COoperative Cyber prOtectiOn for modern power grids

D1.2 Threat Models, Vulnerability Assessment and Risk Profiling

Distribution Level	PU
Responsible Partner	UCY
Prepared by	Dimitris Theocharides, Philippos Isaia, Michael Photiades, Irina Ciornei, Angelos Marnerides
Checked by WP Leader	UCY
Verified by Reviewer #1	Alex Stefanov (TUD) 05/09/2024
Verified by Reviewer #2	Charis Demoulias (AUTH) 15/09/2024
Approved by Project Coordinator	Angelos Marnerides (UCY) 16/09/2024

Co-funded by the European Union

Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Directorate General for Communications Networks, Content and Technology. Neither the European Union nor the Directorate General for Communications Networks, Content and Technology can be held responsible for them.

Deliverable Record

Planned Submission Date	17/09/2024
Actual Submission Date	16/09/2024
Status and version	FINAL

Version	Date	Author(s)	Notes
0.1 (Draft)	01/07/2024	Angelos Marnerides, Irina Ciornei	Initial Structure – ToC
0.2 (Draft)	06/08/2024	Dimitris Theocharides, Philipos Isaia, Irina Ciornei (UCY), Elvira Sanchez Ortiz (ENCS), Georgios Kryonides (AUTH), David Senas Sanvicente (ING), Iordanis Lazoudis (SEL)	Updates of Chapters, 2, 3, 4, and 5
0.3 (Draft)	23/08/2024	Michael Photiades, Dimitris Theocharides, Philippos Isaia, Irina Ciornei (UCY)	Updates of Chapters 1, 3, 4, 5 and 6
0.4	29/08/2024	Irina Ciornei (UCY)	Full integrated draft, edits in all sections
0.5	05/09/2024	Alex Stefanov (TUD), Angelos Marnerides (UCY)	Comments from the 1st reviewer to the full integrated draft
0.6	10/09/2024	Dimitris Theocharides, Philippos Isaia, Irina Ciornei	Updates to address all comments from the 1st reviewer to the full integrated draft
0.7	15/09/2024	Charis Demoulias (AUTH)	Comments from the 2nd reviewer
0.8	16/09/2024	Irina Ciornei (UCY), Angelos Marnerides (UCY)	Updates and final quality checks
1.0 (Final)	16/09/2024	Angelos Marnerides (UCY)	Final quality updates before submission

Table of contents

Definition of Acronyms

ACK	ACKnowledgement
AiTM	Adversary in The Middle
AMI	Advanced Metering Infrastructure
API	Application Programming Interface
APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
ASDU	Application Service Data Unit
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
AV	Attack Vector
BEC	Business Email Compromise
C&C	Command and Control
CKC	Cyber Kill Chain
CNN	Convolutional Neural Networks
COCOON	COoperative Cyber prOtectiON for modern power grids
COMML	COntrol, Measurement, and Monitoring Layer
CPN	COCOON Programable Node
CSL	Cybersecurity Services Layer
CTI	Cyber Threat Intelligence
CTD	COCOON Toolset Dashboard
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DAG	Directed Acyclic Graph
DCS	Distributed Control Systems
DDoS	Distributed Denial of Service
DNP3	Distributed Network Protocol 3
DNS	Domain Name System
DRES	Distributed Renewable Energy Source
DRL	Deep Reinforcement Learning
EMS	Energy Management Systems
EPES	Electrical and Power Energy Systems
EWS	Early Warning System
FAIR	Factor Analysis of Information Risk
FDI	False Data Injection
GOOSE	Generic Object-Oriented Substation Event
HAZOP	Hazard and Operability Study
HMI	Human Machine Interface
ICMP	Internet Control Message Protocol
ICS	Industrial Control System
IEC	International Electrotechnical Commission
ID	Identifier
IED	Intelligent Electronic Device
IOC	Indicator Of Compromise
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System

ISAC	Information Sharing and Analysis Center
IT	Information Technology
SAS	Substation Automation Systems
SCADA	Supervisory Control and Data Acquisition
SDG	SCADA Data Gateway
SMB	Server Message Block
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
SSL-TLS	Secure Sockets Layer-Transport Layer Security
SVM	Support Vector Machine
SYN	SYNchronize
LAN	Local Area Network
LOLBins	Living Off the Land Binaries
LSTM	Long Short-Term Memory
MFA	Multi-Factor Authentication
MiTM	Man in The Middle
MITRE	Massachusetts Institute of Technology Research and Engineering
ML	Machine Learning
MV	Medium Voltage
MV/LV	Medium/Low Voltage
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NVD	National Vulnerability Database
OS	Operating System
OSINT	Open-Source Intelligence
OT	Operational Technology
OTX	Open Threat eXchange
PDP	Programmable Data Plane
PLC	Programmable Logic Controller
PS	Packet Sniffer
PV	PhotoVoltaic
RBAC	Role-Based Access Control
RTU	Remote Terminal Unit
TCP	Transmission Control Protocol
TTP	Tactic Technique Procedure
UDP	User Datagram Protocol
VM	Virtual Machine
VPN	Virtual Private Network
XSS	Cross-Site Scripting

List of Figures

Executive Summary

The growing interconnection and digital transformation of Industrial Control Systems (ICS) within Electric Power and Energy Systems (EPES) has dramatically increased the cyber threat spectrum in such setups. Hence, they underscore the need for robust threat modelling and vulnerability assessment and scoring to safeguard these critical infrastructures. This COCOON deliverable, being the second WP1 technical report, presents a comprehensive framework for threat models fully aligned with cybersecurity and penetration testing industry procedures via the MITRE ATT&CK Framework, specifically tailored for ICS within Electric Power and Energy Systems (EPES). The COCOON threat modelling framework covers taxonomy and definitions, emphasizing the rational and importance of managing cyber vulnerabilities in EPES ICS. Key components of ICS threat models are detailed, including methodologies and practical implementations for Advanced Persistent Threat (APT) modelling. The COCOON threat modelling framework is validated through proof-of-concept examples tailored to the COCOON pilots, demonstrating its applicability in real-world scenarios.

This document also provides a detailed description of the COCOON's Early Warning System (EWS), which underpins the risk profiling and vulnerability assessment processes that are the main focus in this deliverable. The EWS architecture comprises of: (i) the data collection layer, (ii) the data processing and analysis layer, (iii) the decision-making layer, (iv) the communication and response layer, and (v) a continuous monitoring loop. Within this deliverable, the EWS is illustrated through three practical implementations in order to proof its practicality in actual ICS EPES context, such as: (1) the detection of a Distributed Denial of Service (DDoS) attack, from initial sensor detection to the execution of mitigation strategies; (2) the early detection of a botnet infection, starting from the identification of an Internet of Things (IoT) device infected by the botnet to containment measures to prevent botnet spreading; (3) identifying insider threats, beginning with the analysis of suspicious behavior of employee accounts and culminating in measures to protect sensitive data from unauthorized access. The COCOON EWS incorporates the BotPro framework, which is an evolution of early developments and setups from COCOON partners (e.g., VisiBot), which leverages advanced algorithmics such as graph theory, statistical analysis, machine learning (ML) techniques, Natural Language Processing (NLP), and Information Theory.

A significant part of this deliverable is dedicated to the use of the EWS for cybersecurity vulnerability assessment and risk scoring for ICS of EPES. The EWS integrates and processes in real-time Cyber Threat Intelligence (CTI) and Open-Source Intelligence (OSINT) feeds, along with Operational Technology (OT) network scans, to extract meaningful information for operators. This includes state transition examples for the most common communication protocols for ICS, such as Modbus and IEC 104, along with graph-based dependency mapping. This comprehensive approach ensures that operators have a clear and actionable understanding of the cybersecurity posture of their ICS, enabling them to take proactive measures to mitigate risks.

The framework and EWS architecture are designed to be scalable and adaptable, capable of evolving with the changing threat landscape and technological advancements. The practical implementations and proof-of-concept pilots demonstrate the feasibility of the proposed solutions in real-world scenarios. By integrating industry standards and procedures, as well as advanced algorithmics, and real-time threat intelligence, this deliverable intends to offer a clear understanding of the COCOON's holistic approach to enhancing ICS cybersecurity in EPES, and thus ensuring practical reliability, stability, and security.

1 Introduction

1.1 Scope of the Deliverable

Deliverable 1.2 (D1.2) specifies the implementation of cyber threat models, the vulnerability assessment and risk scoring followed in the COCOON project. It also provides an early version of the documentation and software prototypes of the COCOON EWS. The latter is a key component of the COCOON architecture in charge with early detection of cyber vulnerabilities of EPES and their live risk assessment and scoring, along with advisory options for further mitigation actions.

Specifically, this deliverable provides a holistic framework for COCOON threat models fully aligned with the MITRE ATT&CK Framework[1], which is focused on adversary Tactics, Techniques, and Procedures (TTPs) tailored to EPES within the COCOON pilots. By aligning threat modelling with MITRE ATT&CK framework it is ensured that the COCOON approach is grounded in the latest threat intelligence and best industry practices, and thus enhancing the resilience of EPES against sophisticated cyber-attacks.

The MITRE ATT&CK framework is indispensable for threat analysis because it provides a comprehensive, globally recognized model that systematically categorizes TTPs. This enables security professionals to understand and counteract cyber threats more effectively, fostering a proactive and informed defense strategy.

TTPs are the specific methods and strategies used by adversaries to execute cyber-attacks. By focusing on TTPs as part of the COCOON's threat analysis we aim to provide at an early stage of the project implementation a deeper understanding of how attacks are carried out, allowing them to develop more targeted and effective defensive measures. Thus, by working with TTPs, COCOON intends to help EPES stakeholders (e.g., pilot owners) to anticipate on possible mitigation actions for the identified threats, and hence, enhancing their overall cybersecurity posture and resilience against evolving threats. First, this report summarizes several relevant definitions, and the importance of managing vulnerabilities, along with key components, methodologies, and the steps involved in implementing threat models for the ICS of EPES. Subsequently, the report delves into the MITRE ATT&CK Framework exploring its background, the matrix of ICS domain, adversaries TTPs and mitigations, along with the benefits of adopting this framework. To prove its practicality, this report also includes concrete examples, tailored to the COCOON pilots, of potential Attack Vectors (AVs) using a step-by-step approach to outline how they will be used in the context of the pilots. Thus, for each pilot there exists a threat model use-case, the TTP path for a complete AV, and mitigation suggestions for each tactic used. The entire process flow is aligned with globally accepted industry practices entailed within the MITRE ATT&CK Framework.

Furthermore, this report also includes a dedicated section for APTs, explaining their lifecycle and the phases of the Cyber Kill Chain (CKC). The two main types of attacks cover examples, key characteristics and mitigation strategies, that are associated with APTs.

Another significant part of this report details the architecture and logic of the COCOON EWS. The EWS plays a crucial role in vulnerability assessment and risk scoring for EPES by leveraging CTI and OSINT along with real-time network traffic analysis coming from the measurement layer of the pilots, such that to continuously monitor and analyze real-time data for detecting anomalies and potential threats. Thus, the report details on how EWS facilitates timely intervention for ensuring the secure and stable operation of EPES. This report details the evolution of the COCOON's EWS from

[1] https://attack.mitre.org/

the early developments of the BotNet system of UGLA and its upgraded version called BotPro, into a highly modular, flexible, and versatile monitoring and analysis tool for real-time vulnerability assessment and risk scoring of potential cyber threats of EPES. To this end, this report also summarizes some proof-of-concept security assessments to actual network scans for two commonly used communication protocols for ICSs of EPES such as Modbus and IEC 104 which are the prevalent ICS protocols in COCOON pilots and major vessels for a variety of TTPs.

1.2 Relationship with other Work Packages, Tasks and Deliverables

This deliverable is part of the WP1 and reflects the work carried out during Tasks T1.2 (Threat models) and T1.3 (Vulnerability Assessment). On one hand, this deliverable gets as input the system wide requirements (WP2 and WP3), as well as network structure and EPES ICS dataflow logic and asset inventory from the four COCOON pilots (WP5-WP8) for the elaboration of tailored AVs and vulnerability assessment examples. There is also an interdependency between this deliverable (D1.2) and previously released deliverable of WP1, D1.1 (COntrol, Measurement, and Monitoring Layer (COMML) properties), in the sense that COMML provides input to the data collection layer of the EWS detailed in this report. Specifically, D1.1 provided meaningful operational properties for the COMML in intra-domain EPES setups to be used in the context of vulnerability assessment and risk profiling (T1.2, T1.3).

On the other hand, the output of this deliverable will be used as input for Task T1.5 of WP1, and it will also indirectly guide the technology choices to be adopted for anomaly diagnosis solutions to be developed as part of Task T1.4. Figure 1 schematically summarizes the relationship of D1.2 with other deliverables, tasks and WPs.

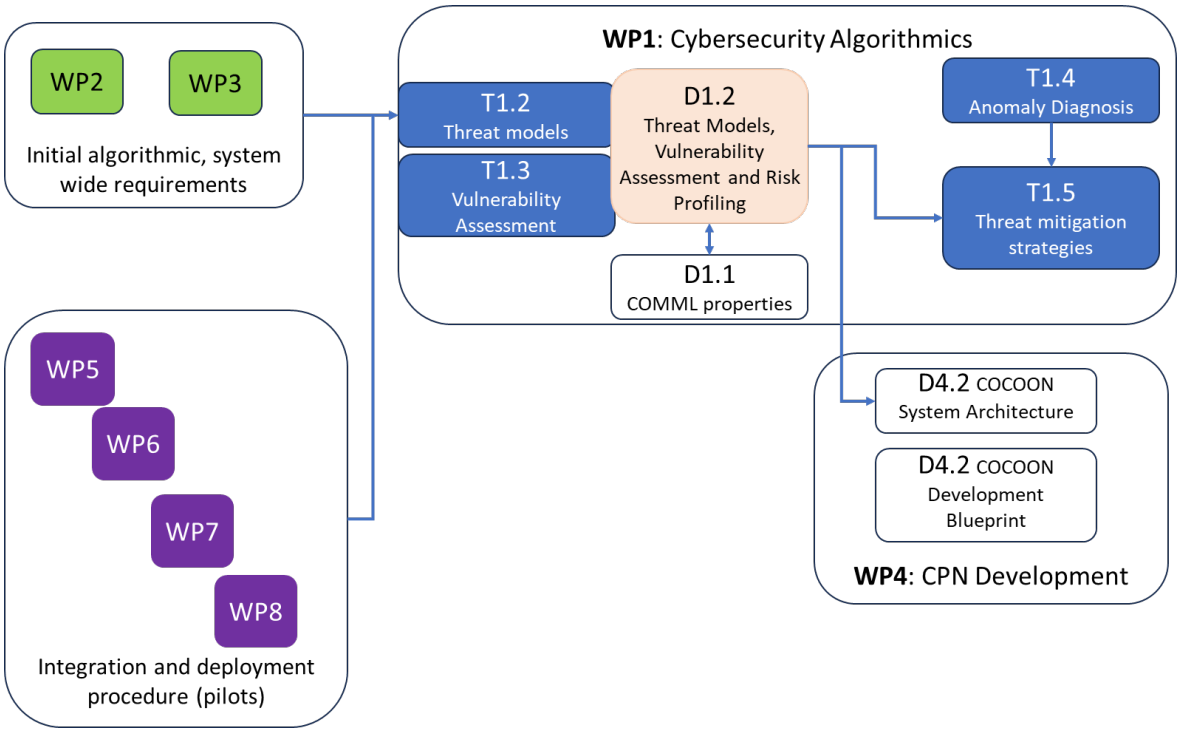

Figure 1: The relationship of D1.2 with other tasks, deliverables and WPs

1.3 Document Structure

Deliverable D1.2 is structured in six chapters. The first chapter introduces the scope of this report, and briefly outlines its position within the whole COCOON project implementation. Chapter 2 describes the methodology adopted for the elaboration of the deliverable which is twofold: (i) a methodology for the construct of the COCOON's threat models aligned with industry practices within

the MITRE ATT&CK Framework, and (ii) a methodology for the COCOON's vulnerability assessment and risk scoring which integrates key features of the COCOON EWS. Chapter 3 delves into the detailed implementation and working examples for the threat modelling approach of COCOON. Chapter 4 provides the architectural vision and implementation details of the EWS, while Chapter 5 exemplifies how the output of the EWS is used for vulnerability assessment and risk scoring with specific examples related to some of the mostly used networking protocols for ICS of EPES, such as Modbus and IEC104.

2 Methodology

In the rapidly evolving landscape of cybersecurity, protecting critical infrastructure such as EPES requires a multi-faceted approach that integrates advanced methodologies and frameworks. Within the COCOON project, a comprehensive methodology for threat modelling fully aligned with the MITRE ATT&CK framework is proposed and it will be briefly described in the current chapter of this deliverable. Further, a robust methodology for vulnerability assessment and risk scoring is proposed which stands on the core functionalities of an EWS to be also detailed in Chapter 4.

The MITRE ATT&CK framework provides a detailed knowledge base of TTPs, enabling the understanding and possible mitigation actions for effectively assessing threats. By aligning the COCOON threat modelling with MITRE ATT&CK, we ensure that our approach is grounded in the latest threat intelligence and best practices.

In addition, this report introduces a sophisticated methodology for vulnerability assessment and risk scoring, leveraging graph-based dependency mapping to map dependencies and assign scoring weights to each Common Vulnerabilities and Exposures (CVE). The proposed approach enhances traditional vulnerability assessment methods by providing a clear understanding of how vulnerabilities can propagate through the system, considering the interdependencies between assets. By integrating data from OSINT search engines (e.g., Shodan[2], Censys[3], etc.) and adhering to established frameworks such as the National Institute of Standards and Technology (NIST)[4], Common Vulnerability Scoring System (CVSS)[5], and Factor Analysis of Information Risk (FAIR) model[6], the COCOON methodology for vulnerability assessment and risk scoring ensures a comprehensive and robust security posture driven by industry standards and best practices. This dual approach of threat modelling and vulnerability assessment aims to equip EPES stakeholders with the necessary tools to proactively identify, prioritize, and mitigate risks, ultimately enhancing their resilience against cyber threats.

2.1 Methodology for the COCOON threat models

The COCOON project's threat modelling methodology offers a structured approach tailored for EPES fully aligned with the MITRE ATT&CK framework and guidelines to ensure comprehensive coverage of potential cyber threats.

The following process diagram identifies the threat model's methodology of COCOON (Figure 2), which includes eight phases, briefly explained below. As illustrated, a feedback mechanism exists since threat models will be documented according to stringent validation and will aid towards the refinement or reconfiguration of components within the Scope Definition phase focusing on the physical assets of a given EPES setup. This methodology can be looped many times until the threat model is improved and reaches the necessary stringent validation.

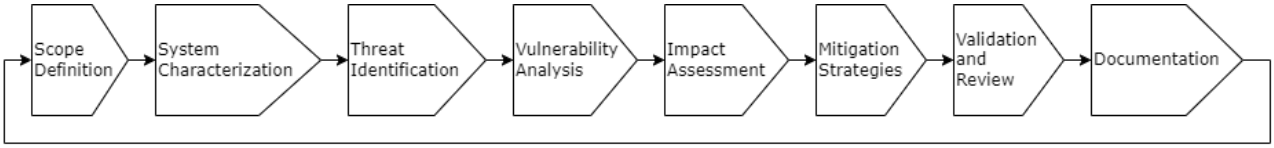

Figure 2: Methodology for COCOON's threat models

[2] https://www.shodan.io/
[3] https://search.censys.io/
[4] https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final
[5] https://nvd.nist.gov/vuln-metrics/cvss
[6] https://www.fairinstitute.org/what-is-fair

1. **Scope Definition**: Within this phase a comprehensive threat modelling process will be conducted which identifies and categorizes the specific systems (Information Technology (IT) and OT) within the EPES to be modelled. This includes the types of ICS, such as Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), Programmable Logic Controllers (PLC), Energy Management Systems (EMS), Substation Automation Systems (SAS), Advanced Metering Infrastructure (AMI), among others.

2. **System Characterization**: This phase refers to the process of understanding the system's components, architecture, data flows, and control flows. Identify the assets (hardware, software, data), their interconnections, and their roles in the system's operation. This includes a detailed understanding of the communication protocols used (e.g. Modbus, DNP3, IEC 60870-5 / IEC 104, IEC 61850 - GOOSE) in the communication network topology scheme.

3. **Threat Identification**: Identify and categorize potential threats using the ICS matrix of the MITRE ATT&CK framework. This matrix provides a list of tactics and techniques commonly used by threat actors against ICS. Threat actors can be internal/external entities that may run malicious actions accidentally even as a victim or as an adversary intentionally. For each asset identified in Step 2, identify relevant tactics and techniques that could be used to compromise it.

4. **Vulnerability Analysis**: Analyse the communication network topology scheme of each industrial partner for vulnerabilities that could be exploited by the identified threats. This includes reviewing the network's design, configuration, and operational procedures for weaknesses. Use tools such as vulnerability assessment scanners and penetration testing to identify system weaknesses.

5. **Impact Assessment**: Evaluate the potential impact of each identified threat on the system's operation and the overall power grid network. This includes assessing the potential for physical damage, operational disruption, data loss, and safety risks matrix. Use a risk scoring system to prioritize the threats based on their potential impact categories.

6. **Mitigation Strategies**: Develop strategies to mitigate each identified threat. This includes implementing preventive security controls, such as firewalls, intrusion detection systems, and access controls. As well as operational procedures, such as incident response plans or disaster recovery plans and system backups to recover from attacks. Align the mitigation strategies with the MITRE ATT&CK guidelines for the ICS matrix to ensure coverage of all identified tactics and techniques.

7. **Validation and Review:** Validate the threat model by testing the effectiveness of the implemented mitigation strategies against simulated attacks. The basic steps are testing and simulation, audit and compliance and continuous monitoring. Review and update the threat model regularly to trace any changes in the system's architecture, operation, and threat landscape.

8. **Documentation:** Document the threat modelling process, which includes the system characterization, threat identification, vulnerability analysis, impact assessment, mitigation strategies and validation. This documentation should be easily accessible by the system operators and cybersecurity personnel for reference and regular updates. Each industrial partner can systematically identify and mitigate cyber threats, thereby protecting their EPES effectively.

The MITRE ATT&CK framework is used to design threat model scenarios based on examples of AVs across all four pilots, to proceed with emulations on their actual communication network topology scheme. The procedure for the design of those is described briefly for each pilot example, the reader is provided with an explanation of the scenario, which is based on the actual communication network topology scheme, the TTPs steps-path of AV, and some mitigation suggestions on how to avoid this AV example for each tactic used.

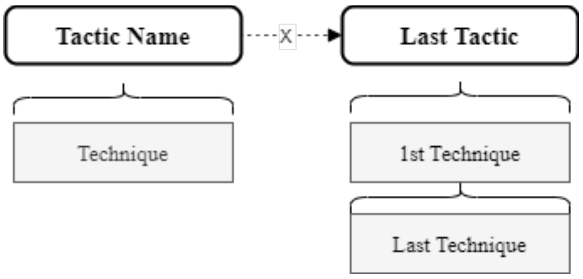

Figure 3: Steps-Pathway of AV Example

Figure 3 is an example and is used specifically to demonstrate the procedure of emulation based on a set of threat model scenarios as used within the COCOON project (Section 3.5). The diagram presents at least one or multiple tactics organized by columns, and techniques at least one for each tactic organized by rows. If there is more than one tactic in the AV path, 'x' represents the step number between the tactics.

2.2 COCOON's methodology for vulnerability assessment and risk scoring

The process flow and components summarizing the COCOON methodology for vulnerability assessment and risk scoring involves ***identifying***, ***quantifying***, and ***prioritizing vulnerabilities*** in systems and networks of EPES. This process is crucial for understanding the security posture of critical infrastructure like EPES. Figure 4 below provides the diagram dataflow of the COCOON approach. This process describes in essence the relationship between ***information gathering***, ***vulnerability identification,*** and ***risk scoring***.

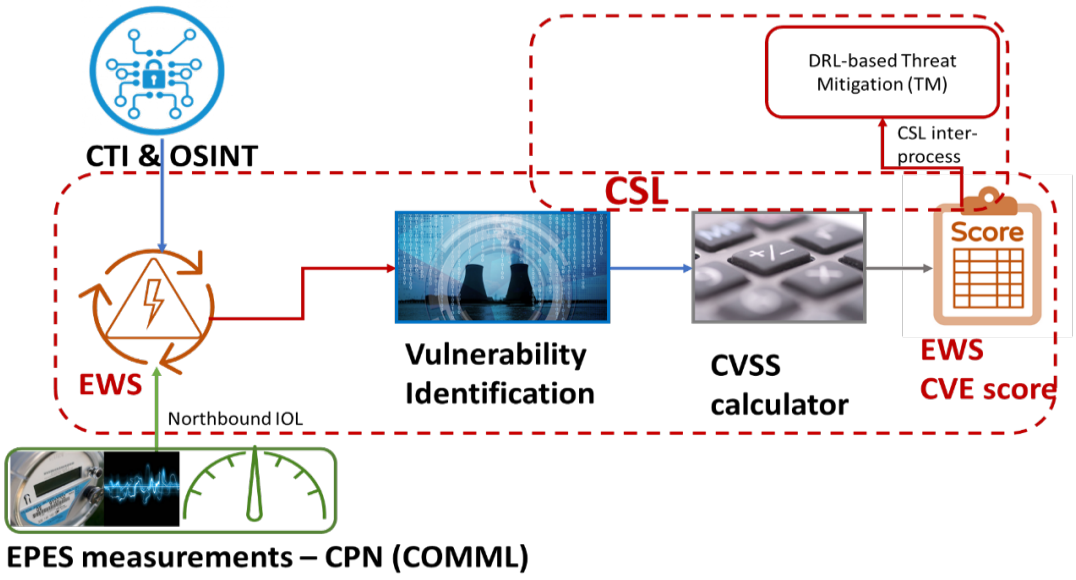

Figure 4: Dataflow for vulnerability assessment and risk scoring as embedded within the COCOON EWS

Specifically, the dataflow for vulnerability assessment and risk scoring adopted by COCOON makes use of trusted search engines for CTI and OSINT which are processed and analyzed by the COCOON EWS, along with offline and live measurements coming from the EPES infrastructure under analysis.

