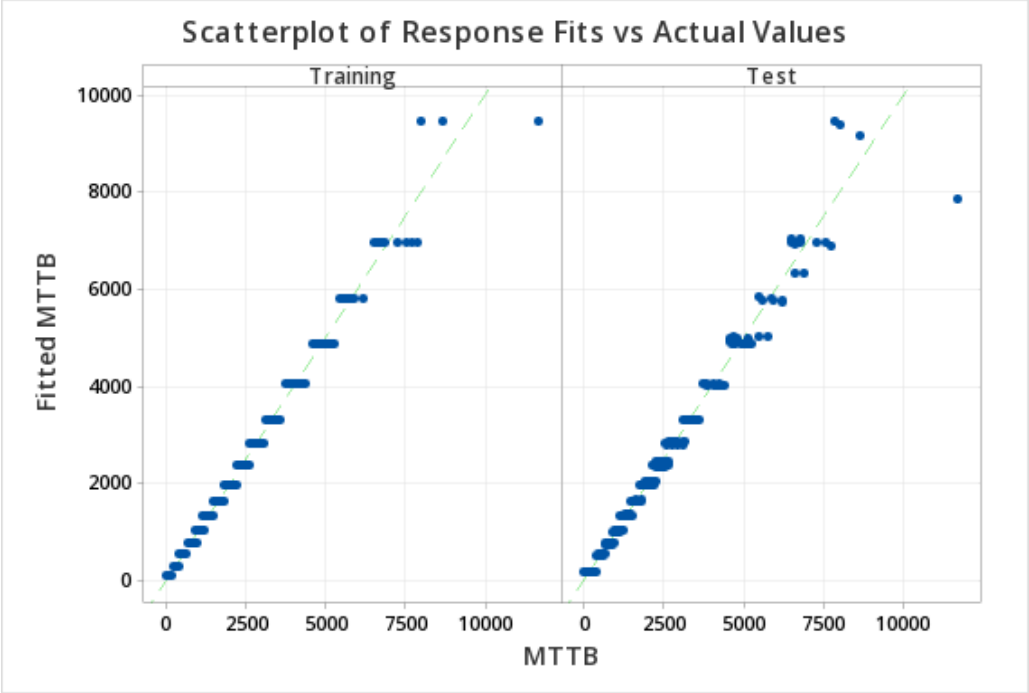


Figure 28 Scatterplot of Response Fits versus Actual Values

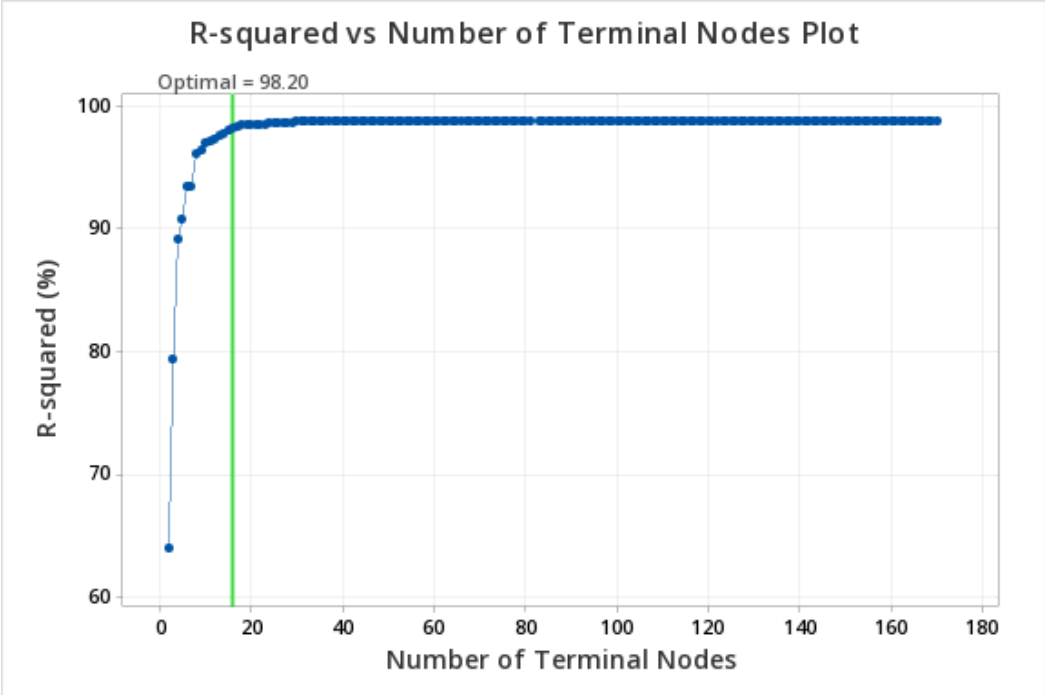


Figure 29 Plot of R-squared versus Terminal Nodes

The node splitting was based on least squared error. The validation, like the classification, was also a 10-fold cross validation and the optimal tree presented in Figure 30 is within 1-standard error of the maximum R-squared.

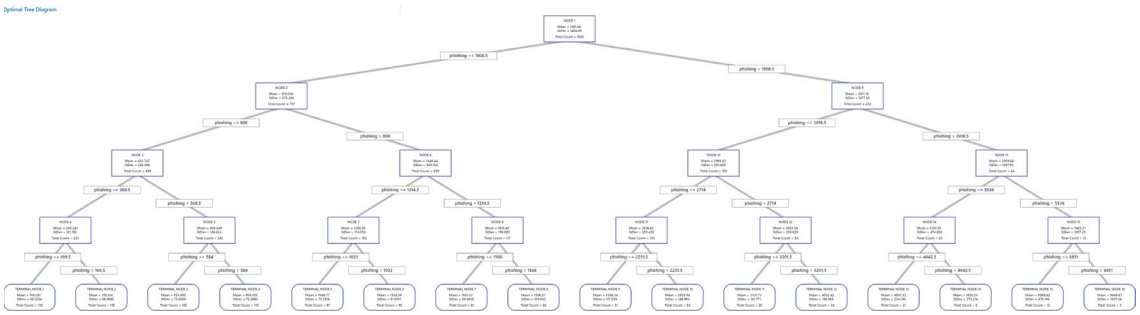


Figure 30 Optimal Regression Tree (CART Analysis MINITAB)- 16-Node tree

4.4 Impact of Deception on Component Access during an Attack

The BACKEND output data file obtained from the simulations, as mentioned previously, presents all the different components included within the model which were impacted by the attack. Figure 31, Figure 32 and Figure 33 illustrate the impact of adding decoys on the system components access during an attack. Figure 32 and Figure 33 present the impact of decoy and honeypot complexity on the components in Figure 31.

The results indicate that as deception is incorporated into the system architecture, the impact on real system components is reduced and this is proportionate to the decoy and honeypot complexity. This is observed in the series of bar graphs below, Figure 31 (A-C), Figure 32(A-C) and Figure 33(A-C). Looking at system components A, for example, it can be seen, by looking at the figures with an A-postscript, that as the number of A-

type decoys increases, the number of A components that appear on the list of components impacted by the attack (collected from Backend file data) is reduced indicating a lessened impact on that real system component, in the presence of the decoys. However, this effect is observed more significantly on the IT side of the infrastructure and reduces as we move to the OT side of the infrastructure, indicating that there is a need for highly sophisticated decoys in address this issue. This can be observed by looking at the number of OT-system components impacted by attacks in Figures 33 C, 34C and 35C.

Section 4.5 presents a proposed solution to this issue.

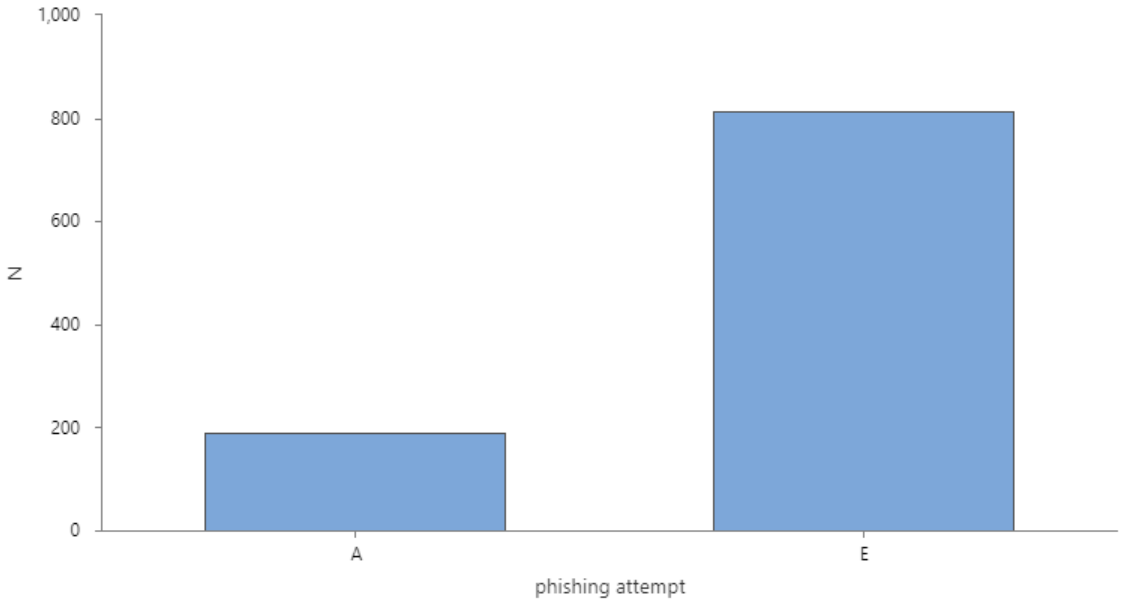


Figure 31-A: Number of Real System Components Impacted by Cyber-attacks -No deception

