

Universidade do Minho
Escola de Engenharia

Paulo Jorge Sousa Costa

Host-based security metrics in industrial environments

January 2025

Host-based security metrics in industrial environments

Paulo Jorge Sousa Costa

UMinho | 2025



Universidade do Minho
Escola de Engenharia

Paulo Jorge Sousa Costa

Host-based security metrics in Industrial environments

Master's Work Dissertation Report
Integrated Master's in Engineering and Management of Information Systems

Work performed under the supervision of
Professor Henrique Manuel Dinis dos Santos

January 2025

Copyright and Terms of Use for Third Party Work

This dissertation reports on academic work that can be used by third parties as long as the internationally accepted standards and good practices are respected concerning copyright and related rights.

This work can thereafter be used under the terms established in the license below.

Readers needing authorization conditions not provided for in the indicated licensing should contact the author through the RepositóriUM of the University of Minho.

License granted to users of this work:



CC BY

https://creativecommons.org/licenses/by/4.0/

Acknowledgements

I would like to thank Professor Henrique Santos for his guidance and rigour throughout the dissertation.

To Sonae Arauco, for allowing me to work on a project in an area new to me and that helped me define the objectives of this dissertation.

Finally, I want to thank my close family, my parents and sister, for their patience and support in these last years which gave me motivation to complete this chapter of my life.

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, Braga, january 2025

Paulo Jorge Sousa Costa

Abstract

In recent years, industrial environments, which support many critical activities in our society, have been adopting new technologies like Industry 4.0 and IIoT, which brings significant benefits to organisations such as improved efficiency of operational processes, greater connectivity, increased quality and quantity of operational data, integration with enterprise systems, among others. However, this evolution of industrial environments is associated with an increased attack surface, resulting higher security risks for organisations. There has also been a notable increase in the number of attacks against these environments, as well as their level of sophistication. As a result, this dissertation aims to develop a security information and event management component, the Argus SIEM, with the requirements of industrial environments in mind, capable of collecting, processing, and correlating events from various relevant sources in industrial environments to detect malicious activity in industrial networks through various methods. This requires the integration of various security tools into a unified system. Finally, with the help of existing frameworks focusing on the security of industrial environments, it was possible to validate the developed system's capabilities against real attacks and scenarios.

Keywords industrial control systems, security information and event management. intrusion detection, network security, cybersecurity

Resumo

Nos últimos anos, os ambientes industriais, que suportam muitas das atividades críticas à nossa sociedade, têm adotado novas tecnologias como a indústria 4.0 e a IIoT, que trazem grandes benefícios às empresas como a melhoria da eficiência dos processos operacionais, maior conectividade, aumento da qualidade e quantidades de dados operacionais, integração com os sistemas empresariais, entre outros. No entanto, esta evolução dos ambientes industriais está associada a um aumento da superfície de ataque, que se traduz num maior risco de segurança para as organizações. Existe também um aumento assinalável do número de ataques a estes ambientes, bem como do seu nível de sofisticação. Perante este cenário, esta dissertação pretende desenvolver um sistema de gestão de informações e eventos de segurança, o Argus SIEM, com os requisitos dos ambientes industrais em mente, e que tenha a capacidade de recolher, processar e correlacionar eventos de vaŕias fontes relevantes em ambientes industriais, detectando, através de vários métodos, atividade maliciosas nas redes industriais. Para isto é necessário integrar diversas ferramentas de segurança num sistema unificado. Finalmente, com a ajuda de frameworks existentes, que focam na segurança de ambientes industriais, foi possível validar as capacidades desenvolvidas do sistema, face a ataques e cenários reais.

Palavras-chave sistema de automação industrial, gestão de informações e eventos de segurança, deteção de intrusões, segurança de redes, cibersegurança

Contents

I Introductory material 1

1 Introduction 2

1.1 Context . . . 2
1.2 Case Study . . . 3
1.3 Motivation and Objectives . . . 4
1.4 Research Methodology . . . 5
1.5 Document Structure . . . 5

2 State of the Art 7

2.1 ICS/OT Environment . . . 7
2.1.1 Components . . . 7
2.1.2 Network architecture . . . 9
2.1.3 Communication protocols . . . 11
2.2 Information security management model . . . 12
2.2.1 Security metrics . . . 13
2.2.2 ISO/IEC 27000 Family of Standards . . . 13
2.2.3 ISA/IEC 62443 . . . 14
2.3 Intrusion Detection Systems . . . 15
2.3.1 Data source . . . 16
2.3.2 Detection Techniques . . . 16
2.4 Security Information and Event Management . . . 17
2.5 Security Orchestration, Automation and Response . . . 19
2.6 MITRE ATT&CK Framework . . . 20
2.6.1 ICS Matrix . . . 21