The scans related to EPES measurements are streamed to the EWS via the Northbound interface of the Control Measurements & Monitoring Layer (COMML) within the larger CPN. The output of the EWS is then fed into the ***Vulnerability Identification*** component, which is responsible for systematic and proactive detection of weaknesses, flaws, and gaps in the EPES system that could be exploited by adversaries. Then, after the identification stage the scoring process makes use of the CVSS such that to adhere to a standardized method for scoring vulnerabilities. The CVSS will be further enhanced by considering process dependencies encapsulated in the vulnerability identification stage. The output of the ***CVSS calculator*** system is then fused to assess the final ***EWS CVE risk score.*** It is to be noted that the EWS CVE score will be used as input for the Threat Mitigation module of the Cybersecurity Service Layer (CSL), which is the scope of the developments in Task T1.5 and the following tests in Tasks T4.1, T4.2, T5.4, T6.4 and T8.4, respectively.

Furthermore, it is to be highlighted that the vulnerability assessment and risk scoring methodology described in this report also integrates the following phases and steps:

1. **Data Collection and Inventory Management**

Asset Inventory: maintain an up-to-date inventory of all IT/OT assets, including hardware, software, and network components. In the ongoing implementation of the EWS of COCOON, the CTI search engines such as Shodan and Censys were used to assist in identifying and cataloging the EPES assets of interest.

Configuration Management: track the configurations and settings of all assets to identify potential vulnerabilities and misconfigurations.

2. **Vulnerability Identification**

Automated Scanning: use automated scanning tools to identify vulnerabilities in IT/OT environments. CTI tools such as Shodan and Censys provide extensive scanning capabilities for detecting exposed devices and services, as such they have been chosen as reference engines for this scope in COCOON.

Manual Assessment: in some circumstances (e.g., equipment or software version not yet listed within the reference CTIs databases), manual assessments to identify vulnerabilities that automated tools may miss will be carried out. This includes reviewing system configurations, network architectures, and security policies.

3. **Risk Scoring and Prioritization**

CVSS Scoring: Utilize the CVSS framework to score identified vulnerabilities based on their severity and potential impact.

FAIR Analysis: Apply the FAIR model to quantify the risk associated with each vulnerability, considering the frequency and magnitude of potential loss events.

Prioritization: Prioritize remediation efforts based on risk scores, focusing on vulnerabilities with the highest potential impact and likelihood of exploitation.

4. **Remediation and Mitigation**

Patch Management: implement a robust patch management program to address identified vulnerabilities promptly.

Security Controls: deploy appropriate security controls, such as firewalls, intrusion detection systems, and access controls, to mitigate risks.

Incident Response: Develop and maintain an incident response plan to address security incidents.

5. **Continuous Monitoring and Improvement**

Real-Time Monitoring: implement real-time monitoring solutions to detect and respond to emerging threats and vulnerabilities.

Regular Assessments: Conduct regular vulnerability assessments and risk reviews to ensure the ongoing security of IT/OT environments.

Feedback Loop: Establish a feedback loop to continuously improve the vulnerability assessment and risk scoring methodology based on new data and insights.

3 COCOON Threat Models

3.1 Background

In the context of ICS, threat modelling [1] is a process that involves identifying, evaluating, and communicating information about potential threats, vulnerabilities, and the impact on the system if these vulnerabilities are exploited, that could affect a specific network. Given the critical nature of ICS in sectors like energy, water etc., the development of a robust threat model is crucial. The goal is to anticipate and mitigate risks before affecting the operation of the industrial environment.

The security practice of threat modelling empowers each team to gain a comprehensive understanding of the threat's characteristics and its potential consequences on the network. There are many important key components of threat models for ICS such as asset identification, threat identification, vulnerability analysis, risk assessment, impact analysis and many more. In addition, threat modelling acts as a tool for assessing risks that threats may pose to applications, considering their potential vulnerabilities. Incorporating risk assessment techniques into the threat modelling process enhances threat prioritization and leads to more concrete outcomes.

In the following sections, we detail the COCOON Threat Modelling Framework, including relevant definitions used along this report, key components, and methodologies of threat modelling in ICS.

3.2 COCOON Threat Modelling Framework

The COCOON Threat modelling framework will establish a comprehensive framework outlining the functional components of inside, external and hybrid attack scenarios to be examined within the demonstrator setups. This framework will be developed in close collaboration with industrial partners, incorporating insights from the design and implementation of the four pilot studies. As already mentioned, it will align with the MITRE ATT&CK practical guidelines to customize adversary TTPs.

Adversary TTPs will guide the development of practical approaches for executing APTs, that exploit specific vulnerabilities and network devices (e.g. firewall bypass). This approach will also address the inherent vulnerabilities of ICS which rely on minimally secure industrial protocols (e.g. Modbus, TCP/RTU, IEC61850 GOOSE). The designed APTs will encompass both volumetric (e.g. DDoS) and stealthy attacks (e.g. malware propagation, phishing) and will cover their entire lifecycle, as outlined in the CKC process.

The effectiveness of these attacks will be validated against and aligned with the implementation guidelines and high-level business objectives provided by the four industrial partners leading the COCOON pilots.

3.2.1 Definitions

Below we provide the definitions of the most relevant concepts, which are adopted in this document.

Operational Technology[7] employs hardware and software to monitor and control industrial equipment and systems. OT is essential for managing advanced specialized systems found in various sectors, including energy, industrial manufacturing, oil and gas, robotics, telecommunications, waste management, and water treatment industries.

[7] Information Technology (IT) vs. Operational Technology (OT) Cybersecurity | Fortinet

Industrial Control Systems[8] are one of the most prominent forms of OT. They control and monitor the performance of industrial processes and deploy systems like SCADA.

The **SCADA** system's vulnerabilities often involve basic bugs such as stack and buffer overflows, as well as issues like information disclosure and others. These vulnerabilities allow adversaries to execute arbitrary code (Remote Code Execution - RCE), perform DDoS, or steal information using some tactics.

The most used protocol for the communication of ICS is the **Modbus protocol**[9], which is accepted as the unofficial industry standard for remote monitoring and control within SCADA. The Modbus protocol operates through a client-server or interrogator-responder relationship. The client or interrogator device is a Human Machine Interface (HMI) or a desktop host running a SCADA management application. The server or responder device can be any PLC or Remote Terminal Unit (RTU), including sensors, valves, and other devices.

The **IEC 61850**[10] is a global standard for communication networks and systems in power utility automation, that defines communication protocols (for example MMS and GOOSE). The IEC 61850 standard includes support for the time stamp feature, which is not available in the Modbus protocol.

The **GOOSE**[11] (Generic Object-Oriented Substation Event) protocol, as specified by the IEC 61850 standard, is a communication model that employs swift and dependable methods to bundle various types of data (such as status and values) into a dataset and transmit it over communication networks. It facilitates interoperability and efficient data exchange between Intelligent Electronic Devices (IEDs) within substations, enabling advanced control, monitoring, and protection functions.

Threat Actors[12] An individual or a group that poses a cybersecurity threat. It could be external entities, internal unprivileged users, or internal privileged users with elevated access rights. These threats could manifest either accidentally or intentionally through malicious actions. There are many types, all with various attributes, motivations, skill levels and tactics. Mapping out the different threats and threat actors, and their path of attack by exploiting vulnerabilities and impact on the security of networks, is an essential step. They can evade security controls, exploit vulnerabilities, manipulate or delete sensitive data or any other objective they want to achieve. Some of the most common types of threats are insider, external, and hybrid actors.

Insider Threats[13] are some security attacks within the industrial partner, where internal users intentionally exposing confidential information willingly sabotage their organization. While accidental exposure is possible, malicious insiders share corporate data or vulnerabilities with external parties. Detecting malicious insider threats can be challenging for industrial partners because these individuals are authorized industrial partner users with legitimate access to corporate systems and networks. To mitigate this risk, industrial partners should monitor network activity for unusual behavior or access patterns, such as users accessing files or systems outside of their typical scope. Such anomalies can be indicators of insider threats. Some internal threats examples are insider attacks, accidental data breaches, poor password management, privilege abuse, and careless behavior.

---

[8] Industrial Control System - Definition | Trend Micro (US)
[9] The Modbus Protocol from an Offensive Security Perspective (redbotsecurity.com)
[10] IEC 61850 – Basics and Applications - OMICRON (omicronenergy.com)
[11] What is IEC 61850 GOOSE messaging? - iGrid Smart Guide (igrid-td.com)
[12] What is a Threat Actor? - Types & Examples (sentinelone.com)
[13] External and Internal Threats | WithSecure | WithSecure™

External threats[14] are any potential danger or risk that comes from outside of an industrial partner. These threats can take many forms, including cyber-attacks or efforts by competitors to sabotage an industrial partner's success. Some external threats examples are malware, phishing, DDoS attacks, Zero-day exploits, and supply chain attacks.

Hybrid threats[15] are the types of cyber-attacks where the adversary combines multiple activities with different goals, using some tools to plan and carry out the attack. A combination of a hybrid attack for example is a dictionary and brute-force attack where the adversary gets a list of potentially credential matches.

An **Attack Vector** [2] is a pathway in which an adversary uses the MITRE ATT&CK framework with specific TTPs depending on the attack scenario to break into network vulnerabilities/weaknesses and exploit them. A few common techniques are Man in The Middle (MiTM/AiTM), Commonly Used Port and DDoS which we will use as an example for demonstration.

Figure 5: AV Procedures Example

An **Attack Surface**[16] is the combination of all the possible AVs that an adversary can exploit. The larger the number of AVs an industrial partner has, the bigger the attack surface. By reducing the number of AVs, an industrial partner can effectively shrink its attack surface. Industrial partners must follow best practices and security measures to mitigate AVs in case to reduce the attack surface, preventing adversaries from exploiting the vulnerabilities/weaknesses and reaching their objective.

Visualizing attack paths[17] is a crucial element in the risk assessment process, as it highlights the potential vulnerabilities in a network infrastructure. By simulating the techniques used by threat actors and visualizing their pathways through network systems and services the security teams can effectively identify the possible routes an adversary might take to compromise the network.

The visualization of these pathways helps to identify and address network misconfigurations, and vulnerabilities/weaknesses that could be exploited. Figure 6 provides an example of a visualisation that indicates alternate critical attack paths consisting of different techniques or methods satisfying

[14] What is External threats? | ITOps Glossary (netenrich.com)
[15] Hybrid threats as a concept - Hybrid CoE - The European Centre of Excellence for Countering Hybrid Threats
[16] What is an attack vector? | Cloudflare
[17] Visualizing attack paths

one of the four selected stages of (i) initial access, (ii) exploitation, (iii) lateral movement, and, (iv) command and control.

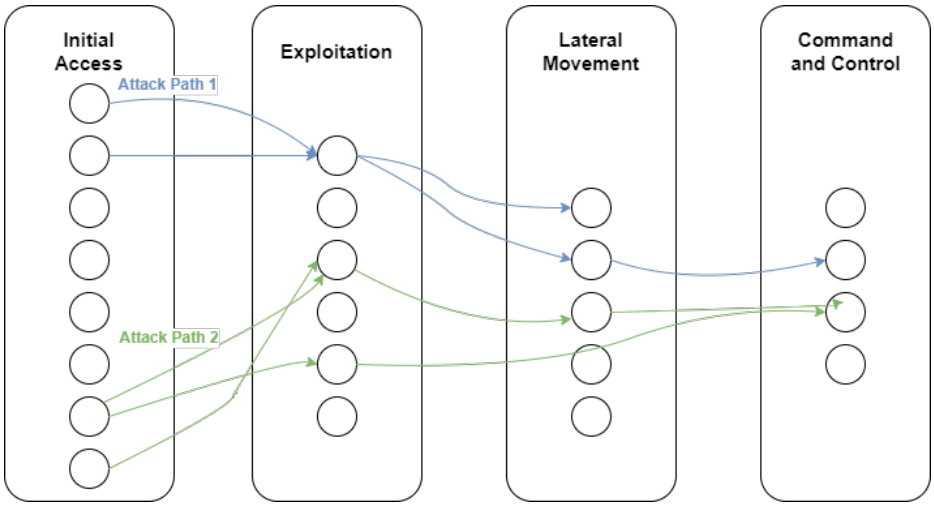

Figure 6: Visualization example of critical attack paths

Vulnerability exploits[16]: A vulnerability is a flaw in software or hardware. For example, think of it as being like a defective device that does not work properly, enabling an adversary who knows where the faulty device is to enter a secured network. When an adversary successfully uses a vulnerability to enter a network, this is called a vulnerability "exploit."

Zero-Day vulnerabilities are unknown vulnerabilities, with no available fix. They can be exploited before even being identified and provide a patch to be fixed.

3.2.2 Importance of Managing Vulnerabilities

1. Update regularly, applying the software or hardware updates can fix most vulnerabilities.
2. Zero-Day awareness, employ security measures like intrusion detection systems to monitor and mitigate potential exploits.
3. Security Best Practices, implement security strategies including regular audits, developers and operators training, and incident response plans to enhance overall protection against exploits.

3.2.3 Key Components of ICS Threat Modelling

1. **Asset Identification**:
   a. Critical Systems: Identify the critical components of the ICS, such as PLCs, SCADA systems, RTUs and HMIs.
   b. Data Flow: Map out the data flow between different components of the ICS.
2. **Threat Identification**:
   a. External Threats: Identify any potential threats outside the industrial partners, such as a cyber-attack from adversaries or competitors.
   b. Internal Threats: Consider threats from within the industrial partner including insider threats or malicious insiders.
   c. Environmental Threats: Include natural disasters, hardware failures and other physical threats.
3. **Vulnerability Analysis**:
   a. Software Vulnerabilities: Identify vulnerabilities in software, including operating system's (OSs) and application software.
   b. Hardware Vulnerabilities: Consider vulnerabilities in ICS hardware components.
   c. Network Vulnerabilities**:** Examine vulnerabilities in the communication networks used within the ICS.
4. **Risk Assessment**:
   a. Likelihood: Estimate the likelihood of different threats exploiting vulnerabilities.

   b. Impact: Assess the potential impact of successful attacks on the ICS operations, safety, and data integrity.
   c. Risk Matrix: Use a risk matrix to categorize and prioritize risks based on their likelihood and impact.
5. **Countermeasure Development**:
   a. Preventive Measures: Implement security controls to prevent threats, such as firewalls, intrusion detection systems, intrusion prevention systems (IPS), and regular software updates.
   b. Detective Measures: Develop capabilities to detect attacks in progress, including monitoring and logging systems.
   c. Responsive Measures: Plan for response and recovery in the event of an attack, including incident response plans and disaster recovery protocols.
6. **Impact Analysis**:
   a. Operational Impact: Distribution of industrial processes, loss of control of devices or production downtime.
   b. Safety Impact: Potential harm to human life or the environment due to malfunctioning equipment.
   c. Financial Impact: Economic losses due to production halts, or equipment damage.
   d. Reputation Impact: Loss of trust from stakeholders, customers, or the public.

3.2.4 Methodologies of ICS Threat Modelling

1. **STRIDE** (Spoofing, Tampering, Repudiation, Information Disclosure, DDoS, and Elevation of Privilege).
2. **Attack Trees** - Root node, branches, and leaves.
3. **MITRE ATT&CK** - Adversaries TTPs, which we will align with this methodology.
4. **Kills Chain Analysis -** Stages of attack.
5. **HAZOP** (Hazard and Operability Study) - Deviation and guide words.

3.2.5 Steps of Implementing ICS Threat Modelling

1. **Define scope and objectives**: Clearly define the scope of the threat modelling exercise and the objectives you aim to achieve.
2. **Assemble a team**: Gather a team with expertise in ICS, cybersecurity, and risk management.
3. **Gather data**: Collect detailed information about the ICS architecture, devices, and communication flows.
4. **Identify threats and vulnerabilities**: Use the chosen methodologies to identify potential threats and vulnerabilities.
5. **Assess Risks**: Perform a risk assessment to determine the likelihood and impact of identified threats.
6. **Develop mitigation strategies**: Create and implement mitigation strategies for identified risks.
7. Validate and test: Validate the threat model through testing and simulations to ensure its effectiveness.
8. **Update regularly**: Regularly update the threat model for new threats, vulnerabilities and changes.

By following these steps and utilizing the appropriate methodologies, the COCOON project via its industrial partners can create effective threat models for the envisaged pilots.

3.3 Alignment with MITRE ATT&CK Guidelines

3.3.1 Background

The MITRE ATT&CK[18] framework was created in 2013 after an experiment to study behavior patterns of adversaries, by a group of researchers who simulated a scenario involving blue-red team roles as part of a research project. An internationally available knowledge base of TTPs is catalogued in real-world observations. The framework reflects on various phases of an adversary attack lifecycle and the platform they are known to target. The cybersecurity industry utilizes the ATT&CK knowledge base, to develop specific threat models and methodologies.

ATT&CK stands for Adversarial Tactics, Techniques, and Common Knowledge[19], emphasizing the focus on real-world observation and experiences rather than hypothetical scenarios.
Adversarial in this context refers to attackers who are also known as adversaries, threat actors and commonly known as hackers.

The tactic or "why" is the highest-level objective adversaries are trying to achieve, they are exploits they use. The techniques or "how" they use those exploits to achieve their objectives and "what" adversaries seek to gain with their actions. Each technique encompasses a range of procedures. Consequently, an adversary's objective is achieved through a sequence of tactics, each employing one or more techniques. This progression continues with subsequent tactics and their associated techniques until the objective is reached. This layering of general tactics down to specific procedures is where we get TTPs.

Finally, the CK stands for Common Knowledge since this is a grouping of data information and reports that MITRE collects which are open to the public and accessible to both adversaries and defenders. Users and researchers submit the information and then they are catalogued.

3.3.2 MITRE ATT&CK Domains

The MITRE ATT&CK framework covers three distinct technology domains:

1. **Enterprise**: For enterprise IT environments.
2. **Mobile**: For mobile devices and environments.
3. **ICS**: Industrial environments.

Each domain features unique TTPs, so there might be some slight overlaps. These domains are distinct, resulting in differences in adversarial behavior. The attack surfaces and the adversary's objectives also differ based on the chosen domain, requiring tailored TTPs.

The MITRE ATT&CK framework can be used by different types of users based on the scope-objective. Commonly the groups are:

- **Blue team** are those on the defense, like security analysts. They would identify different data sources like assets and capabilities both logical and physical including things like OSs servers and types of protocols on the network. Designing and executing adversary emulation exercises to test threat models. They use this framework to determine how good or bad the defenses are and change things to strengthen network protection.
- **Red team** consists of those on the offensive, such as penetration testers and those who actually hack the network and test security by exploiting known vulnerabilities.

[18] MITRE ATT&CK®
[19] https://www.trellix.com/security-awareness/cybersecurity/what-is-mitre-attack-framework/

3.3.3 MITRE ATT&CK Purposes

1. **Threat Intelligence**: Map and understand the behavior of adversaries, enabling better threat intelligence.
2. **Detection and Response**: Leverage the framework to enhance their detection and response strategies by identifying gaps and vulnerabilities/weaknesses in their current defenses.
3. **Red Teaming**: Provides a foundation for developing realistic adversary emulation scenarios, allowing red teams to simulate attacks based on known TTPs.
4. **Security Assessments**: Assess the effectiveness of security controls and guide the development of new defensive measures.

3.3.4 MITRE ATT&CK Matrix ICS

There are 12 phases of an adversary attack lifecycle which are also known as Tactics that the MITRE ATT&CK matrix[20] framework has, instead of 7 phases of the Lockheed Martin CKC, and the 12 phases of ICS CKC. All frameworks offer different models of threat behaviors and scopes-objectives.

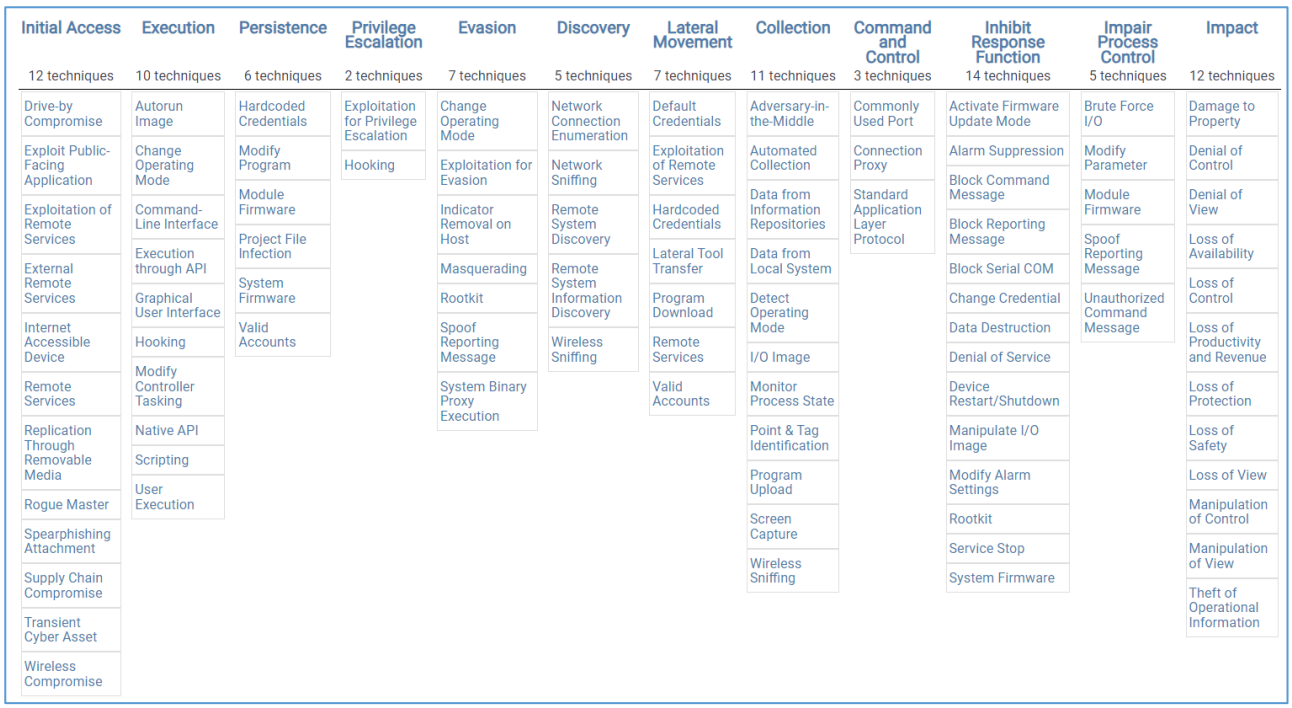

Figure 7: MITRE Matrix ICS

MITRE ATT&CK matrix layout for ICS domain: Tactics are organized by columns and techniques by rows on the above Figure 7. Refers to a domain comprising 94 techniques without sub-techniques.

3.3.5 MITRE ATT&CK Tactics ICS

The MITRE ATT&CK Tactics for ICS[21] adopted by COCOON are as follows:

1. **Initial Access**: Gaining a foothold in the target environment.
2. **Execution**: Running malicious code on the target.
3. **Persistence**: Maintaining access to the target environment.
4. **Privilege Escalation**: Gaining higher-level permissions.
5. **Evasion**: Avoiding detection and removal.
6. **Discovery**: Understanding the target environment.

[20] ATT&CK® Navigator (mitre-attack.github.io)
[21] Tactics - ICS | MITRE ATT&CK®

7. **Lateral Movement**: Moving through the target environment.
8. **Collection**: Gathering data of interest within the environment.
9. **Command and Control (C&C)**: Communicating with compromised systems.
10. **Inhibit Response Function**: Prevent your safety, protection, quality assurance, and operator intervention functions from responding to a failure, hazard, or unsafe state.
11. **Impair Process Control**: Manipulate, disable, or damage physical control processes.
12. **Impact**: Disrupting or destroying systems and data.

3.3.6 MITRE ATT&CK Mitigations

The framework provides security measures and technology solutions designed to prevent the successful execution of specific techniques or sub-techniques. Some mitigations[22] are:

1. **Secure Sockets Layer-Transport Layer (SSL-TLS) Security Inspection**: Break and inspect SSL-TLS sessions to look at encrypted traffic for adversary activity.
2. **Out-of-Band Communications Channel**: Have alternative methods to support communication requirements during communication failures and data integrity attacks.
3. **Exploit Protection**: Use capabilities to detect and block conditions that may lead to or be indicative of a software exploit occurring.
4. **Encrypt Network Traffic**: Utilize strong cryptographic techniques and protocols to prevent eavesdropping on network communications.
5. **Vulnerability Scanning:** Used to find potentially exploitable software vulnerabilities to remediate them.

3.3.7 MITRE ATT&CK Benefits

There are also several benefits[23] of MITRE ATT&CK which can contribute to the COCOON integrated threat analysis framework, and they are listed below:

1. **Common Language**: Provides a common framework and language for discussing and addressing cyber threats across different sectors and disciplines.
2. **Real-World Relevance:** Based on actual real-world observations of adversary behavior, ensuring relevance and applicability.
3. **Comprehensive Coverage:** Covers various tactics and techniques, allowing for thorough threat modelling and defense planning.
4. **Community Contribution**: Continuously updated with contributions from the global cybersecurity community, ensuring it remains current and comprehensive.

By leveraging the MITRE ATT&CK knowledge base, industrial partners can develop more robust and effective cybersecurity strategies, threat models and methodologies. Enhancing their ability to detect, respond to, and mitigate cyber threats across various stages of the attack lifecycle and target platforms against TTPs.

