

Attack Scenario Generator for Industrial Control System



Mazyounah Alfagham

Supervisor: Dr Charles Morisset

School of computing
Newcastle University

This thesis is submitted for the degree of
Doctor of Philosophy

This thesis is dedicated to my loving family and friends, who have stood by me every step of the way.

To my MOTHER, [Moadiah] whose sacrifices, love have been my guiding light throughout this journey. Your belief in me has fuelled my determination to succeed.

To my FATHER, [Hayef] In the face of illness, you demonstrated resilience and courage that I can only aspire to emulate. To you, my hero, who taught me the value of determination, and who continues to inspire me to reach for the stars.

To my extraordinary siblings, To my caring Charismatic **[Bader]**, my optimistic **[Albandry]**, my Kind-hearted **[Bander]**, my backbone **[Bedoor]**, my funny generous **[Abdulaziz]**, my responsible diligent **[Abdullah]**, my easygoing inspirational **[Saud]**, my support system **[Zoz]** for their unwavering support, encouragement. Your belief in my potential has been my driving force. Your insights and wisdom have guided me through moments of uncertainty.

To My Daughters, [Mishael and Sarah] From the moment you came into my life, you brought immeasurable joy. May this work stand as a testament to the power of dreams, the strength of family, and the endless possibilities that lie ahead.

Declaration

I, Mazyounah Alfagham, declare that this thesis, titled Attack Scenario Generator for Industrial Control system , submitted for the degree of Doctor of Philosophy at Newcastle University, represents my own original work. All sources of information and material used in this research have been properly acknowledged and cited in accordance with the established academic citation standards. I have adhered to all ethical and academic guidelines and regulations set forth by Newcastle university throughout the research. I am aware that any breach of academic integrity, including plagiarism or unethical conduct, may result in disciplinary actions, including the potential revocation of my degree. I assume full responsibility for the content and methodology of this research.

Mazyounah Alfagham

June 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my thesis advisor, **Dr.CHARLES MORISSET**, whose expertise, guidance, and patience have shaped me into the scholar I am today. I am grateful to the **University of Hafr Al Batin** for their financial support, which made this research possible. Additionally, I extend my appreciation to the **Newcastle University** for providing access to their extensive library resources and research facilities. I want to acknowledge my friends and family for their constant encouragement and understanding during this demanding journey. Their belief in me kept me motivated

I am indebted to my fellow graduate students and research colleagues at **Secure and Resilient Systems (SRS) research group** for their stimulating discussions, brainstorming sessions, and the sense of camaraderie that enriched my academic experience.

Abstract

Attack scenarios are hypothetical or planned sequences of events that describe how an attacker might target a system, organization, or network. Their primary goal is to carry out malicious activity. Attack scenarios help in identifying potential threats and understanding the possible consequences of a successful attack. Security analysts traditionally create attack scenarios manually. They may use graphical security models such as attack graphs, trees, or frameworks such as cyber kill chain or a combination of these. Security analysts heavily rely on their knowledge and experience to carry out this manual approach. However, the manual approach is a challenge for complex systems, such as Industrial Control Systems (ICSs). Indeed, ICSs have various requirements coming from the plurality of structures, devices, protocols and application contexts. In addition, the threat landscape for ICSs is constantly evolving due to their increased use. The manual creation of an attack scenario for a given ICS against a given threat landscape might therefore be complex, error-prone and quickly outdated.

The proposed novel general methodology can be effectively used by security analysts to define attack scenarios for ICSs. The proposed methodology gathers the raw data from vast sources to prepare the data and initiate the inferential analysis. Furthermore, it structures and creates the attack sequence to generate the scenario and then simulate the attack scenario. The method was first tested by manually analyzing a complex case study. Human analysts were relied upon to review previous reports and map them with ICS cyber kill chain to generate a scenario and identify the relationship between the attacks. Next, it was demonstrated that this method could be automated. Both a threat-based approach (by automating the cyber threat knowledge base to generate attack scenarios) and a system-based approach (by using the static system state to create attack scenarios) were used. These two approaches were combined in a new tool called the Attack Scenario Generator (ASG). The ASG can generate and optimize attack scenarios based on the cyber kill chain, and predict the techniques, software, and groups behind the attacks in just a few seconds with high accuracy. This saves time, effort, and assists ICS owners efficiently.

Table of contents

List of figures	xvii
List of tables	xxi
1 Introduction	3
1.1 Problem Formulation	4
1.1.1 Aim of the Research Study	5
1.2 Research Questions	5
1.3 Research Challenges	7
1.4 Proposed Solution	8
1.4.1 The Rationale of Using Manual/State/Threat phases	10
1.4.2 Major Contributions	10
1.4.3 Thesis Structure	11
2 Background	13
2.1 Definitions	13
2.2 Introduction	13
2.3 Industrial Control Systems (ICSs)	14
2.3.1 ICS Heterogeneity	14
2.3.2 Securing Heterogeneous ICS with Attack Scenarios	17
2.3.3 Cyber Incidents in ICS	18
2.4 Attack Scenarios Generation	22
2.5 Attack Scenarios Related Works	26
2.6 Synthesis and Gaps in Literature	28
2.7 Chapter Discussion	29
2.8 Conclusion	30
3 Manual Generation of Attack Scenarios	31
3.1 Introduction	31

3.2	Related Works	32
3.2.1	The Diamond Model	32
3.2.2	The OWASP Threat Model	33
3.2.3	Attack Graph	33
3.3	Destructive Wiper Malware Attacks	33
3.4	Review of Destructive Malware Attacks	34
3.4.1	Aramco Attacks	35
3.4.2	South Korea Attack	37
3.4.3	Sony Pictures Entertainment (SPE) Attack	38
3.5	Method	38
3.6	The ICS Cyber Kill Chain	39
3.7	Chapter Discussion	40
3.8	Conclusion	41
4	Attack scenario Generation: Threat-based	45
4.1	Introduction	45
4.2	Related Works	46
4.3	Methodology	48
4.4	Data ICS ATT&Ck	49
4.5	Data Preparation (preprocessing)	49
4.6	ICS Attack Description Analysis	51
4.7	Features Extraction	52
4.7.1	TF-IDF Vectorizer Model	52
4.7.2	Similarity	52
4.8	Developing ICS Prediction Model	52
4.9	Training Performance of ICS models	54
4.9.1	Technique Model	54
4.9.2	ICS Incident Model	55
4.9.3	Group Pipeline Model	55
4.10	Testing of ICS models	57
4.11	Validate ICS Models	58
4.12	Hierarchical Sequence for Scenarios Generation	59
4.13	ICS Scenario Generation Model	61
4.13.1	Checking ICS Attack Scenarios	63
4.14	Chapter Discussion	64
4.15	Conclusion	64

5	Attack Scenario Generation: State-Based	65
5.1	Introduction	65
5.2	Related Works	66
5.3	Methodology	67
5.4	State Based ASG Architecture	68
5.5	System State	70
5.6	Attack Techniques	72
5.7	Generate Attack Scenario Process	74
5.7.1	Sequence Structure for Scenario Generation	74
5.8	Validation	80
5.9	Attack Scenario Generator Interfaces Implementation	80
5.10	Chapter Discussion	81
5.11	Conclusion	84
6	General Methodology To Create Attack Scenario For ICS	87
6.1	Introduction	87
6.2	Attack Scenario Generation Methodology	87
6.2.1	Data Collection Related to Attacks	90
6.2.2	Data Processing	90
6.2.3	Inferential Analysis	90
6.2.4	Structure Attack Techniques Sequence	90
6.2.5	Generate Attack Scenario	91
6.2.6	Scenarios Validation	91
6.3	ASG Architecture	93
6.4	Correctness of Data Generated	93
6.5	Evaluation of the Proposed Methodology for Generating Attack Scenarios	93
6.6	Use Cases	96
6.6.1	Use Case 1: Adversary Emulation	96
6.6.2	Use Case2: Cyber Threat Intelligence Enrichment	102
6.6.3	Use Case3: Red Teaming	104
6.7	Validation of the General Methodology Using Expert Interviews	105
6.8	Experts Interviews	105
6.8.1	Using the Methodology to Generate Attack Scenarios	106
6.8.2	Participant Background	110
6.8.3	Usability Study Validation	110
6.8.4	Results	111
6.8.5	Response to the Insights	111

6.9	ASG Graphical User Interface(GUI)	112
6.10	Application Programming Interface (API) Configurations	120
6.10.1	Architecture of the ASG Application	120
6.10.2	Cloud API Server	120
6.10.3	Authentication and Authorization	121
6.10.4	Services Deployment	121
6.10.5	ASG Web-based Application	122
6.11	Chapter Discussion	125
6.12	Conclusion	126
7	Discussion and conclusion	127
7.1	Introduction	127
7.2	Discussion	128
7.3	Challenges	129
7.4	Limitations	130
7.5	Conclusion	130
7.6	Future Work	131
	References	133
	Appendix A Use cases for ASG	143
A.1	Motivation	143
A.2	Use Cases	144
A.3	Usability Study	145
A.4	Use Case in Graph	150
	Appendix B Assessing Guidelines for Destructive Malware Response	153
2.1	Incident Response Practices	153
2.1.1	United States National Cybersecurity and Communications Integra- tion Centre Response plan	153
2.1.2	European Union Agency for Cybersecurity Response Plan	155
2.1.3	Analysis of Incident Response Guidelines	155
2.1.4	Aramco Cyber Response	157
2.1.5	Aramco Cyber Incident Response Assessment	157
2.2	Conclusion	158

3	Attack Scenario Generator	161
3.1	Introduction	161
3.2	Algorithms	161
3.2.1	Data Production Algorithm	161
3.2.2	NLP Process Algorithm	163
3.2.3	Check Scenario Algorithm	165
3.2.4	General Algorithm	166
3.3	Extra Details for ASG Architecture	167
4	Data Generated for validation using AI tools	171
4.1	Software Generated Using Bing	171
4.2	Group Generated Using Bing	174
4.3	Software Generated Using Bing	177
4.4	Techniques Generated	180

List of figures

1.1	ASG Architecture Including State-Based and Threat-Based	9
3.1	Timeline of Destructive Malware Undocumented Attacks	35
3.2	Declare Infection Flow	37
3.3	The ICS Cyber Kill Chain by M. Assante and R. Lee [1]	39
4.1	Data Production Model	50
4.2	Developing ICS Prediction Model	53
4.3	Learning Curve Technique Prediction Model	55
4.4	Learning Curve Incident Prediction Model	55
4.5	Learning Curve Group Prediction Model	56
4.6	Validation Score SVM with Techniques	57
4.7	Validation Score SVM with Incidents	58
4.8	Validation Score SVM with Groups	59
5.1	ASG Tool Architecture	69
5.2	Visual Representation of State-Based Scenario Generation Process	76
5.3	State-Based Scenario Generation Process: Port Scanning	77
5.4	State-Based Scenario Generation Process: CVE-2016-3714 Exploitation	77
5.5	State-Based Scenario Generation Process: System Investigation	78
5.6	State-Based Scenario Generation Process: Download (Wget)	78
5.7	State-Based Scenario Generation Process: Scenario Generated	79
5.8	The example of the process of scenario generation: completion of generating a scenario	80
5.9	Main Interface	81
5.10	OS Selection from the State Drop Down	82
5.11	Linux Selection from OS State	82
5.12	Techniques Selection	83
5.13	Pre and Post Conditions	83

6.1	Attack Scenario Generation Methodology	89
6.2	Adversary Emulation Use case 1.a	98
6.3	Adversary Emulation Use case 1.b	99
6.4	Adversary Emulation Use case 1.c	99
6.5	Adversary Emulation Use Case 2.a	100
6.6	Adversary Emulation Use Case 2.b	100
6.7	Adversary Emulation Use Case 2.c	101
6.8	Cyber Threat Intelligence Enrichment Use Case 1.a	102
6.9	Cyber Threat Intelligence Enrichment 1.b	103
6.10	Cyber Threat Intelligence Enrichment 1.c	103
6.11	Red Team Use Case 1.a	104
6.12	Red Team Use Case 1.b	105
6.13	Resource Exhaustion Attack on ICS devices over the network	109
6.14	man-in-the-middle (MITM) attacks	110
6.15	ASG Graphical User Interface 1	112
6.16	ASG Graphical User Interface 2	113
6.17	Example of Using the Technique Prediction Model	114
6.18	Interface for Incident Prediction Model	115
6.19	Example of Using the Incident Prediction Model	115
6.20	Interface for Group Prediction Model	116
6.21	Example of Using the Group Prediction Model	117
6.22	Interface for Threat-Based Scenario Generation Model	118
6.23	Interface for Generated Scenarios	119
6.24	Interface Presenting Information About a Specific Scenario	119
6.25	ASG WEB Interface	122
6.26	ASG WEB Interface Options	122
6.27	ASG WEB Chat AI	123
6.28	ASG Chat AI: Scenario Generation	124
6.29	ASG Chat AI: Generated Scenarios Analysis	124
A.1	Common Behavior Focused on Adversary use case	144
A.2	Behavioral Analytic Development Use Case Using ASG AI1	145
A.3	Behavioral Analytic Development Use Case Using ASG AI2	146
A.4	Behavioral Analytic Development Use Case Using ASG AI3	147
A.5	The Questionnaire Template	148
A.6	Results	149
A.7	Use Case Explaining Threat-Based Model	151

3.1	threat based model scenario generation using tree	168
3.2	System State Model scenario generation using Graph	169
3.3	ASG web based Architecture	170
3.4	Attack scenario Generation Methodology	170

List of tables

2.1	Noticeable Cyber Incidents in ICS	19
2.2	Mapping of Related Work to Research Gaps and Challenges	29
3.1	ICS Cyber Kill Chain Steps Mapped to Shamoon Attack	42
3.2	ICS Cyber Kill Chain Mapped to Destructive Wiper Malwares	43
4.1	Inputs Describing the MITRE ATT&CK of ICS	51
4.2	Performance of ICS Models	60
4.3	Validation of ICS Models	60
5.1	Initial System States	72
5.2	Attack Techniques	73
6.1	General Methodology for Attack Scenario Generation	92
6.2	Comparison of Methodologies Across Key Metrics	95
6.3	Comparison of Methodologies for Analyzing Attack Scenarios	97
6.4	Interviewer result using manual methodology	107
6.5	Interviewer result using manual methodology	108

Glossary

Term	Definition
TF-IDF Vectorizer Model	Transforms text into meaningful numerical representations used for fitting machine learning algorithms.
System State	A data structure describing the initial state, including IP addresses, ports, acquired privileges, and operating system state related to the target system.
Adversary Emulation	Simulates real-world attack scenarios to identify vulnerabilities and enhance cybersecurity posture.
API	Application Programming Interface; software with a distinct function that enables applications to interact.
NLP	Algorithms that enable computers to understand, interpret, and generate human language.
Cyber Incident	An attack attempt or breach of security measures in digital systems, networks, or services.
CTI	Cyber Threat Intelligence, which reconstructs past attacks to study tools, patterns, and techniques.
TSP	The Travelling Salesman Problem, a mathematical problem involving nodes, edges, weights, and rules.
DWM	Destructive Wiper Malware targeting vulnerabilities in ICS infrastructures to damage and destroy data.
APT	Advanced Persistent Threat, referring to advanced computer system threats by nation-state actors.
TTPs	Tactics, techniques, and procedures to refine defense measures and improve security training.
FSM	Finite State Machines based on automata theory for language generation.

SpaCy	AI model for natural language processing, handling large text sets for semantic and syntactic analysis.
SVM	Support Vector Machine, a versatile machine learning algorithm for classification, regression, and anomaly detection.
STRIPE	Technology company building economic infrastructure for the internet.
Technique Prediction Model	Processes input text, focusing on synonyms and phrases using pre-trained NLP models.
Incident Prediction Model	Enhances knowledge by interpreting techniques to predict incidents.
Group Prediction Model	Processes group descriptions to predict their behavior.
Attack Scenarios	Hypothetical situations illustrating adversary actions to identify vulnerabilities and test defenses.
Techniques	Tools or methods used by attackers, such as phishing, malware, and brute-force attacks.
Tactics	Strategies attackers use to achieve goals, including reconnaissance and privilege escalation.
ICS	Industrial Control Systems that monitor and control industrial processes, often critical infrastructure.
ASG	Attack Scenario Generator detecting past attacks and suggesting techniques and tactics to counter them.

Chapter 1

Introduction

Industrial Control Systems (ICS) emerged as a popular field and refers to the computerized systems attuned to monitor and control physical processes in various industries. There is a large number of industries such as power plants, water treatment facilities, manufacturing plants, and transportation systems. ICSs operate widely within the Operational Technology (OT) environment and the industrial correspondent to Information Technology (IT) and telecommunication which makes it a highly heterogeneous environment [2].

ICSs operate widely within the Operational Technology (OT) environment and the industrial correspondent to Information Technology (IT) and telecommunication which makes it a highly heterogeneous environment [3]. Due to various requirements coming from the plurality of structures, devices, protocols and application contexts, communication subsystems must be increasingly heterogeneous.

ICSs itself is heterogeneous by nature. This heterogeneity can be attributed to numerous factors such as differences in technological frameworks, communications with a variety of sensors and interfaces with a completely changed set of architecture, interactions with legacy codes and many such scenarios. This heterogeneity however creates a complex security landscape that attackers can potentially exploit and interrupt the overall functionalities along with financial damage to the industry. For instance, if a system has different communication protocols and each protocol is autonomously secured, the whole system is not automatically guaranteed to be secure. When they are integrated with each other, the possibility of a new kind of vulnerability coming into existence cannot be disregarded and may be prone to a variety of attacks [4]. In case of cross-protocol attacks that may happen due to integrating different protocols, the security analyst further investigates the problem. The security analyst is responsible for the investigation, as the analyst's initial roles are monitoring computer networks for security issues, investigating security breaches and other cyber security incidents,

in addition to installing security measures, especially in the cyber domain. The analyst may trace the attacker's technique and recreate the attack scenario for a deeper understanding.

It is pertinent to understand what actually is an attack scenario. It may be considered a hypothetical representation of a cyberattack [5]. It can be represented as a sequence of attack techniques that help the adversary to compromise a system [6]. Attack scenarios may be sequences of events or actions that potential attackers might undertake to exploit vulnerabilities in a system, network, organization, or any other target[7]. Attack scenarios are typically created to help security professionals, analysts, or organizations to identify potential threats and consequences of a successful attack, assess risks, prioritize security measures, and develop appropriate countermeasures to mitigate the potential damage. Unfortunately, there are no standard practices or methodologies to create attack scenarios other than relying on their knowledge and experience, which may be different from person to person [8]. The lack of standardization and automation of creating attack scenarios and the reliance on static templates of graphical security models makes the process non-flexible and somewhat vulnerable to an advanced set of attacks[9]. This PhD thesis aims to propose a primitive general methodology that guides security analysts to create attack scenarios based on analyzing the previous attacks that reaches the similarities between these attacks and extracting guidelines to solve the previous problems related to attack scenarios. This chapter mainly discusses the research motivation, problem formulation, research questions, aim of the study, challenges, proposed solution and expected contribution followed by the thesis outline.

1.1 Problem Formulation

As discussed in the previous section, ICSs are large-scale, mission-critical systems that operate within diverse industrial environments. Due to their operational significance, they form an essential part of national infrastructure and are thus attractive targets for cyberattacks [10][11]. The convergence of digital technologies, legacy systems, and heterogeneous communication frameworks makes ICSs particularly vulnerable to exploitation.

Cybersecurity teams, particularly analysts, play a vital role in mitigating these threats by reconstructing cyberattacks as part of forensic investigations or proactive defense exercises. These reconstructions are typically represented through *attack scenarios*, which help understand the attacker's strategy and test the resilience of ICS environments. It may be noted that the construction of such scenarios is currently performed manually, relying on individual expertise, historical reports, or anecdotal evidence. This manual labor of scenario construction leads to several critical limitations:

- **Lack of standardization:** There is no unified method or formal model that guides how attack scenarios should be constructed or validated.
- **Subjectivity and inconsistency:** Due to differences in experience and interpretation, analysts may produce different scenarios for the same attack case.
- **Inability to respond dynamically:** Static models fail to adapt to changes in system state or emerging attacker behaviors.

Given the potential physical and financial consequences of cyberattacks on ICSs, it is imperative to have a more structured, data-driven, and adaptive mechanism for generating attack scenarios. The absence of such a mechanism not only delays the detection and mitigation process but also increases the risk of overlooking advanced attack vectors.

Therefore, this thesis addresses the problem of generating accurate and adaptable attack scenarios in ICSs by proposing a general methodology informed by historical data, system state analysis, and technique classification. This methodology aims to improve the consistency, accuracy, and effectiveness of scenario generation for both proactive defense and post-incident analysis.

1.1.1 Aim of the Research Study

The aim of this study is to analyse the significant ICSs attacks. After thorough analysis, compile the findings into a comprehensive representation, and make it able to create a general methodology that aids in the process of generating attack scenarios. These may include the attack scenarios that may take place and may have similarities to past attacks. By this, an attack can be effectively reconstructed and will also let the analyst explore new directions to investigate.

1.2 Research Questions

The aim of this research is to develop a standardized, adaptable, and data-driven methodology for generating realistic cyber-attack scenarios in Industrial Control Systems (ICSs). To address this aim, the following research questions (RQs) are formulated. Each question is grounded in the specific research challenges identified (See 1.3) in the thesis and is systematically addressed in the subsequent chapters.

RQ1: Can a standardized and systematic methodology for ICS attack scenario generation be developed?

RQ1 addresses the absence of a unified framework in existing literature for generating structured and repeatable attack scenarios in ICS environments. By designing a reproducible methodology, this research aims to establish a consistent foundation for attack scenario synthesis.

Linked Challenge: Scenario Generation Mechanism (Challenge 2)

Addressed in: Chapter 3 (Methodology Design)

RQ2: How can the classification and alignment of attack techniques with the ICS cyber kill chain improve scenario relevance and accuracy?

Effective scenario generation requires accurate mapping between techniques and the phases of an attack lifecycle. RQ2 explores how such classification, particularly using the ICS cyber kill chain, can enhance the contextual precision of generated scenarios.

Linked Challenge: Scenario Generation Mechanism (Challenge 2)

Addressed in: Chapter 3 (Attack Technique Classification Framework)

RQ3: Can previous ICS cyber incident data be leveraged to enhance the predictive capabilities of scenario generation?

RQ3 examines the use of real-world or synthetic incident data to predict attack patterns. Given the scarcity of high-quality ICS datasets, the work investigates data augmentation techniques using machine learning.

Linked Challenge: Lack of Data (Challenge 1)

Addressed in: Chapter 4 (Data Augmentation and Scenario Prediction)

RQ4: How effectively can real-time system state information be used to generate dynamic and adaptable attack scenarios?

Unlike static approaches, RQ4 investigates incorporating real-time or near-real-time system state data to dynamically construct attack paths that reflect evolving ICS environments.

Linked Challenges: Time Complexity (Challenge 3), Lack of Host-Level Mapping (Challenge 4)

Addressed in: Chapter 5 (State-Aware Scenario Generation)

RQ5: How can past incident data be used to optimize and tailor generated attack scenarios?

RQ5 builds on RQ3 but shifts focus from prediction to optimization. It evaluates how historical data can be used to refine and validate generated scenarios, making them more realistic and efficient.

Linked Challenge: Lack of Data (Challenge 1)

Addressed in: Chapter 4 (Scenario Optimization and Refinement)

Validated in: Chapter 6

RQ6: To what extent can threat actor prediction improve the authenticity of generated attack scenarios?

Different adversary groups exhibit different behavioral patterns and strategies. RQ6 evaluates how integrating threat actor profiling and prediction can increase the realism and applicability of the generated scenarios.

Linked Challenge: Lack of Host-Level Mapping (Challenge 4)

Addressed in: Chapter 5 (Threat Actor Prediction)

Validated in: Chapter 6

These six research questions collectively target the technical and methodological gaps identified in existing literature, particularly regarding data limitations, model flexibility, system-state awareness, and actor realism. The structured framework proposed in this thesis answers these questions through a combination of machine learning, graph theory, system modeling, and empirical validation.

1.3 Research Challenges

Some challenges associated with the attack scenario generations have been discussed briefly. Challenges may include a lack of expert knowledge and a limitation of time required. Furthermore, while undertaking the study, it was observed that there are more challenges that are important to discuss separately and need urgent research attention. In order to get better insights into the challenges, a detailed para-wise discussion is presented.

1. **Lack of Data:** One of the major challenges is the absence of data. The bulk of related data is deemed necessary to create modules to predict suitable techniques. These techniques can be derived from text or system statuses.
2. **The mechanism of obtaining scenarios** If the input is related to all nodes, it is impossible to generate a scenario, analogous to The Travelling Salesman Problem

(TSP), which is a well-known mathematical problem, and also in computer science as an NP-hard problem. While it is beyond the scope of this thesis to discuss the TSP, there are nodes and edges involved, along with weights and some set of rules. This study uses similarity between the nodes as a threshold to create relationships between nodes that will decrease the relation between the techniques.

3. **Time complexity** Formally, time complexity is the amount of time required for an algorithm to execute. In the context of this thesis, it describes the amount of time required to generate a scenario. It is challenging as the delay in extracting the information could lead to more attacks.
4. **Lack of host-level information** In the host-level generating scenario, an issue emerged while mapping between techniques and the state-level key values that were represented in the proposed solution as pre-condition and post-condition. Similarly, in the state-level generation scenario, the lack of state-level information caused difficulties in mapping techniques to these key values.

From the above mentioned aim, research questions, and identified challenges, the severity of the problem can not be ignored, and it demands a robust solution. The proposed solution is provided to address these issues.

1.4 Proposed Solution

In the first place, the proposed solution initially gathers the major incidents of cyber attacks on major companies, as they are well-representative of the effect of cyber attacks on governmental levels, e.g., Saudi Aramco[12]. The attacks will be compiled and documented, including the recent attacks such as the ZeroCleare attack[13]. Furthermore, ICS *cyber kill chain* will be employed to characterize three different attacks and investigate their similarities. The practice is to create and develop a Bespoke destructive wiper malware taxonomy, which can detail each step of such attacks. This taxonomy will help security professionals to ensure that their systems are secured against such attacks.

Secondly, using the ICS *cyber kill chain* and threat attack knowledge base (such as MITRE ATT&CK). The classification of attacks and their techniques(attack steps) in each step has to be figured out, and it has to be investigated which techniques are being used at what stage of the attack by leveraging the power of the ICS Cyber Kill Chain. The "Attack Scenario Generator" as shown in Figure 1.1, will be responsible for generating the attack scenarios on the textual level, by searching the techniques database. After that, the results are combined to understand all the aspects as defined above. Furthermore, the attack scenario

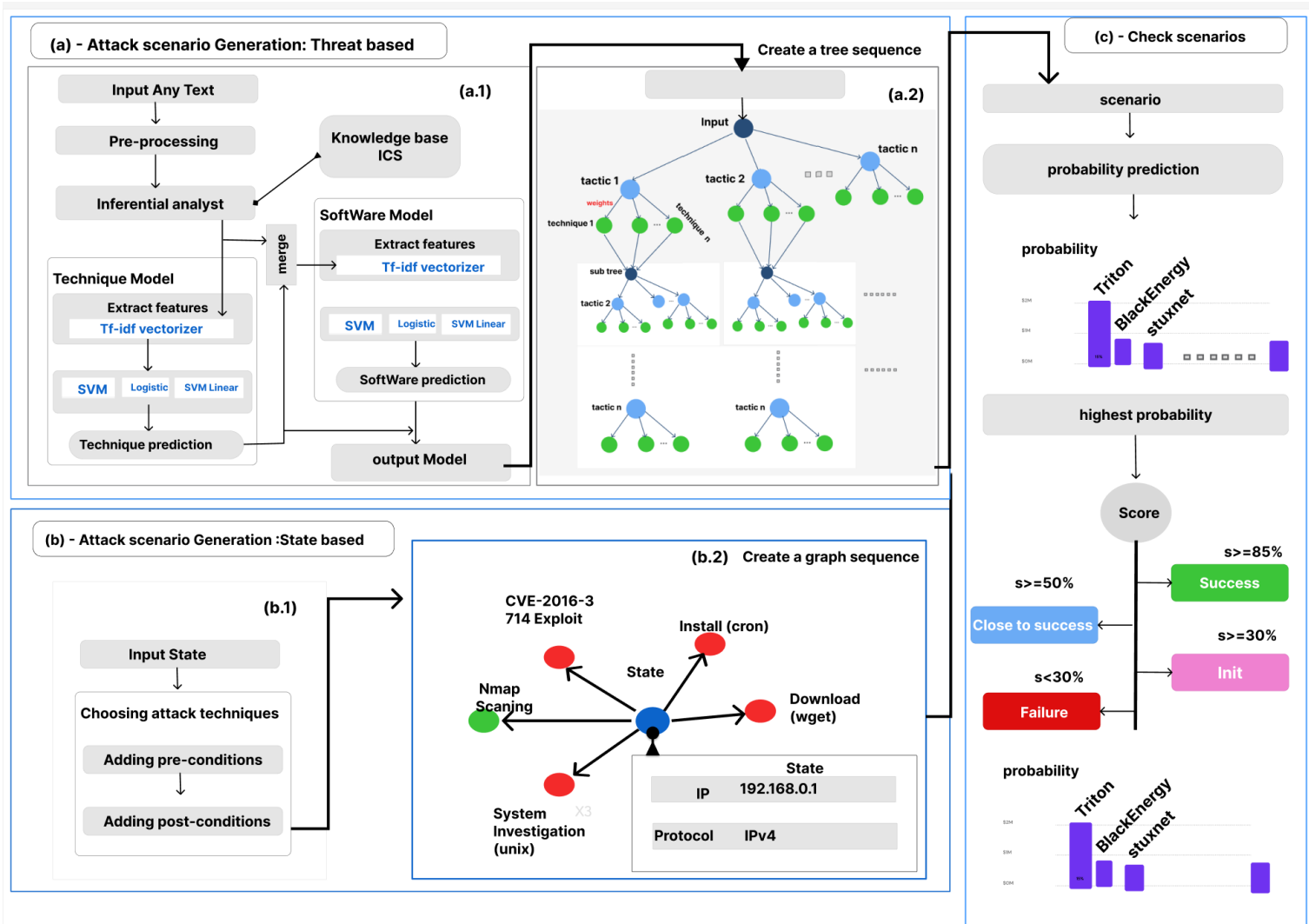


Fig. 1.1 Attack Scenario Generator Architecture Including State-Based and Threat-Based

generator may take into account the system state initially. With the inspiration from STRIPS, which is an automated planning method, two primary concepts are defined.

- **State:** The state information makes the attack hypothesis generator understand the current situation of a target system using key values (e.g., IP address, port list, and gained privilege).
- **Attack Technique:** The attack technique is a unit of attacks, and it has pre-conditions and post-conditions.

Now, the attack scenario generator employs the attack technique having a higher success chance based on the state information. It may be noted that the overall process depends on the system state. Finally, a general methodology is extracted for generating attack scenarios from three different models, namely Manual, Threat-Based Based and State-Based; details of them are covered in Chapter 6 of the thesis.

1.4.1 The Rationale of Using Manual/State/Threat phases

The rationale behind this research stems from the critical need for improved methods in generating attack scenarios for ICS environments. Currently, there is a significant lack of standardization and automation in this field. Therefore, the researcher followed a three-layer methodology that started with manual analysis of the previous attacks to identify common attack patterns, state-based and threat-based layers to create an Attack Scenario Generator (ASG) tool specifically designed for ICS environments. The ASG tool offers significant advancements by automating scenario generation, predicting potential attacker groups, and optimizing scenarios based on historical data.

1.4.2 Major Contributions

Below are some of the major contributions of the research work.

- Proposed a general methodology to generate attack scenarios of ICS. This contribution answers the first research question "How to Create an Attack Scenario for ICS?" which is discussed in chapter 6.
- Proposed a manual methodology for analysing ICS cyber incidents to investigate the common grounds by leveraging the ICS cyber kill chain framework. This specific contribution tries to relate the ICS's previous attacks. This contribution answers research question 2, "How can the classification and alignment of attack techniques

with distinct stages of the ICS cyber kill chain enhance the precision and relevance of generated attack scenarios?" which is discussed in Chapter 3.

- Development of Attack Scenario generator(ASG) tool, that has the capacity to automatically generate an attack scenario and predicts the adversary group being the attacks by utilising the threat intelligence knowledge base of the previous attacks. This contribution answers the sub-research question "How can threat intelligence knowledge base of the previous attacks be utilized to generate attack scenarios?" which is found in Chapter 4.
- Proposed a data production model for attack techniques along with software, which can be effectively used in other applications. It is a result of utilizing attack scenario generation and it is also found in chapter 4.
- A model is created that can optimise the attack scenarios that have been generated depending on ICS's previous cyber incidents. It is one of the contributions of the study conducted in chapter four.
- Proposed a model using system state information to create an attack scenario, and this answers the research question " How far can the attack scenarios be generated and the attack techniques be predicted using system state? " which is answered in Chapter 5.

1.4.3 Thesis Structure

In this research thesis, Chapter 2 defines the background and related work. It also discusses the ICS, its heterogeneity and some famous ICS-related cyber incidents. Chapter 3 highlights the Destructive Wiper Malware and its detailed review with reference to major companies. Furthermore, this chapter also elaborates on the Cyber Kill Chain, Manual Attack Scenarios Creation is discussed in the same Chapter. Chapter 4 is based on Threat-based attack scenario generation. Chapter 5 defines the State-based attack scenario generation. Chapter 6 enlightens on the General Methodology to create attack scenarios for ICS. Finally, Chapter 7 concludes the thesis. The appendix provides any additional information deemed necessary to be included in this thesis.

Chapter 2

Background

In order to better comprehend this chapter, it is vital to be educated on the basic definitions of some common terms within the context of this thesis.

2.1 Definitions

An attack scenario: describes a potential sequence of actions an attacker might take to compromise a system. It outlines the attacker's goals, methods, tools, and potential outcomes [14].

Techniques: According to the Information Systems Security Association (ISSA) (2023), attack techniques in cybersecurity encompass the specific methods attackers utilize to exploit system vulnerabilities and achieve their objectives. These techniques range from basic social engineering tactics to intricate code injection attacks. A thorough understanding of common attack techniques is essential for security professionals to develop efficient defense strategies[15].

Tactics: Cybersecurity tactics are specific actions or countermeasures taken to defend against cyber threats and vulnerabilities. They focus on directly mitigating risks and preventing attacks from succeeding. Tactics are implemented within a broader cybersecurity strategy, which outlines the overall goals and approach to protecting information systems [16].

2.2 Introduction

This chapter presents relevant background knowledge concerning the established contributions. This chapter also explains ICS security challenges and highlights the current situation in terms of attack scenario generation for ICS.

The following stepwise sections elaborate background of the overall thesis. Section 2.3 discusses the different approaches for ICS and their heterogeneous nature in terms of architecture, communication, and data. Furthermore, cyber incidents are presented along with a brief description. Section 2.4 follows on describing attack scenario generation using a variety of available frameworks such as MITRE and ATT&CK. Some notable related work is presented in the section 2.5

2.3 Industrial Control Systems (ICSs)

ICSs are used in crucial infrastructures like heavy industries of giant ships, towering skyscrapers, steel plants, power stations, water treatment facilities, and nuclear/thermal plants, besides numerous other industries. The term "ICS" includes various control systems, such as Supervisory Control and Data Acquisition (SCADA), Distributed Control System (DCS), and Programmable Logic Controller (PLC) configurations. ICS are wireless and control components that include control loops, Human Machine Interfaces (HMIs), and network protocols used in industries (e.g., manufacturing, transportation). Industrial Control System is used to refer to all components.

In the context of ICS, the SCADA system is used to collect and present data to the operators, make decisions about a specific process with the assistance of operator input, and control plant functions based on those decisions. The DCS's role in plant automation is similar to the SCADA system. A PLC is a ruggedized and adapted industrial computer for the control of manufacturing processes, such as assembly lines, machines, robotic devices, or any activity that requires high reliability, ease of programming, and process fault diagnosis.

Until long the ICSs were not utilising the advantages of the Internet to the full extent. However, recent advancements let the remote components communicate by using specialized protocols and private networks, including Modbus RTU, Modbus TCP, Wi-Fi, Z-Wave, and Zigbee. Similar protocols include CIP, AC-i, DeviceNet, and HART, which are widely used for the sake of automation. Industries currently recognize the advantages of the Internet and IT and are also leveraging the power of cloud computing. With such a diverse nature of the ICS, there comes the integration of different components, leading to a heterogeneous nature of the ICS.

2.3.1 ICS Heterogeneity

ICS is the backbone of any modern nation's Critical Infrastructure (CI), such as the water, energy, transportation, and Information and Communication Technology (ICT) sectors.

There has been a significant interest in computer network technologies recently, leading to satisfactory research in the same domain. Industrial communication, particularly, is taking serious steps to adopt the state-of-the-art technologies. Despite the new technologies, businesses that have evolved from the old systems are still using the old technologies, which on one hand are mature enough to run the businesses but lack to cope with the recent technological advancements.

The majority of these systems were developed in the early 90s, making them hard, if not impossible, to integrate with newer technologies. The latest trends and innovations in operational and information technologies bring new products that may be integrated into the existing systems or may be used as an add-on to the already developed products.

Businesses that lagged in adopting the latest trends are not providing strong incentives to refit (brownfield method), maintain, and run outdated systems in whole new ways, with solutions emerging from architectural paradigms generated from recent research activities. Therefore, the most crucial difficulty in such an advancement process is improvements in industrial networks and communication technologies that can function in a heterogeneous environment. ICS heterogeneity has different major forms. It can be in architecture, communication, or data. The different forms of heterogeneity are elaborated in the preceding sections.

- **Heterogeneity in Architecture**

Heterogeneity in Architecture may spread across various layers. The **Physical layer** is composed of field devices such as sensors and actuators, which interfaces with the physical processes, the Programmable Logic Controllers (PLC) and Remote Terminal Units (RTUs). These can be controlled locally or remotely and are supported by the **Control layer**. The Human Machine Interface (HMI), data historians, and other workstations are housed in the **Supervision layer** to allow real-time monitoring and control. Enterprise servers and applications for production management are placed in the **Corporate layer**.

With so many layers working under the architecture. Controlling them remotely mandates the use of networks, i.e., wired or wireless networks. Therefore, ICS architecture makes use of several protocols. An attacker can manipulate loopholes and exploit the protocols to damage the system or take control of the system. Also, protocols may be safe when used independently, but may be vulnerable to attacks when used in conjunction with other protocols, e.g., Wifi and Zigbee.

- Heterogeneity in Communication

As discussed in the previous section, the ICS systems are spread across various layers and with different devices deployed in different locations, they are interconnected with each other via networks. It is not necessary that the devices may be deployed in a single location or that the whole system will exist in a unified manner. The distributed architecture is employed to manage such scattered systems. These systems form a group of several nodes that are spatially scattered throughout the industry and linked by a communication network.

The features needed for the management and maintenance of a manufacturing process are provided by a comprehensive system, which can be conceived of as a holistic but abstract entity. Such a system frequently deals with data that comes from an industrial process in which the data is coded as a collection of facts and allocated as application variables in the memory of subsystem devices. To ensure a proper outcome from the standpoint of the application, the necessary subset of these data is cooperatively transferred across nodes. Each local system communicates with other systems both internally and externally to exchange data. Industrial networks are the main option that can provide local and remote communication services with real-time transfers and a high level of dependability. In practice, it is critical that IO data be distributed in a timely and reliable manner to the appropriate units. These parts, which serve as the distributed system's nodes, are often real-time controllers. A system application requires a proper communication network to share common data entities and for mutual synchronisation since programs in these distributed nodes handle data locally or remotely. Theoretically, the ideal networking solution in this scenario would be to have a single global network that can be used across all levels and provide a wide range of coordinated services, but this is not practically possible.

Currently, a combination of well-known and novel technologies from the OT and IT domains allows for the creation of dispersed system applications that can work in a multi-functional and heterogeneous scope. This makes it possible to develop distributed data processing solutions for both local and remote communication.

- Heterogeneity in Data

There are numerous approaches to categorise data heterogeneity. Some related approaches are discussed below.

- **Syntactic Heterogeneity** occurs when two data sources are not described using the same knowledge representation formalism. For example, when combining ontologies, F-logic [17], and OWL [18].

- **Terminological Heterogeneity** is when different names for the same thing may be used. This may be due to the context of the problem or domain dependency, or simply when standards are not followed for naming conventions.
- **Semantic Heterogeneity** occurs when numerous models are applied to the same topic of inquiry (for instance, when various axioms are employed to describe concepts)
- **Semiotic Heterogeneity** refers to varying interpretations of the same entity by different people.

With that much diversity and heterogeneity, these systems are vulnerable to attacks. In the next section, some cyber attacks are summarised.

2.3.2 Securing Heterogeneous ICS with Attack Scenarios

Attack scenarios are constructed to overcome the heterogeneity of ICS (ICS). According to a study presented in [19], a sequence of attack techniques can help to understand the threat landscape of heterogeneous ICS. The researchers suggest a model that aggregates attack means into limited attack techniques according to word clustering to address the classification challenge caused by unknown attacks. Then, similarity and statistical methods are employed for anomaly detection and classification, utilizing algorithms like decision trees, neural networks, and clustering techniques. Different such methods and models are combined to predict the next attack technique. Subsequently, the hidden Markov model is used to map attack techniques and security states. condition of a system, device, or network, in terms of its own security posture, are incorporated to obtain the forecasting results of the next security state. Attack scenarios can help to identify common vulnerabilities. ICSs are often heterogeneous, meaning that they use different hardware and software from different vendors. This can make it difficult to identify and patch common vulnerabilities. Attack scenarios can help to identify these vulnerabilities by simulating attacks on different ICSs [20]. They can also help to prioritize security efforts. Not all ICSs are created equally; some are more critical than others and need to be protected more heavily. Attack scenarios can help to prioritize security efforts by identifying the ICSs that are most vulnerable to attack. In summary, attack scenarios can help overcome the heterogeneity of ICS by providing a framework for threat intelligence and formulating defence strategies. By using a defence model for similar scenarios, it is possible to predict future attack types and calculate risk values based on these predictions. This can help organizations prepare for potential attacks and minimize their impact.

2.3.3 Cyber Incidents in ICS

Due to the heterogeneous nature of ICS, this always leads to repetitive cyber attacks. As an example, Shamoon attack was repeated three times, which was a highly destructive cyberattack that primarily targeted critical infrastructure and energy sector organizations, particularly in the Middle East. Different attacks are represented in the Table 2.1.

Table 2.1 Noticeable Cyber Incidents in ICS

Name	Year	Description
ACAD/Medre.A	2012	It is an administrative data-stealing worm. The worm collects AutoCAD files that contain drawings. It has the potential to be used for industrial espionage.
Backdoor.Oldrea	2013	It is a modular backdoor that has been employed by [Dragonfly] since at least 2013 against energy businesses. It was disseminated through supply chain compromise and contained specialised modules to list and map ICS-specific systems, procedures, and protocols.
Bad Rabbit	2017	It is an infectious ransomware that hit the Ukrainian transportation industry in 2017. It has also targeted Russian businesses and customers.
BlackEnergy	2015	It is a malware toolset that is malicious and has also employed APT attackers. It has been around since at least 2007, and while its original purpose was to build botnets for use in Distributed Denial of Service (DDoS) attacks, it has now developed to support a number of plug-ins. It is well recognised for having been used to attack Ukrainian institutions and Georgia and Russia during their 2008 conflict.
Conficker	2008	It is a computer virus spread via the MS08-067 Windows vulnerability to attack Microsoft Windows.
Duqu		It is a virus platform that extends functionality after deployment inside a target network by using a modular strategy.
EKANS	2019	It is a Golang-written ransomware version that has been used against a variety of industries, including the manufacturing of automobiles, healthcare, and the energy industry. In some cases, this has led to severe operational interruptions.
Flame	2010	It is a complicated toolkit that has been used to gather information that largely targets Middle Eastern countries.

Industroyer		It is a sophisticated virus designed to cause an impact on the working processes of ICS, particularly components used in electrical substations.
KillDisk	2018	It is a disk-wiping tool designed for overwriting files with random data to render the OS in a state which can not be booted.
PLC-Blaster	2016	is a piece of proof-of-concept malware that runs on Siemens S7 PLCs. This worm locates other Siemens S7 PLCs on the network and attempts to infect them. Once this worm has infected its target and attempted to infect other devices on the network, the worm can then run one of many modules.
REvil	2019	It is ransomware used against organizations in the manufacturing, transportation, and electric sectors. It is highly configurable and shares code similarities with the GandCrab RaaS.
Ryuk	2018	It is a ransomware designed to target enterprise environments that have been used in attacks, shares code similarities with Hermes ransomware.
Stuxnet	2010	It targets ICS devices. is a large and complex piece of malware that utilized multiple different behaviours including multiple zero-day vulnerabilities, a sophisticated Windows rootkit, and network infection routines.
Triton	2017	It is an attack framework built to interact with Triconex Safety Instrumented System (SIS) controllers.
VPNFilter	2018	It is a multi-stage, modular platform with versatile capabilities to support both intelligence-collection and destructive cyberattack operations. It can collect traffic that passes through an infected device, allowing the theft of website credentials and monitoring of Modbus SCADA protocols.
WannaCry	2017	It is ransomware that affected more than 150 countries. It contains worm-like features to spread itself across a computer network using the SMBv1 exploit EternalBlue.

Shamoon	2012	
Shamoon 2	2016	The virus was famous because of how costly it was to recover from and how devastating it was. A machine that is infected can transmit Shamoon to other machines connected to the network.
Shamoon 3	2017	Once a machine has been infected, the virus keeps a list of the files from particular locations on the system, uploads them to the attacker, and then deletes them. Finally, the virus overwrites the computer's master boot record, rendering it useless.
Dark Seoul	2013	One single entity was responsible for the attacks on six organisations. Malicious software rather than distributed denial-of-service (DDoS) attacks, as was initially thought, targeted the networks. It looked like it was simply using hard drive overwrites.[5] 32,000 PCs and servers belonging to media and financial institutions were damaged in this cyberattack.
Destover	2014	"Destover," also known as "Wipall," has been identified as the destructive "wiper" virus that was used to infect and wipe hard drives at Sony Pictures Entertainment.
Stone Drill	2017	It is a highly destructive wiper malware that targets Saudi Arabian companies, including those in the public sector, business, transportation, and telecommunications. Attacks in Europe were also carried out using this malware.
Olympic Destroyer	2018	The virus targeted computers which have information about; the International Olympic Committee (IOC), South Korean people and officials, Olympic athletes, partners, and tourists, in the 2018 PyeongChang Winter Olympics.
LockerGoga	2019	This virus changes the passwords for the user accounts on the infected system. Additionally, it tries to log off people who are currently logged in. Then it would go into a temporary folder and rename itself via the command line (cmd).

MegaCortex	2019	It is a kind of malware that restricts users' access to their computer or personal files and requests ransom payments to allow access
ZeroCleare	2020	It is a malicious software. It has been designed to destroy a disk's partitioning by wiping the master boot record section.

Table 2.1 contains a list of the significant cyber incidents to the ICSs. The threat types include targeted attacks, malware, cyber-intrusion campaigns, and cyber-threat groups. It is pertinent to mention that Table 2.1 focuses on notable cyber threats, incidents, and campaigns that target ICS equipment and critical infrastructure. This is not an exhaustive list; rather, it portrays the risks that the ICSs systems are facing. The attacks cited in Table 2.1 caused significant damage to ICS systems financially and, in some instances, disrupted their overall process, halting or even resulting in a complete shutdown for proper maintenance.

2.4 Attack Scenarios Generation

As discussed earlier, there is a dilemma in terms of ICS cybersecurity due to its complexity and heterogeneity. To elaborate further, this section explains the idea of attack scenarios and how they contribute to cybersecurity development. Attack scenarios are a foundational component of effective cybersecurity strategies [21]. Attack scenarios allow organizations to anticipate, prepare for, and mitigate cyber threats by identifying vulnerabilities, improving defences, and enhancing incident response capabilities. By regularly testing and refining attack scenarios, cybersecurity professionals can stay ahead of cyber-criminals and safeguard digital assets and sensitive information.

To simulate prospective cyberattacks and evaluate an organization's vulnerabilities, attack scenarios are generated methodically using a variety of approaches and frameworks. Some prevalent methodologies and approaches, which are discussed shortly, include the cyber kill chain, MITRE ATT&CK Framework, STRIDE Model, Attack Trees, Threat modeling and Red team exercises, Penetration Testing, Tabletop exercises, NIST, and Scenario-based Risk Assessment.

- Cyber Kill Chain

The cyber kill chain, a foundational model in cybersecurity, outlines the distinct stages attackers follow during an intrusion. As described in Lockheed Martin's influential 2011 whitepaper

[22], it typically consists of seven phases: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. By understanding these phases, security professionals can implement targeted countermeasures at each stage to disrupt the attack lifecycle and prevent attackers from achieving their goals. Research article published in 2019 by Shackelford et al.[23] further emphasizes the importance of extending the kill chain beyond technical measures to incorporate human factors and deception techniques, creating a more holistic approach to defense.