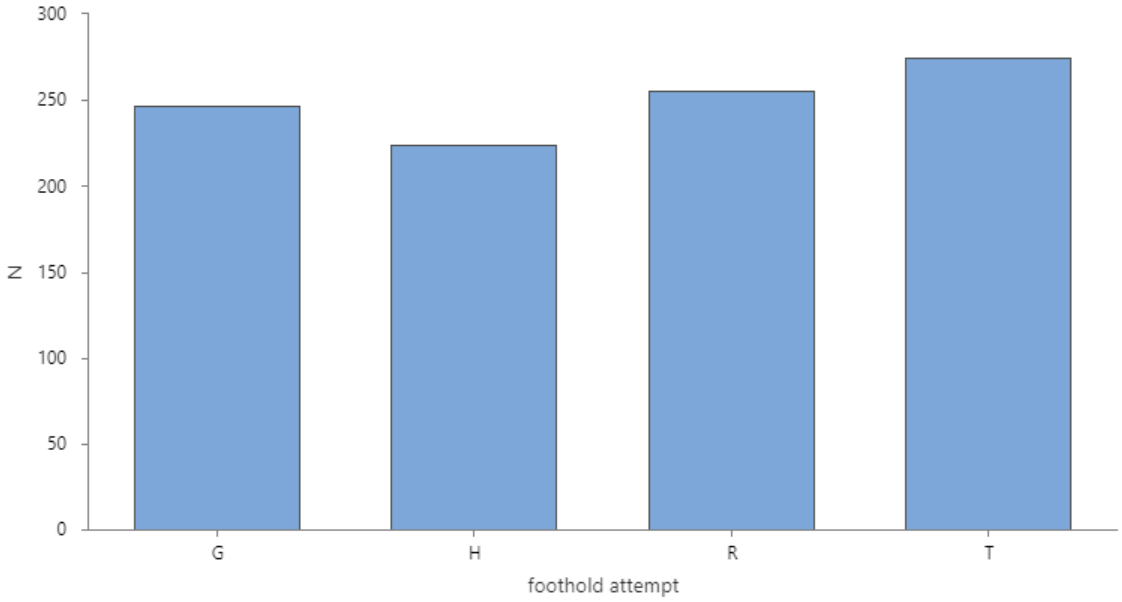


Figure 31-B: Number of Real System Components Impacted by Cyber-attacks -No deception