3.4 COCOON's APT Lifecycle and Attack Types

3.4.1 Advanced Persistent Threats

An APT[24] is a sustained and focused cyber-attack where an adversary infiltrates a network and remains stealthy for a significant duration. The primary objective of an APT is to extract highly sensitive information, rather than to inflict immediate damage on the target's network. The difference

[22] Mitigations - ICS | MITRE ATT&CK®
[23] ATT&CK Fundamentals Training | MITRE ATT&CK®
[24] What is an advanced persistent threat (APT)? | Definition from TechTarget, available online: https://www.techtarget.com/searchsecurity/definition/advanced-persistent-threat-APT

from regular cyber-attacks such as ransomware is that APTs aim for long-term access rather than a quick intrusion and exit.

APTs are typically executed manually with careful preparation, requiring substantial effort and resources. Adversaries are choosing high-value targets to steal valuable information over an extended period. As a result, APTs are often carried out by well-funded nation-state groups rather than individual adversaries.

Implementation of APTs refers to sophisticated and sustained cyber-attack campaigns where adversaries gain unauthorized access to a network and remain undetected for an extended period. These adversaries typically target specific networks to steal sensitive data, conduct espionage, or cause disruption. The key characteristics of APTs include advanced attack methods, persistence over long durations, and a high level of stealth to avoid detection.

By understanding the nature of APTs and implementing robust mitigation strategies, industrial partners can better protect themselves against these sophisticated and persistent cyber threats.

The lifecycle of APTs can be mapped to the MITRE ATT&CK framework, which provides a comprehensive matrix of TTPs used by adversaries. By understanding these stages and the associated TTPs, industrial partners can better detect, prevent, and respond to APTs.

There are two main types of adversary AVs, stealthy-passive attacks, and volumetric-active attacks.

3.4.2 Stealthy – Passive AV[25]

A **stealthy - passive AV** involves an adversary observing a network to identify open ports or vulnerabilities, aiming to collect information and understand the communication network topology scheme and potential entry points. These types of attacks are designed to evade detection and operate undetected within a target network. Can be difficult to detect because they do not involve altering data or system resources, they just behave stealthily without any interaction. These attacks often involve sophisticated techniques and tools that allow adversaries to remain hidden while they propagate malware, conduct phishing campaigns, or other malicious activities. The primary scope is to achieve their objectives without being noticed by the users or operators.

Stealthy-Passive AV Examples

1. **Malware Propagation**: The spread of malicious software through a network while avoiding detection.
   - **Fileless**: Malware that resides in the memory rather than on the disk, making it harder to detect with traditional antivirus software.
   - **Rootkits:** Tools that hide the presence of malware by modifying the OSs core functions.
   - **Polymorphic:** Malware that changes its code to avoid detection by signature-based antivirus solutions.

2. **Phishing:** Deceptive attempts to obtain sensitive information such as usernames, passwords, and details by masquerading as a trustworthy entity.
   - **Spear:** Specific type of targeted phishing attacks directed at individuals or industrial partners, often personalized to increase credibility.
   - **Clone:** Using a legitimate email that has been previously sent to the victim, modifying it slightly with malicious content, and sending it again to the same recipient.

[25] What is an Attack Vector? Types & How to Avoid Them (fortinet.com)

– **Business Email Compromise (BEC):** Adversaries impersonate executives or trusted industrial partners to deceive them into transferring money or sensitive information.

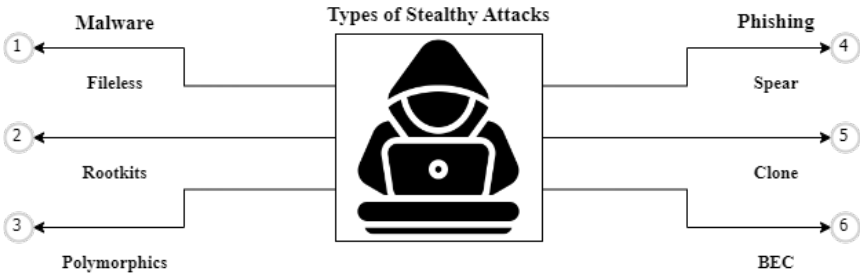

Figure 8: Types of Stealthy Attacks

Stealthy-Passive AV Key Characteristics

1. **Low Visibility:** Stealthy attacks aim to blend in with normal network traffic or system operations to avoid raising suspicion.
2. **Sophistication:** They often employ advanced techniques and technologies to bypass security measures.
3. **Persistence:** Adversaries maintain a presence within the network over extended periods without being detected.

By understanding and addressing the tactics and techniques associated with stealthy-passive attacks, and implementing robust mitigation strategies, industrial partners can better protect the confidentiality of their data and reduce the risk of undetected intrusions by adversaries.

Stealthy-Passive AV Mitigation Strategies

1. Use strong encryption protocols for data in transit to prevent unauthorized access.
2. Network segmentation, to limit the exposure of sensitive information.
3. Security audits and monitoring network traffic, that analyses behavior for unusual patterns that may indicate reconnaissance activities and any anomalies.
4. Implement strict access controls, to ensure that sensitive information is only accessible to specific authorized users.
5. Train employees on the importance of security best practices and the risks of passive reconnaissance.
6. Implement comprehensive endpoint protection solutions, that include anti-malware, anti-phishing, and advanced threat detection capabilities.
7. Require Multi-Factor Authentication (MFA), to add an extra layer of security for accessing sensitive systems and data.

3.4.3 Volumetric – Active AV[26]

A volumetric-active AV sets out to disrupt or cause damage to a network or affect regular operations. This includes adversaries launching attacks against network vulnerabilities. The volumetric-active attack characteristics such as a disruptive nature to disrupt regular operations, causing damage or stealing sensitive data and information. These attacks focus on consuming all available bandwidth or resources, causing service disruptions and downtime. The detection where are more often noticeable than stealthy-passive attacks due to their disruptive effects and direct engagements. The last characteristic is that are intent, as the main objective is to exploit vulnerabilities, gain unauthorized access etc.

[26] What Is a Volumetric Attack? | How Volumetric DDoS Attacks Work | Akamai

Volumetric-Active AV Examples

A type of volumetric-active attack example is **DDoS** where the adversary overwhelms the network with excessive traffic to make it unavailable. A **Malware** attack that uses Viruses-malicious code, Trojan-malicious activities or even ransomware where the victim's system is held hostage until they agree to pay a ransom to the adversary. Another example is a **False Data Injection** (FDI) attack which attempts to disrupt the system within a short time interval for momentary gains, while a covert attack allows an adversary to feed FDI into a system such that the attack effects usually happen in the long term.

An example of a volumetric-active attack known as the User Datagram Protocol (UDP) Flood. Adversaries send a huge number of requests UDP packets to random ports on the target, causing it to check, listening ports and reply with Internet Control Message Protocol (ICMP) responses. That overwhelms the target bandwidth and resources, leading to service disruption.

Volumetric-Active AV Key Characteristics

1. **High Traffic Volume:** Adversaries send an excessive amount of data or requests to the target to exhaust its resources.
2. **Bandwidth Consumption:** The primary objective is to consume the available bandwidth, preventing normal traffic from reaching the target.
3. **Distributed Nature:** Launch from multiple compromised devices-systems (botnets) making them harder to mitigate.

Volumetric-Active AV Mitigation Strategies

1. Deploy EWS to detect and prevent malicious activities in real-time.
2. Regular patching to keep all devices of the network updated with the latest updates or security patches.
3. Use a strong authentication mechanism to implement an MFA.
4. User awareness and regular training sessions to educate everyone about any new and common attack methods that adversaries are using.
5. Access control and least privilege principle - Different permissions for users and operators based on their group role will have necessary access rights.
6. Use network firewalls and IPS to filter out malicious traffic and block IP addresses associated with the attack.
7. Employ specialized DDoS mitigation services that can absorb and filter large volumes of attack traffic.
8. Utilize anycast routing to distribute incoming traffic across multiple data centers, making it harder for adversaries to overwhelm a single target.
9. Network segmentation to contain potential breaches and limit the spread of the attack.

By understanding the nature of and mitigating volumetric-active AV, industrial partners can enhance cybersecurity defenses, ensuring networks and data remain secure against potential disruptions and unauthorized access.

3.4.4 Cyber Kill Chain[27]

The CKC is a framework developed by Lockheed Martin to describe the stages of a cyber-attack, particularly focusing on APTs. It outlines the sequence of actions adversaries take to achieve their objectives. Understanding this lifecycle helps industrial partners identify and disrupt these stages, enhancing their cybersecurity defenses.

[27] What is The Cyber Kill Chain and How to Use it Effectively (varonis.com)

Phases of the CKC

1. **Reconnaissance**: Gathers information about the target to identify potential vulnerabilities and entry points.
2. **Weaponization**: Create a malicious payload by combining an exploit with a backdoor or other malware.
3. **Delivery**: Delivers the payload to the target.
4. **Exploitation**: Delivered payload exploits a vulnerability to execute code on the target.
5. **Installation**: Adversaries malware or other AV will be installed on the system.
6. **C&C**: Establish a C&C to remotely manage system.
7. **Actions on objectives**: Achieve the final objective, such as data exfiltration, espionage, or disruption.

Industrial Control System CKC

ICS CKC[28] the framework proposed by ICS-SANS, which outlines the stages of a cyber-attack specifically targeting ICSs, such as those used in critical infrastructure like power grids or water treatment facilities. It extends the CKC from IT to OT which includes several phases: reconnaissance, initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, C&C, and impact. This model helps industrials understand and defend against potential threats by identifying vulnerabilities and implementing protective measures at each stage of an attack.

3.5 COCOON Exemplar APT Design for Specific Vulnerabilities

3.5.1 Example of AV for Phishing – Spoofing GOOSE and SCADA

TTPs to GOOSE Spoofing AV

GOOSE Spoofing attack can be performed by reaching the primary substation, an adversary can pretend to be a publisher IED and broadcast a multicast GOOSE message throughout the substation network. Injecting spoofed traffic into the substation can cause the opening of circuit breakers or the tripping of protection relays.

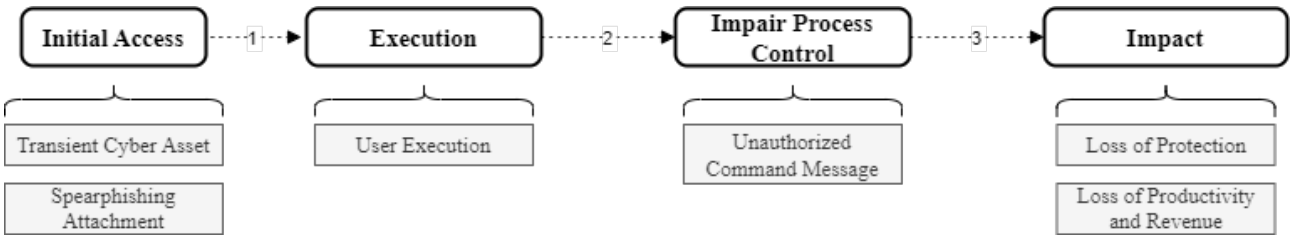

Figure 9: Steps-Pathway of GOOSE Spoofing AV

• **Step 1**: Initial access to the substation network.

Initial Access[29] An adversary must reach the substation network to perform a GOOSE spoofing attack. An adversary trying to gain a foothold within an ICS environment falls under the MITRE ATT&CK Framework's Initial Access tactic.

The technique considered in this scenario is *Transient Cyber Asset*[30]. Transient assets are commonly needed to support management functions and may be more common in systems where a remotely

[28] The Industrial Control System Cyber Kill Chain (icscsi.org)
[29] Initial Access, Tactic TA0108 - ICS | MITRE ATT&CK®
[30] Transient Cyber Asset, Technique T0864 - ICS | MITRE ATT&CK®

managed asset is not feasible, external connections for remote access do not exist, or 3rd party contractor/vendor access is required.

Transient assets may be infected by malware and when connected to an ICS environment the malware performs actions on the target or propagates onto other systems. In this case, an infected engineering laptop could connect to substation equipment for maintenance.

This also means that the adversary, in order to compromise transient assets, might also use techniques such as *Spearphishing Attachment*[31]. A malicious file is attached to a spearphishing email and usually relies upon user execution to gain execution and access.

- **Step 2**: User interaction with email.

Execution[32] the tactic will be used in the case of spearphishing email, the adversary needs the targeted user to interact with the malicious attachment, a technique *User Execution*[33]. The execution usually consists of opening the email attachment.

The users need to be careful in this kind of email with attachments, even if the source is trusted always need to cross-check with the sender, using the official site communication method. Useful mitigation is *User Training* to counteract this technique, which consists of training users to be aware of common phishing and spearphishing techniques.

- **Step 3**: Control and distract normal behavior of ICS processes.

Impair Process Control[34] & **Impact**[35] tactics will be used, once the adversary reaches the substation network, they can inject spoofed GOOSE messages. In this case, the adversary is using the technique of *Unauthorized Command Message*[36]: they are sending unauthorized command messages to instruct control system assets to perform actions outside of their intended functionality, or without the logical preconditions to trigger their expected function.

Opening circuit breakers or tripping protection relays can cause a *Loss of Protection*[37], which can in turn cause a *Loss of Productivity and Revenue*[38].

TTPs to Compromise SCADA Systems from Enterprise IT AV

Based on the below communication network topology scheme example (Figure 10), to perform a scenario where the adversary penetrates the network from the Enterprise IT layer with the final objective of compromising SCADA systems.

[31] Spearphishing Attachment, Technique T0865 - ICS | MITRE ATT&CK®
[32] Execution, Tactic TA0104 - ICS | MITRE ATT&CK®
[33] User Execution, Technique T0863 - ICS | MITRE ATT&CK®
[34] Impair Process Control, Tactic TA0106 - ICS | MITRE ATT&CK®
[35] Impact, Tactic TA0105 - ICS | MITRE ATT&CK®
[36] Unauthorized Command Message, Technique T0855 - ICS | MITRE ATT&CK®
[37] Loss of Protection, Technique T0837 - ICS | MITRE ATT&CK®
[38] Loss of Productivity and Revenue, Technique T0828 - ICS | MITRE ATT&CK®

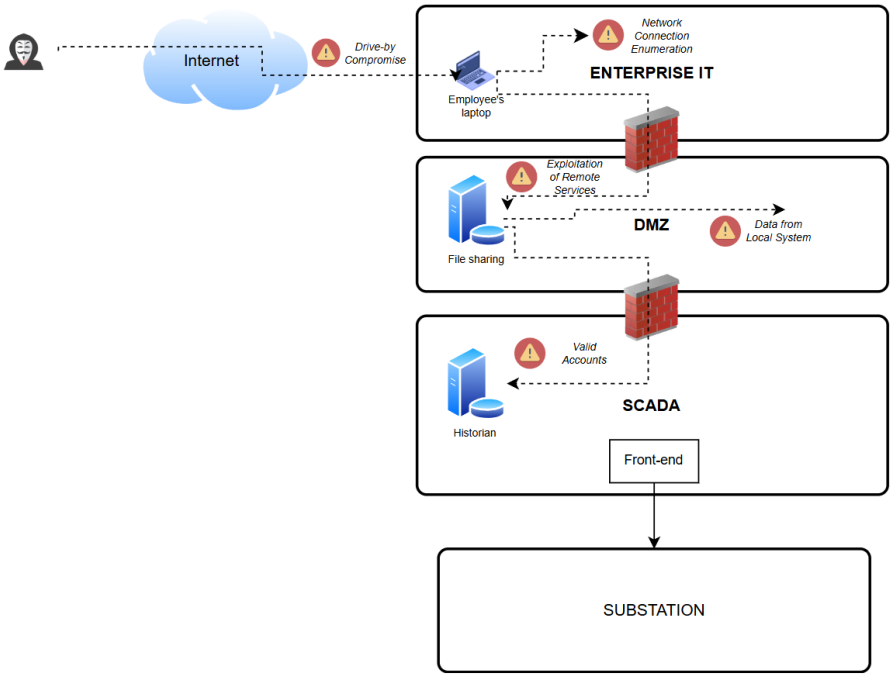

Figure 10: Attacker moving through the network topology scheme

For the above scenario Figure 10 we will use the TTPs of MITRE ATT&CK Framework of the potential attack path to be the one described below Figure 11.

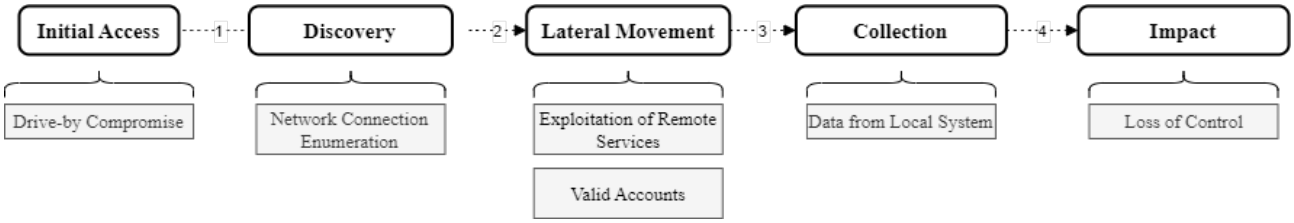

Figure 11: Steps-Pathway of AV to compromise SCADA Systems from Enterprise IT

- **Step 1**: The attacker would target, and compromise websites that people in the industry often visit. With the **Initial Access** tactic - *Drive-by Compromise*[39] technique, the user's web browser is targeted and exploited simply by visiting the compromised website. This kind of targeted attack relies on a specific interest and is known as a watering hole attack[40].

This can be avoided using a mitigation of *Exploit Protection*, to prevent activities that may be exploited through malicious websites.

- **Step 2**: Usual enumeration would follow **Discovery**[41] tactic - *Network Connection Enumeration*[42] technique by using common tools (e.g. netstat, ipconfig).

The mitigation is *Limited or Not Effective*, since it is based on the abuse of system features, so it cannot be easily mitigated.

[39] Drive-by Compromise, Technique T0817 - ICS | MITRE ATT&CK®
[40] What Is a Watering Hole Attack? | Fortinet
[41] Discovery, Tactic TA0102 - ICS | MITRE ATT&CK®
[42] Network Connection Enumeration, Technique T0840 - ICS | MITRE ATT&CK®

- **Step 3**: The adversary can exploit network services to move further in the network using the **Lateral Movement** tactic - *Exploitation of Remote Services*[43] technique for instance using **CVE-2017-7494**[44] a remote code execution vulnerability for the Samba file server.

There is a mitigation of *Vulnerability Scanning* that recognises any new or potentially vulnerable services during regular scans.

- **Step 4**: Further data collection would follow as the adversary has now reached a different segment in the network (**Collection**[45] tactic - *Data from Local System*[46] technique) and usual enumeration (*Network Connection Enumeration* technique). The adversary finally compromises a system in the SCADA network, the Historian, by re-using valid credentials looted during the previous data collection step.

In case to protect sensitive data such as credentials some mitigations that can be used are *Data Loss Prevention* or *Encrypt Sensitive Information* which are recommended by MITRE ATT&CK.

- **Step 5**: Once logged into the SCADA network, the adversary can deploy different techniques under the **Impair Process Control** or **Impact** tactics, such as *Loss of Control*[47] technique.

A mitigation strategy is *Data Backup*, always keeping backups regularly from end users' systems and critical servers. Ensure that backups are kept separate from the internal/organisational network. Can perform a fast recovery and response from adversarial action that impacts control, view, or availability.

3.5.2 Example of AV on firewall bypass and adversary SCADA control

An AV example is to attempt multiple TTPs on the below communication network topology scheme example, to check for SSL-TLS vulnerabilities and exploit them known as adversary emulation. The scope is to bypass the Firewall, and then get access to the control server (SCADA) which uses the Modbus protocol to communicate with all main devices of the network in case to have control.

It's essential to evaluate SSL-TLS[48] protocols for potential vulnerabilities, as these are responsible for the encryption of your network connections. Some of the most frequent issues related to SSL-TLS are self-signed certificates, the expiration of certificates, relying on outdated OpenSSL versions, keeping default settings without any customization, setting up incorrect trust chains, and misconfiguring the used protocol. Ensuring protocols are properly configured and performing regular vulnerability scans will keep you on top of your SSL-TLS setups, helping to prevent common attacks.

[43] Exploitation of Remote Services, Technique T0866 - ICS | MITRE ATT&CK®
[44] NVD - CVE-2017-7494 (nist.gov)
[45] Collection, Tactic TA0100 - ICS | MITRE ATT&CK®
[46] Data from Local System, Technique T0893 - ICS | MITRE ATT&CK®
[47] Loss of Control, Technique T0827 - ICS | MITRE ATT&CK®
[48] What are SSL and TLS Vulnerabilities | Veracode

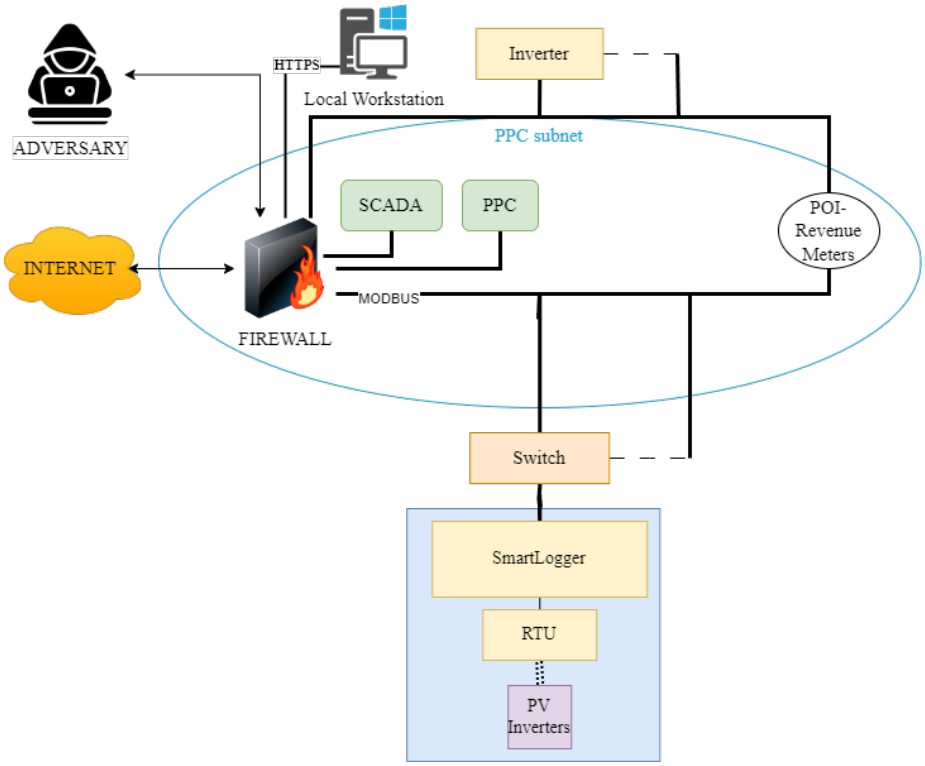

Figure 12: General Topology Scheme with Firewall

TTPs to Bypass Firewalls and Control SCADA AV

Based on the above communication network topology scheme example (Figure 12) An out-of-bound known vulnerability in the SSL - Virtual Private Network (VPN) of the Firewall will be used **CVE-2024-21762**[49]. The vulnerability could enable a remote, unauthenticated adversary to execute arbitrary code or commands by sending specially crafted requests.

Figure 13 illustrates the TTPs of MITRE ATT&CK framework that will be used for the scenario.

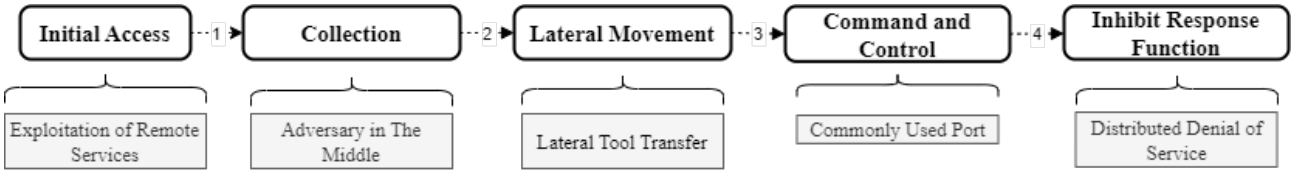

Figure 13: Steps-Pathway of AV Firewall

• **Step 1**: Get access to the local network through the Local Workstation / Firewall Switch.
Initial Access tactic consists of techniques that adversaries may use as entry vectors to gain an initial foothold within an ICS environment.

– Using the *Exploitation of Remote Services* technique, adversaries might exploit a software vulnerability to leverage a programming error in an application, service, or even within the OS software or the kernel itself allowing them to abuse remote services.

Procedure Example - Bad Rabbit[50] is a self-propagating ransomware that initially infected IT networks.

[49] NVD - CVE-2024-21762 (nist.gov)
[50] Bad Rabbit, Software S0606 | MITRE ATT&CK®

The industrial partner can use *Network Segmentation* mitigation to split the network into segments protecting it by reducing access to critical systems and communication services.

However, by exploiting a vulnerability (EternalBlue-MS17-010) in the SMBv1 network file-sharing protocol on Microsoft computers, it was able to spread to industrial networks as well.

• **Step 2**: Collect data from the private network (devices-controllers).
Collection This tactic involves techniques adversaries use to collect domain knowledge and obtain contextual feedback within an ICS environment.

– Adversaries with privileged network access may seek to modify network traffic in real-time using the technique of *Adversary in The Middle (AiTM)* [51]. The AiTM attack technique allows the adversary to intercept traffic to and/or from a particular device on the network. The objective of the adversary is to have the ability to block, log, modify, or insert traffic into the communication network.

Procedure Example - VPNFilter[52] is a multi-stage, modular platform equipped with versatile capabilities to support both intelligence collection and destructive cyber-attack operations. VPNFilter modules such as its Packet Sniffer (PS) can collect traffic that passes through an infected device, allowing the adversary to get the credentials and monitor the Modbus of SCADA.