2.6.2 Use Cases . . . 22

3 Prototype **23**

3.1 Requirements . . . 23

3.2 Event collection, processing and storage . . . 24

3.2.1 Beats . . . 24

3.2.2 ECS Schema . . . 25

3.2.3 Logstash . . . 26

3.2.4 Elasticsearch . . . 27

3.3 Network monitoring with Zeek . . . 27

3.4 Deployment Architecture . . . 28

4 Argus **31**

4.1 Overview . . . 31

4.2 System Architecture . . . 32

4.3 Rule Engine . . . 34

4.3.1 Rule definition . . . 35

4.3.2 Sigma Rules . . . 39

4.3.3 Event Correlation . . . 40

4.4 Alert System . . . 40

4.4.1 Alert Response Model . . . 41

4.4.2 Alert Analysis . . . 43

4.5 Asset Discovery and Monitoring . . . 44

4.5.1 Passive Monitoring . . . 45

4.5.2 Usage . . . 46

4.5.3 Limitations . . . 47

4.6 Baselining . . . 47

4.6.1 Analytics . . . 48

4.6.2 Data Flows . . . 49

4.6.3 Assets . . . 50

4.7 MITRE ATT&CK Integration . . . 50

4.7.1 Alert Insights . . . 51

4.7.2 Learning Support . . . 51

4.7.3 Coverage Assessment 51

5 System Evaluation **54**

5.1 Simulation Environment Setup 54

5.1.1 ICS Test bed 55

5.1.2 Test environment 55

5.1.3 Tools configuration 57

5.2 Test Plan 58

5.2.1 TTPs Overview 59

5.2.2 Test Preparation 59

5.3 Results 61

5.3.1 Initial Access 61

5.3.2 Command and Control 63

5.3.3 Persistence 65

5.3.4 Discovery 70

5.3.5 Lateral Movement 73

5.3.6 Credential Access 82

5.3.7 Collection 84

5.3.8 Inhibit Response Function 86

5.3.9 Impair Process Control 89

5.3.10 Impact 91

5.3.11 Other Use Cases 94

6 Conclusions and future work **98**

6.1 Conclusions 98

6.2 Prospects for future work 100

List of Figures

1 Representation of the Purdue model 10
2 ATT&CK ICS matrix 21
3 Event search with/without ECS 26
4 Argus internal deployment architecture 29
5 Argus architecture representation 32
6 Rule engine execution representation 34
7 Sigma rule importer 39
8 Alert response model 42
9 Profiler metric visualization 44
10 Asset discovery and monitoring dashboard 47
11 Baseline asset data flows 49
12 ATT&CK matrix rule coverage 52
13 Test environment architecture 56
14 Caldera agents page 63
15 Dnscat2 Windows shell command execution 64
16 C2 DNS tunnelling alert 65
17 Registry run key contents 66
18 Startup or run key registry alert 67
19 New Executable in windows user/system startup folder alert 67
20 Windows service registry key 69
21 Windows service creation/modification via windows registry alert 69
22 Ping sweep enumeration alert 71
23 Single port scan on multiple assets alert 72
24 Full port scan on a single asset alert 73

25 Windows failed authentication attempts profiling 74
26 Windows RDP bruteforce rule alert 75
27 Account authentication outside business hours alert 76
28 Account authentication attempts in multiple systems alert 77
29 Windows asset authentication account profiling 78
30 Suspicious file transfer EQL rule 81
31 Potential lateral tool transfer through an SMB share alert 81
32 Suspicious file transfer through an SMB share alert 81
33 Credential dumping activity via LSASS alert 83
34 Jump machine LSASS dump with Caldera "Powershell Mimikatz" ability 84
35 andrew's NTLM hash crack with hashcat 84
36 OpenPLC web UI interface 87
37 PLC start/stop outside business hours alert 88
38 PLC start/stop via modbus coil modification alert 89
39 Modbus write command by asset profilling 91
40 Modbus unauthorized write baseline analytic alert 91
41 OpenPLC control logic parameters modification 92
42 Operational process physical destruction 93
43 PLC new program upload modification alert 94
44 Unknown asset discovery alert 95

List of Tables

1 Basic rule attributes 36
2 Threshold rule paremeters 37
3 Selected TTPs 60

Acronyms

AD Active Directory.

AI Artificial Intelligence.

APT Advanced Persistent Threat.

C2 Command & Control.

DCS Distributed Control Systems.

DLL Dynamic-Link Library.

DMZ Demilitarized Zone.

DNS Domain Name System.

DSL Domain Specific Language.

DSRM Design Science Research Methodology.

ECS Elastic Common Schema.

EQL Event Query Language.

FQDN Fully Qualified Domain Name.

FTP File Transfer Protocol.