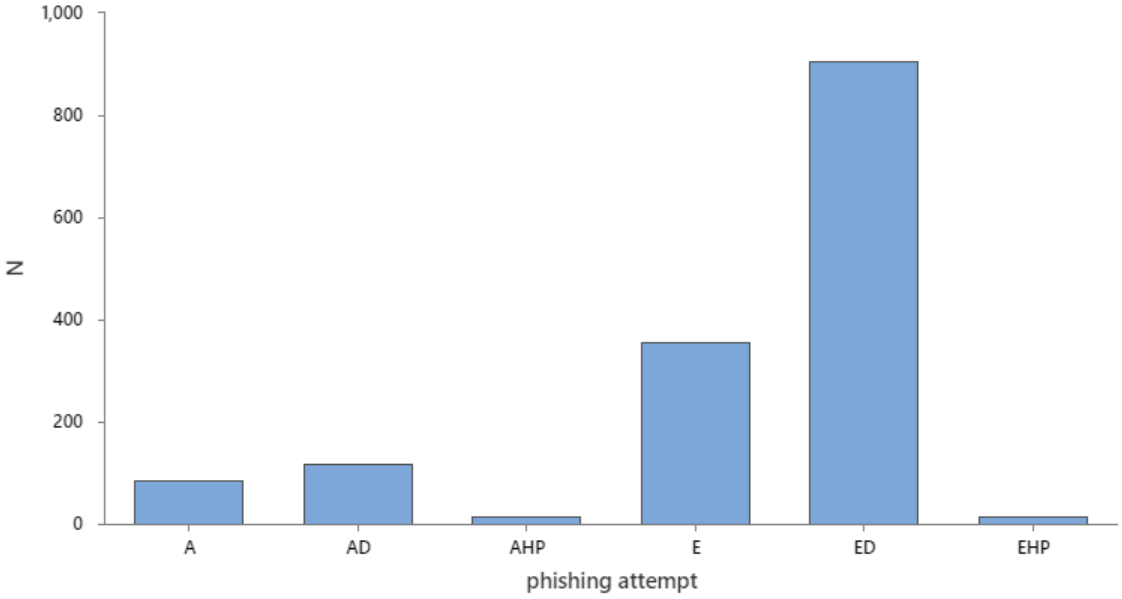


Figure 32-A: Honeypots 15% Decoys 50% and 10 times as many as real

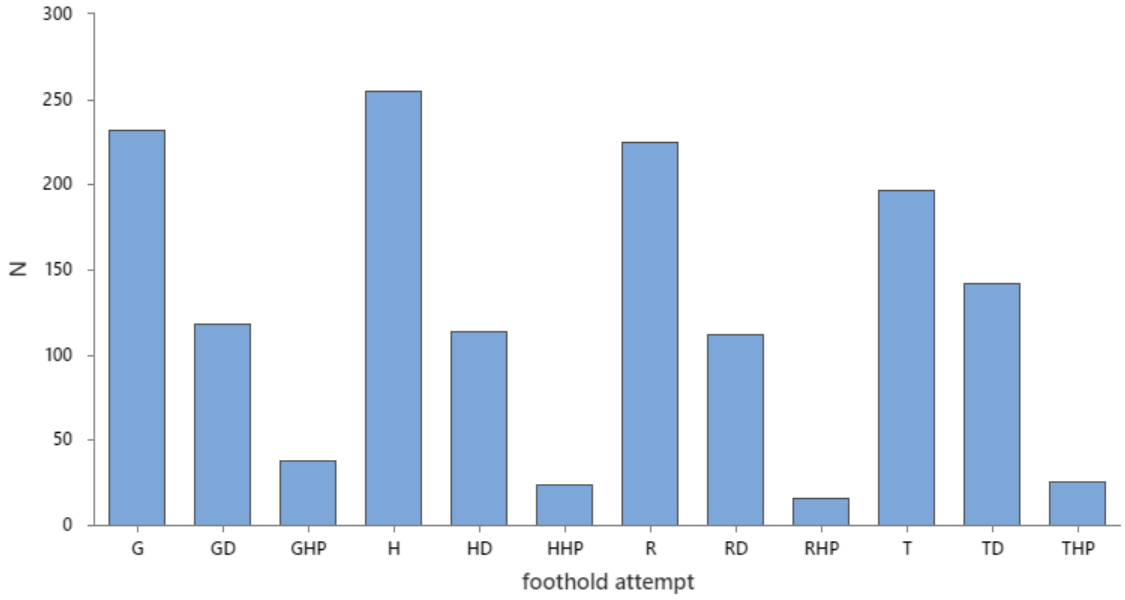


Figure 32B: Honeypots 15% Decoys 50% and 10 times as many as real

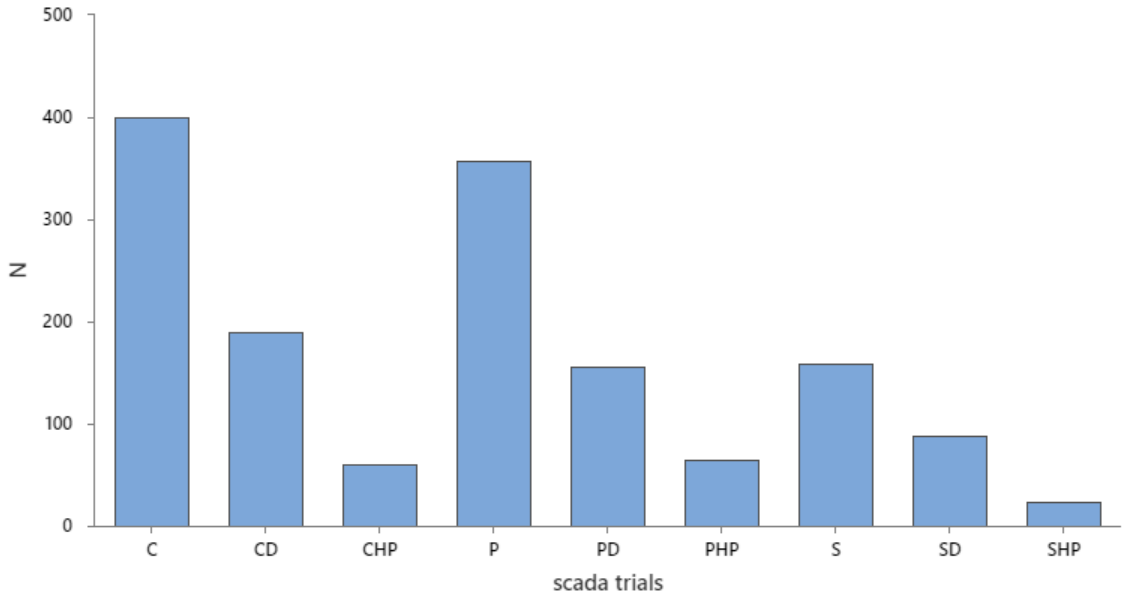


Figure 32-C: Honeypots 15% Decoys 50% and 10 times as many as real

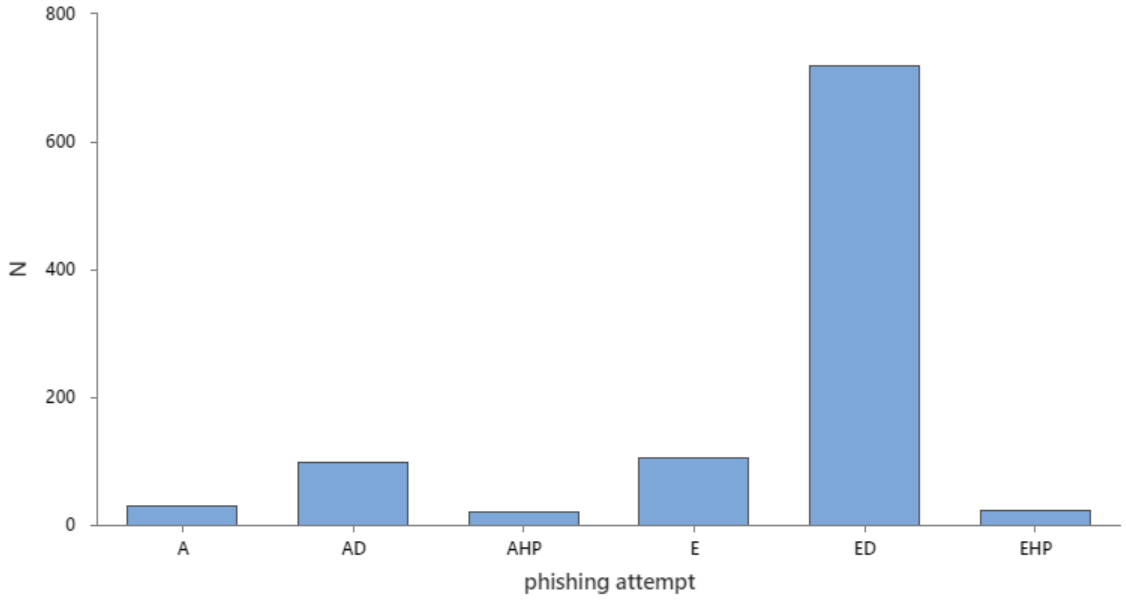


Figure 33-A: Starting to observe a difference at OT end (When decoys are 10 times more and at 100% complexity (0.80 0.85 and 120% real), but can we achieve this)

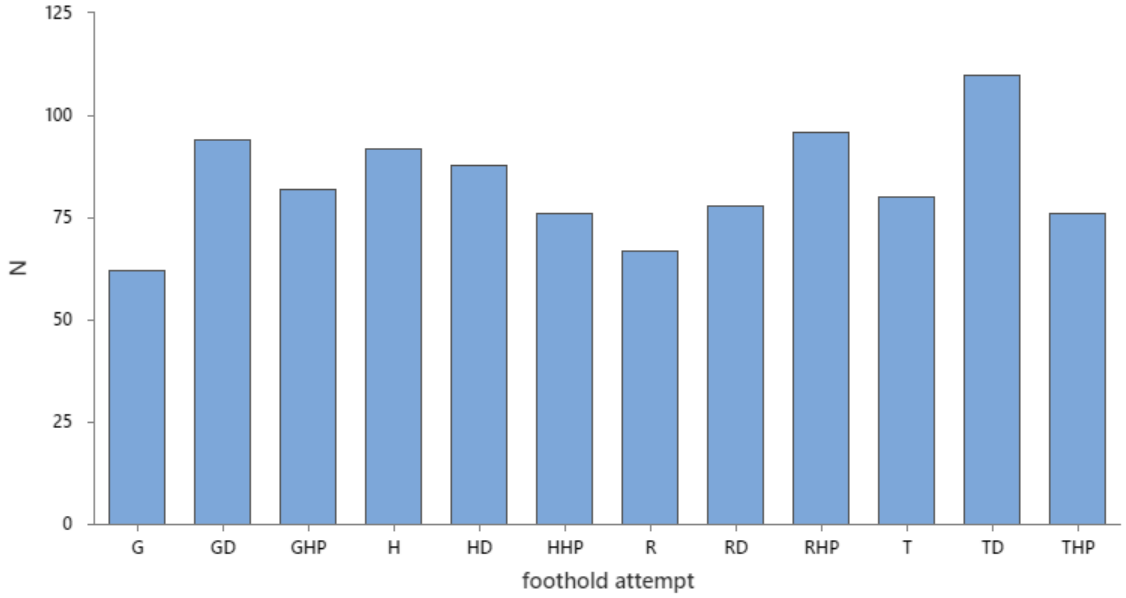


Figure 33-B: Starting to observe a difference at OT end (When decoys are 10 times more and at 100% complexity (0.80 0.85 and 120% real), but can we achieve this)

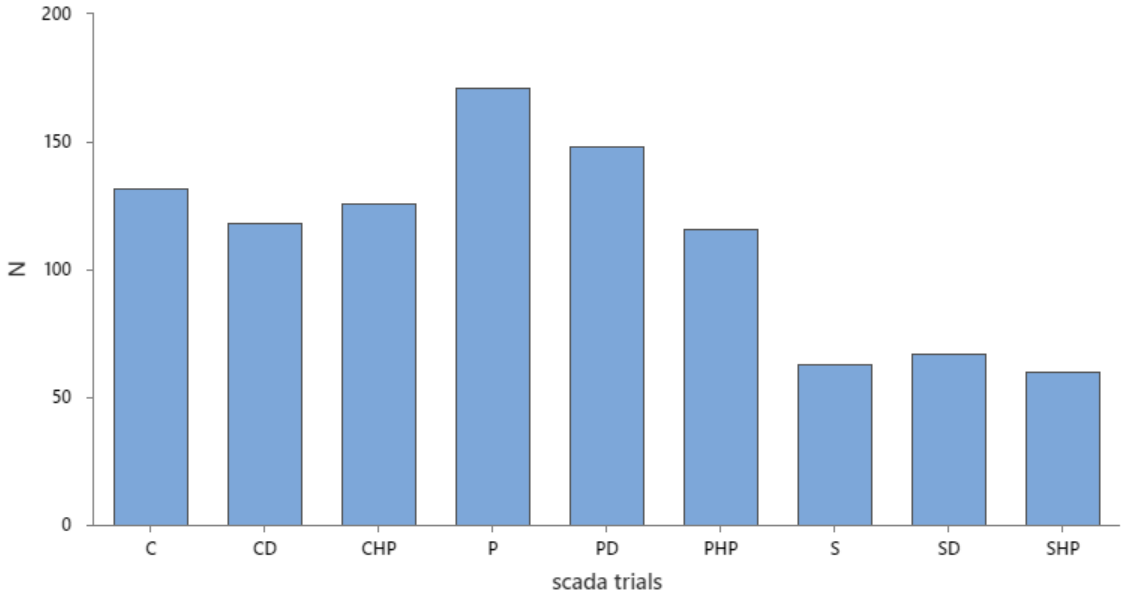


Figure 33 Starting to get a difference at OT end (When decoys are 10 times more and at 100% complexity (0.80 0.85 and 120% real), but can we achieve this)

4.5 Improving Deception on the Operational Technology-side of the Architecture- Deception through the use of Identical PLCs and SCADAs:

Figure 34 summaries a trend presented through the praxis; when complexity or quantities are increased, the impact on the IT layers of the model is significantly more than the impact on the DMZ and OT layers.

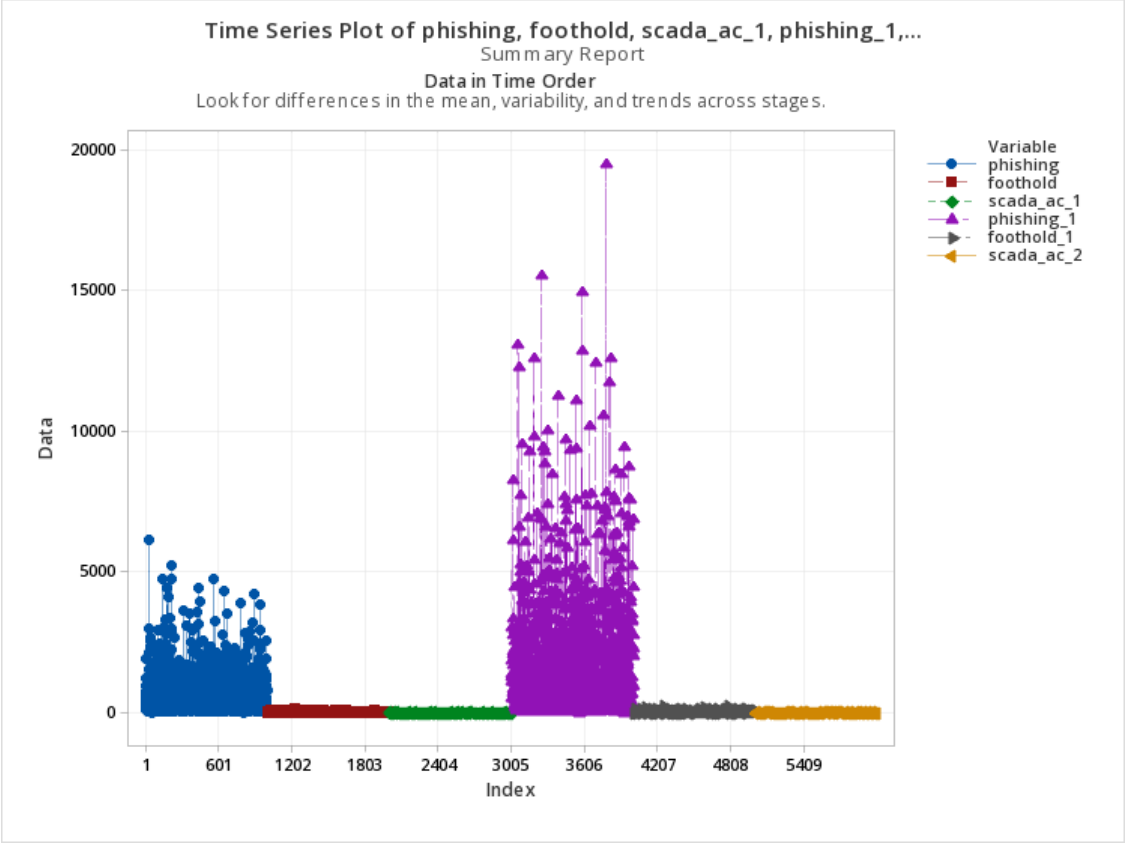


Figure 34 Impact of layers of deception complexity on phases of attack

When this PLC vulnerability is used as a means for defense the outcome can possibly be different. SCADAs will continue to be attacked and because there is typically no redundancy, and even when there is it isn't to the level of redundancy on the IT-side of the infrastructure, causing components on the OT-side will continue to be more vulnerable, until defense methodology is changed.