- MITRE ATT&CK Framework

The ATT&CK matrix provides a reliable source for institutions to develop reality-based attack scenarios depending on real-world adversary procedures and tactics. This enables security teams to better align their defenses with the actual threats they might face. However, it is important to highlight that the ATT&CK matrix is not exhaustive and may not cover all expected attack vectors. Additionally, it requires consistent updates to reflect the threat landscape and may not fully capture the complexity of the sophisticated attacks.

- STRIDE Model

The STRIDE methodology provides security professionals with a structured approach towards developing attack scenarios. It involves brainstorming expected threats within each of its six categories (Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service, and Elevation of Privilege) and afterwards, evaluating their impact. As outlined in [16], this method offers several advantages: it fosters a systematic approach to threat classification, it is adaptable to different applications and systems, and it helps in prioritizing specific threats. However, limitations also exist. Categorization can be subjective, with some scenarios that might fit into multiple categories. Additionally, STRIDE doesn't encompass all expected attack vectors and lacks a step-by-step and comprehensive simulation of these scenarios.

- Attack Trees

While attack trees offer benefits such as visualizing attack pathways, identifying vulnerabilities, and assessing attack complexity, they also come with limitations. Constructing intricate trees can be time-consuming, and their complexity can make them difficult to manage and comprehend. Additionally, attack trees may not effectively capture the dynamic and evolving nature of attacker behavior, potentially overlooking novel attack strategies.

- Risk Modelling

Threat modeling offers a structured approach to analyzing system architectures and trust boundaries, providing valuable insights into potential vulnerabilities and attack vectors. This proactive technique enhances understanding of potential threats and helps prioritize security measures. However, it does require familiarity with specific methodologies, and its focus on architectural analysis might not fully capture the intricate details of sophisticated attack scenarios. Additionally, detailed simulations may not be as prevalent within the traditional threat modeling process.

- Red Team Training

Red teaming offers valuable insights into an organization's security posture by simulating real-world attacks and exposing vulnerabilities and weak points. This allows defenders to assess the effectiveness of their existing defenses against actual threats. However, it requires significant time and effort, demanding meticulous planning and execution to avoid disrupting production systems. Additionally, the success of red teaming heavily relies on the ingenuity and knowledge of the red team, which can introduce potential limitations if their capabilities are insufficient.

- Penetration Testing

Penetration testing offers a valuable approach to cybersecurity by actively uncovering vulnerabilities through simulated attacks. It provides a practical method for identifying security flaws and delivers actionable insights that directly contribute to remediation efforts. However, limitations exist. Penetration testing may not accurately capture the intricate details of highly sophisticated attacks, and it can be resource-intensive to conduct. Additionally, these tests often focus on specific targets within a system, potentially overlooking broader vulnerabilities present in the overall environment.

- Tabletop Exercises

While tabletop exercises offer a number of benefits, such as fostering collaboration among stakeholders, evaluating response plans, and enhancing an organization's overall cyber readiness, they also have limitations. Firstly, they may not fully capture the dynamic nature of real-world attacks, potentially leading to complacency or overlooking unpredictable scenarios. Additionally, these exercises often assume participants possess a comprehensive understanding of various attack situations, which might not always be the case, potentially hindering the effectiveness of the exercise.

- Scenario-based Risk Assessment

Security teams conduct systematic risk assessments using scenario-based risk assessment, which involves developing and analysing fictitious attack scenarios. Organisations can use these scenarios to prioritise risks, allocate resources, and create risk mitigation plans [24].

- NIST Modelling

The National Institute of Standards and Technology (NIST) illustrates that risk modeling is a systematic process of recognition, analysis, and assessment of potential threats, vulnerabilities, and their expected impacts on an organization's information systems. NIST represents a risk-based approach to security, where organizations prioritize security measures according to the likelihood and severity of potential risks. This approach includes identifying assets, threats, vulnerabilities, and calculating potential impacts. It helps organizations to make informed decisions about resource allocation and security controls to mitigate risks effectively. [25]

- Automated Attack Scenario Generation

Automated Attack Scenario Generation quickly develops a variety of attack scenarios. It incorporates well-known strategies, methods, and weaknesses. Furthermore, it aids organisations in staying vigilant about spotting problems. However, it may not accurately reflect cutting-edge or innovative attack techniques and may not have the depth of comprehension that is offered by threat intelligence [26].

Apart from the above-mentioned frameworks, there are industry-specific frameworks. These frameworks are customised to meet industry-specific needs and compliance standards. They give advice that is pertinent to the context of the organisation and provide models and best practices. Like any other framework, it also comes with some limitations. It might not include every possible hazard that is particular to an organisation. Furthermore, if not updated frequently, can become outdated. It mainly focuses on aspects unique to a given business rather than more general assault scenarios [27].

The selection of the most appropriate methodology depends on the research objectives, the resources available, and the specific requirements of the organization. Each approach has advantages and disadvantages, and the most thorough and efficient strategy for creating attack scenarios is most probably a combination of methodologies.

2.5 Attack Scenarios Related Works

Concerning the manual analysis of attack scenarios, Park and Lee's work[6] came up with several attack scenarios for I&C systems in Nuclear Power Plants (NPPs) to quantitatively evaluate the risk assessment for NPP. Within the power domain, Li et al [28] discuss well-known previous cyber incidents such as false data injection, denial of service attacks, man-in-the-middle, and replay attacks against power systems. They generalized the scenario for each attack method to assess their consequences. Studying further, Haas and Moller [7] created attack scenarios to test the cyber-security measures and to gauge the consequences, depending on use cases of connectivity. These case studies include e-mobility, car sharing, and a special case of autonomous driving that was for automated valet parking. They discussed the cyber-attack surfaces and some possible attack scenarios. While using attack graphs visualization, Noel et al [29] created attack scenarios. They mapped intrusion events to known exploits in the network attack graph and correlated the events through the corresponding attack graph distances. Identifying the natural application of automated planning to security [30] identifies the attacks as a sequence of steps that ultimately achieve a goal, analogous to a plan. They argue that by using automated planning, one can construct tests that can run against the systems to identify weaknesses. The authors further specifically identify how planning can be used to assess web applications by using the SQL injection technique and the SSL/TLS protocols. Other recent applications include using automated planning and plan recognition to identify larger attack paths, citeamos2017efficient, and vulnerability assessment [31]. Hoffman in their detailed survey [32] identifies the two main dimensions of existing research:

- How does the approach handle uncertainty from the point of view of the adversary?
- How do the attack components interact with each other?

Hoffman similarly enumerates eight key assumptions and surveys the literature, mapping each to its appropriate assumptions and how the approach maps to the two dimensions being identified. These assumptions include:

1. **Technological Determinism:** Technology drives societal change and influences security measures;
2. **Rationality:** Actors in the cyber domain are rational and seek to maximize their goals;
3. **Interdependence:** Cybersecurity is highly interdependent, with actions in one domain affecting others;
4. **Complexity:** The cyber domain is complex and difficult to understand fully;

5. **Uncertainty:** There is inherent uncertainty in cyber threats and vulnerabilities;
6. **Evolution:** Cyber threats and defenses are constantly evolving;
7. **Asymmetry:** There is an asymmetry between attackers and defenders;
8. **Externalities:** Cyber threats often have unintended consequences.

These assumptions create the basis for Hoffmann's analysis of cyber security issues and give insights into strategies for risks mitigation [32]. Miller et al [33] introduce the CALDERA automated adversary emulation system, which takes the system state as input and emulates the adversary action as a result. CALDERA, while a powerful adversary emulation tool, has certain limitations. One of which is its reliance on predefined adversary tactics, techniques, and procedures (TTPs). Although CALDERA provides a library of TTPs, it may not be able to accurately simulate the new or customized tactics employed by real-world adversaries [34]. OCTAVE and PASTA models are also additional threat modelling techniques that are worth mentioning.

- OCTAVE

An operational framework for risk assessment and defence against cyberattacks is the OCTAVE technique (Operationally Critical Threat, Asset, and Vulnerability Evaluation). This risk-based approach to strategic planning and assessment was created by the Computer Emergency Response Team (CERT). Because OCTAVE is a self-directed approach, individuals within an organisation take ownership of determining the security plan for the company [35].

- PASTA

Process for Attack Simulation and Threat Analysis is referred to as PASTA [35]. One of the characteristics of PASTA threat models is that they are risk-centric. The purpose of adopting a threat model is to identify, categorise, and concentrate on the risks that pose the greatest harm to the organisation. Furthermore, it can run simulations, which implies that they can be run with threats that have been recognised, evidence that has been gathered, etc. There are seven phases in PASTA. Every step contributes to the body of knowledge regarding the topic under study and its technical context.

It is evident from the literature and also to the best of our knowledge that despite all of the work dedicated to using automated planning for penetration testing, little has been done to investigate how the planning portion of the problem relates back to the acting portion for the problem as most of the approaches assume that the plan will be generated before execution, with the "acting" portion merely following the plan's script.

2.6 Synthesis and Gaps in Literature

From the reviewed literature, it is evident that existing works have significantly contributed to modeling and simulating attack scenarios in ICS environments. However, several gaps remain about the research challenges addressed in this thesis:

- **Lack of Data (Challenge 1 / RQ3 & RQ5):** While some systems like CALDERA [35] offer adversary emulation, they heavily rely on predefined TTPs, limiting adaptability to new threats. Similarly, tools reviewed in [31, 32] do not leverage synthetic or expanded datasets to improve coverage. This motivates our use of ML-based data augmentation (Chapter 4) to overcome data scarcity.
- **Scenario Generation Mechanism (Challenge 2 / RQ1 & RQ2):** Existing frameworks (e.g., OCTAVE, PASTA) focus more on risk assessment rather than generating interconnected attack steps using systematic rules. Our work addresses this through a graph-based approach that considers node similarity and connectivity, inspired but not fully supported by existing literature.
- **Time Complexity (Challenge 3 / RQ4):** The literature lacks discussions on the computational cost of generating multiple scenarios dynamically. Our approach proposes parallel processing techniques to reduce scenario generation time, which is not covered in prior frameworks.
- **Lack of Host-Level Mapping (Challenge 4 / RQ4 & RQ6):** While automated planning has been used to sequence attacks [32, 33], they often overlook low-level state-to-action transitions at the host level. This challenge is addressed in our work through the use of pre- and post-conditions for better alignment with real-time system states.

Table 2.2 Mapping of Related Work to Research Gaps and Challenges

Literature Reviewed	Gap Identified	Related Challenge / Research Question
CALDERA, Planning Tools [33, 36]	Reliance on predefined TTPs; limited adaptability to novel or customized threats; lack of data synthesis.	Lack of Data (Challenge 1), RQ3, RQ5
OCTAVE, PASTA [35]	Focus on risk assessment frameworks; limited support for scenario generation using structured, rule-based models.	Scenario Generation Mechanism (Challenge 2), RQ1, RQ2
Attack Graph Approaches [29]	Limited focus on reducing execution complexity or time when generating multiple scenarios.	Time Complexity (Challenge 3), RQ4
Hoffman's Survey [32]	High-level theoretical modeling; lack of practical host-level mapping or real-time scenario construction.	Lack of Host-Level Mapping (Challenge 4), RQ4, RQ6
Automated Planning Approaches [37]	Weak link between plan generation and system-state-aware acting; lack of dynamic adjustment.	RQ4, RQ6

In summary, while existing works provide the foundation, they fall short in dynamically generating realistic, adaptive attack scenarios using real-time system state, synthesized data, and optimization techniques. This thesis fills these gaps by proposing a structured, ML-enhanced, and computationally efficient methodology, as detailed in Chapters 3 to 6.

2.7 Chapter Discussion

In this chapter, an in-depth analysis of the ICS is presented. Also, the chapter aimed to develop a background knowledge to better comprehend the research work. The ICS is defined in great detail and the different dimension of the ICS in terms of its heterogeneous nature is

elaborated. We also looked into the previous attacks, and how they were generated. In the next two chapters, the ASG with respect to manual land threat-based will be discussed.

2.8 Conclusion

This chapter highlighted the security aspects of ICS from the heterogeneity perspective. ICS heterogeneity has been described in three sections: heterogeneity in communication, architecture, and data, which leads to vulnerability. This may lead to system attacks, and that is what has been discussed thoroughly through listing previous cyber incidents in Industrial Control Systems. Finally, it was crucial to identify the previous efforts exerted to analyze attack scenarios, either manually or automatically, to find the gap and build upon it to contribute to the ICS attack scenario analysis and generation.

Chapter 3

Manual Generation of Attack Scenarios

3.1 Introduction

In this chapter, manual attack scenario generation is discussed with the help of real-life examples and attacks that have caused millions of dollars in damages to large companies, such as Saudi Aramco. A variety of attacks on different countries are highlighted. Furthermore, the methodology for manually generating the attack scenario involves mapping incidents using the ICS cyber kill chain to create an attack scenario. Destructive Wiper Malware (DWM) is exemplified in this chapter as a type of attack that targets vulnerabilities in ICS infrastructures, aiming to destroy and exploit them to wipe out and damage data. This assists in analysing the manually generated attack scenarios and concludes how they are related to DWM. As a result of this methodology, a general attack scenario for DWM is represented.

DWM is a type of attack that targets vulnerabilities in ICS infrastructures, aiming to destroy and exploit them to wipe out and damage the data. The first evidence of this kind of malware was Stuxnet [38, 39]. Stuxnet was notably recognised due to its attack on Iran's nuclear facilities, and since then, it has evolved. After that, more such attacks targeted multiple facilities, including Aramco, the largest oil company in Saudi Arabia. Saudi Aramco is one of the largest companies in the world, dealing in Saudi Arabian public petroleum and natural gas, and is based in Dhahran. Following the Stuxnet cyber attack, a series of subsequent attacks continued through the Shamoon malware [40] on Aramco, as documented in 2012. Later on, the predecessor of Shamoon, known as Shamoon2 [41], was used to attack Aramco. In 2016, again, an attack was carried out using Shamoon3 [41], and in 2017, the Triton [42] and StoneDrill 2017 [43] worms were used to attack Aramco. Continuing the attacks, ZeroCleare [44] was reported to be used to attack Aramco in 2019.

Analogous to Shamoon, a new virus, ZeroCleare, was discovered. According to the IBM X-Force Incident Response and Intelligence Services (IRIS) [45], ZeroCleare is a destructive

attack that bears similarity to the Shamoon attacks. ZeroCleare's similarity to Shamoon is observed through its purpose of overwriting the master boot records (MBRs) and disk partitions on Windows-based machines [45].

More research and studies are needed in this area; therefore, the aim of this chapter is to thoroughly document the cyberattacks against various companies, particularly Saudi Aramco. Furthermore, the ZeroCleare attack, which has not been documented in the literature, is also discussed. Documenting attacks is a nontrivial process, as it is essential to identify the key vulnerabilities and attack steps that must occur for an attack to be successful. To this end, in this thesis, the ICS kill chain was utilised to develop a bespoke DWM taxonomy that can encompass each step of the attacks discussed thus far. Through the kill chain, it becomes much easier for a security professional to ascertain whether their system is secure against a set of attacks and to evaluate the coverage of security guidelines. An overview of the kill chain is provided in Section 3.6, and it also serves as the basis for the manual attack scenario generation process.

The focus of this chapter is on the methodology for creating an attack scenario manually by mapping it to the constructed kill chain. Furthermore, it defines a detailed taxonomy of destructive malware attacks. This chapter highlights each phase's step within the context of the ICS kill chain [1].

3.2 Related Works

The ICS kill chain is a cybersecurity framework that involves the various phases attackers follow to compromise critical infrastructure [46]. If analysts follow these phases, potential vulnerabilities can be understood, and mitigation strategies at each stage to disrupt the attack can be followed [47]. This framework builds upon the traditionally known cyber kill chain model, incorporating additional phases that are specific to the uncommon characteristics of ICS environments [47].

3.2.1 The Diamond Model

The Diamond Model of Intrusion Analysis, introduced by Caltagirone et al.[48] in 2013, is a framework that analyses cyberattacks through the key components of adversary, capability, infrastructure, and victim. This model provides a comprehensive view of attacks, aiding in the development of effective threat identification strategies. However, the model's simplicity can be a disadvantage, as it potentially overlooks the complexities of advanced cyberattacks.

3.2.2 The OWASP Threat Model

The OWASP Threat Dragon (OWASP Threat Dragon) is a threat modelling framework used in cybersecurity to systematically label and mitigate vulnerabilities in software applications (OWASP Threat Modeling Project). This approach offers advantages such as early detection of security risks during the Secure Development Lifecycle (SDLC) (OWASP Threat Modeling Process), leading to cost-effective remediation. However, it can be time-consuming and requires specialized knowledge, potentially hindering its adoption by smaller development teams[49].

3.2.3 Attack Graph

An attack graph in cybersecurity visually depicts the potential pathways an attacker can exploit to breach a system, ultimately achieving a desired state, such as gaining access to critical data. This visualisation offers valuable insights for security professionals, but it can be challenging to construct and maintain manually [50].

In the analysis conducted to develop a taxonomy of Drive-by-Wire Malware (DWM), attack scenarios were manually constructed by mapping attack steps, documented in laboratory reports, to the ICS cyber kill chain. For each incident, a distinct scenario was created. By synthesizing insights from these three individual attack scenarios, a generalized scenario for DWM was derived.

3.3 Destructive Wiper Malware Attacks

A DWM aims to disable access to data or destroy the system functionality. It has evolved from a simple novel concept to an epidemic targeting large companies across multiple sectors, which as documented in [51], has shown a 200% increase in the first half of 2019 when compared to the second half of 2018. As discussed in [52], DWMs have the following four characteristics that attract attackers:

1. Easy to implement.
2. Fast to execute.
3. Fast to spread across the network.
4. Almost invisible until it is too late. which means the system has already been overwritten.

As previously discussed, DWMs are designed to overwrite and wipe files essential for the system to run by overwriting the master boot record (MBR), rendering it inoperable. In the case of Shamoon and Zeroclear, the attacker used the EldoS RawDisk driver to wipe the victim's hard drive. RawDisk, which is signed by EldoS, is a legitimate tool that grants direct access to the internals of operating systems. The attacker manipulates this strategy and replaces the existing EldoS RawDisk driver with a vulnerable driver (DRDISK in the ZeroClear attack case), which allows them to bypass Windows security restrictions, and the attacker ultimately gains access to both read and write rights on the hard drive. While in StoneDrill, wiper modules are injected into the user's browser preferences, allowing for a watering hole attack that enables the system to crash and delete the MBR. In a few cases, such as Stuxnet and BlackEnergy, they are multifunctional wipers that aim to destroy industrial equipment by sending random messages to the engines, causing them to break down and become inoperable. In some cases, the attacks have been primarily used by nation-state actors to deliver political messages. Notable incidents included Stuxnet, Shamoon, BlackEnergy, and DarkSeoul. These are more desirable and popular choices for advanced persistent threats (APTs), a term used to describe a sophisticated threat to computer systems, where the threat actor is typically a nation-state group. These attacks enable nation-states to send powerful messages and destroy their targets, with the ability to plausibly deny the claims.

As shown in Fig. 3.1, the majority of the attacks are carried out by nation-state groups. However, more recently, DWMs, such as LockerGoga [53] and MegaCortex [53], have been used by cyber-criminals to target commercial entities.

This chapter focuses on the attacks that are considered destructive wiper malware for two reasons. Firstly, the study shows that destructive attacks have occurred frequently over the past few years and are still ongoing. Secondly, the study aims to narrow down the analysis of cyber incident responses. The cyber incident response in operational technology (OT) differs slightly in terms of practices and procedures from that in information technology (IT). Therefore, this study will exclude the Triton (TRISIS) attack[54], which was a deadly goal attack that targeted Aramco's industrial control safety system.

3.4 Review of Destructive Malware Attacks

In this section, the attacks will be described and also the general taxonomy of a DWM attack will be modelled using the ICS cyber kill chain.

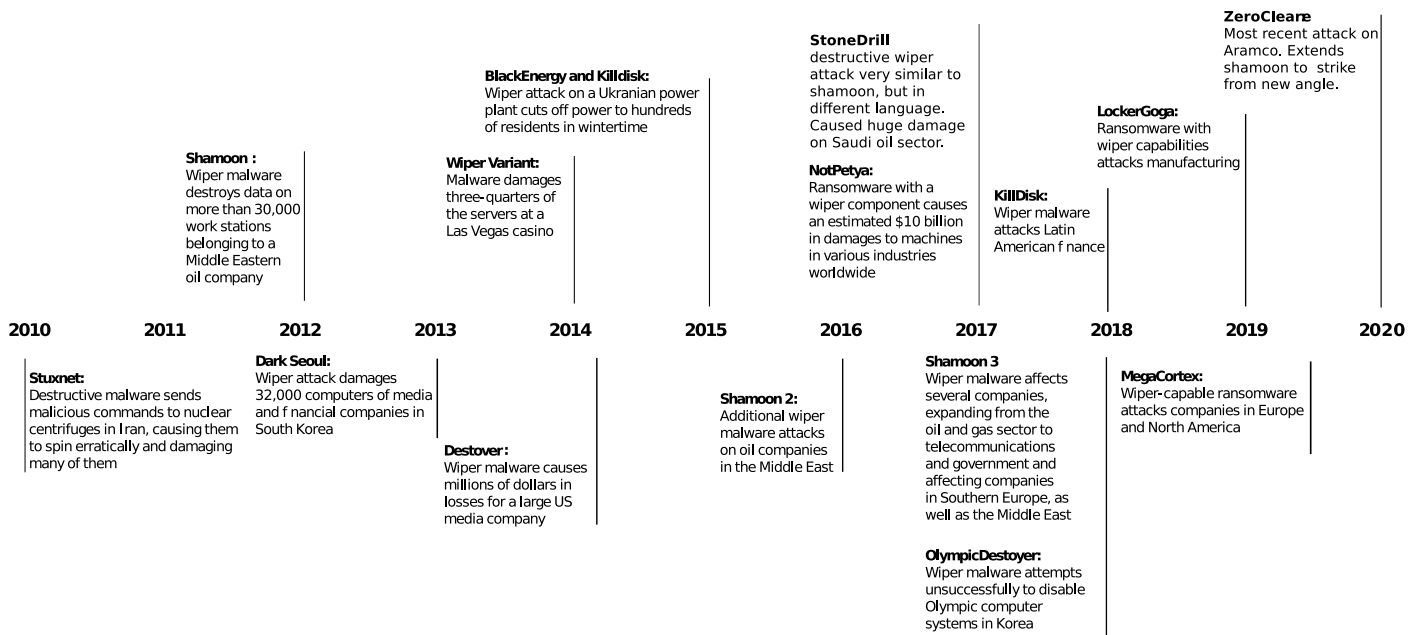


Fig. 3.1 Timeline of Destructive Malware Cyber Attacks Adapted from [51], to Include Undocumented Attacks.

3.4.1 Aramco Attacks

Aramco has been the target of a series of destructive malware attacks over the past few years, resulting in significant damage to its infrastructure and finances. These attacks were mainly caused by Shamoon, Shamoon 2, Shamoon 3, StoneDrill, and ZeroCleare. The continuous successful targeting with relatively slight variations triggers the need for a comprehensive incident response plan to avoid such kinds of attacks on any organisation.

- Shamoon 1 Attack

Shamoon (also known as Disttrack) is a destructive wiper malware used in a targeted attack against the energy sector in the Middle East, including Aramco. In 2012, Shamoon destroyed more than 30,000 workstations, and the company took over two months to fully recover from the attack's consequences. The most interesting component in Shamoon was the wiper, which was responsible for wiping all files on the hard drive, the MBR, and active disk partitions. The attacker specifically chose a holiday to give more time for the malware to spread before discovery.

- Shamoon 2 and 3 Attacks

The Shamoon 2 attack was carried out at three different times and targeted five government organisations: Saudi Aramco, Saudi Arabian Monetary Authority (SAMA), Saudi Ministry of Labor and Social Development, Saudi National Industrial Cluster Development Program, and Saudi General Authority of Civil Aviation (GACA). The first attack was carried out on the 17th of November 2016, followed by another attack on the 29th of November 2016, and the last attack took place on the 23rd of January 2017 [41]. In 2018, Shamoon 3, which was almost identical to Shamoon 2, was discovered. In total, the damages cost the government 2.8 billion Saudi Riyals to recover from and overcome the consequences [55]. All of the Shamoons variations' main components were the dropper, wiper, and reporter. However, apart from Shamoon 1, Shamoon 2 and Shamoon 3 have fully functional ransomware that encrypts the entire disk and spreads over the network automatically, without the need to communicate with the control centre.

- StoneDrill Attack

While Kaspersky Lab experts were investigating Shamoon 2, they discovered a new wiper called StoneDrill [43]. StoneDrill is a destructive wiper attack, but it has structural components that differ from Shamoon. Unlike Shamoon, it does not use a driver for deployment but relies on memory injection into the victim's browser. Shamoon 2 and Stonedrill were two distinct waves of attacks that occurred within the same timeframe between 2016 and 2017. Stonedrill also shares similarities to another attack called Newsbeef [56], conducted on a European company in 2014.

- ZeroCleare Attack

The flow of ZeroCleare in a step-wise manner is provided in the Figure 3.2. ZeroCleare used the EldoS RawDisk driver to wipe the victim's hard drive. This driver is a legitimate tool that provides direct access to the internals of operating systems, allowing ZeroCleare attackers to bypass Windows security restrictions. ZeroCleare bypasses Windows controller driver signature enforcement by utilising the Turla Driver Loader (TDL) open-access tool and installs a compromised driver. Afterwards, EldoS is used to issue a license key that can activate the disk management driver. Finally, ZeroCleare executes the wiping function, which creates a buffer of random bytes and wipes the victim's hard drive by overwriting MBR, partitions, and files on the system with junk data. ZeroCleare has 32-bit and 64-bit versions. In the 32-bit version, driver signature enforcement is disabled, making it easier to implant. However, the ZeroCleare 32-bit version crashed while trying to execute the wiping

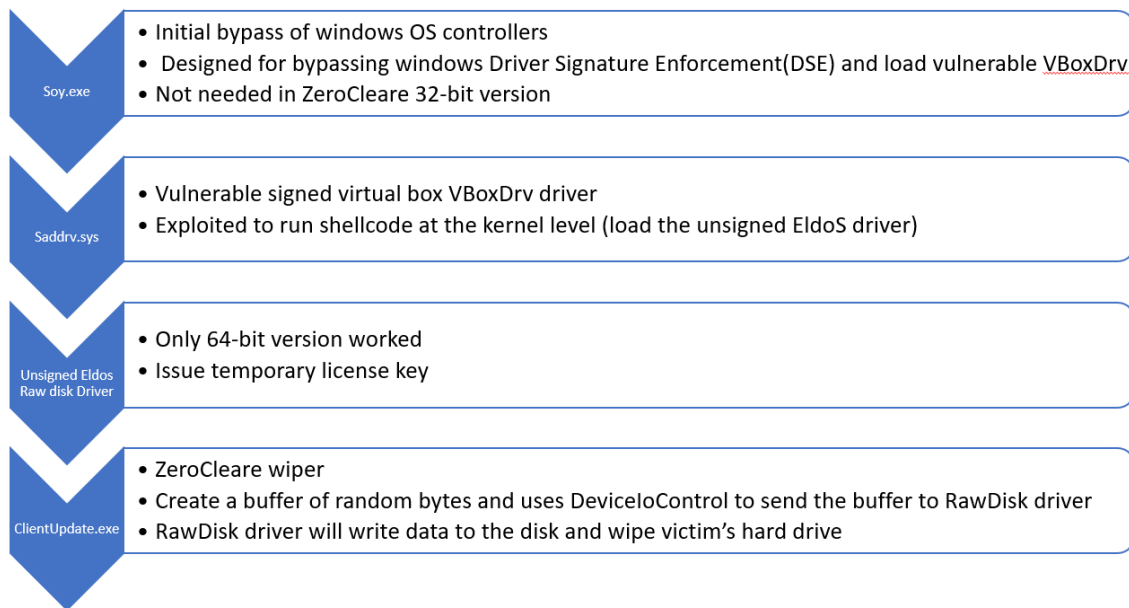


Fig. 3.2 Declare Infection Flow

functions. The 32-bit version of ZeroCleare crashed primarily due to driver compatibility issues, insufficient error handling, resource limitations inherent to 32-bit systems, and possible oversight in testing and development. These factors combined to create an unstable execution environment, resulting in frequent crashes during operation. As a result, only the 64-bit version worked. ZeroCleare targets strategic devices and causes real-time economic harm, whereas Shamoon spreads randomly throughout the network, attempting to affect as many devices as possible. However, indirect costs related to leaked films, lost intellectual property, and reputation damage are difficult to quantify.

3.4.2 South Korea Attack

At 2 p.m. on March 20, 2013, South Korea suffered a major cyberattack that caused a denial-of-service attack across several major banks and broadcasters [57]. The attack, usually referred to as *Dark Seoul*, generated lots of attention because it was very effective, had low sophistication [57, 58], and was extremely precise in time [59]. According to the Korean Internet Security Agency (KISA), approximately 48,000 computers were affected, and organisations required weeks to restore functionality [57]. The attack caused about US\$750 million in damages [60]. Some analysts suggest that the attack could have been prevented if the affected organisations had just performed software updates and used antivirus properly [57]. *Dark Seoul* prevented affected computers to start by overwriting the MBR [59].

3.4.3 Sony Pictures Entertainment (SPE) Attack

In 2014, Sony Pictures was also targeted with destructive malware, including Shamoon and ZeroClear. The attacker aimed to wipe all the company servers using the RawDisk driver to overwrite the MBR. Approximately 100 terabytes of internal data, comprising 38 million files and thousands of personal emails, were released to the public [61]. Sony estimated that the direct costs of the attack were over US\$41 million [62]. The Sony Pictures attack generated widespread debate and publicity, not only because of the magnitude of the attack, but also due to its political nature [61] and the controversial technical issues that revolved around such a reputable international company. The official Sony response, created by the Mandia Security firm [62], alleged that the attack was an "unparalleled crime" carried out by an "organised group," and neither SPE nor other companies could have been fully prepared [63] for the attack.

However, some analysts contradicted this response, alleging an insufficiently small security team comprising three analysts overseen by eight managers in the hierarchy, a flat network architecture having no internal firewalls or barriers to provide resistance, and a scarcity of password-protected or internally encrypted files [63].

According to post-technical analysis, attackers knew details about the network [61] and some speculation has led to the belief that they possibly exploited the Server Message Block (SMB) protocol[64], extracted data and installed wipers[65] that rendered computers unusable after a planned reboot[66].

In the next section, a well-formulated method for creating attack scenarios and its steps are elaborated.

3.5 Method

To create an attack scenario, the following steps are followed. The associated documents are reviewed with previous cyber incidents, and the techniques utilised by the attackers in each occurrence are extracted. This involved analysing technical lab documentation for all the incidents to comprehend the methods employed by the attackers, such as phishing or other techniques. These techniques were combined to create an attack scenario using the Cyber Kill Chain framework.

The Cyber Kill Chain framework not only provides information on attack sequence techniques but also helps in understanding the tactics used. Each technique is linked to a specific tactic, outlining the steps taken by the attackers to penetrate the system and execute their attack. Using this framework, scenarios were proposed for the Shamoon, Stone Drill, and ZeroClear cyber incidents. These three incidents were analysed to identify the

techniques used and create a general taxonomy for "Destructive Wiper Malware." This malware is designed to wipe and destroy important corporate data, resulting in significant political and economic damage.

By following the steps mentioned above, a well-formulated attack scenario can be created for the three discussed incidents.

3.6 The ICS Cyber Kill Chain

The current approach first creates a cyber kill chain for each attack and then combines them to create a general *DWM kill chain*, which is documented in Table 6.5. The ICS cyber kill chain is an applied kill chain specially designed to keep the ICS context [1]. A cyber kill chain (CKC) details the steps that an adversary will take to successfully attack their desired target. In the context of ICS, specific phases should be in place to facilitate the initial successful breach and subsequent attacks. Figure 3.3 provides a graphical illustration of the process.

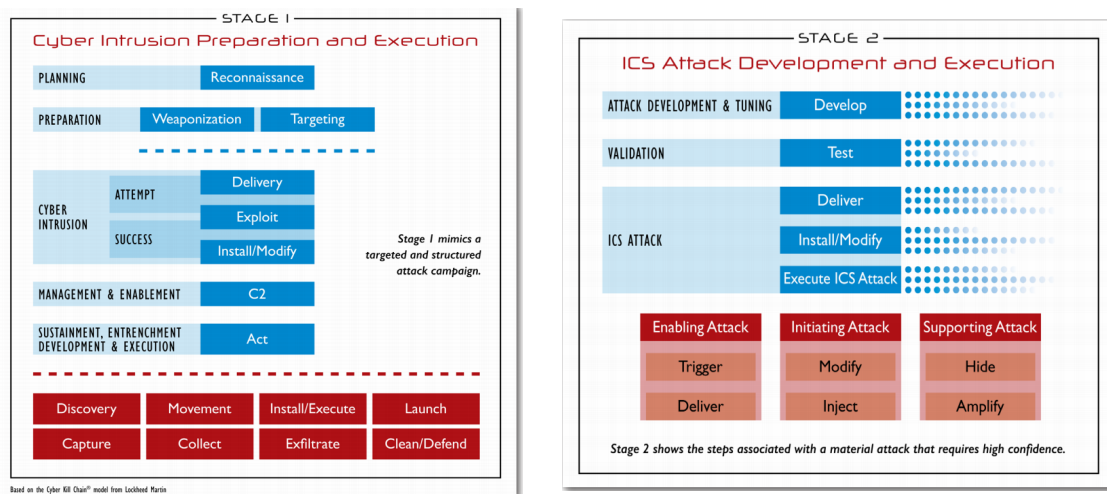


Fig. 3.3 The ICS Cyber Kill Chain by M. Assante and R. Lee [1]

The attacks on Aramco have affected both the IT and the ICS domains of the organisation. This research adapted the ICS cyber kill chain to investigate the steps attackers followed to perform a high-confidence destructive attack that successfully affected the IT and aimed to affect the ICS [12] in Aramco.

The analysis focused on the Shamoon attack and is detailed in stages, along with steps in each stage. The whole process is summarised in the Table 3.1.

The first stage is **Planning**, which includes reconnaissance strategy. There is clear evidence that the attacker was observing the system: the timing of the attack coincided with a holiday, and the phishing email was sent to one of the IT members in Aramco, understanding the vulnerability in the Windows workstation [67].

The second step is **Preparation**, which further splits into Weaponization and Targeting. In Shamoon, the attacker used a Microsoft Word document as a weapon to embed the malware, combined with a PowerShell script to target the victim's device.

During the **Cyber Intrusion** phase, consisting of Delivery, Exploit, and Install/Modify, the malware was delivered using an email sent to a member of IT [67]. When the individual opened the document, the executable file with encrypted resources exploit was executed [68]. The file consisted of three components: Dropper, Wiper, and Reporter. The dropper is responsible for installing the malware and related components.

During the **Management and Enablement** phase, consisting of Command & Control (C&C), the attacker used basic C&C to spread the attack.

In the **Sustainment, Entrenchment, Development & Execution** phase, the intruder took over devices and destroyed them and gained access to specific system functionality through an installation of an unauthenticated driver (i.e., Eldos RawDisk).

In stage two, during the **Attack Development** phase, the attacker launched the services inside the network looking for workstations. Once the target was found in the **Validation** phase, the malware opened and closed random files many times to check and validate whether it had gained access and whether it was an ICS controller or not. Next, in the **Attack** phase, it executed itself and spread into other systems through the network. At this point, the malware checks whether the system is not related to ICS, then it simply wipes its data and moves on to other systems. The attack succeeded through the IT environment, wiping all the files, MBR, and active disk partitions in the targeted workstation; however, it did not spread into the ICS. Assessing guidelines for destructive malware have been detailed in Appendix B

3.7 Chapter Discussion

In this chapter, a new cyber incident (Zero Cleare) was manually analysed, keeping in mind a previous cyber incident (Shamoon). Analysing cyber incidents using attack scenarios is a process for understanding, responding to, and mitigating cyber threats. Security analysts primarily rely on their expertise to create attack scenarios, which enables them to gain a deeper contextual understanding of an incident. However, there is no standardisation for this practice. In this chapter, the main focus was to introduce a methodology that can serve as a base for manual attack scenario generation by using a new undocumented incident (Zero

Cleare) in the presented analysis. After gathering data about the ZeroCleare incident and extracting its attack techniques, it was compared with the previous well-known cyber incident (Shamoon). For the sake of comparison between Shamoon and Zero Cleare, Shamoon attack techniques were extracted and mapped to both incidents using the ICS cyber kill chain framework. As a result, the full sequence of attack techniques was detailed, which ultimately led to the attack scenarios for both Zero Cleare and Shamoon. This process helps in understanding the commonality between the two incidents and also assists in carrying out a deeper analysis and generalising a taxonomy for destructive wiper malware.

Another incident that was added in the process was StoneDrill, to validate the work. Furthermore, a timeline and a description giving examples from real previous ICS cyber incidents were provided, which can be used as literature for destructive wiper malware. This chapter is considered a roadmap for the manual generation of attack scenarios. However, the process of manual generation has two major issues: it is time-consuming and relies on the assessor's knowledge. Therefore, automation is required to overcome these issues and limitations. Automating the process and providing expert knowledge can help individuals and organisations create attack scenarios and draw conclusions related to cyber incidents. In the following two chapters, the automation of attack scenario generation depending on threat intelligence and system state will be discussed.

3.8 Conclusion

Destructive malware poses an increasing threat that has a huge impact on governments and organisations alike. Despite existing guidelines, these attacks have been carried out successfully for several years, with no signs of slowing down. For this reason, the researcher reviewed and analysed all DWM attacks in the last 10 years, including the previously undocumented ZeroCleare, and developed a new way of assessing guidelines through the use of bespoke cyber kill chains. The methodology used to create the DWM taxonomy is manual, involving the creation of attack scenarios with specific parameters. Therefore, this study can be considered the core reference for researchers aiming to develop an automatic model that predicts upcoming cyberattacks based on previously provided data, which is the major objective of the researcher's current research.

Table 3.1 ICS Cyber Kill Chain Steps Mapped to Shamoon Attack

Attack Steps	Shamoon Attack
	<i>Stage 1</i>
<i>Reconnaissance</i>	attack was on holiday; phishing email sent to IT member; knew of vulnerability in the equipment;
<i>Weaponization</i>	MS word document to embed the malware;
<i>Targeting</i>	control of targeted IT staff computer;
<i>Delivery</i>	successful email phishing;
<i>Exploit</i>	hidden executable file.
<i>Install/Modify</i>	took over workstation using vulnerable driver
<i>Command and Control</i>	operation point of the attack;
<i>Act</i>	wiped the disk; disabled devices;
	<i>Stage 2</i>
<i>Develop</i>	scouted for further vulnerabilities;
<i>Test</i>	tested files to check behaviour;
<i>Deliver</i>	searched system for ICS controller;
<i>Install/Modify</i>	wiped all machines on the way;
<i>Execute</i>	attempted ICS controller takeover;

Table 3.2 ICS Cyber Kill Chain Mapped to Destructive Wiper Malwares

Attack Steps	General DWM	Shamoon	ZeroCleare	StoneDrill
Stage 1				
<i>Reconnaissance</i>	workstation vulnerability	IT member Email [67] workstation vulnerability [68] timing of the attack [69]	workstation vulnerability [45]	workstation vulnerability [43]
<i>Weaponization</i>	compound document; watering Hole;	compound document[67]	compound document[45]	Watering Hole [43]
<i>Targeting</i>	bypass security controls	bypass user account controls[70]	bypass user account controls; shellcode in kernel:[45]	sandbox evasion [71]
<i>Delivery</i>	phishing; process memory injection;	phishing [69] [67]	phishing[45]	process memory injection [72]
<i>Exploit</i>	hidden executable	hidden executable [73]	hidden executable [45]	hidden executable [72]
<i>Install/Modify</i>	fake rawdisk driver	fake rawdisk driver; bypass signature; [68]	fake rawdisk driver; bypass signature; [45]	fake rawdisk driver; bypass signature; [43]
<i>C&C</i>	remote system control	application layer (http); remote system discovery; remote file copy:[73]	common tcp port; file deletion; remote file copy; [45]	windows instrumentation; remote system discovery; remote file copy:[72]
<i>Act</i>	data destruction; disk structure wipe;	data destruction; disk structure wipe:[12]	data destruction; disk structure wipe; [45]	data destruction; disk structure wipe; [71]
Stage 2				
<i>Develop</i>	new service; scripting;	new service [73]	unknown	scripting [72]
<i>Test</i>	service execution; execute script;	service execution [73]	unknown	execute script [74]
<i>Deliver</i>	windows admin share; network config exploration; valid accounts;	windows admin share; network config exploration; [73]	network config exploration; [45]	network config exploration; valid accounts; [72]
<i>Install/Modify</i>	scheduled ICS tasks	scheduled ICS tasks [75]	unknown	script in workstation [74]
<i>Execute</i>	takeover ICS controller	attempt to takeover SCADA [75]	unknown	failed [74]

Chapter 4

Attack scenario Generation: Threat-based

4.1 Introduction

Cyberattacks are becoming more frequent, sophisticated, and intelligent. With more robust security measures in place, attackers are also adapting their technologies and tactics to circumvent the new security systems. Cyber threat intelligence (CTI) analyses previous attacks and reconstructs them to understand the tools, patterns, and techniques used by attackers in the past. This strategy provides analysts with a more effective way to study the attacks. CTI can help analysts understand the rationale, goals, and methods of relevant attacks and their perpetrators. Following the CTI protocols, this chapter presents attack scenarios created based on cyber threat intelligence.

As discussed in the previous chapter, the process of attack scenario generation needs to be automated, as it offers various advantages that empower security professionals to effectively identify vulnerabilities, evaluate defence processes, and respond to potential threats quickly. The automation process could also save time and effort. To automate attack scenario generation, artificial intelligence and machine learning technologies can play a vital role in enhancing cybersecurity. AI has been used to improve threat detection capabilities, prevent attacks, recognise the probable behaviours of hackers, analyse user behaviour, and develop self-diagnosis and automatic response systems for attacks [76]. It is therefore considered to automate attack scenario generation using the latest technologies such as AI and ML.

4.2 Related Works

Due to the continually changing and evolving cyber threat landscape, traditional methods, which involve physical control and manual methods of discovering attack scenarios, are considered less safe. CTI, which is shared throughout organisations, helps with strategic decision-making and accelerates the defence line against new threats. However, the necessity to transfer the acquired CTI into operational steps is one of the most crucial problems. The majority of traditional AI methods ignore the information gained through CTI sharing. This chapter demonstrates the use of CTI for security operations.

Because CTI is evidence-based threat intelligence that aids decision-making, it bridges the gap between human understanding and attack prediction of cyber threats to enterprises[77]. Therefore, CTI enables enterprises to respond more effectively to ever-increasing cyber threats. Organisations use CTI to log suspicious or malicious activity, share information between different organisations, and automate the process of detecting and investigating attacks against them. According to [78], there are four types of CTI: operational threat intelligence, strategic threat intelligence, tactical threat intelligence, and technical threat intelligence. Operational threat intelligence is primarily concerned with real-time threats, facilitating immediate incident response and threat-hunting activities through detailed analysis of security logs and network traffic. This type of intelligence is critical for the prompt identification and mitigation of active threats. Strategic threat intelligence offers a broader, long-term perspective, informing the development of overarching security strategies and policies by analysing geopolitical trends, technological advancements, and industry developments. Tactical threat intelligence focuses on the specifics of adversary tactics, techniques, and procedures (TTPs), thereby aiding in the refinement of defensive measures and enhancing security team training. Lastly, technical threat intelligence provides granular data, such as IP addresses, domain names, and file hashes, which can be used to automate threat detection and enable real-time protection through integration with security tools. By leveraging these diverse forms of threat intelligence, organisations can achieve a comprehensive understanding of the threat landscape, enabling them to implement robust and adaptive security measures that address both immediate and long-term challenges. Tactical indicators are behaviour-oriented tools, tactics, and techniques associated with a pattern of activities or methods used by a specific threat actor or group of threat actors[79]. The hierarchy of tactical and technical metrics is represented by the Pyramid of Pain (PoP) as described in the article[80]. Technical signs, such as IP addresses, hash values, domain names, and network/host artefacts, are found at the bottom tiers of the PoP. These indications have correct numbers, but the attacker can readily change them. For example, an attacker can easily change IP addresses and file hashes with little effort. Due to the availability of free DNS providers and lax registration

standards, even domain names can be easily altered. The attackers can use anonymous proxy services, such as Tor, to change IP addresses, flip bits on unused resources, or add spaces. Recognising the high level of attack footprint is more challenging due to complex patterns and the resemblance to normal behaviour. Modelling attacker behaviour is essential to better understand attack methods and improve detection and analysis capabilities.

Apart from the related discussion, Obst et al. [81] mentioned Structured Threat Intelligence Expression (STIX) as a comprehensive CTI framework, which encompasses various cyber ontology elements, including adversaries, tactics, techniques, and procedures (TTPs), courses of action, indicators, incidents, and vulnerabilities. The STIX element is called a STIX Domain Object (SDO). MITER ATT&CK uses different terminology from STIX. For example, *techniques* in ATT&CK are called *attack patterns* in STIX. Similarly, *groups* in ATT&CK may refer to *campaigns*, *intrusion sets*, and *threat actors* in STIX. The immediate need for using CTI is the analyst's ability to search and view the collected CTI and explore it by enabling search with graphics from STIX, which provides a visual representation of elements and charts.

In this study [82], security analysts claim that the proposed CTI visualisation helped the investigation process by building a platform that integrates CTI with their innovative search-enabled searching on a variety of nodes' attributes and multiple sets of nodes (subgraphs).

Attack reconstruction, which is often the output of a successful threat-hunting procedure, refers to describing a threat by presenting the different steps the attacker successfully executed. The security analyst should have the ability to explain how each step was achieved by pointing to the relevant events based on the collected evidence and their analysis. Milajerdi et al. [83] presented a framework for real-time APT detection by analysing kernel audit logs. The framework produces a high-level scenario graph representation with associations between TTPs for each detected APT. Bhatt et al. [84] also suggested an APT detection model. The suggested model is used to improve hypothesising ongoing attacks given correlated events and the cyber kill chain. Rubinshtein et al. [5] created an attack ontology and reconstructed attack phases based on this ontology.

Polatidis et al. [85] proposed an approach for a cyberattack prediction using visualised attack graph representations and recommender systems; however, the modelling of an attacker's behaviour was based on exploiting vulnerabilities, regardless of the attacker's goals.

Kotenko and Saenko [86] proposed a cybersecurity ontology, which includes attack, malefactor, cost, topology, zeroday, and system metrics, along with environmental components. Following an attack detection, an analyst can implement countermeasures using the metrics and the relationships between metrics of the components in the attacked environment.

Giura et al. [5] proposed the idea of an attack pyramid that aims at capturing the movements of an attacker through CKC phases (represented by the levels of the pyramid) and an organisation's environments as represented by the planes of the pyramid (e.g., physical, network, user, and application). The attacker's goal is found at the top of the pyramid, and it can be reached by stepping from one event to another. The pyramid enables tracing the attack and the attacker's goal. Iqbal et al. [87] built a unified graph representation for CKC and PoP models by extracting entities from textual reports. Two variants of the same attack are compared.

Pinchinat et al. [88] introduce ATSyRA. ATSyRA is a tool that is applied on top of Eclipse to assist security experts in communicating and interacting with a user-friendly environment when designing Attack Trees. The major motivation for applying the tool was to secure military buildings. As a first step, the security expert defines the system, including a description of the building, the strength of the attacker, and the purpose behind the attack. The following step is to run the process of generating attack scenarios. This specific step compiles the input feature and specifications into an Attack Graph.

Another method for automatically generating attack trees is presented by Bryans et al. [89]. It takes place from the description of the network and a set of templates. Each one of the templates represents a step in the attack. The templates use variables that are replaced by the system's components under investigation. Kumar et al. [90] also presented the use of an Attack Tree template as a feature diagram, or as it is called, formal and graphical notations, to provide solutions for the non-standardisation problem related to Attack Trees.

4.3 Methodology

Creating an attack sequence through the analysis of the ICS ATT&CK knowledge base[91] is desirable as it was developed for adversary emulation and serves as a framework and knowledge base for attackers to identify tactics and techniques based on real incidents. ATT&CK summarises attacker behaviours in terms of tactics, techniques, and groups. It is essential to organise the attack methods used in cyberattacks and unify their names and scopes. The attack method names consist of a sequence of techniques to generate a scenario. To understand the ATT&CK knowledge base, the Spacy NLP model is employed, specifically designed to handle large amounts of text for data analysis and processing.