HMI Human-machine interface.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

IACS Industrial Automation and Control Systems.

ICMP Internet Control Message Protocol.

ICS Industrial Control Systems.

IDS Intrusion Detection System.

IED Intelligent Electronic Device.

IIoT Industrial Internet of Things.

IoC Indicators of Compromise.

IoT Internet of Things.

IP Internet Protocol.

IPS Intrusion Prevention System.

ISMS Information Security Management System.

IT Information Technology.

LSASS Local Security Authority Subsystem Service.

MAC Media Access Control.

MSDT Microsoft Support Diagnostic Tool.

NTLM New Technology LAN Manager.

OS Operating System.

OT Operational Technology.

OUI Organizationally Unique Identifier.

PLC Programmable Logic Controllers.

RDP Remote Desktop Protocol.

RTU Remote Terminal Unit.

SCADA Supervisory Control and Data Acquisition.

SIEM Security Information and Event Management.

SMB Server Message Block.

SOAR Security, Orchestration, Automation and Response.

SOC Security Operations Center.

SPAN Switched Port Analyzer.

TTP Tactics, Techniques, and Procedures.

VPN Virtual Private Network.

Part I

Introductory material

Chapter 1

Introduction

1.1 Context

Industrial Control Systems (ICS) are responsible for monitoring and automating operations in an industrial environment, supporting various critical infrastructures that our society relies upon, such as the energy sector, water treatment, oil, gas, manufacturing, etc. In the last decade, it has been possible to observe a serious increase in attacks directed against industrial environments, beginning with Stuxnet in 2010, attacks against Ukraine's energy grid in 2015 and 2016, attacks on a petrol and oil station in Saudi Arabia, and many others, with devastating results and huge losses [61].

In recent years, industrial companies from different sectors of activity have been under a digital transformation process, increasingly adopting new trends and technologies such as Industry 4.0 and Industrial Internet of Things (IIoT), with the goal of enhancing the quality and efficiency of operational processes, better-supporting tools and visualization of the operational environment, which represents a crucial element to achieve strategic goals and increase business innovation and profitability [27]. As a result of this trend, new and important questions related to the cybersecurity domain have emerged, specifically in ICS and Operational Technology (OT) networks.

Consequently, OT networks have evolved to a more complex architecture, involving not only operational devices but also Information Technology (IT) support systems, internet-connected sensors/devices, and others [65]. With IIoT, there is also a need to fulfil scaling requirements because organisations need to have the ability to add new devices to an operational network seamlessly. These requirements and communication channels require robust network architecture design and solid infrastructures that can handle all the communication, storage, and data processing. This enables advanced functionalities and use cases, such as substituting certain mechanical mechanisms, which require physical systems to be integrated with IT [65], extensive access to real-time data from beginning to end of the value chain, analytics, big data, and cloud use cases [63]. On the other hand, all of these new opportunities come

at the cost of a wider attack surface in industrial environments, resulting in the need to develop and implement better and more resilient security controls.

However, despite the increasing adoption of IT technologies in operational environments, there are a set of distinct fundamental characteristics between IT and ICS/OT, for instance the difference in the evaluation of risk and priorities, which in the case of ICS/OT, there is a great concern and prioritization in the safety of human lives and maintaining a continuous production, to avoid financial losses, as well as different performance and availability requirements and new concerns in the implementation of security controls, which respect and guarantee the integrity and availability of the operating environment [65]. Additionally, concerning digital technologies, the maturity and expertise levels in OT environments are lower, demanding special user requirements for cybersecurity-related tools [56].

Therefore, due to the underlying impact and the growing attention from malicious actors to ICS and the fundamental differences between the IT and OT domains, it is necessary to tackle these new challenges differently, tailored to the ICS/OT reality, which requires the development of new skills, as well as adaption of the risk analysis process and the implementation of security controls, where the security metrics will play a key role.

Faced with this paradigm shift, this dissertation intends to develop a Security Information and Event Management (SIEM) with features adapted to the IT and ICS/OT environments, with the goal of creating a more comprehensive and capable defense security solution that can be used to tackle the security challenges present in industrial environments.

1.2 Case Study

This thesis is part of the Sonae Arauco Advanced Decision Making in Productive Systems Through Intelligent Network project, ADM.IN, for short, whose main goal is to develop smart tools to support the decision-making process by increasing the quality and value of Sonae Arauco products and their operational availability.

This thesis contribution goes to the cybersecurity component of the project, whose purpose was to implement an information security management system that would comply with the specific security requirements of an ICS/OT environment, desirably within the main reference standards. The selected case study for this component was remote access to operational production lines for maintenance activities.

1.3 Motivation and Objectives

This dissertation topic choice was motivated by a long-last interest in cybersecurity and its challenges, as well as the opportunity to work on a real project in Sonae Arauco that allowed me to learn and comprehend the concepts, problems, and challenges behind the OT domain.