In Figure 35, below, the result of an attack on the OT system architecture is presented. However, in this attack the real SCADA components is named C and the real PLC component is names P. Decoys have Siemens, Allen Bradley or Schneider Electric prefixes. Note that no real system components are accessed in this attack even though the decoys are not high-fidelity ones.

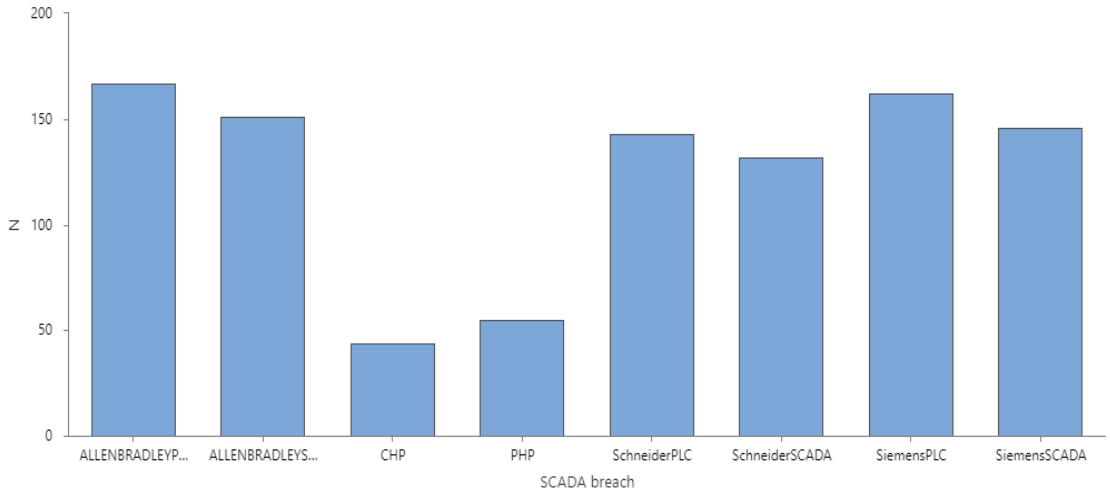


Figure 35 Impact of deception using real components for decoys

The Figure offers that when "digital twins" or decoys that mimic real system components are used and when these components use industry-wide codes that are known such as Siemens when the real system components are Schneider-Electric or vice versa, the engineering manager and operator are able to build more deception into the architecture compensating for the higher rates of deception achieved at the IT levels in comparison to the OT levels of the architecture thus satisfying their specific needs for tailored defense against cyber-attack.

In this case the risk was shifted through psychological deception because the decoys utilize industrial known equipment and code for deception.

4.6 Hypothesis Testing Summary

This section summaries the results presented in Sections 4.3 through 4.5

- **Hypothesis 1 (H1):**

Hypothesis 1 was aimed at testing the impact of the incorporation of complex decoys into the system architecture on the Mean Time to Breach. *The Null Hypothesis was that the MTTB was significantly impacted by the increase of the complexity of decoys in the system.* Starting with the time series comparison of Figure 21 of the MTTB versus deceptive component complexity, it is observed that the MTTB increased proportionately to the complexity of the deceptive components. As the complexity of the honeypots and decoys increased from 15% and 50% respectively to 85% and 100%, the MTTB increase by 10 order of magnitude from 698.29 to 5680 as presented in the statistical summary of Figure 21 C and the ANOVA analysis of Table 15. The statistical analysis, the Welsh and Games-Howell tests and the graphical results illustrated through the Interval plot and the boxplot of the ANOVA analysis in Figure 22, Figure 23 and Figure 24 indicate consistently that there is a significant difference between the MTTB of the four deception conditions, thus allowing the acceptance of the claim made through Hypothesis 1.

Comparing the results of the Base model to those of the best results obtained from the simulations (highest complexity and 10 times as many decoys as real system components), Table 18 summaries the results of the comparison. The table shows that there the number of successful phishing attacks fell from 1.4% to 0.095%. The number of successful foothold attacks fell from 6% to 0.3% and most significantly, the number of successful SCADA attacks fell from 20% to 3%.

	Phishing	Foothold	SCADA
Number of attempted attacks (Base Model)	69453	15403	4978
Number of attempted attacks (With Decoy Defense)	1048574	283393	33860
Reduction in probability of successful attacks	*Probability of successful attack drops from* **1.4** *to* **0.095%**	*Probability of successful attack drops from* **6%** *to* **0.3%**	*Probability of successful attack drops from* **20** *to* **3%**

Table 18 Reduction in number of successful cyber-attacks

- **Hypothesis 2 (H2):**

Hypothesis 2 testing was aimed at testing the impact of the presence of a network of decoys on the attackers' access to real system components and as such the probability of successful cyber intrusions. *The Null Hypothesis was that decoys incorporated into layer of the critical infrastructure system architecture will reduce the probability of successful cyber intrusions.* The results presented in Sections **Error! Reference source not found.** and **Error! Reference source not found.**present the results of testing this hypothesis. The results of Figure 32 present that with no deception incorporated into the system, the impact is 100% on the real system components. The impact on each group of components is proportionate to their quantities within the architecture. On the IT side of the architecture, the Level 4 components are most impacted compared to the Level 5 components consistent with attacks described by (Janicke, Cook, Nicholson, & Jones, 2016) (Alladi, Tejasvi ; Chamola, Vinay & Zeadally, Sherali. , 2020) and many more cited throughout the Praxis. As we proceed further down the layers of the architecture it become more evident that the components are more impacted compared to the upper layers. In literature this is explained by the vulnerability of these components to attacks due to lack of means of defense that are tailored to these layers of the architecture.

Comparing the results of Figure 32 to those of Figure 33 and Figure 34, it is evident that as the complexity of the decoys increases the impact on the real system components diminishes and as such proving the validity of Hypothesis 2. As Hypothesis 2 was proven, however, it was observed that the impact on the OT-side of the architecture was not as prominent as that on the upper layers of the architecture. This result led to the further analysis within Section **Error! Reference source not found.**. The results of Section **Error! Reference source not found.** aim to present a solution to the defense issue observed in Section 4.4. The results of the Section offer that the more the decoys incorporated into the OT-side of the architecture resemble real system components, the more successful they can be in defensing the infrastructure, further providing evidence to the validity of H_1 and H_2.

5 CHAPTER V DISCUSSION and CONCLUSIONS

The research demonstrated the use of an engineering management tool which can be used to assess the vulnerability of a critical infrastructure to cyber-attacks, given the estimated risk of successful attacks at each level of the Purdue System Architecture. In this praxis, probabilities of success where used from probabilities calculated or cited in literature (Black, Scarfone , & Souppay, 2008) (Bobbio, Egidi, Terruggia, Ciancamerla, & Minichino, 2013) (Security, 2022). The research also demonstrated the use of this engineering management tool in assessing the impact of a network of honeypots and decoys on the Mean Time to Breach the architecture starting at the topmost IT layers and ending at the SCADA/PLC layers. In real life scenarios, this tool can be used to assess the vulnerability caused by a specific component within the system, the impact of that component on the overall system resilience assessed through the MTTB and then in the devising of a defense methodology, based on the incorporation of decoys and honeypots into the system architecture. To achieve this the user can enter the probability of accessing levels of the infrastructure and vulnerabilities of components.

5.1 Simulation (Base Model) insights

The research demonstrated that an attacker can gain full access to the SCADA system of an unprotected critical infrastructure, given time. Even with the resistance created by the Demilitarized Zone and the Intrusion Detection system and the firewalls which act as a virtual airgap, an attacker will eventually, successfully, gain full access to the SCADA of a critical infrastructure, in this case the command-and-control signals, PLCs and SCADA. The Base model demonstrated that the full OT system breach takes an average of 68

attempts (mean value in population of 1000), with the attempts acting as a proxy for the Mean-Time to Breach of the full system with a 20% chance of success. This result is illustrated through the time-series graphs and statistical analysis which illustrates the time (attempts) the attacker takes to breach each level of the Purdue Model.

A significant finding of the research is that deceptive defense within the IT layers of the infrastructure is critical in defending the SCADA system components or the OT layers. Referring back to Figure 19, repeated below for ease, it becomes evident that prolonging the time to successfully breaching the IT layers (the phishing attack) the first DMZ firewall provide the most significant defense for the architecture, both IT and OT sides.