Alternatively, we can use **Wireshark**[53] software which is used for the first part of launching the AiTM attack. Wireshark will help us to monitor the traffic inside the network.

Providing *Communication Authenticity* mitigation ensures that any messages tampered with through AiTM can be detected and can be used to prevent many procedures or malicious actions.

• **Step 3**: Move within the network to the SCADA system.
Lateral Movement[54] tactic provides the ability for the adversary to try to move through the ICS environment. This tactic includes techniques adversaries use to access and control remote systems on a network. The adversary using techniques of this tactic can move to their next target within the environment, positioning themselves where they want or need to be.

– Using the *Lateral Tool Transfer*[55] technique, adversaries can transfer tools or other files between systems. Copying of files may also occur laterally between internal victim systems to facilitate Lateral Movement with remote Execution, using inherent file-sharing protocols like file sharing over SMB to connected network shares.

Procedure Example - WannaCry[56] can move laterally in industrial networks through the SMB service. WannaCry is ransomware that contains worm-like features, to spread itself across a computer network using the SMBv1 exploit EternalBlue.

Network Intrusion Prevention mitigation is crucial to use network signatures, in recognizing traffic of adversary malware or unusual data transfer over tools and protocols at the network layer.

• **Step 4**: Get control of the SCADA system and exploit it.

---

[51] Adversary-in-the-Middle, Technique T0830 - ICS | MITRE ATT&CK®
[52] VPNFilter, Software S1010 | MITRE ATT&CK®
[53] Wireshark · About
[54] Lateral Movement, Tactic TA0109 - ICS | MITRE ATT&CK®
[55] Lateral Tool Transfer, Technique T0867 - ICS | MITRE ATT&CK®
[56] WannaCry, Software S0366 | MITRE ATT&CK®

C&C[57] tactic consists of techniques that adversaries use to communicate with and send commands to systems, devices, controllers, and platforms with specialized applications used in ICS environments.

Adversaries often utilize commonly available resources and imitate expected network traffic to avoid detection and suspicion. Maybe established to varying degrees of stealth, usually depending on the victim's network structure and defenses.

– Using the *Commonly Used Port*[58] attack technique, adversaries may communicate over a commonly used port to bypass firewalls or evade network detection systems and to blend in with normal network activity, to avoid more detailed inspection. They might use the protocol associated with the port or an entirely different protocol. Often, they utilize commonly open ports, such as those listed below.
  - TCP:80 - HTTP
  - TCP:443 - HTTPS
  - TCP:502 - MODBUS
  - TCP: 20000 - DNP3
  - TCP:44818 - Ethernet/IP

A good practice to avoid this kind of attack is to always close ports that are NOT in use. The mitigation *Disable or Remove Feature or Program* makes sure that unnecessary ports and services are closed to prevent any risk.

Procedure Example - 2015 Ukraine Electric Power Attack[59], Sandworm Team used port 443 to communicate with their servers. They used the BlackEnergy[60] malware toolkit and the KillDisk[61] disk-wiping tool to target and disrupt transmission and distribution substations within the Ukrainian power grid. Similarly, in our experiment, we will try the 502 port that the Modbus protocol is using.

Alternatively, **Ettercap** software can be used which is a security tool for implementing AiTM in Local Area Network (LAN) using Address Resolution Protocol (ARP) spoofing/poisoning techniques.

• **Step 5**: Impact operations.
Inhibit Response Function[62] This tactic involves techniques adversaries use to prevent your safety, protection, quality assurance, and operator intervention functions from responding to a failure, hazard, or unsafe state. Adversaries may modify or update system logic, or even outright prevent responses with a DDoS. They may result in the prevention, destruction, manipulation, or modification of programs, logic, devices, and communications.

– They may execute *DDoS*[63] attack technique to disrupt normal device operations. For instance, an adversary overloads a server with internet traffic using multiple machines also known as a botnet.

This could include overwhelming the device with a high volume of requests in a short time period and sending requests to the device that cannot process. That will cause a crash on the device that

[57] Command and Control, Tactic TA0101 - ICS | MITRE ATT&CK®
[58] Commonly Used Port, Technique T0885 - ICS | MITRE ATT&CK®
[59] 2015 Ukraine Electric Power Attack, Campaign C0028 | MITRE ATT&CK®
[60] BlackEnergy, Software S0089 | MITRE ATT&CK®
[61] KillDisk, Software S0607 | MITRE ATT&CK®
[62] Inhibit Response Function, Tactic TA0107 - ICS | MITRE ATT&CK®
[63] Denial of Service, Technique T0814 - ICS | MITRE ATT&CK®

receives the request, and the network will face critical issues. Such actions can disrupt the device state, causing temporary unresponsiveness, which may require a reboot to resolve.

This can be mitigated through firewalls to prevent malicious traffic and defense tools, for example risk assessments. Another mitigation is *Watchdog Timers* that restart the process and system when a timeout occurs or is unresponsive.

Procedure Example - Industroyer[64] malware framework is designed to disrupt the operational processes of ICS, particularly targeting components used in electrical substations. It represents the first publicly known malware explicitly designed to target and impact operations within the electric grid.

3.5.3 Example of AV for DNS Spoofing and LOLbins-Fileless

Based on the below communication network topology scheme example (Figure 14), we will create some examples of AVs scenarios following the TTPs of MITRE ATT&CK framework to perform emulations of AVs acting as Lenovo over DNS and acting as developers using Living off the Land Binaries (LOLBins)-Fileless attack type.

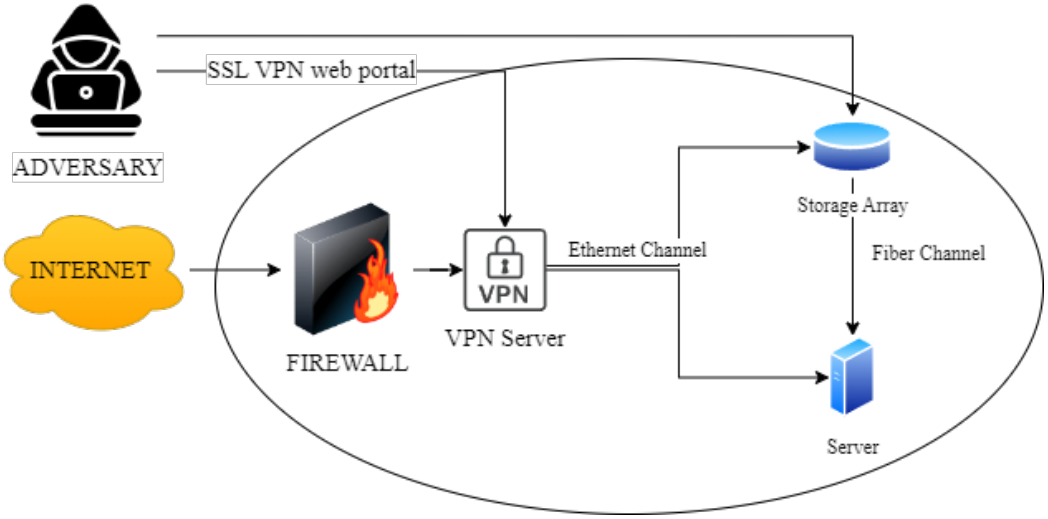

Figure 14: General Topology Scheme with VPN

TTPs to DNS Spoofing AV

An AV example for the scenario where the adversaries will get access and manipulate the Storage Array/Server when they have access to Lenovo Servers, using a tactic of spoofing attack and acting as Lenovo over DNS.

Figure 15: Steps - Pathway of AV VPN Server DNS Spoofing

- **Step 1**: Gain access to the network
  The **Initial Access** tactic will first initiate access to the industrial network.
  – Using the *Exploit Public-Facing Application*[65], where the adversary focuses on the network's access. The adversary exploits the internet-facing software that may be users' applications, weak

[64] Industroyer, Software S0604 | MITRE ATT&CK®
[65] Exploit Public-Facing Application, Technique T0819 - ICS | MITRE ATT&CK®

defenses etc. Publicly exposed applications can be found through online tools that check for open ports and services. This type of application may be used to have the ability to target specific known vulnerabilities.

Exploit Protection mitigation detects and blocks conditions that lead to or are traced to a software exploit. Application Isolation and Sandboxing limit an exploited target's access to other processes and system features. Built-in examples include software restriction policies AppLocker for Windows *and SELinux or AppArmor for Linux.*

Procedure Example - *Sandworm Team*[66] *where the adversaries as actors exploited vulnerabilities in different kind of software which had been exposed directly to the internet.*

• **Step 2**: Avoid detection and removal.
Evasion[67] the tactic consists of techniques that adversaries employ to bypass security defenses to remain stealthy and removal of compromise indicators, communication spoofing and the exploitation of software vulnerabilities within an ICS environment.

– Using the **Spoof Reporting Message**[68] technique, adversaries might falsify reporting messages in ICS to evade detection and disrupt process control. These messages include telemetry data such as I/O values, that reflect the current state of equipment and the industrial process. They can affect the control systems in various ways, such as sending falsified messages indicating that the process is functioning correctly to avoid detection.

Procedure Example - Maroochy Water Breach[69] where the adversary sends false data and instructions, affecting the control system and forcing them with wrong data. Back in 2000 an incident on wastewater control system affected and released 800,000 liters into the local community.

With **Domain Name System (DNS) spoofing**[70], an adversary manipulates DNS records to redirect traffic to a fraudulent or "spoofed" website. Once on the fraudulent site, victims may enter sensitive information that the adversary can then use or sell. Additionally, the adversary might construct a poor-quality site with derogatory or inflammatory content to damage the reputation of the industrial partner.

In a DNS spoofing attack, the adversary takes advantage of the fact that the user thinks the site they are visiting is legitimate, which exploits the user's trust. This allows the adversary to commit crimes in the name of an innocent industrial partner, at least from the visitor's viewpoint.

To prevent DNS spoofing, ensure your DNS servers are always up-to-date. Adversaries aim to exploit vulnerabilities in DNS servers, and the latest software versions often contain fixes that close known vulnerabilities.

• **Step 3**: Collecting information.
Discovery tactic is critical for adversaries to understand the network by collecting information, they can tailor their AV, identify weaknesses, find valuable data and move through the network.

---

[66] Sandworm Team, ELECTRUM, Telebots, IRON VIKING, BlackEnergy (Group), Quedagh, Voodoo Bear, IRIDIUM, Seashell Blizzard, FROZENBARENTS, Group G0034 | MITRE ATT&CK®
[67] Evasion, Tactic TA0103 - ICS | MITRE ATT&CK®
[68] Spoof Reporting Message, Technique T0856 - ICS | MITRE ATT&CK®
[69] Maroochy Water Breach, Campaign C0020 | MITRE ATT&CK®
[70] Top 20 Most Common Types Of Cyber Attacks | Fortinet

– Using the **Network Sniffing**[71] technique, adversaries might attempt to sniff the tragic to collect important information such as user credentials. ARP and DNS poisoning can be used to capture credentials of websites, proxies and internal systems by redirecting traffic to adversaries.

Procedure Example - INCOTROLLER[72] is a specialize malware composed of multiple modules designed to target ICS devices. IT specifically focuses on PLCs and can interact with industrial protocols like ModBus. Has the ability to identify targeted devices, download logic on the devices, and exploit specific vulnerabilities in those systems.

To avoid the sniffing technique there is a mitigation *Privileged Account Management* which is a restricted root or administrator access on the user account that limits the ability to capture traffic on a network using common packet capture tools by the adversaries.

TTPs to LOLBins-Fileless AV

An AV example for the other scenario is where the adversaries act as developers via an application on a Virtual Machine (VM) that are affecting the system. The developer needs external libraries which are online, downloads a credential dumping tool or malware LOLBins-Fileless[73] which is infected (i.e. supply chain software vulnerabilities) and executes it for the installation, but the malware in the background is performing some other actions.

The LOLBins is an attack type that leverages a trusted application, allowing adversaries to remain hidden-stealthy running camouflage actions, making it harder for security measures to detect and respond quickly.

Fileless is a malware type that exists as memory-based, leaving minimal or no trace on the hard drive. Because these procedures are not installing standard malicious software is particularly challenging for antivirus tools to detect them. This characteristic makes fileless malware more difficult to be addressed compared to other types. However, since it doesn't write to the disk it removes once the system is rebooted.

Below on Figure 16 are the TTPs of MITRE ATT&CK Framework that will be used.

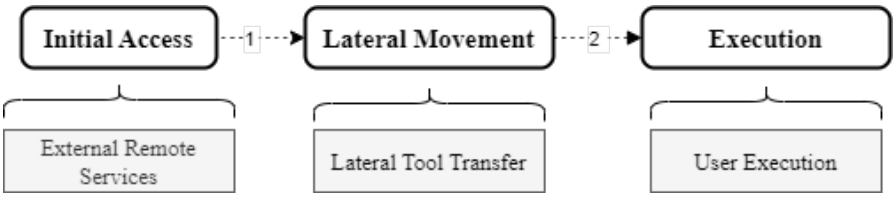

Figure 16: Steps - Pathway of AV VPN Server LolBins-Fileless

- **Step 1**: Firstly, the adversaries are using credential dumping tool or malware, which exploits a vulnerability in Fortigate VPN servers (**CVE-2018-13379**)[74] in case to steal the authentication credentials using Mimikatz an open-source application. They can use the **Initial Access** tactic with *External Remote Services*[75] technique where adversaries can have access to connect to the network externally, as they monitor for an entry point on the ICS network. For example, VPN implementations at trusted 3rd party networks or within remote support developer connections where the split tunnelling feature is active.

[71] Network Sniffing, Technique T0842 - ICS | MITRE ATT&CK®
[72] INCONTROLLER, Software S1045 | MITRE ATT&CK®
[73] What Are LOLBins and How Do Attackers Use Them in Fileless Attacks? (cynet.com)
[74] Ransomware crooks are targeting vulnerable VPN devices in their attacks | ZDNET
[75] External Remote Services, Technique T0822 - ICS | MITRE ATT&CK®

- **Step 2**: After that provide them with the ability to move within the network using the **Lateral Movement** tactic - *Lateral Tool Transfer* technique described in the 3.5.2 section.
- **Step 3**: Installing the infected external libraries such as Backdoor.Oldrea[76]/Bad Rabbit using the **Execution** tactic - *User Execution* technique described in the 3.5.1 section.

To prevent adversaries from performing this kind of AV, there are some examples of mitigations, that the MITRE ATT&CK framework provides are strong MFA[77] - prevent adversaries from gaining access and update regularly Antivirus/Antimalware[78] which are used to detect infected applications.

3.5.4 Example of AV for FDI on a Database

The communication network topology scheme example below includes the configuration of the energy community pilot, which consists of PhotoVoltaic (PV) plants connected to Medium Voltage (MV) electrical power and an EPES through medium/low-voltage (MV/LV) step-up transformers.

A SCADA data gateway (SDG) is physically attached to the cellular gateway using a wired connection. All the data exchanged between SDG and each PV plant is stored in a local Structured Query Language (SQL) server database as shown in Figure 17: General Topology Scheme with SQL Server DatabaseFigure 17: General Topology Scheme with SQL Server Database.

Using a 4G cellular network, the 4G router of all PV plants communicates wirelessly with a cellular gateway located at the operator's premises. A 4G router is physically attached to each smart logger using a wired connection. This router belongs to the distribution system operator and provides access to the smart logger, thus enabling the remote monitoring and control of each PV plant of the energy community. The IEC 60870-5-104 standard is used as the main communication protocol.

[76] Backdoor.Oldrea, Software S0093 | MITRE ATT&CK®
[77] Multi-factor Authentication, Mitigation M0932 - ICS | MITRE ATT&CK®
[78] Antivirus/Antimalware, Mitigation M0949 - ICS | MITRE ATT&CK®

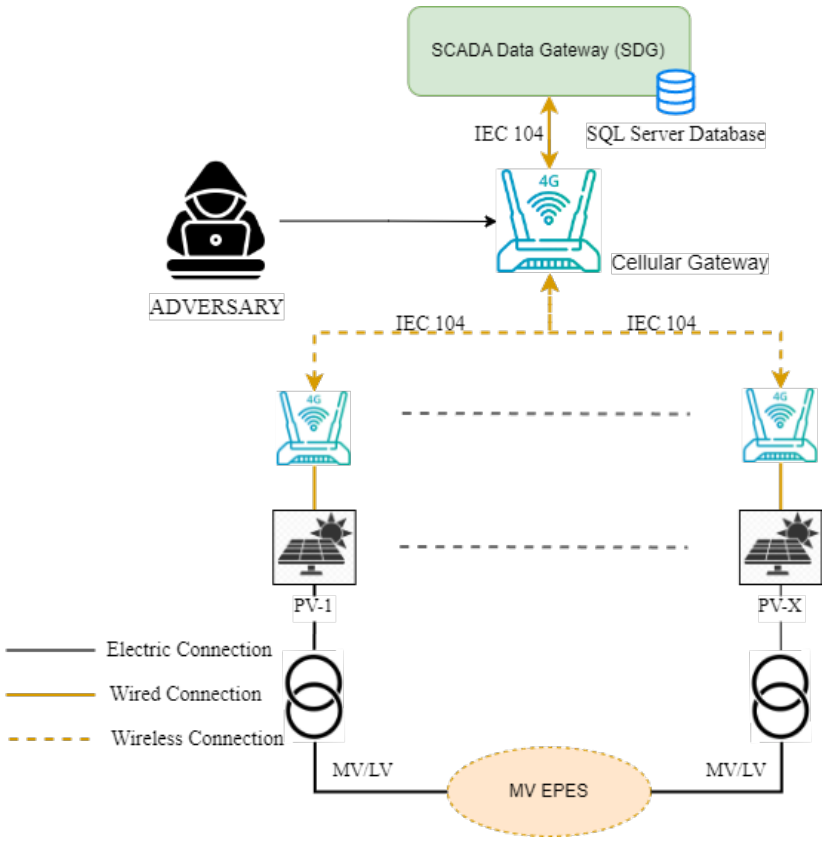

Figure 17: General Topology Scheme with SQL Server Database

TTPs to Bypass Gateway and attempt FDI AV

Below we will describe the steps of an AV that we will emulate, as an example of a scenario where an adversary has a scope to interrupt and impact the network. The main objective is to try to manipulate the data with FDI in the database, to transmit incorrect/faulty instructions from the working station to the setpoints, resulting in typical erroneous behavior. The false data and instructions affect the control system and force them with the wrong actions.

Initially using a known "Oncell Gateway Firmware" vulnerability CVE-2012-3039[79] on the cellular gateway device with installed firmware below version 1.4, which doesn't use a sufficient source of entropy for Secure Shell (SSH) and Secure Sockets Layer (SSL) keys. The adversaries can obtain access easier by leveraging knowledge of a key from a product installation from somewhere else.

Figure 18: Steps-Pathway of AV SQL Server Database Figure 18 below shows the TTPs of MITRE ATT&CK Framework that will be used.

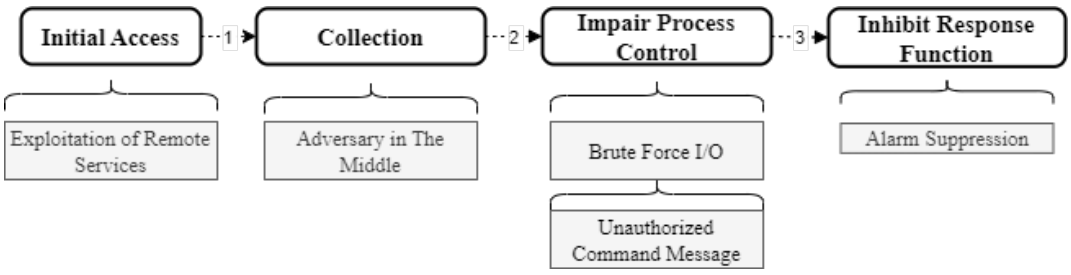

Figure 18: Steps-Pathway of AV SQL Server Database

[79] NVD - CVE-2012-3039 (nist.gov)

- **Step 1**: Attempt to exploit a software vulnerability for initial access and lateral movement to move within the network using the **Initial Access** tactic - *Exploitation of Remote Services* technique described in the 3.5.2 section. For this example, a different procedure Stuxnet[80] will be used that execute malicious SQL commands on the database server of remote systems in case to propagate. Another mitigation for this example procedure can be used the *Update Software,* routinely by implementing patch management for internal enterprise endpoints and servers.
- **Step 2**: Then the adversary can perform the same procedure example VPNFilter of 3.5.2 section that is using the **Collection** tactic – *AiTM* technique between the gateway and the SQL Server to perform the FDI attack, to manipulate the data that will be stored in the database. Another mitigation that can be used is an *Out-of-Band Communications Channel,* in case to validate the integrity of data.
- **Step 3**: Also, they can proceed with the **Impair Process Control** tactic – *Brute Force I/O*[81] the technique or the *Unauthorized Command Message* (similar example of 3.5.1 section) disrupt the normal process functionality of the system by receiving the wrong data on the database, forcing the devices to differently behavior. A procedure example is the Industroyer2[82] that created to cause impact to high-voltage substations.
  A mitigation strategy for that is to Filter Network Traffic which blocks access when excessive I/O connections are detected during a specified time.

- **Step 4**: Finally using the **Inhibit Response Function** tactic – *Alarm Suppression*[83] the adversaries can prevent any protection function alarms to notify the operators that something is going wrong with the process functionality or any critical condition. Using methods such as tampering or altering devices display warnings and logs. The scope-objective is to evade detection, so the operators don't proceed with any actions to respond to an error warning/log occurring. The procedure example is Maroochy Water Breach described in the 3.5.3 section.
  Another mitigation for this example procedure that can be used is *Network Allowlists*. These allowlists help to restrict unnecessary connections to network devices and services. For the specific device, they also enforce a limitation of simultaneous sessions they report.

[80] Stuxnet, Software S0603 | MITRE ATT&CK®
[81] Brute Force I/O, Technique T0806 - ICS | MITRE ATT&CK®
[82] Industroyer2, Software S1072 | MITRE ATT&CK®
[83] Alarm Suppression, Technique T0878 - ICS | MITRE ATT&CK®

4 COCOON Early Warning System

4.1 EWS Overview

In order to effectively track, profile and quantify the risk of threats as well as be in a position to effectively enable early detection at the onset of an AV it is crucial to orchestrate “on the fly” logic through a software component. The COCOON EWS will play this pivotal role by achieving the aforementioned properties and it will be instantiated as a service to operators within the COCOON Dashboard Toolset (CDT) via the COCOON Cyber Services Layer (CSL).

The EWS functions as a real-time monitoring and alerting system that continuously assesses the cyber-physical security posture of the power grid. It collects and analyses data from multiple sources within the network, applies sophisticated algorithms for anomaly detection, and triggers alerts for potential threats. Furthermore, it integrates a robust risk scoring framework, as discussed in Chapter 5, to ensure that identified threats are properly assessed and prioritized. This risk scoring system enables the EWS to make informed decisions about threat severity and necessary response actions in real-time. It enables timely intervention by operators, helping to prevent disruptions and maintain the stability and reliability of power grid operations.

The key objective of the EWS include:

1. Real-time threat detection as it is designed to detect cyber-physical threats as they occur, providing immediate alerts to system operators. This real-time capability is crucial for mitigating potential impacts before they escalate into significant disruptions.
2. Anomaly diagnosis in the network with the use of advanced ML techniques. By distinguishing between normal operational variances and actual threats, the system reduces false positives and enhances the accuracy of threat detection.
3. Proactive threat mitigation with the incorporation of threat mitigation mechanisms. It dynamically adapts to evolving threat landscapes, deploying appropriate countermeasures to neutralize identified risks.
4. Enhanced operator training components to improve the readiness of grid operators. Through simulated scenarios and practical exercises, operators are better prepared to respond to real-world cyber-physical incidents.
5. Inter-domain secure information exchange with secure communication across different domains within the power grid infrastructure, maintaining the integrity and confidentiality of critical data.
6. Integration with existing systems allowing it to work seamlessly with existing IT and OT systems. Thus, EWS leverages existing infrastructure while enhancing overall security capabilities.

The core features of EWS include:

7. Comprehensive data collection by continuously gathering data from various sensors and devices across the power grid, providing a holistic view of the network's operational state.
8. Advanced analytical tools which include sophisticated algorithms and ML models to analyze collected data, identifying patterns indicative of potential threats.
9. Automated alerting and reporting with detailed incident reports, providing operators with actionable insights for quick decision-making.
10. User-friendly interface with a dashboard that presents information in an intuitive format, allowing operators to easily monitor system status and respond to alerts.
11. Scalability and flexibility as it is designed to scale with the size and complexity of the power grid. EWS can be customized to meet the specific needs of different operational environments.

By integrating these objectives and features, the EWS stands as a comprehensive solution for enhancing the cyber-physical security of modern power grids. Its proactive, real-time approach ensures that threats are not only detected but also effectively mitigated, thereby safeguarding the critical infrastructure that underpins the stability and reliability of power systems.

4.2 EWS Architecture

The EWS architecture is designed to be adaptive, scalable, and compliant with industry guidelines and standards such as MITRE ATT&CK and CVSS. The EWSs architecture enhances its ability to continuously monitor, learn, and respond to evolving threats. To allow for better data handling and analysis, it consists of several layers as shown in Figure 19Figure 19. A description of each layers is given below and in Sections 4.3 (EWS Components) and 4.4 (EWS Data Flow).