At the beginning, the initial dissertation focus was on the different host and network metrics that could be collected and analysed to detect malicious activities in ICS/OT environments. However, throughout the development of the dissertation, the focus shifted to the development of a full-blown SIEM, as it solved the main challenges and requirements for the proper defence of industrial environments, involving not only the analysis of host and network metrics but also logs coming from different assets. These changes could even deem a dissertation title change, focusing on the development of a SIEM adapted to modern ICS/OT environments.

Therefore, this dissertation aims to develop a SIEM system, tailored to monitor and protect ICS/OT environments, respecting their requirements to enhance security. Due to increasing convergence between the ICS and IT domains, this solution's detection capabilities must cover both domains.

To achieve this goal, the new detection system will be designed with ICS requirements in mind, as it will integrate multiple technologies and tools to collect, process, and analyse events that can be correlated from multiple sources to detect malicious behaviour in the environment.

Thus, the main objectives that are intended to be achieved in the course of the dissertation are:

- Increase the knowledge about ICS/OT environments, how they are structured, how they work, and how they can be attacked and protected.
- Identification of the main attacks and threats present in ICS/OT environments, with a special focus in vectors that come from the IT domain, like remote access.
- Exploration of the available security tools, both at the host and network level, that can be used and implemented to support the final solution.
- Development of the Argus SIEM prototype.
- Validate the solution detection capabilities with a comprehensive test plan.

1.4 Research Methodology

The research methodology used throughout this dissertation is the Design Science Research Methodology (DSRM), which provides a structured and iterative framework that supported the development of this dissertation [3]. This methodology is comprised by six phases: problem identification, objectives definition, design and development, demonstration, evaluation and communication.

As such, the dissertation starts by clearly defining the research problem, which is the lack of security solutions that are capable of protecting emerging ICS/OT environments.

To solve this problem, the objective of the dissertation was defined, which is mainly the development of a SIEM system adapted to the ICS/OT requirements.

The next step was the conceptualization and design of such security solution. It started by defining how security logs and metrics were to be collected, from which systems, the selected software, how they will processed and stored. Then the Argus SIEM was conceptualized, including which functionalities it will have, how the subsystems work and how they interact with each other. Finally, the development of the SIEM started, implementing all the defined functionalities and capabilities that will help attain the dissertation goal.

Once the development was completed, it was time to create a test plan that will assess how the security solution behaves and detects IT and OT attacks in ICS/OT environments.

Finally, the final SIEM evaluation was performed, and the results were produced and shared.

1.5 Document Structure

This dissertation is structured into six chapters. The first chapter is an introductory chapter that explains the dissertation context, motivation, methodology, and main goals.

The second chapter explains the fundamental concepts of modern ICS/OT environments and their components, as well as the information security management models that can help protect them. It also analyses the types of systems that can be implemented to detect intrusions effectively according to the ICS/OT requirements. Finally, the MITRE ATT&CK framework is explored to understand how its use cases can help defensive teams better protect their environments.

The third chapter addresses the prototype requirements and the technologies and tools that are the basis of the proposed solution. It also presents an example of the solution's deployment architecture.

The fourth chapter dives into the design and development of the Argus SIEM, describing its system

architecture, especially the core of the solution, the rule engine, and how all components interact and work together. After this, other subsystems that incorporate the final solution are also presented, namely the passive asset monitor, the baseline system, the alert system, the MITRE ATT&CK framework integration, and other management functionalities.

The fifth chapter explains the system evaluation process and results, including how the simulation environment was built and structured. Afterward, the test methodology and plan are presented. The chapter ends with the results section containing all the tests and their corresponding results.

Finally, the sixth chapter lists the dissertation's main conclusions and suggestions for future works.

Chapter 2

State of the Art

This chapter presents the state of the art of cybersecurity in industrial environments, including an overview of common industrial control systems, the available security management models, and the technologies and systems deployed in such environments.

2.1 ICS/OT Environment

OT environments are the foundation of industrial processes and critical infrastructure, connecting physical machinery equipment to industrial control systems. It is responsible for ensuring operational safety, efficiency, reliability, and availability across many industries that we, as a society, rely upon.

Achieving these objectives often involves proper machinery and equipment communication, monitoring, and automation. By integrating sensors, actuators, controllers, and specialized software, OT environments enable precise control over variables such as temperature, pressure, and other critical parameters, that are part of the operational process. These industrial devices also use specialized network protocols to communicate with each other, and they are an important factor to take into account when the objective is to digitally secure these systems.