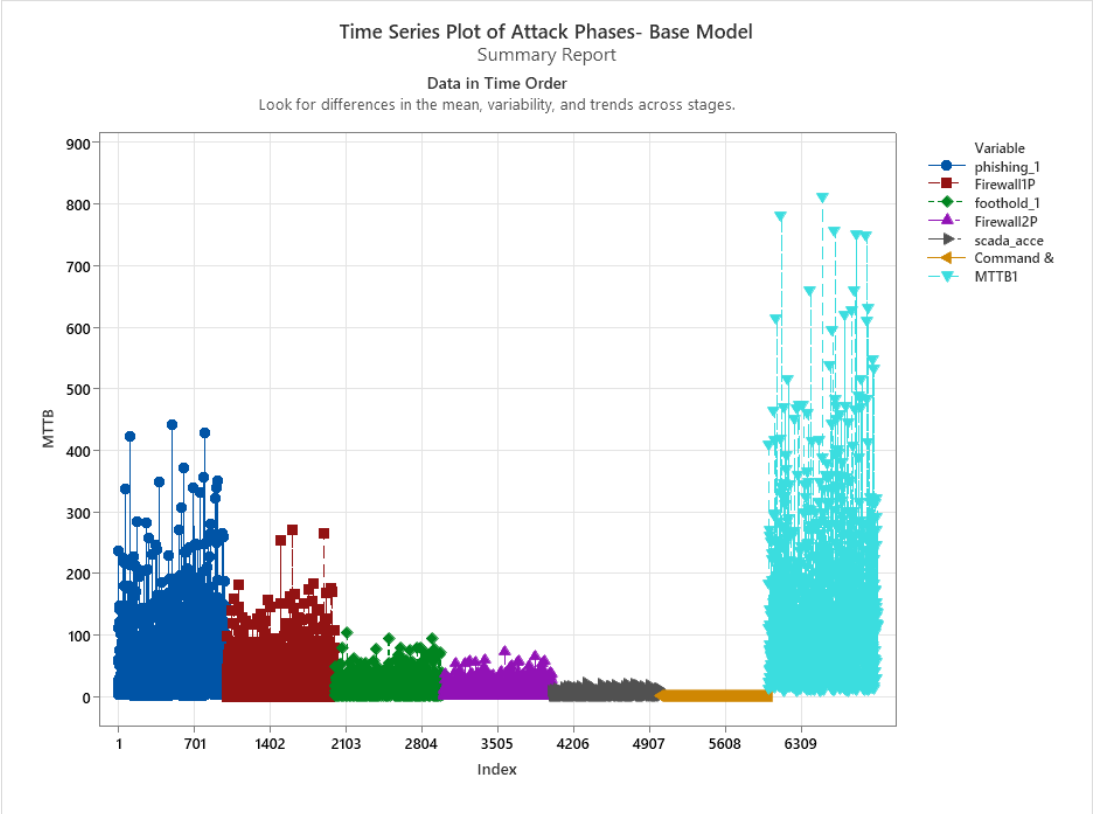


Figure 19 Time Series Plot of Base Model by Attack Phase and overall MTTB

Once these defenses are penetrated, it becomes significantly easier to access the SCADA system components. The first four Defense Cases which were run as part of the Edge Cases to establish the optimum location for the decoys also yielded the results that placement of decoys within the OT system, even though it increases the Mean Time to Breach, does not match the increase in mean time to breach caused by prolonging the time the attacker spends in phishing which in turn influences the overall MTTB for the entire architecture, providing more resilience. This extended time to a successful attack combined with intrusion detection systems allows the engineering manager the time to make more informed decisions regarding managing cyber risks.

5.2 Hypothesis I Testing Insights:

The model and simulations allowed for the proof of Hypothesis I. A significant finding from Hypothesis 1 testing was that a combination of quantities and complexity of decoys yields the best resilience results. The figures and tables of Section **Error! Reference source not found.** demonstrate that as the model complexity increases through the incorporation of decoys, the total time to fully breach the SCADA system, starting at the enterprise level of the network, increases. There is a significant **(α =0.05)** difference between the mean of the phishing attack on the infrastructure which was not cyber decoy protected and that which is. The results of the hypothesis testing indicate that there is a significant **(α =0.05)** difference between an infrastructure architecture protected by a network of decoys and one that is not protected.

5.3 Hypothesis II Testing Insights

The purpose of Hypothesis II was to illustrate that when decoys are incorporated into the architecture of an IT/ OT system, reduce the probability of the attacker's access to real system components and in turn the probability of a successful attack. The results of the testing and statistical analysis presented in Section **Error! Reference source not found.** depict a significant **(α =0.05)** reduction in the number of times the attacker was successful in accessing real system components when the decoy network was incorporated. This is a 10-fold reduction in the success of attacks when a complex network of decoys is incorporated into the model compared to when no decoys are incorporated.

5.4 CONCLUSION

As technological advancement continues and as more aspects of our daily lives continue to interface with the internet the defense of critical infrastructures will become more crucial to the security of our nation. Literature indicates that the defense currently in place for critical infrastructures is not sufficient because it does not match the complexity of the critical infrastructures systems, the fact that the components are accessible through the internet or the improving skills of attackers. Defending critical infrastructures requires a well thought out means of deception which involves the use of decoy networks which resemble real system components and are connected to them in a parallel series of networks at points of vulnerabilities in the network. This leaves the engineering manager responsible for creating architecture resilience with challenges and no tools. The praxis

research presented a tool that the engineering manager can use to 1) access the resilience of a critical infrastructure 2) determine areas of vulnerability in the architecture 3) determine points where deception needs to be deployed and 4) assess optimal defense conditions.

5.5 Future research

The research was based around the assumption that PLCs are observed as merely pieces of code that can be replicated to deceive attackers. Future research should include the utilization of the cyber resilience and engineering concepts presented in the praxis to create resilience through active defense in addition to the sophistication in the design of cyber decoys which enable defenders to create resilience in critical infrastructures. Future research recommendations include cyber decoy design and the need for a sophisticated cyber deception, which combines both psychological deception and deception through the implementation of a network of decoys designed to look like real system components.

Bibliography

Maltby, H., Pakornrat, W., & Jackson, J. (2022). *Markov Chains*. Retrieved from Brilliant: https://brilliant.org/wiki/markov-chains/

McQueen, M., Boyer, W., Flynn, M., & Beitel, ,. (2006). Quantitative Cyber Risk Reduction Estimation Methodology for a Small SCADA Control System. *Proceedings of the 39th Hawaii International Conference on System Sciences* .

Surnin, O., Hussain, F., Hussain, R., Ostrovskaya, S., Polovinkin, A., Young Lee, J., & Fernando, X. (2019). Probabilistic Estimation of Honeypot Detection in Internet of Things Environment. *Workshop on Computing, Networking and Communications.*

A. Cherepanov and R. Lipovsky. (2017). *Industroyer: Biggest threat to industrial control systems since stuxnet.* WeLIVESECURITY ESET.

Alladi, Tejasvi ; Chamola, Vinay & Zeadally, Sherali. . (2020). Industrial Control Systems: Cyberattack Trends and Countermeasures. Computer Communications.

Allen, T. T. (2021). The Ohio State Model For ICS Cybersecurity. *International Conference on Maintenance and Intelligent Asset Management (ICMIAM)* , 1-4.

Alves, Thiago; Morris, Thomas;. (2018). OpenPLC: An IEC 61,131–3 compliant open source industrial controller for cyber security research. *Computers & Security*, Pages 364-379.

American Water Works Assciation- Risk, Resilience and Cyber Security Guidance. (n.d.). Retrieved from www.awwa.org: https://www.awwa.org/Resources-Tools/Resource-Topics/Risk-Resilience/Cybersecurity-Guidance

Applied Power Technologies Inc. (2022, 9). *water-system-cybersecurity-vulnerabilities*. Retrieved from https://www.apt4power.com: https://www.apt4power.com/2021/10/07/water-system-cybersecurity-vulnerabilities/

B. Galloway and G. P. Hancke. (2013). “Introduction to industrial control networks,”. *Communications Surveys &Tutorials, IEEE*, vol. 15, no. 2, pp. 860–880.

B. Galloway and G. P. Hancke. (2013). Introduction to industrial control networks. *Communications Surveys &Tutorials, IEEE,*, vol. 15, no. 2, pp. 860–880.

B. Zhu, A. J. (2011). A taxonomy of cyber attacks on scada systems. *Interntional Conference on the Internet of Things and 4th International Conference on Cyber, Physical and Social Computing IEEE*, 380-388.

Beresford, D. (2011). Exploiting siemens simatic s7 plcs.

Biswas, S. (2022). *6 Major SCADA Attacks That Happened And Their Consequences*. Retrieved from Hackernoon: https://hackernoon.com/6-major-scada-attacks-that-happened-and-their-consequences

Black, P., Scarfone , K., & Souppay, M. (2008). *CYBER SECURITY METRICS AND MEASUREMENTS.* Gaithersburg, Maryland: National Institute of Standards and Technology.

Bobbio, A., Egidi, L., Terruggia, R., Ciancamerla, E., & Minichino, M. (2013). WEIGHTED ATTACK TREES FOR THE CYBERSECURITY ANALYSIS OF SCADA SYSTEMS. *Proceedings of the International Defense and Homeland Security Simulation Workshop.*

Bodeau, D., & Graubart, R. (2017). *Cyber Resiliency Design Principles.* Bedford, MA: MITRE.