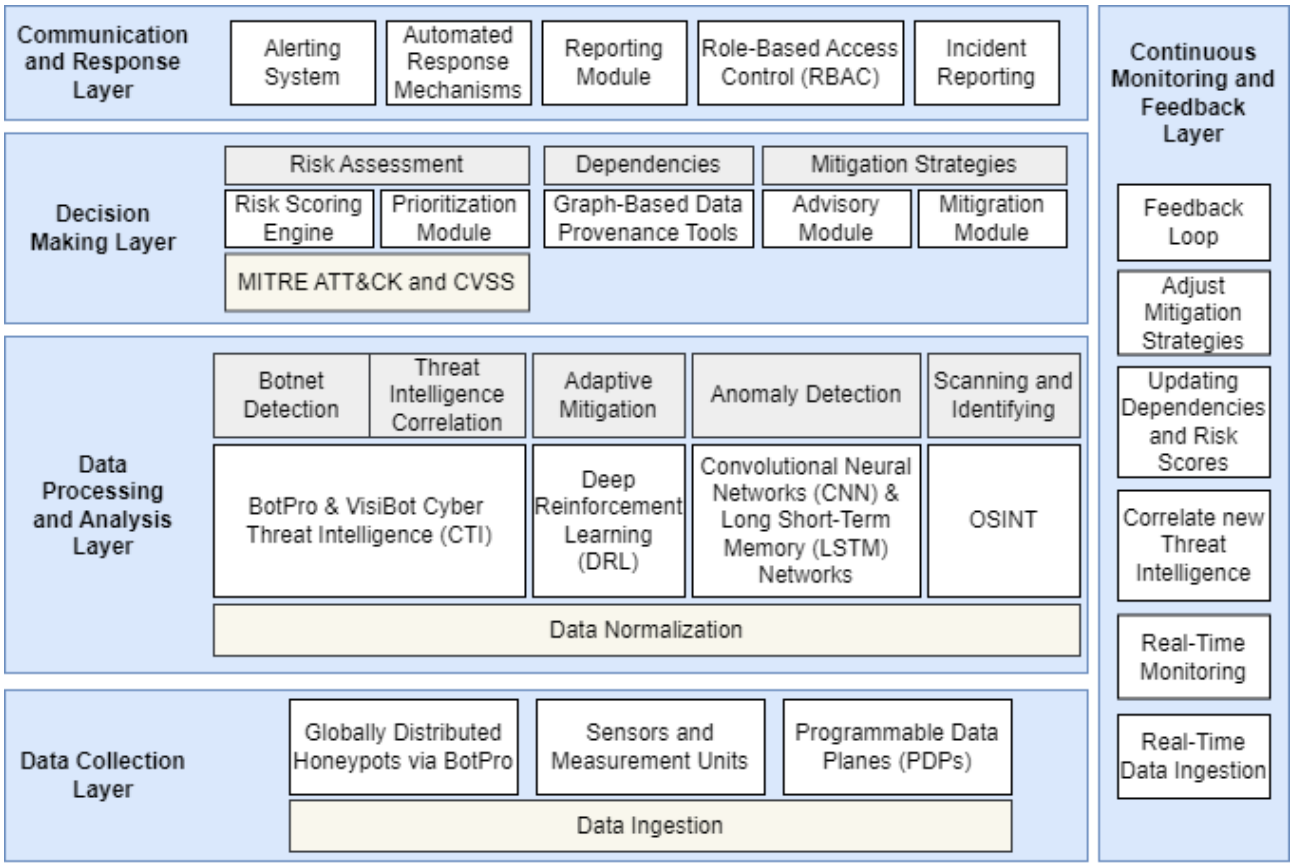

Figure 19: EWS Architecture

4.2.1 Data Collection Layer

The Data Collection Layer forms the foundation of the EWS by gathering critical data from various network sources. Programmable Data Planes (PDPs) play a crucial role in this layer, capturing packet-level data within the COMML. PDPs perform essential tasks such as packet parsing, flow statistics updates, and in-network aggregation, providing a granular view of network traffic that is indispensable for detecting anomalies.

Additionally, Sensors and Measurement Units are strategically installed across the grid infrastructure to monitor both cyber and physical parameters in real-time. These sensors ensure a comprehensive understanding of the operational environment, capturing data related to system performance, potential security threats, and environmental conditions.

Globally Distributed Honeypots via BotPro enhance this layer by attracting and detecting botnet activities and other malicious behavior. These honeypots are strategically deployed worldwide, feeding data directly into the BotPro module, which facilitates early detection of automated threats.

All data collected through PDPs, sensors, and honeypots undergoes Data Ingestion and Normalization, a critical process that ensures consistency and prepares the data for further analysis. This step is vital for standardizing the data across various formats and sources, making it ready for the sophisticated processing and analysis that follows in subsequent layers.

4.2.2 Data Processing and Analysis Layer

The Data Processing and Analysis Layer is where raw data collected from the network is transformed into actionable insights through advanced analysis tools and algorithms. BotPro is central to this layer, specializing in detecting botnet-related activities. It employs a combination of methods, including information theory, statistical analysis, ML, and graph theory, to identify and respond to threats, particularly those related to botnets.

VisiBot CTI complements BotPro by aggregating and correlating external threat intelligence from sources like OSINT feeds, Internet topology measurements, and IP blacklists. This tool provides a broader context for the detected threats, enriching the EWSs ability to respond to both internal and external security challenges.

Deep Reinforcement Learning (DRL) is another key component, used to implement adaptive threat mitigation strategies. DRL dynamically adjusts the system's responses based on the outcomes of previous actions, ensuring continuous improvement in threat detection and response.

Convolutional Neural Networks (CNN) & Long Short-Term Memory (LSTM) Networks are employed to classify known events and detect anomalies in multivariate time-series data. These ML models enhance the system's ability to identify unusual or potentially harmful behavior.

Additionally, tools like Shodan and Censys are used for scanning and identifying exposed devices and services within the IT/OT environment. These tools help map potential attack surfaces, providing critical insights for proactive defense strategies. All data is first subjected to Data Normalization to ensure compatibility and accuracy across different data sources, facilitating effective correlation and comparison.

4.2.3 Decision-Making Layer

The Decision-Making Layer is responsible for turning processed data into informed decisions regarding risks and mitigation strategies. Central to this process is the Risk Scoring Engine, which evaluates the potential impact of detected threats using frameworks such as the Common Vulnerability Scoring System (CVSS) and the risk scoring framework detailed in Chapter 2. By assigning risk scores to each threat, this engine helps prioritize which issues require immediate attention, ensuring that the most critical threats are addressed first.

The Prioritization Module takes these risk scores and ranks threats according to their severity, impact, and urgency. This prioritization is crucial for optimizing response efforts and ensuring that resources are allocated to the most pressing security challenges.

Graph-Based Data Provenance Tools provide a visual representation of network dependencies and risks. These tools map the relationships between various network components, highlighting how vulnerabilities might propagate through the system. This visual approach aids in understanding the broader impact of specific threats and supports more informed decision-making.

The Advisory Module plays a key role in recommending mitigation actions based on the identified threats and their risk scores. These recommendations ensure that responses are not only timely but also effective in neutralizing the specific nature of the threat. Finally, the Mitigation Module executes these recommended actions, which may include isolating affected systems, reconfiguring network settings, or applying security patches, thereby ensuring a swift and effective response to security incidents.

4.2.4 Communication and Response Layer

The Communication and Response Layer is vital for managing the dissemination of information and ensuring that appropriate actions are taken in response to detected threats. The Alerting System is the first line of communication, responsible for sending real-time alerts to relevant stakeholders, including security teams and system administrators. These alerts provide crucial information about detected threats and recommended actions, enabling quick and informed decision-making.

Automated Response Mechanisms are another critical component, designed to execute predefined actions automatically in response to certain types of threats. These actions may include blocking malicious IP addresses, updating firewall rules, or isolating compromised systems. By automating these responses, the system can minimize the window of exposure and limit the potential damage caused by an attack.

The Reporting Module generates detailed reports on the detected threats, the actions taken, and the overall security posture. These reports are essential for auditing, compliance, and post-incident analysis, helping industrial partners refine their security strategies and improve their response to future threats.

Role-Based Access Control (RBAC) is implemented to ensure that only authorized personnel can access sensitive information and perform critical actions within the EWS. This control is fundamental to maintaining the integrity and security of the system by minimizing the risk of unauthorized access or actions.

Finally, Incident Reporting manages the communication of incident details, ensuring that all relevant parties are informed promptly and accurately. Effective incident reporting is crucial for coordinating a swift and effective response, ultimately minimizing the impact of security breaches.

4.2.5 Continuous Monitoring and Feedback Loop

The Continuous Monitoring and Feedback Loop is a critical component of the EWS, ensuring that the system remains effective and adaptable over time. Real-Time Monitoring is at the core of this layer, continuously assessing network and system performance to detect any anomalies or potential threats. This continuous monitoring feeds data back into the system for immediate analysis and response, enabling the EWS to detect and respond to threats as they emerge.

The Feedback Loop integrates the outcomes of threat detection and mitigation actions back into the system, allowing the EWS to refine its detection algorithms and improve future responses. This ongoing feedback ensures that the system learns from past experiences, enhancing its ability to deal with new and evolving threats.

As part of this adaptive capability, the system Adjusts Mitigation Strategies based on the feedback received. By continuously refining these strategies, the EWS ensures that its responses remain effective against the latest threats, maintaining a robust defense posture.

The system also continuously Updates Dependencies and Risk Scores, revising these elements based on new data and insights. This process ensures that the EWS accurately reflects the current risk environment, staying up-to-date with the latest threat intelligence.

Finally, the EWS works to Correlate New Threat Intelligence with existing data, ensuring that it remains aware of and prepared for the latest threats. This continuous integration of new intelligence helps the system stay ahead of emerging threats, maintaining its effectiveness in protecting the network.

4.3 EWS Components

The EWS is built on a sophisticated and multi-faceted infrastructure designed to ensure the highest levels of cybersecurity for modern power grids. At its core, the EWS relies on a robust data collection and monitoring infrastructure. This includes PDP that enable packet-level primitives within the COMML. These PDPs facilitate various essential functions such as packet parsing, flow statistics updates, and in-network aggregation. Complementing this are network sensors and measurement units strategically installed within the grid infrastructure to gather real-time data on network performance, physical parameters, and operational states.

To tackle the ever-evolving landscape of cyber threats, the EWS incorporates advanced threat models and vulnerability assessment tools. The MITRE ATT&CK Framework plays a crucial role here, providing a structured approach to design functional attack scenarios, including APTs that target specific vulnerabilities. In tandem, the BotPro CTI Tool leverages data-driven techniques to correlate exogenous sources like OSINT feeds and Internet topology measurements, thereby tracking vulnerabilities and exploits in real-time with high precision.

Central to the EWSs functionality is its anomaly diagnosis and classification capability, which employs state-of-the-art deep learning methods. By utilizing CNN and LSTM networks, the system can classify known events and identify deviations in multivariate time-series data. This process is further enhanced through convergence with OT parameters, including voltage, frequency, and power flows, ensuring accurate and reliable anomaly detection.

The EWS also integrates sophisticated threat mitigation strategies to neutralize detected threats. It leverages DRL to implement dynamic and adaptive threat mitigation at the packet level. This involves deploying policies such as packet dropping, load balancing, and traffic reshaping based on the identified threats, thus providing a comprehensive defense mechanism. Additionally, real-time protection schemes ensure the cyber-secure operation of critical infrastructure elements like digital substations and Distributed Renewable Energy Source (DRES) deployments.

To facilitate effective communication and response, the EWS features automated incident reporting and communication tools. The COCOON Toolset Dashboard (CTD) offers a user-friendly interface for operators to receive alerts, view detailed incident reports, and initiate mitigation actions. The automated incident reporting function generates comprehensive reports on detected threats, detailing the nature of the threat, affected systems, and recommended mitigation steps, thereby ensuring a coordinated and informed response to cyber incidents.

Integration with existing security frameworks is another key aspect of the EWS. It adheres to RBAC to ensure that access to EWS functionalities is restricted based on user roles. Additionally, the EWS complies with relevant cybersecurity standards like NIST and IEC 62351, facilitating interoperability and secure communication within the grid infrastructure.

Functionality and implementation of the EWS prioritize high detection accuracy and rapid response times to minimize the impact of cyber-physical attacks. The system undergoes extensive simulation and validation in controlled environments before real-world deployment to ensure its robustness and reliability. Pilot studies and real-world applications, such as secure energy communities and digital

substation security, demonstrate the EWSs capabilities in managing and protecting power grid infrastructures against sophisticated cyber threats.

In summary, the EWS in the COCOON project is a comprehensive cybersecurity solution that combines advanced data collection, threat modelling, anomaly diagnosis, and threat mitigation to protect modern power grids. Its proactive and real-time approach ensures that threats are effectively detected and neutralized, maintaining the stability and reliability of critical power infrastructure.

4.4 EWS Data Flow

To better understand the data flow within the EWS, it's helpful to consider practical scenarios that illustrate how each component functions in real-world situations. After analyzing the individual components of the EWS, we can apply this knowledge through specific use cases. These examples provide a clear picture of how data moves through the system from detection to response.

1. Detection of a DDoS Attack (Section 4.4.1 Use Case 1: Detection of a DDoS Attack): This use case explores how the EWS identifies and mitigates a large-scale DDoS attack targeting critical infrastructure, from initial detection by sensors to the execution of mitigation strategies.
2. Early Detection of a Botnet Infection (Section 4.4.2 Use Case 2: Early Detection of a Botnet Infection): In this scenario, the EWS detects an IoT device infected by a botnet, analyzes the threat, and implements containment measures to prevent the botnet from spreading.
3. Identifying Insider Threats (Section 4.4.3 Use Case 3: Identifying Insider Threats): This example demonstrates how the EWS detects suspicious behavior from an employee's account, assesses the risk of an insider threat, and responds to protect sensitive data from unauthorized access.

4.4.1 Use Case 1: Detection of a DDoS Attack

In a scenario where a DDoS attack is launched against critical infrastructure within a smart grid, the EWS plays a crucial role in detecting and mitigating the threat. The attack typically begins with a significant increase in network traffic, aimed at overwhelming resources and disrupting services. The process starts in the Data Collection Layer, where PDPs and Sensors and Measurement Units detect unusual traffic patterns that indicate the onset of a DDoS attack (the data flow in EWS is shown in Figure 20). These components provide a granular view of the traffic, capturing the increase in data flow. Simultaneously, Globally Distributed Honeypots via BotPro collect data from botnets contributing to the attack, providing early insights into the nature of the threat.

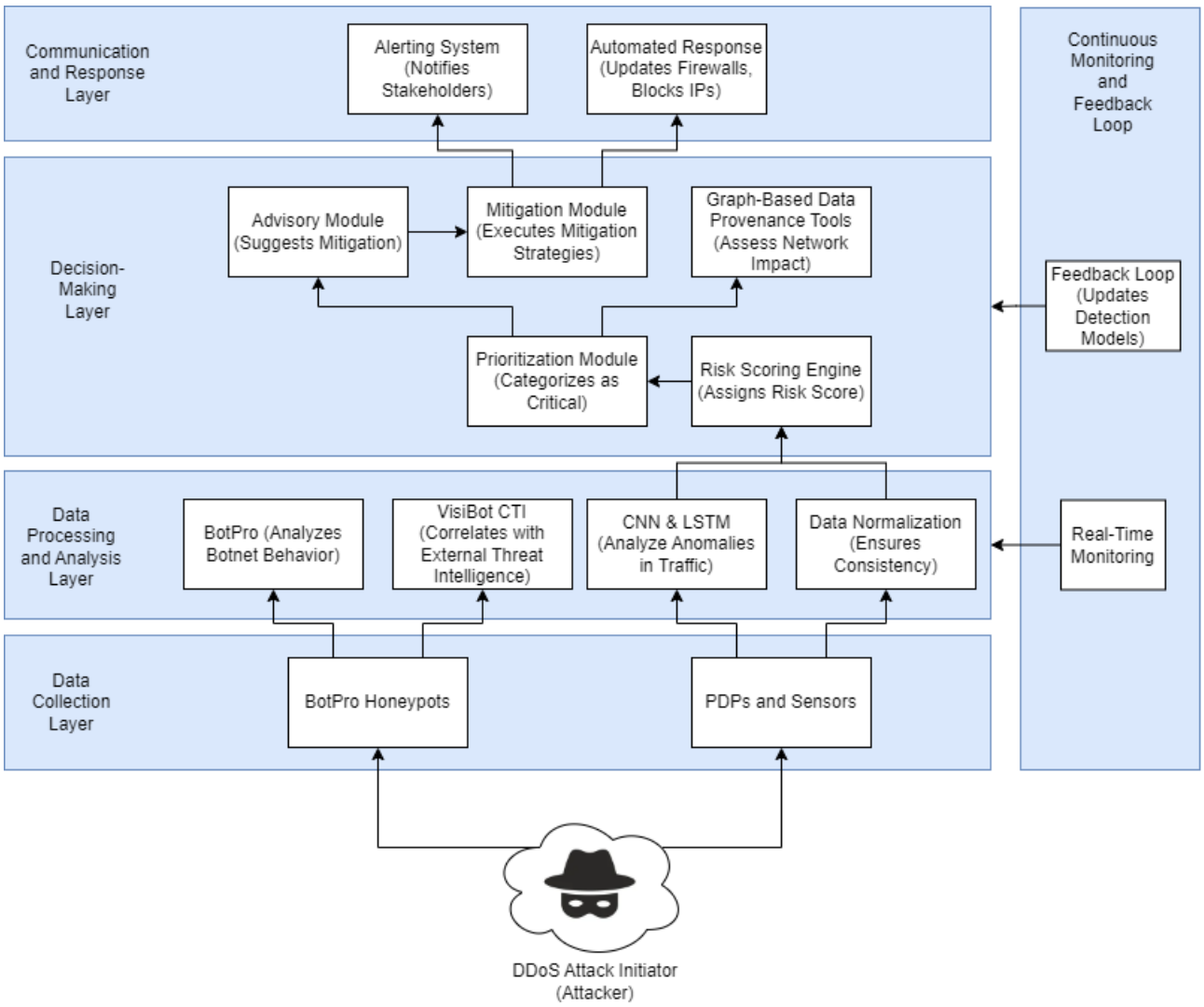

Figure 20: Use Case 1 EWS Data Flow

As the data moves into the Data Processing and Analysis Layer, BotPro analyzes the behavior of the botnets involved, identifying C&C communications and the distribution of attack sources. VisiBot CTI correlates this information with external threat intelligence feeds, recognizing known malicious IPs and attack patterns associated with DDoS activities. ML models, particularly CNN and LSTM Networks, are employed to analyze traffic data for anomalies, confirming the presence of a DDoS attack. All data collected undergoes normalization to ensure consistency, facilitating further analysis.

In the Decision-Making Layer, the Risk Scoring Engine assigns a high-risk score to the detected DDoS attack, based on the volume and impact of the traffic. This score triggers the Prioritization Module, which categorizes the event as critical, prompting immediate response protocols. Graph-Based Data Provenance Tools assess the attack's impact on the network, identifying vulnerable nodes and systems. The Advisory Module then recommends mitigation strategies, such as rate limiting, traffic rerouting, or isolating targeted systems. These strategies are executed by the Mitigation Module to neutralize the threat.

As the attack progresses, the Communication and Response Layer ensures that the security team and relevant stakeholders are notified through the Alerting System, providing real-time updates on the situation. Automated Response Mechanisms are activated, implementing additional countermeasures like updating firewall rules to block the malicious IPs identified by VisiBot CTI. The Reporting

Module documents the entire incident, detailing the attack, the response actions taken, and the overall impact, which is crucial for post-incident analysis.

Finally, the Continuous Monitoring and Feedback Loop plays a vital role in tracking the attack's progress and the effectiveness of the mitigation measures. Real-time monitoring continues to observe network activity, while the Feedback Loop updates the system's understanding of DDoS attack patterns, enhancing future detection capabilities. As the nature of the attack evolves, the EWS dynamically adjusts its mitigation strategies to ensure continued effectiveness, thereby safeguarding the infrastructure against ongoing threats.

4.4.2 Use Case 2: Early Detection of a Botnet Infection

Consider a scenario where an IoT device within the network becomes infected by a botnet, posing a significant threat to the entire infrastructure. The EWS is designed to detect and respond to such threats with precision and speed (the data flow in EWS follows the same path as with use case 1 as shown in Figure 20). The detection process begins in the Data Collection Layer, where Sensors and Measurement Units identify abnormal behavior in the IoT device, such as unusual outbound traffic or attempts to connect to known malicious IPs. This initial detection is crucial in identifying the early stages of botnet activity. Simultaneously, Globally Distributed Honeypots via BotPro capture similar behavior from other compromised IoT devices, providing additional context and confirming the presence of a widespread infection.

Once the data is collected, it is processed in the Data Processing and Analysis Layer. Here, BotPro identifies the specific botnet's signature and analyzes its communication patterns, determining the scope of the infection. This analysis is enhanced by VisiBot CTI, which correlates the gathered information with external threat intelligence to identify the botnet variant and its known capabilities. The system's ML Techniques are then applied to classify the behavior of the infected device, confirming its participation in the botnet. During this process, Data Normalization ensures that all information is consistent and accurately formatted, enabling precise analysis.

In the Decision-Making Layer, the Risk Scoring Engine evaluates the severity of the botnet infection based on the botnet's known capabilities and the criticality of the infected IoT device. This assessment allows the Prioritization Module to categorize the incident as high priority, given the potential for the botnet to spread across the network. Graph-Based Data Provenance Tools are employed to map the dependencies of the infected device, identifying other vulnerable systems that might be at risk of infection. The Advisory Module then suggests appropriate mitigation actions, such as isolating the infected device, updating its firmware, or performing a network-wide scan to detect and address similar infections. These recommendations are promptly executed by the Mitigation Module to contain the infection and prevent further spread.

As these actions are carried out, the Communication and Response Layer ensures that the IT and security teams are informed through timely alerts issued by the Alerting System. Automated Response Mechanisms may also be triggered, leading to actions such as disconnecting the infected device from the network or deploying security patches to other potentially vulnerable devices. The Reporting Module logs the incident, detailing the infection's origin, the response actions taken, and any lessons learned, which are essential for future reference and improving security protocols.

In the final stage, the Continuous Monitoring and Feedback Loop plays a critical role in ensuring the EWS remains vigilant. Real-time monitoring continues to track network activities, looking for any signs of further botnet-related behavior. The Feedback Loop refines the detection models based on the observed behavior of the botnet, enhancing the system's ability to identify similar threats in the future. Additionally, the system dynamically adjusts its mitigation strategies as new information

about the botnet becomes available, ensuring that the EWS remains effective against evolving threats. This continuous cycle of monitoring, feedback, and adaptation is vital in maintaining the security and integrity of the network.

4.4.3 Use Case 3: Identifying Insider Threats

In a scenario where suspicious behavior from an employee's account suggests a possible insider threat, such as unauthorized access to sensitive data, the EWS is crucial for timely detection and response (the data flow in EWS is shown in Figure 21). The process begins in the Data Collection Layer, where PDPs and Sensors collect comprehensive logs of network activities, including access to critical files and systems. These logs, along with data from HR systems, access control logs, and network activity records, are aggregated through Data Ingestion, which consolidates the information from various sources.

The collected data is then processed in the Data Processing and Analysis Layer. VisiBot CTI plays a critical role by providing external context, correlating internal data with known insider threat tactics, and identifying if the suspicious behavior aligns with known patterns. ML Techniques are employed to analyze the user's behavior, comparing it against established baselines to detect anomalies that might indicate malicious intent. Additionally, NLP techniques are used to analyze communication logs, such as emails and messages, to detect sentiment changes or language that could signal a potential threat. The Data Normalization process ensures that all data is consistently formatted, facilitating accurate analysis.

In the Decision-Making Layer, the Risk Scoring Engine evaluates the potential risk based on the nature of the accessed data and the user's role within the industrial partner. This evaluation allows the Prioritization Module to assess the urgency of the threat, particularly in cases where sensitive data is at risk. Graph-Based Data Provenance Tools are used to map the relationships between accessed files, identifying potential paths of data leakage and helping to determine the broader implications of the insider activity. The Advisory Module then provides recommendations for immediate actions, such as limiting the user's access, closely monitoring the account, or launching a full investigation. These recommended actions are implemented by the Mitigation Module, which may involve restricting access or notifying HR and security teams for further investigation.

As the situation unfolds, the Communication and Response Layer ensures that all relevant parties are kept informed. The Alerting System sends notifications to security and HR teams about the potential insider threat, detailing the actions being taken and any immediate concerns. In some cases, Automated Response Mechanisms may be activated, such as temporarily suspending the account's access to sensitive resources until a thorough investigation is completed. The Reporting Module documents the incident, including all actions taken and outcomes, which is crucial for compliance, auditing, and future prevention strategies.