2.1.1 Components

There are multiple types of ICS, each with a unique level of usefulness for different industrial sectors. Supervisory Control and Data Acquisition (SCADA) systems and Distributed Control Systems (DCS) are the most common types. In many different implementations, it is typical to find a hybrid approach where one or more control systems work simultaneously to achieve operational goals [65].

As such, the most common control systems that can be found in industrial environments are:

- **Supervisory Control and Data Acquisition (SCADA)** - SCADA systems integrate geographically distributed industrial systems such as a Programmable Logic Controllers (PLC), Human-

machine interface (HMI), Remote Terminal Unit (RTU), and others, working as a centralized component that monitors and administers operational processes by analysing operational data from different operational inputs and outputs gathered by the system. It collects information from local devices, stores it, and displays, in real-time, contextual information to an operator in a single interface, allowing him to make decisions based on the information that the system provides.

- **Distributed Control Systems (DCS)** - Distributed control systems are used to monitor and control industrial systems that are all in the same geographic location. These systems are composed of multiple subsystems, each one controlling a single production process that are managed by a supervisory control component, effectively monitoring the complete production chain. It adopts a modular system architecture, which limits the impact on production and system availability in case of failure of one of the various subsystems.

- **Programmable Logic Controllers (PLC)** - A programmable logic controller is one of the most common control systems found in an operational environment. Its purpose is to execute operation decisions based on a defined program and the input data from various industrial devices.

In addition to the industrial control systems themselves, OT environments incorporate different industrial components and field devices, each with a specific role in the operational process [65]:

- **Actuators and Sensors** - These are some of the most important components found in any OT environment. Sensor devices are responsible for generating measurements based on real-world attributes, which are then transferred to the corresponding control system. On the other hand, actuators like motors or valves, receive commands from the control system and directly influence the operational behaviour of the underlying equipment.

- **Human-machine interface (HMI)**—Human-machine interfaces combine software and hardware components to present a user interface that gives an operator insights into the current process status, historical data, KPI performance, and other process-related information. They also provide control to the process operation, allowing it to make configuration and parameter changes, and they also work as emergency controls as they enable manual overrides.

- **Remote Terminal Unit (RTU)** - A remote terminal unit is a field device that bridges a hardware device and an industrial system such as SCADA or DCS. Its purpose is to forward sensor data, both analogue and digital, from various processes to one of these centralized monitoring systems.

- **Intelligent Electronic Device (IED)** - An integrated microprocessor-based controller that receives and processes sensor data and sends it to an external system, like SCADA, makes it an essential component of such a system. Based on this gathered data, it can act precisely, like turning off a power switch, if any fault is detected.

- **Data historian** - A data historian is a centralized database that stores process-related information. It can perform statistical analysis on the stored data, and it is normally used to transfer the gathered operational data to the IT enterprise systems.

2.1.2 Network architecture

As stated before, the convergence of IT and OT domains brings many problems and challenges to the table, particularly in terms of how ICS/OT networks are structured to ensure the safety of the operational network and its devices. Because of this, ICS networks changed from isolated networks, which required physical access to plant sites, to fully interconnected networks, that are completely exposed to a new and vast attack surface if no access or authorization controls are implemented [51].

Protecting the operational network from the untrusted enterprise network is necessary, so segmentation emerges as one of the fundamental access control mechanisms for securing ICS networks [65].

The Purdue reference model is a well-established structural model that helps design a secure ICS network architecture by placing the different IT and OT devices into different zones or levels [76]. With the help of different access controls like demilitarized zones, it is possible to ensure the isolation between the IT domain (level 4-5) and the ICS/OT domain (0-3 levels). Figure 1 depicts a high-level representation of the different zones or levels, and devices that comprise the Purdue model.