Bodeau, D., & Graubart, R. (2017). Cyber Resiliency Design Principles: Selective Use Throughout the Lifecycle and in Conjunction with Related Disciplines.

Bodeau, D., & Graubart, R. (2017). yber Resiliency Design Principles: Selective Use Throughout the Lifecycle and in Conjunction with Related Disciplines.

Boral, S. (2021, January 31). *The Challenge to Explain IoT Architecture*. Retrieved from IoT TechTrends: https://www.iottechtrends.com/iot-architecture-explained/#:~:text=At%20the%20moment%2C%20there%20are%20two%20kinds%20of,the%20bottom%20of%20this%20along%20with%20the%20gateways.

Burns, K., Hermes, A., Keleher, M., Kirpichevsky, Y., Dechant,, J., & Morgeson, D. (2016). *The Common Risk Model Volumes 1-V.* Institute for Defense Analysis.

Cai, N., Wang, J., & Yu, X. (2008). SCADA System Security: Complexity, History and New Developments. *IEEE International Conference on Industrial Informatics.* Daejeon Korea.

Cai, N., Wang, J., & Yu, X. (2008). SCADA System Security: Complexity, History and New Developments. *6th IEEE International Conference on Industrial Informatics* (pp. 569-574). IEEE.

Caltagirone, S., Pendergast, A., & Betz, C. (2013). The Diamond Model of Intrusion Analysis. *Computer Science*.

CISA.gov. (2022, 7 13). *CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY*. Retrieved from CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY Sectors Website: https://www.cisa.gov/water-and-wastewater-systems-sector

Climek, D., Macera, A., & Tirenin, W. (2016). *Cybersecurity & Information Systems Information Analysis Center.* Retrieved from CSIAC: https://csiac.org/articles/cyber-deception/

Colarik, &. J. (2007). *Cyber warfare and cyber terrorism.* Idea Group Reference.

Colelli, R., Panzueru, S., & Pascucci, F. (2020). Exploiting system model for securing CPS: the anomaly based IDS perspective.

Cook, A., Janicke, H., Smith, R., & Maglaras, L. (2021). The Industrial Control System Cyber Defence Triage. *Cyber Technology Institute, De Montfort University*.

Coretelligent. (2022, 7 13). *Coretelligent Power of IT*. Retrieved from https://coretelligent.com/insights/what-is-the-cia-triad-and-why-does-your-cybersecurity-position-depend-on-it/#:~:text=What%20is%20the%20CIA%20Triad%3F%20The%20CIA%20Triad,CIA%20Triad%20stand%20for%20Confidentiality%2C%20Integrity%2C%20and%20Availability

Cremer, F., Sheehan, B., Fortmann,, M., & Kia, A. (2022). Cyber risk and cybersecurity: a systematic review of data availability. *The Geneva Papers on Risk and Insurance - Issues and Practice*, pages698–736.

Cyberattack at med-tech conglomerate Hoya slowed production at Thai factory by 60 percent. (2019, April 8). Retrieved from CyberScoop: https://www.cyberscoop.com/hoya-cyberattack-cryptojacking-thailand/

Cybersecurity and Infrastructure Security Agency. (2022, July 13). Retrieved from Homepage | CISA. https://www.cisa.gov/. Accessed 13 July 2022

D. Ding, Q.-L. H.-M. (2018.). A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing, vol. 275*, pp. 1674–1683.

D. Serpanos, M. T. (2018). Designing safe and secure industrial control systems: a tutorial review. *IEEE Design & Test,vol. 35, no. 3*, pp. 73–88.

Dahad, N. (2018). Designer's Guide to Industrial Internet of Things. *EE Times (Design Lines)*.

Darshana Upadhyay, S. S. (2020). SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations,. *Computers & Security*. doi:https://doi.org/10.1016/j.cose.2019.101666

Devasia, A. (2022, 9 21). *control.com/*. Retrieved from Digital-twinning-and-its-use-in-scada-systems: https://control.com/technical-articles/digital-twinning-and-its-use-in-scada-systems/

Dragonfly: Western Energy Companies Under Sabotage Threat. (2014). Retrieved from https://www.symantec.com/connect/blogs/dragonfly-western-energycompanies-under-sabotage-threat: https://www.symantec.com/connect/blogs/dragonfly-western-energycompanies-under-sabotage-threat

Fan, W., Du, Z., Fernandez, D., & Villagr , V. (2018). Enabling an anatomic view to investigate honeypot systems: A survey. *IEEE*.

Ferguson, K. (2018). Oh look, a butterfly.

Ferguson-Walter, K., Maxine , M., Van Bruggen, D., Furgate, S., & Gutzwiller, R. (2019). The World (of CTF) is Not Enough Data: Lessons Learned from a Cyber Deception Experiment. *IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*, p.346-353.

Fire Eye. (2021). *Double Dragon: APT41, a Dual Espionage and Cyber Crime.* Retrieved from Fireeye: https://content.fireeye.com/apt-41/rpt-apt41/

FORTINET. (2020). *Use Cyberdeception To Lure Attackers from Critical Assets-Proactive Cybersecurity for Operational Technology.* FORTINET.

Franco, J., Aris, A., Canberk, B., & Uluagac, S. (2021). *A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems.*

G. Dan, H. S. (July 2012). Challenges in power system information security. *Security Privacy, IEEE,*.

Gollmann, M. K. (2013). Industrial control systems security:What is happening? *11th IEEE International Conference on Industrial Informatics (INDIN).*, pp. 670–675.

Greenberg, A. (2018.). *SANDWORM : A NEW ERA OF CYBERWAR AND THE HUNT FOR THE KREMLIN'S MOST DANGEROUS HACKERS (The untold story of notpetya, the most devastating cyberattack in history).* Penguin Random House.

H. Holm, M. K. (2015). A survey of industrial control system testbeds. *Nordic Conference on Secure IT Systems. Springer*, pp. 11–26.

Haddad, H., PajouhaAli, P., & Dehghantanhaa, R. (2021). A survey on internet of things security: Requirements, challenges, and solutions: Requirements, challenges, and solutions. *Internet of Things*.

Heindal Security. (2022, 09 04). *Heindal Security - Advanced Persistent Threat*. Retrieved from Heindal Security: https://heimdalsecurity.com/blog/advanced-persistent-threat-apt/

Hubbard, D. (2009). *Failure of Risk Managemet: What is Broken and How to Fix it.* Wiley.

Jajodia, S., Subrahmanian, V., & Swarup, V. (2016). *Cyber Deception: Building the Scientific Foundation.* Cham: Springer International Publishing AG,.

Janicke, H., Cook, A., Nicholson, A., & Jones, K. (2016). Risks, threats and mitigation strategies for SCADA systems. In *Cyber-Physical-Social Systems and Constructs in Electric Power Engineering* (pp. p.1-1).

John Riggi, S. A. (2022, 7 13). *AHA Center for American Healthcare Innovation*. Retrieved from American Hospital Association: https://www.aha.org/center/cybersecurity-and-risk-advisory-services/ransomware-attacks-hospitals-have-changed#:~:text=WannaCry%20is%20illustrative%20of%20the%20modern%20cyber%20threats,investigation%20by%20the%20FBI%20and%20the%20U.S.%20

K. Kimani, V. O. (2019). Cyber security challenges for iot-based smart grid networks. *International Journal of Critical Infrastructure Protection, vol. 25*, pp. 36–49.

Kakareka,, A. (2020). *Defense in Depth - an Overview | ScienceDirect Topics. What is Vulnerability Assessment.* ScienceDirect.

Kaspersky. (2022, 1). *advanced-persistent-threats-in-2022-what-to-look-out-for-next-year*. Retrieved from Kaspersky: https://www.kaspersky.com/about/press-releases/2021_advanced-persistent-threats-in-2022-what-to-look-out-for-next-year

KIM, S., Yannakogeorgos, P. A., & Lowther, A. B. (2013). AMERICAN CYBERSECURITY TRIAD: Governmentwide Integration, Technological Counterintelligence, and Educational Mobilization. *In Conflict and Cooperation in Cyberspace*, 291–310.

Kim, Z. (2014). *Countdown to Zero Day.*

Krutz, R. L. (2006). *Securing SCADA Systems.*

Kumar, M. (2018, august 07). *TSMC Chip Maker Blames WannaCry Malware for Production Halt.* Retrieved from thehackernews.com: https://thehackernews.com/2018/08/tsmc-wannacry-ransomware-attack.html

Lehto, M. (2022). APT cyber-attack modelling-building a general model. University of Jyvaskyla, Finland.

Locke, G., & Gallagher, P. (2009). *NIST Framework and Roadmap for Smart Grid Interoperability Standards.* NIST & Department of Commerce. Retrieved from https://www.nist.gov/system/files/documents/2017/05/09/smartgrid_interoperability.pdf#:~:text=The%20NIST%20Smart%20Grid%20Conceptual%20Reference%20Model%20identifies,and%20applications%20within%20each.%20The%20reference%20model%20also

Lowe, E. B. (2004). The myths and facts behind cyber security risks for industrial control system. *Proceedings of the VDE Kongress*, 213-218.

Luallen, M. (2014). *Breaches on the Rise in Control Systems: A SANS Survey*. Retrieved from SANS: https://www.qualys.com/docs/sans-survey-breaches-rise-control-systems.pdf#:~:text=2Breaches%20on%20the%20Rise%20in%20Control%20Sys

tems%3A%20A,educational%2C%20legal%20and%20government%20services%20to%20this%20industry.