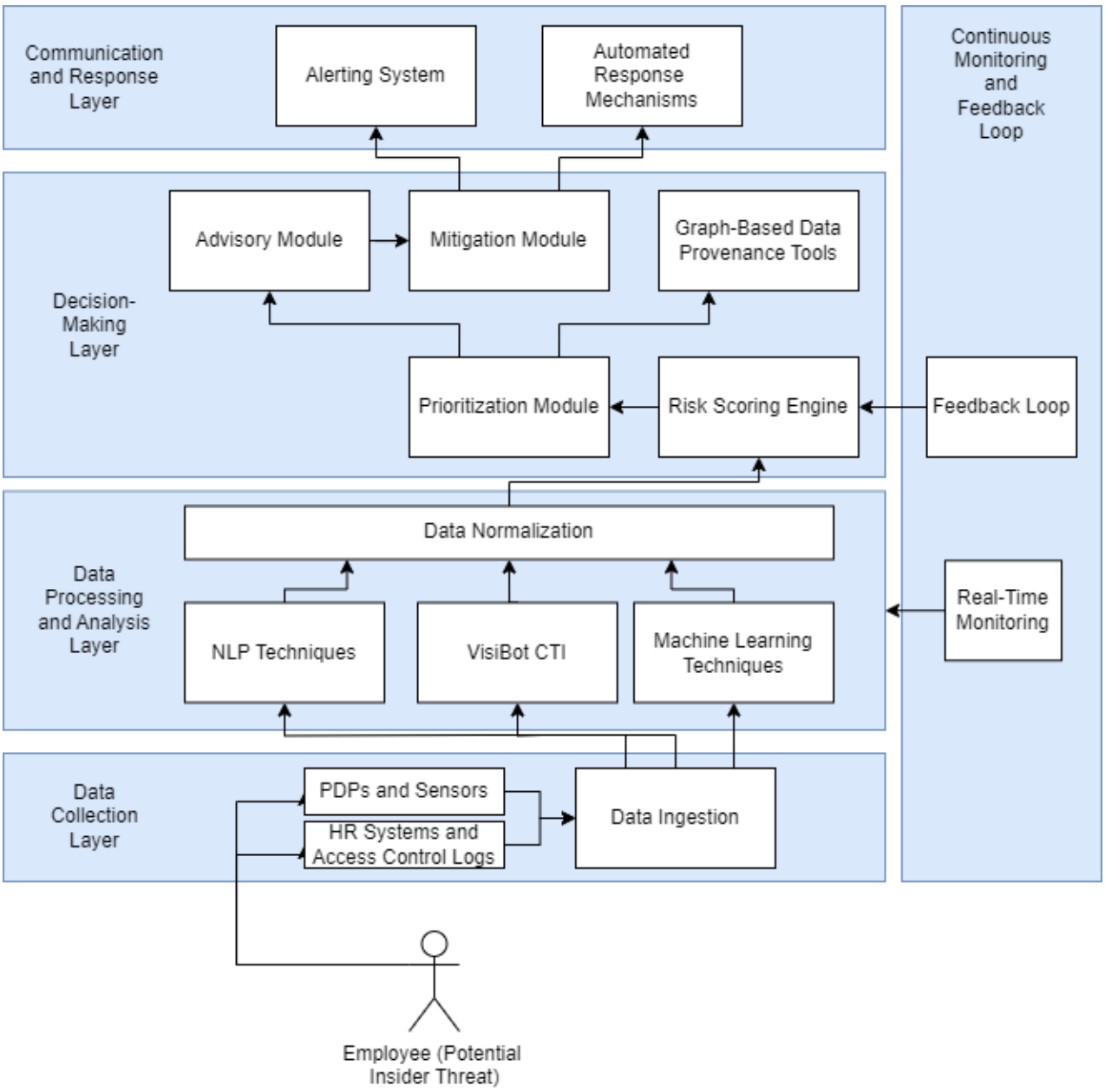

Figure 21: Use Case 3 EWS Data Flow

Finally, the Continuous Monitoring and Feedback Loop plays an essential role in ensuring that the EWS remains effective against insider threats. Real-time monitoring continues to observe the user's activities, looking for further signs of suspicious behavior. The Feedback Loop updates the system's behavioral models based on the insights gained from the incident, improving the EWSs ability to detect similar threats in the future. As new information is gathered, the system dynamically adjusts its mitigation strategies, ensuring a responsive and adaptable defense against insider threats. This continuous process of monitoring, feedback, and adaptation is critical for maintaining the integrity and security of the industrial partner, particularly in protecting against the nuanced and often complex nature of insider threats.

4.5 BotPro

BotPro [[14][17]] is a comprehensive data-driven framework developed by UCY and UGLA to profile IoT botnet behavior. It aims to capture and highlight the behavioral properties of IoT botnets with respect to their structural and propagation properties across the global Internet. The BotPro framework is implemented using real-world data obtained from the measurement infrastructure of the

CPN (e.g., the COMML layer of the CPN) deployed on the actual EPES infrastructure of the COCOON pilots. This infrastructure gathers data from various sources, including globally distributed honeypots, regional Internet registries, global IP blacklists, and routing topology. This diverse dataset forms a strong foundation for profiling IoT botnet activity, ensuring that the analysis accurately reflects the behavioral patterns of botnets in real-world scenarios. BotPro has been integrated into EWS and plays a pivotal role to various parts of EWS as described in Section 4.2 EWS Architecture.

The BotPro framework employs a variety of methods to profile IoT botnets, including information theory, statistical analysis, NLP, ML, and graph theory. These diverse methods provide a comprehensive approach to understanding the structure, behavior, and evolution of IoT botnets.

4.5.1 BotPro Framework Key Components

BotPro framework has 5 key components starting from data collection up to real-time analysis as shown in Figure 22. An overview of each component is given in the list below.

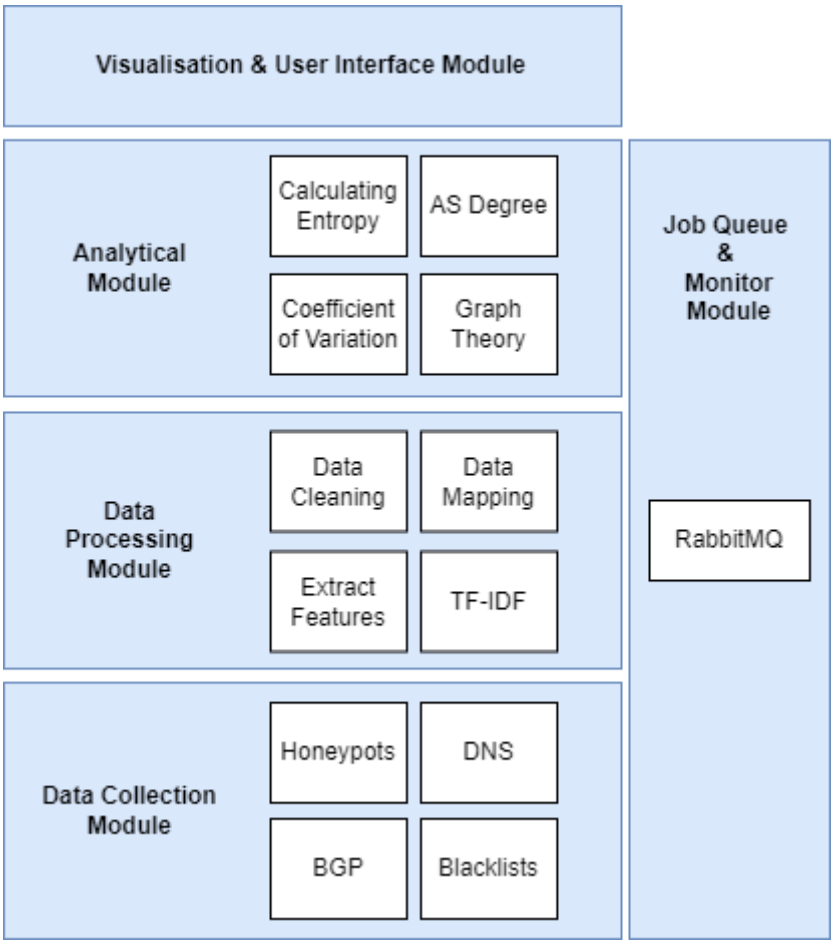

Figure 22: BotPro System Architecture

1. **Data Collection Module**
   BotPro's data collection is powered by a sophisticated measurement infrastructure that gathers real-world data from multiple sources. These include globally distributed honeypots, regional Internet registries, global IP blacklists, and routing topology data.
   a. The honeypots are deployed globally to capture real-time botnet activity. They provide crucial data on attack patterns, botnet size, and the geographical distribution of botnet nodes.

b. IP Blacklists and BGP Data: BotPro integrates data from global IP blacklists and BGP (Border Gateway Protocol) routing information to understand the geographical and topological aspects of botnet propagation.

2. **Data Processing Module**
   Collected data is first normalized to ensure consistency across different data sources. This step is crucial for the accurate analysis and correlation of data.
   a. BotPro employs graph theory to analyze the structural properties of IoT botnets, including the relationships between botnet nodes and the Autonomous Systems (ASes) they reside in. Statistical tools are also used to identify patterns and trends in the data, helping to uncover the underlying dynamics of botnet operations.
   b. BotPro utilizes ML models, including clustering algorithms and anomaly detection techniques, to classify botnet activities and predict future behavior. These models help identify new botnet variants and their evolving strategies.
3. **Analytical Module**
   a. The core of BotPro's analytical capabilities lies in its ability to profile botnet behavior. This includes analyzing the scanning and propagation strategies of botnets, understanding the role of botnet loaders, and evaluating the effectiveness of IP blacklists in capturing botnet activities.
   b. BotPro provides a macroscopic view of how different ASes contribute to botnet propagation. It examines the structural characteristics of ASes and their relationships to understand how botnets exploit these networks for spreading malware.
   c. A significant focus is placed on understanding botnet loaders, which are critical for the propagation of IoT botnets. BotPro analyzes the distribution and behavior of malware binaries across different ASes, providing insights into the tactics used by botnet loaders to evade detection and enhance botnet resilience.
4. **Visualization and User Interface Module**
   a. BotPro features a user-friendly interface with interactive dashboards that allow security analysts to visualize botnet activity in real-time. This includes geographic maps showing the distribution of botnet nodes, network topologies illustrating AS relationships, and time-series graphs of botnet activity.
   b. The system provides real-time monitoring capabilities, with alerts triggered by significant changes in botnet behavior or the detection of new botnet variants. This helps industrial partners respond promptly to emerging threats.
5. **Real-Time Data Processing and Analysis**
   a. For real-time data processing, BotPro integrates RabbitMQ, a message queuing system that ensures efficient handling of large volumes of data. This allows BotPro to process and analyze data as it is collected, providing immediate insights into botnet activity.
   b. BotPro is designed to handle large-scale IoT botnets, with optimizations for both speed and accuracy. The framework can scale to process data from a vast number of IoT devices, making it suitable for monitoring global botnet activities.

4.6 BotPro Algorithmic Properties

The algorithmic properties of BotPro are critical to its effectiveness in profiling IoT botnet behavior. These properties are built on advanced statistical and computational techniques that ensure the accuracy, efficiency, and scalability of the framework.

4.6.1 Graph Theory and Centrality Measures

Graph theory forms a crucial component of BotPro's approach to analyzing the structural dynamics of IoT botnets. Within BotPro, botnets are represented as graphs, where each node corresponds to an

individual device, and edges signify the connections between these devices. The application of centrality measures is key to understanding the influence and importance of specific nodes within the botnet. Degree centrality highlights nodes with the most connections, identifying devices that are heavily involved in botnet communication and potentially serving as hubs for coordinating attacks. Betweenness centrality is used to identify nodes that act as bridges within the network, controlling the flow of information between different parts of the botnet. These nodes are often critical to the botnet's operation and are prime targets for disruption. Closeness centrality measures how quickly information can spread from a node to all other nodes in the network, helping to identify devices that can efficiently propagate commands across the botnet. By using these centrality measures, BotPro can pinpoint nodes that are crucial to the botnet's functionality, enabling more effective targeting of interventions. Disrupting these key nodes can significantly impair the botnet's ability to operate, making graph theory a powerful tool in the fight against IoT botnets.

4.6.2 Statistical Analysis

Statistical analysis in BotPro is essential for identifying patterns, trends, and anomalies in the data collected from IoT botnets. The framework employs a variety of statistical techniques to make sense of the vast amounts of data it processes. Clustering algorithms group similar botnet behaviors together, revealing patterns that might indicate a coordinated effort or a common origin among different botnets. This is particularly useful in identifying new botnet variants or understanding how botnets evolve over time. Regression analysis is employed to quantify relationships between variables, such as the frequency of attacks and the geographical distribution of botnet nodes. By understanding these relationships, BotPro can predict how certain factors influence botnet behavior, which can inform strategic decisions about where to focus defense efforts. Hypothesis testing is used to assess the significance of observed behaviors, distinguishing between normal network traffic and potential botnet activity. This statistical rigor ensures that BotPro can reliably detect deviations from expected behavior, which is critical for early threat detection. The ability to uncover these hidden patterns and relationships makes statistical analysis a powerful tool in BotPro's arsenal, enabling it to provide deep insights into the operational strategies of IoT botnets and to anticipate future actions.

4.6.3 Machine Learning Techniques

Machine Learning (ML) is integral to BotPro's advanced analytical capabilities, enabling it to classify, predict, and adapt to IoT botnet behaviors with high accuracy. BotPro utilizes both supervised and unsupervised learning methods to process and analyze the vast datasets it collects. In supervised learning, models like decision trees, Support Vector Machines (SVMs), and neural networks are trained on labeled datasets to recognize and classify known botnet activities. These models can identify specific types of botnet behaviors and categorize them based on learned patterns, which is crucial for recognizing known threats quickly and accurately. On the other hand, unsupervised learning techniques, such as clustering and anomaly detection, are used to discover previously unknown patterns in the data. These methods do not rely on labeled data and are therefore particularly effective at identifying new and emerging threats that have not yet been categorized. Anomaly detection is especially valuable in spotting behaviors that deviate from the norm, which could indicate a novel AV or an evolving botnet strategy. BotPro's ML algorithms are designed to improve over time by learning from both historical and real-time data. This continuous learning process enhances BotPro's ability to stay ahead of evolving threats, making it a dynamic and robust tool for IoT botnet detection and mitigation. Scanning profiling patterns generated by BotPro are shown in Figure 23.

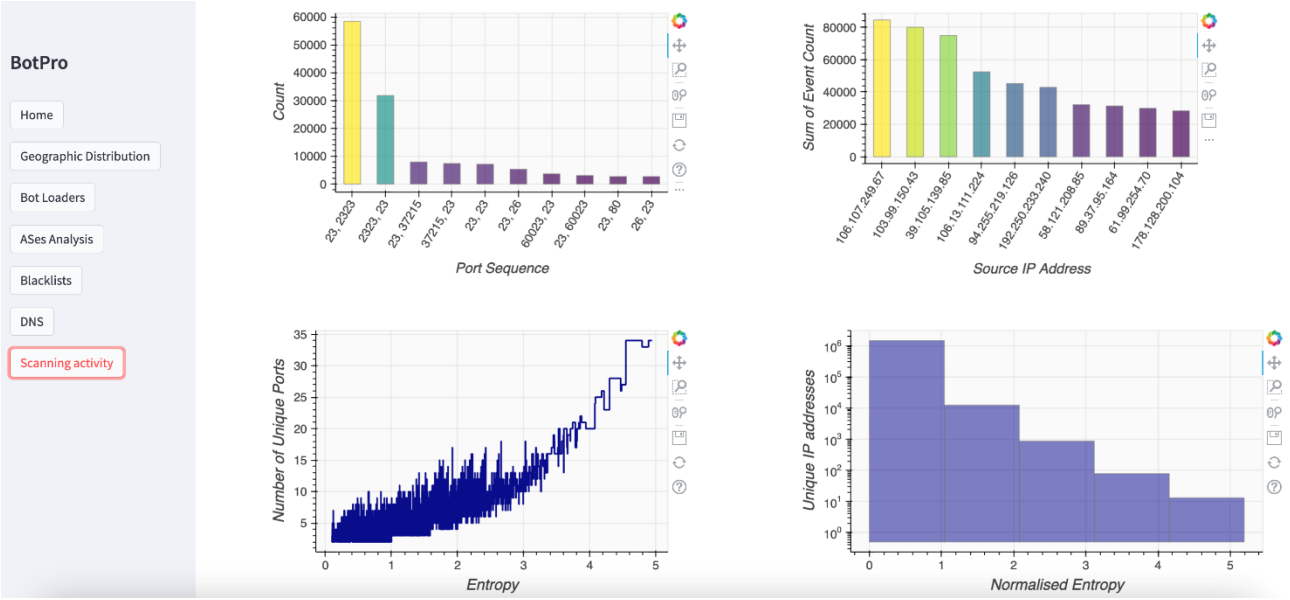

Figure 23: Common scanning patterns generated by IoT botnets and observed by BotPro[84]

4.6.4 Natural Language Processing

Natural Language Processing (NLP) in BotPro is utilized to analyze the vast amounts of text-based data associated with IoT botnets, such as logs, C&C communications, and other related textual information. NLP techniques enable BotPro to extract meaningful insights from this unstructured data, which is often crucial for understanding the intent and strategies of botnet operators. Tokenization breaks down large bodies of text into smaller, more manageable pieces, such as words or phrases, which are easier to analyze. This process is followed by stemming and lemmatization, which reduces words to their root forms, helping to standardize the text and facilitate comparisons across different documents. These steps are essential for creating a uniform dataset that can be analyzed consistently. Sentiment analysis is another powerful NLP technique employed by BotPro. By analyzing the tone and sentiment of C&C communications, BotPro can infer the intent behind certain commands, such as whether a botnet is preparing for a large-scale attack or simply performing routine maintenance. Understanding these nuances can provide valuable context for other analytical efforts and help prioritize responses to different types of threats. Overall, NLP enables BotPro to convert unstructured text into actionable intelligence, offering deep insights into the strategic and operational aspects of IoT botnet activities.

4.6.5 Information Theory

Information theory is a critical component of BotPro's algorithmic framework, providing tools to quantify and analyze the complexity and uncertainty within IoT botnet behaviors. Entropy is a key concept from information theory applied in BotPro to measure the randomness or unpredictability in the actions of botnets. High entropy levels suggest that a botnet is attempting to randomize its operations to avoid detection, making its behavior more challenging to predict and counter. By quantifying this unpredictability, BotPro can assess the sophistication of a botnet and adjust its detection strategies accordingly. Mutual information is another important metric used to determine the amount of shared information between different variables in the dataset. This measure helps BotPro identify which features are most informative when distinguishing between normal and malicious network traffic. For example, mutual information can reveal how closely certain network behaviors are correlated with botnet activities, allowing for more focused monitoring of those

[84] https://github.com/almazarqi/BotPro/blob/main/images/Scanning_activity.png

behaviors. Additionally, information theory helps in optimizing the data processing pipeline by identifying and prioritizing the most critical data points, ensuring that BotPro remains efficient even when dealing with large volumes of information. By applying these principles, BotPro enhances its ability to understand and predict the behavior of IoT botnets, making it a more effective tool in the ongoing effort to secure networks against these sophisticated threats. (Network topologies generated by BotPro are shown in Figure 24)

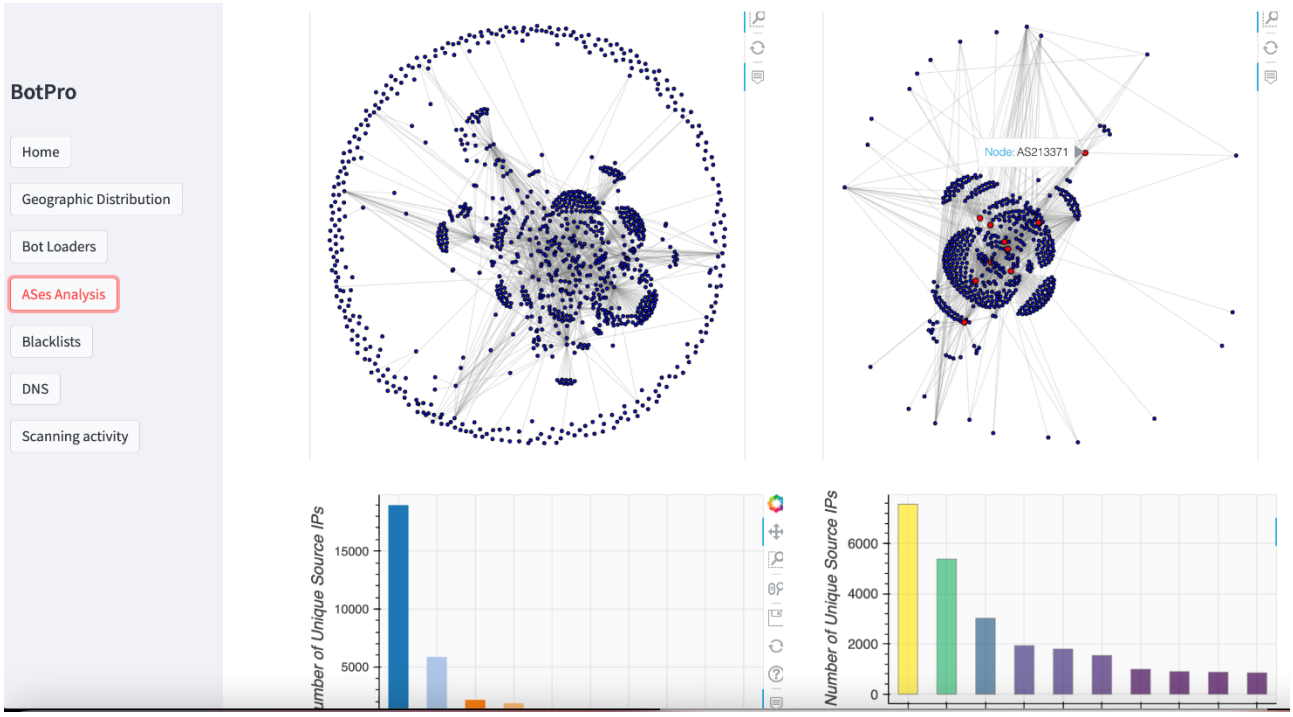

Figure 24: Network topologies for ASes generated by BotPro, suggesting that nodes identified by centrality metrics are more effective at spreading malicious content throughout the Internet[85]

[85] https://github.com/almazarqi/BotPro/blob/main/images/ASes%20analysis.png

5 Vulnerability Assessment and Risk Scoring

Vulnerability assessment and risk scoring are essential practices in cybersecurity, particularly for ICS systems in EPES. Specifically, the integration of vulnerability assessment and risk scoring is crucial for EPES due to the critical nature of these systems. Thus, any disruption (including those coming from cyber space) can lead to significant consequences, including power outages, economic losses, and potential safety risks. However, by regularly conducting vulnerability assessments along with proper risk scoring for each of the identified vulnerabilities, EPES stakeholders can proactively prepare measures which could mitigate or reduce the potential damage which these vulnerabilities might create to their daily business, and thus ensuring the reliability and security of their ICS systems.

Vulnerability assessment [18] refers to the systematic process to identify, quantify, and prioritize vulnerabilities in ICS systems of EPES. This process involves the identification of vulnerable assets by cataloging all components within the ICS of EPES under analysis (e.g., including SCADA, systems, PLCs, RTUs, etc.) and using specialized CTI tools to scan the network and devices for known vulnerabilities, such as outdates software, misconfigurations, or unpatched systems among others. Following the identification phase, tailored penetration testing, using specific threat models could be applied. For this purpose, cybersecurity professionals often use sandboxing cyber-physical testing environments are used for simulating specific cyber-attacks in relation with the identified vulnerabilities, aiming to further understand the systems' defenses and the potential entry points for the attackers. The next step in the vulnerability assessment process refers to a detailed analysis and reporting of identified vulnerabilities, their potential impact and possible recommendation for mitigation/remediation actions.

Risk scoring [19] refers to standardized methods used for quantifying the potential impact and likelihood of identified vulnerabilities being exploited. In the context of COCOON, the Common Vulnerability Scoring System (CVSS), a free and open industry standard commonly used by cybersecurity professionals for assessing the severity of IT/OT security vulnerabilities is adopted.

In the following we provide a comprehensive example on how the COCOON EWS along with external CTI and real-time analysis of network scans could be used for vulnerability assessment and risk scoring of ICS of EPES.

5.1 Cyber Threat Intelligence Feeds

Cyber Threat Intelligence (CTI) feeds are streams of data that provide information about potential or current cybersecurity threats to organizations or products. These feeds can include information about known malicious Internet Protocol (IP) addresses, Uniform Resource Locator (URLs), domains, file hashes[86], and other indicators of compromise (IOCs) [4]. They can also include information about threat actors, their TTPs, and the motivations behind their attacks. CTI feeds can be obtained from various sources, including commercial providers, open-source platforms (such as Malware Information Sharing Platform (MISP)[87], AlienVault Open Threat eXchange (OTX)[88]) among others), Information Sharing and Analysis Centers (ISACs) such as EE-ISAC[89], and government agencies.

In the context of EPES, CTI feeds can be particularly valuable due to the increasing interconnectivity of OT and IT networks. OT networks, which include ICS, are traditionally air-gapped but are now

[86] https://www.cisa.gov/sites/default/files/FactSheets/NCCIC%20ICS_Factsheet_File_Hashing_S508C.pdf
[87] https://www.misp-project.org/
[88] https://otx.alienvault.com/
[89] https://www.ee-isac.eu/

increasingly connected to IT networks for improved operational efficiency. This interconnectivity, however, also increases the attack surface and the risk of cyber-attacks.

In the context of COCOON, algorithmic tools such as the EWS, described in the previous chapter, is used to assess vulnerabilities in the ICS by integrating OT and IT network scans. The COCOON EWS tool scans the binaries of the software running on the analyzed ICS devices such that to identify any known vulnerabilities. To identify any known IOCs, the EWS uses the ICS scans (received from the EPES measurements via the COMML component of the CPN) and compare them against available CTI feeds. This integration of the COCOON EWS with CTI and OSINT search engines provides a more comprehensive view of the potential threats to the ICS.

The scans should include both passive and active scanning. Passive scanning involves monitoring the network traffic to identify the devices on the network and their characteristics. Active scanning involves sending probes to the devices to gather more detailed information. The scans should be conducted regularly to account for changes in the network and the threat landscape.

In the context of the COCOON project, CTI and OSINT trusted search engines for internet-connected devices such as Shodan and Censys are used to identify IPs or devices that might be exposed to potential cybersecurity vulnerabilities. It is to be highlighted that the data processing and analysis layer of the EWS is flexible enough to integrate a broad range of CTI and OSINT sources beyond the specific current stage of its implementation. Thus, while there are also other popular search engines for internet connected devices, such as ZoomEye[90], Fofa[91] or BinaryEdge[92], among many others, the rational for selecting Shodan and Censys search engines as out trusted CTI/OSINT sources for the early version of the vulnerability assessment and risk scoring of COCOON's is because they provide a greater range of measurements across the global IPv4.