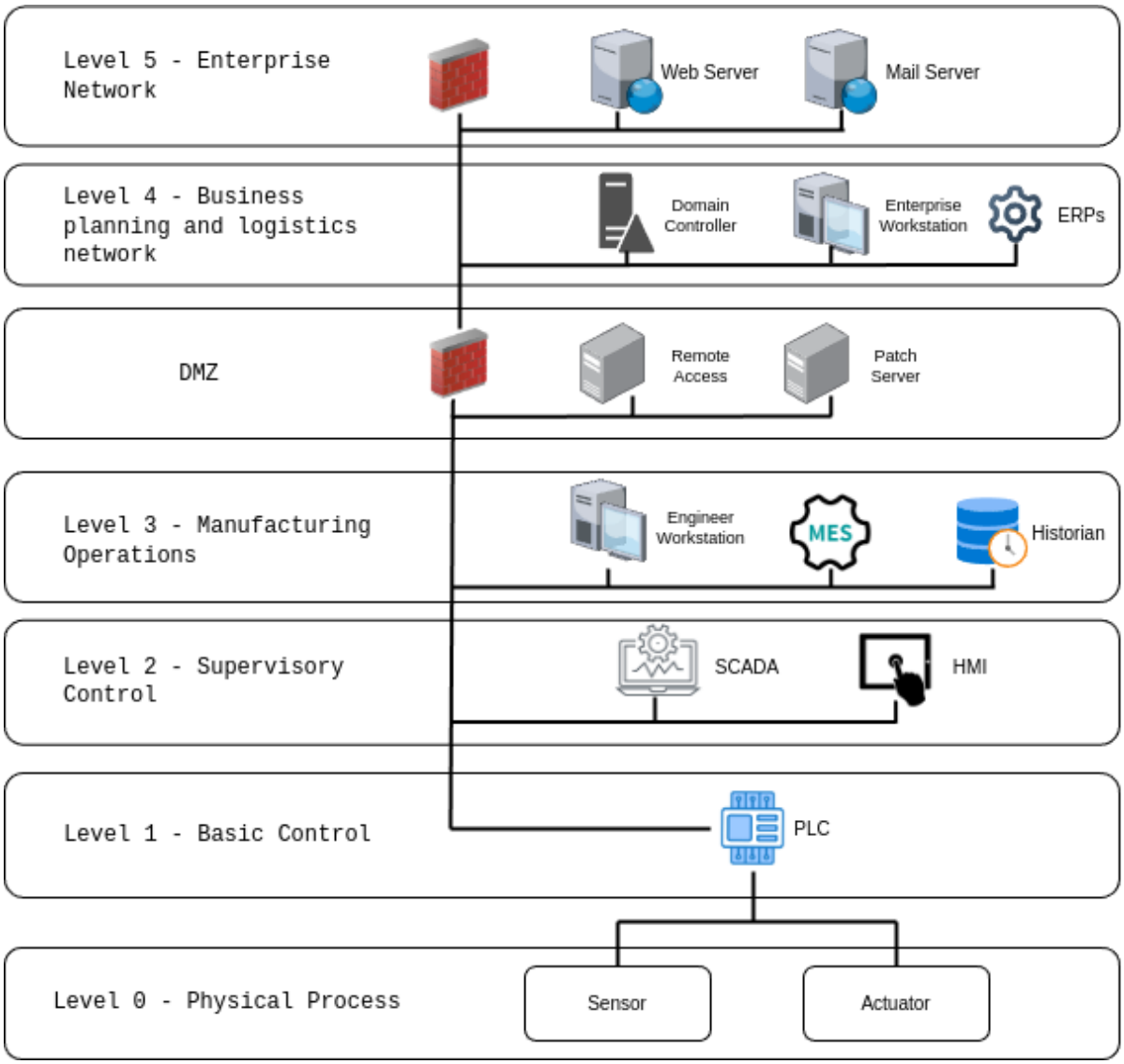


Figure 1: Representation of the Purdue model

The highest element in the model is the Enterprise zone, subdivided into levels 5 (enterprise network) and 4 (business planning and logistics network). This zone includes both external network connectivity and internal business networks that usually contain application servers, email servers, ERPs, Domain Name System (DNS) servers, business desktops, databases, and other IT systems and devices that support the operational processes management.

Between the enterprise and manufacturing zones is the plant demilitarized zone, whose responsibility is to prevent direct and unrestricted access from IT devices in the enterprise zone to OT devices that are located in the manufacturing zone. By positioning intermediate systems like data historians or jump machines, it is possible to access the required industrial process information without exposing directly the OT systems.

Right after the Demilitarized Zone (DMZ) is the manufacturing operations zone at level 3. This zone contains the manufacturing execution systems (MES) and supports the management, supervision, and control of operational workflows running on the plant floor. It is also responsible for collecting and aggregating operational data coming from lower-level devices that will be forwarded to the IT systems at level 4. At this level, it is possible to find the data historians, application servers, domain controllers, engineer workstations, etc. The cell security zone (level 0-2), includes systems that monitor and control physical processes such as SCADA, DCS and even HMIs (level 2), systems that control and send commands like RTUs and PLCs (level 1) to production field devices, sensors and actuators (level 0).

2.1.3 Communication protocols

To communicate with each other, industrial devices use ICS-specific communication protocols that were created and designed with operational process automation as its core. They are an essential component of how OT environments operate because they fulfil the OT communication requirements. These protocols need to be reliable because industrial environments can be very different from each other, as they operate under difficult conditions, for example, high temperatures and electronic interference that affect the behaviour of electronic devices. They also need to be performant, as these industrial control systems exchange high volumes of data in short time frames that need to arrive rapidly at their destination without missing a beat. Finally, the communication protocols must be interoperable, as OT environments have many different types of devices from different vendors, which all need to seamlessly communicate with each other without further configuration.