Madnick, A. N. (2015.). A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet. *IEEE Transactions on Dependable and Secure Computing*.

McLaughlin, S. (82–84). Securing control systems from the inside: A case for mediating physical behaviors. *IEEE Security & Privacy, vol. 11, no. 4*, 2013.

Mei, Y., Han, W., Li, S., & Wu, X. (2021). A Survey ofAdvanced Persistent Threats Attack and Defense. *IEEE Sixth International Conference on Data Science in Cyberspace (DSC)*.

Mohapatra, S., & Bhuyan, J. (2016). A Solution Framework for Managing Internet of Things (IOT). *International journal of Computer Networks & Communications.*, 73-87.

Monroe, M., & Olinger, J. (2020). Mission-Based Risk Assessment Process for Cyber MRAP-C). *The ITEA Journal of Test and Evaluation*, 229-232.

Moore, D., PAXSON , V., SAVAGE, S., & STUART, C. (n.d.). Inside the Slammer Worm. *IEEE Security and Privacy*.

Morgeson, D., Meeson, R., Simpson, W., Dechant, J., & Hermes, A. (2016). *The Common Risk Model for Dams, Volume V Cyber Security for Industrial Control Systems.* Institute for Defense Analysis.

Murphy, G. (2022, 01 18). *Forbes - INNOVATION*. Retrieved from Forbes - A Reimagined Purdue Model For Industrial Security Is Possible: https://www.forbes.com/sites/forbestechcouncil/2022/01/18/a-reimagined-purdue-model-for-industrial-security-is-possible/?sh=79febfb81162

N. H. C. Guzman, M. W. (2019). “Conceptualizing the key features of cyber-physical systems in a multi-layered representation for safety and security analysis,”. *Systems Engineering*, pp. 1–22.

Nguyen, T. (2022, 5). Lead Architect and Developer, "Cyber-Physical Systems Platform Security Laboratory," Naval Postgraduate School. (U. C. Professional, Interviewer)

Operators, W. T. (2022, 01). Water Treatment System Operators. (N. N. Taha, Interviewer)

Patterson, D. (2018, 3 6). PODCAST: ICS DETECTION CHALLENGE INTERVIEWS. (CyberX., Interviewer)

Pawlick, C., & Zhu, Q. (2019). A Game-theoretic Taxonomy and Survey of Defensive Deception for Cybersecurity and Privacy. ACM Computing Surveys, 52(4), 1–28.

https://doi.org/10.1145/3337772. *ACM Computing Surveys, 52(4)*, 1–28. doi:https://doi.org/10.1145/3337772

Petrenki?? (n.d.).

Petrenko, S. (2019). *Cyber Resilience.* Gistrup, Denmark: River Publishers.

Preferred IT Group. (2019, 8 27). *Preferred IT Group*. Retrieved from https://www.preferreditgroup.com/2019/08/27/the-three-goals-of-cyber-security-cia-triad-defined/

Qian, C., & Abdelwahed, S. (2014). Towards realizing self-protecting SCADA systems. *9th Cyber and Information Security Research Conference.*

R. Khan, P. M. (2016). Threat analysis of blackenergy malware for synchrophasor based realtime control and monitoring in smart grid. *ICS-CSR, vol. 16*, 1-11.

R. M. Lee, M. J. (2014.). German steel mill cyber attack. *Industrial Control Systems, vol. 30*, p. 62.

Real Pars. (2022). Retrieved from RealPars: https://realpars.com/

Reddy, M. C. (2022, 7 13). *Global Commission on the Stability of Cyberspace- 2019 Report*. Retrieved from Global Commission on the Stability of Cyberspace: https://cyberstability.org/report/

Robinson, M. (2013). The SCADA threat landscape. *1st International Symposium for ICS & SCADA Cyber Security Research* (pp. 16-17). Leicester, UK: Workshops in Computing, H. Janicke and K. Jones, (Eds) BCS, [Online].

Rohozinsk, J. P. (pp. 23–40). Stuxnet and the future of cyber war. *Survival, vol. 53, no. 1*, 2011.

Rowe, [. B. (2012.). "A survey scada of and critical infrastructure incidents." . *RIIT, vol. 12*, 51–56.

Rowe, B. M. (2012). "A survey scada of and critical infrastructure Incidents. *RIIT*, 51-56.

Rubio, J., Alcaraz, C., & Roma, R. (2019). Current cyber-defense trends in industrial control systems. *Computers & Security*. doi:https://doi.org/10.1016/j.cose.2019.06.015.

Ryan, J., & Mazzuchi, T. (2012). Quantifying information security risks using expert judgment elicitation. *Management Science*.

S. Kriaa, L. P.-C. (2015). A survey of approaches combining safety and security for industrial control systems. *Reliability engineering & system safety vol. 139*, 156–178.

S. McLaughlin, C. K.-R. (2016.). The cybersecurity landscape in industrial control systems. *Proceedings of the IEEE, vol. 104, no. 5*, 1039– 1057.

Sahlin, D. J. (2022, March). Metrics determination. (N. N. Taha, Interviewer)

Sahlin, J. (2022, February 11). *www.gdit.com*. Retrieved from General Dynamics Information Technology: https://www.gdit.com/perspectives/latest/when-everything-is-code-everything-bounces-fwd/

Sectigo. (2022). *42 Cyber-attack-statistics-by-year-a-look-at-the-last-decade*. Retrieved from https://sectigostore.com/blog/42-cyber-attack-statistics-by-year-a-look-at-the-last-decade/

Securicon. (2022, 04 04). *Securicon White Paper Case Studies*. Retrieved from Securicon: https://www.securicon.com/wp-content/uploads/2020/12/Part-1-Securicon-D6-White-Paper-compressed.pdf

Security Insights. (2022, March). */www.securityinsights.net/*. Retrieved from Cyber Threats on Critical Infrastructure: https://www.securityinsights.net/top-7-cyber-threats-on-critical-infrastructure/#:~:text=Top%207%20cyber%20threats%20on%20critical%20infrastructure%201,...%207%207.%20Weak%20or%20stolen%20credentials%20

Security, P. (2022). *2022 Cyber Security Statistics*. Retrieved from PurpleSec: https://purplesec.us/resources/cyber-security-statistics/#:~:text=34%25%20of%20companies%20experienced%20damage,result%20of%20an%20APT%20attack.

Seda-Sanabria, D. (2022, 02 01). Dr. Seda-Sanabria Interview. (N. N. Taha, Interviewer)

Seda-Sanabria, Y., & Morgeson, J. D. (2016). An Integrated Approach for Physical and Cyber Security Risk Assessment: The U.S. Army Corps of Engineers Common Risk Model for Dams.

Service, H. a. (2022, september). *using-honeypots-network-intrusion-detection*. Retrieved from www.hhs.gov: https://www.hhs.gov/sites/default/files/using-honeypots-network-intrusion-detection.pdf

Singh, P., Garg, S., Kumar, V., & Saquib, Z. (2015). A Testbed for SCADA Cyber Security and Intrusion Detection. *International Conference on Cyber Security of Smart Cities, Industrial Control and Communications*.

Stouffer, K., Falco, J., & Scarfone, K. (2011). *Guide to Industrial Control Systems (ICS) Security - Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers.* Special Publication by National Institute for Standards and Technology.

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., & Hahn, A. (2011). *Guide to Industrial Control Systems (ICS) Security.* NIST Special Publication 800-82 Revision 2.

Taha, N. N. (2022, 4 4). *Google Colab Platform- Nesrine Taha Water Treatment Model Python Simulation*. Retrieved from Google Colab: https://colab.research.google.com/

Thryft, A. R. (2022). *Critical Infrastructure Cyber Attacks on the Rise*. Retrieved from www.eetasia.com: https://www.eetasia.com/critical-infrastructure-cyber-attacks-on-the-rise/

United States Computer Emergency Readiness Team. (2011). *US-CERT incident response summary 2009-2011.* Department of Homeland Security.

V. Hassija, V. C. (2019). V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures,". *IEEE Access, vol. 7*, pp. 82 721.

Verve, & Livingston, J. (2021, February 23). *The Future of OT Security: OT System Management*. Retrieved from Verve: https://verveindustrial.com/resources/blog/the-future-of-ot-security-ot-systems-management/

Wagner, R., Fredrikson, M., & Garlan, D. (2017). *An Advanced Persistent Threat Examplar.* Pittsburgh: Institute for Software Research- School of Computer Science -Carnegie Mellon University.

Wang , H., Lau, N., & Gerdes , R. (2018). Examining Cybersecurity of Cyberphysical Systems for Critical Infrastructures Through Work Domain Analysis. Human Factors. *Human Factor*, 699-718.

warner, m. (2012). Cybersecurity: A Pre-history. *Intelligence and National Security*, 781–799.

Williams, T. (1994). The Purdue enterprise reference architecture. *Computers in Industry*, 141-158.

Williams, T. J. (September 1994). The Purdue enterprise reference architecture. *Computer in Induustry*, Volume 24, Issues 2–3 Pages 141-158.

Xu, H. (2019). A Survey of the Internet of Things. *IEEE*.

Z. Drias, A. S. (2015). Analysis of cyber security for industrial control systems. *International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC). IEEE*, 1-8.

Zanutto, A., & Follis, K. (2017). The Shadow Warriors: In the no man's land between industrial control systems and enterprise IT systems. *WSIW 2017 3rd Workshop on Security Information Workers* .

Zeadally, C. A. (2015). Critical infrastructure protection: Requirements and challenges for the 21st century. *International journal of critical infrastructure protection*, 53-66.