To process such a large amount of text, there is a need for initial processing steps on the dataset relevant to ICS techniques and incidents. With such a large amount of data, it is preferable to use Finite State Machines (FSM), which is a method based on automata theory for language generation. Through FSM, the names, descriptions of techniques, incidents,

groups, tools, and data sources are identified. Next, the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer model is employed to transform the text into meaningful numerical representations used for machine learning algorithm fitting. Subsequently, pipeline models are developed to teach the machine to predict ICS techniques, incidents, and groups based on textual inputs. Learning progress was monitored using learning curves that reveal the impact of adding more samples during training. Afterwards, testing was conducted on the test dataset to evaluate the classification model's performance, reflecting its ability to predict ICS attack scenarios. Confusion matrices were employed to verify accuracy and precision. External data were generated using three AI models for techniques, incidents, and ICS groups to assess the models' learning capability. After predicting the technique, incident, and group for the input text using prediction models, an attack sequence is developed to generate a scenario.

In this study, the attack sequence is handled through a tree model, based on the order of tactics in ATT&CK as a fundamental truth in the dataset for exploring attack sequences. The structure of the attack sequence is depicted in the Figure 5.1 part A. Attack methods are ordered according to a tactic execution order. It is assumed that techniques are interrelated in a hierarchical sequence to ensure accurate and incorrect scenarios. Next, the pathways are compared using various techniques based on similarity ratios or as an ICS incident. Finally, the attack scenario based on actual previous incidents in ICS is verified.

4.4 Data ICS ATT&Ck

The MITRE ATT&CK framework for ICS is an invaluable resource for understanding and mitigating cyber threats in operational technology (OT) environments. It provides a structured knowledge base of adversary tactics, techniques, software, and procedures (TTPs) specifically tailored for ICS attacks.

4.5 Data Preparation (preprocessing)

For machine learning techniques, large datasets are needed to create a module that can predict appropriate techniques and incidents, relying on the representation of inputs as metadata (text) provided by the user. Using different methods, data can be produced, which is the first step towards creating an attack scenario builder. To achieve this, it involves performing initial processing on the dataset related to techniques and ICS incidents. This includes removing unnecessary information, symbols, and numbers from text data. For the sake of data production, a well-known method is the Finite State Machine (FSM), which is based on

the automate theory and a method of machine learning to produce language. FSM is used to increase the dataset by identifying information that characterises techniques and to produce potential inputs.

In this research work, the MITRE ATT&CK of ICS is employed, consisting of 78 techniques and 19 incidents (softwares). This helps analyze the MITRE ATT&CK® and produce the appropriate dataset to describe the techniques and ICS incidents. Figure 4.1

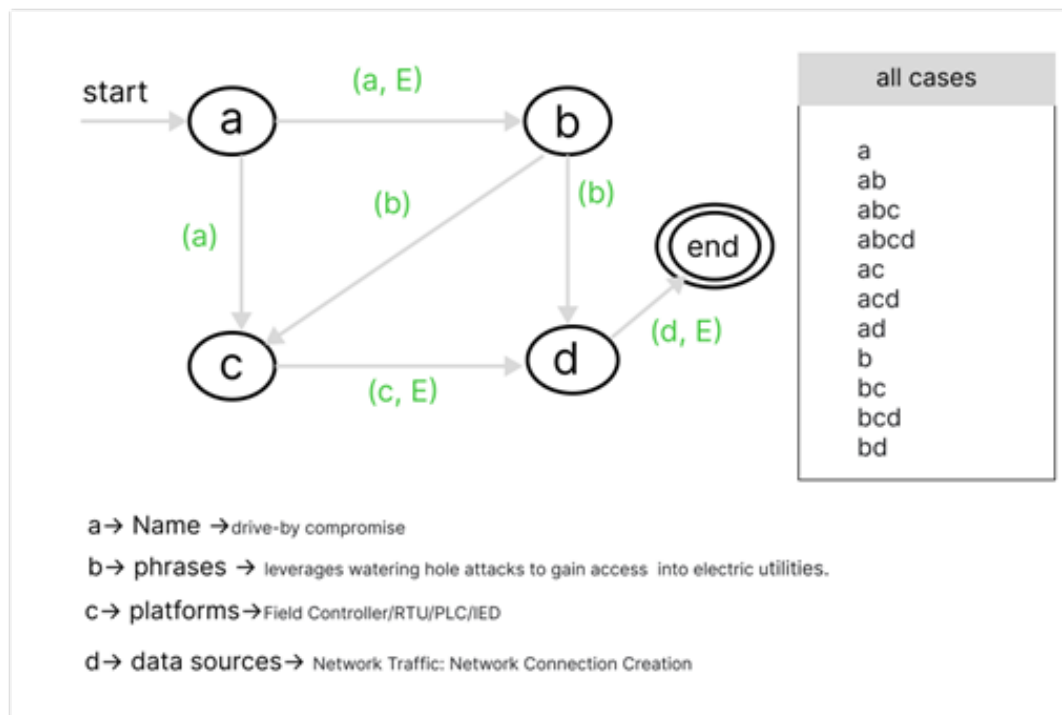


Fig. 4.1 Data Production Model

illustrates how to produce data from a MITRE ATT&CK, as it is a state machine that produces possible inputs represented by phrases or sentences extracted from the description of the technique and utilities. The machine works as follows:

- Entries start from node A, which indicates the name of the technique or incident.
- Status (E) represents empty input, which means that if the input does not include the name, you can move to node B, Where node B represents a description of the technique or incident.
- Node C means the utilities used with the attack. It is required to read the case from A to C, while it can take a null (empty) value from A to B.

- The data represented in this project are based on scientific constants and not random data.
- The input cannot be the C utility set or the data source D itself.
- The machine must read either the name or description with C and D so that the inputs are distinct to the techniques and incidents of ICS directly.

This stage, designed to predict user input, is a contribution that can be utilised in a cybersecurity application that relies on a MITRE attack. Applying such techniques increases the number of inputs from 78 cognitive descriptions of techniques to approximately 23,597 descriptions. For a group of 19 ICS incidents, approximately 2000 were produced that can be considered as inputs describing the MITRE ATT&CK of ICS. The results are provided in the Table 4.1.

Table 4.1 Inputs Describing the MITRE ATT&CK of ICS

Model	Data(n)	Data Train (n) 70%	Features(1*n)	Outputs
Techniques	23000	16099	1928	78 techniques
Incidents	2000	1400	1123	19 incidents
Groups	1152	0.76	656	15 groups

4.6 ICS Attack Description Analysis

Textual input to the model is either generated by the data production process or received by users. The input requires proper analysis and understanding to distinguish whether the text represents a technique or an ICS incident.

To resolve this issue, a free and open source library, SpaCy Model, is used [92]. It is an artificial intelligence model that handles natural language, capable of processing large sets of text. It is used to find the meaning of words in context, synonyms, and to perform semantic analysis of sentences. The use of NLP [93] as a semantic model facilitates the handling of large amounts of textual input and its comprehension. Furthermore, it allows the user to enter the text without restrictions, enabling users to analyse the text and extract significant words using the ICS attack, as well as identify the sources of these words, etc.

4.7 Features Extraction

In machine learning, pattern recognition, and natural language processing, feature extraction begins from an initial set of measured data and builds on derivative (features) intended to be informative and not redundant, thereby facilitating subsequent learning and generalisation steps. This stage aims to extract the expressive qualities of the ICS attack from the text data using two methods: the TF-IDF Vectorizer Model and similarity.

4.7.1 TF-IDF Vectorizer Model

TF-IDF Vectorizer [94] takes text data after dividing it into word sources using the NLP Spacy model, where it encodes each word or abbreviation into a unique number (See Figure 4.2). This is used to fit the machine for a prediction algorithm that can be defined as calculating how relevant is a word in a string or group to the text. The meaning appears more strongly proportionally to the number of times in the text that the word appears but is compensated by the frequency of the words in the dataset.

The application of (TF-IDF) on processed data produces 1928 distinct words for each input that have a technical significance after being converted into a digital vector that contains the repetitions of distinctive words. The same applies to the significant inputs of ICS incidents resulting in 1123 distinct words for each entry. It also applies to the group's data, where 678 distinct words were produced for each entry.

4.7.2 Similarity

Since the data describing techniques and attacks in ICS are textual, the cosine similarity algorithm was used [95] in word processing is a technique that aims to measure the degree of convergence or contrast between two or more texts in terms of meaning, shape or structure. The purpose of finding the similarity ratio between the sum of the attack data is to search for a common relationship between the techniques according to the description information and the tools they have in common. This method works by distinguishing the association of each technique with the next, which in turn helps in the process of producing the scenarios according to the resulting similarity ratio between the techniques.

4.8 Developing ICS Prediction Model

Defenders require knowledge of the first attack creator that describes the technique and the incident that defines the attack. However, it can be challenging to differentiate this

information with an abundance of techniques and incidents, particularly if the attacker employs various methods to carry out the attack. However, it is necessary to create multiple

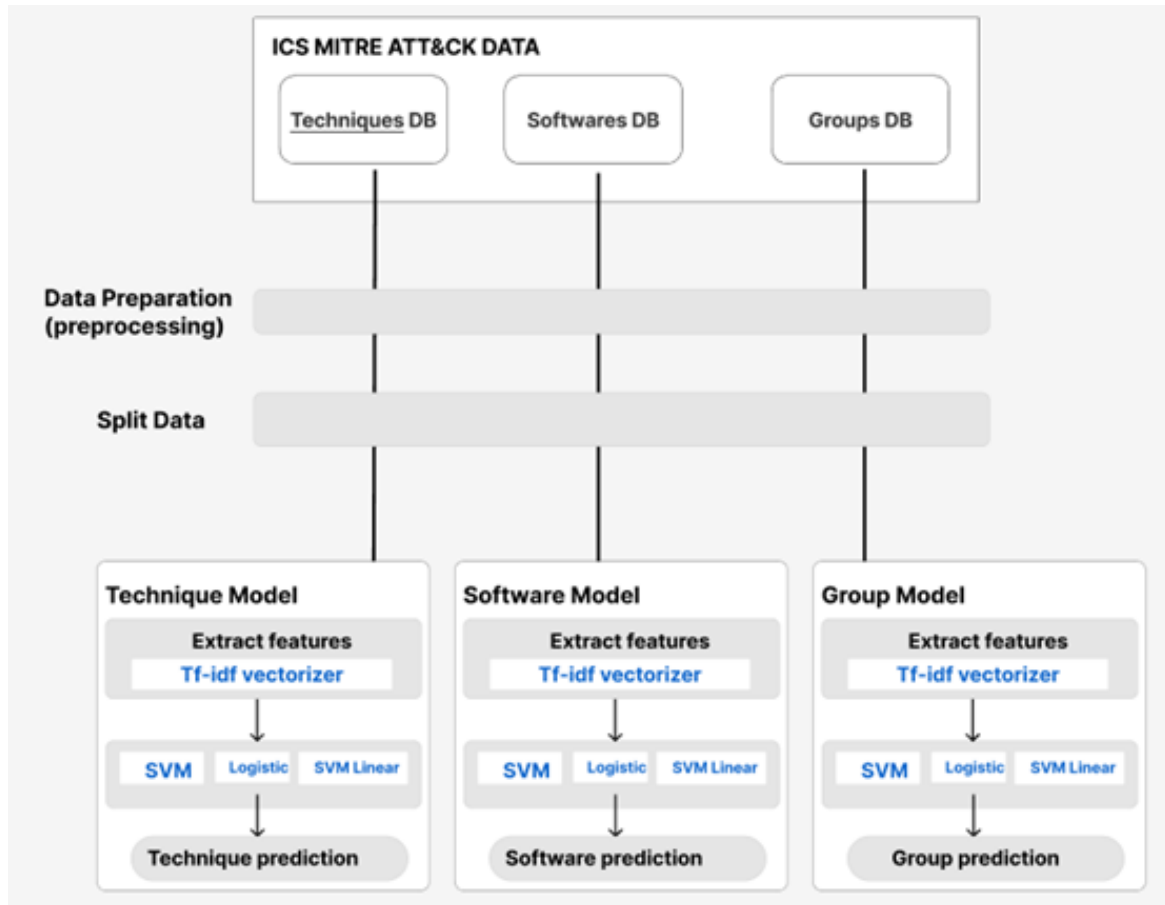


Fig. 4.2 Developing ICS Prediction Model

attack sequences from fragmented information related to a technique or incident. The problem is that it is difficult to create these sequences manually. Therefore, the ATT&CK knowledge base is processed and analyzed in order to examine the textual inputs that describe ICS incidents, extract their properties, use them with classification methods and finally train them on the processed data set.

A pipeline model of three different models is proposed to recognize ICS incidents that express textual inputs. Initially, a model is used to predict the technique that corresponds to the nature of the text input analyzed and has been provided with a set of information that describes techniques and utilities. The next model is the incident prediction model that corresponds to the input and technique, as this model depends on enhancing its knowledge of the technique expressing the input. The third model predicts the group to which the incident expressing the textual input belongs, predicting the technique, incident, and the

group described by the textual input. This is the main stage for generating attack scenarios. This dependency of models can be learned effectively using the application of halving models, i.e. a supervised approach to machine learning algorithms such as Linear, nonlinear SVM [96] and logistic regression algorithms [97] being used to confirm the correctness of the proposed classification. The most viable model of the learning algorithm for the current problem is Linear SVM.

4.9 Training Performance of ICS models

The tests were performed on the test datasets to detect the performance of classification models, which reflects the ability of the models to perform the prediction task as an ICS attack. Since the stage of data production and extraction of characteristics involved arranging and formatting the data so that each category has distinctive characteristics from the rest, all models showed good performance. Results indicate that the prediction model with the technique reached the greatest prediction accuracy with (SVC (Kernel=linear)) 96% and the average validity rate by 96%. The incident prediction model reached the greatest prediction accuracy with (SVC (Kernel = linear)) 99.2% with an average accuracy of 99% due to the fact that (SVC (Kernel=linear)) is especially useful when the data contains many features, or when there is a clear margin for separation in the data.

4.9.1 Technique Model

Figure 4.3 shows the technique prediction model that expresses the textual input. This begins with the processing of the input text and ensuring that synonyms and phrases are significant in techniques using the pre-trained NLP model. The words related to the techniques were analyzed and synonyms were replaced with a description that expresses the technique. This procedure helps to give the model flexibility and the ability to understand the textual input. After that, the properties were extracted to convert the data from textual descriptive information to digital information. The use of previous learning models to train the model contributes towards 70% of the processed data. The greatest concern is providing a large set of data describing the techniques, as the model may not deal with the input directly. Therefore, predicting good technique has a role in the origin of the attack scenario. It also has a role to get to know the incident and the group.

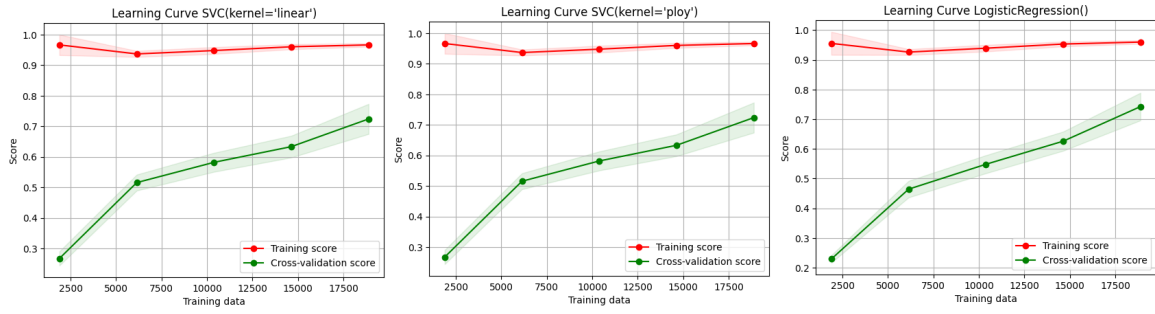


Fig. 4.3 Learning Curve Technique Prediction Model

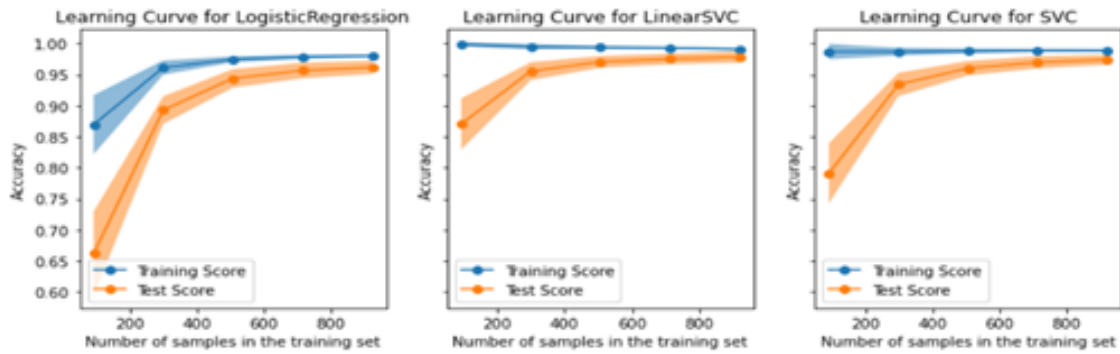


Fig. 4.4 Learning Curve Incident Prediction Model

4.9.2 ICS Incident Model

The architecture of this model does not differ from the model of predicting the technique, but it depends on its knowledge significantly, as the model enhances its knowledge through the technique expressing the input as shown in Figure 4.4 that shows the architecture of the model, where it integrates the textual input with the expected performance of the incident prediction ICS. During the stage of data production and processing, the model was trained on a group. This model works in two stages, the first includes predicting the nearest significant incident at the input and the second is using it in the scenario validation stage by finding the probability of the prediction ratio in ICS.

4.9.3 Group Pipeline Model

Figure 4.5 illustrates the architecture of the incident prediction model, it is developed as a pipeline model based on the knowledge of techniques and incidents. The dataset here is the same as the dataset that describes techniques and incidents and the group information. The model is trained on a group and to predict the group, it is required at the beginning to

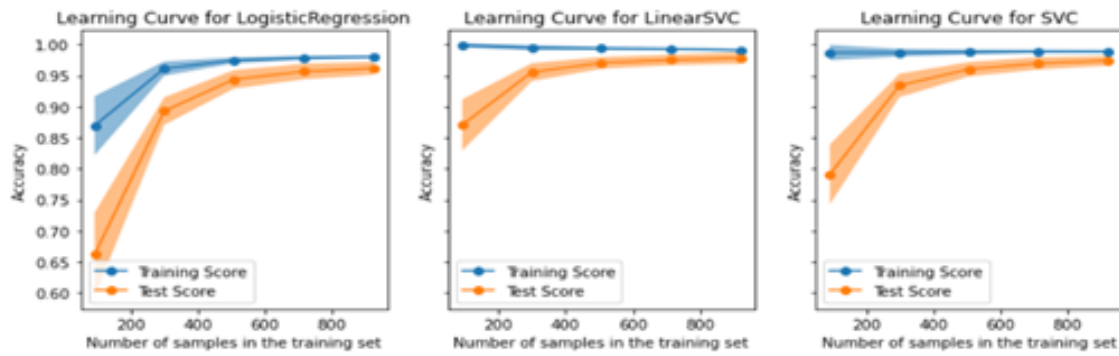


Fig. 4.5 Learning Curve Group Prediction Model

ascertain the technique and the incident expressing the inputs, so the performance of this model is dependent on the previous two models.

The learning process is monitored using learning curves that reveal the effect of adding more samples during the training process. The effect is evident by examining the statistical performance of the models in terms of training and test scores.

Here, the learning curve of a Linear SVC, an SVC classifier(kernel=linear) and a classifier(Logistic Regression) using a processed dataset is studied. Analysis of the learning curve of classifiers indicates that the training score is very high when a few samples are used for training and decreases when the number of samples is increased, while the test score is very low at the beginning and then increases when samples are added. Training and test scores become more realistic when all samples are used for training.

We see another typical learning curve for an SVM classifier with a kernel (RBF, Linear). The training score remains high regardless of the size of the training set. On the other hand, the test score increases with the size of the training dataset.

To validate the models at the training stage, a scoring function to determine the quality of predictions and the accuracy of classifiers is needed. The correct way to choose multiple hyperparameters for an estimator is to look at learning weight parameters or similar methods such as adjusting the super-parameters of an estimator that determine the super-parameter to the maximum score in a validation group or multiple validation sets or if it is inappropriate for some super-parameter values.

The validation curve function helps in this plot, seeing the training scores and SVM validation scores for the different values of the kernel parameter gamma. Figures 4.6,4.7 and 4.8 indicate that the degree of training and verification performance is good, where the gamma value decreases and the accuracy ratio increases.

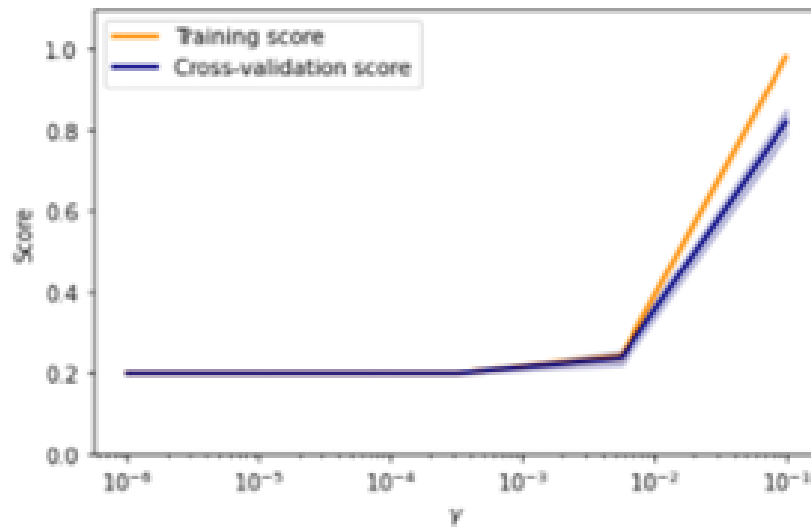


Fig. 4.6 Validation Score SVM with Techniques

4.10 Testing of ICS models

The test was conducted on the test dataset to evaluate the performance of classification models that reflect the models' ability to perform the task of detecting ICS cyberattacks. The result of validation using internal data is as follows as summarized in Table 4.2. Regarding the **technique prediction** models' performance, the Linear Support Vector Classifier (Linear SVC) demonstrates an accuracy of 95%, precision of 97%, and an F1 score of 95%. Meanwhile, the Support Vector Classifier with a linear kernel (SVC with Kernel=linear) exhibits an accuracy of 96%, precision of 94%, and an F1 score of 92%. As for the logistic regression, accuracy = 95% , precision= 88% , F1 score = 85%. Regarding the **Incident prediction** models' performance, the Linear Support Vector Classifier (Linear SVC) demonstrates an accuracy of 99%, precision of 98%, and an F1 score of 99%. Meanwhile, the Support Vector Classifier with a linear kernel (SVC with Kernel=linear) exhibits an accuracy of 99%, precision of 98%, and an F1 score of 99%. As for the logistic regression, accuracy = 98% , precision= 99% , F1 score = 98%.

Regarding the **Group prediction** models' performance, the Linear Support Vector Classifier (Linear SVC) demonstrates an accuracy of 95%, precision of 97%, and an F1 score of 95%. Meanwhile, the Support Vector Classifier with a linear kernel (SVC with Kernel=linear) exhibits an accuracy of 99%, precision of 99%, and an F1 score of 92%. As for the logistic regression, accuracy = 95% , precision= 88% , F1 score = 85%.

We used the confusion matrix to evaluate the True and False prediction performance of the three models, which is shown in Figure ().

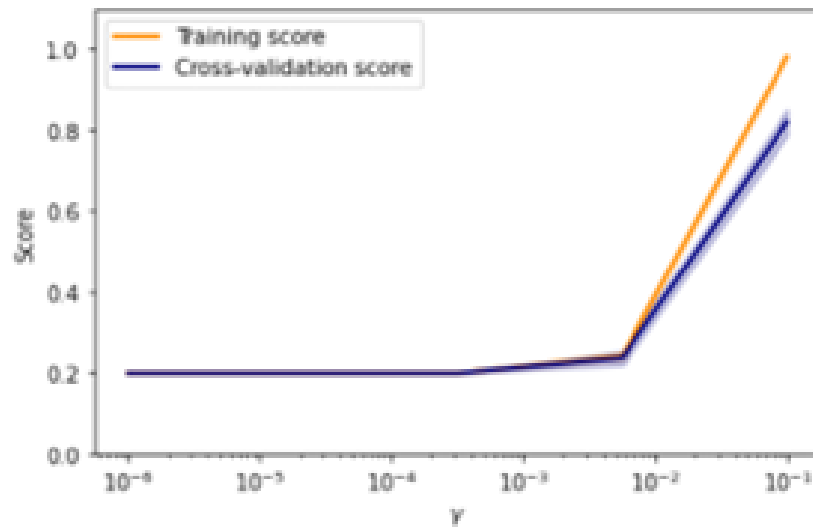


Fig. 4.7 Validation Score SVM with Incidents

confusion matrix through the main diagonal indicates that the models are able to predict well.

4.11 Validate ICS Models

It is important to evaluate the performance of the classification model on a new dataset that is not used in training. This will help ensure that the models are not overfitting and that they have the ability to generalize their results to new data. External data about techniques were obtained along with incidents, and groups using AI models that excel in data inference such as ChatGPT, Google Bard, the snippets are provided in Appendix 4. A total of 172 descriptions of techniques, 39 related to incidents, and 20 about groups were collected. Subsequently, the classification capabilities of the SVC model (Kernel = Linear) are calculated, as it exhibited the highest accuracy during the training and testing phases. The models' performance is assessed based on the verification error and accuracy among the used models.

The results presented in Table 4.3 indicate that the highest prediction accuracy using external data was achieved with the group prediction model, with a prediction accuracy of 0.91. This means the model classified correctly with 91%. A score of 0.80 was found for overall precision and F1 score alike. This signifies that the model can classify group data with external ATT&CK data accurately.

The overall accuracy of the model predicting techniques was also satisfactory, reaching an accuracy score of 0.85. This means the model correctly classified 85% of the samples in

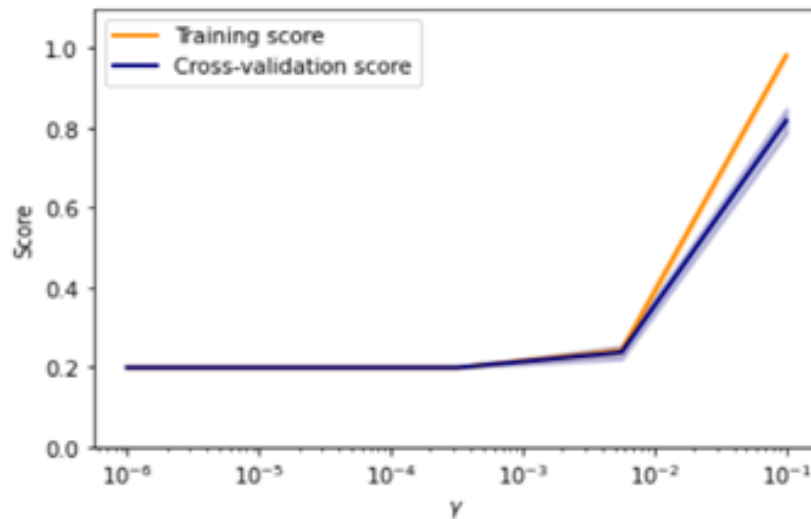


Fig. 4.8 Validation Score SVM with Groups

the external dataset. The overall precision and F1 score are 0.84 and 0.79, respectively. This implies that the model has a good overall performance across all categories.

The prediction accuracy for incidents showed average performance in event prediction, with an overall accuracy of 0.76. This means the model correctly classified 76% of the samples in the external dataset. The overall precision and F1 score are 0.80 and 0.74, respectively.

In general, the results of model verification using external data indicate that the models exhibited good performance in predicting techniques, groups, and incidents. This could be attributed to the process of generating data from ATT&CK knowledge base, which plays a role in extracting meaningful data for ICS events. However, it is important to validate external data using different datasets to assess model performance more accurately.

4.12 Hierarchical Sequence for Scenarios Generation

ICS attack scenarios consist of 72 techniques divided into 12 tactics. The attacks here are based on a hierarchy according to the order of tactics as an ICS attack. Generating an attack scenario requires linking each technique with the techniques in the next tactic, resulting in a graphical structure that can be represented in the form of a tree forest as in 5.1. This approach helps in arranging and producing all possible possibilities for scenarios. If it is assumed that the techniques are related to each other according to the tree sequence, the scenario can be generated by simply jumping between the tree nodes, where the scenario tree consists of three basic nodes. The first node represents the next entrance by the user,

Table 4.2 Performance of ICS Models

Model		Accuracy	Precision	F1-Score
	Linear SVC (C=1.0)	0.95	0.97	0.95
	SVC (Kernel=linear)	0.96	0.94	0.92
	Logistic Regression	0.95	0.88	0.85
	Linear SVC (C=1.0)	0.992	0.98	0.99
	SVC (Kernel=linear)	0.994	0.99	0.99
	Logistic Regression	0.983	0.99	0.98
	Linear SVC (C=1.0)	0.95	0.97	0.95
	SVC (Kernel=linear)	0.992	0.99	0.92
	Logistic Regression	0.95	0.88	0.85

Table 4.3 Validation of ICS Models

Model	Data(n)	Accuracy	Precision	F1-score
Group prediction	25	0.91	0.80	0.80
Technique prediction	87	0.85	0.84	0.79
Incident prediction	39	0.76	0.80	0.74

which is called the root. The second node represents the tactic where all tactics are linked as a parent contract to Root. Each tactic is associated with a set of nodes, which is the third node type and represents the techniques. Finally, each technique is associated with a partial tree, starting with the next tactic. If the entrance is significant in all techniques, this results in generating all scenarios as paths starting from the root node and ending with the child nodes. Obtaining text, that is based on a series of techniques, difficult to represent directly, otherwise. Therefore, the previous tree structure is presented to ensure that all scenarios, true or false, are recognized. What matters the most is to determine the paths that generate the scenarios. Then, it is assumed that the inputs will complete all these paths. The purpose is to depend on the total number of possible scenarios for any entry. which is calculated according to the following formulae.

$$y_n = a_n \quad (4.1)$$

$$y_{(n-1)} = a_{(n-1)}(1 + y_n) \quad (4.2)$$

$$y_{(n-2)} = a_{(n-2)}(1 + y_{(n-1)} + y_n) \quad (4.3)$$

$$y_1 = a_1(1 + y_2 + \dots + y_{(n-1)} + y_n) \quad (4.4)$$

$$y = (y_1 + y_2 + \dots + y_{(n-1)} + y_n) \quad (4.5)$$

Where $y_1 \dots y_n$ indicates the number of scenarios that can be produced for all techniques and for each tactic. y refers to all the possible scenarios to represent an input with all tactics. The approach presented earlier is intended to arrange the possible paths for each scenario, however, it is necessary to find optimal scenarios that express the entrance. Therefore, a set of weights is added to each technical node through which the path with significance on the entrance is determined. These weights are extracted during the trait extraction phase. The input node is related to the techniques through the compatibility of weights and conditions between them according to the set of descriptors extracted from ICS incidents. If the entrance node expresses the technique, the technique node is created and linked to the input. After that, comes the partial tree, which is related to the technique. The search for optimal scenarios depends on the similarity between the techniques and the characteristics of ICS incidents, which are represented in two cases:

1. Choosing the largest percentage similar to the weights that link the technique with the techniques in the next tactic, which results in optimal scenarios.
2. Determining the value of the threshold, which determines the lowest percentage of weights for the techniques associated with the entrance, where if the weights are greater or equal to the value, scenarios will be generated, but if the values of the weights are less, it is not allowed to generate the scenario.

4.13 ICS Scenario Generation Model

5.1 refers to the architecture of the scenario generation model, which it consists of two basic parts, as described.

The first part is responsible for dealing with the entrance, processing, analyzing the texts of the attacks, and predicting the technique and the appropriate incident.

The second part passes the disassembly and the incident in the tree of scenarios, which in turn works to produce a series of techniques such as an ICS attack.

The overall process of generating attack scenarios is outlined in the Algorithm 1. First of all, it calls "*input_text()*" as shown in step 1, to get the textual entry. Then, "*nlpCle-*

anData(Input)" performs initial processing and semantic analysis on the input text and replaces the synonyms of semantic words with attack, in step 2. In step 3, the form returns "None" if the text does not express an ICS attack. If there are outputs from the NLP model, the process of searching and predicting the appropriate technique is initiated by the "*model_technique(cinput)*" as can be observed in step 4. The prediction of the technique and the search for the tactic to which it belongs is confirmed through "getTacticByTechnique(tech)" at step 7. Then "*model_software(Cinput,tech)*" predicts the incident that is related to the technique and entrance, in step 9. Afterwards, in step 13, the process of generating scenarios begins through "*generate_tree_scenarios(tactic,tech)*". That conducts the tree search process according to the search case, which includes the largest percentage of similarity or threshold value to determine the paths chosen as appropriate scenario paths for the entrance. Finally at step 17, "*cheak_scenario(scenario,cinput)*" checks the status of scenarios, which depends on the percentage of belonging of the series of techniques to the incident scenarios of the ICS attack.

Algorithm 1 ICS Models

```

1: Input: input ← input_text()
2: cinput ← nlpCleanData(input)
3: if cinput is not None then
4:   tech ← model_technique(cinput)
5:   if tech is Technique then
6:     Print: Technique : tech
7:     tactic ← getTacticByTechnique(tech)
8:     Print: start Tactic search to scenario tactic
9:     soft ← model_software(Cinput, tech)
10:    if soft is Software then
11:      Print: input same as Software is : soft
12:    end if
13:    scenarios ← generate_tree_scenarios(tactic, tech)
14:    if scenarios is not empty then
15:      states ← {}
16:      for scenario in scenarios do
17:        states[scenario] ← cheak_scenario(scenario, cinput)
18:      end for
19:    end if
20:  end if
21: end if

```

4.13.1 Checking ICS Attack Scenarios

Figure 5.1 represents the scenario status check pane that looks for the closest incident that can represent the scenario and identify the group related to that incident, where each scenario produced by the model undergoes an important examination process that determines the percentage of its affiliation as an ICS attack. This process depends on the probability ratio of the series of techniques as being one of the ICS incidents. The Algorithm 2 illustrates this verification method. "*mode_software_ptedictprob(cinput + scenario)*", in step 2, is based on estimating the probability ratio of the scenario as an ICS incident, this method is adjusted with the incident prediction model and then determines the closest incident to the scenario. If the score value is greater than or equal to 85%, it means that the scenario represents an ICS incident as shown in step 3. This also implies that all the series of techniques must belong to this incident and mark the status of this scenario as (success). If the score value is greater than or equal to 65%, this means that the series of techniques expressing the scenario can represent an incident, but some of them do not belong to the group of techniques of the incident. However, by evaluating them based on the percentage of similar techniques, this condition is called (close to success), the condition is depicted in step 6. Similarly, when the score value is greater than or equal to 50%, this indicates that there are some techniques from the scenario series that do not belong to the incident and are not similar to its set of techniques. This condition is called (init). Finally, for the rest of the cases, the scenario is considered a failure as shown in step 12, which means that the scenario cannot be represented as an incident or even part of it, where all techniques do not belong and are not similar to one of the techniques of ICS incidents.

Algorithm 2 Check Scenario

```

1: procedure CHECK_SCENARIO(scenario, cinput)
2:   rate ← model_software_predictprob(cinput + scenario)
3:   if rate ≥ 85 then
4:     return "Scenario is success"
5:   end if
6:   if rate ≥ 65 then
7:     return "Scenario is close to success"
8:   end if
9:   if rate ≥ 50 then
10:    return "Scenario is init"
11:  end if
12:  return "Scenario is failure"
13: end procedure

```

4.14 Chapter Discussion

Modelling and analyzing cyberattacks automatically is a crucial task for improving information security. These methods can help understand how attacks work, identify system weaknesses, and develop effective defensive measures.

There are various different methods for modelling and analyzing cyberattacks. These methods differ in several aspects, including comprehensiveness, accuracy, applicability, and ease of use. It can be said that methods based on the ATT&CK knowledge model are the most comprehensive and accurate. However, they are also the most complex and may be more challenging to use.

On the other hand, methods that rely on graphical representations, such as attack tree models, have the advantage of being easy to use. However, they may be less comprehensive, accurate and static than methods based on the ATT&CK knowledge model.

Ultimately, the best method depends on the specific application. For example, the method based on the ATT&CK knowledge model may be suitable for analyzing broad security threats, while graphical representation-based methods may be suitable for analyzing specific attacks.

The thread-based model relies on AI and ML techniques to automate the creation of attack scenarios. This is a promising approach because it has the ability to improve the accuracy, scalability, and efficiency of creating attack scenarios. According to the results of classification models for techniques, incidents, groups, and hierarchical sequence (Mesh graph) of ATT&CK knowledge-based scenarios, the model can be used as an effective tool to create a large number of attack scenarios. This can be useful for security analysts and researchers trying to understand and mitigate the risks posed by ICS attacks.

In general, the model contributes to the field of industrial system security. The proposed method is a promising approach for creating attack sequences and can be used to enhance the security of industrial systems.

4.15 Conclusion

As has been mentioned in this chapter, threat-based attack scenario generation was executed and tested using internal data and has been validated using external data. An interface was created for this model. Based on the percentages, the accuracy results for this model are proven to be high and satisfactory.

Chapter 5

Attack Scenario Generation: State-Based

5.1 Introduction

As detailed in Chapter 4, attack scenarios were generated as an ICS attack, which contributed to identifying all possible paths generated as attack scenarios, provided that each possible attack scenario is based on the method of sequencing tactics such as an ICS attack. However, it is plausible that attackers need to develop new attack patterns to find another way to hack and dodge the security measures in place. Therefore innovative means of ICS attack tests are required, as routine testing is meaningless because the tests will generate monotonous results. The cyber security team need to constantly update the security system. They need to check the potential new attack path whenever new vulnerabilities or attack patterns are detected. This nature of defensive operations makes the overall security setup quite time-consuming. To overcome the manual setup, generating attack scenarios by automated means seems a promising replacement.

However, there are numerous attack scenarios that are hard to anticipate or implement. Therefore, this research thesis has worked on an attack scenario-generating model based on security analyst knowledge, which lets the possibility of generating attack scenarios according to a set of attack techniques. Its primary goal is to provide the most likely attack for a routine threat by evaluating it using known vulnerabilities. This model performs a lot of iterations based on attack scenarios to scan existing vulnerabilities and manages a series of attacks in order, based on attack scenarios. In the next section, some related works are provided to get a better understanding.

5.2 Related Works

Concerning the previous studies conducted in the state-based solutions, this section elaborates on some significant work, such as CALDERA. The CALDERA is developed by MITRE corporation and it performs automatic attacks on the targeted systems using techniques and tactics described in the ATT&CK framework [61]. Applebaum et al. [98] introduced the application and design of CALDERA. Miller et al. [33] illustrated how to deal with the planning issue of adversary emulation by CALDERA, when CALDERA needs to handle the uncertainties of the targeted systems. The CALDERA analyzes automatically mistrusted behaviours and recognizes their maliciousness with the ATT&CK framework. It mainly focuses on analyzing the detected threats.

Ou et al. [99] also introduce a study based on MulVAL [100]. MulVAL is a system for automatically determining security vulnerabilities in networks. The core idea of MulVAL is that the configurations could be introduced as security semantics and Datalog tuples. Most attacks can be introduced as Datalog rules. Each rule can be seen as a logical formula. It evaluates the Datalog rules using XSB, a prologue system. The researchers made some modifications on the MulVAL system to record the evaluation track and send it to the graph-building algorithm, where the Attack Graph is created. In each successful query, a function notices the successful deviation to the trace file. The researchers also represented an algorithm to change the traces into an attack graph. The transformation takes each trace step and changes it into a piece of a graph. This study focuses on network security, which means that the generated Attack graphs do not cover General Attacks.

Another effective study is conducted by Ibrahim et al. [101]. They introduce a solution encompassing Hybrid Attack Graphs (HAG). In general, HAGs notice the changes in logical parameters, as identified by the pre and post-conditions, representing the state or situation of the system under attack. HAGs also capture the resilience level, which is a number that indicates the system under a sequence of attacks dynamic response. The worst-case scenario is chosen to be included in the Attack Graph, and that is based on the resilience level. Automated HAGs can be visually represented with a Java-based tool. This step requires a formal description of the model of the system and the security property.

The graph represented by Tippenhauer et al. [102] is called a “Goal-System Attacker” graph and it is another contribution to be put into consideration. To generate this graph, three steps are followed. The Goal graph or G-graph, which depicts the security goal and the description of the workflow, is initially constructed using the system’s workflow and security goal. In order to create the goal-system graph (GS-graph), the result is coupled with the system description. The attacker model is then added to generate the GSA graph (goal-system-attacker graph). Every time a procedure is done, the graph is updated with new

data about the environment. Using the paradigm mentioned above, the authors discovered various patterns. Some templates were applied using these patterns. Then, using the given templates, they built local extensions to generate these graphs gradually. The template is added to the main graph if a matching node is found.

A method to create an attack graph, where each node stands for a network state, was proposed by Swiler et al. [103]. The inputs required for the program are the network configurations, the attacker's profile, and the attack templates. The attack templates describe the phases of well-known attacks or transitional tactics. The program puts together the input data and alters the general template attacks in accordance with the attacker profile and network setup. Every graph contains a few variables that demonstrate the network's state. A new edge is added to the network and new nodes are produced when a node in a graph meets the template's requirements.

Adding to all the previous contributions, a technique to facilitate developing an attack graph using Machine Learning and Deep Learning is presented by Koo et al. in [104], where the network topology and system information act as an input into the final model. By utilising the input data, feature extraction is used to get an attack graph-generating model. Finally, the path predicted is evaluated by the researchers using an evaluation measure. In order to build an attack graph starting with knowledge of the network and the system, the researchers provide a binary classification problem together with a multi-output learning technique.

This related work details all the previous studies conducted in the scope of state-based solutions related to attack scenarios. In the next Section i.e., 5.3 tackles the methodology, highlighting all the technicalities carried out in the model.

5.3 Methodology

Generating attack sequences automatically based on system state. The two key factors referred for automated attack planning are the 'Initial State' and the 'Attack Technique.' These attacks should be linked to the actions of the security analyst, who must be aware of the initial attack state. This is represented by a data structure containing a description of the initial state, such as lists of IP addresses, ports, acquired privileges and the operating system state, all related to the system's target. The security analyst then plans and selects attack techniques based on an ICS attack. Attack techniques have both preconditions and postconditions, dependent on the initial state. This set of conditions is outlined in Section 5.6. Subsequently, the process of generating attack sequences begins by exploiting attack techniques according to the system's state. Attack sequence generation are illustrated as a graph model, where chosen techniques are connected to the initial state. . Attack methods are ordered based

on updating the preconditions and postconditions for each attack technique with the initial state. Parallel programming is employed to mitigate link paths by simultaneously exploring multiple paths. To solve the path repetition problem, the Ant Colony Optimization algorithm is used to distinguish traversed paths and generate scenarios of technique sequences. Finally, the attack scenarios are verified based on actual previous incidents in ICS using the validation model discussed in Chapter 4.

5.4 State Based ASG Architecture

In this section, we present the attack scenario generator architecture and how to create sequences of attacks automatically. Two main factors are referred to for planning automated attacks inspired by (STRIPS and Caldera) are the "starting state" and the "attack technique". The security analyst must be aware of the attack situation at the starting point, which is a data structure that includes a description of the starting state (for example lists of IP addresses, ports, acquired privileges and operating system state) related to the system target. Then, the security analyst works on planning and selecting attack techniques according to the ICS attacks. Attack techniques have preconditions and post-conditions depending on the starting state, this set of conditions is described in section (5.6). After that, the process of generating attack scenarios begins by exploiting attack techniques according to the system State. For example, the security analyst wants to see whether it is possible, by extracting the system state from the log, IP and Protocol to produce an attack scenario. For example, the target source IP:192.168.0.1 address and Protocol:IPv4 are specified as the starting state and the automated system generates attack scenarios includes (PoetRAT).

Figure 5.1(b) shows the architecture of State based attack scenario generator. The process relies on a set of sequential steps that require the identification of a customised start state as an initial attack that determines the type of access. Next, select and identify techniques intended for generating attack scenarios and verify them if they represent an ICS attack. Its different components are described here, while an in-depth discussion of each component is followed in the preceding sections.

State Form

It is a set of descriptive inputs on which the model relies in determining the state of the system represented by lists, they are more identical to key-value pairs.

(**Address** : IP address),

(**Protocol** : protocol used in the port),

(**OS** : Operating System info.),

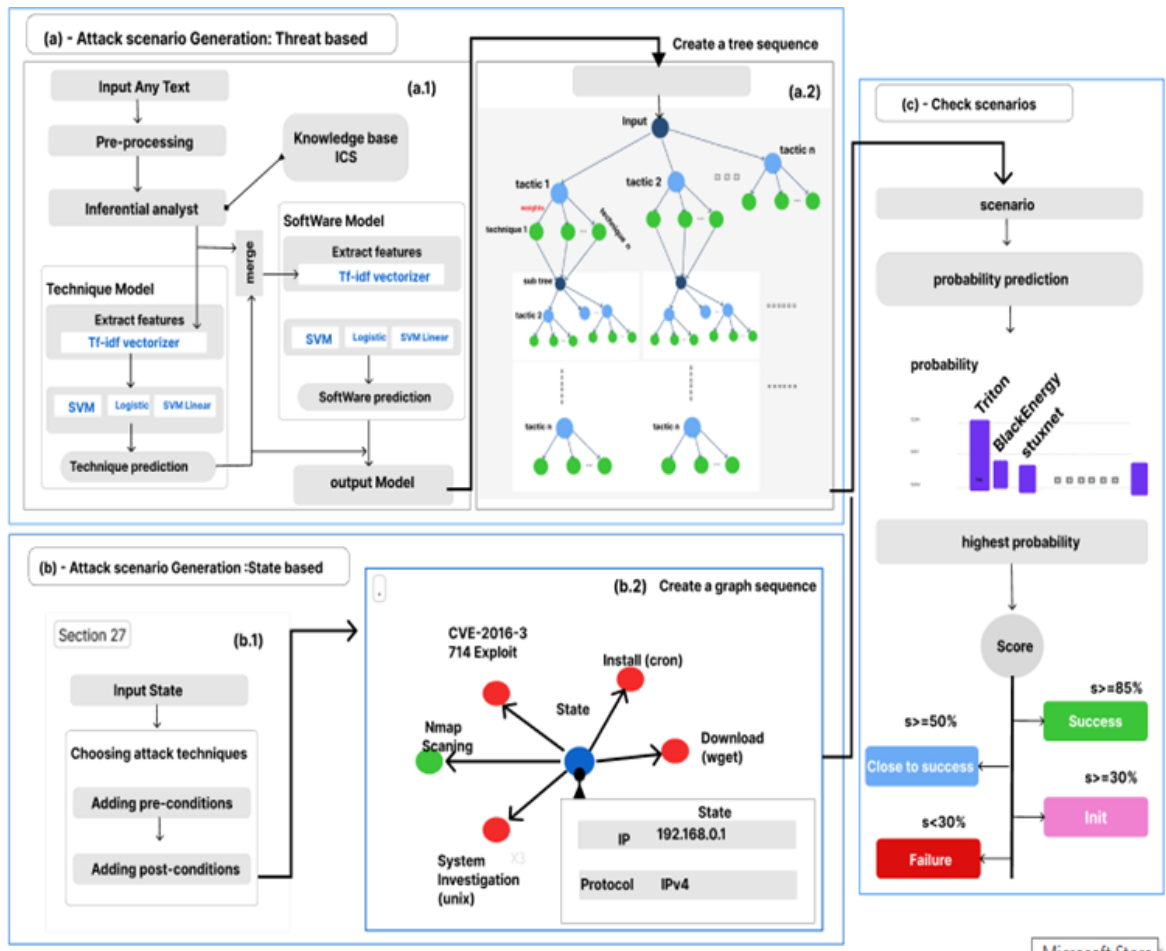


Fig. 5.1 ASG Tool Architecture

- (Ports : list of open port),
- (Shell : ‘TEMPORAL’, ‘DOWNLOADED’, ‘PERMANENT’),
- (Privilege : superuser, regular user),
- (Binaries : list of binaries installed on the target node),
- (CWD : current working directory),
- (Misudo : exploitable sudo configurations).

The state entry form allows the security analyst to add attack state fields, for example, the analyst can specify a state starting from IPv4:IP:192.186.0.1, Protocol and can add more entries as needed. Section 5.5 explains the above key-value pairs and their settings in detail.

Attack Techniques Database

It is a list of selected techniques determined by the security analyst in order to create the sequence of attacks based on the starting state. It relies on the ICS attack techniques repository. Each technique comes with a set of fields that include the name of the technique and tactic to which it belongs, in addition to the fields of the pre and post-conditions, and a set of functions described in Section 5.6.

Attack Scenario Generator

The attack scenario generator obtains the attack techniques and the starting state identified by the security analyst to achieve a goal. For example (generating Stuxnet attack scenarios) and then working to generate attack sequences where the process of generating scenarios begins by linking all attack techniques to the start state directly and linking techniques with each other forming a group. Further in-depth discussion is followed in Section 5.7.

Attack Scenario Validation

Attack scenario validation model described in Figure 5.1 , is used to examine the sequence of attacks produced by the model. The verification process determines the similarity of scenarios represented to ICS incidents as defined earlier.