Unfortunately, many of these communication protocols were created many years ago, when security requirements were not a priority, resulting in insecure protocols that do not comply with modern security necessities [71]. For example, the Modbus protocol was created in 1979. These systems were also air-gapped and isolated from other networks, which would require physical access from potential attackers. Nowadays, this is not exactly always the case.

As such, it is important to identify the most used and relevant ICS protocols:

- **Modbus** - One of the oldest protocols used for process automation, Mobdus is a request-response communication protocol that supports serial interfaces(RS232/RS485) and the TCP/IP network stack as its transmission methods. In its master/slave model, a master can communicate with multiple slaves that only transmit data to the master once explicitly requested. Through the years, due to this protocol's age, new iterations were developed, namely Modbus-F2009, Modbus-S2015, and Modbus-A2018, to add authentication and encryption capabilities to ensure the confidentiality,

integrity, and availability of the systems that use this protocol, more commonly used by PLCs [71].

- **DNP3** - Developed for SCADA systems used in the energy sector, DNP3 is an open standard protocol that optimizes data transmission between critical infrastructure devices. It implements a master-slave architecture, supports transmission over a TCP/IP network or a serial bus connection, and is commonly used by HMIs(master) and PLCs and RTUs (slaves) devices. It also provides extensive capabilities such as multiple data types in the same message, time synchronization, unprompted request response, and others [8].

- **OPC UA** - OPC is a client/server protocol commonly used by HMI devices where the client retrieves the necessary information from the server. It has multiple interfaces for different purposes. UA, or Unified Architecture, is the newest released in 2006, supported by the TCP/IP network stack, and has become one of the most important standards for Industry 4.0 and IIoT adoption [71].

- **S7Comm** - A Siemens proprietary protocol explicitly developed for PLCs, allowing access to data within a PLC, data exchange between PLCs, and PLC programming [73].

- **MQTT** - A lightweight message transmission protocol with a publish/subscribe model designed for IoT and M2M devices with limited network and processing resources. In this model, the broker(server) is responsible for receiving data from publishers(clients) and relaying it to other clients that need that data, in this case, the subscribers. Due to the increasing adoption of IIoT in OT networks, the importance and usage of this protocol for industrial use cases will only grow in the future [75].

2.2 Information security management model

An information security management model focuses on the risk of failure of an information system resource, such as a critical component of an information system and affecting one or more of the fundamental security properties: confidentiality, integrity, and availability. This risk can be evaluated and correctly prioritized by defining a set of actions that aims to mitigate the identified risk [57].

To implement an information security management model, it is necessary to execute the following set of tasks: identification of resources, vulnerabilities, threats, attacks, and risk assessment [57]. The first step aims to identify the most critical resources in the organisation, which requires, when possible, quantifying the value of these same resources, information that will be important in the later stages of the implementation. The vulnerability identification phase identifies vulnerabilities in the resources previously

selected. The third step consists of identifying potential threats and attacks to which the resources are vulnerable and the probability of this attack being executed. Finally, the risk assessment phase is based on the information collected in the previous steps; its purpose is to identify the real security risk that the resources are exposed to, describing how the risk should be addressed, which can be eliminated, shared, or mitigated through the use of security controls that exist to reduce the associated risk to an acceptable level, in which it can be monitored and controlled.

Consequently, security controls are crucial components in information security management, some of which are considered fundamental, with significant impact on the three main security properties, such as access controls, firewalls, and intrusion detection systems [57]. However, the security controls implemented by an organisation must be efficient and effective, resulting in the need to evaluate the performances of these controls, which requires adequate metrics.

2.2.1 Security metrics

Metrics, in the context of information systems security, are a way to evaluate the implementation, efficiency, and effectiveness of security controls present within an organisation. For a metric to be considered good, it needs to fulfil a set of fundamental characteristics; it must be expressive, objective, and repetitive, and its cost should never exceed the value of what it is measuring. Through measuring, organisations can monitor the state of different activities, providing data that facilitates its improvement through the execution of corrective actions based on the measures. Furthermore, it also increases accountability, improves the effectiveness of information system controls, demonstrates compliance, and provides quantifiable inputs to allocate resources [6].

There are three types of metrics: implementation, efficiency or effectiveness, and impact [6]. Implementation metrics demonstrate the progress in implementing an information system, which may be related to associated controls, policies, or procedures. Effectiveness or efficiency metrics are responsible for monitoring the processes and security controls, verifying that they are correctly implemented, operating according to what is expected, and fulfilling the objective for which they were implemented. Finally, impact metrics are relegated to the specific impact on the organisation's mission.