Zhenxin Zhan, Maochao Xu, & Xu, S. (2013). Characterizing Honeypot-Captured Cyber Attacks: Statistical Framework and Case Study. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.* IEEE.

zscaler. (2022, 7 14). Retrieved from .zscaler.com: https://www.zscaler.com/

APPENDIX A

Python Model

```
import random
#LEVEL 5
totalA = 10 #  number of terminals at Enterprise Level 5
totalAHP = 100
totalAD = 100


#LEVEL 4
totalE = 100 #  number of terminals at Enterprise Level 4
totalEHP = 1000
totalED = 1000


#Demilitarized Zone
totalH = 2 # DMZ Level 3 # Data Historian Server (H)
totalHHP = 20
totalHD = 20
totalHPasswodEnclave = 0

totalG = 2 # Patching Server (G)
totalGHP = 20
totalGD = 20

totalR = 2 # Application server (R),
totalRHP = 20
totalRD = 20

totalT = 2 #  Remote access server (T)
totalTHP = 20
totalTD = 20


#SCADA
totalC = 1 # SCADA   Level 2
totalCHP = 10
totalCD = 10
```

```
#PLC
totalP = 6 # PLC Level 1
totalPHP = 60
totalPD = 60


#Sensors
totalS = 240 # Sensors Level 0
totalSHP = 2400
totalSD = 2400

totals_phishing = [totalA, totalAHP, totalAD, totalE,totalEHP,totalE
D]
phishing_attack = [ 'A','AHP','AD', 'E','EHP','ED']
phishing_attack_prob = [1,0.15,0.4,  1,0.15,0.4, ]
phishing_wt= [a * b for a,b in zip(totals_phishing,phishing_attack_p
rob)]

totals_lateral_movement = [totalA, totalAHP, totalAD, totalE,totalEH
P,totalED]
lateral_movement =['A','AHP','AD', 'E','EHP','ED']
lateral_movement_prob = [1,0.15,0.4,  1,0.15,0.4]
lateral_movement_wt= [a * b for a,b in zip(totals_lateral_movement,l
ateral_movement_prob)]

FireWall1Penetration_failure = [True, False]
FireWall1Penetration_probability = [0.8, 0.2]

totals_foothold = [ totalH,totalHHP,totalHD, totalG, totalGHP,totalG
D, totalR, totalRHP, totalRD, totalT,totalTHP ,totalTD ]
foothold_attack = [ 'H', 'HHP','HD','G','GHP','GD','R','RHP','RD','T
','THP','TD' ]
foothold_attack_prob = [1,0.15,0.4, 1,0.15,0.4, 1,0.15,0.4, 1,0.15,0
.4 ]
foothol_wt= [a * b for a,b in zip(totals_foothold,foothold_attack_pr
ob)]

FireWall2Penetration_failure = [True, False]
FireWall2Penetration_probability = [0.8, 0.2]

SCADAaccess_success = [True, False]
SCADAaccess_probability = [0.3, 0.7]

totals_SCADA = [totalC,totalCHP,totalCD, totalP,totalPHP,totalPD, to
talS,totalSHP,totalSD]
```

```
SCADA = ['C','CHP','CD', 'P','PHP','PD', 'S','SHP','SD']
SCADA_prob = [1,0.15,0.4, 1,0.15,0.4, 1,0.15,0.4]
SCADA_wt= [a * b for a,b in zip(totals_SCADA,SCADA_prob)]

Field_Equipment = ['S', 'SHP','SD' ]
Field_Equipment_prob = [1,0.15,0.4]



from collections import Counter
trials = Counter()



def do_trials():

  trials = Counter()
  phishing_trials = []
  foothold_trials = []
  scada_trials = []
  FW1_trials =[]
  FW2_trials = []
  sensor_trials = []



  #STEP1:
  while True:
    phishing = random.choices(phishing_attack,phishing_wt, k=1)[0]
    trials['phishing'] += 1
    phishing_trials = phishing_trials + [phishing]



    while  phishing != "A" and phishing != "E":
     phishing = random.choices(phishing_attack,phishing_wt, k=1)[0]

     trials['phishing'] += 1
     phishing_trials = phishing_trials + [phishing]


    if ((phishing == 'A') or (phishing == 'E')) and (random.random()
>0.32):
```

```
        phishing_trials = phishing_trials + [phishing]
        trials['phishing'] += 1
        continue
   #STEP 2: Lateral movement
    if (phishing == 'E'):

        trials['phishing'] += 1
        trials['1st-attempted-attack'] =  phishing
        phishing_trials = phishing_trials + [phishing]

        while  phishing != "A":
          phishing = random.choices(lateral_movement,lateral_movemen
t_prob, k=1)[0]
          trials['phishing'] += 1
          phishing_trials = phishing_trials + [phishing]
        trials['Enterprise level breached'] =  phishing



    else:
        phishing_trials = phishing_trials + [phishing]
        trials['phishing'] += 1
        trials['1st-attempted-attack'] =  phishing
        trials['Enterprise level breached'] =  ''


#FW1 Breach:

    if random.choices(FireWall1Penetration_failure, FireWall1Penetra
tion_probability, k=1) [0]:
       trials['Firewall1Penetration'] +=1
       print ('firewall breached')
       foothold = random.choices(foothold_attack,foothold_attack_pro
b, k=1)[0]
    else:
          print ('failed access attempt-FW!')
          while random.choices(FireWall1Penetration_failure, FireWal
l1Penetration_probability, k=1) [0]:
              trials['Firewall1Penetration'] +=1
              continue


    #STEP3: foothold_attack
```

```
    foothold = random.choices(foothold_attack,foothold_attack_prob,
k=1)[0]
    print(foothold)
    foothold_trials = foothold_trials + [foothold]
    trials['foothold'] += 1
    trials['DMZ breach'] = foothold
    if (foothold == 'TD') or (foothold == 'RD') or (foothold == 'GD'
) or  (foothold == 'HD') or (foothold == 'THP') or (foothold == 'RHP
') or (foothold == 'GHP') or  (foothold == 'HHP'):
        trials['foothold'] += 1
        foothold_trials = foothold_trials + [foothold]
        continue
    if  ((foothold == 'T') and (random.random()> 0.3)) or ((foothold
 == 'R') and (random.random()> 0.3)) or ((foothold == 'G') and (rand
om.random()> 0.3)) or  (foothold == 'H') and (random.random()> 0.3):
        trials['foothold'] += 1
        foothold_trials = foothold_trials + [foothold]
        continue
    elif ((foothold == 'T') and (random.random()<= 0.3)) or ((footho
ld == 'R') and (random.random()<= 0.3)) or ((foothold == 'G') and (r
andom.random()<= 0.3)) or  (foothold == 'H') and (random.random()<=
0.3):
        trials['foothold'] += 1
        trials['DMZ breach'] = foothold
        foothold_trials = foothold_trials + [foothold]
#FW2

    if random.choices(FireWall2Penetration_failure, FireWall2Penetra
tion_probability, k=1) [0]:
       trials['Firewall2Penetration'] +=1

    else:

          while random.choices(FireWall2Penetration_failure, FireWal
l2Penetration_probability, k=1) [0]:
              print ('failed access attempt-FW!' )
              trials['Firewall2Penetration'] +=1
          trials['Firewall2Penetration'] +=1

#STEP4: SCADA ATTACK
    scada_access = random.choices(SCADA,SCADA_prob, k=1)[0]
    scada_trials = scada_trials + [scada_access]
    trials['scada_access'] += 1
    trials['SCADA breach'] = scada_access
```

```
    if (scada_access == 'CD') or (scada_access == 'PD') or (scada_ac
cess == 'CHP') or (scada_access == 'PHP') :
       scada_trials = scada_trials + [scada_access]
       trials['scada_access'] += 1

       continue
    if ((scada_access == 'C') and (random.random()> 0.3)) or ((scada
_access == 'P') and (random.random()> 0.5)):
        scada_trials = scada_trials + [scada_access]
        trials['scada_access'] += 1
        trials['SCADA breach'] = scada_access
    if ((scada_access == 'C') and (random.random()<= 0.3)) or ((scad
a_access == 'P') and (random.random()<= 0.5)):
        scada_trials = scada_trials + [scada_access]
        trials['scada_access'] += 1
        trials['SCADA breach'] = scada_access
#Sensor access
        sensor_access = random.choices(Field_Equipment,Field_Equipme
nt_prob, k=1)[0]
        sensor_trials = sensor_trials + [sensor_access]
        trials['Command & Control accessed'] += 1
        trials['sensor'] = sensor_access
        if (sensor_access == 'SD') or (sensor_access == "SHP"):
           continue

        break
  return trials ,phishing_trials, foothold_trials,scada_trials



many_trials = []
many_phishing_trials = []
many_foothold_trials = []
many_scada_trials = []
for i in range(1000):
  trials, phishing_trials,foothold_trials,scada_trials = do_trials()

  many_trials.append(trials)
  many_phishing_trials = many_phishing_trials + phishing_trials
  many_foothold_trials = many_foothold_trials + foothold_trials
  many_scada_trials = many_scada_trials + scada_trials


import pandas as pd
```

```
table = pd.DataFrame.from_records(many_trials)
table2 = pd.DataFrame({'phishing attempt':pd.Series(many_phishing_tr
ials), 'foothold attempt':pd.Series(many_foothold_trials), 'scada tr
ials':pd.Series(many_scada_trials) })
table.to_csv('NNT-PraxisData-MainData-File .csv')
table2.to_csv('NNT-PraxisData-BACKENDDATA-File.csv')
```