5.5 System State

The system state is represented by the initial attack determined by the security analyst with a set of fields that describe the starting state displayed in Table 5.1, where it is an approach inspired by (Strips), famous for the automated planning method A 9-digit system state has been proposed and is as follows:

- **IP address:** addresses of the target devices. The address is entered as a text variable that is checked to ensure that the IP is correct and has a numeric(octagon) variable value starting from 0 to 255.
- **Protocol:** It represents a list of internet protocols that are used as plugs to connect the network between devices or others. More than one protocol can be added, for example, TCP and UDP.
- **Ports:** A list of "ports" is basically their port numbers. These port numbers are the point of access to the target node by the security analyst. The list contains open port

numbers, and the list can contain one or more open ports if the target server manages one or more services. This field is valid only if the Protocol is IPv4 or IPv6.

- **Operating Systems:** The type of the operating system affects the set of commands or vulnerabilities that are issued only for the process of accessing, executing or downloading from an attack technique. Because the commands that will be sent are different depending on the operating system, determining the operating system of the target system is essential. Using specific Operating System information, different commands can be used to attack the host OS. Furthermore, the analyst can specify more than one operating system.
- **Shell:** indicates the state of the shells. It contains three values: "TEMPORAL", "DOWNLOADED" and "PERMANENT". "TEMPORAL" is a state that temporarily takes the current cover. If the connected process is terminated, "downloaded" is the case where the security analyst sent for example RAT to the target node to get the shell permanently. "Permanent" is a state that means that the download state has expired. The shell is used to access the target node. For example, using "cron" job, the program for scheduling jobs in an operating system similar to Unix, it is possible to set the installed binaries to be restarted periodically if terminated.
- **Privilege:** It is a level of authority gained over the target node. The desirable privilege is an administrative authority i.e., the superuser. Which is the highest level of permission granted. The other type of privilege may be a regular user or with a lower authority.
- **Binaries:** It is the list of Binaries or executable programs, installed on the target node. agent successfully infiltrates the target node.
- **CWD:** is a Current Working Manual for the projectile. Based on this information, the security expert can know the absolute path and relative path of the file system on the target node.
- **Misudo:** It is a set of exploitable sudo configurations. For example, "sudofind" can be an element of this array, since it makes the "find" command execute without the root password.

The state of the system gives the analyst the ability to determine the real conditions for generating attack scenarios. For example, a security analyst needs to plan an initial attack that involves using Protocol and IP to access the target system. System state can also ensure that the model is suitable for more than one purpose, for example, it can help the security team or the Red Team to generate their attack scenarios where the Red Team agent can determine

Table 5.1 Initial System States

Address IP	A list of target machine addresses
Protocols	A list of users protocols to reach the goal
Ports	List of port numbers used to reach the target
OS	List of operating system types for target devices, for example (Windows, Linux, etc.)
Shell	State of the shell to access the target node (TEMPORAL or DOWNLOADED or PERMANENT)
Privilege	List of a set of privileges earned on target devices
Binaries	A list of a set of Binaries that are implemented in target devices.
CWD	Current working directory of the shell which have accessed to the target node
Misudo	Array of exploitable sudo configurations

the state of initial attacks. In addition, the model has been given the ability to link directly with cybersecurity applications such as (SIEMs). It can also be combined with a Listener to identify the start state.

5.6 Attack Techniques

A security analyst uses ICS attack techniques to launch automated attacks. The analyst's job is to decide the attack method in the initial case and then to suggest techniques. Each technique comes with a set of fields and functions, chosen by the analyst, including the name of the technique and the name of the tactic to which it belongs, the pre and post-conditions, and the search and matching functions, all of which are shown in the Table 5.2 The preconditions and postconditions are a partial state of the system state consisting of 9 fields shown in table (5.1) With each technique, the "pre-conditions" and "post-conditions" are defined. **Preconditions** are a set of descriptive conditions that define the requirements for an attack technique, where each precondition represents a partial description of the state and can have more than one condition. The Start state field can be a factor representing **IP: 192.168.0.1** and **Protocol: tcp**, and there are two states for it: **SC and INCLUDE**. As an example of a precondition(IP:192.168.0.1, Protocol: tcp) if the precondition matches the IP and protocol that means the state is the same and it is called SC. whereas if the precondition matches the IP and protocol however it has extra content as example port: TCP, UDP, HTTP that's called **Include**. As for

Table 5.2 Attack Techniques

Technique Name	The name of the attack technique
Tactic Name	The name of the tactic to which the attack technique belongs
Pre-conditions	The requirements of the crisis are met in order to exploit the technique of attack
Post-conditions	Conditions that are updated when the pre-technical conditions are met with the attack state
Verification()	The method responsible for verifying that the prerequisites for the attack technique are met
Update()	The method responsible for updating the attack status with the post-condition fields of the attack technique

the **post-conditions**, they represent a partial state also determined by the security analyst in order to update the attack state that has been met after the success of the attack method. For example, the analysts define post-conditions, which represent Linux OS: Thus, if the pairing process matches the pre-conditions, the starting state is updated (**IP: SC, Protocol: SC, OS: Linux**) directly, which produces a new attack state looking for a new association with another technique.

The method (**validation**) is responsible for the process of verifying the compatibility of the pre-conditions with the State that relied in the matching process on two states. One is when the security analyst needs to define strict sequences based on direct equality conditions only and does not include a small percentage of similarity, for example, if a starting state containing **IP: 192.168.10.1, port:80, OS:Linux** is specified. The pre-condition for the attack technique must be completely identical to the state, otherwise, the production of attack scenarios will be ignored. The other state does not guarantee the conformity of all cases of pre-conditions with the case, but only the availability of one condition at most, for example, the correspondence of IP, Port.

The method (**Update**) works by updating the state according to post conditions, for example, if the previous state is true, it is updated by adding (**shell:TEMPORAL**) so the attack state becomes (**IP: 192.168.10.1, port:80, OS: Linux, shell:TEMPORAL**)

5.7 Generate Attack Scenario Process

The process of generating scenarios depends on the set of techniques of the ICS attack. The first step is to determine the start state. The state itself when represented in a graphical shape reveals nine rows, each row representing a box from the previously mentioned states.

The starting state is an object of this graphic type and needs to contain one of its digits on the information at most, for example (IP, port) and then the technique that fulfils these conditions is searched. The analysts are provided the ability to define the conditions for the start state, where they can share their expertise to define the state that represents each technique. Next, to start with the attack scenario generation process. The task of the automated system begins here in the generation process because the starting state will pass through all the techniques at the same time, and each technique must be linked to all the techniques, forming a mesh or graph, as shown in Figure 5.1. This leads to taking a large time to generate a set of scenarios if possible, parallel programming is leveraged to reduce the time. Where the thread line was linked with each initial state with the techniques to produce many scenarios at the same time. This procedure will be accompanied by the possibility of producing repeated scenarios, by marking the paths that have been taken, the issue of repeatedly producing scenarios can be resolved. This is very similar to the search method used in the ant colony optimization algorithm (ACO), which identifies signs on the nodes and the paths that it took in order to choose the best paths. The point of concern here is about numbering or marking paths so that if a scenario is started to be produced, it is rejected directly.

5.7.1 Sequence Structure for Scenario Generation

In Figure 5.1, a directed graph represents the set of techniques used for generating scenarios. Each technique is connected to all other techniques, forming a network of nodes. Hypothetically, each node is linked to the starting state to enable the production of all possible scenarios. This type of scenario does not rely on the specific order of events, such as an ICS attack, but rather is compatible with the given state. The purpose of this representation is to facilitate the creation of all possible scenarios.

This type of generated graph often results in unsolvable problems, provided when there is a large number of techniques. Which makes it similar to the Travelling Salesman Problem (TSP). Therefore, the option is to rely on pre-conditions and post-conditions. Which represent states that correspond to the state of the system to loosen the link of paths between the nodes of techniques.

This section shows an example of how a security analyst can create attack scenarios. Where

the security analyst determines the starting state and a set of attack techniques in addition to the Pre and post-conditions. After that, each technique is linked to the state directly and the process of searching for a match with the starting state is completed. Figures from (5.2 to 5.7) show one example of state-based scenario generation. Each step consists of three parts as defined.

"**Techniques Pool**" is a graph of attack techniques that are associated with the start state to generate attack scenarios, while "**Update and validation State**" part is responsible for the verification and update process of the state. The "Update and validation" part performs the process of matching the post-conditions of the attack techniques, and updating the starting state with post-conditions, where the state is called "Update-State". The last part, "**Scenario Techniques Sequence**", collects a sequence of techniques that express the attack scenario.

Figure 5.2 describes the initial state of scenario generation. It displays the start state that stores the IP address and protocol information of the target system which is **192.168.10.1** and **IPv4** respectively, referred to as "State" and are directly related to all attack techniques. The search for an available attack technique initializes from the set of techniques, that are associated with the state, where the pre-conditions that correspond to the attack state are validated. A *red* technique node means that the verification process has failed to match their preconditions with the state. The *green* node indicates an attack technique "*Nmap Scanning*", which matches the preconditions with the "State". Because its pre-conditions are met i.e., where IP is **IP:192.168.0.1**, and protocol is **IPv4**. The state is updated with the post-conditions of the *Nmap Scanning* technique and is created by the method (Update) and then the state becomes "Update-State", updating the (IP and Protocol) fields with "SC". Updating to "SC" means that they have been overtaken, in the sense that they will agree with any new technique even if they do not appear within their pre-conditions. Figure 5.3 shows the next step, where the starting state is replaced with the "Update-State" and the search process begins between the attack techniques. This process also takes into account the exclusion of the *Nmap Scanning* technique and adds it to the part of the sequence of techniques that represent the scenario. Next, the scenario creator performs a validation of the preconditions to find the next attack of the technique based on its preconditions. It indicates that "**CVE-2016-3714 Exploit**" matches the state. Therefore, the state will be updated by adding fields (ports, OS, shell) so the same actions will occur as the previous state. In the next step, the start state is updated and linked with the other techniques, as shown in Figure 5.4. This figure shows the "**System Investigation (Unix)**" state match dependency on the preconditions (**OS:SC, Shell: TEMPORAL**). The state fields will be updated (Privilege, Binaries, CWD) so that their value becomes (UNDEFINED), which allows ignoring the

fulfilment of these cases. The process is also repeated for forms to choose attack techniques for the sequence.

Finally, Figure 5.7 shows that the scenario generator has finished the sequence of scenario techniques that appear in full in the "Scenario Techniques Sequence" and also shows the final attack state called "End-State". From the above process and knowing the starting state i.e., (**Ip, protocol**) the scenario has been generated (poet rat) which consists of five attack techniques and this represents the work of the model, State-based ASG. State-based ASG helps generate scenarios based on the state of attack and the model can produce more complex scenarios in the same way as the scenario (Poet RAT), which is one of the techniques (Stuxnet) and therefore can produce a scenario for each technique of ICS incidents producing a mesh or graph of attack scenarios.

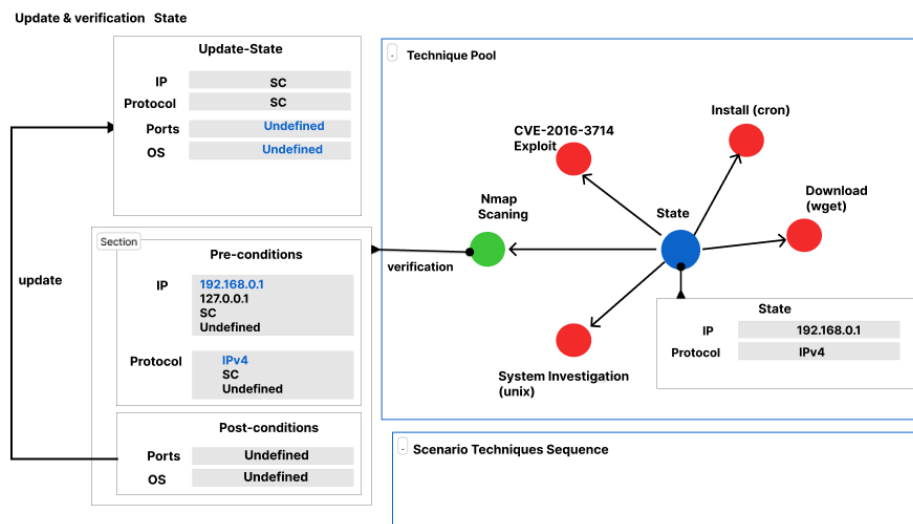


Fig. 5.2 Visual Representation of State-Based Scenario Generation Process

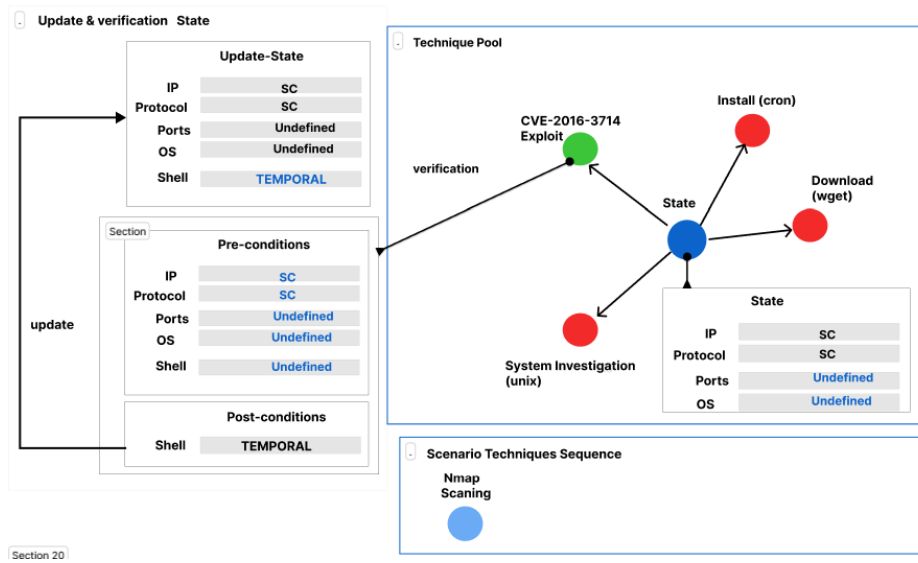


Fig. 5.3 Visual Representation of State-Based Scenario Generation Process: Port Scanning

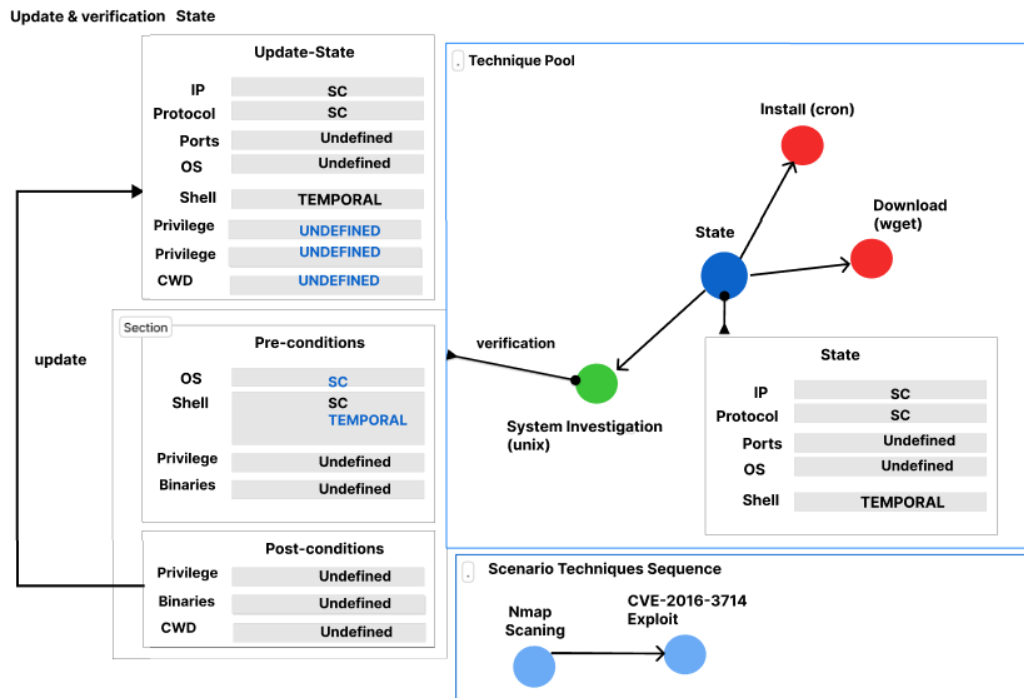


Fig. 5.4 Visual Representation of State-Based Scenario Generation Process: CVE-2016-3714 Exploitation

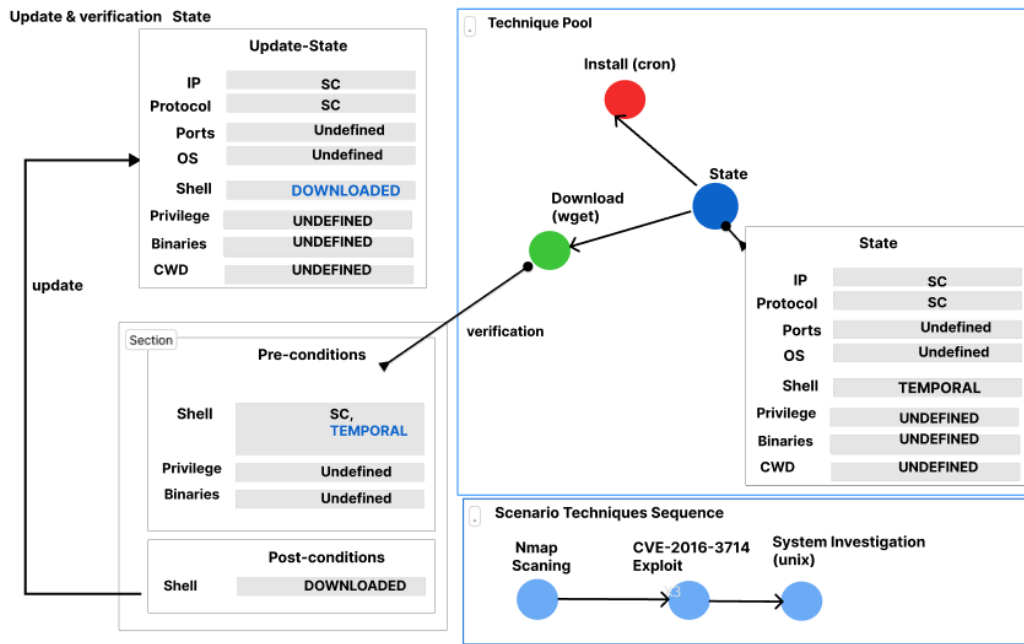


Fig. 5.5 Visual Representation of State-Based Scenario Generation Process: System Investigation

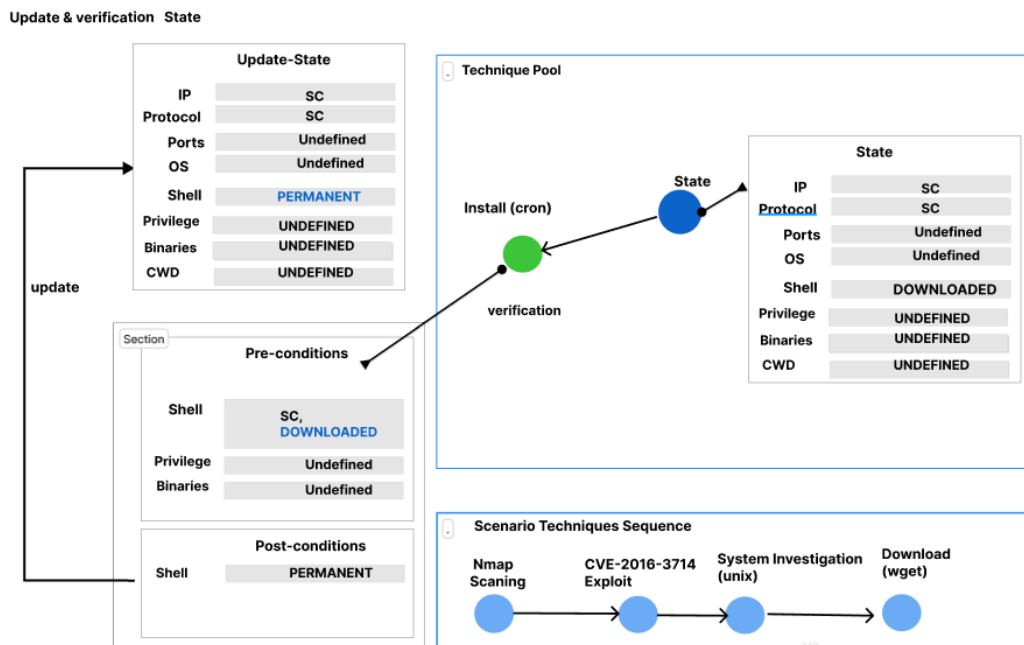


Fig. 5.6 Visual Representation of State-Based Scenario Generation Process: Download (Wget)

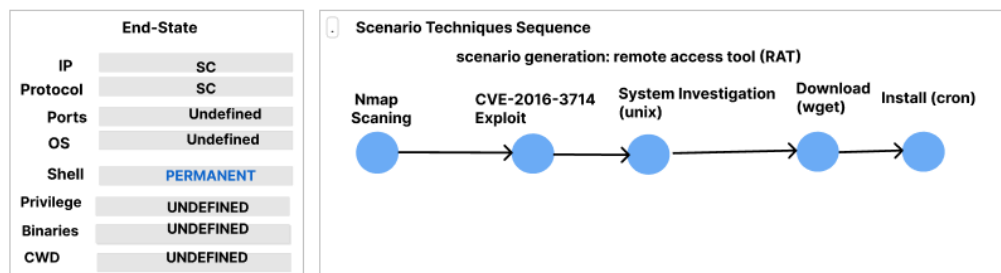


Fig. 5.7 Visual Representation of State-Based Scenario Generation Process: Scenario Generated

5.8 Validation

Since a database of scenario-generating system states is not available, which typically relies on the security analyst's expertise in selecting and assembling techniques, we validated our system state-based model by comparing the attack scenarios it generates with attack scenarios of how a red team agent generates attack scenarios, as shown in the Figure 5.8 We used the

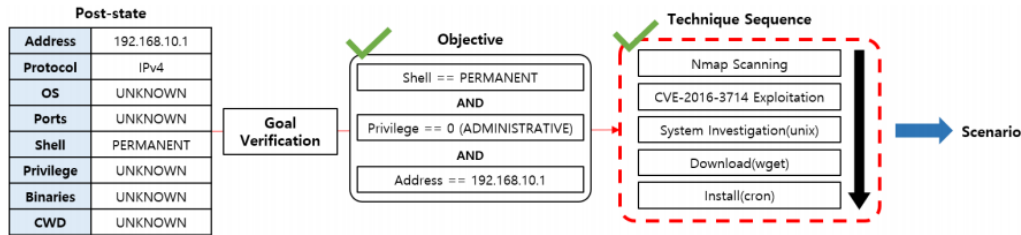


Fig. 5.8 Example Red Team of the process of scenario generation: completion of generating a scenario.

same system states in both cases [105]. The comparison showed that the model is capable of generating realistic and diverse attack scenarios, as shown in part (). The attack scenario validation model in Figure 5.1 has been linked with the attack scenario generator in order to give the analyst the possibility to compare the generated attack scenario with previous (real) ICS incidents.

5.9 Attack Scenario Generator Interfaces Implementation

Dealing with attack scenario generator models relies on a large number of inputs to determine the state and set of conditions for the techniques. The interfaces for all models in order to test and validate the performance of the model are presented. Figure 5.9 shows the main interface, which displays stat based scenario generator. It shows two basic steps that the security analyst must identify. The first step is to add the starting state where a set of inputs is available as shown in the figure, which represents a form that allows the security analyst to enter the start state where it can include more than one row and each row can contain more than one input condition, for example, you can be the state (IP, Port, OS). Also, each field has a set of inputs that are displayed as a set of columns to add more than one condition, for example, the security analyst can through the state menu choose the OS box and press the add button, which shows a box in which OS can be added e.g., (OS=Windows) and again add a new box (OS=Linux). This helps to produce a multiple start state that can deal with more than one environment or system as shown in Figure 5.10 and 5.11. The other step taken

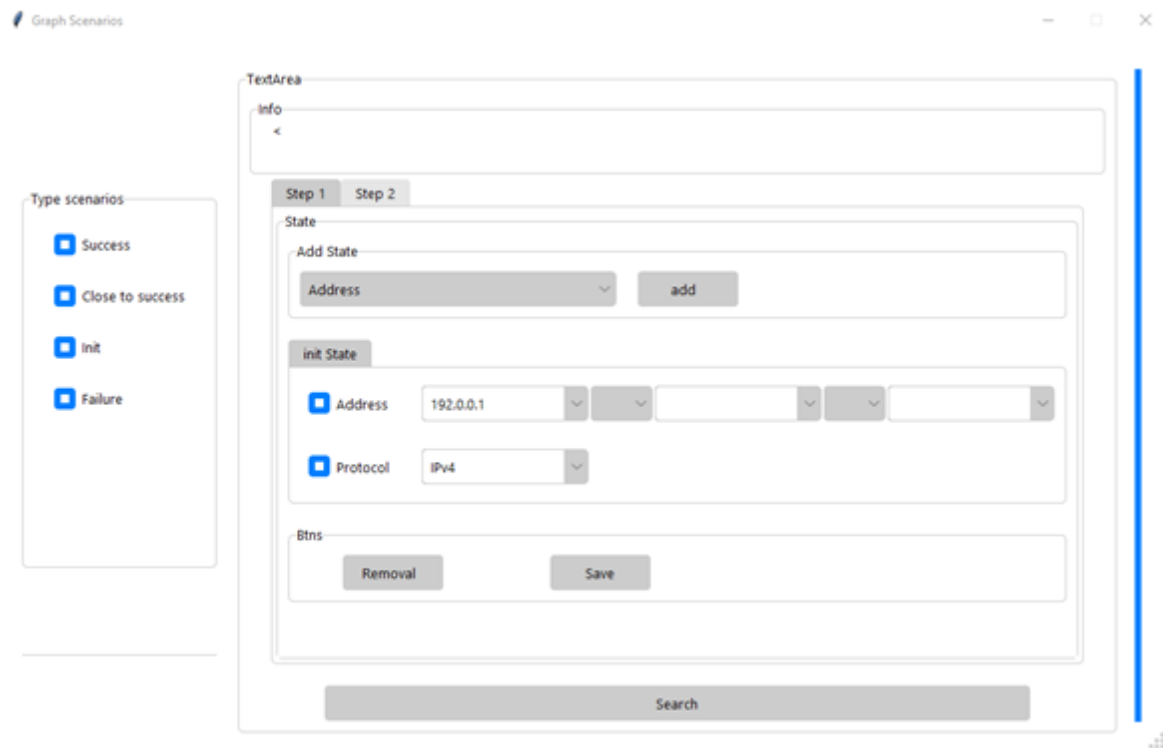


Fig. 5.9 Main Interface

by the security analyst is to identify the attack techniques from which to generate attack scenarios where the list of techniques provides all ICS attacks. More techniques can be selected and added to the list of selected attack techniques as shown in Figure 5.12. Also, the pre and post-conditions can be added by double-clicking on the technique in the list of selected attack techniques, which shows a new form window that allows him to add and modify those conditions as indicated in the figure 5.13. In order to overcome the challenge of not having sufficient data for the state of the system, security analysts are provided with the flexibility in dealing with the model.

5.10 Chapter Discussion

The threats targeting cybersecurity have become more frequent and intelligent. Attackers continually develop new techniques and tools to exploit vulnerabilities in systems and networks. This makes it challenging for cybersecurity experts to keep up with the latest threats.

We have introduced a state-based model that provides a solution for generating attack scenarios based not only on knowledge but also on the system's state. It helps identify

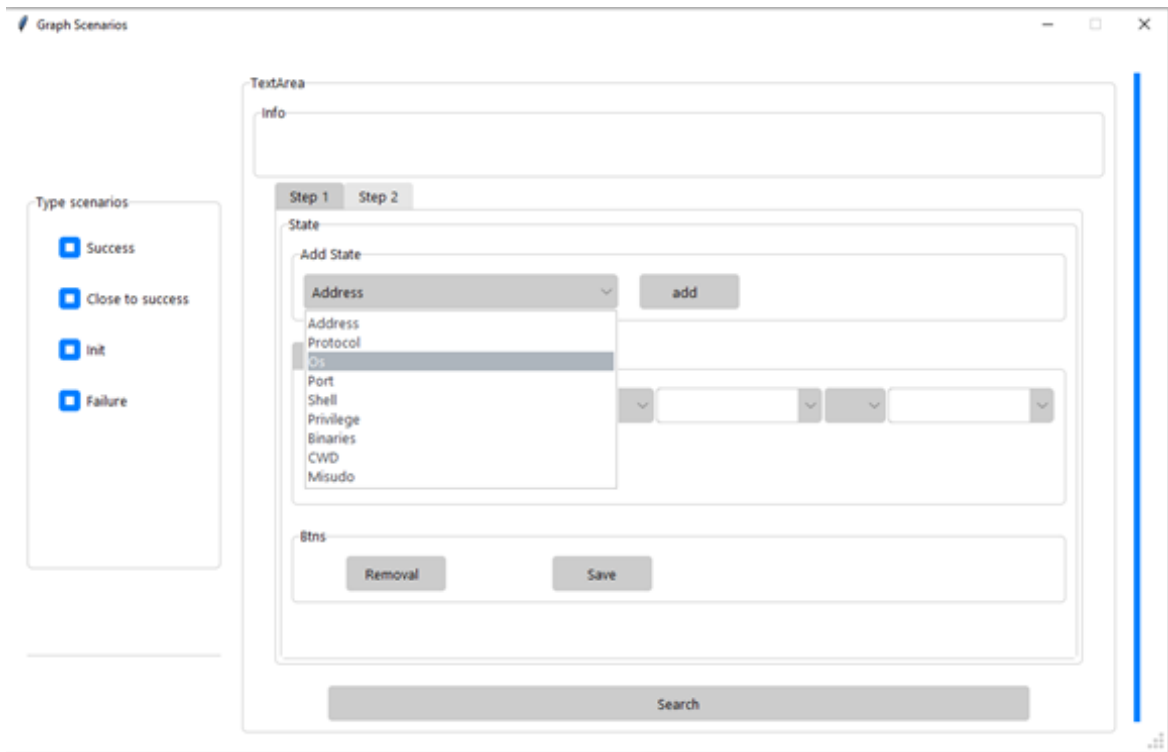


Fig. 5.10 OS Selection from the State Drop Down

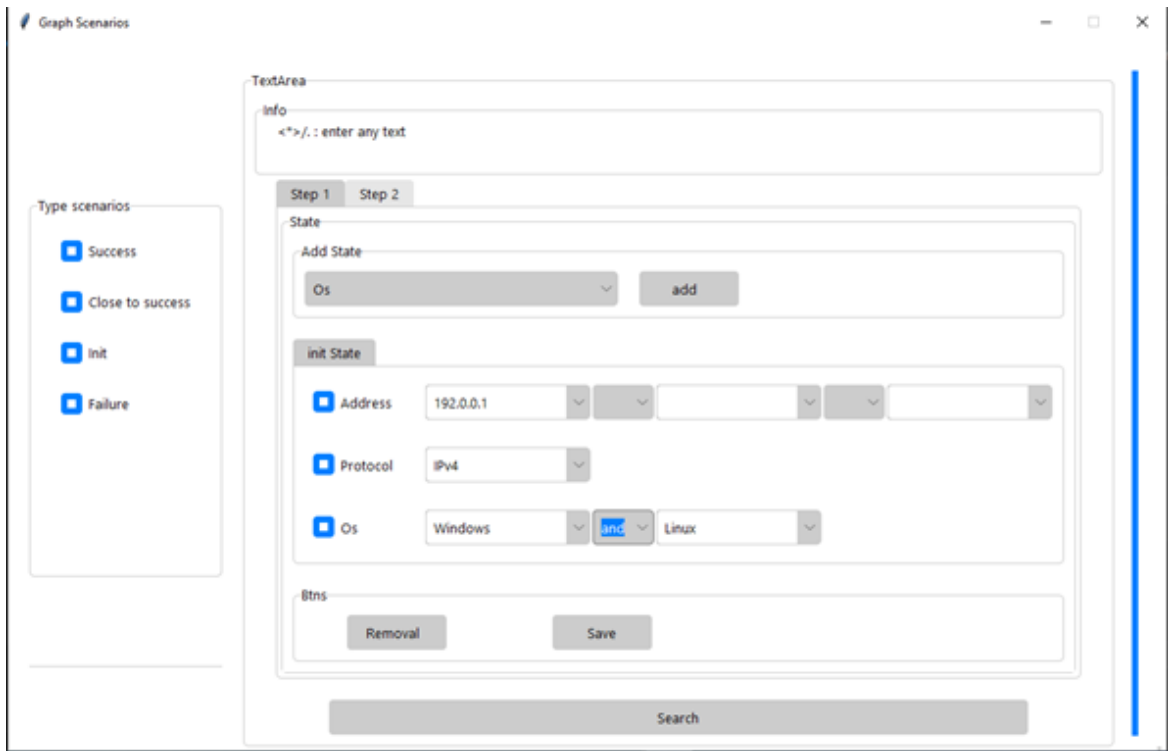


Fig. 5.11 Linux Selection from OS State

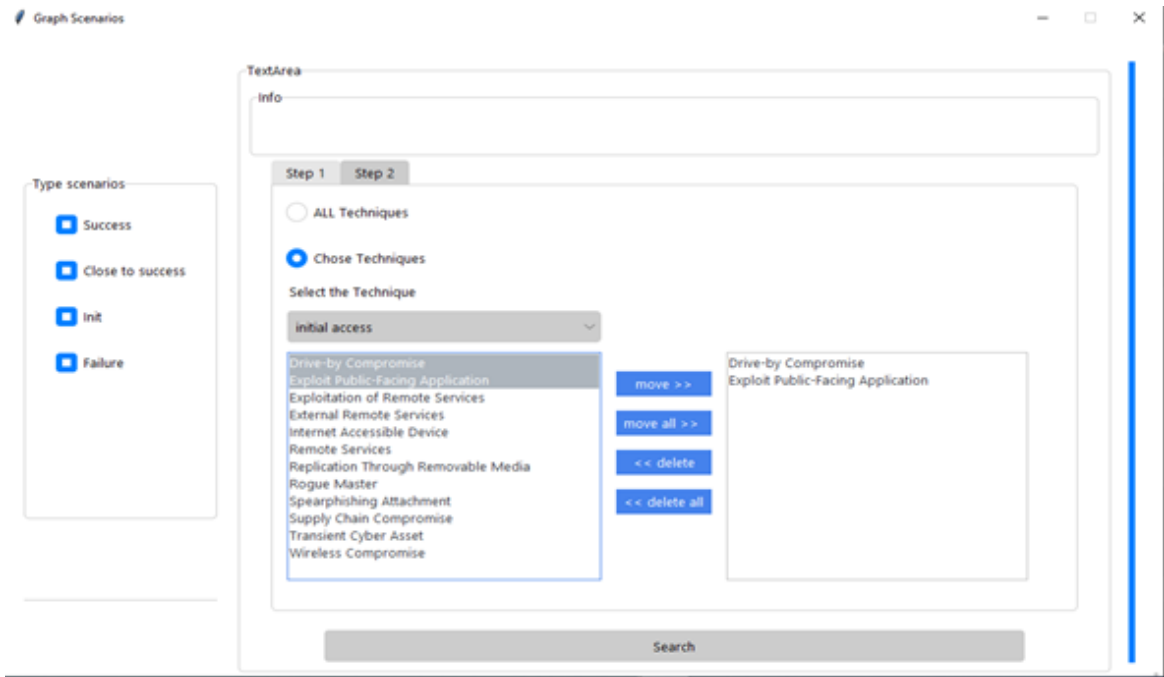


Fig. 5.12 Techniques Selection

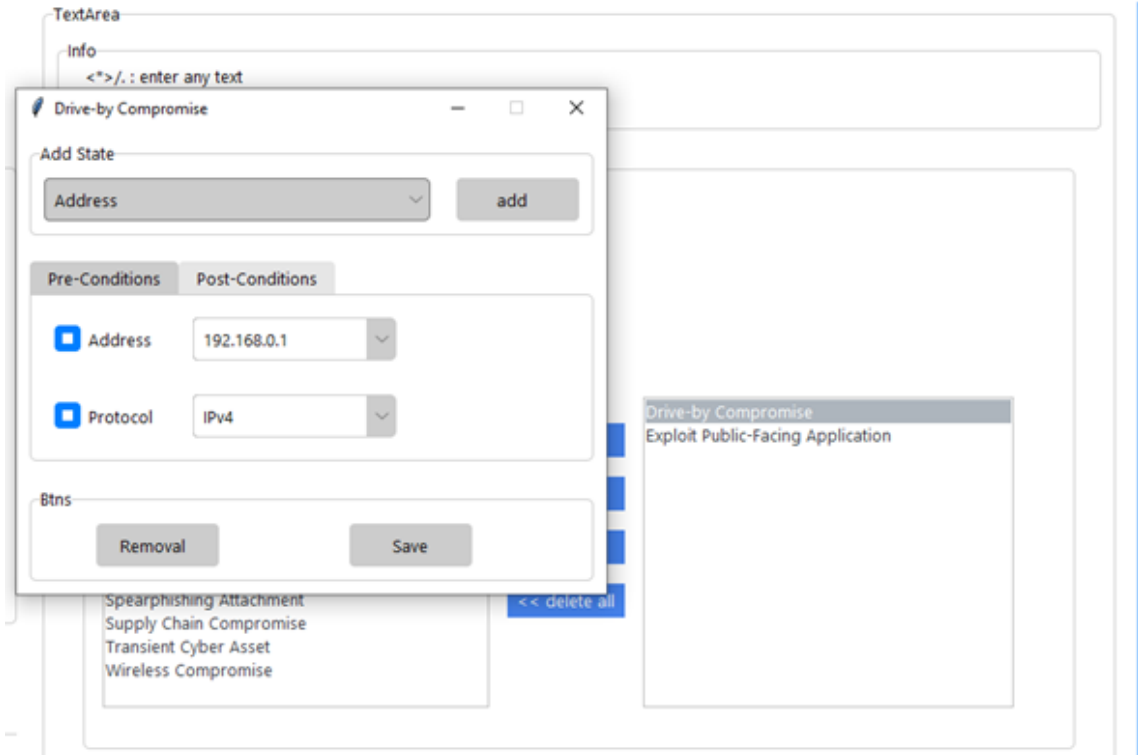


Fig. 5.13 Pre and Post Conditions

security vulnerabilities through ICS techniques to derive diverse attack scenarios. This assists security analysts in controlling the most likely attack scenarios for routine threats by assessing them using known security vulnerabilities.

Our model draws inspiration for state based from many of cyber security systems for example Strips , caldera but differs in creating attack sequences. The majority of system-based tool relies on a brute-force approach to creating sequences of attacks, randomly selecting attack techniques and steps to execute. This allows the tools to generate a wide range of attack sequences, but they may not always be realistic or practical.

On the other hand, the scenario generator relies on a state-based model to create attack sequences. The scenario generator takes into account the current system state when creating attack sequences. This makes it possible for the scenario generator to create attack sequences that are more likely to be successful in the real world.

Our evaluation results show that our model effectively performs attacks and automated defences. While we currently have limitations due to testing a small number of scenarios, we believe that the proposed model can be an effective tool for understanding potential attack paths. Additionally, the model provides graphical interfaces for configuring system state settings and attack techniques, improving the usability for scenario creators. This makes it easy for security analysts to use the scenario generator to create different sequences of attacks.

In general, the state-based architecture represents an improvement over previous state-based solutions for creating attack scenarios. It is more realistic and comprehensive, addressing challenges related to scalability and accuracy.

5.11 Conclusion

This chapter presents the methodology used by the author to generate state-based attack scenarios using the STRIPE inspired model. The model is used to identify security threats and vulnerabilities in a system by considering the system's state. This methodology is based on the assumption that a security breach can occur when an attacker exploits a vulnerability in a system.

The author utilized the attack graph technique during the analysis phase to identify the possible attack paths that an attacker can take to exploit vulnerabilities in a system. Attack graphs are used to model the relationships between the system's components and the possible attack paths that an attacker can take to reach their target.

The results of the analysis phase were used to generate state-based attack scenarios. The author classified these scenarios based on the system's state and the type of attack that an

attacker can execute. The generated scenarios were evaluated based on their likelihood of occurrence, impact, and severity.

Finally, the author developed an interface for the model to make it more accessible to security professionals. The interface provides an easy-to-use platform for generating state-based attack scenarios and evaluating their likelihood of occurrence, impact, and severity. It is hoped that this interface will enable security professionals to identify and mitigate security threats more effectively.

Chapter 6

General Methodology To Create Attack Scenario For ICS

6.1 Introduction

Analysis of cyber incidents using attack scenarios is an important process for understanding, addressing, and preventing cyber threats. Security experts can gain a deeper understanding of incidents by creating attack scenarios that define the attacker's steps and techniques. This chapter discusses the general methodology of attack scenarios and the application of the technique prediction model, group prediction model, threat-based scenario generation model, and state-based scenario generation model. Furthermore, this chapter validates the attack scenario general methodology through using case studies, expert interviews, and literature. This chapter also introduces the usability study for the ASG tool and the results with the help of case studies.

6.2 Attack Scenario Generation Methodology

This study has come up with an outcome of extracting a general methodology from those solutions that security analysts can follow to create attack scenarios. The process begins with the collection of raw data from laboratory documentation and incident reports, followed by the identification of a relevant knowledge base. Subsequently, system-related information is retrieved, encompassing both the system's operational state and the conditions under which the attack occurred. Afterwards, data pre-processing takes place by determining the attack incidents and tactics, identifying descriptive texts that express techniques or systems, or determining the system state log. The inferential analysis is carried out by processing data,

arranging the techniques manually through direct comparison with related attacks, using the Natural Language Processing (NLP) model to link the relationships between techniques according to similarity in terms of usage, and connecting the techniques according to the system state. The following step is to structure attack sequences that represent the scenario usage. The selection process is executed according to previous attack policies using a Machine Learning Model (ML) to connect all techniques like a series of trees and determine the optimal ways to represent a scenario. Followed by linking all the techniques' series consequences, classifying the ideal paths from the scenarios tree based on the usage, and searching for the paths that link the techniques with the system state. The final step is the validation stage, which takes place through a direct comparison between the series of techniques that represent the scenario and the incidents of previous attacks. A classification model is used that determines the percentage of belonging to the scenario as ICS incidents, and determines the APT group that the incident belongs to. The stages of the attack scenario generation methodology are illustrated in Figure 6.1, while a summary is provided in Table 6.1.

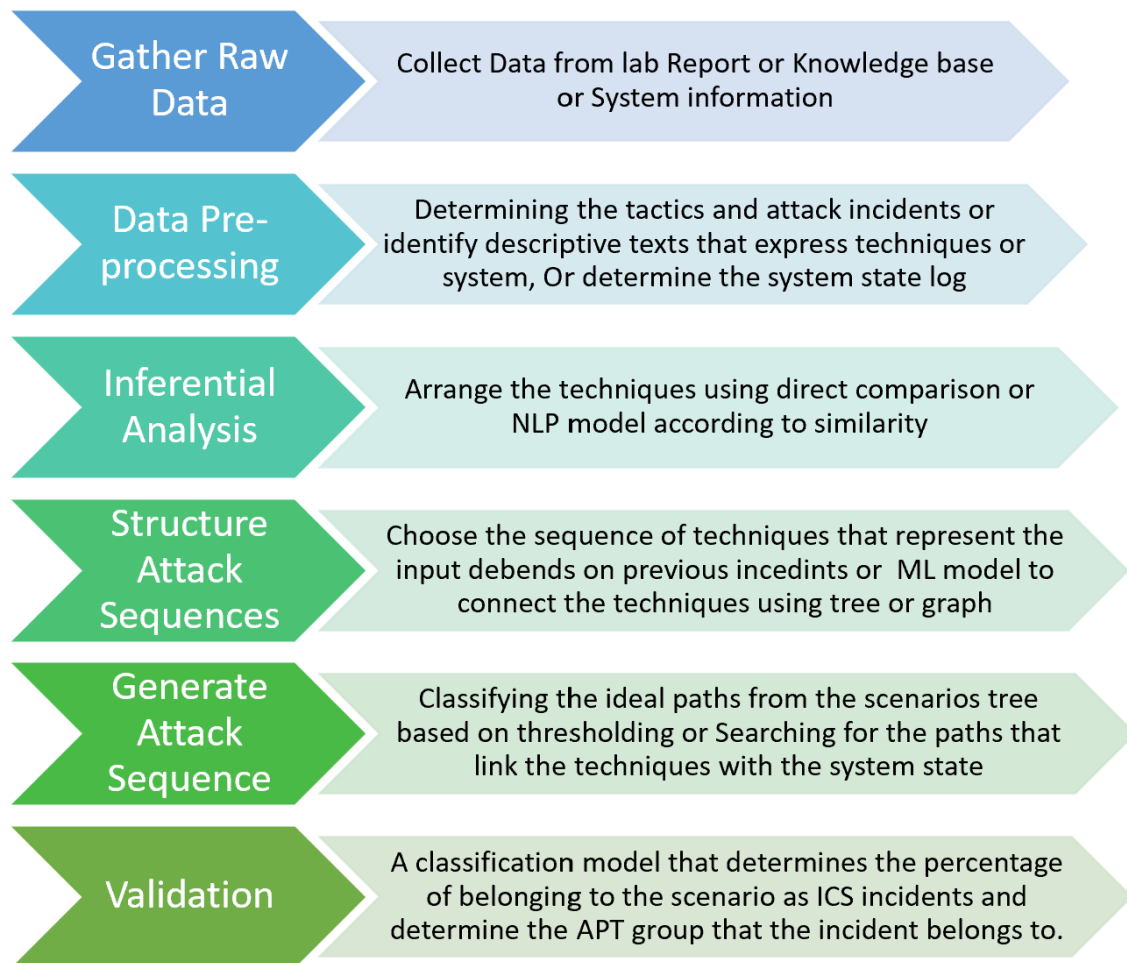


Fig. 6.1 Attack Scenario Generation Methodology

6.2.1 Data Collection Related to Attacks

The data collection is performed for each method as detailed:

- **Manual:** Investigate the reports and Lab documentation.
- **Threat-based:** Find a knowledge base.
- **State-based:** Search for system information. Which includes the system state and the attack conditions.

6.2.2 Data Processing

Data processing for each method is highlighted in the following bullets.

- **Manual:** determining the tactics and attack incidents or identifying.
- **Threat-based:** descriptive texts that express techniques or system.
- **State-based:** determine the the system's state log.

6.2.3 Inferential Analysis

The inferential analysis is carried away as elaborated:

- **Manual:** Arrange the techniques manually through direct comparison with related attacks.
- **Threat-based:** Use NLP model to link the relationships between techniques according to the similarity in terms of usage
- **State-based:** Connecting the techniques according to the system state.

6.2.4 Structure Attack Techniques Sequence

Attack scenarios can be created for each as follows:

- **Manual:** Choose the series of techniques that represent the scenario usage. Selection is according to previous attack policies
- **Threat-based:** Use Model ML to connect all techniques like a series of trees and determine the optimal ways to represent a scenario
- **State-based:** Connecting Techniques That Achieve System State

6.2.5 Generate Attack Scenario

- **Manual:** Mapping the techniques with ICS CKC.
- **Threat-based:** Classifying the ideal paths from the scenarios tree based on the usage.
- **State-based:** Searching for the paths that link the techniques with the system state.

6.2.6 Scenarios Validation

Scenarios can be validated by the following methods as described below.

- **Manual:** Direct comparison between the series of techniques that represent the scenario, and the incidents of previous attacks
- **Threat-based and State-based:** A classification model that determines the percentage of belonging to the scenario as ICS incidents and determines the APT group that the incident belongs to.

Table 6.1 General Methodology for Attack Scenario Generation

	Manual	Threat-based	State-based
Gathering Raw Data	View the reports and lab documentation	Find a knowledge base: ICS ATT&CK	Search for system information: <ul style="list-style-type: none"> • System state • Attacks post and pre-conditions
Data Pre-Processing	determining : <ul style="list-style-type: none"> • Tactics • Attack incidents 	Identify descriptive texts that: <ul style="list-style-type: none"> • Techniques • Tactics • Incidents • Groups APT 	determining: <ul style="list-style-type: none"> • System state log • Techniques
Inferential Analysis	Arrange the techniques manually through direct comparison with related attacks	Use NLP model to link the relationships between techniques according to similarity in terms of usage	Comparison of the starting state with the pre-conditions of the techniques
Structure Attack Sequences	Choose the series of techniques that represent the scenario usage. Selection is according to previous attack policies	Use Model ML to link all techniques, like a sequence of trees and determine the optimal ways to represent a scenario	Link the techniques to the system state to create a mesh graph.
Generate Attack Sequence	Mapping the techniques with ICS CKC manually	Classifying the ideal paths from the scenarios tree based on the usage	Searching for the paths that link the techniques with the system state
Validation	Direct comparison with previous incidents	A classification model that determines the percentage of belonging to the scenario as ICS incidents and determines the APT group that the incident belongs to.	