As part of Chapter 2 the methodology followed in COCOON for vulnerability assessment and risk scoring was presented and also how CTI&OSINT streams along with real-time network scans will be used by the EWS for the actual evaluation of these vulnerabilities in terms of severity to the EPES operations and business continuation. For instance, Figure 4 showed the Dataflow for the COCOON's vulnerability assessment and risk scoring. Nonetheless, Figure 25 presents an early proof of concept implementation and showcases in greater level of detail the dataflow which used for extracting the relevant risk scoring associated with a specific EPES vulnerability associated to a particular asset with an IP address.

[90] https://www.zoomeye.hk/data-store
[91] https://en.fofa.info/
[92] https://www.binaryedge.io/

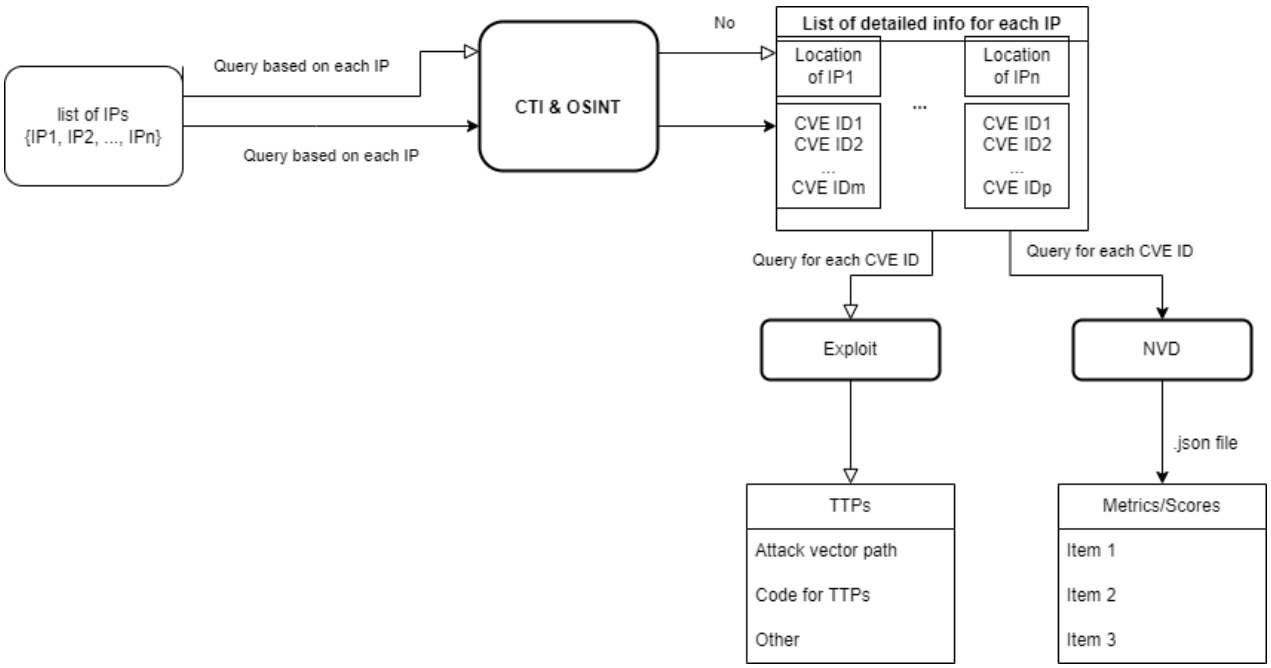

Figure 25: COCOON dataflow for risk scoring

As indicated, CTI and OSINT feeds are correlated with the list of IP addresses within a CPN-enabled EPES deployment and detailed vulnerability-related metadata is generated per asset. Metadata is stratified in terms of Common Vulnerabilities and Exposures (CVE) tags. For each CVE tag, a search on past and on-going associated exploits and their corresponding TTPs is conducted with the use of the widely and most prevalently used Exploit-DB database[93] . In parallel, a correlation with the National Vulnerability Database (NVD) is also initiated such as to obtain metrics and information attributing the risk score per CVE. Risk score composition is done through the Common Vulnerability Scoring System (CVSS) which acts as a standard for assessing the severity of known vulnerabilities. It provides a score between 0 and 10, with 10 being the most severe. The score is based on various metrics, including the exploitability of the vulnerability, its impact, and the scope of the impact.

Within the COCOON solution, CVSS scoring for each CVE tag is employed in vulnerabilities associated to devices and protocols identified via OT and IT network scans. Hence, helping to prioritize the mitigation efforts based on the severity of the vulnerabilities. The CVSS scores in COCOON are updated regularly to account for changes in the threat landscape and the availability of patches or workarounds for the vulnerabilities in an EPES of interest.

5.2 OT Network Scans and Security Assessment

OT network scans are an integral part of cybersecurity practices within ICS, including those related to EPES, and they are particularly useful to identify vulnerabilities and assess the security posture of such critical infrastructures. Within the COCOON solution, OT scans will be correlated against the CTI and OSINT feeds in order to tailor risk profiling based on the explicit properties of an EPES deployment of interest. However, the application of OT network scans for comprehensive risk assessment is notoriously challenging. Unlike IT environments, OT networks often involve a mix of legacy systems and proprietary protocols such as Modbus, which are not always compatible with standard IT scanning tools [8]. Moreover, OT environments require a high degree of reliability and uptime, making active scanning methods, which could disrupt operations, risky [9]. Literature on ICS

[93] Exploit Database, Exploit-DB

security highlights these difficulties; for instance, standard scanning techniques can fail to detect certain vulnerabilities or cause unintended system outages, which poses a significant barrier to effective risk management [10].

As already mentioned, we have developed a refined approach to address these challenges by integrating OT network scans into our risk assessment framework, while minimizing operational disruptions. Our methodology includes both passive and active scanning techniques tailored specifically for OT environments. Passive scanning allows us to monitor network traffic and identify devices without sending any probing signals, thus reducing the risk of interference. Active scanning, when used selectively and under requirements of active mitigation components (e.g., attack mitigation using Deep Reinforcement Learning), provides a deeper insight into network configurations and vulnerabilities. Additionally, by employing the CVSS within the COCOON scanning framework, we can normalize vulnerability data, facilitating a consistent and standardized risk scoring process. This dual approach not only improves our ability to identify vulnerabilities accurately but also aligns with established cybersecurity frameworks. The analysis of ICS protocols of interest based on their scanning properties broadens up the attribution of vulnerabilities and potential exploits according to their specific steady or transitioning states which is a core element within the overarching operation of the COCOON EWS. All pilots within the COCOON project use Modbus TCP and IEC104 in their OT deployments. We thus following provide a security assessment of these protocols via analysis of their session states via state transition diagrams.

5.2.1 State Transition Example for Modbus

As illustrated via Figure 26, the combined state transition diagram for the Modbus server integrated with the TCP protocol outlines the comprehensive process from connection establishment to request handling and error management. Initially, the server starts in the TCP Idle state, awaiting to initiate or accept a connection. When a connection request SYNchronize (SYN) is received, the server transitions to the TCP Listen state. Upon receiving a SYN-ACK and ACKnowledgement (ACK), the connection is established, moving the server to the TCP Established state. At this point, the Modbus protocol begins, and the server enters the Modbus Idle state, where it waits for a request from the client. Upon receiving a valid request, the server transitions to the Receive Request state, and then to the Process Request state where the request is processed. After processing the request, the server moves to the Send Response state to send the response back to the client. Once the response is sent, the server returns to the Modbus Idle state, ready to handle new requests. If a close request is initiated after sending a valid response, the server transitions to the TCP Close Wait state, then to the TCP Last Ack state upon receiving an ACK, and finally to the TCP Closed state, indicating the connection is closed.

If an invalid request or error occurs at any point during the communication process, the server transitions to the Error Handling state. In this state, the server addresses the issue by handling the invalid request or error condition. Once the error is resolved, the server transitions back to the Modbus Idle state, ready to receive new requests. This systematic approach ensures robust error management and reliable communication between the Modbus server and client in the context of the TCP protocol.

In case of an attack, several states in the combined state transition diagram could be affected, leading to potential security breaches and operational disruptions. For example, during a Man-In-The-Middle attack, an attacker intercepts and potentially alters communication between the client and server [20]. For example, in the Receive Request state, the attacker might modify the requests sent by the client, leading the server to process incorrect or malicious commands. This could cause the Process Request state to handle invalid data, resulting in erroneous operations or system malfunctions. Additionally, in the Send Response state, the attacker might intercept and alter the server's responses, leading the

client to receive false information, which can disrupt the overall system's integrity and reliability. The Error Handling state might also be triggered more frequently due to the tampered data causing processing errors. Furthermore, in the TCP Established state, the attacker could disrupt the connection stability, forcing premature transitions to the TCP Close Wait or TCP Last Ack states, leading to unexpected connection closures and communication failures.

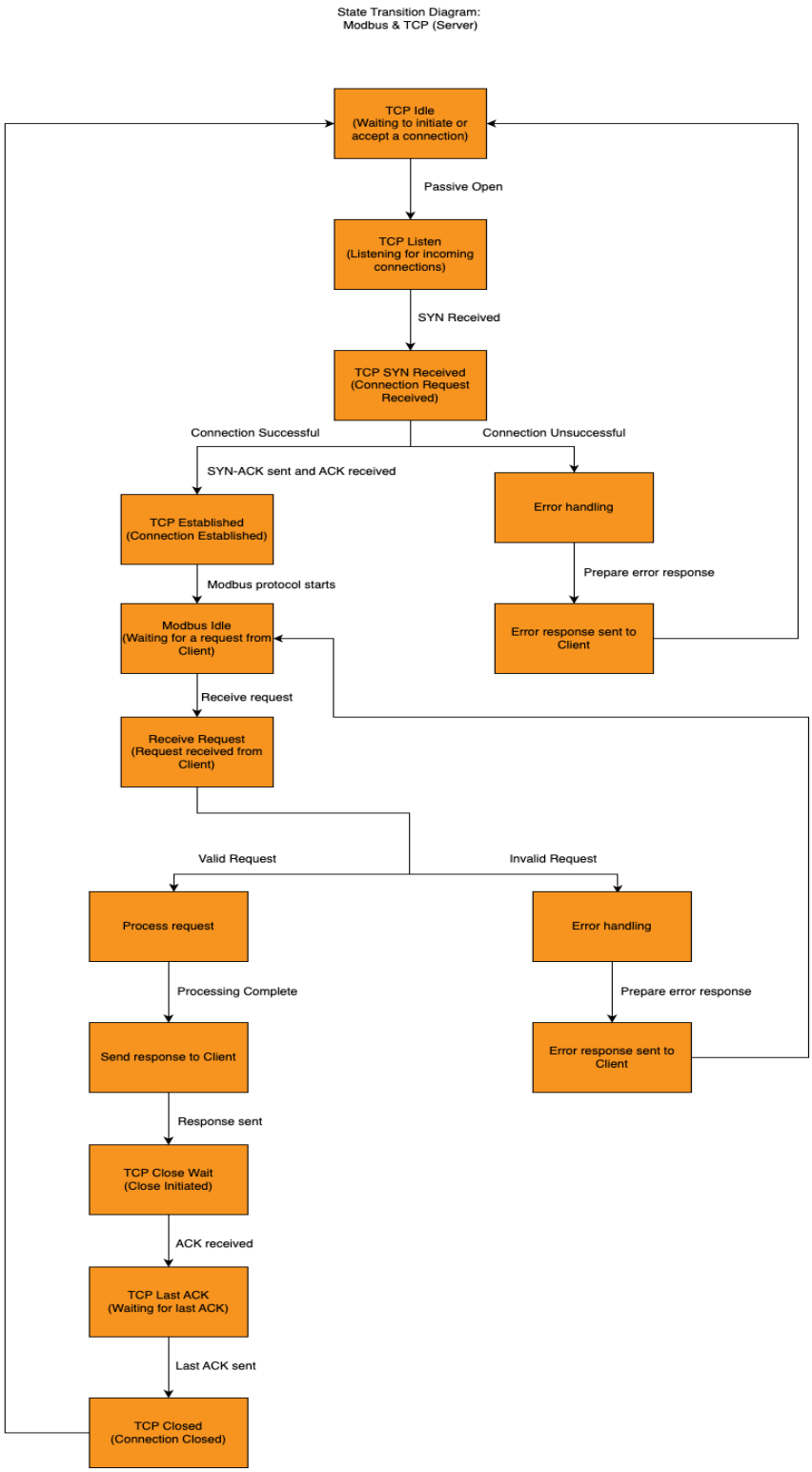

Figure 26: Transition diagram for Modbus TCP Server

In Figure 27, the state transition diagram for the Modbus client integrated with the TCP protocol outlines the detailed process from connection initiation to request handling, response processing, and error management. Initially, the client starts in the TCP Idle state, ready to initiate a connection. Upon sending a SYN request, the client transitions to the TCP Syn Sent state, waiting for a SYN-ACK from the server. Once the SYN-ACK is received, the client moves to the TCP Syn Receive state and sends an ACK, establishing the connection and transitioning to the TCP Established state. At this point, the Modbus application protocol is initiated, and the client enters the Modbus Idle state, where it is ready to send requests to the server. When the client sends a request, it transitions to the Send Request state, and after the request is sent, it moves to the Wait for Response state, waiting for a response from the server.

Upon receiving a response, the client transitions to the Process Response state to validate and process the received data. If an error occurs while sending the request or waiting for a response, the client transitions to the Error Handling state to address the issue. After handling the error, the client returns to the Modbus Idle state. When the client decides to terminate the connection, it sends a FIN and moves to the TCP Fin Wait 1 state, waiting for an ACK from the server. Upon receiving the ACK, the client transitions to the TCP Fin Wait 2 state, waiting for a FIN from the server. After receiving the FIN, the client sends an ACK and moves to the TCP Time Wait state, waiting for a timeout to ensure all packets have been properly transmitted before transitioning to the TCP Closed state, indicating the connection is closed.

In the event of a MiTM attack or other types of cyber-attacks on the Modbus protocol, several states in the combined state transition diagram for the client could be impacted, leading to security breaches and operational disruptions [20]. During a MiTM attack, an attacker might intercept and alter the client's requests or the server's responses. For example, in the Send Request state, the attacker could modify the requests being sent to the server, causing the Wait for Response state to receive malicious or incorrect data. This would subsequently affect the Process Response state, where the client might process falsified information, leading to erroneous decisions or actions. Additionally, if the attack involves disrupting the communication, it could cause frequent transitions to the Error Handling state due to unexpected errors or timeouts. The TCP Established state might also be compromised, leading to premature or unauthorized transitions to the TCP Fin Wait 1 or TCP Closed states, disrupting the connection and causing potential data loss or communication failures.

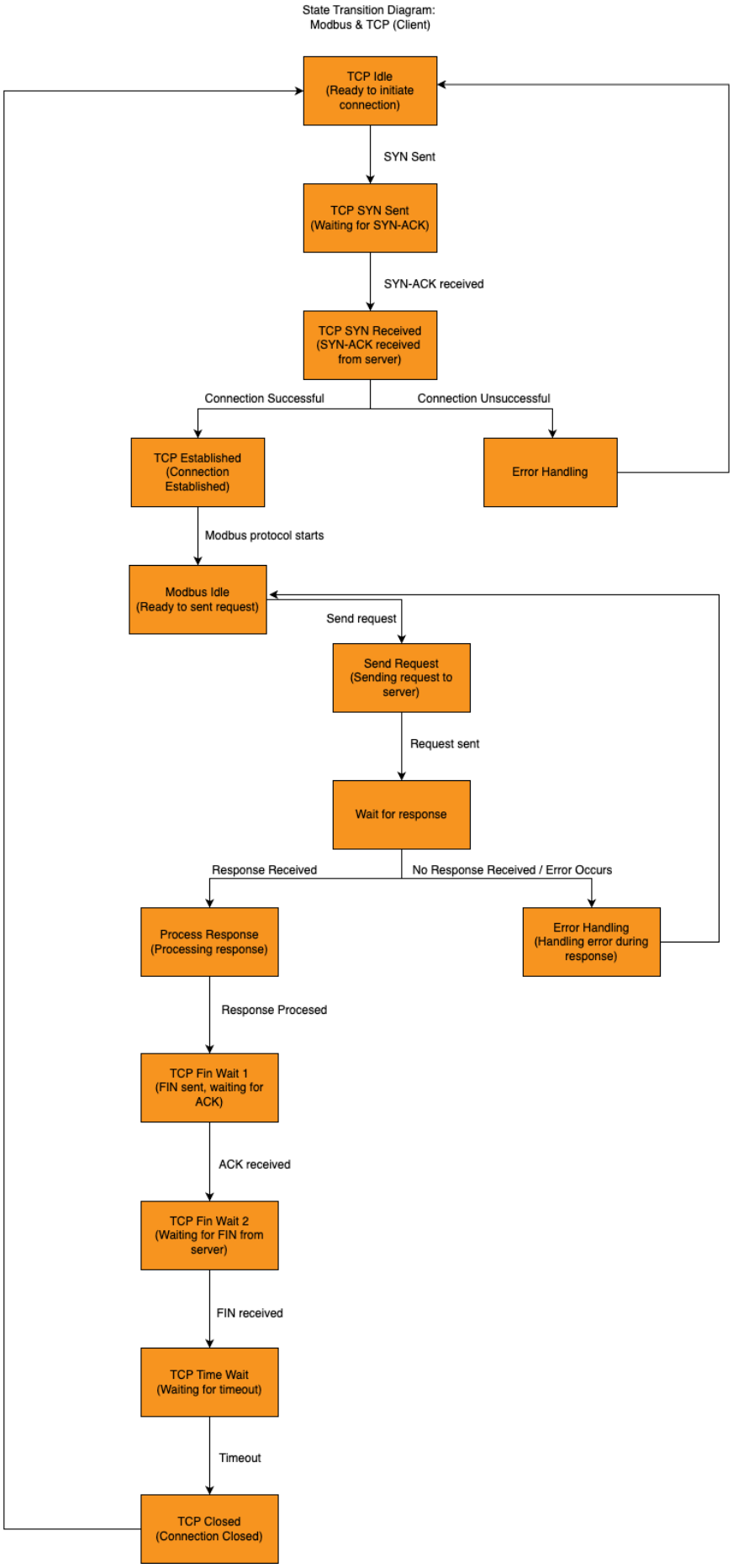

Figure 27: State transition diagram for Modbus TCP Client

5.2.2 State Transition for IEC 104

The IEC104 client state transition diagram in Figure 28 illustrates the sequence of states the client undergoes while establishing a connection and exchanging data with a server. Initially, the client is in the Idle state, ready to initiate a connection. Upon sending a connection request, the client transitions to the Connection Request Sent state, where it awaits confirmation from the server. Once the connection is established, the client moves to the Connected state. From here, the client can transition to the Send ASDU state to send Application Service Data Units (ASDUs) to the server. After sending the data, the client waits in the Wait for Acknowledgment state for a response from the server. If an ACK is received, the client returns to the Connected state, ready to send further data or disconnect and return to the Idle state when the session is complete.

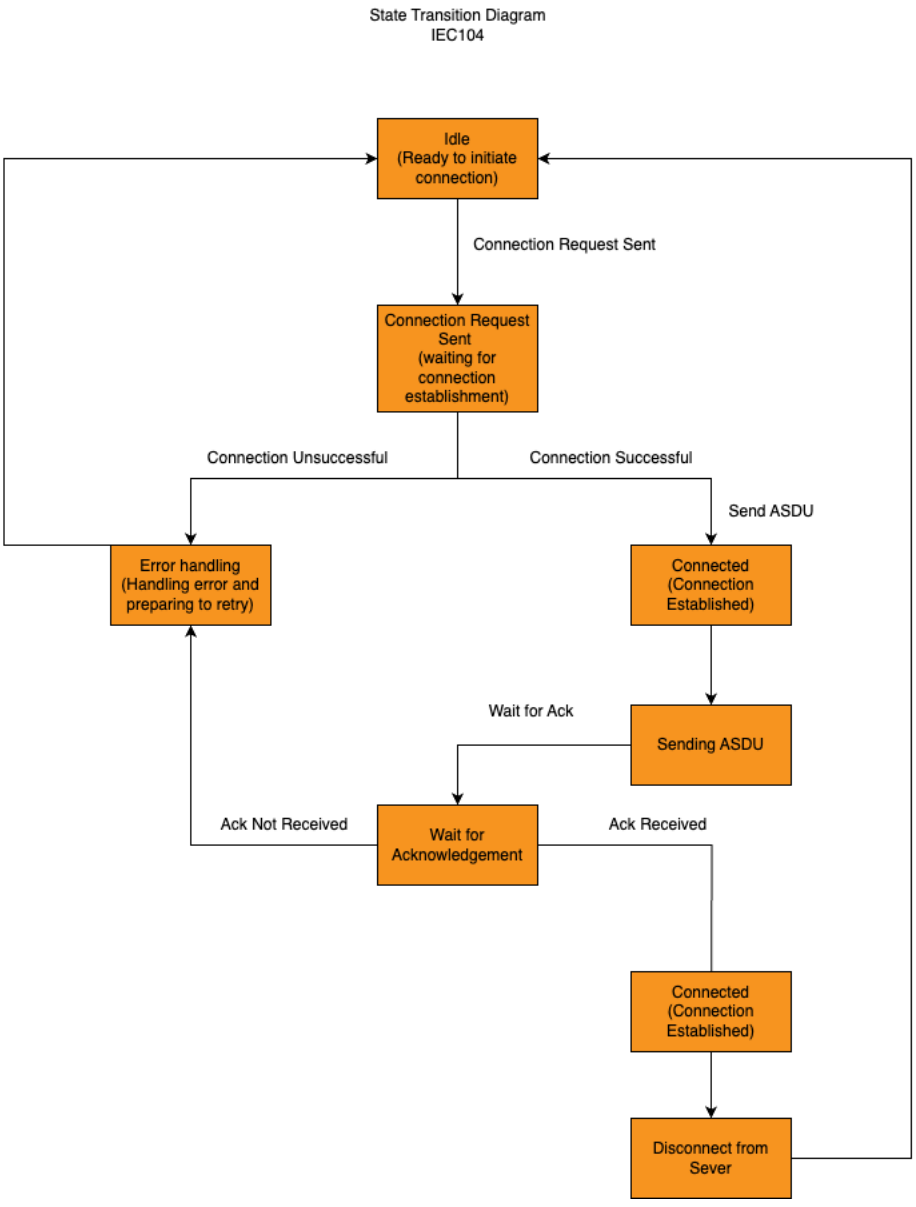

Figure 28: State transition diagram for IEC 104

If an error occurs at any point during the communication process, such as receiving an invalid response or encountering a timeout, the client transitions to the Error Handling state. This state is responsible for addressing the error, which might involve retrying the request, logging the error, or performing other corrective actions. Once the error has been handled, the client transitions back to the Idle state, prepared to initiate a new connection or resume normal operation. This systematic

approach ensures robust error management and reliable communication between the IEC104 client and server.

During an attack, certain states are particularly susceptible to exploitation by attackers, notably during the connection establishment and data transfer phases. One of the most vulnerable states is the Connection Request Sent state [21], where attackers can intercept and manipulate connection requests, potentially leading to unauthorized access or denial of service attacks. This state is critical as it lays the foundation for subsequent communication; thus, compromising it can lead to a breakdown in the secure transmission of data.

Another critical state frequently targeted by attackers is the Wait for Acknowledgment state [22]. During this phase, the client waits for an ACK from the server after sending an ASDU (Application Service Data Unit). Attackers can exploit this state by injecting false ACKs or delaying legitimate ones, causing the client to either process incorrect data or timeout, leading to unnecessary retransmissions and potential service disruption. Additionally, the Send ASDU state is vulnerable to command injection attacks, where malicious commands are sent to the server, potentially causing unauthorized actions within the control system.

5.3 Graph-Based Dependency Mapping

Graph-based dependency mapping plays a crucial role in enhancing vulnerability assessment and risk scoring by providing a clear understanding of the interdependencies within IT/OT environments. Specifically, graph-based dependency mapping involves creating a graph where nodes represent assets (e.g., devices, services, applications) and edges represent dependencies or relationships between these assets. This approach provides a visual and analytical representation of the system's architecture and interdependencies. By integrating this approach with established frameworks like NIST, CVSS, and FAIR, industrial stakeholders using the COCOON solution can develop a comprehensive and robust methodology for protecting their systems in a proactive manner. Hence, the COCOON EWS graph-based dependency mapping functionality ensures that vulnerabilities are prioritized based on their potential impact on the overall system, enabling effective remediation and mitigation strategies. Via leveraging data from diverse CTI &OSINT search engines (e.g., Shodan, Censys, ZoomeEye, BotPro) and correlating them with OT scans as described earlier, a deployment-specific graph-based dependency mapping is achieved within the Vulnerability Assessment and Risk scoring framework realised within the COCOON EWS. Therefore, allowing the enhancement of the security posture of EPES Stakeholders while contributing to the overall resilience via risk preparedness for their explicit operational deployments.

For the sake of simplicity, Figure 29 provides a data flow diagram showing in practice the instantiation of the COCOON Vulnerability Assessment and Risk Scoring Framework with the exemplar use of Shodan feeds. To be noted, that OSINT is also considered as the output of the BotPro framework discussed earlier as well as other engines such as Censys, ZoomEye and Greynoise which are also considered within the COCOON implementation of the EWS. Hence, this diagram is an overview of the procedure for evaluating and archiving IP addresses in relation with their Shodan feeds. As shown, the process is initialized with an input list of IP addresses related to the EPES setup. During the initialization phase OSINT techniques are used to initialize application programming interface (API) keys for Shodan, NVD along with the creation of folders for each IP. Subsequently, Shodan is used to fetch host data where geolocation information and a list of CVE identifiers (IDs)

are retrieved for each IP address. CVE data is then enriched by gathering detailed information from the NVD, including CVSS metrics, and by retrieving exploit data from exploitDB[94].