6.3 ASG Architecture

This section presents an architectural model for ASG. The architecture consists of three essential components.

1. The first component (**a-Threat-based**) creates a sequence of attacks for ICS by receiving textual inputs and analyzing them to classify the technique, incident, and relevant group from the text. The model then creates attack sequences in a tree structure, based on the execution order of ICS tactics. Chapter 4 details the architecture in depth.
2. The second component (**b-State-based**) is explained in Chapter 5, where it generates automated attack sequences based on the system's state and attack techniques. It allows security analysts to control attack sequences, input system conditions, and select attack techniques. The model then generates scenarios in a mesh graph form.
3. The final component (**c-Check scenarios**) is detailed in Chapters 4 and 5. It is used for validating the generated attack scenarios according to models (a, b) by comparing the proportion of technique sequences representing the scenario with previous actual ICS incidents (See 5.1).

6.4 Correctness of Data Generated

The validity of the text data that was generated to verify the performance of the scenario model was evaluated. The BLEU (Bilingual Evaluation Understudy)[106] scale was incorporated, which is used in machine learning techniques to compare texts, determine the percentage of similarity, and prevent excessive agreement between the generated text and the basic text. The model had a coefficient value of 0.8, which shows that the similarity of the generated data to the original data is 80%.

6.5 Evaluation of the Proposed Methodology for Generating Attack Scenarios

To assess the effectiveness and distinctiveness of the proposed methodology for generating attack scenarios, it is essential to situate it within the domain of existing threat modeling and simulation approaches. This section presents a comparative analysis of prominent methodologies, highlighting their strengths, limitations, and operational requirements. Through this

comparative approach, the proposed methodology is evaluated across key dimensions, with particular focus on its suitability for complex cyber-physical systems such as ICS.

- Theoretical Background

Various methodologies underpin threat modeling and analysis. Attack trees rely on expert intuition. The Kill Chain model leverages information from each attack stage. MITRE ATT&CK is grounded in comprehensive threat intelligence and real-world attack observations. Adversary emulation and red teaming necessitate detailed threat actor tactics, techniques, procedures, and in-depth environmental understanding. Simulation-based approaches depend on accurate input data for effective modelling. The proposed methodology explains and illustrates where it stands compared to the previous methodologies. The first is the cyber kill chain, a foundational model in cybersecurity that outlines the distinct stages attackers follow during an intrusion. As for ATT&CK matrix, it provides a reliable source for institutions to develop reality-based attack scenarios depending on real-world adversary procedures and tactics. This enables security teams to better align their defenses with the actual threats they might face. Attack trees offer benefits such as visualizing attack pathways, identifying vulnerabilities, and assessing attack complexity. Concerning red teaming, it offers valuable insights into an organization's security posture by simulating real-world attacks and exposing vulnerabilities and weak points. Adversary emulation provides significant advantages in enhancing cybersecurity posture by simulating real-world attack scenarios. It effectively identifies vulnerabilities and weaknesses within an organization's defenses, enabling proactive mitigation strategies. Simulation-based approaches use models to simulate attack scenarios in a controlled environment. This can involve software tools that mimic network conditions and attack behaviours. The approaches are compared based on 7 main criteria [107]: complexity, realism, scalability, applicability, ease of use, data requirements and adaptability. These pillars are represented in the table below:

Based on the table, simulations are complex tools requiring substantial technical expertise to develop, implement, and interpret. While their realism is contingent on the accuracy of underlying models, they offer high scalability, allowing for diverse scenarios and adaptable configurations. Their broad applicability spans training, testing, and research domains. However, achieving accurate simulations necessitates significant data input. Despite these complexities, the potential to explore various conditions makes them invaluable assets in numerous fields

Table 6.2 Comparison of Methodologies Across Key Metrics

Methodology	Complexity	Realism	Scalability	Applicability	Ease of Use	Data Requirements	Adaptability
Attack Trees	Moderate	High	Moderate	High	Moderate	Low to moderate	High
Kill Chain Analysis	Low to moderate	Moderate	High	High	High	Moderate	Moderate
MITRE ATT&CK	High	High	High	Very high	Moderate	High	Very high
Adversary Emulation	High	Very high	Moderate	High	Low	High	High
Red Teaming	Very high	Very high	Low to moderate	Very high	Low	High	High
Simulation-Based	High	Moderate to high	High	High	Moderate	High	High

- ASG General Methodology

As mentioned earlier, each method related to attack scenario generation and evaluation has its advantages and disadvantages. The following section illustrates the common methodologies compared to the proposed methodology for attack scenario generation based on: data collection, data processing, inferential analysis, Structure Attack Techniques Sequence, Attack Scenario generation, and scenario validation. The comparison is represented in a table 6.3 to summarize the main criteria.

- The Proposed Methodology Features Compared to Other Methodologies

This section discusses where our methodology stands compared to other methodologies based on specific criteria. First of all, comprehensive data collection; the inclusion of manual, threat-based, and state-based methods ensures a wide range of data sources, enhancing the depth and breadth of the collected information. Next is advanced data processing, which tailors the processing approach to the data type and ensures precise and relevant data interpretation. After that, a sophisticated inferential analysis is carried out, where the use of NLP and system state connectivity provides a dynamic and precise analysis, enabling a more accurate understanding of attack relationships. In addition, adaptive structuring techniques

are considered, where machine learning and system state connections allow for a flexible and responsive structuring of attack scenarios. In the innovative scenario generation, the methodology's focus on classifying paths and linking them with system states results in realistic and context-specific attack scenarios. Finally, the quantitative validation makes use of classification models to validate and ensure a high degree of precision and reliability. The methodology employs a multifaceted approach to data collection, which is a strength as it ensures a comprehensive dataset. This is comparable to MITRE ATT&CK and adversary emulation, which also rely on diverse and detailed data sources.

The proposed methodology for generating attack scenarios shows significant strengths in terms of comprehensive data collection, advanced data processing, sophisticated inferential analysis, adaptive structuring of techniques, innovative scenario generation, and quantitative validation. When compared to traditional methodologies like Attack Trees, Kill Chain Analysis, MITRE ATT&CK, Adversary Emulation, Red Teaming, and Simulation-Based methods, the proposed approach provides a more advanced, flexible, and context-sensitive framework. This makes it particularly well-suited for complex environments such as ICS, ensuring realistic, dynamic, and highly relevant attack scenarios.

6.6 Use Cases

In the use cases, potential scenarios are provided in the form of case studies to get a better understanding of what can be expected and how the system's reaction will be. The use cases are diverse and associated with different organizations or companies.

6.6.1 Use Case 1: Adversary Emulation

Considering an imaginary Shark Oil Company, with almost 3000 employees in its branches globally. Due to the lack of punctuality and proper time management by some of the staff members, the company decided to monitor them closely and warn them. Furthermore, the company deducted part of their salaries last year in February, for some reason. There is an issue with the mentioned scenario that some of the monitored staff members are working in the IT department and have access to the company CHAT API, and have access to user credentials. The cybersecurity team needs to be proactive in order not to face any incident that could happen from malware injections to their chat API or by using user access control.

Therefore, Shark Company makes use of ASG (Attack Scenario Generator) to be able to detect and predict attack scenarios to create adversary emulation for testing and verifying defences against the expected common adversary techniques. The techniques may be

Table 6.3 Comparison of Methodologies for Analyzing Attack Scenarios

Criteria	Proposed Methodology	Kill Chain Analysis	MITRE ATT&CK	Adversary Emulation	Simulation-Based
Data Collection	Manual: Reports and documentation. Threat-based: Knowledge bases. State-based: System state and attack conditions.	Data from attack lifecycle phases.	Threat intelligence and real data.	Threat actor TTPs.	Accurate simulation data.
Data Processing	Manual: Identify tactics. Threat-based: Descriptive texts. State-based: Analyze system logs.	Categorize by phases.	Categorize TTPs.	Emulate attack behaviors.	Model and simulate attacks.
Inferential Analysis	Manual: Compare techniques. Threat-based: Use NLP to link techniques. State-based: Connect techniques to states.	Analyze phase sequences.	Link TTPs with intelligence.	Direct adversary emulation.	Simulate for inference.
Structure Techniques	Manual: Select techniques based on policies. Threat-based: Use ML for tree and graph. State-based: Link techniques achieving goals.	Structure by phases.	Structure by TTPs.	Mimic threat actors.	Structure in simulations.
Generate Scenarios	Manual: Map techniques with ICS CKC. Threat-based: Classify paths. State-based: Link techniques to states.	Develop from lifecycle phases.	Create scenarios from TTP links.	Mimic adversary TTPs.	Generate in simulations.
Scenario Validation	Manual: Compare with past incidents. Threat/State-based: Classify and identify APT groups.	Validate lifecycle phases.	Validate techniques with real data.	Match adversary behaviors.	Validate in simulations.

inputting the 'have access to API' or 'gaining user credentials' to generate possible attack scenarios.

- Results

- The security analyst needs to identify potential attacks by an employee working on the messaging system API within Company Shark Oil, to cause harm to the system by executing malicious programs, as illustrated in Figure 6.2, 6.3 and 6.4.
- The security analyst identifies potential attacks for an employee working on the account management interface, where the employee has specific permissions that do not allow them to create administrative accounts. How can the employee access the system and

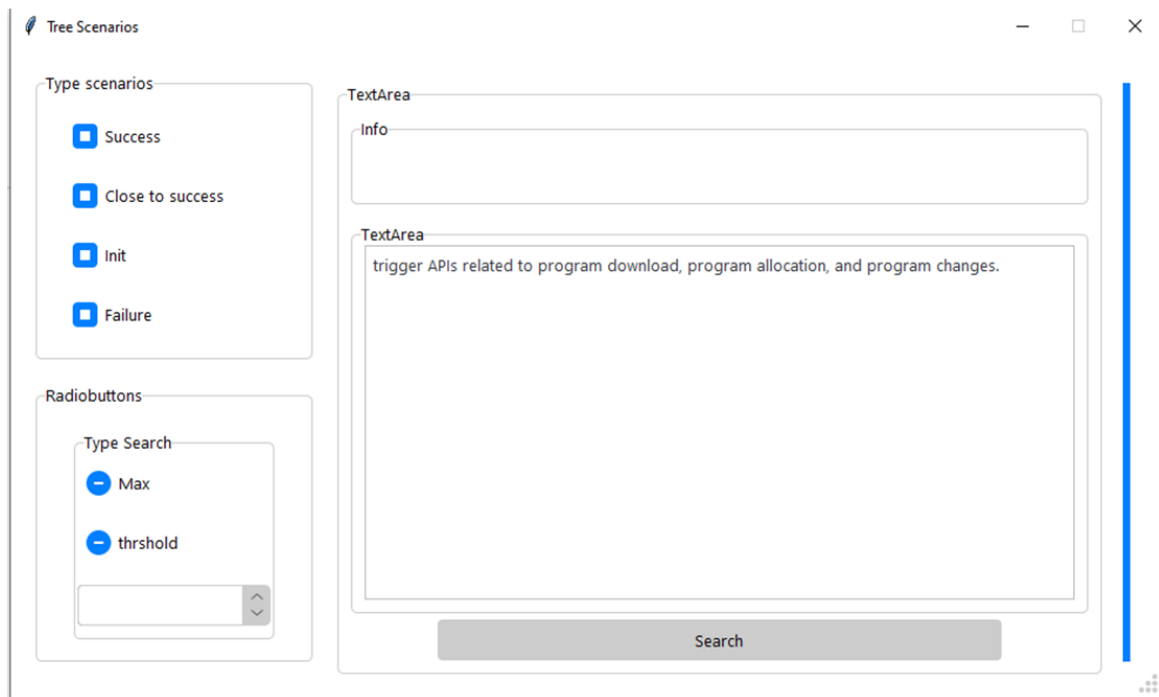


Fig. 6.2 Adversary Emulation Use case 1.a

create an administrative account capable of modifying their privileges? As illustrated in Figure 6.5, 6.6 and 6.7

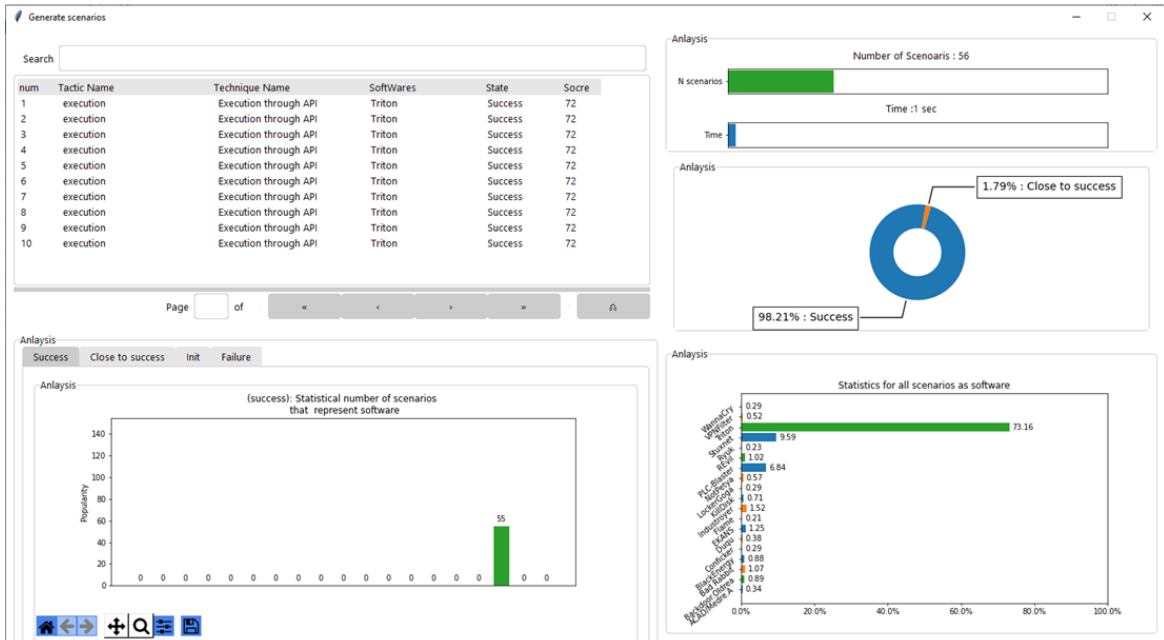


Fig. 6.3 Adversary Emulation Use case 1.b

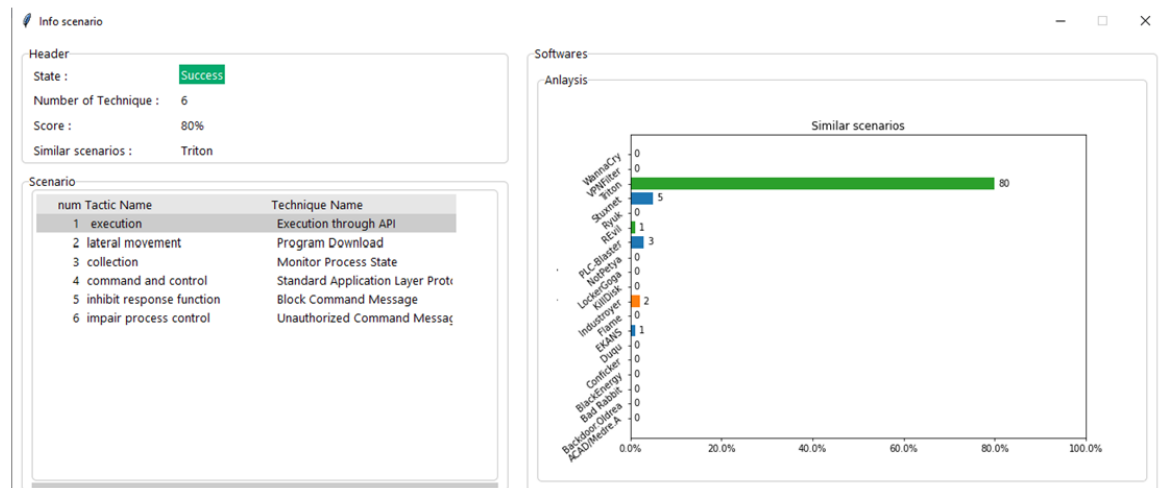


Fig. 6.4 Adversary Emulation Use case 1.c

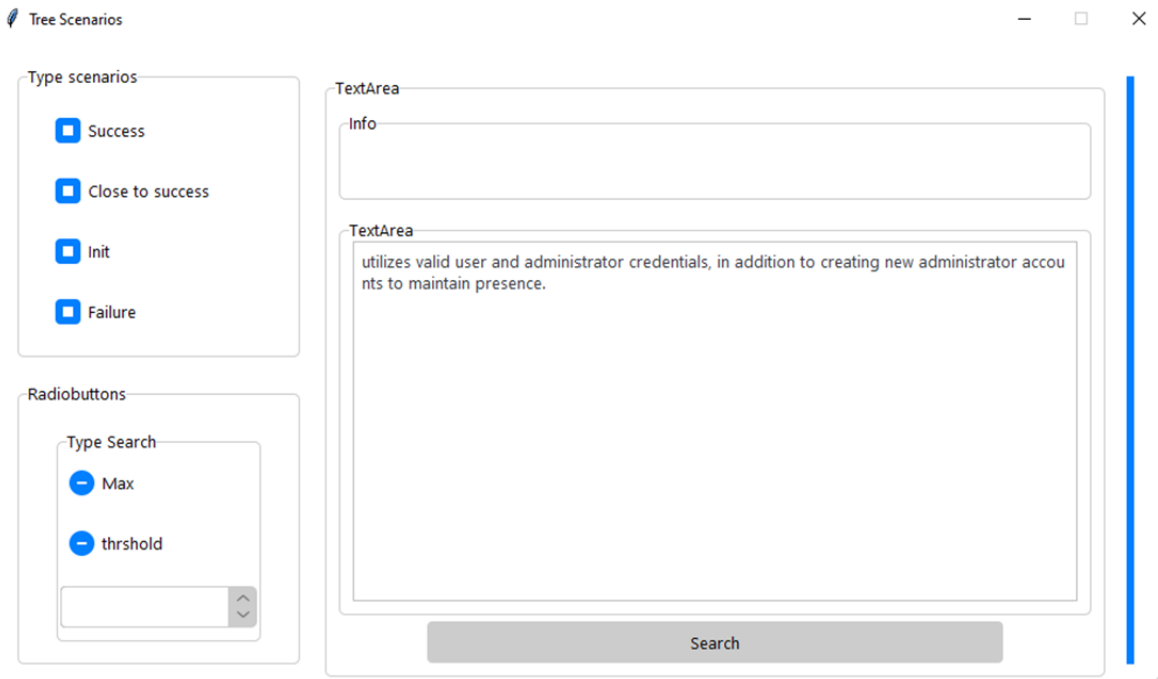


Fig. 6.5 Adversary Emulation Use Case 2.a

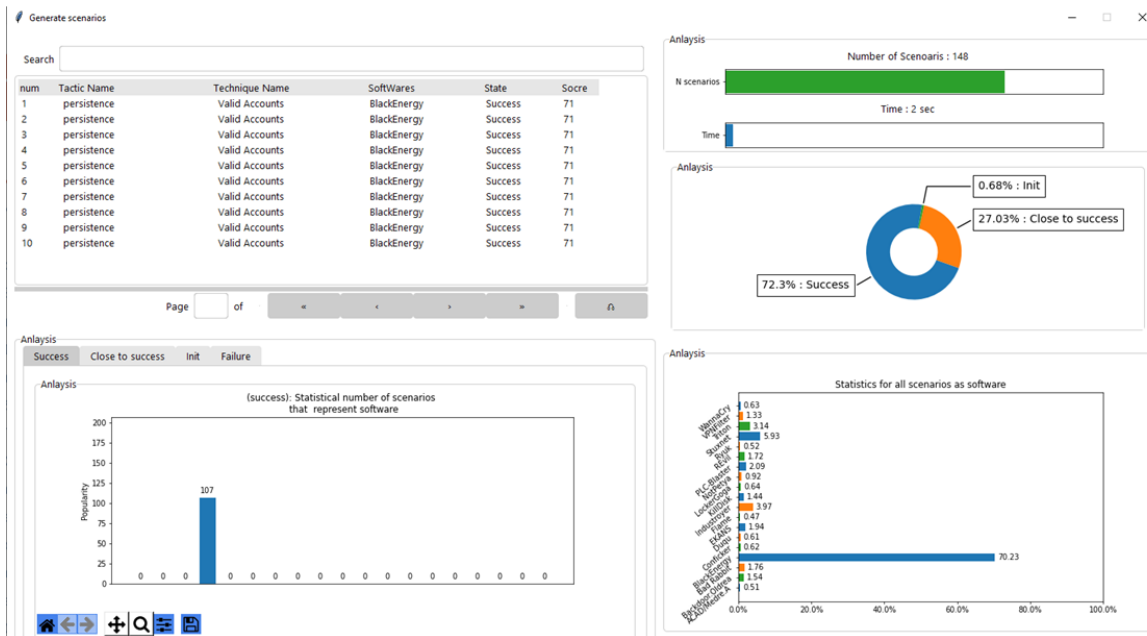


Fig. 6.6 Adversary Emulation Use Case 2.b

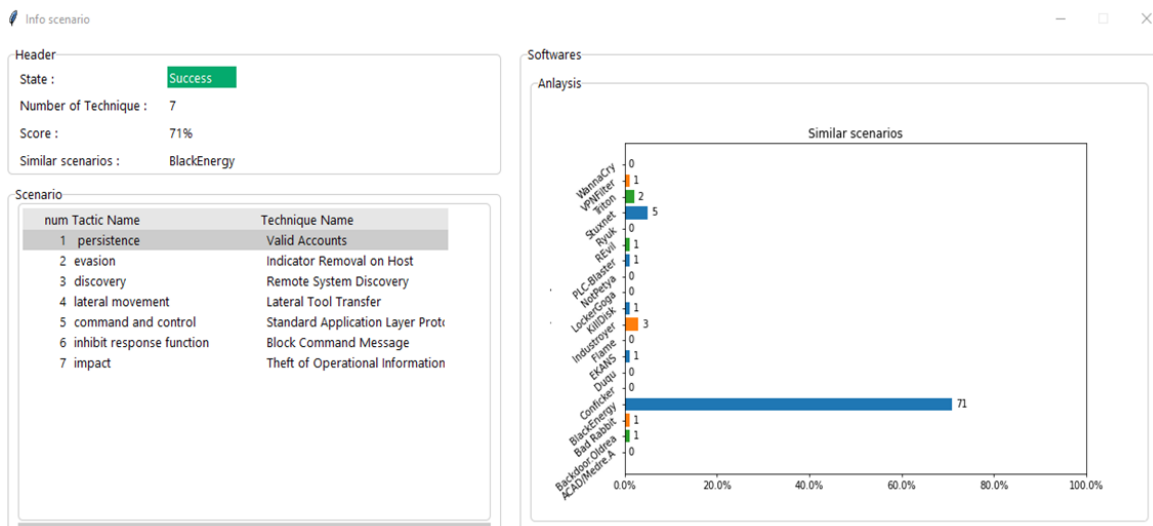


Fig. 6.7 Adversary Emulation Use Case 2.c

6.6.2 Use Case2: Cyber Threat Intelligence Enrichment

In this case, in 2023, Fossilia Oil company wanted to get the reports related to the Shamoon 2013 and ZeroCleare 2019 attacks to investigate the techniques used by the attackers. Their motive for this investigation is to create an effective map defence. ASG provides an automated live experience for the attack scenarios, based on the previous attacks for other companies, to assist analysts and defenders in devising an effective map defence.

- Results

Fossilia Oil's cybersecurity team is actively investigating ICS-related incidents believed to be caused by Shamoon malware. This insidious software has been designed to circumvent User Account Control (UAC) remote control restrictions by manipulating the system's registry. Once it gains access to the network share, it activates the remote registry service on the target system, allowing it to connect and modify the registry settings. By doing so, it disables the UAC restrictions through configuration settings. Scenarios are depicted in Figures 6.8, 6.9 and 6.10.

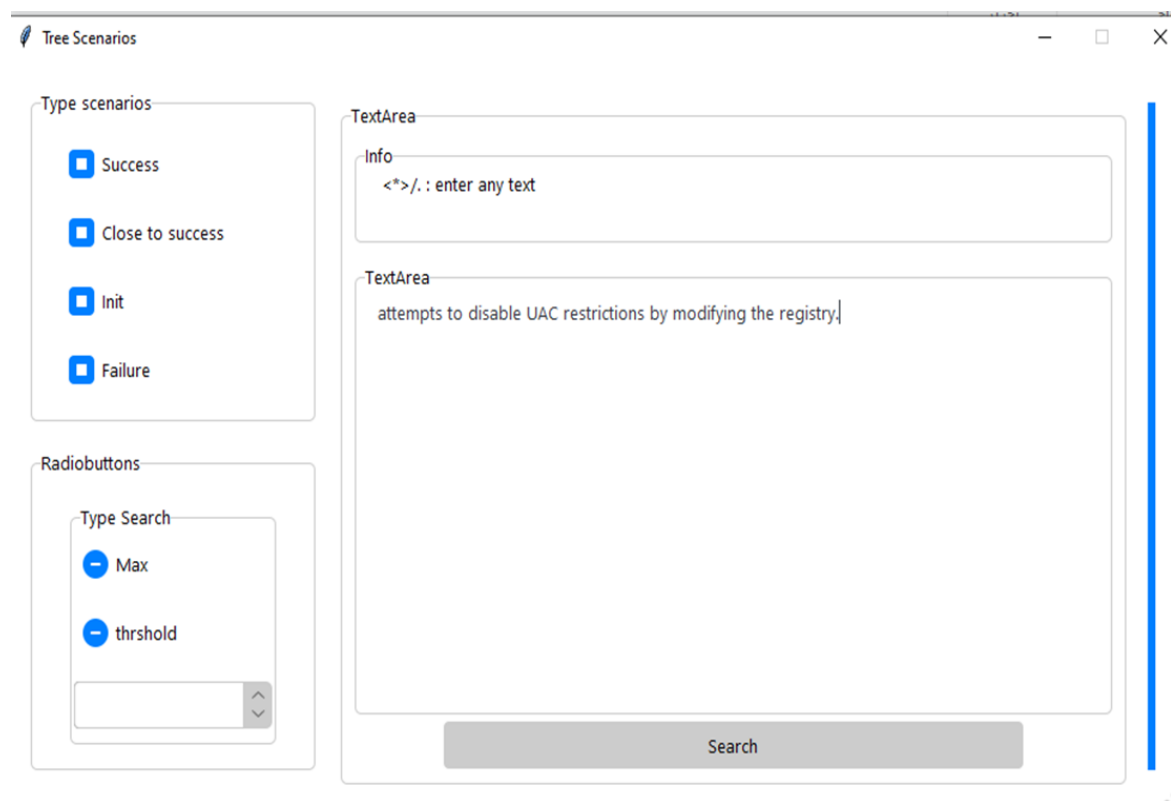


Fig. 6.8 Cyber Threat Intelligence Enrichment Use Case 1.a

6.6.3 Use Case3: Red Teaming

Aero-crafts company *Red Team* needs to test the complexity of the current modern system. They keep on attacking the system easily. Unfortunately, there are not enough provided scenarios or complex tests to be performed in order to better train for future defence. To achieve that, the company decides to consult the ASG to create red team plans and organize operations to avoid certain defensive measures that may be in place within a network. Through the tool, Red Team used it as a research roadmap to develop new ways of performing actions that may not be detected by common defences as shown in Figures 6.11 and 6.12.

The screenshot displays the 'Tree Scenarios' application window. On the left, there are two main sections: 'Type scenarios' and 'Radiobuttons'. The 'Type scenarios' section contains four checkboxes: 'Success', 'Close to success', 'Init', and 'Failure', all of which are checked. The 'Radiobuttons' section contains a 'Type Search' dropdown menu with 'Max' and 'threshold' options, where 'threshold' is selected. Below this is a numeric input field containing the value '1'. The main area of the application contains two 'TextArea' components. The top 'TextArea' is labeled 'Info' and contains the text '<*>/. : enter any text'. The bottom 'TextArea' is labeled 'TextArea' and contains the text 'access api'. Below the text areas is an 'Automatic language recognition' section with a checked checkbox for 'Translation'. At the bottom center, there is a 'Search' button.

Fig. 6.11 Red Team Use Case 1.a

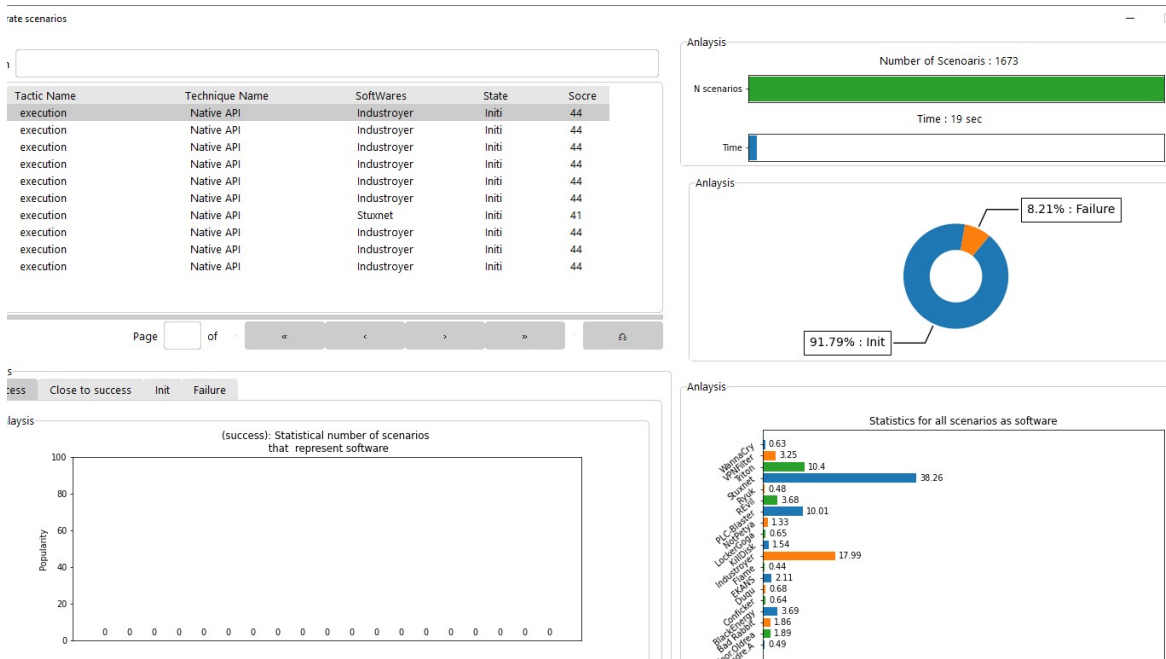


Fig. 6.12 Red Team Use Case 1.b

6.7 Valiadation of the General Methodology Using Expert Interviews

To obtain constructive and unbiased feedback, insights were gathered from domain experts to proactively address potential issues that might be identified. The validation, as previously mentioned, is conducted through 3 main steps: literature review and comparing our methodology with the other methodologies, trying to extract the point of strength and points of consideration. The second is through the case studies and how the methodology is applied, and finally, the experts’ interviews to validate the proposed methodology by the triangulation technique.

6.8 Experts Interviews

To gather in-depth insights about the general methodology, the study employed a semi-structured interview format, conducted via Zoom. This platform assisted remote human participants, ensuring accessibility and convenience. During the interviews, participants shared their perspectives and experiences. This approach allowed for flexibility in following up on emergent themes while maintaining consistency across the interviews. The interviewees

were asked to use manual, state-based, or threat-based methodologies to create attack scenario generation.

6.8.1 Using the Methodology to Generate Attack Scenarios

In total, five interviews were conducted and their feedbacks were noted. The interviewee's introduction and their feedback are compiled below.

- First Interviewee

The interviewee is the CEO of a future platform company who employed a manual-based methodology to analyze IsaacWiper, a destructive malware targeting Ukrainian organizations during the 2022 Russia-Ukraine war.

- Interviewees Feedback

The interviewee highlighted several challenges encountered during the analysis, including the difficulty in identifying specific attack techniques, particularly when relying solely on technical laboratory reports, and the lack of in-house expertise. The interviewee also expressed concerns regarding the methodology's unclear validation process and its reliance on a general Data Wipe Mechanism (DWM) extracted by the thesis author. While acknowledging the methodology's utility in structuring the attack stages, the interviewee indicated a preference for using it as a supplementary tool rather than as the primary analysis framework. Ultimately, the application of this manual methodology resulted in the generation of an IsaacWiper Hermetic Wizard attack scenario, detailed in the appended laboratory reports.

- Second Interviewee

The second interviewee was from the Saudi CTI society and was a CTI analyst, participated in generating attack scenarios. This analyst utilized a threat intelligence-based tool to extract scenarios.

- Interviewees Feedback

The interviewee mentioned that the methodology's general applicability to various platforms, including enterprise and mobile systems, was acknowledged, although the initial model was primarily based on ICS data. The automation of threat intelligence, particularly with respect to MITRE ATT&CK, was highlighted as a significant potential time-saver. The analyst noted that the tool's automated results were comparable to manual analysis using MITRE

Table 6.4 Interviewer result using manual methodology

Attack Steps	General DWM	IsaacWiper
Stage 1		
<i>Reconnaissance</i>	Workstation vulnerability	Workstation vulnerability
<i>Weaponization</i>	Compound document; Watering Hole	Compound document
<i>Targeting</i>	Bypass security controls	Bypass user account controls
<i>Delivery</i>	Phishing; Process memory injection	Stealth
<i>Exploit</i>	Hidden executable	Hidden executable
<i>Install/Modify</i>	Fake rawdisk driver	Bypass signature
<i>C&C</i>	Remote system control	Application layer (HTTP); Remote system discovery; Remote file copy
<i>Act</i>	Data destruction; Disk structure wipe	Wipe data, relying on overwriting data rather than encrypting or manipulating the Master Boot Record (MBR)
Stage 2		
<i>Develop</i>	New service; Scripting	Scripting
<i>Test</i>	Service execution; Execute script	Service execution
<i>Deliver</i>	Windows admin share; Network config exploration; Valid accounts	Network config exploration
<i>Install/Modify</i>	Scheduled ICS tasks	Scheduled ICS tasks
<i>Execute</i>	Takeover ICS controller	Takeover SCADA

ATT&CK, suggesting a strong reliance on this framework. While the analyst attempted to validate results using state-based methodologies, they were deemed unsuitable for complex attacks like ICS incidents. Ultimately, the analyst validated the results using their prior knowledge and expressed enthusiasm for integrating the methodology and tool into their

team's practices. The analyst recommended clarifying the methodology's specific role within the Security Operations Center (SOC) analysis to facilitate its adoption.

- Third Interviewee

The interviewee and a cybersecurity analyst employed a manual automation methodology to analyze Fuxnet 2024, a novel Industrial Control System (ICS) malware utilized by the Ukrainian hacking group Blackjack to target Russian infrastructure in June 2024. This malware, specifically designed to compromise sensor gateways, poses a significant threat to industrial systems, potentially leading to widespread disruption and operational failures. Notably, Fuxnet 2024 represents an emerging threat with limited public documentation and analysis available at the time of the interview.

Table 6.5 Interviewer result using manual methodology

Attack Steps	fuxnet
Stage 1	
<i>Reconnaissance</i>	External Remote Services
<i>Weaponization</i>	Internet Accessible Device
<i>Targeting</i>	Bypass User Account Controls
<i>Delivery</i>	unknown
<i>Exploit</i>	Hidden Executable
<i>Install/Modify</i>	Bypass Signature
<i>C&C</i>	common used port;
<i>Act</i>	Data destruction;
Stage 2	
<i>Develop</i>	unknown
<i>Test</i>	unknown
<i>Deliver</i>	Network Config Exploration
<i>Install/Modify</i>	Brute Force I/O
<i>Execute</i>	Loss of View

- Fourth Interviewee

The interviewer, a cybersecurity specialist at STC, employed an automation-based threat modeling methodology to analyze a resource exhaustion attack on ICS devices over the network using the T0833 (Network Denial of Service) technique.

- Interviewee Feedback

The interviewee illustrated that this approach proved highly advantageous due to its automated nature, significantly reducing time consumption. The matrix generated by the methodology facilitated its practical application in the interviewer's work. The interviewer emphasized the tool's unique value proposition, differentiating it from other tools in the market. The automated generation of attack scenarios for the Network Denial of Service technique was a key feature, and the model showed its performance in generating nearly 4 scenarios, including actual scenarios as shown in Figure 6.13. It saves time and serves a diverse audience, including those with no technical background.

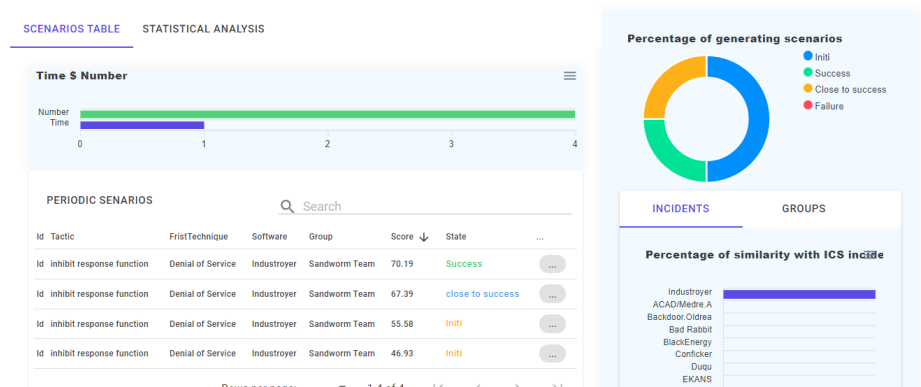


Fig. 6.13 Resource Exhaustion Attack on ICS devices over the network

- Fifth Interviewee

A PhD researcher at Newcastle University was interviewed and used state-based threat modelling methodology to analyse man-in-the-middle (MITM) attacks, through injection attacks, where false data is injected into the signal to influence the system to behave in unexpected ways.

- Interviewee Feedback

The Interviewee demonstrated that this approach did not show any results due to the lack of state precondition and postcondition for each technique, which has been mentioned in

Chapter 5 as a limitation for the state-based approach. The respondent recommended that the methodology be considered as a starting point for creating a comprehensive dataset that links the pre- and post-conditions for each technique according to the MITRE ATT&CK framework. These data can improve the accuracy of the analysis and improve the depth of understanding of the attack techniques, since the results of the generation of attack scenarios showed attacks close to the actual ones, as shown in Figure 6.14. The PhD scholar indicated

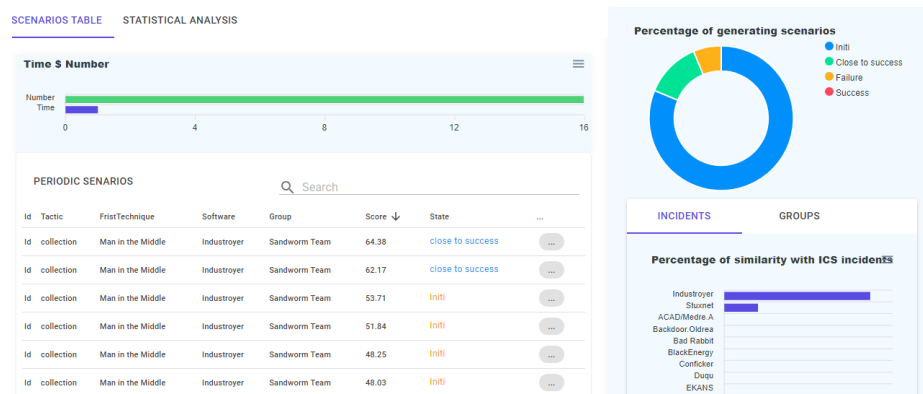


Fig. 6.14 man-in-the-middle (MITM) attacks

that applying the methodology to real systems instead of simulated environments may facilitate obtaining accurate data and support the creation of large databases for analyzing similar systems.

Despite the difficulty in applying the methodology, the respondent emphasized its importance as a promising first step, especially in the field of automation, to link state information to attack techniques. He also considered it a future research area of interest for researchers in the field of cybersecurity of ICS.

6.8.2 Participant Background

In order to maintain the best outcomes and ensure the quality of the undertaken research, a usability study was conducted following an approach of both quantitative and qualitative methodology, based on a questionnaire and an interview. An estimated 20 experts took part in the process, working in multinational companies in the fields of cybersecurity and IoT. They were also consulted to test and validate the tool holistically.

6.8.3 Usability Study Validation

An online session was conducted by the author itself. During the online session, there was a demonstration of the general methodology. This demonstration was presented to 20

participants in the form of a video via Zoom. It took place right after finishing the Attack Scenario Generation tool. The participants of the study were asked several questions. The nature of the questionnaire was anonymous to get authentic feedback right after watching the demonstration.

6.8.4 Results

The results of the questionnaire illustrate the following; 80% of the respondents confirmed that the ASG tool is effective and beneficial for their organizations, specifically, and for cybersecurity generally. They also agreed that the tool will assist the organizations in many ways, such as planning for penetration testing and red teaming, risk assessments, and vulnerability assessments.

The remaining 20% considered the tool a great effort. However, it needs more features like showing threat feeds and any historical numbers for the threats. They also mentioned that they are not fully aware of the tool's capabilities. Also, a few respondents mentioned that it would be of great benefit if languages other than English were added. One respondent added that the tool doesn't recognize synonyms, for example, "social engineering" or "hacker", as it deals with more technical lexical items. Finally, an expert mentioned that the tool lacks an integration plan.

6.8.5 Response to the Insights

After getting expert feedback, some corrections or updates were made to the tool.

- Firstly, concerning supporting different languages, the issue was solved using Google Translate API, which supports various languages, including Arabic.
- Secondly, to address synonyms, the solution was achieved by adding the spaCy database, which is an open-source tool for natural language processing
- As for the integration plan, it will be considered in future research, since more information about cybersecurity systems organizations is needed.

It may be noted that, the more population group is asked for the feedback, the more suggestions can be collected. There might be some suggestions from non-experts that can jeopardize the security. Therefore, not all suggestions can be accommodated. The integration may also face issues and will need tough scrutiny by other platforms.

6.9 ASG Graphical User Interface(GUI)

Graphical interfaces were developed for use with ASG scenario generation models. Using the Thinker package, a Python library that provides tools for desktop GUI development, the model is integrated with interfaces to facilitate user-friendly access to scenario generation models. Handling these models might be challenging for non-technical users who might struggle to understand model parameters or interpret results. To make scenario generation models more user-friendly, graphical interfaces are developed. These interfaces allow users to specify model parameters and monitor results through a simple GUI. For instance, the interface might include a drop-down menu for selecting the type of scenario to generate and a slider to adjust the level of realism. The interface might also display the generated scenarios in a viewing window.

The main interface of ASG presents five fundamental tabs for different scenario generation models (Techniques, incidents, Groups, and Scenarios, according to either a tree-like or mesh graph structure). Some of the GUI components are displayed in Figure 6.15 and 6.16.

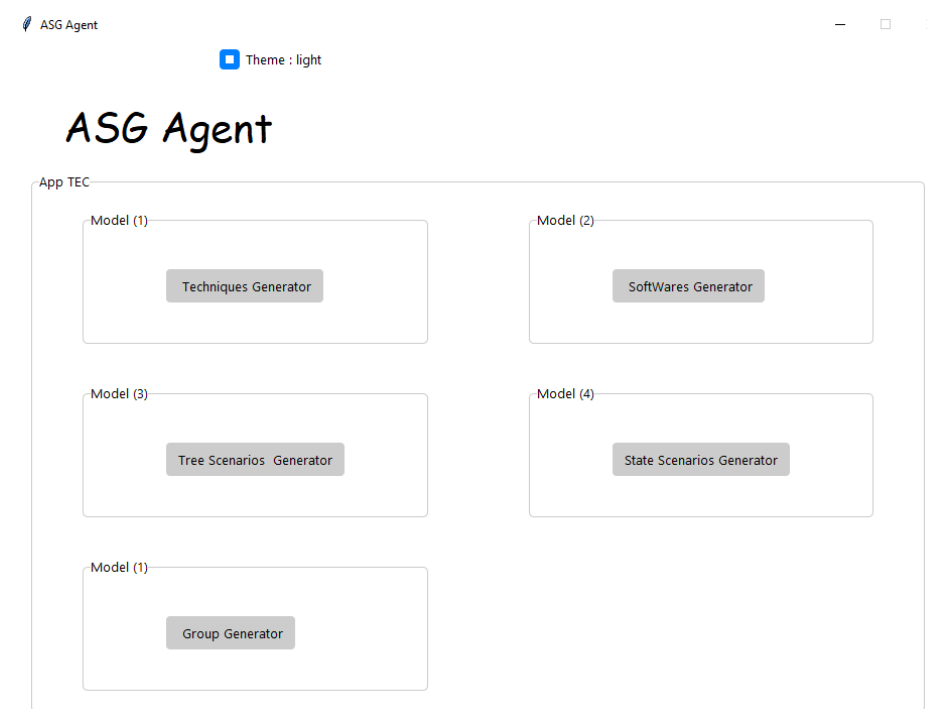


Fig. 6.15 ASG Graphical User Interface 1

The Figures 6.15 and 6.16 illustrate an interface for interacting with the Technique Prediction Model. This interface enables users to input descriptions of techniques or several techniques serving the same purpose. It also allows for the description of ICS incidents associated with the technique and for predicting the most suitable technique. The prediction

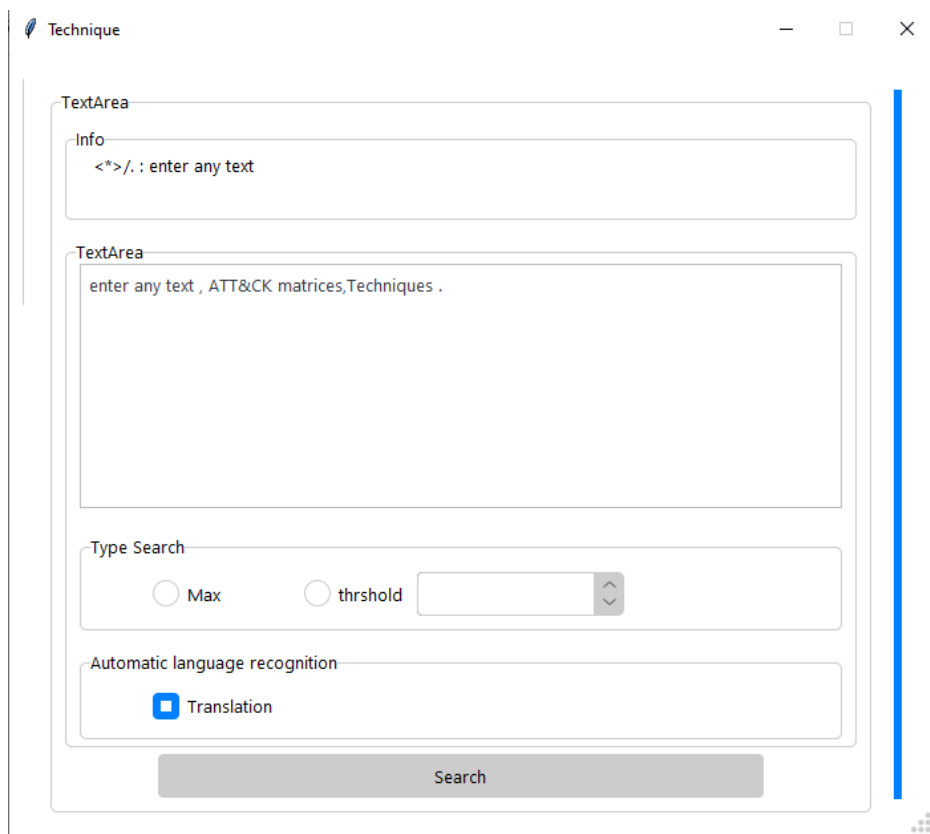
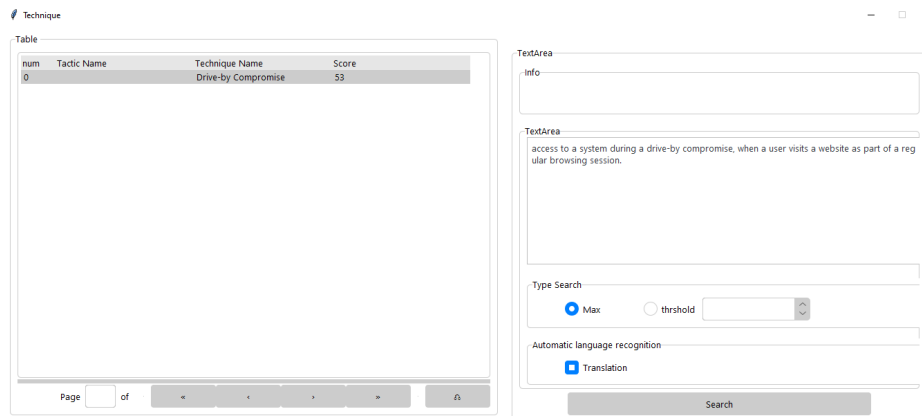


Fig. 6.16 ASG Graphical User Interface 2

model can be based on a Type Search, which presents two approaches: first, by finding the closest technique similar to the input nature, and secondly, relying on the probability of predicting techniques. The model can also translate input text using Automatic Language Recognition, such as the Google Translate API for text classification.

Figure (6.17) demonstrates an example of using the Technique Prediction Model. In this example, a description related to a technique is entered, and upon clicking the search button, the model displays a table of techniques along with their "Score," indicating the likelihood of accuracy. In this example, the model predicts that the technique "Drive Compromise" is the most suitable technique. It has a "Score" value of 53, which means that it is likely to be the correct technique.

In Figure (6.18), an interface for interacting with the incident prediction Model is shown. Users can input incident descriptions or several incidents, or even descriptions of techniques serving the same purpose. The model then predicts the most suitable ICS incident. Similar to the technique model, this model also allows for the prediction method and translation of the input.



The screenshot displays a web interface for the Technique Prediction Model. On the left, a table titled "Table" shows the following data:

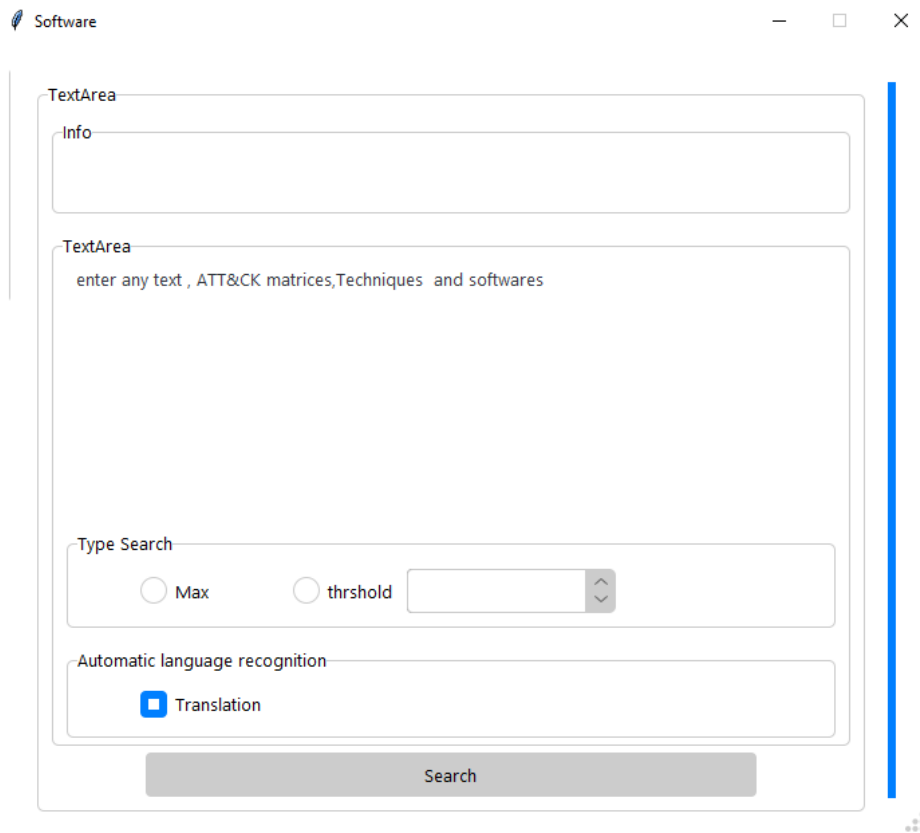
num	Tactic Name	Technique Name	Score
0		Drive-by Compromise	53

Below the table is a pagination control showing "Page 1 of 1" and navigation buttons. On the right, a "TextArea" form is visible, containing an "Info" field, a "Type Search" section with radio buttons for "Max" (selected) and "threshold", and an "Automatic language recognition" section with a checked "Translation" option. A "Search" button is located at the bottom of the form.

Fig. 6.17 Example of Using the Technique Prediction Model

Figure (6.19) provides an example of using the incident Prediction Model, demonstrating the input of an incident description and the subsequent display of a table with incidents along with their corresponding scores. In this example, the model predicts that the incident "Bad Rabbit" is the most suitable event. It has a "Score" value of 83, which means that it is likely to be the correct incident, as shown in 6.19.

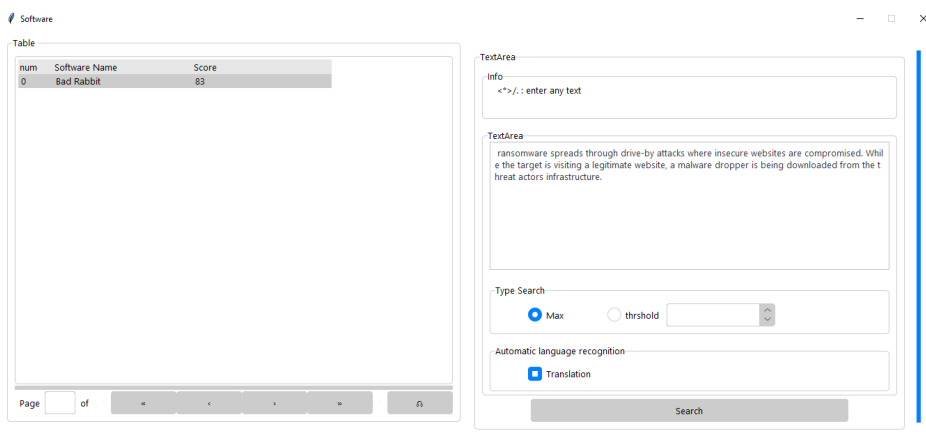
In Figure (6.20), an interface for interacting with the Group Prediction Model is displayed. Users can input descriptions of techniques, incidents, groups, or even descriptions of groups serving the same purpose. The model then predicts the most suitable group. Similar to the other models, this one also includes the prediction method of the group and translation capabilities for input. Figure (6.21) showcases an example of using the Group Prediction Model, where a group description is input, and the model displays a table of groups along with their scores. In this example, the model predicts that the "Sandworm Team" group is the most suitable. It has a "Score" value of 87, which means that it is likely to be the correct group.



The screenshot shows a web application window titled "Software". It contains several input fields and controls:

- A "TextArea" labeled "Info" with a placeholder "Info".
- A larger "TextArea" with the placeholder text "enter any text , ATT&CK matrices, Techniques and softwares".
- A "Type Search" section with two radio buttons: "Max" (selected) and "thrshold". Next to "thrshold" is a small input field with a dropdown arrow.
- An "Automatic language recognition" section with a checked radio button for "Translation".
- A large grey "Search" button at the bottom.

Fig. 6.18 Interface for Incident Prediction Model

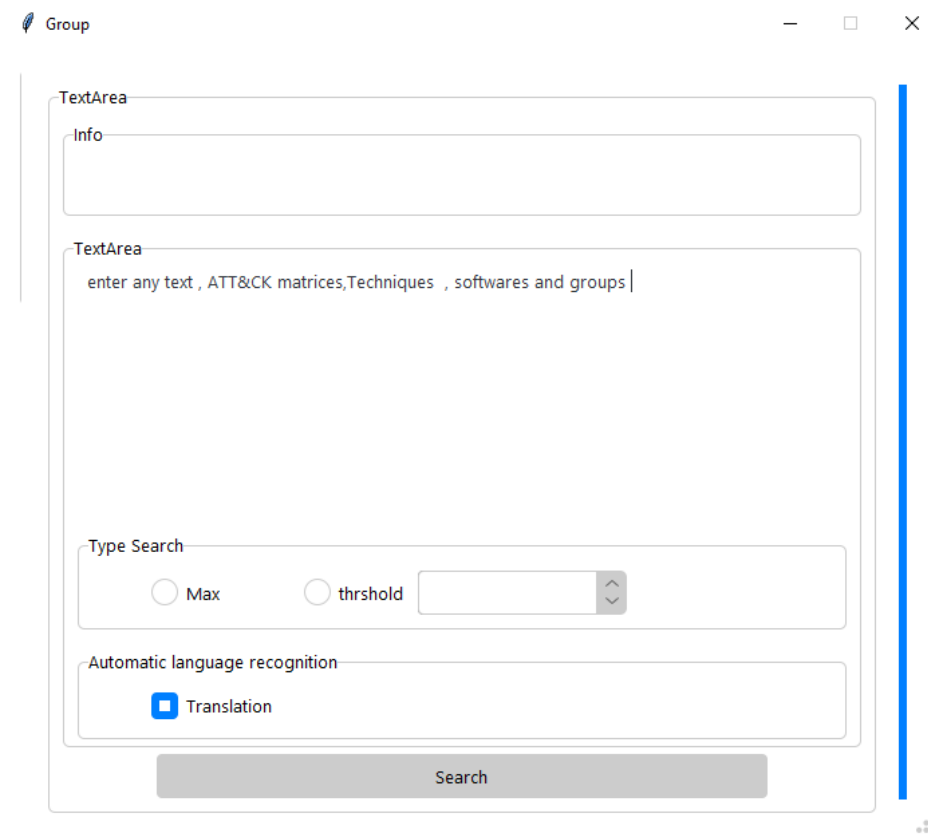


The screenshot shows the same web application window after a search. On the left, a table displays the results:

num	Software Name	Score
0	Bad Rabbit	83

Below the table is a pagination control showing "Page 1 of 1" and navigation buttons. On the right, the input fields from the previous screenshot are visible, with the "Info" field containing the HTML code "<*>; : enter any text". The "Max" radio button is still selected in the "Type Search" section.

Fig. 6.19 Example of Using the Incident Prediction Model



The image shows a web-based interface for a Group Prediction Model. The interface is contained within a window titled "Group" with standard window controls (minimize, maximize, close). The main content area is divided into several sections:

- TextArea Info:** A text input field labeled "Info" at the top.
- TextArea:** A larger text input field with the placeholder text "enter any text , ATT&CK matrices,Techniques , softwares and groups |".
- Type Search:** A section containing two radio buttons: "Max" and "thrshold". To the right of the "thrshold" radio button is a small input field with a dropdown arrow.
- Automatic language recognition:** A section containing a checked checkbox labeled "Translation".
- Search:** A large, grey, rectangular button at the bottom of the interface.

A vertical blue bar is visible on the right side of the interface, and a small grid icon is located at the bottom right corner of the window.

Fig. 6.20 Interface for Group Prediction Model

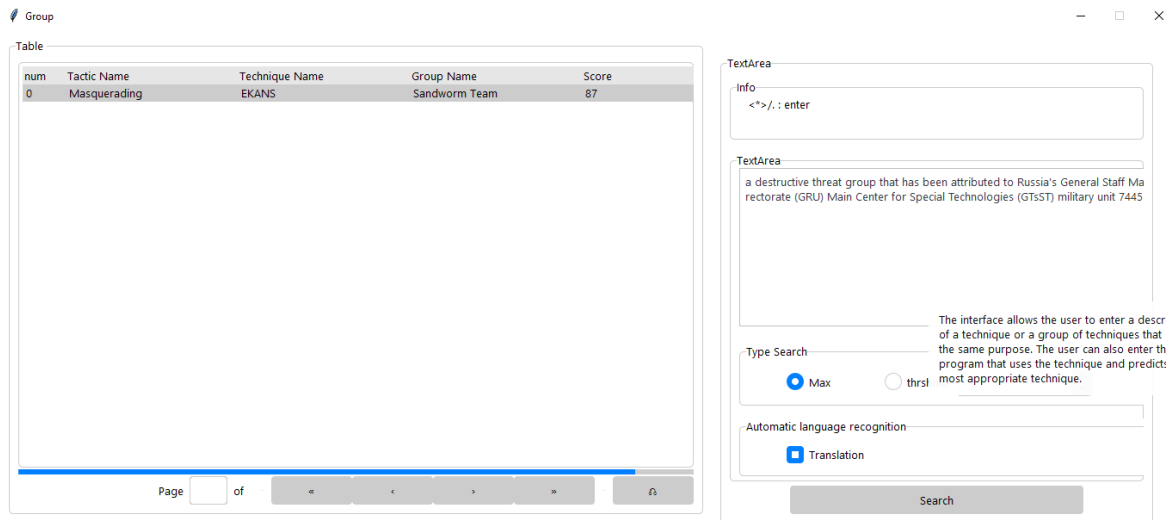


Fig. 6.21 Example of Using the Group Prediction Model

Figure (6.22) presents the interface for the Threat-Based Scenario Generation Model. The model searches for user-provided input, identifies and verifies their scenarios based on the previous ICS incidents, as identified in the MITRE framework. Users can input descriptions of techniques, incidents, and groups to predict scenarios accordingly.

The search and prediction method of the model differs from the prediction model technique. In the technique prediction model, the search is conducted for the most suitable technique based on its match to the specified description. In the attack scenario generator model, the search is conducted for all possible scenarios based on their match to the specified description. The method of determining the optimal scenarios depends on the value of "Max". If the value of "Type search" is "Max", then all possible scenarios will be generated that require each technique to be associated with only one technique from the following tactics, according to the choice of the highest similarity percentage. This often achieves the creation of sequence techniques that are similar to previous incidents. The method of determining similar techniques depends on the value of the "threshold".

If the value of "threshold" is "0", all techniques will be considered similar. That means the model will generate all the possible scenarios, which may be correct or otherwise.

If the value of "threshold" is greater than "0" and less than "1", techniques with a similarity degree greater than the "threshold" value will be considered similar, and scenarios can be generated from them.

Users can specify the desired "scenario type," which is (successful or close to successful), and the model searches for the most suitable scenarios. The interface displays generated scenarios in a table, showing techniques, the primary tactic, similar incidents, and the likelihood score. Additionally, the interface offers analysis sections for the scenarios, including

statistical graphs for scenario types and their distribution based on verification against ICS incidents. TextArea in Figure 6.22 illustrates an example where the user intends to create intrusion scenarios for an ICS system. The user can input a scenario description, such as "Utilizing the Drive-by Compromise exploit in an ICS system to download command and control files." Additionally, the user can also specify a value for "Max"

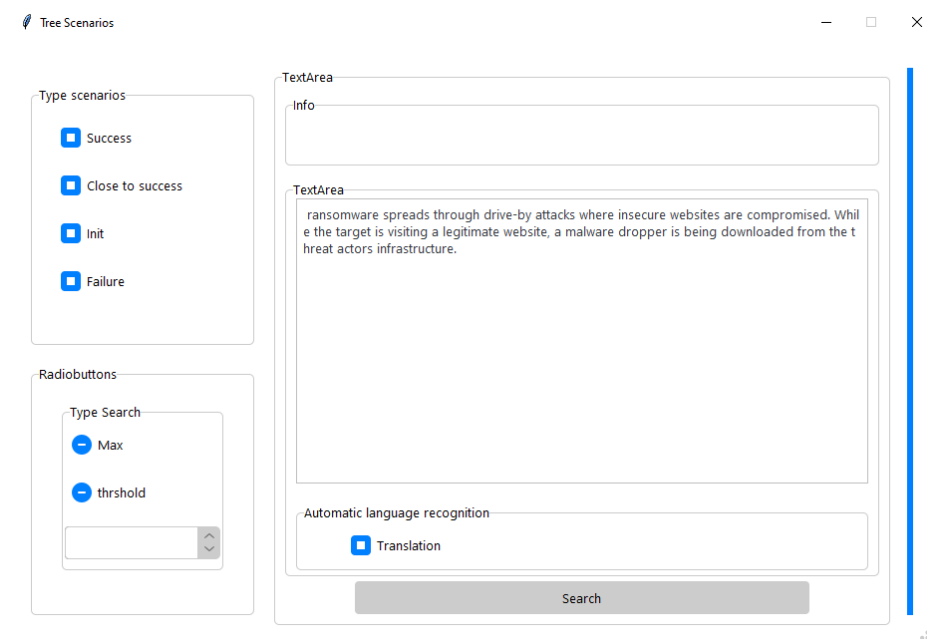


Fig. 6.22 Interface for Threat-Based Scenario Generation Model

Based on these settings, the model will search for all possible scenarios. Once the model completes the search process, an interface for scenarios will be displayed, as shown in Figure (6.23). This includes a table listing all generated scenarios, presenting the primary technique and tactic of each scenario, along with the corresponding matching incidents and their probability. The interface also features four main sections for scenario analysis.

The *top left section* displays a chart illustrating the number of created scenarios and the time taken for the generation process. The *upper right corner* of Figure 6.23 shows statistics related to determining the number of scenarios according to their four specified types (success, initial, close to success, and failure), as detailed in Chapter 4. The *left bottom section* presents a statistical chart depicting the distribution of scenarios based on their verification against ICS incidents, as discussed in Chapter 4. Lastly, an analysis of the scenario types, identifying their relevance to ICS incidents and determining the percentage of their association with ICS incidents, is illustrated.

The user can obtain detailed information about each scenario by double-clicking on the rows in the scenario, as shown in Figure 6.24. This opens an interface presenting information

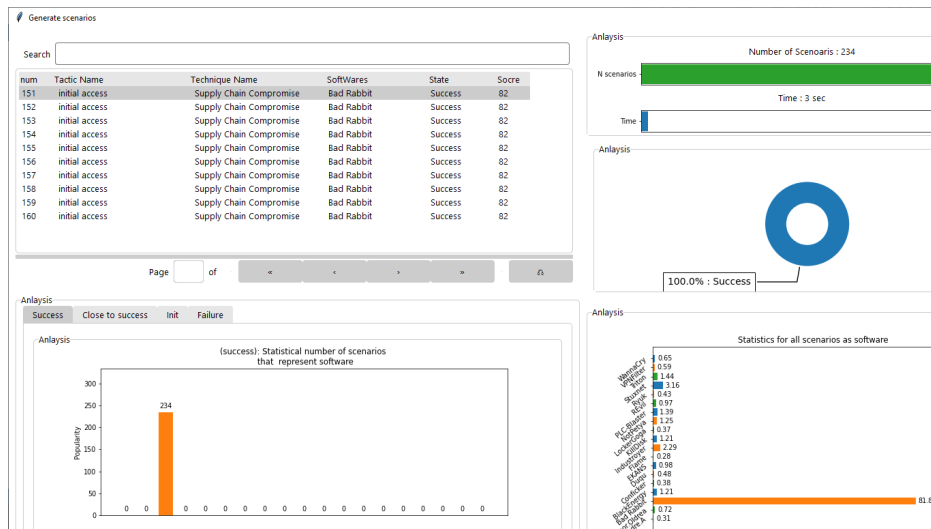


Fig. 6.23 Interface for Generated Scenarios

about the scenario, including its status, the number of techniques, and the likelihood of accuracy regarding the scenario’s representation of ICS incidents. The interface further provides a table of techniques representing the scenario, along with their associated tactics. It also indicates the likelihood of similarity between the scenario and all ICS through a statistical graph.

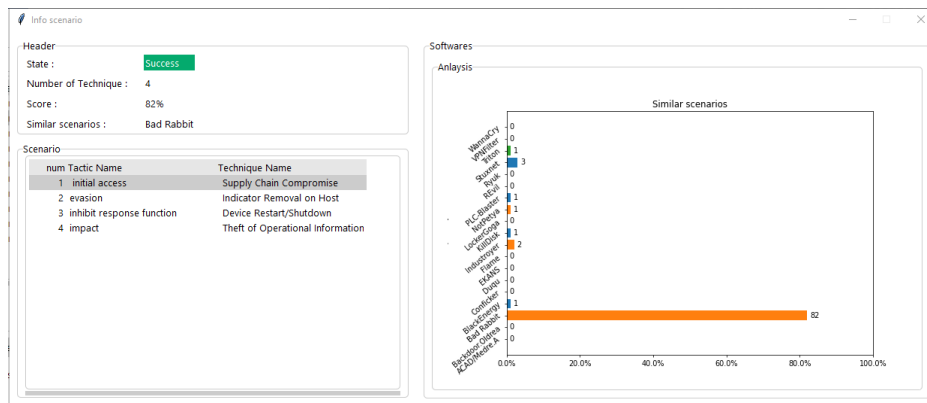


Fig. 6.24 Interface Presenting Information About a Specific Scenario

6.10 Application Programming Interface (API) Configurations

In this section, the API architecture and different usability options are highlighted. Along with the architecture, the cloud-based API server is also defined. How the authentication and authorization of users are tackled to keep the system safe and secure, is briefed. Furthermore, the deployment of services and implementation in terms of the web-based interface is also explained.

6.10.1 Architecture of the ASG Application

The Representational State Transfer (REST) architecture is used in the development of this API, which is an architectural approach that defines a set of constraints that will be used in the creation of web services. REST API is the way to access web services in a simple and flexible way without the need for any processing. It also helps in the use of cloud applications; the cloud system is responsible for maintaining the API and the database that is associated with the applications. The architecture provides flexibility in terms of choosing the type of client and enabling the system to cater to multiple clients at the same time. For example, a simple web application contains all control logic, and its data is stored in one place (cloud), allowing the user to create and generate automated scenarios according to the methods of ASG models.

6.10.2 Cloud API Server

The API service is developed by Django, which is an open-source web framework based on Python. Django provides a powerful foundation and a set of tools that can be used to create fast and scalable web applications. This Cloud API Server works as an intermediary and is responsible for traffic, to connect it with the web main interface. The interface is developed to deal with ASG models and their different components. The API is responsible for accessing and controlling the data materials of classification models and scenario builders, such as chat to predict ICS techniques and events through receiving text inputs from the user. This service puts the system in a state of independence and the ability to develop the process more simply and easily because the process of adding or deleting does not affect the other system services. It can be concluded so far that the API Server represents the gateway that enables users to access ASG models and interact with application interfaces. From a development point of view, the website is developed using ASP.NET Blazor technology. Visual Studio 2022 has been used as the development environment for the system because it is characterized by

providing templates for dealing with websites and different applications. C# is used as a dynamic language to connect the site with API ASG. The Blazor framework is characterized by Razor components, which are small, reusable pieces of code that can be used to create HTML, CSS, and JavaScript. Blazor components can be combined to create complex user interfaces, and they can be shared and reused easily across projects.

6.10.3 Authentication and Authorization

The process of accessing services through the Cloud API Server is not directly accessible. A user must register prior to using the web-based interface. It is mandatory to acknowledge the identity of users and define their permissions in the system before allowing access to the ASG API. To achieve so, it is important to add a middle layer to the system to make it secure. This layer or component is an independent layer, specifically designed for security purposes, and is responsible for processing login, authentication, and authorization of users. Specifically, IdentityServer 6 is implemented as the middle layer to authenticate and authorize API users and allow requests to connect ASP.NET Core-based APIs to applications¹. IdentityServer6 enables the admins of the website to define which users and applications should be enabled and disabled, and defines the authorization policy that the user must follow to access ASG API services.

6.10.4 Services Deployment

The ASG API Server services have been deployed on PythonAnywhere² hosting. PythonAnywhere is a cloud development environment that allows Python code to be executed in the cloud. It is a good option for developers demanding a fully managed environment for running Python applications. It also provides a variety of tools for testing Python code. The API deployment included all the classification models for techniques, events, groups, the scenario builder, the database, and user and authentication services. As for the website, its services have been deployed on SmarterASP³ hosting. SmarterASP is a web hosting company specializing in Microsoft ASP.NET. It offers a variety of hosting plans, including shared hosting, managed virtual hosting, and dedicated hosting. The website services can be deployed directly by Visual Studio. The website can be accessed via the link <https://asgmodel-001-site1.etempurl.com>.

¹[Docs.identityserver.io](https://docs.identityserver.io)

²www.pythonanywhere.com/

³SmarterASP.net

6.10.5 ASG Web-based Application

The ASG website has been developed to make it easier for users to interact with the model services. It provides the ability to use and test the scenario builder and allows access to it from multiple application interfaces. Figures 6.25 and 6.26 show the home page of the ASG website, which displays a descriptive overview of the model in three main sections. If the user clicks on "Prediction Models", they will be taken to the Chat AI page, which allows them to enter text input and predict the related events, groups, and techniques. They can also use the different interface options e.g., to navigate to the scenario builder.

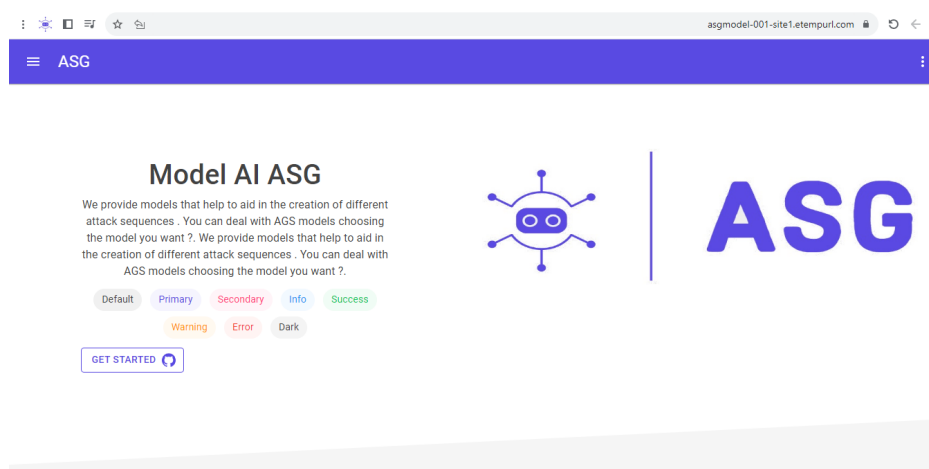


Fig. 6.25 ASG WEB Interface

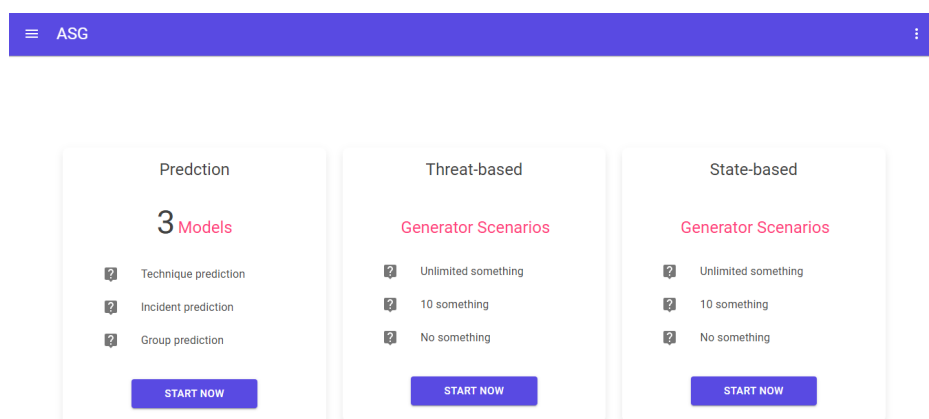


Fig. 6.26 ASG WEB Interface Options

Users can interact with ASG models through the Chat AI page. As shown in Figure 6.27, the page displays a chat interface where users can communicate with the models to predict attack techniques, events, and groups, create attack scenarios, and handle text input. Once

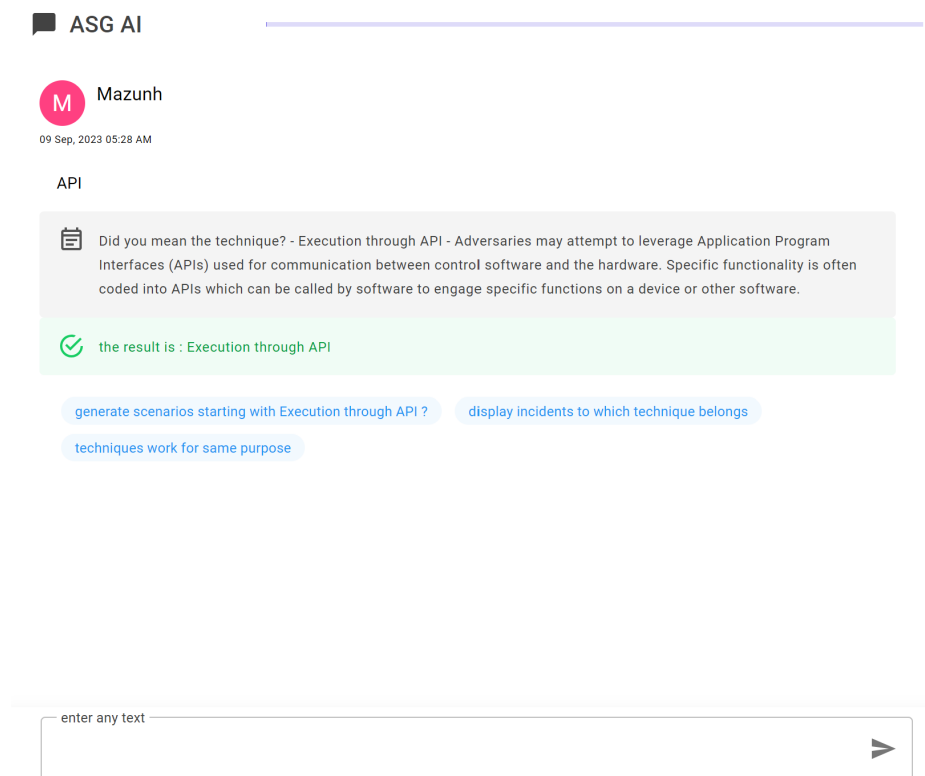


Fig. 6.27 ASG WEB Chat AI

the user enters technical texts, the model will search for attack techniques and display them along with a set of suggestions on whether they want to search for similar techniques as well or generate attack scenarios starting from the suggested technique. It will also display real incidents associated with the suggested technique. If the user clicks on the "Generate Attack Scenario" button, the model will respond by providing a series of techniques for the suggested attack scenarios and selecting the incidents that are most similar to the scenarios, as shown in Figure 6.28 6.29. Through the previous application interfaces and the Chat AI page, users can search for attack scenarios in different ways. They can also communicate with the models to control the production of attack scenarios and display the incidents that are most similar to the proposed scenarios.

ASG AI

Tactic	Technique	Software
execution	Native API	
Stuxnet is state & score : Initi 37.84		
execution	Native API	
impact	Denial of Control	
Industroyer is state & score : Initi 30.59		
execution	Native API	
impact	Loss of Control	

Rows per page: 10 1-10 of 19

Percentage of generating scenarios



Fig. 6.28 ASG Chat AI: Scenario Generation

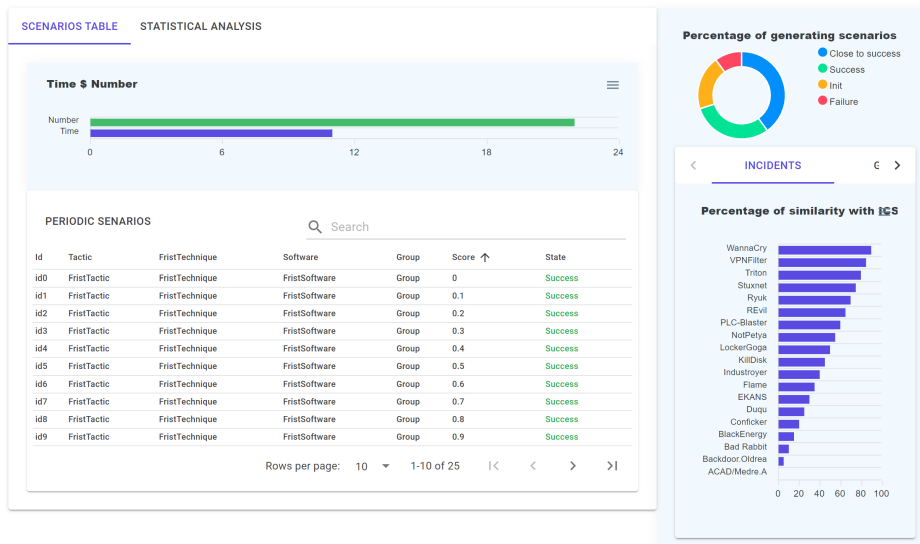


Fig. 6.29 ASG Chat AI: Generated Scenarios Analysis

6.11 Chapter Discussion

Analysis of cyber incidents using attack scenarios is an important process for understanding, addressing, and preventing cyber threats. Security experts can gain a deeper understanding of incidents by creating attack scenarios that define the attacker's steps and techniques. We present a general methodology for generating attack scenarios using three different approaches: manual, threat-based, and state-based.

The manual approach involved collecting data about the incident, extracting attack techniques, and comparing them to previous incidents. This helped in identifying commonalities between incidents and developing a more comprehensive understanding of attacker techniques. However, the manual creation of attack scenarios can be time-consuming and may not be feasible for all incidents. Additionally, the quality of attack scenarios will depend on the knowledge and expertise of the analyst.

On the other hand, a threat-based approach can be used to automate the process of creating attack scenarios. This approach trains a model on a comprehensive knowledge base of attack techniques and incident data. AI-based solutions promise to address these challenges. These solutions are used to generate attack scenarios automatically, saving time and effort for cybersecurity experts. They are also used to generate more comprehensive and accurate attack scenarios. There are still some challenges that need to be addressed before AI and machine learning can be used to automate attack scenario generation.

The state-based approach takes into account the current state of the system when generating attack sequences. This means that it can generate more realistic and actionable attack scenarios than scenarios that are generated without considering the current state of the system. State and attack techniques can also be controlled by security analysts and experts. Graphical user interfaces (GUIs) make attack scenario generation models more user-friendly by providing a simple and easy-to-understand way to interact with the model. This can be especially useful for non-technical users who may not be familiar with the technical details of the model.

Developing GUIs to deploy the general methodology will help make it more accessible to the general public. This will make creating attack scenarios easier and more efficient. Developing APIs to provide access to model services from more than one application interface will help make the general methodology more flexible and adaptable. This will allow users to create attack scenarios using an application interface that meets their specific needs. Overall, a general methodology for generating attack scenarios is presented which is flexible, adaptable and can support a variety of goals and needs.

6.12 Conclusion

As it has been mentioned, this chapter has covered all the details about the extracted general methodology which the security experts and analysts can use and the steps to be followed in order to create attack scenarios. All the challenges, contributions and limitations along with the final concluding remarks are discussed.

Chapter 7

Discussion and conclusion

7.1 Introduction

This section presents a comprehensive overview of the research methodology and findings in response to the study's primary research questions. The core objective of this research work was to develop a standardised and systematic approach for generating attack scenarios that accurately reflect the complexities of ICS environments. To achieve this, the first question related to standardisation of the methodology, "Can a standardised and systematic methodology be developed to facilitate the generation of attack scenarios?" is answered clearly in Chapter 6, Section 6.2, through the development of a structured approach established to create attack scenarios. Concerning classification and association of attack techniques, the question "How can the classification and alignment of attack techniques with distinct stages of the ICS cyber kill chain enhance the precision and relevance of generated attack scenarios?" has been investigated to conclude manual attack scenario generation by mapping attack techniques with the ICS cyber kill chain. A well-known attack incident, such as Shamoon, is used, as well as a new incident that has not been previously illustrated in the literature, namely ZeroCleare, in addition to Stone Drill. As a result of analysing scenarios, a contribution was made towards the Wiper Malware Taxonomy in Chapter 3. The question "Can data from previous ICS cyber incidents be used for prediction?" is related to the utilisation of incident data. To answer this question, the potential of leveraging previous cyber incident data was explored to enhance the predictive capabilities of the scenario generation process. The details and deep investigation are presented in Chapter 4. It can be confidently stated that we utilised the threat intelligence database and proposed models to predict attack techniques, identify the groups behind the attacks, generate attack scenarios, predict incidents, and optimise the generated scenarios by comparing them with previous incidents. In addition, for the dynamic scenario generation based on system state, the key question is "How effectively can the use

of real-time system state information contribute to the dynamic and adaptable generation of attack scenarios?” The researcher has thoroughly illustrated this in Chapter 5. The proposed word generates attack scenarios using seven key values that represent the state conditions and mentions that the integration of system state data to create scenarios that adapt to evolving circumstances has been explored. The last question that has been answered in this research is related to optimisation of generated attack scenarios: “In what manner can the data of previous ICS cyber incidents be employed to optimise and tailor the generated attack scenarios?” To answer this question, methods for enhancing the effectiveness of attack scenarios based on insights from previous cyber incidents have been investigated in Chapter 4. To sum up, this study developed a comprehensive methodology for generating attack scenarios. We formulated a generalised framework applicable across various contexts through an in-depth exploration of manual, threat-based, and state-based attack scenario generation approaches. The validity and efficacy of this proposed methodology were rigorously assessed through a combination of use case analysis, comparative literature review, and expert interviews. These evaluations collectively substantiate the robustness and practical utility of our approach.

7.2 Discussion

The overall aim of this research work is to provide a novel, verified, and validated methodology for generating attack scenarios. These attack scenarios are primarily designed to prevent the recurrence of cyber incidents in the future. Throughout this research, existing solutions were studied, and our efforts contribute to the development of secure cyber systems and ICS. Chapter 2 extensively surveyed the literature to assess the current state-of-the-art systems, identify gaps, and propose means to uniformly evaluate the attack scenarios in ICSs.

Furthermore, it was observed that there is a significant lack of standardisation and automation for attack scenario generation. Considering this research gap, a general methodology was proposed, which follows seven main steps: gathering raw data, preparing the data, conducting inferential analysis, structuring the attack sequence, creating the attack sequence, generating the attack scenario, and verifying the scenarios. This methodology shows the feasibility of building ICS attack scenarios manually and automatically. The manual method analyses ICS cyber incidents to identify commonalities using the ICS cyber kill chain framework, which is detailed in Chapter 3. Based on the automatic method, the researcher created an attack scenario generator (ASG) tool that can generate attack scenarios, predict the adversary group behind the attack, and optimise those incidents depending on previous ones. This endproduct was achieved by creating an input of text, logs, or new incident information, generating all possible attack scenarios, and checking the success probability of the generated

attack scenario. At the state level, the tool generates attack scenarios that utilise system state information.

In Chapter 4, the threat-based methodology for generating attack scenarios is presented. Chapter 5 presented the system state methodology for dealing with attack scenario generation, validating the ASG tool that aims to fill the gaps related to cyber threat intelligence. The tool has been tested using internal data and external data. The external data are generated using AI tools, such as Bing, Google Bard, and ChatGPT (details in the Appendix). For further enhancement, a user and API were created to make the tool accessible and user-friendly. The overall experience was not without challenges, and numerous obstacles were encountered during the research process. Some of the challenges are highlighted in Section 1.3.

7.3 Challenges

In this section, some challenges related to the attack scenario generation are briefly illustrated.

- **Lack of Data:** A major challenge is the absence of data. The related data are necessary to create modules for predicting suitable techniques, which are derived from text or system statuses. One suggested solution for the lack of data is to access data through machine learning (ML). In this study, ML was used to produce the data from 78 rows, and as an output, approximately 23597 rows were generated, as previously discussed in depth in Chapter 4.
- **Scenarios Obtaining Mechanism:** In this thesis scenario, it is impossible to produce the scenario if the input is connected to every node. Additionally, the degree of similarity between nodes is taken into account as a threshold for the development of a relationship between nodes and a decrease in the interaction between techniques.
- **Time Complexity:** The time needed for an algorithm to complete its execution is known as time complexity. It describes the time required to create a scenario within the framework of this thesis. While there are numerous ways to reduce time complexity, the method of parallel programming is suggested in this thesis to create multiple scenarios simultaneously.
- **Lack of host-level information:** An issue was faced when trying to map approaches to state-level key values at the host-level, producing a scenario. The issue was addressed in the solution as pre- and postconditions (i.e., when the user feeds the model, it becomes flexible to learn).

7.4 Limitations

Due to certain constraints, some acknowledged limitations are presented in this section. This research thesis developed a general methodology to generate attack scenarios for ICSs. However, the model itself is not capable of executing the generated attack scenario. To perform the attack scenario, a compromised system and a listener are required to create a model that executes the generated attack scenario. Although the AGS tool has been tested and validated using internal and external data, yielding highly accurate results, it has not been tested for more than one user at a time because it is built on a local server. Finally, the ASG tool is an open-source tool; therefore, no fees are required. However, it may lead to data limitations because the research largely depends on open-source documentation and knowledge bases.

The research also discusses the performance of various prediction models using external data. Specifically, it evaluates three types of models: technique prediction, incident prediction, and group prediction, all of which use a support vector classifier with a linear kernel. The technique prediction model achieved an accuracy of 85%, a precision of 84%, and an F1 score of 79%. The incident prediction model had an accuracy of 76%, a precision of 80%, and an F1 score of 74%. Lastly, the group prediction model achieved an accuracy of 91%, a precision of 80%, and an F1 score of 80%. These results suggest high model efficiency based on statistical representation. Additionally, the research highlights several key outcomes, including a methodology for creating attack scenarios, a literature review on attack scenario generation, the feasibility of practical solutions, a data production model, and automation for model validation using external data. Numerous use cases collected from AI tools are also mentioned in the Appendix.

7.5 Conclusion

In terms of the broader implications, this research addresses two critical issues in attack scenario generation: standardisation and automation. The proposed general methodology offers a structured approach for security analysts to efficiently create attack scenarios. Moreover, the ASG tool represents a significant step towards automation in this domain, with high accuracy in predicting techniques, incidents, and adversary groups.

In conclusion, this study provides a valuable framework for generating attack scenarios and presents a practical tool for enhancing cybersecurity practices in ICSs. The general methodology and ASG tool can serve as valuable references for security experts and organisations seeking to improve their cybersecurity posture.

7.6 Future Work

The proposed methodology and AGS tool demonstrate the feasibility and effectiveness of automating the creation and optimisation of attack scenarios for ICSs. However, several avenues for future research and development remain.

The current approach relies on static data collection and analysis. Future work will integrate real-time data sources, such as security information and event management (SIEM) systems and ICS monitoring tools. This will enable the ASG to dynamically adapt scenarios based on live system status and evolving threats, improving the timeliness and relevance of the output. While this work focused on ICS environments, the methodology has the potential to be generalised to other critical infrastructure domains, including smart grids, healthcare cyber-physical systems, and intelligent transportation networks. Further research is required to validate the adaptability of the framework across these sectors, each of which presents unique operational and security challenges.

Despite the focus on automation, a human-in-the-loop component remains crucial for operational environments. Future iterations of the tool will explore mechanisms for analyst feedback and scenario refinement, supporting interactive scenario generation and validation. The proposed methodology and tool should be evaluated further across a broader range of case studies. Collaborations with industrial partners could provide access to real-world ICS environments and historical incident data, offering a stronger empirical basis for performance and usability assessments.

Aligning the ASG with international standards such as MITRE ATT&CK for ICS, NIST SP 800-82, and the IEC 62443 series will be a critical step towards regulatory compliance and broader adoption. Work is underway to ensure that the output of the ASG tool can be mapped to these standards for integration into existing security assessment workflows.

The promising results obtained so far will be further developed and submitted for publication in reputable journals and conferences focused on cybersecurity, industrial systems, and the protection of critical infrastructure. Such dissemination includes venues such as IEEE Transactions on Industrial Informatics, IEEE Access, and conferences such as ACSAC, RAID, and CRITIS. Publishing this work will help validate the methodology within the academic community and support its dissemination to industrial stakeholders.

References

- [1] Michael J Assante and Robert M Lee. The industrial control system cyber kill chain. *SANS Institute InfoSec Reading Room*, 1, 2015.
- [2] Tejasvi Alladi, Vinay Chamola, and Sherali Zeadally. Industrial control systems: Cyberattack trends and countermeasures. *Computer Communications*, 155:1–8, 2020.
- [3] Muammer Semih Sonkor and Borja García de Soto. Operational technology on construction sites: A review from the cybersecurity perspective. *Journal of Construction Engineering and Management*, 147(12):04021172, 2021.
- [4] Cas Cremers. Feasibility of multi-protocol attacks. In *First International Conference on Availability, Reliability and Security (ARES'06)*, pages 8–pp. IEEE, 2006.
- [5] Aviad Elitzur, Rami Puzis, and Polina Zilberman. Attack hypothesis generation. In *2019 European Intelligence and Security Informatics Conference (EISIC)*, pages 40–47. IEEE, 2019.
- [6] Jong Woo Park and Seung Jun Lee. A quantitative assessment framework for cyber-attack scenarios on nuclear power plants using relative difficulty and consequence. *Annals of Nuclear Energy*, 142:107432, 2020.
- [7] Roland E Haas and Dietmar PF Möller. Automotive connectivity, cyber attack scenarios and automotive cyber security. In *2017 IEEE International Conference on Electro Information Technology (EIT)*, pages 635–639. IEEE, 2017.
- [8] Yong Peng, Yu Wang, Chong Xiang, Xifeng Liu, Zhe Wen, Dongqing Chen, and Chaojun Zhang. Cyber-physical attack-oriented industrial control systems (ics) modeling, analysis and experiment environment. In *2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pages 322–326. IEEE, 2015.
- [9] Alyzia-Maria Konsta, Beatrice Spiga, Alberto Lluch Lafuente, and Nicola Dragoni. A survey of automatic generation of attack trees and attack graphs. *arXiv preprint arXiv:2302.14479*, 2023.
- [10] Adam Hahn. Operational technology and information technology in industrial control systems. *Cyber-security of SCADA and other industrial control systems*, pages 51–68, 2016.
- [11] Keith Stouffer, Joe Falco, Karen Scarfone, et al. Guide to industrial control systems (ics) security. *NIST special publication*, 800(82):16–16, 2011.

- [12] Christopher Bronk and Eneken Tikk-Ringas. The cyber attack on saudi aramco. *Survival*, 55(2):81–96, 2013.
- [13] Wen Tian, Miao Du, Xiaopeng Ji, Guangjie Liu, Yuwei Dai, and Zhu Han. Honeypot detection strategy against advanced persistent threats in industrial internet of things: A prospect theoretic game. *IEEE Internet of Things Journal*, 8(24):17372–17381, 2021.
- [14] National Institute of Standards and Technology (NIST). Cybersecurity framework, 2023. URL <https://www.nist.gov/cyberframework>.
- [15] Information Systems Security Association (ISSA). Cybersecurity essentials, 2023.
- [16] A. C. Jones and B. D. Smith. *Cybersecurity for everyone: Practical steps to stay safe online*. ABC Publishing, 2023.
- [17] Michael Kifer and Georg Lausen. F-logic: A declarative logic programming language for frame-based knowledge representation. In *Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data*, pages 127–138, 1992.
- [18] Owl cyber defense, 2024. URL <https://www.owlcyberdefense.com/>.
- [19] Muhammad Azmi Umer, Chuadhry Mujeeb Ahmed, Muhammad Taha Jilani, and Aditya P Mathur. Attack rules: an adversarial approach to generate attacks for industrial control systems using machine learning. In *Proceedings of the 2th Workshop on CPS&IoT Security and Privacy*, pages 35–40, 2021.
- [20] Yichuan Wang, Zhaoqi Lv, Yaling Zhang, Yequi Xiao, and Xinhong Hei. Petri net-based attack modeling for industrial control system networks. In *China Intelligent Robotics Annual Conference*, pages 3–19. Springer, 2022.
- [21] Toshiyuki Fujikura and Ryo Kurachi. An attack scenario generation method using the behavior model. In *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 59–591. IEEE, 2019.
- [22] Lockheed Martin Threat Research Labs. Kill chain: A precise approach to defeat advanced threats, February 2011.
- [23] J. T. Shackelford, M. R. Sargor, and J. R. C. Nurse. Deception for cyber security. *Information Security Journal: A Global Perspective*, 28(1):75–94, 2019.
- [24] Paul Kirshen, Samuel Merrill, Peter Slovisky, and Norman Richardson. Simplified method for scenario-based risk assessment adaptation planning in the coastal zone. *Climatic Change*, 113:919–931, 2012.
- [25] National Institute of Standards and Technology. Guide for conducting risk assessments. Technical Report 800-30 Rev. 1, sep 2012.
- [26] Hyojeong Lee, Jeff Seibert, Endadul Hoque, Charles Killian, and Cristina Nita-Rotaru. Turret: A platform for automated attack finding in unmodified distributed system implementations. In *2014 IEEE 34th International Conference on Distributed Computing Systems*, pages 660–669. IEEE, 2014.