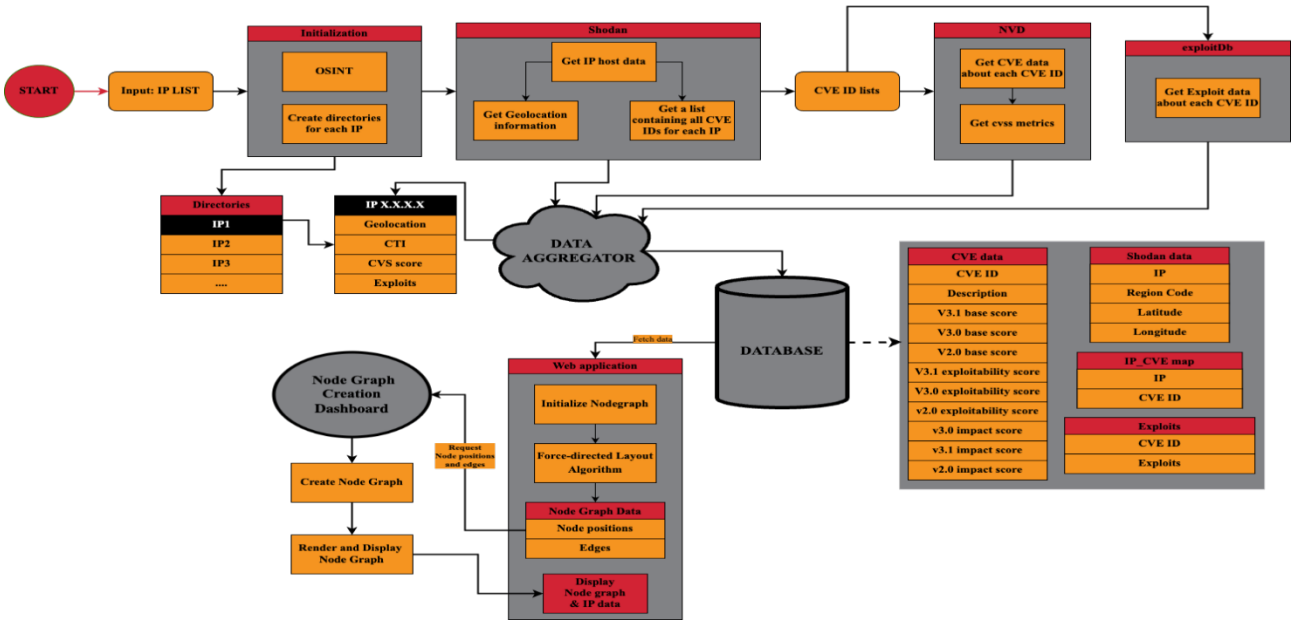

Figure 29: Detailed implementation version of the data flow diagram for risk scoring and Node Graph

Following data collection, the information is arranged before being kept in the directories made specifically for each IP. After that, the information is aggregated into a database with an organized schema. The tables that comprise the database are the following:

- 'Shodan data' for storing IP-related / host information such as region code and geolocation details,
- 'CVE data' for storing details about each CVE including CVSS scores,
- 'IP_CVE map' for mapping IPs to their respective CVE IDs, and
- 'Exploits' for storing exploit data related to specific CVE IDs.

Finally, a node graph is generated to visually represent the vulnerabilities and exploits of the analyzed IP addresses, providing a clear view of potential threats. This graph is rendered and displayed using a node graph dashboard, allowing for dynamic interaction with the threat data.

5.3.1 Graph Construction

Graph Construction refers to the process of building a dependency graph where nodes represent assets and edges represent dependencies. This graph can be enriched with attributes such as asset criticality, vulnerability severity, and connectivity.

Several crucial phases are involved in the development of the Node Graph, such as:

1) Data Retrieval

The first step is to retrieve all the necessary data that the node graph will show. The data consists of a list of IP addresses, a mapping of these IP addresses to CVEs, Descriptions of the CVEs, CVSS scores for each CVE and information about exploits.

2) Graph Initialization

[94] https://www.exploit-db.com/

After data collection, the graph needs to be initialized. The graph is initialized by using NetworkX[95] library for Python, which is a useful tool when creating, manipulating, and studying the structure, dynamics and functions of complex networks.

The initialized graph is directed, i.e., the connections (edges) among nodes start from a specific node and point directly to another node in the graph. So, the relationships between these nodes are shown with directed edges.

There are three types of nodes: (i) IP node, (ii) CVE node & (iii) Exploit node.

To show that a CVE is linked to an IP address in the system, there is an edge starting from an IP node pointing to a CVE node. An edge connecting a CVE node to an exploit node suggests that the exploit may be able to leverage that vulnerability.

3) Node Positioning

When the initialization process ends, nodes need to be placed in certain positions to create a meaningful depiction of them along with their relationships. So, nodes positions in this stage are established with the Spring Layout algorithm which results in an attractive display, making sure that the user is provided with an understandable evaluation of the system. We adopt the Spring Layout Fruchterman-Reingold algorithm implementation pseudocode as in [12] and illustrated in Figure 30.

[95] https://networkx.org/documentation/stable/reference/index.html

Spring Layout Algorithm

area = $W \times L$ ▷ W and L are the width and length of the frame
G = (V, E) ▷ Vertices are assigned random initial positions
k = $\sqrt{\text{area}/|V|}$

function FA(x)
    return x^2/k ▷ Calculate the attractive force based on x
end function

function FR(x)
    return k^2/x ▷ Calculate the repulsive force based on x
end function

if pos = None **then** ▷ Initialize positions if not provided
    pos = random
end if

if fixed ≠ None **then** ▷ Ensure positions are set for fixed nodes
    for each node in fixed **do**
        if node ∉ pos **then**
            fixed_positions.append(pos[node])
        end if
    end for
end if

domain_size = the largest coordinate value among all nodes' positions ▷ Determine the size of the existing layout area

if domain_size = 0 **then** ▷ Set to one if domain size is zero to avoid scaling issues
    domain_size = 1
end if

pos = random positions for all nodes, scaled by domain size ▷ Generate random positions scaled by domain size and centered

for $i = 1$ **to** iterations **do**
    for each v in V **do** ▷ Calculate repulsive forces
        v.disp = 0
        for each u in V **do**
            if $u \neq v$ **then**
                $\delta = v.pos - u.pos$
                distance = $|\delta|$
                v.disp = v.disp + $(\delta/\text{distance}) \times$ fr(distance)
            end if
        end for
    end for

    for each e in E **do** ▷ Calculate attractive forces
        $\delta = e.v.pos - e.u.pos$
        distance = $|\delta|$
        e.v.disp = e.v.disp − $(\delta/\text{distance}) \times$ fa(distance)
        e.u.disp = e.u.disp + $(\delta/\text{distance}) \times$ fa(distance)
    end for

    for each v in V **do** ▷ Limit max displacement to temperature t and prevent displacement outside the frame
        disp_length = $|v.disp|$
        if disp_length > 0 **then**
            v.pos = v.pos + $(v.disp/disp\_length) \times \min(disp\_length, t)$
            v.pos.x = $\min(W/2, \max(-W/2, v.pos.x))$
            v.pos.y = $\min(L/2, \max(-L/2, v.pos.y))$
        end if
    end for

    t = cool(t) ▷ Reduce the temperature as the layout approaches a better configuration
end for

if fixed = None **and** scale ≠ None **then** ▷ If needed, rescale layout to fit within the specified area
    pos = rescale_layout(pos, scale)
end if

return pos

Figure 30: Spring Layout (Fruchterman-Reingold) algorithm implementation pseudocode

As shown, the Spring Layout algorithm—also referred to as the *Fruchterman-Reingold algorithm* [12]— is a force-directed layout algorithm that is used to place the nodes in a structured form. It simulates a force-directed representation of the network treating edges as springs holding nodes close, while treating the nodes as repelling objects. The algorithm starts by first arranging nodes at random points in a second space, then continuously modifies their placements in response to the attraction and repulsion between connected nodes. Nodes with edges connecting them are dragged closer together, while those without edges are pushed apart, based on the rules of two forces:

- *Attractive Forces* (Hooke's Law): Similar to the tension in a spring, an attractive force acts between nodes connected by an edge.
- *Repulsive Forces* (Coulomb's Law): Similar to the attraction between like-charged particles, each node in the network creates a repulsive force on every other node.

To minimize the overall energy of the system, the algorithm seeks to find a state where the forces between nodes and edges are balanced. The system's energy is a measurement of how far the nodes are now arranged from this optimal, balanced state. This process repeats until it reaches the equilibrium state. Equilibrium is reached when the positions of the nodes stop changing significantly because the forces acting on them are balanced. Over time, the adjustments to the node positions get smaller and smaller, until eventually, the nodes settle into a stable layout where they no longer change positions. The representation of the graph is produced using the nodes' positions in this equilibrium. The result is a visually appealing graph layout that enhances understanding of the relationships and interconnections between the nodes for the viewer.

4) Enhancing Visual Clarity

Several parameters are added for the *Spring Layout algorithm* which significantly change the result. The first parameter defines the spacing between nodes. The higher the value, the more dispersed the node will be, which helps in reducing clutter. The second parameter is in charge for adjusting (modifying) the graph's scale. This parameter is based on the number of CVE nodes existing in the graph. Thus, more CVE nodes result in the graph's scale to be bigger and all nodes to be placed in such a manner that they can all be properly seen. The third and final parameter is the number of iterations the algorithm does to place the nodes. More iterations will result in a more stable and refined layout, but it will also take a longer time to compute.

After positioning the nodes and having a final view of the nodes' graph using the *Spring Layout algorithm*, the direct positioning and appearance for the three different types of nodes (IP, CVE & Exploits) is adjusted to enhance visual clarity. These properties make sure that the differences between the nodes are clear and it also ensures a better context about what data is represented by each node. Specifically, IP Nodes are positioned centrally, indicating their role as the central point of the network, with connections spreading outward to vulnerabilities and exploits. The CVE Nodes show the CVSS score, and they are arranged around the IP nodes. The way they are colored conveys the risk severity level for a given CVE and also its relationship with a given exploit that has been directly related to.

Node Color	CVSS Score	Qualitative Rating
Red	9.0 – 10.0	Critical
Orange	7.0 – 8.9	High
Yellow	4.0 – 6.9	Medium
Green	0.0 – 3.9	Low

Table 1 Color-Score-Rating Representation of Nodes

Table 1 indicates the color-coded system with respect to node graph visualization. The colors are based on their CVSS scores. Nodes are colored to represent the different levels of risk: Red indicates critical vulnerabilities with CVSS scores between 9.0 and 10.0, Orange represents high severity (7.0 to 8.9), Yellow shows medium severity (4.0 to 6.9), and Green signifies low severity (0.0 to 3.9)

5) Node Graph Presentation

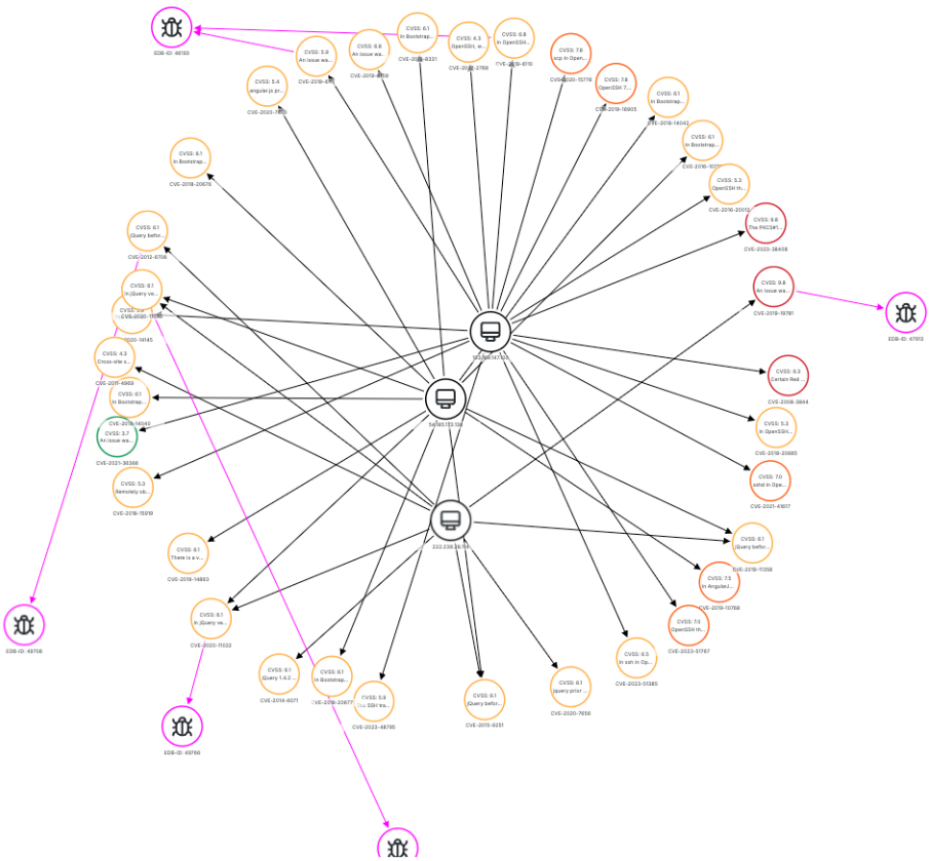

Figure 31: Example of a COCOON' s Node Graph visualization within the EWS

The final step is to present the graph, and the data needed for generating it, including all node positionings, attributes and customizations. The EWS compiles the graph into a response format (JSON) and entails a Grafana instance[96](a powerful open-source platform specifically designed for monitoring and visualizing data from various sources) used to render and visualize the node graph based on the response file created for both nodes and edges. For additional inspection as well as for ensuring interoperability amongst diverse configurations, the response along with the graph can be exported to other systems or applications that could be customized within a given EPES stakeholder. This procedure ensures that the complex relationships between IPs, and vulnerabilities along with their corresponding CVSS scores and exploits are efficiently shown and well abstracted. Thus, enhancing comprehension, and improving the examination process of the network's architecture to an EPES operator.

An example of the Node Graph generated, depicting the relationships between various IP addresses, their associated vulnerabilities (CVEs), and potential exploits can be seen in Figure 31. The central

[96] https://grafana.com/grafana/

nodes represent IP addresses, which are connected to various CVEs, based on the vulnerabilities they possess. Additionally, the magenta edges leading to bug-like icons represent the known exploits associated with specific CVEs.

5.3.2 Propagation Analysis in the EWS Graph Dependency Mapping

Propagation analysis refers to the process of interpreting how vulnerabilities can propagate through the system by examining the dependencies. This involves identifying critical paths and potential cascading effects. Figure 32 presents an example of a zoomed-in Graph-Node for a given EPES as generated with the COCOON's Graph Dependency Mapping tool within the EWS. As shown, the IP address 54.185.173.138 visible on the top right of Figure 32, depicted as a node with title "54.185.173.138" in the Node Graph shows an asset in a network that relates to multiple vulnerabilities (CVEs) and open to corresponding exploits. This IP poses a significant security risk because of its several vulnerabilities, including CVE-2020-11022, which has a score of 6.1 and could easily expose the system to potential attacks. The associated exploit for this vulnerability is indicated by the magenta-colored edge pointing to the exploit node with the bug icon (bottom left of Figure 32), which is identified by EDB-ID: 49766. This exploit is a reference to a jQuery version 1.2 Cross-Site Scripting (XSS) vulnerability that lets attackers insert malicious scripts into webpages, possibly leading to data theft or unauthorized access.

Additionally, another IP address in the network, 222.239.28.114 (middle right in Figure 32), also shares this vulnerability (CVE-2020-11022). This can clearly be seen as an edge connecting this specific IP node to the same CVE node as the previous IP did. This indicates that multiple assets within the network are at risk from the same AV.

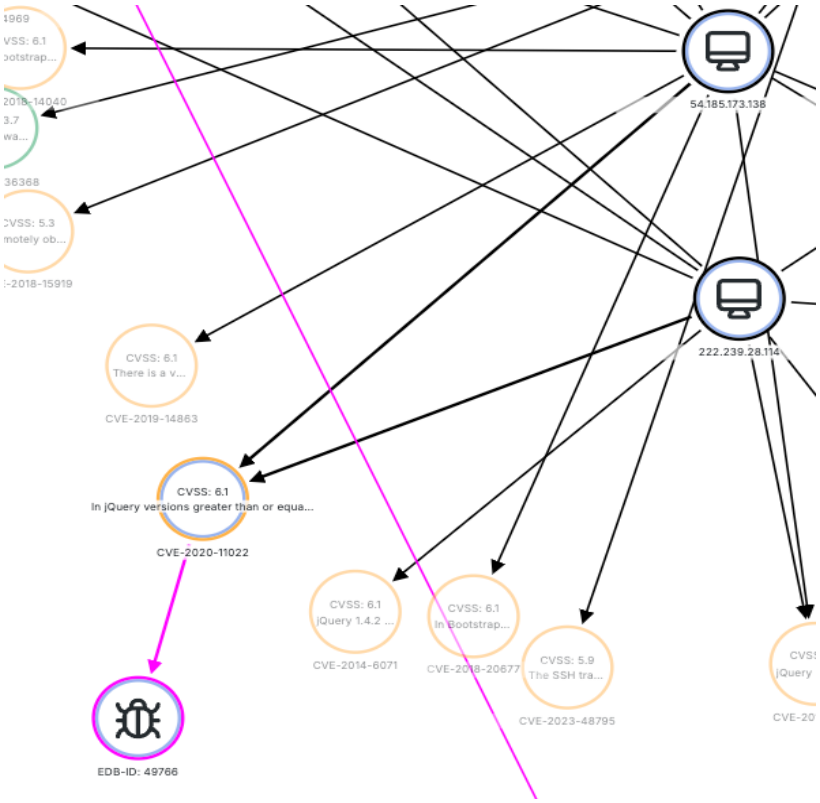

Figure 32: Zoom into a section of the Graph Node example within the COCOON EWS.

6 Conclusions

The digital transformation of EPES via the growth and the convergence of traditionally isolated ICS with Internet-enabled protocols and services expose these infrastructures to cyber threat vulnerabilities and severe exploits. This report composed the COCOON framework to address the need for robust threat modelling, vulnerability assessment and risk scoring fully aligned with the industry standards along with technology and research advancements on data analytics and algorithms from ML, NLP graph theory to safeguard these critical infrastructures. Specifically, Chapter 3 of this report detailed the COCOON framework for threat models as an adaptation and evolution of the MITRE ATT&CK Framework, tailored specifically for ICS within EPES and to be heavily utilised within the envisaged COCOON pilot demonstrators. The COCOON threat modelling framework encompassed methodologies and practical implementations for several APT threat modelling while strengthening the importance and rationale behind managing sophisticated cyber threats which might exploit vulnerabilities in the ICS of EPES. Key components of ICS threat models were also elaborated, while the applicability of the COCOON threat modelling framework was demonstrated through proof-of-concept in real-world scenarios linked to the specificities of the COCOON pilots.

This deliverable also delved into the architecture of the COCOON EWS which is a key expected outcome of COCOON designed to bolster the cybersecurity posture of ICS in EPES. The modular, flexible and adaptive architecture of the EWS allows for further extension and integration of technical advancements which could be easily mapped to its layers: (i) the data collection layer, (ii) the data processing and analysis layer, (iii) the decision-making layer, (iv) the communication and response layer, and, (v) a continuous monitoring loop. Further, this report provided several practical implementations which offer a credible proof-of-concept for the applicability of the EWS in real-life scenarios in-line with the COCOON demonstrators. Thus, first, the detection of a DDoS attack, from initial sensor detection to the execution of mitigation strategies was elaborated. Second, the early detection of a botnet infection, starting from the identification of an IoT device infected by the botnet to containment measures to prevent botnet spreading, was also presented as a detailed practical example. Third, identifying insider threats, beginning with the analysis of suspicious behavior of employee accounts and culminating in measures to protect sensitive data from unauthorized access was detailed. These three examples aimed to offer a broad perspective of the role of EWS within the vulnerability assessment process and its generic use across typical EPES deployments.

It is worth highlighting that the COCOON EWS integrates BotPro for tracking large-scale vulnerabilities and exploits and processes CTI and OSINT feeds, along with OT network scans in order to extract meaningful information for EPES operators. The results analysis is offered using advanced visualization tools which make use of state transition diagrams and graph-based dependency maps. Further Chapter 5 of this report provided concrete examples on how the COCOON EWS and its following process for vulnerability assessment and risk scoring can be applied. Thus, practical implementation examples for two commonly used communication protocols for ICS of EPES, such as Modbus and IEC104 were demonstrated. In conclusion, this holistic approach ensures that EPES operators along with communication and cybersecurity engineers serving the EPES have a clear and actionable understanding of the cybersecurity posture of their ICS, enabling them to take proactive measures to mitigate risks.

In general, this report may adequately be used as a valuable resource for cybersecurity professionals, offering a comprehensive framework and practical tools for managing and mitigating cyber threats in ICS of EPES via the COCOON solution. The detailed descriptions, practical implementations, and real-world examples emphasize the applicability of the proposed practical mechanisms.

References

[1] S. M. Khalil, H. Bahsi, T. Korõtko, “Threat modeling of industrial control systems: A systematic literature review”, *Computers & Security*, Volume 136, no. 103543, Jan. 2024.

[2] Xiong, W., Legrand, E., Åberg, O. et al. “Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix”, Software System Model, vol. 21, pp. 157–177, Feb.2022.

[3] B. Al-Sada, A. Sadighian and G. Oligeri, "Analysis and Characterization of Cyber Threats Leveraging the MITRE ATT&CK Database," in *IEEE Access*, vol. 12, pp. 1217-1234, 2024

[4] M. Asiri, N. Saxena, R. Gjomemo, and P. Burnap, “Understanding Indicators of Compromise against Cyber-attacks in Industrial Control Systems: A Security Perspective”, *ACM Transaction on Cyber-Physical Systems,* vol. 7, issue 2, Article 15, pp. 1-33, April 2023.

[5] Petr Matoušek, “Security analysis of the IEC 60870-5-104 protocol,” Technical Report, Brno University of Technology, 2020. Available: hTTP’s://www.fit.vut.cz/research/publication-file/11570/TR-IEC104.pdf

[6] Introduction to Modbus TCP/IP: ProSoft Technology, “Introduction to Modbus TCP/IP,” [Online]. Available: https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf. [Accessed: 07-Aug-2024].

[7] Modbus over Serial Line: Modbus Organization, “Modbus over Serial Line,” Version 1.02, December 2006. [Online]. Available: https://modbus.org/docs/Modbus_over_serial_line_V1.pdf. [Accessed: 07-Aug-2024].

[8] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, B. Pranggono, and H. F. Wang, “A survey on industrial control system testbeds for security research,” IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1649–1675, 2019.

[9] T. Morris and W. Gao, Industrial Control System Cyber Attacks. Cham, Switzerland: Springer, 2017.

[10] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on SCADA systems,” in Proc. IEEE Int. Conf. Internet of Things (iThings/CPSCom), Dalian, China, 2011, pp. 380-388.

[11] V. Uher, P. Gajdo and V. Snáel, "The Visualization of Large Graphs Accelerated by the Parallel Nearest Neighbors Algorithm," *2016 IEEE Second International Conference on Multimedia Big Data* (BigMM), pp. 9-16, Taipei, Taiwan, 20-22 April 2016.

[12] T. M. J. Fruchterman and E. M. Reingold, “Graph Drawing by Force-Directed Placement,” *Software Practice and Experience*., vol. 21, no. 11, pp. 1129–1164, 1991.

[13] Stephen G. Kobourov, “Force-Directed Drawing Algorithms,” Handbook of Graph Drawing and Visualization, pp.383-408, 2013

[14] H. A. Almazarqi, M. Woodyard and A. K. Marnerides, "Macroscopic Insights of IoT Botnet Dynamics Via AS-level Tolerance Assessment," ICC 2024 - IEEE International Conference on Communications, Denver, CO, USA, 2024, pp. 5244-5249, doi: 10.1109/ICC51166.2024.10622782.

[15] H. A. Almazarqi, M. Woodyard, T. Mursch, D. Pezaros and A. K. Marnerides, "Tracking IoT P2P Botnet Loaders in the Wild," ICC 2023 - IEEE International Conference on Communications, Rome, Italy, 2023, pp. 5916-5921, doi: 10.1109/ICC45041.2023.10279593

[16] Hatem A Almazarqi, Angelos K Marnerides, Troy Mursch, Mathew Woodyard and Dimitrios Pezaros, "Profiling iot botnet activity in the wild", 2021 IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2021.

[17] Angelos K. Marnerides, Vasileios Giotsas, and Troy Mursch. 2019. Identifying infected energy systems in the wild. In Proceedings of the Tenth ACM International Conference on Future Energy Systems (e-Energy '19). Association for Computing Machinery, New York, NY, USA, 263–267. https://doi.org/10.1145/3307772.3328305

[18] R. C. Parks and E. Rogers, "Vulnerability Assessment for Critical Infrastructure Control Systems," in IEEE Security & Privacy, vol. 6, no. 6, pp. 37-43, Nov.-Dec. 2008.

[19] W. Wang, F. Shi, M. Zhang, C. Xu and J. Zheng, "A Vulnerability Risk Assessment Method Based on Heterogeneous Information Network," in *IEEE Access*, vol. 8, pp. 148315-148330, Aug. 2020.

[20] I. Siniosoglou, P. Radoglou-Grammatikis, et.al., "A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1137-1151, June 2021.

[21] P. Radoglou-Grammatikis, P. Sarigiannidis, et. al., "Attacking IEC-60870-5-104 SCADA Systems," *2019 IEEE World Congress on Services*, Milan, Italy, pp. 41-46, 8-13 July 2019.

[22] Erdodi, L.; Kaliyar, P.; Houmb, et. al., “Attacking Power Grid Substations: An Experiment Demonstrating How to Attack the SCADA Protocol IEC 60870-5-104”, In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES 2022), Vienna, Austria, 23–26 August 2022.