- [27] Marina Dabic, Ana Colovic, Olivier Lamotte, Mollie Painter-Morland, and Silvana Brozovic. Industry-specific csr: analysis of 20 years of research. *European Business Review*, 28(3):250–273, 2016.
- [28] Feng Li, Xinteng Yan, Yunyun Xie, Zi Sang, and Xiaoshu Yuan. A review of cyber-attack methods in cyber-physical power system. In *2019 IEEE 8th International Conference on Advanced Power System Automation and Protection (APAP)*, pages 1335–1339. IEEE, 2019.
- [29] Steven Noel, Eric Robertson, and Sushil Jajodia. Correlating intrusion events and building attack scenarios through attack graph distances. In *20th Annual Computer Security Applications Conference*, pages 350–359. IEEE, 2004.
- [30] Josip Bozic and Franz Wotawa. Planning the attack! or how to use ai in security testing. In *Iwaise: First international workshop on artificial intelligence in security*, volume 50, 2017.
- [31] Saad Khan and Simon Parkinson. Towards automated vulnerability assessment. 2017.
- [32] Jörg Hoffmann. Simulated penetration testing: From "dijkstra" to "turing test++". In *Proceedings of the international conference on automated planning and scheduling*, volume 25, pages 364–372, 2015.
- [33] Doug Miller, Ron Alford, Andy Applebaum, Henry Foster, Caleb Little, and Blake Strom. Automated adversary emulation: A case for planning and acting with unknowns. *MITRE CORP MCLEAN VA MCLEAN*, 2018.
- [34] National Institute of Standards and Technology (NIST). Adversary emulation framework. Technical Report 800-162 Rev. 1, 2021.
- [35] Livinus Obiora Nweke and Stephen Wolthusen. A review of asset-centric threat modelling approaches. 2020.
- [36] Adam Amos-Binks, Joshua Clark, Kirk Weston, Michael Winters, and Khaled Harfoush. Efficient attack plan recognition using automated planning. In *2017 IEEE symposium on computers and communications (ISCC)*, pages 1001–1006. IEEE, 2017.
- [37] F. Hoffmann. *Cyberterrorism: A challenge to homeland security*. Routledge, 2005.
- [38] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security Privacy*, 9(3): 49–51, May 2011. ISSN 1558-4046. doi: 10.1109/MSP.2011.67.
- [39] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.
- [40] Christopher Bronk and Eneken Tikk-Ringas. Hack or attack? shamoon and the evolution of cyber conflict. 2013.
- [41] Hamad Al-Muhaimid. cyber attacks targeting saudi sites.. information technology and the communications authority warns. <http://www.alekhbariya.net/ar/node/27138>, Jan 2017. Accessed on Jan 2020.

- [42] Martin Giles. Triton is the world’s most murderous malware, and it’s spreading. *Online: <https://www.technologyreview.com/s/613054/cybersecurity-critical-infrastructure-triton-malware>*, 2019.
- [43] Costin Raiu, Mohamad Amin Hasbini, Sergey Belov, and Sergey Mineev. From shamoon to stonedrill–wipers attacking saudi organizations and beyond. *Kaspersky Lab, March*, 2017.
- [44] Oxford Analytica. Business will be in the cross-hairs of gulf cyberwar. *Emerald Expert Briefings*, (oxan-es), 2019.
- [45] IBM. New destructive wiper “zeroclear” targets energy sector in the middle east. Technical report, IBM, Dec 2019.
- [46] Inc. Author Name(s) or Dragos. Title of the blog post. *Dragos Blog*, 2023. URL [URLoftheBlogPost](#).
- [47] SANS Institute. The industrial control system cyber kill chain. Technical report, SANS Institute, 2023. URL <https://www.sans.org/white-papers/36297/>.
- [48] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. The diamond model of intrusion analysis. Technical Report CMU/SEI-2013-TR-001, Carnegie Mellon University, Software Engineering Institute, August 2013.
- [49] OWASP. Threat modeling process. URL https://owasp.org/www-community/Threat_Modeling_Process.
- [50] Radhakrishna Jha, Jeannette M. Wing, and Uma Shankar. Two formal analyses of attack graphs. *IEEE Explore*, Oct 2002. URL <https://ieeexplore.ieee.org/document/1021806>.
- [51] IBM. Combating destructive malware: Lessons from the frontline. Technical report, IBM, 2019.
- [52] USCERT. Destructive malware response. Technical report, National Cybersecurity and Communications Integration Centre, 2009.
- [53] IBM. What is destructive malware? Technical report, IBM, 2019.
- [54] Sumayah Al-Rabiaah. The “stuxnet” virus of 2010 as an example of a “apt” and its “recent” variances. In *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pages 1–5. IEEE, 2018.
- [55] Abdullah Al-Jadeea. Al-zaid to al-akhbariya.net: shamoon 2 targets 15 institutions. <http://www.alekhbariya.net/ar/node/27138>, Jan 2017. Accessed on Jan 2020.
- [56] GReAT. Twas the night before. <https://securelist.com/twas-the-night-before/91599/>, Jul 2019. Accessed on Jan 2020.
- [57] Jonathan AP Marpaung and HoonJae Lee. Dark seoul cyber attack: Could it be worse? Conference of Indonesian Student Association in Korea. Retrieved from [http . . .](http://...), 2013.

- [58] A L Johnson. four years darkseoul cyberattacks against south korea continue anniversary korean war, June 2013. URL <https://www.symantec.com/connect/blogs/four-years-darkseoul-cyberattacks-against-south-korea-continue-anniversary-korean-war>.
- [59] Dan Goodin. Hard drive-wiping malware that hit south korea tied to military espionage, September 2013. URL <https://arstechnica.com/information-technology/2013/07/hard-drive-wiping-malware-that-hit-s-korea-tied-to-military-espionage/>.
- [60] *Mutual Security in the Asia-Pacific: Roles for Australia, Canada and South Korea*. McGill-Queen's University Press, 2015. URL <http://www.jstor.org/stable/j.ctt1jktr6v>.
- [61] Meredith Alexander, Ashley Chase, Kelly Chase, and James S O'Rourke. *Sony pictures entertainment, inc.: A cybersecurity attack from north korea (a)*. The Eugene D. Fanning Center for Business Communication, Mendoza College of . . . , 2015.
- [62] Nicholas Horton and Antonio DeSimone. Sony's nightmare before christmas: The 2014 north korean cyber attack on sony and lessons for us government actions in cyberspace. Technical report, JHUAPL Laurel United States, 2018.
- [63] Walter Houser. Could what happened to sony happen to us? *IT Professional*, 17(2): 54–57, 2015.
- [64] Mike Lennon. Hackers used sophisticated smb worm tool attack sony, December 2014. URL <https://www.securityweek.com/hackers-used-sophisticated-smb-worm-tool-attack-sony>.
- [65] The hack of sony pictures what you need to know, December 2014. URL <https://www.trendmicro.com/vinfo/ph/security/news/cyber-attacks/the-hack-of-sony-pictures-what-you-need-to-know>.
- [66] An analysis of the destructive malware behind fbi warnings, 12 2014. URL <https://blog.trendmicro.com/trendlabs-security-intelligence/an-analysis-of-the-destructive-malware-behind-fbi-warnings/>.
- [67] Chris Kubecka and CEO HYPASEC. How to implement it security after a cyber meltdown, 2015.
- [68] Sami Zhioua. The middle east under malware attack dissecting cyber weapons. In *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*, pages 11–16. IEEE, 2013.
- [69] Zakariya Dehlawi and Norah Abokhodair. Saudi arabia's response to cyber conflict: A case study of the shamoon malware incident. In *2013 IEEE International Conference on Intelligence and Security Informatics*, pages 73–75. IEEE, 2013.
- [70] R Falcone. Shamoon 2: return of the distrack wiper, 2018.
- [71] J O'Leary, J Kimble, K Vanderlee, and N Fraser. Insights into iranian cyber espionage: Apt33 targets aerospace and energy sectors and has ties to destructive malware. *Threat Research Blog*, 2017.
- [72] MITRE. Stonedrill. <https://attack.mitre.org/software/S0380/>, . Accessed on Jan 2020.

- [73] MITRE. Shamoan. <https://attack.mitre.org/software/S0140/>, . Accessed on Jan 2020.
- [74] Zeinab Bakhshi, Ali Balador, and Jawad Mustafa. Industrial iot security threats and concerns by considering cisco and microsoft iot reference models. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 173–178. IEEE, 2018.
- [75] Roberto Godreau. *SCADA systems and their vulnerabilities within the Smart Grid: Can they be defended from a cyber attack*. PhD thesis, Utica College, 2013.
- [76] Kamrul Hasan, Sachin Shetty, and Sharif Ullah. Artificial intelligence empowered cyber threat detection and protection for power utilities. In *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*, pages 354–359. IEEE, 2019.
- [77] Rob McMillan. Definition: threat intelligence. *Gartner.com*, 2013.
- [78] Wiem Tounsi and Helmi Rais. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & security*, 72:212–233, 2018.
- [79] github. mettagithub. URL <https://github.com/uber-common/metta>.
- [80] David Bianco. The pyramid of pain. *Enterprise Detection & Response*, 2013.
- [81] Leo Obrst, Penny Chase, and Richard Markeloff. Developing an ontology of the cyber security domain. In *STIDS*, pages 49–56. Citeseer, 2012.
- [82] Yali Gao, Xiaoyong Li, Hao Peng, Binxing Fang, and S Yu Philip. Hincti: A cyber threat intelligence modeling and identification system based on heterogeneous information network. *IEEE Transactions on Knowledge and Data Engineering*, 34(2): 708–722, 2020.
- [83] Sadegh M Milajerdi, Birhanu Eshete, Rigel Gjomemo, and VN Venkatakrishnan. Poirot: Aligning attack behavior with kernel audit records for cyber threat hunting. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 1795–1812, 2019.
- [84] Rachit, Shobha Bhatt, and Prakash Rao Ragiri. Security trends in internet of things: A survey. *SN Applied Sciences*, 3:1–14, 2021.
- [85] Nikolaos Polatidis, Elias Pimenidis, Michalis Pavlidis, Spyridon Papastergiou, and Haralambos Mouratidis. From product recommendation to cyber-attack prediction: Generating attack graphs and predicting future attacks. *Evolving Systems*, 11:479–490, 2020.
- [86] Igor Kotenko and Andrey Chechulin. Fast network attack modeling and security evaluation based on attack graphs. *Journal of Cyber Security and Mobility*, pages 27–46, 2014.
- [87] Zafar Iqbal and Zahid Anwar. Ontology generation of advanced persistence threats and their automated analysis. 2016.

- [88] Sophie Pinchinat, Mathieu Acher, and Didier Vojtisek. Atsyra: An integrated environment for synthesizing attack trees: (tool paper). In *Graphical Models for Security: Second International Workshop, GramSec 2015, Verona, Italy, July 13, 2015, Revised Selected Papers 2*, pages 97–101. Springer, 2016.
- [89] Jeremy Bryans, Lin Shen Liew, Hoang Nga Nguyen, Giedre Sabaliauskaite, Siraj Shaikh, and Fengjun Zhou. A template-based method for the generation of attack trees. In *Information Security Theory and Practice: 13th IFIP WG 11.2 International Conference, WISTP 2019, Paris, France, December 11–12, 2019, Proceedings 13*, pages 155–165. Springer, 2020.
- [90] Rajesh Kumar. An attack tree template based on feature diagram hierarchy. In *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*, pages 92–97. IEEE, 2020.
- [91] MITRE ATT&CK. Mitre att&ck. URL: <https://attack.mitre.org>, 2021.
- [92] Yuli Vasiliev. *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press, 2020.
- [93] Karen Sparck Jones. What is the role of nlp in text retrieval? *Natural language information retrieval*, pages 1–24, 1999.
- [94] Vipin Kumar and Basant Subba. A tfidfvectorizer and svm based sentiment analysis framework for text data corpus. In *2020 National Conference on Communications (NCC)*, pages 1–6. IEEE, 2020.
- [95] Yihua Chen, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10(3), 2009.
- [96] Haifeng Wang and Dejin Hu. Comparison of svm and ls-svm for regression. In *2005 International conference on neural networks and brain*, volume 1, pages 279–283. IEEE, 2005.
- [97] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.
- [98] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018.
- [99] Xinming Ou, Wayne F Boyer, and Miles A McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 336–345, 2006.
- [100] Xinming Ou, Sudhakar Govindavajhala, Andrew W Appel, et al. Mulval: A logic-based network security analyzer. In *USENIX security symposium*, volume 8, pages 113–128. Baltimore, MD, 2005.
- [101] Mariam Ibrahim and Ahmad Alsheikh. Automatic hybrid attack graph (ahag) generation for complex engineering systems. *Processes*, 7(11):787, 2019.

- [102] Nils Ole Tippenhauer, William G Temple, An Hoa Vu, Binbin Chen, David M Nicol, Zbigniew Kalbarczyk, and William H Sanders. Automatic generation of security argument graphs. In *2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing*, pages 33–42. IEEE, 2014.
- [103] Laura P Swiler, Cynthia Phillips, David Ellis, and Stefan Chakerian. Computer-attack graph generation tool. In *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, volume 2, pages 307–321. IEEE, 2001.
- [104] Kijong Koo, Daesung Moon, Jun-Ho Huh, Se-Hoon Jung, and Hansung Lee. Attack graph generation with machine learning for network security. *Electronics*, 11(9):1332, 2022.
- [105] Reda Yacouby and Dustin Axman. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of NLP systems*, pages 79–91, 2020.
- [106] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002.
- [107] Rui Zhuang, Su Zhang, Scott A DeLoach, Xinming Ou, Anoop Singhal, et al. Simulation-based approaches to studying effectiveness of moving-target network defense. In *National symposium on moving target research*, volume 246. Citeseer, 2012.
- [108] Y. A. Abramovich, C. D. Aliprantis, and O. Burkinshaw. Another characterization of the invariant subspace problem. *Operator Theory in Function Spaces and Banach Lattices*. The A.C. Zaanen Anniversary Volume, *Operator Theory: Advances and Applications*, 75:15–31, 1995. Birkhäuser Verlag.
- [109] J. B. Conway. *A Course in Functional Analysis*. Springer-Verlag, New York, second edition, 1990.
- [110] Ju. I. Ljubič and V. I. Macaev. On operators with a separable spectrum. *Amer. Math. Soc. Transl. (2)*, 47:89–129, 1965.
- [111] ENISA. Study on csirt landscape and ir capabilities in europe 2025. Technical report, European Union Agency for Network and Information Security, Feb 2019.
- [112] Paul Cichonski, Tom Millar, Tim Grance, and Karen Scarfone. Computer security incident handling guide. *NIST Special Publication*, 800(61):1–147, 2012.
- [113] National Cybersecurity and Communications Integration Centre. National cyber incident response plan. Technical report, NCCIC, 2016.
- [114] National Cybersecurity and Communications Integration Centre. Recommended practice: Developing an industrial control systems cybersecurity incident response capability. Technical report, NCCIC, 2009.
- [115] ENISA. Good practice guide for incident management. Technical report, European Union Agency for Network and Information Security, 2010.

-
- [116] ENISA. Actionable information for security incident response. Technical report, European Union Agency for Network and Information Security, 2019.
 - [117] Hannes Tschofenig. Report from the internet of things software update (iotsu) workshop 2016, Feb 2017. URL <https://tools.ietf.org/html/draft-iab-iotsu-workshop-01>.
 - [118] US government. How to protect your networks from ransomware. Technical report. URL <https://arstechnica.com/information-technology/2013/07/hard-drive-wiping-malware-that-hit-s-korea-tied-to-military-espionage/>.
 - [119] MCIT. National cyber security strategy of saudi arabia. Technical report, ENISA, 2013.

Appendix A

Use cases for ASG

A.1 Motivation

According to the National Cyber Security Center (NCSC)¹, which states,

"You must report a cyber security incident within 72 hours of becoming aware of it".

Reporting an incident to government reporting services and relevant organizations is highly recommended by law enforcement in the UK. Reporting such incidents helps other organizations to prepare for attacks of the same nature. Reporting incidents further helps security analysts to analyze the reported incident and to create a hypothetical attack scenario and to combat such attacks. However, such practices always pose a significant challenge to recreate and mitigate the attack.

The reason that creating an attack scenario is challenging is that it requires extensive practical experience. Furthermore, such scenarios are widely based on accumulative threat intelligence (TI) factors. These factors may include the likelihood, strength and actors of the incident and whether or not this incident is applicable to the system. Although many state-of-the-art technologies support threat intelligence, however, it still requires a lot of manual labour by cyber security experts.

This approach besides being labour intensive has its own set of challenges, such as system complexity, time pressure, lack of experts and the increase of the threat landscape. Therefore, there is a pressing demand to standardise and utilize the procedure for creating attack scenarios to support system security and minimize manual work by experts [108–110].

¹<https://www.ncsc.gov.uk/>

A.2 Use Cases

Defensive Gap Assessment

A defensive gap assessment allows an organization to determine what parts of its enterprise lack defences and/or visibility.

Solidarity factories for trains manufacturing cybersecurity team need to examine their current defence software efficiency. The team decided to use the Common Behavior Focused adversary model to assess the defence tools. The problem is that experts are needed to provide this kind of assessment. Due to the lack of experts, the cybersecurity team used [my tool...] to assess tools, monitor, and mitigate existing defences within an organization's enterprise with less expenses, time and better expertise.

num	Tactic Name	Technique Name	Group Name	Score
0	Loss of Productivity and Revenue	Ryuk	APT33	62

Page 1 of 1

Info
<*/. : enter any text

TextArea
APT33

Type Search
 Max thrshok

Automatic language recognition
 Translation

Fig. A.1 Common Behavior Focused on Adversary use case

Behavioral Analytic Development

The year 2022 has witnessed an increase of the threat landscape; around 39% of UK businesses reported suffering a cyber-attack and the cost of data breaches was reported to be an average of 4.35 million USD in only one year. For that reason, by going beyond traditional indicators of compromise (IoCs) or signatures of malicious activity, the proposed tool in this research thesis can be used to identify potentially malicious activity within a system or network that may not rely on prior knowledge of adversary tools and indicators. Since the

tool depends on MITRE ATT&CK, it is faster and more updated than the traditional tools that leverage how an adversary interacts with a specific platform to identify and link together a suspicious activity that is agnostic or independent of specific tools, that may be used.

ASG AI

DDos

Did you mean the technique? - Denial of Service - Adversaries may perform Denial-of-Service (DoS) attacks to disrupt expected device functionality. Examples of DoS attacks include overwhelming the target device with a high volume of requests in a short time period and sending the target device a request it does not know how to handle. Disrupting device state may temporarily render it unresponsive, possibly lasting until a reboot can occur. When placed in this state, devices may be unable to send and receive requests, and may not perform expected response functions in reaction to other events in the environment. Some ICS devices are particularly sensitive to DoS events, and may become unresponsive in reaction to even a simple ping sweep. Adversaries may also attempt to execute a Permanent Denial-of-Service (PDoS) against certain devices, such as in the case of the BrickerBot malware. (Citation: ICS-CERT April 2017) Adversaries may exploit a software vulnerability to cause a denial of service by taking advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Vulnerabilities may exist in software that can be used to cause a T1023 or denial of service condition. Adversaries may have prior knowledge about industrial protocols or control devices used in the environment through [Remote System Information Discovery](https://attack.mitre.org/techniques/T0888). There are examples of adversaries remotely causing a [Device Restart/Shutdown](https://attack.mitre.org/techniques/T0816) by exploiting a vulnerability that induces uncontrolled resource consumption. (Citation: ICS-CERT August 2018) (Citation: Common Weakness Enumeration January 2019) (Citation: MITRE March 2018) In the Maroochy attack, the adversary was able to shut an investigator out of the network. (Citation: Marshall Abrams July 2008)

the result is : Denial of Service

generate scenarios starting with Denial of Service ?

display incidents to which technique belongs

techniques work for same purpose

enter any text



Fig. A.2 Behavioral Analytic Development Use Case Using ASG AI1

A.3 Usability Study

In order to maintain the best outcomes and ensure the quality of the research, a usability study following a mixed methods approach was conducted. The approach included both quantitative and qualitative methodology based on a questionnaire and an interview session comprising of twenty experts working in multinational companies in the fields of cyber security and IoT. The main purpose of the interview was to test and validate the tool holistically.

ASG AI

Tactic	Technique	Software
inhibit response function	Denial of Service	
		Industroyer is state & socre : Initi 41.87
inhibit response function	Denial of Service	
impact	Denial of Control	
		Industroyer is state & socre : Initi 56.7
inhibit response function	Denial of Service	
impact	Denial of View	

Rows per page: 10 1-10 of 23

Percentage of generating scenarios



enter any text

Fig. A.3 Behavioral Analytic Development Use Case Using ASG AI2

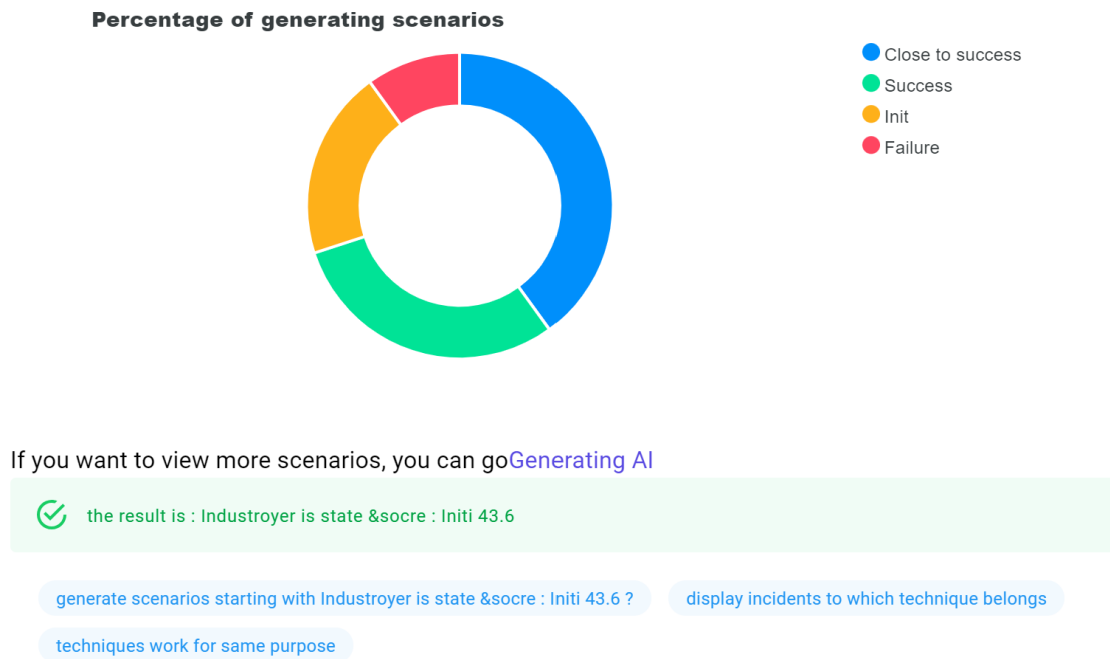


Fig. A.4 Behavioral Analytic Development Use Case Using ASG AI3

Online Interviews

The online demo of the tools was also conducted to demonstrate the tool and to educate the participants on, how it works and the steps to follow. This demonstration was presented to the participants in the form of a video interview online, via Zoom. The length of each interview remained around 30 minutes. It took place right after finishing the Attack Scenario Generation tool. The data collected for the participants is defined in the Section A.3 and are shown in the Figure A.5

The Questionnaire Components

General Information:

Participant Information:

Name (optional):

Occupation:

Did you find the ASG tool beneficial? Yes or NO

Did you find ASG tool beneficial? *

YES

NO

Do you think that ASG tool will be effective for your organization? *

Yes

No

Do you have any comments related to the ASG tool?

Long-answer text
.....

Fig. A.5 The Questionnaire Template

Overall Satisfaction and Comments:

The participants of the study were asked a number of questions in an anonymous questionnaire to be able to get authentic feedback right after watching the demonstration.

Results and General Comments

The results of the questionnaire illustrate the following; 80% of the respondents confirmed that ASG tool is effective and beneficial for their organizations specifically and for cybersecurity generally. They also agreed that the tool will assist the organizations in many ways, such as planning for penetration testing and red teaming, risk assessments, and vulnerability assessments. The remaining 20% commented saying that this tool is considered a great effort, however, it needs more features like showing threats feeds, and any historical number for the threats. They also mentioned that they are not fully aware of the tool's capabilities. Another respondent added that the tool doesn't recognize synonyms, for example, "social engineering" or "hacker" as it deals with more technical lexical items. Finally, an expert mentioned that the tool lacks an integration plan. Results of the survey are presented in the A.6

Did you find ASG tool beneficial?

Do you think that ASG tool will be effective for your organization?

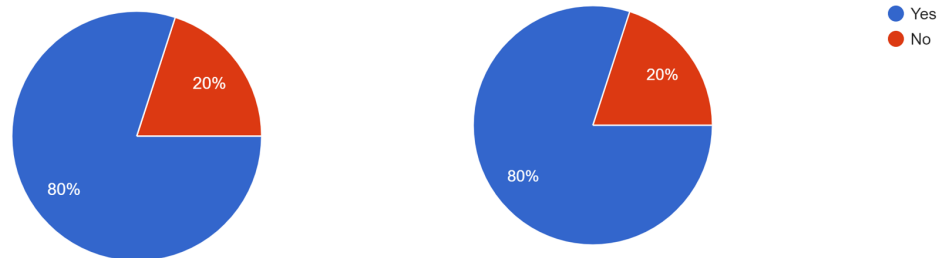


Fig. A.6 Results

Addressing the comments

In order to address the concerns of the users. Some efforts were made to address these issues. In order to address the issue of different languages, Google Translator API was integrated that supports various languages including Arabic. Next, to solve the problem related to synonyms and to make the ASG tool be aware of the synonyms. The SpaCy database, which

is an open-source tool for natural language processing, was added. It is more robust than NLTK library, especially in avoiding ambiguity. As for the integration plan, due to certain limitations, it was decided to consider it in future research work, since more information about cybersecurity systems organizations is needed.

A.4 Use Case in Graph

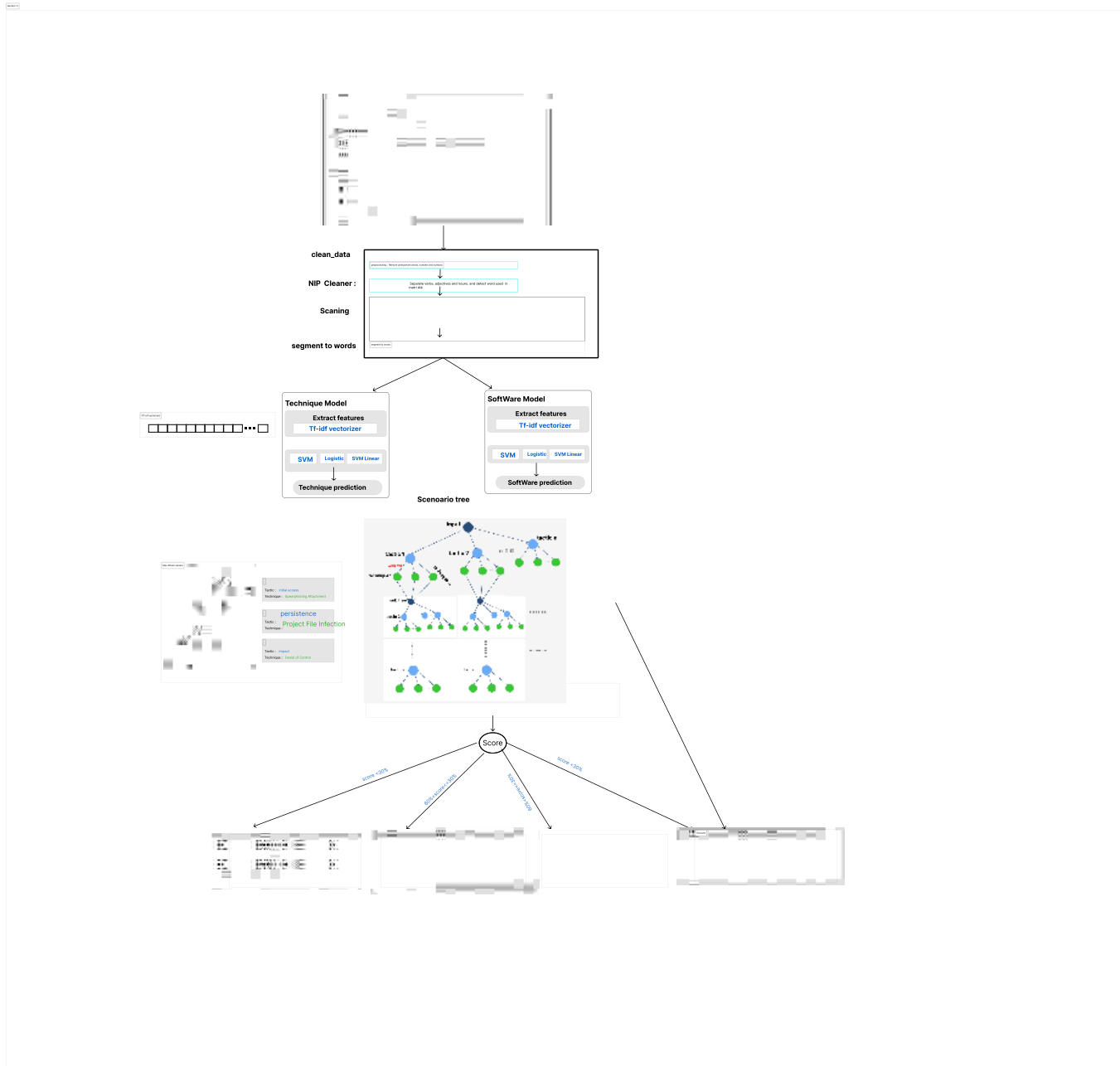


Fig. A.7 Use Case Explaining Threat-Based Model

Appendix B

Assessing Guidelines for Destructive Malware Response

begincomment

2.1 Incident Response Practices

There are currently two main international standards for cyber malware response. ENISA incident response practices [111] and NIST Computer Security Incident Handling Guide [112]. Both provide a detailed overview of how to respond to a cyber malware incident and provide guidelines on handling the attack. After reviewing both documents some limitations were observed in certain aspects, as they did not cover certain elements of the destructive malware kill chain. This report specifically focused on the NIST standard as it provides specific guidelines for the handling of destructive malware, and discusses what aspects of the attack are not covered.

2.1.1 United States National Cybersecurity and Communications Integration Centre Response plan

In the US national cyber incident guideline that follows NIST 800-61 regulations [112] the response plan depends on the cyber incident assessment. A cyber incident is assessed from a level 0 baseline, when the incident is inconsequential, to a level 5 Emergency, when the incident is of imminent threat to wide-scale critical infrastructure. The destructive attacks against oil and gas companies under this scheme are considered as Level 4 Severe [113]. As a general practice for cyber incident response, US-CERT [114] has defined some key elements that organizations should follow when a cyber-attack happens:

- **Planning:** The initial step is planning i.e., putting all the factors together to prevent the attack if possible or to be ready to respond when it happens. The planning phase includes building teams, establishing policies, indicating procedures, developing the plan itself, and evaluating the process.
- **Incident prevention/detection:** Incident prevention is mitigating the cyber incident by following NIST standards and procedures in parallel with using software tools and managing the patches. Incident management includes detecting the incident and reporting both suspected and unknown incidents.
- **Containment:** Containment is preventing the spread and effects of malware.
- **Remediation:** Remediation is the efforts in the period between the incident and the recovery including the eradication of the malware and rebuilding the system.
- **Recovery:** Includes restoring backup data, removing temporary containment actions and restarting all systems.
- **Post incident analysis and forensics:** The last and core phase according to US-CERT is post-incident analysis and forensics which consists of three areas: Lessons learned, Recurrence prevention which is analysis of the incident and initiating the action, and forensics which is collecting the evidence to use it for legal actions.

In regards to destructive malware, US-CERT [52] indicates that

- The system attacked by destructive malware is likely to have been overwritten or encrypted,
- The possibilities of noticing that the malware is running before it has completed is very low and
- The scope of the destructive malware increases at an alarming rate.

They recommend that the victim disconnect all non-critical infected devices before the malware spreads to the network. It also recommends data recovery and forensic analysis of the affected system be done offline. If the malware can spread to other devices on the local network without the internet (as in Shamoon), they suggested preserving the system logs and memory which may allow to detection of other affected devices and other malware used by the attacker. It also highlighted that asset owners need to prepare before the incident. In particular, asset owners should create up-to-date backups and safeguard these backups offline. As well as maintain the antivirus system frequently, monitor the network traffic and operate system patches.

2.1.2 European Union Agency for Cybersecurity Response Plan

There are 363 CERT teams, or computer security incident response teams (CSIRT) worldwide that, follow the ENISA incident response practices [115, 111]. In this study, the "Good practice guide for incident management" ENISA" [115] is followed, and assesses the guidelines in terms of effectiveness in the case of a real destructive wiper malware incident response. ENISA's practice workflow for cyber incidents consists of 15 activities starting from incident response and registration. ENISA advises the CERT to ask two questions: Is it a real incident? is it from your constituency? if yes, then register the incident and carry on the process. [Next step is the triage process, it assesses the incident to help CERT to understand how urgently they should respond. Afterwards it deals with the incident cycle which includes: incident resolution, data analysis, the action proposed, the action performed, recovery and closing of the incident. The final part of the process is the incident analysis cycle which is determined by five processes: final information, final classification, incident archiving, post-analysis and improvement proposal. ENISA hasn't published any specific guidelines or practices for destructive malware [116]. The incident assessment in ENISA is different than NCCIC and US-CERT (as explained in Sec. 2.1.1). In ENISA the incident depends on 5 levels, whereas EU-CERT assesses the incident on two aspects: severity and category. Severity is an incident considered against availability, confidentiality and integrity. Category is determined by which type of incident it is e.g. Denial of service, destructive attack, or phishing.

2.1.3 Analysis of Incident Response Guidelines

As the mapping of the guidelines to the attack breakdown provided in Sec. 3.6 proceeds, it has been identified that some of the key components of the attack are not being addressed by the current guidelines. The analysis suggests that even if a company were to follow the guidelines word for word, they might still be vulnerable to such attacks. Therefore, certain changes have been recommended to the guidelines to address these gaps and establish a global standard. These examples are taken directly from the descriptions of the attacks carried out on Aramco. Whilst this research provides recommendations that should be included in the official guidelines, these are just some of the ways that the guidelines could be improved and do not *yet* claim the list to be exhaustive.

Offline Communication

In stage 1, during the *Management and Enablement* phase, in Shamoon, the attacker took control of the victim's workstation through C&C and was able to spread it offline in Shamoon2.

During *Sustainment and Execution* stage, the attacker wiped the MBR and made the workstation inoperable. The guidelines suggest that one should disconnect the system, as was done by Aramco, still resulting in a successful attack. What the guidelines do not address is the importance of setting up offline communication. It has been established by the literature that communication is the most important aspect of a response [53, 113, 116]. Therefore guidelines should include the step of identifying key channels of communication to be left online or to identify alternative communication means in the case of a destructive malware incident e.g. phone or key targets to talk to.

Cyber Incident Response Team 24/7

As for the *Entrenchment* phase in the case of Shmoon1/2/3 and ZeroCleare, the attackers hard coded the date and time to start the attack during holidays or non-office hours, this was a huge contributor to the attack success [67]. The guidelines haven't taken this into account and haven't addressed that CERT teams should be always on hand even on holiday and the system should be monitored 24/7.

Fake Raw disk

In stage 2, during the *Attack Development and Tuning* stage in Shmoon, ZeroCleare, StoneDrill, and DarkSeoul the attacker uses a vulnerable signed driver. This allows the attacker to bypass security controls and gain access to the MBR. One of the core security vulnerabilities of any ICS is updates and installation of software [117], as recognized by the Internet Engineering Task Force (IETF). The specific guidelines for destructive malware have not discussed any authentication requirements for installations which could have prevented many of these attacks.

Backup Assurance

Finally, during the *Install/Modify* phase, there is no mention of backup assurance. If malware spreads quickly and undetected as in the case of all these attacks, backups could be compromised and malware could be included in them. In the guideline, they mentioned that one needs to backup, secure the backup and recover. However, additionally, we suggest that the recovered data should be accurate, complete and free of any malware. Data assurance techniques should be clarified in the guidelines to ensure no further spread.

Aramco Cert

It should be noted that the effectiveness of attacks is not altogether due to the lack of guidelines. In the case of Aramco, their emergency response team was founded after the Shamoon attack in 2012. After Shamoon the company went offline and all the offices were unplugged from the Internet. Aramco Cert team has three sub-teams:

- **Red, Reactive;** The red team is responsible for incident alert, warning, coordination, support, analysis and response;
- **Yellow, Quality Management;** The yellow team is responsible for announcements, information dissemination, risk management and dealing with international security services;
- **Green, Proactive;** the green team is responsible for awareness building, education training and publishing security handouts.

These follow on from some of the suggested practices in the guidelines.

2.1.4 Aramco Cyber Response

After Shamoon, the company went offline, disconnecting from the Internet. Kubecka, former consultant for Saudi Aramco, said “it was a massive army from cyber security specialist, I never seen anything like that”. Kubecka also indicated that the company bought 50,000 new hard drives and built incident response team and plan from zero.

2.1.5 Aramco Cyber Incident Response Assessment

During the assessment of Aramco’s cyber incident response, we found a baseline combination of NCCIC and ENISA recommendations. The cyber incident response plan consists of three phases: Plan, Response and Learn, and has been used to evaluate Aramco’s cyber incident response.

Plan: Aramco has a cyber incident response plan that includes identifying the CERT team, and coming up with a cyber security threat module. Aramco has not categorised cyber incidents like NCCIC, or prioritised them like ENISA, but has considered the implications of process, technology and information. Aramco has the privileges of admin controllers listed. However, logistic companies under Aramco all use the same email domain and system portal and have admin accounts. Aramco also sets to ensure regular maintenance of anti-virus and system patches and has regular backups for its systems, but no evidence is presented about safeguarding offline backups as NCCIC suggested.

Response: Aramco CERT is capable of identifying the cyber incident, even though they have coordination with international security companies to analyse the cyber incident. Aramco CERT is also capable of taking appropriate action, and of undertaking restoration and recovery.

Learn: Aramco is investigating the incident more while using expert help in the investigation. Aramco regularly updated the cyber incident plan after the investigation, and never publicly addressed the incident. No evidence was found that Aramco reported the incident to other companies in the sector, or warned them as ENISA suggested.

Recommended Security Controls

However, analysing the attacks, this report observed that several proposed guidelines were not followed by organisations targeted by the attacks and if they were followed the severity of the attack could have been greatly mitigated, the attack vector is analysed and some recommended solutions are provided:

1. Credentials Compromise (Sony Attack), implement multi-factor authentication, reduce admin privilege accounts [114, 116, 51].
2. Watering Hole (DarkSeoul and StoneDrill): patch the system frequently [116, 51, 113].
3. Phishing (Shamoon, ZeroCleare), authenticate emails using technology like sender policy framework [118].

2.2 Conclusion

Destructive malware are an increasing threat that has a huge impact on governments and organisations alike. Despite existing guidelines, these same attacks have been carried out successfully for several years with no signs of slowing down. This research report has reviewed and analysed these kinds of attacks extensively and presented a breakdown of destructive wiper malware in the last 8 years including previously undocumented ZeroCleare. Furthermore, it is stressed that our review finds that the effectiveness of these attacks is also due to the fact that companies are still not conforming to suggested practices whether due to negligence or conflicting global and national standards. Even though all regulations (EU NIS directive, NIST 800-61, and ISO/IEC 27001) require companies to disclose any cyber incident, Aramco never publicly addressed the incidents. This is due to a conflict of regulations, as the Saudi National Information Security Regulation (NISS) does not require the company to disclose their incident [119]. In the case of Aramco, they prioritised local

authority regulations. As an outcome of this analysis, current guidelines were investigated in the context of whether they are enough to cover and mitigate these attacks and have found key areas that should be addressed. The attacker's behaviour is very adaptive and is quickly able to change to adapt to new security measures, stressing the need for an adaptive security solution to match the attacks. We conclude our analysis by proposing the need for global guidelines for destructive malware attacks taking into consideration the existing guidelines and the suggestions we have provided in this work, as the current trends show that it is more likely that these attacks will increase rather than die down.

Chapter 3

Attack Scenario Generator

3.1 Introduction

This appendix contains all extra algorithmic representations that are highlighted in the body of the dissertation.

3.2 Algorithms

3.2.1 Data Production Algorithm

Data production is the first step to create an attack scenario builder. Therefore, it involves performing initial processing on the data set related to techniques and ICS incidents. This includes removing unnecessary information, symbols and numbers from text data. In addition to the use of (finite state machines FSM), which is a method of data production based on computational theory, and a method of machine learning to produce language. It was used as a method to increase the data set by identifying information that characterizes techniques and working to produce potential inputs. This helps analyze the MITRE ATT&CK® and produce the appropriate dataset to describe the techniques and ICS incidents. In the data production algorithm, the machine reads the data from the knowledge base, and extract the data is needed, such as technique names, description, platforms and data sources. The relationship between software is extracted. Next, the data to link the data together is tokenized to create our data list.

Algorithm 3 Data production

```

1: procedure PRODDATA(file)
2:   data ← readexecle(file, datasets from MITRE ATT&CK)
3:   techniques ← data['techniques'] ▷ Get all techniques with description, platforms,
   and datasources
4:   softwares ← data['software'] ▷ Get all software with description
5:   relationships ← data['relationships']
6:   docs ← create_list()
7:   for all technique in techniques do
8:     add_list(docs, technique → name)
9:     add_list(docs, technique → description)
10:    phrases ← sent_tokenize(technique → description) ▷ Split the description into
   phrases
11:    platforms ← technique→platforms ▷ get all tools used with techniques
12:    datasources ← technique→datasources ▷ get all tools used with techniques
13:    for all  $ph_j \in$  phrases do
14:      add_list(docs,  $ph_j$ )
15:      for all  $pf_k \in$  platforms do
16:        add_list(docs,  $ph_j + pf_k$ )
17:        add_list(docs,  $pf_k$ )
18:        for all  $ds_h \in$  platforms do
19:          add_list(docs,  $ph_j + pf_k + ds_h$ )
20:          add_list(docs,  $ph_j + ds_h$ )
21:          add_list(docs,  $ds_h$ )
22:        end for
23:      end for
24:    end for
25:  end for
26: end procedure

```

3.2.2 NLP Process Algorithm

NLP algorithms are the core components that enable computers to understand, interpret, and generate human language. Here's a high-level overview of the typical steps in an NLP process algorithm:

Text Acquisition:

Input: Raw text data from various sources such as documents, websites, social media, or user input. **Process:** Gather, collect, or retrieve the text data to be processed. Text Preprocessing:

Input: Raw text data.

Process: Clean and prepare the text data for analysis by performing tasks such as:

- **Tokenization:** Splitting text into individual words or tokens.
- **Lower Casing:** Converting all text to lowercase for uniformity.
- **Removing Stop Words:** Eliminating common, non-informative words (e.g., "the," "and," "is").
- **Stemming or Lemmatization:** Reducing words to their root form. Removing Punctuation and Special Characters.

Text Analysis: Input: Preprocessed text data.

- **Process:** Extract insights and information from the text through various NLP techniques, including:
- **Sentiment Analysis:** Determining the emotional tone of the text (positive, negative, neutral).
- **Named Entity Recognition (NER):** Identifying and classifying entities (e.g., names, locations, organizations).
- **Part-of-Speech (POS) Tagging:** Labeling words with their grammatical categories.
- **Text Classification:** Categorizing text into predefined classes or categories.
- **Topic Modeling:** Discovering underlying topics within the text using techniques like Latent Dirichlet Allocation (LDA).

Feature Extraction:

- **Input:** Analyzed text data.

- **Process:** Transform the text into numerical features that can be used for machine learning, including techniques like:
- **Bag of Words (BoW):** Representing text as a vector of word frequencies.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Assigning weights to words based on their importance.
- **Word Embeddings (e.g., Word2Vec, GloVe):** Capturing semantic relationships between words.

Machine Learning or Deep Learning:

- **Input:** Extracted features and labeled data (if applicable).
- **Process:** Train machine learning models (e.g., classification, regression, clustering) or deep learning models (e.g., recurrent neural networks, transformers) to perform specific NLP tasks, such as text classification, language translation, or chatbot responses.

Post-Processing and Interpretation:

- **Input:** Model predictions or results.
- **Process:** Interpret the model's output, perform post-processing tasks, and present the results in a human-readable format.

Feedback and Improvement:

- **Input:** User feedback and data updates.
- **Process:** Continuously refine and improve the NLP system based on user interactions and new data.

Deployment:

- **Input:** Deployable NLP model.
- **Process:** Integrate the NLP system into applications, chatbots, or platforms for end-users to interact with.

Monitoring and Maintenance:

Process: Regularly monitor the NLP system's performance, retrain models with new data, and address any issues or changes in the data distribution. The specific algorithms and techniques used in each step of the NLP process can vary depending on the task and objectives of the NLP application. NLP is a rapidly evolving field, and researchers continue to develop new algorithms and models to improve language understanding and generation capabilities.

Algorithm 4 Algorithm 2: NLP preprocess data

```

procedure NLP_CLEANDATA(docs)
  nlp ← spacy_load('en_core_web_lg')           ▷ Initialize NLP model
  cleandata ← create_list()
  reg ← ([0 - 9])|(@\[A - Za - z0 - 9]+)|([^\0 - 9A - Za - z\t])|(\w+ :
  \/\S+)|^rt|http.+?"
  for all  $d_i \in$  docs do
    doc ← regex_sub( $d_i$ , reg)
    tokens ← nlp(doc)
    temp ← create_vector()
    for all  $t_j \in$  tokens do
      if  $t_j$  is not punct, stop, alpha, and space then
        add_vector(temp,  $t_j$ )
      end if
    end for
    if temp is not None then
      add_list(cleandata, temp)
    end if
  end for
end procedure

```

3.2.3 Check Scenario Algorithm

It refers to a process or set of steps used to assess or monitor the status or progress of a particular scenario or situation. The specific algorithm can vary widely depending on the context and the nature of the scenario being monitored. It illustrates this verification method. "*mode_software1_ptedictprob(cinput + scenario)*", in step 2, which is based on estimating the probability ratio of the scenario as an ICS incident, this method is adjusted with the incident prediction model and then determines the closest incident to the scenario. If the score value is greater than or equal to 85%, it means that the scenario represents an ICS incident as shown in step 3. This also implies that all the series of techniques must belong to this incident and mark the status of this scenario as (success). If the score value is greater than or equal to 65%, this means that the series of techniques expressing the scenario can represent an incident, but some of them do not belong to the group of techniques of the incident. However, by evaluating them based on the percentage of similar techniques, this condition is called (close to success), the condition is depicted in step 6. Similarly, when the score value is greater than or equal to 50%, this indicates that there are some techniques from the scenario series that do not belong to the incident and are not similar to its set of techniques. This condition is called (init). Finally, for the rest of the cases, the scenario is considered a failure as shown in step 12, which means that the scenario cannot be

represented as an incident or even part of it, where all techniques do not belong and are not similar to one of the techniques of ICS incidents. After a user specifies the scenario status, `getGroupBySoftware(soft)` searches for the group to which this incident belongs.

Algorithm 5 Algorithm: CHECKSCENARIO

```

1: procedure CHECK_SCENARIO(scenario, cinput)
2:   rate  $\leftarrow$  mode_softwareI_ptedictprob(cinput + scenario)
3:   if rate  $\geq$  85 then
4:     return "Scenario is success"
5:   end if
6:   if rate  $\geq$  65 then
7:     return "Scenario is close to success"
8:   end if
9:   if rate  $\geq$  50 then
10:    return "Scenario is init"
11:  end if
12:  return "Scenario is Failure"
13: end procedure

```

3.2.4 General Algorithm

Refers to the architecture of the scenario generation model, which consists of two basic parts, as described.

- **The first part** is responsible for dealing with the entrance, processing, analyzing the texts of the attacks, and predicting the technique and the appropriate incident.
-
- **The second part** passes the disassembly and the incident in the tree of scenarios, which in turn works to produce a series of techniques such as an ICS attack.

The overall process of generating attack scenarios is outlined in the Algorithm 1. First of all, it calls "`input_text()`" as shown in step 1, to get the textual entry. Then, "`nlpCleanData(Input)`" performs initial processing and semantic analysis on the input text and replaces the synonyms of semantic words with attack, in step 2. In step 3, the form returns "None" if the text does not express an ICS attack. If there are outputs from the NLP model, the process of searching and predicting the appropriate technique is initiated by the "`model_technique(cinput)`" as can be observed in step 4. The prediction of the technique

and the search for the tactic to which it belongs is confirmed through "getTacticByTechnique(tech)" at step 7. Then "*model_software(Cinput,tech)*" predicts the incident that is related to the technique and entrance, in step 9. Afterwards, in step 13, the process of generating scenarios begins through "*generate_tree_scenarios(tactic,tech)*". That conducts the tree search process according to the search case, which includes the largest percentage of similarity or threshold value to determine the paths chosen as appropriate scenario paths for the entrance. Finally at step 17, "*cheak_scenario(scenario,cinput)*" checks the status of scenarios, which depends on the percentage of belonging of the series of techniques to the incident scenarios of the ICS attack.

Algorithm 6 Algorithm: ICS MODELS

```

1: Input: input ← input_text() ▷ Enter any text
2: cinput ← nlpCleanData(Input)
3: if cinput is not None then
4:   tech ← model_technique(cinput)
5:   if tech is Technique then
6:     Print: Technique : tech
7:     tactic ← getTacticByTechnique(tech)
8:     Print: start Tactic search to scenario tactic
9:     soft ← model_software(Cinput, tech)
10:    if soft is Software then
11:      Print: input same as Software is : soft
12:    end if
13:    scenarios ← generate_tree_scenarios(tactic, tech)
14:    if scenarios is not empty then
15:      states ← {}
16:      for all scenario ∈ scenarios do
17:        states[scenario] ← cheak_scenario(scenario, cinput)
18:      end for
19:      plot_analysis(scenarios, states)
20:    end if
21:  end if
22: end if

```

3.3 Extra Details for ASG Architecture

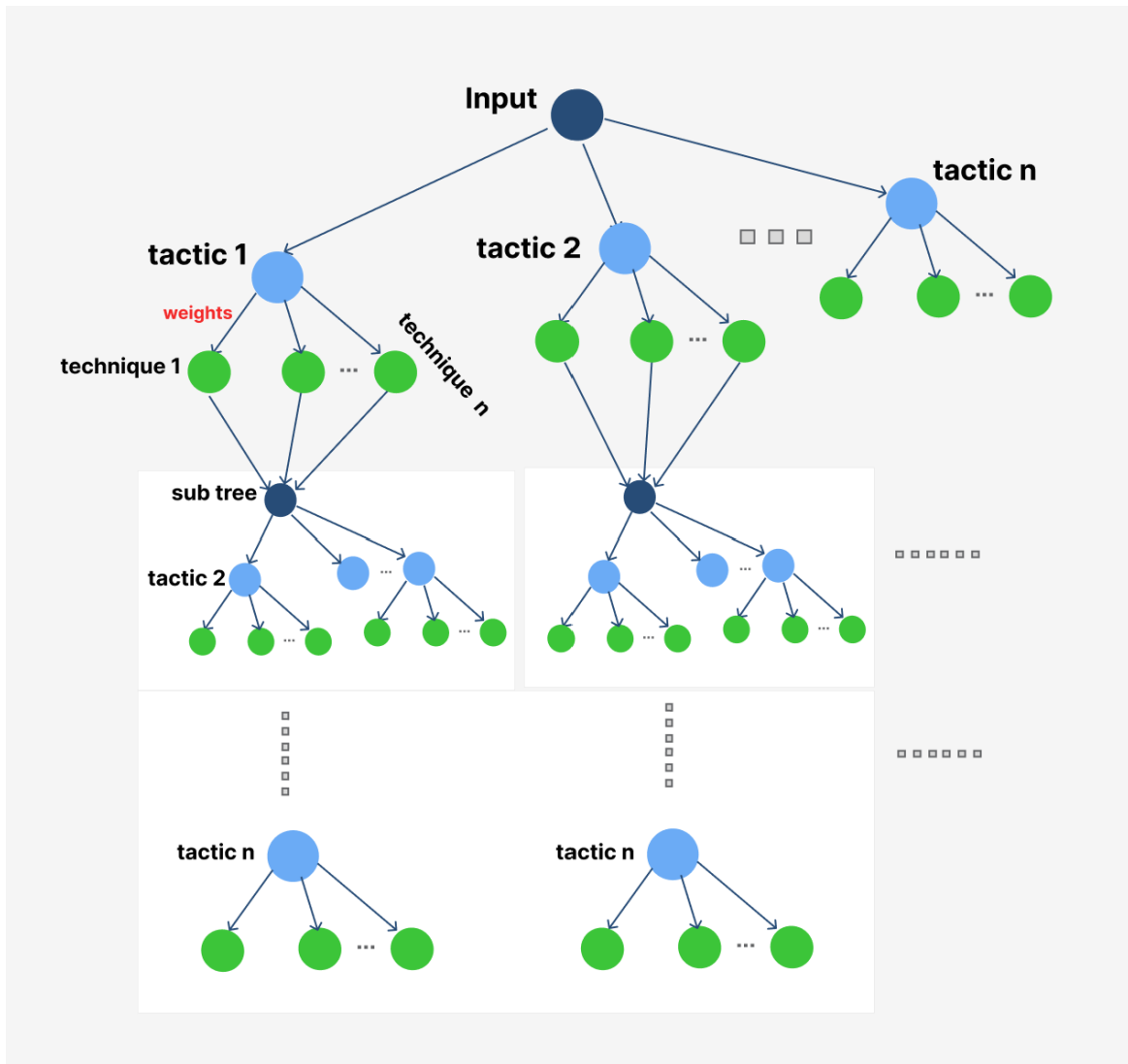


Fig. 3.1 threat based model scenario generation using tree

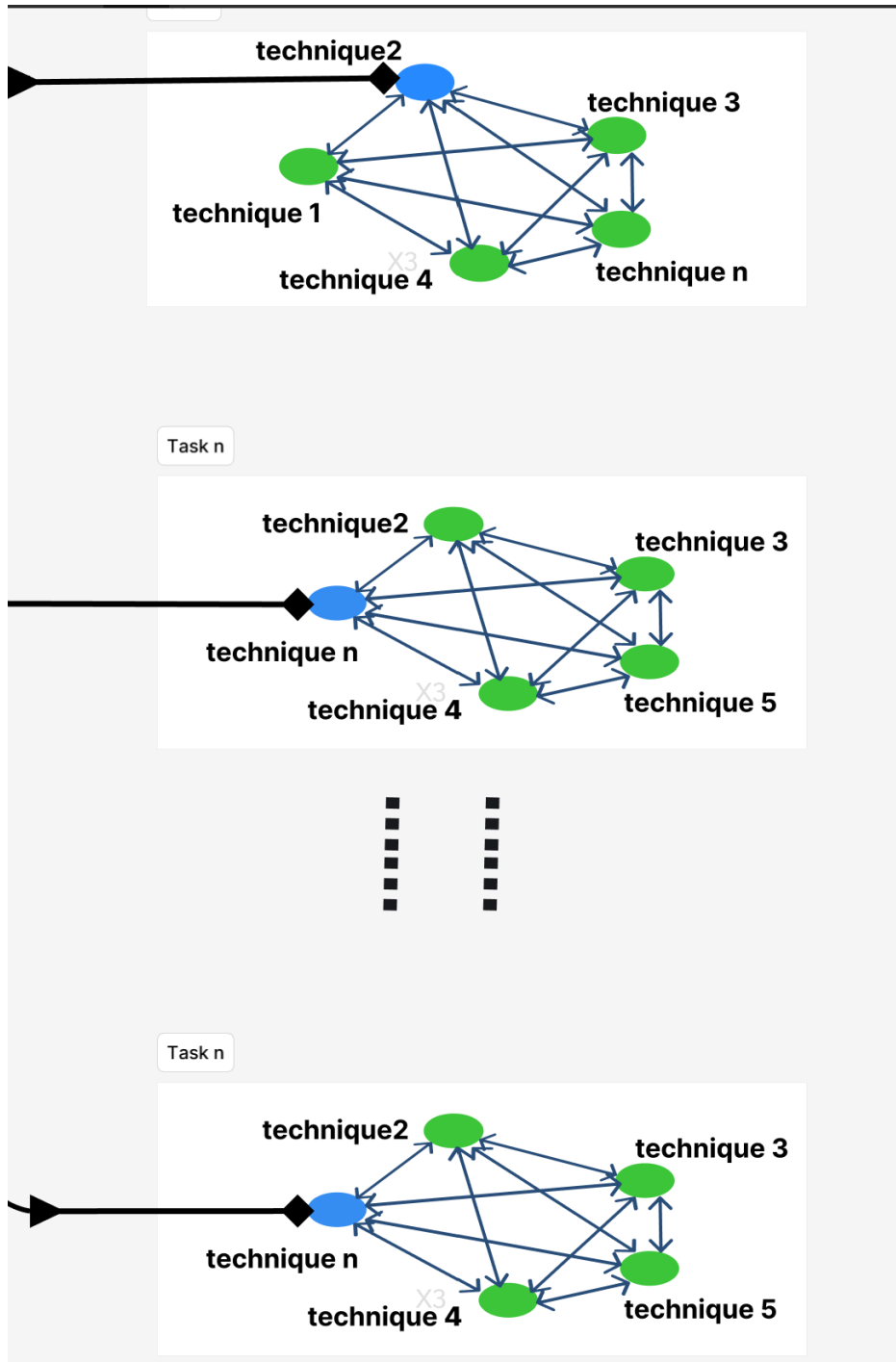


Fig. 3.2 System State Model scenario generation using Graph

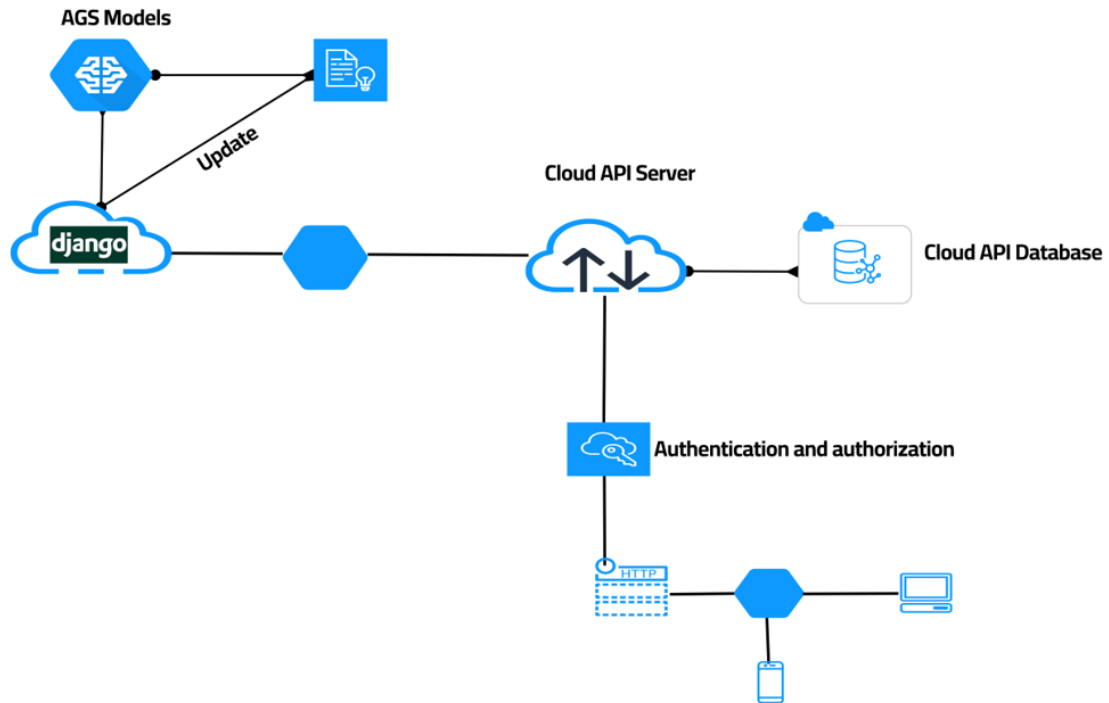


Fig. 3.3 ASG web based Architecture

Section 2

Attack scenario generation general Methodology			
State-based	Threat-based	Manual	General Methodology
Search for system information. Which includes the system state and the attacks post and pre conditions	Find a knowledge base	View the reports and Lab documentation	Gathering Raw Data
determine the the system state log	identify descriptive texts that express techniques or system	determining the tactics and attack incidents	Data Prepossessing
Connecting the techniques according to the the system state	Use NLP model to link the relationships between techniques according to similarity in terms of usage	Arrange the techniques manually through direct comparison with related attacks.	Inferential Analysis
link the techniques to system state to create mesh graph	Use Model ML to link all techniques like a sequence of trees and determine the optimal ways to represent a scenario	Choose the series of techniques that represent the scenario usage. Selection is according to previous attack policies	Structure Attack Sequences
Searching for the paths that link the techniques with the system state	Classifying the ideal paths from the scenarios tree based on the usage	mapping the techniques with ICS CKC manually	Generate Attack Sequence
A classification model that determines the percentage of belonging to the scenario as ICS incidents. And determine the APT group that the incident belongs to.	A classification model that determines the percentage of belonging to the scenario as ICS incidents. And determine the APT group that the incident belongs to.	direct comparing with previous incident	Validation

Fig. 3.4 Attack scenario Generation Methodology

Chapter 4

Data Generated for validation using AI tools

4.1 Software Generated Using Bing

Name	Description	Sources
ACAD/Medre.A	ACAD/Medre.A is a worm that infects files used by AutoCAD application, a software for computer-aided design (CAD). The worm steals AutoCAD drawings and sends them to email accounts located in China ¹²³ . It is suspected to be used for industrial espionage ²⁴ . ESET, a security company, has uncovered this worm in 2012 and has worked with Chinese ISPs and Autodesk to stop the transmission of the files ² . ESET has also made a free stand-alone	<ol style="list-style-type: none"> 1. attack.mitre.org 2. eset.com 3. virusradar.com 4. welivesecurity.com
ACAD/Medre.A	It sounds like you are describing ACAD/Medre.A, a worm that steals operational information. The worm collects AutoCAD files with drawings from infected systems and sends them by email to accounts based in China ¹² . This worm can be used for industrial espionage ³ . Some possible ways to protect yourself from this worm are: - Use an updated antivirus software and scan your system regularly. - Avoid opening suspicious attachments or links from unknown sources. - Backup your important files and data regularly. - Use strong passwords and encryption for your email accounts and files.	<ol style="list-style-type: none"> 1. scmagazine.com 2. attack.mitre.org 3. attack.mitre.org
ACAD/Medre.A	It is a worm written in AutoLISP, a dialect of the LISP programming language used in AutoCAD ¹ . It infects AutoCAD files by copying itself to the startup folder and modifying the acad.lsp, acadoc.lsp and acad.vlx files ²¹ . It uses SMTP to send the stolen files to email accounts in China ²³¹ . It has a list of 44 email accounts that it tries to use ³ .	<ol style="list-style-type: none"> 1. eset.com 2. virusradar.com 3. attack.mitre.org 4. spywareremove.com
Backdoor.Oldrea	Backdoor.Oldrea is a modular backdoor that has been used by a cyber espionage group called Dragonfly against energy companies since at least 2013 ¹ . Backdoor.Oldrea allows the attackers to extract data and install further malware on the victim's computer ⁵ . Backdoor.Oldrea may also have specialized modules to enumerate and map ICS-specific systems, processes, and protocols ¹ . Backdoor.Oldrea may be distributed via supply chain compromise or spear phishing emails ¹ . Backdoor.Oldrea may copy itself to a certain location on the infected PC and modify the Windows Registry to run automatically at startup ³⁴ .	<ol style="list-style-type: none"> 1. attack.mitre.org 2. github.com 3. spywareremove.com 4. enigmasoftware.com 5. attack.mitre.org
Backdoor.Oldrea	Backdoor.Oldrea may drop and run malicious files on the infected computer, such as a keylogger or a password stealer ¹² . Backdoor.Oldrea may gather personal information from the victimized computer user and send it to a remote attacker for malicious purposes ¹² . This may include the hostname, OS version, OS architecture, CPU information, and network configuration ³ . Backdoor.Oldrea may also collect information about the installed software on the victim's computer, such as antivirus products, firewalls, or ICS-related applications ³ .	<ol style="list-style-type: none"> 1. attack.mitre.org 2. spywareremove.com 3. attack.mitre.org 4. enigmasoftware.com
Backdoor.Oldrea	Backdoor.Oldrea is a modular backdoor that has been used by a cyberespionage group called Dragonfly against energy companies since at least 2013 ^[^1^] . Backdoor.Oldrea was distributed via supply chain compromise , and included specialized modules to enumerate and map ICS-specific systems, processes, and protocols ^[^1^] . Backdoor.Oldrea allows attackers to extract data and install further malware on the infected computer ^[^5^] . It also opens a back door for remote unauthorized access and control ^[^3^] ^[^4^] . Backdoor.Oldrea appears to be custom malware , either written by the group itself or created for it ^[^5^] .	<ol style="list-style-type: none"> 1. attack.mitre.org 2. github.com 3. spywareremove.com 4. enigmasoftware.com
Bad Rabbit	Bad Rabbit is a strain of ransomware that first appeared in 2017 and is a suspected variant of Petya ¹ . Like other strains of ransomware, Bad Rabbit virus infections lock up victims' computers, servers, or files preventing them from regaining access until a ransom—usually in Bitcoin—is paid ¹² . Bad Rabbit spreads through drive-by downloads on compromised websites, meaning victims could be exposed to the virus simply by visiting a malicious or compromised website ² . The malware is disguised as an Adobe Flash installer and is embedded into websites using JavaScript injected into the site's HTML code ¹² . If a person clicks on the malicious installer, Bad Rabbit ransomware encrypts files and presents users with an austere black-and-red message ² . The majority of incidents have been recorded in Russia and Ukraine, where it has affected news or media outlets, the Kiev Metro, Odessa airport, and other organizations ² . There have been no reports of Bad Rabbit attacks in the	<ol style="list-style-type: none"> 1. proofpoint.com 2. wired.co.uk 3. youtube.com
Bad Rabbit	It was called Bad Rabbit ¹² . It was also reported that Bad Rabbit has targeted organizations and consumers in Russia ¹ .	<ol style="list-style-type: none"> 1. attack.mitre.org 2. wired.com 3. csoonline.com 4. gettyimages.com
BlackEnergy	BlackEnergy is a malware that was first reported in 2007 as an HTTP-based toolkit that generated bots to execute distributed denial of service (DDoS) attacks ¹ It has evolved over the years to include capabilities beyond DDoS, such as cyber espionage and information destruction ² A Russian-based group known as Sandworm (aka Voodoo Bear) is attributed with using BlackEnergy in targeted attacks against industrial control systems (ICS) and energy sectors around the world ³ Some of the notable incidents involving BlackEnergy are: The 2015 cyberattack on the Ukrainian power grid, which caused a blackout for about 225,000 customers ⁴ The 2016 cyberattack on the Ukrainian central bank and other financial institutions, which destroyed data and disrupted operations ⁵ The 2017 cyberattack on the Ukrainian airport, railway, and government agencies, which	<ol style="list-style-type: none"> 1. en.wikipedia.org 2. attack.mitre.org 3. kaspersky.com 4. usa.kaspersky.com

BlackEnergy	<p>BlackEnergy is a malware toolkit that has been used by both criminal and APT actors. It dates back to at least 2007 and was originally designed to create botnets for use in conducting Distributed Denial of Service (DDoS) attacks, but its use has evolved to support various plug-ins¹</p> <p>BlackEnergy has three major versions: BlackEnergy 1 (BE1), BlackEnergy 2 (BE2), and BlackEnergy 3 (BE3). Each version has different features, capabilities, and techniques²</p> <p>BlackEnergy 2 uses modern rootkit/process-injection techniques, strong encryption and a modular architecture. It can execute local and remote files, update itself and its plugins, and execute die or destroy commands²</p> <p>BlackEnergy 3 simplifies the malware code and drops the main DLL component directly to the local application data folder. It can steal credentials, files, screenshots, and network information, execute commands and scripts,</p>	<ol style="list-style-type: none"> 1. attack.mitre.org 2. secureworks.com 3. en.wikipedia.org
Conficker	<p>Conficker is a computer worm that targets the Microsoft Windows operating system. It was first detected in November 2008 and has infected millions of computers since then¹ It exploits a vulnerability in the Windows Server service (MS08-067) and also spreads through dictionary attacks on administrator passwords and removable media² It forms a botnet that can receive commands and updates from its creators via pseudorandom URLs²</p> <p>Conficker has five variants: Conficker A, B, C, D, and E. Each variant has different features and defense mechanisms, such as blocking DNS lookups, disabling Safe Mode, deleting System Restore points, and terminating security software¹ Conficker E can also download and install other malware, such as spambots and scareware²</p> <p>Conficker was expected to cause significant damage on April 1, 2009, when it was supposed to contact its creators</p> <p>You should use an antivirus product to remove the Conficker malware from the system. Or, follow the steps in the “Manual steps to remove the Win32/Conficker virus” section of this Microsoft Knowledge Base article to manually remove the malware from the system¹</p> <p>You can also download a free removal tool from AVG that will scan your disks and try to heal the infected files. If a virus is found, you will be asked to restart your computer, and the infected file will be repaired during startup²</p> <p>Another option is to download a free removal tool from Sophos that will detect and remove Conficker variants from your home computers. Sophos has been securing business computers against Conficker for over a decade³</p> <p>To prevent Conficker infection, you should use strong and unique administrator passwords, avoid logging on with domain admin credentials, apply the latest security updates, disable the Autoplay features, and remove excessive</p>	<ol style="list-style-type: none"> 1. en.wikipedia.org 2. hellotech.com 3. eweek.com
Conficker	<p>Duqu is a collection of computer malware that was discovered in 2011 and is thought to be related to the Stuxnet worm that targeted Iran’s nuclear program¹ Duqu was designed to gather intelligence data from entities such as industrial control manufacturers in order to launch future attacks on industrial control facilities² Duqu exploited a zero-day vulnerability in Microsoft Windows and used stolen digital certificates to sign its components¹ Duqu was also linked to a sophisticated cyberattack on Kaspersky Lab in 2015, which was dubbed Duqu</p>	<ol style="list-style-type: none"> 1. support.microsoft.com 2. avg.com 3. home.sophos.com 4. techyard.net
Duqu	<p>Duqu has a nomenclature that distinguishes between its malware components, its flaw, and its operation¹</p> <p>Duqu has a unique programming language that is not known to researchers and is dubbed “Duqu framework”¹</p> <p>Duqu is nearly identical to Stuxnet in terms of code and structure, but has a different purpose of gathering intelligence rather than sabotaging systems¹</p> <p>Duqu was used in a cyberattack on Kaspersky Lab in 2015, which was detected by the company itself and named Duqu 2.0. This attack used a zero-day vulnerability in Windows to infect up to 100 computers in 30 countries^{2,3}</p>	<ol style="list-style-type: none"> 1. en.wikipedia.org 2. techopedia.com 3. kaspersky.com 4. industrialcybersecuritypulse.com
Duqu	<p>A Poison-type Pokémon introduced in Generation I that evolves into Arbok at level 22¹</p> <p>A snake-like Pokémon with purple and yellow coloring, a large mouth, and a rattle at the tip of its tail²</p> <p>A common sight in grassland and other habitats, where it moves silently and stealthily to hunt for prey, such as bird eggs^{1,2}</p>	<ol style="list-style-type: none"> 1. en.wikipedia.org 2. informationweek.com 3. resources.infosecinstitute.com
EKANS	<p>EKANS has two possible abilities: Intimidate, which lowers the opponent’s Attack stat, or Shed Skin, which heals status conditions. It can also have the hidden ability Unnerve, which prevents the opponent from using Berries^{1,2}</p> <p>EKANS is a palindrome, meaning its name is spelled the same backwards and forwards. It is also an anagram of snake²</p> <p>EKANS can learn a variety of moves, such as Poison Sting, Bite, Glare, Wrap, and Acid Spray. It can also learn some moves by TM, such as Sludge Bomb, Earthquake, and Dark Pulse^{1,2}</p>	<ol style="list-style-type: none"> 1. pokemondb.net 2. bulbapedia.bulbagarden.net 3. pokemon.fandom.com
EKANS	<p>The visible, gaseous part of a fire that is caused by a highly exothermic chemical reaction taking place in a thin zone¹</p> <p>A state of blazing combustion that can produce heat, light, and plasma. The color and temperature of a flame depend on the type of fuel and oxidizer involved in the reaction^{1,2}</p> <p>A condition or appearance suggesting a flame or burning, such as burning zeal, passion, brilliance, brightness, or a strong reddish-orange color²</p>	<ol style="list-style-type: none"> 1. bulbapedia.bulbagarden.net 2. bulbapedia.bulbagarden.net 3. pokemon.com 4. pokemon.fandom.com 5. facts.net
Flame	<p>Flame has different zones depending on the temperature and the fuel/oxidizer ratio. The hottest zone is the blue inner core, where complete combustion occurs. The yellow outer zone is where incomplete combustion produces soot and other radicals. The dark zone is where unburned gases escape from the fuel source¹</p> <p>Flame can produce plasma, which is a state of matter consisting of ionized gas particles. Plasma can emit light of different colors depending on the temperature and the chemical composition of the flame^{1,2}</p> <p>Flame can have different effects on the environment and human activities. Fire can stimulate growth, maintain</p>	<ol style="list-style-type: none"> 1. en.wikipedia.org 2. merriam-webster.com 3. britannica.com
Flame	<p>A malware framework that was used in the cyberattack on Ukraine’s power grid on December 17, 2016, which caused a blackout in Kyiv for one hour^{1,2}</p> <p>The first malware ever seen to have been specifically designed to attack electrical grids. It targets industrial control systems, especially those used in electrical substations, and exploits their communication protocols^{1,3}</p> <p>A modular malware that consists of a main backdoor, an additional backdoor, a launcher component, four</p>	<ol style="list-style-type: none"> 1. en.wikipedia.org 2. en.wikipedia.org
Industroyer	<p>A malware framework that was used in the cyberattack on Ukraine’s power grid on December 17, 2016, which caused a blackout in Kyiv for one hour^{1,2}</p> <p>The first malware ever seen to have been specifically designed to attack electrical grids. It targets industrial control systems, especially those used in electrical substations, and exploits their communication protocols^{1,3}</p> <p>A modular malware that consists of a main backdoor, an additional backdoor, a launcher component, four</p>	<ol style="list-style-type: none"> 1. eset.com 2. en.wikipedia.org 3. kaspersky.com

4.2 Group Generated Using Bing

Name	Descriptions	Sources
ALLANITE	Allanite is a mineral that belongs to the sorosilicate group of minerals within the broader epidote group. It contains a significant amount of rare-earth elements, such as cerium, lanthanum, neodymium and yttrium. It is usually black or brown in color and has a hardness of 5.5–6 on the Mohs scale. It is often radioactive due	1. en.wikipedia.org 2. mindat.org 3. alexstrekeisen.it 4. attack.mitre.org
ALLANITE	Allanite is a monoclinic silicate mineral allied to epidote and has the general formula (Ce,Ca,Y,La) 2 (Al,Fe +3) 3 (SiO 4) 3 (OH). It is also called orthite1. Allanite is often coated with a yellow-brown alteration product, likely limonite, and has a grey streak. It is also pyrognomic, meaning that it becomes	1. merriam-webster.com 2. en.wikipedia.org 3. crystallographygems.com 4. mindat.org
APT33	APT33 is a suspected Iranian threat group that has carried out cyber espionage operations since at least 2013. The group has targeted organizations across multiple industries in the United States, Saudi Arabia, and South Korea, with a particular interest in the aviation and energy sectors1	1. attack.mitre.org 2. en.wikipedia.org 3. mandiant.com 4. brandefense.io
APT33	APT33 has been linked to the Shamoon malware attacks in 2018, which wiped data from thousands of computers in the Middle East and Europe. The group may have used Shamoon as a retaliation for sanctions imposed on Iran12 APT33 has used domain generation algorithms (DGAs) to create dynamic C2 domains that are difficult to block or track. The group has also used	1. cfr.org 2. boozallen.com 3. mandiant.com 4. brandefense.io 5. attack.mitre.org
APT33	APT38 is a North Korean state-sponsored threat group that specializes in financial cyber operations. The group has been attributed to the Reconnaissance General Bureau, North Korea’s intelligence agency1 APT38 has been active since at least 2014 and has targeted banks, financial institutions, casinos, cryptocurrency exchanges, SWIFT system endpoints, and	1. attack.mitre.org 2. mandiant.com
APT33	APT38 is also known as Nickel Gladstone, BeagleBoyz, Bluenoroff, and Stardust Chollima. These names may reflect different subgroups or clusters within the broader APT38 umbrella1 APT38 has used a malware tool called EASTCash to compromise payment switch	1. avertium.com 2. attack.mitre.org 3. mandiant.com
Dragonfly	A flying insect belonging to the infraorder Anisoptera below the order Odonata. Dragonflies have two pairs of transparent wings, large compound eyes, and elongated bodies. They are predatory insects that feed on other flying insects, and are found near freshwater habitats throughout most of the world12	1. en.wikipedia.org 2. britannica.com 3. nationalgeographic.com 4. gettyimages.com
Dragonfly	Dragonflies have a life cycle that consists of three stages: egg, nymph, and adult. The eggs are laid in or near water, and hatch into aquatic nymphs that undergo several molts before emerging as winged adults. The nymphs have extendable jaws that they use to catch prey, such as tadpoles, fish, and other insects. The	1. funfactoday.com 2. britannica.com 3. en.wikipedia.org
FIN6	FIN6 is a cybercrime group that has stolen payment card data and sold it for profit on underground marketplaces. This group has aggressively targeted and compromised point of sale (PoS) systems in the hospitality and retail sectors1	1. attack.mitre.org 2. zdnet.com 3. zdnet.com
FIN7	FIN6 has been active since 2015 and is primarily known for targeting companies operating high-traffic POS (Point-of-Sale) payment terminals, where it compromises internal networks to install POS malware that steals payment card information. The group has made millions of dollars by selling the stolen card	1. know.netenrich.com 2. attack.mitre.org
FIN8	FIN7 is a financially-motivated threat group that has been active since 2013, primarily targeting the U.S. retail, restaurant, and hospitality sectors, often using point-of-sale malware to steal payment card data. The group has also sold sensitive information stolen from companies on underground forums12	1. en.wikipedia.org 2. attack.mitre.org 3. techrepublic.com
FIN8	FIN7 has been called one of the most successful criminal hacking groups in the world, having stolen millions of customer credit and debit card numbers from over 3,600 business locations in 47 states and the District of Columbia, as well as abroad. The group has made millions of dollars by selling the stolen card data on	1. en.wikipedia.org 2. fbi.gov 3. attack.mitre.org 4. wired.com
GOLD SOUTHFIELD	GOLD SOUTHFIELD is a cybercriminal threat group that operates the REvil ransomware on behalf of various affiliates12. It has been active since at least 2018 and has targeted various sectors and regions with its malware campaigns1. It is also known as Pinchy Spider1.	1. attack.mitre.org 2. malpedia.caad.fkie.fraunhofer.de
GOLD SOUTHFIELD	They are also known as Pinchy Spider1, a name derived from their use of spider-themed names for their malware and infrastructure2. They operate the REvil ransomware as a service (RaaS), meaning they provide the malware and the backend infrastructure to affiliates who conduct the actual	1. attack.mitre.org 2. malpedia.caad.fkie.fraunhofer.de

HEXANE	HEXANE is a chemical compound with the molecular formula C ₆ H ₁₄ . It is a straight-chain alkane with six carbon atoms and 14 hydrogen atoms ¹ . It is a colorless and odorless liquid when pure, and it has a boiling point of approximately 69 °C ¹ . It is widely used as a non-polar solvent in various	1. en.wikipedia.org 2. sigmaaldrich.com 3. byjus.com 4. vedantu.com
HEXANE	It has five isomers, which are molecules that have the same molecular formula but different structures. The isomers of HEXANE are 2-methylpentane, 3-methylpentane, 2,2-dimethylbutane, 2,3-dimethylbutane, and n-hexane ¹ . The last one is the most common and the simplest form of HEXANE ^{1,2} .	1. en.wikipedia.org 2. vedantu.com 3. beyondthemagazine.com
Lazarus Group	Lazarus Group is a cybercrime group that is believed to be run by the North Korean government ¹² . The group has been active since at least 2009 and has been involved in many cyberattacks against various targets, such as governments, banks, media, and cryptocurrency platforms ^{12,3} . The group uses a	1. en.wikipedia.org 2. attack.mitre.org 3. makeuseof.com
Lazarus Group	The group is also known by other monikers, such as Guardians of Peace, Whois Team, Hidden Cobra, Zinc, and Nickel Academy ¹² . These names are given by different cybersecurity organizations or the group itself to disguise its identity or claim responsibility for its attacks ¹² .	1. en.wikipedia.org 2. bbc.com 3. darkreading.com 4. theregister.com
OilRig	OilRig is a cyber threat group that is suspected to be linked to the Iranian government ¹² . The group has been active since at least 2015 and has targeted Middle Eastern and international victims in various sectors, such as financial, government, energy, chemical, and telecommunications ¹² . The group's main	1. attack.mitre.org 2. intights.com 3. oasdom.com
OilRig	The group is also known by other names, such as Cobalt Gypsy, Twisted Kitten, Crambus, and Helix Kitten ¹² . These names are given by different cybersecurity organizations or the group itself to disguise its identity or claim responsibility for its attacks ¹ . The group is believed to be linked to the Iranian Intelligence agency and the	1. intights.com 2. medium.com 3. attackiq.com 4. oasdom.com 5. en.wikipedia.org
Sandworm Team	Sandworm Team is a destructive Russian-based threat group that has been attributed to Russian GRU Unit 74455 by the United States Department of Justice and United Kingdom National Cyber Security Centre ¹² . The group has been active since at least 2009 and has been involved in many cyberattacks against	1. attack.mitre.org 2. en.wikipedia.org 3. docs.rapid7.com
Sandworm Team	The group is also known by other names, such as Voodoo Bear, Iron Viking, Telebots, and Fancy Bear ^{12,3} . These names are given by different cybersecurity organizations or the group itself to disguise its identity or claim responsibility for its attacks ^{1,2,3} .	1. docs.rapid7.com 2. attack.mitre.org 3. en.wikipedia.org
TEMP.Veles	TEMP.Veles is a Russia-based threat group that has targeted critical infrastructure ^{12,3} . The group has been observed utilizing TRITON, a malware framework designed to manipulate industrial safety systems ^{12,3} . The group's main goal is to conduct cyber espionage and sabotage its adversaries' interests	1. attack.mitre.org 2. malpedia.caad.fkie.fraunhofer.de 3. lansinginstitute.org
TEMP.Veles	The group is also known by other names, such as XENOTIME, ATK91, and G0088 ^{12,3} . These names are given by different cybersecurity organizations or the group itself to disguise its identity or claim responsibility for its attacks ^{12,3} . The group is believed to be linked to the GRU and is the primary threat to the	1. whereandwhen.net 2. attack.mitre.org 3. malpedia.caad.fkie.fraunhofer.de
Wizard Spider	Wizard Spider is a Russia-based financially motivated threat group that has been active since at least 2016 ¹² . The group is known for the creation and deployment of TrickBot, a malware that can steal banking credentials, harvest emails, and	1. en.wikipedia.org 2. attack.mitre.org 3. malpedia.caad.fkie.fraunhofer.de
Wizard Spider	The group is also known by other names, such as Trickbot, Cobalt Spider, and G0102 ^{12,3} . These names are given by different cybersecurity organizations or the group itself to disguise its identity or claim responsibility for its attacks ^{12,3} . The group is based in and around Saint Petersburg in Russia, and some members	1. attack.mitre.org 2. theregister.com 3. en.wikipedia.org 4. attackevals.mitre-engenuity.org

4.3 Software Generated Using Bing

Name	Description	Sources
ACAD/Medre.A	<p>ACAD/Medre.A is a worm that infects files used by AutoCAD application, a software for computer-aided design (CAD). The worm steals AutoCAD drawings and sends them to email accounts located in China¹²³. It is suspected to be used for industrial espionage²⁴. ESET, a security company, has uncovered this worm in 2012 and has worked with Chinese ISPs and Autodesk to stop the transmission of the files². ESET has also made a free stand-alone cleaner available at: http://download.eset.com/special/ESETACADMedreCleaner.exe².</p>	<ol style="list-style-type: none"> 1. attack.mitre.org 2. eset.com 3. virusradar.com 4. welivesecurity.com
ACAD/Medre.A	<p>It sounds like you are describing ACAD/Medre.A, a worm that steals operational information. The worm collects AutoCAD files with drawings from infected systems and sends them by email to accounts based in China¹². This worm can be used for industrial espionage³. Some possible ways to protect yourself from this worm are:</p> <ul style="list-style-type: none"> - Use an updated antivirus software and scan your system regularly. - Avoid opening suspicious attachments or links from unknown sources. - Backup your important files and data regularly. - Use strong passwords and encryption for your email accounts and files. 	<ol style="list-style-type: none"> 1. scmagazine.com 2. attack.mitre.org 3. attack.mitre.org
ACAD/Medre.A	<p>It is a worm written in AutoLISP, a dialect of the LISP programming language used in AutoCAD¹. It infects AutoCAD files by copying itself to the startup folder and modifying the acad.lsp, acadoc.lsp and acad.vlx files²¹. It uses SMTP to send the stolen files to email accounts in China²³¹. It has a list of 44 email accounts that it tries to use³. It was first detected in 2012 and mainly affected users in Peru and neighboring countries¹⁴. It can be removed by using an updated antivirus software and deleting the infected files²⁴.</p>	<ol style="list-style-type: none"> 1. eset.com 2. virusradar.com 3. attack.mitre.org 4. spywareremove.com
Backdoor.Oldrea	<p>Backdoor.Oldrea is a modular backdoor that has been used by a cyber espionage group called Dragonfly against energy companies since at least 2013¹. Backdoor.Oldrea allows the attackers to extract data and install further malware on the victim's computer⁵. Backdoor.Oldrea may also have specialized modules to enumerate and map ICS-specific systems, processes, and protocols¹. Backdoor.Oldrea may be distributed via supply chain compromise or spear phishing emails¹. Backdoor.Oldrea may copy itself to a certain location on the infected PC and modify the Windows Registry to run automatically at startup³⁴. Backdoor.Oldrea is also known as Havex or Energetic Bear RAT⁵. Backdoor.Oldrea is considered a threat to the western energy sector¹⁵.</p>	<ol style="list-style-type: none"> 1. attack.mitre.org 2. github.com 3. spywareremove.com 4. enigmasoftware.com 5. attack.mitre.org
Backdoor.Oldrea	<p>Backdoor.Oldrea may drop and run malicious files on the infected computer, such as a keylogger or a password stealer¹². Backdoor.Oldrea may gather personal information from the victimized computer user and send it to a remote attacker for malicious purposes¹². This may include the hostname, OS version, OS architecture, CPU information, and network configuration³. Backdoor.Oldrea may also collect information about the installed software on the victim's computer, such as antivirus products, firewalls, or ICS-related applications³. Backdoor.Oldrea may communicate with a command and control (C2) server using HTTP or HTTPS protocols³. The C2 server may send commands to Backdoor.Oldrea to download and execute additional malware, update itself, or uninstall itself³.</p>	<ol style="list-style-type: none"> 1. attack.mitre.org 2. spywareremove.com 3. attack.mitre.org 4. enigmasoftware.com
Backdoor.Oldrea	<p>Backdoor.Oldrea is a modular backdoor that has been used by a cyberespionage group called Dragonfly against energy companies since at least 2013^[^1^]. Backdoor.Oldrea was distributed via supply chain compromise, and included specialized modules to enumerate and map ICS-specific systems, processes, and protocols^[^1^]. Backdoor.Oldrea allows attackers to extract data and install further malware on the infected computer^[^5^]. It also opens a back door for remote unauthorized access and control^[^3^] ^[^4^]. Backdoor.Oldrea appears to be custom malware, either written by the group itself or created for it^[^5^]. Backdoor.Oldrea is also known as</p>	<ol style="list-style-type: none"> 1. attack.mitre.org 2. github.com 3. spywareremove.com 4. enigmasoftware.com

Bad Rabbit	<p>Bad Rabbit is a strain of ransomware that first appeared in 2017 and is a suspected variant of Petya¹. Like other strains of ransomware, Bad Rabbit virus infections lock up victims' computers, servers, or files preventing them from regaining access until a ransom—usually in Bitcoin—is paid¹². Bad Rabbit spreads through drive-by downloads on compromised websites, meaning victims could be exposed to the virus simply by visiting a malicious or compromised website². The malware is disguised as an Adobe Flash installer and is embedded into websites using JavaScript injected into the site's HTML code¹². If a person clicks on the malicious installer, Bad Rabbit ransomware encrypts files and presents users with an austere black-and-red message².</p> <p>The majority of incidents have been recorded in Russia and Ukraine, where it has affected news or media outlets, the Kiev Metro, Odessa airport, and other organizations². There have been no reports of Bad Rabbit attacks in the UK².</p> <p>Bad Rabbit is not related to Bad Bunny, who is a Puerto Rican rapper and singer³.</p>	<ol style="list-style-type: none"> 1. proofpoint.com 2. wired.co.uk 3. youtube.com
Bad Rabbit	<p>It was called Bad Rabbit¹². It was also reported that Bad Rabbit has targeted organizations and consumers in Russia¹.</p>	<ol style="list-style-type: none"> 1. attack.mitre.org 2. wired.com 3. csoonline.com 4. gettyimages.com
BlackEnergy	<p>BlackEnergy is a malware that was first reported in 2007 as an HTTP-based toolkit that generated bots to execute distributed denial of service (DDoS) attacks¹ It has evolved over the years to include capabilities beyond DDoS, such as cyber espionage and information destruction² A Russian-based group known as Sandworm (aka Voodoo Bear) is attributed with using BlackEnergy in targeted attacks against industrial control systems (ICS) and energy sectors around the world³</p> <p>Some of the notable incidents involving BlackEnergy are:</p> <p>The 2015 cyberattack on the Ukrainian power grid, which caused a blackout for about 225,000 customers⁴</p> <p>The 2016 cyberattack on the Ukrainian central bank and other financial institutions, which destroyed data and disrupted operations⁵</p> <p>The 2017 cyberattack on the Ukrainian airport, railway, and government agencies, which deployed a destructive payload called NotPetya that encrypted and wiped data from infected machines.</p> <p>BlackEnergy uses various techniques to infect target machines, such as phishing emails with malicious attachments, compromised websites, and removable media. It also uses a modular architecture that allows it to download and execute plugins from its command and control (C2) servers. Some of the plugins enable BlackEnergy to:</p> <ul style="list-style-type: none"> Steal credentials, files, screenshots, and network information Execute commands and scripts Kill processes and services Wipe files and disks Interact with ICS devices and protocols Encrypt data and display ransom notes <p>BlackEnergy is a sophisticated and persistent threat that poses a significant risk to critical infrastructure and national security. It is important to implement security measures to prevent, detect, and respond to BlackEnergy attacks, such as:</p> <p>Educating users about phishing and social engineering</p>	<ol style="list-style-type: none"> 1. en.wikipedia.org 2. attack.mitre.org 3. kaspersky.com 4. usa.kaspersky.com
BlackEnergy	<p>BlackEnergy is a malware toolkit that has been used by both criminal and APT actors. It dates back to at least 2007 and was originally designed to create botnets for use in conducting Distributed Denial of Service (DDoS) attacks, but its use has evolved to support various plug-ins¹</p> <p>BlackEnergy has three major versions: BlackEnergy 1 (BE1), BlackEnergy 2 (BE2), and BlackEnergy 3 (BE3). Each version has different features, capabilities, and techniques²¹³</p> <p>BlackEnergy 2 uses modern rootkit/process-injection techniques, strong encryption and a modular architecture. It can execute local and remote files, update itself and its plugins, and execute die or destroy commands²</p> <p>BlackEnergy 3 simplifies the malware code and drops the main DLL component directly to the local application data folder. It can steal credentials, files, screenshots, and network information, execute commands and scripts, kill processes and services, wipe files and disks, interact with ICS devices and protocols, encrypt data and display ransom notes³</p>	<ol style="list-style-type: none"> 1. attack.mitre.org 2. secureworks.com 3. en.wikipedia.org

4.4 Techniques Generated

Name	Technology	Description	Sources
Activate Firmware Update Mode	Bard google	<p>Locate the firmware update mode button or combination. This button or combination is usually found on the device's control panel or in the device's documentation.</p> <p>Turn off the device. This is important to do before entering firmware update mode, as it will prevent any data loss.</p> <p>Press and hold the firmware update mode button or combination. Hold the button or combination for the amount of time specified in the device's documentation.</p> <p>The device will enter firmware update mode. You will usually see a message on the device's display indicating that it is in firmware update mode.</p> <p>Connect the device to a computer. You will need to connect the device to a computer in order to update the firmware.</p> <p>Use the appropriate software to update the firmware. The software to update the firmware will be provided by the device manufacturer.</p> <p>Once the firmware has been updated, the device will automatically exit firmware update mode.</p>	
Activate Firmware Update Mode	Bard google	<p>Make sure that the device is properly connected to the computer before you start the update process.</p> <p>If the device does not enter firmware update mode, try pressing and holding the button or combination for a longer period of time.</p> <p>If you are still having trouble activating firmware update mode, consult the device's documentation or contact the device manufacturer for assistance.</p>	
Alarm Suppression	Bard google	<p>Alarm suppression is a technique used to temporarily disable or silence alarms. This can be useful in a number of situations, such as:</p> <p>To prevent nuisance alarms from flooding the operator's console. Nuisance alarms are alarms that are not critical and do not require immediate attention. They can often be caused by transient conditions, such as a momentary power outage or a spike in the process variable.</p> <p>To allow for maintenance or troubleshooting without being interrupted by alarms. When an operator is performing maintenance or troubleshooting, they may need to disable alarms that would otherwise distract them.</p> <p>To prevent alarms from being triggered by transient conditions. As mentioned above, nuisance alarms can often be caused by transient conditions. Alarm suppression can be used to prevent these alarms from being triggered, which can help to keep the operator's console clear of unnecessary alarms.</p> <p>There are a number of different ways to implement alarm suppression. Some common methods include:</p> <p>Shelving: This is a manual method of suppressing alarms. The operator can temporarily disable an alarm by shelving it. Shelved alarms can be easily restored by the operator.</p> <p>Time-based suppression: This method suppresses alarms for a specified period of time. This can be useful for suppressing nuisance alarms that are only temporary. For example, an alarm that is triggered by a power outage may only need to be suppressed for a few minutes.</p> <p>Condition-based suppression: This method suppresses alarms based on certain conditions. For example, an alarm can be suppressed if the process variable is within a specified range.</p> <p>Alarm suppression can be a valuable tool for managing alarms. However, it is important to use it carefully. If alarms are suppressed too often, operators may miss important warnings.</p>	
Alarm Suppression	Bard google	<ul style="list-style-type: none"> * Use alarm suppression sparingly. Only suppress alarms that are not critical. * Restore suppressed alarms as soon as possible. * Document all alarm suppressions. <p>Alarm suppression can be a complex topic, but it is an important tool for managing alarms. By understanding the different methods of alarm suppression and the best practices for using it, operators can help to ensure that they are not overwhelmed by nuisance alarms and that they are able to respond to critical alarms in a timely manner.</p>	
Automated Collection	Bard google	<p>Automated collection is the process of collecting data or information automatically, without the need for human intervention. This can be done through a variety of methods, such as:</p> <p>Sensors: Sensors can be used to collect data about physical conditions, such as temperature, pressure, or humidity.</p> <p>Scanners: Scanners can be used to collect data from documents, such as barcodes, images, or text.</p> <p>Web scraping: Web scraping can be used to collect data from websites.</p> <p>APIs: APIs can be used to collect data from other applications.</p>	

Automated Collection	Bard google	<p>Automated collection can be used in a variety of industries, such as manufacturing, healthcare, and logistics. It can help to improve efficiency, reduce costs, and improve accuracy.</p> <p>Here are some of the benefits of automated collection:</p> <p>Increased efficiency: Automated collection can free up human resources to focus on other tasks, such as analysis and decision-making.</p> <p>Reduced costs: Automated collection can help to reduce costs by eliminating the need for manual data entry.</p> <p>Improved accuracy: Automated collection can help to improve accuracy by reducing the risk of human error.</p> <p>Real-time data: Automated collection can provide real-time data, which can be used to make informed decisions.</p> <p>Here are some of the challenges of automated collection:</p> <p>Data quality: The quality of the data collected can be affected by the quality of the sensors, scanners, or other devices used.</p> <p>Security: The security of the data collected can be a concern, especially if the data is sensitive.</p> <p>Compliance: The data collected may need to be compliant with certain regulations, such as those governing privacy or data protection.</p>	
Block Command Message	Bard google	<p>The block command message is a security feature that can be used to prevent unauthorized users from sending commands to a device. When a block command message is received, the device will not execute the command. This can help to protect the device from malicious attacks.</p> <p>The block command message can be used in a variety of scenarios. For example, it can be used to prevent unauthorized users from sending commands that could shut down a device or modify its configuration. It can also be used to prevent users from sending commands that could access sensitive data.</p>	
Block Command Message	Bard google	<p>The block command message is a valuable security feature that can help to protect devices from malicious attacks. If you are concerned about the security of your devices, you may want to consider enabling the block command message feature.</p> <p>Here are some additional details about the block command message:</p> <p>The block command message is typically sent by a network administrator or security officer.</p> <p>The block command message can be sent to a specific device or to a group of devices.</p> <p>The block command message can be configured to expire after a certain amount of time.</p> <p>The block command message can be overridden by an authorized user.</p>	
Block Reporting Message	Bard google	<p>A block reporting message is a security feature that can be used to prevent unauthorized users from sending reporting messages to a device. When a block reporting message is received, the device will not send the reporting message. This can help to protect the device from malicious attacks.</p> <p>The block reporting message can be used in a variety of scenarios. For example, it can be used to prevent unauthorized users from sending reporting messages that could contain sensitive data. It can also be used to prevent users from sending reporting messages that could be used to track their activity.</p>	
Block Reporting Message	Bard google	<p>The block reporting message is a valuable security feature that can help to protect devices from malicious attacks. If you are concerned about the security of your devices, you may want to consider enabling the block reporting message feature.</p> <p>Here are some additional details about the block reporting message:</p> <p>The block reporting message is typically sent by a network administrator or security officer.</p> <p>The block reporting message can be sent to a specific device or to a group of devices.</p> <p>The block reporting message can be configured to expire after a certain amount of time.</p> <p>The block reporting message can be overridden by an authorized user.</p>	