2.2.2 ISO/IEC 27000 Family of Standards

One of the most known and applied information security management models in IT is centred on the ISO/IEC 27000 family of standards. It provides guidelines for implementing, monitoring, and improving an information security management system, as well as managing information security risks, with the

main goal of steering organisation efforts to protect all business information [58]. It uses a risk-based approach to achieve this, helping organisations identify, examine, and evaluate risks in an information system. The identified risks are then prioritized, and for each one, a set of security controls is selected that will decrease the risk to an acceptable level [53]. The most well-known standard of this family is the ISO/IEC 27001, which specifies the requirements for establishing, implementing, maintaining, and continually improving an Information Security Management System (ISMS). Furthermore, it is the only one linked to a certification process in information security.

However, as stated, the ISO/IEC 27000 standards focus primarily on IT security, addressing the IT requirements, problems, and challenges. As such, although it covers a wide range of use cases, overlapping with some industrial organisation requirements [53], it is still necessary to use OT-oriented models to approach the risk-based process with the industrial environment characteristics at its core.

2.2.3 ISA/IEC 62443

The ISA/IEC 62443 series of standards was created to tackle cybersecurity problems and challenges for Industrial Automation and Control Systems (IACS). It also uses a risk-based approach and defines a comprehensive set of models, terms, requirements, security controls, and processes that can be used to ensure the safety and security of IACS and, consequently, the confidentiality, integrity and availability of operations. It builds upon previous works by considering the IT security requirements defined in the ISO 27000 series in an industrial environment [53], but adapted to OT environment's particular requirements and needs.

The ISA/IEC 62443 series comprises different documents that can be grouped into four groups [28]:

- **General group** - Includes the IEC 62443-1-1 document, establishes common terminology and concepts that will be used as a basis through the standard series, providing, in this way, fundamental knowledge to understand the knowledge shared in the series.

- **Policies and procedures group** - Includes the IEC 62443-2-1, IEC 62443-2-3, and IEC 62443-2-4 parts, addresses how an organisation can, more precisely, secure IACS as it defines the policies and procedures that for example, should be implemented in the cybersecurity management system or how to patching IACS software should be managed and executed.

- **System requirements group** - Includes the IEC 62443-3-1, IEC 62443-3-2, and IEC 62443-3-3 parts, outlines the requirements that industrial systems should fulfil to achieve different security levels.

- **Components requirements group** - Includes the IEC 62443-4-1 and IEC 62443-4-2 parts, defines the requirements necessary to develop software tailored for an IACS component, such as an embedded, host, or network component,

Furthermore, the increasing IIoT trend adoption in industrial environments comes with the promise of more efficient and intelligent operations at the cost of a much more extended attack surface, where the line between traditional IT and OT segregation of responsibilities becomes fuzzy, which introduces more challenges in designing industrial environments architectures, and consequently protecting them adequately. The ISA/IEC 62433 series, although not designed with IIoT in mind, lays the foundations and defines some controls that will help address the new challenges brought by IIoT. Some examples are the security zones and network segmentation, which separates the devices, directly controlling the IACS and supporting IT internet-connected devices into different networks, providing additional protection. On the other side, there are clear limitations that should be addressed in the future, such as the complex management of network segmentation rules and patch management due to the rapidly increasing number of IIoT devices that are being deployed in industrial environments, as well the necessity of end-to-end encryption in communication protocols between devices, since the boundaries of the industrial networks are constantly changing [30].

For this reason, the ISA/IEC 62433 series of standards should address the IIoT-specific requirements and challenges to help organisations ensure the safety and protection of their systems.

2.3 Intrusion Detection Systems

Intrusion Detection System (IDS) are tools that detect attacks and abnormal behaviours by monitoring systems and network activity. IDS collects data from various relevant sources, such as application and security logs, network traffic captures, and others, that are analysed and correlated afterwards, using different methods to detect suspicious and malicious activities. The inherent passivity in this type of system is essential in industrial environments since any reactive action to potential malicious activity could disturb or compromise the continuous industrial operation [65]. This kind of environment disturbance can lead to huge losses for industrial companies as they can directly impact the operational process.

However, intrusion detection systems were originally designed and developed with IT systems in mind, not considering the unique properties and characteristics of ICS/OT environments[26]. For this reason, once more, it is necessary to adapt these systems to work correctly and safely in such a complex and new environment. Additionally, IDS are prone to generate false positives if they are not properly config-

ured to the organisation's environment characteristics. This requires extensive testing and fine-tuning of the signatures or threshold that defines an event as malicious. Due to the inherent characteristics of IC-S/OT environments, its absolutely essential that this environment-oriented fine-tuning processed is done with the upmost care, taking in consideration all the peculiarities of the organization operational process environment.

Traditional intrusion detection systems can be grouped according to the data source and the detection method. In the following subsections, those types are further detailed.

2.3.1 Data source

When grouped by the data source, intrusion detection systems can be defined as host-based or network-based. A host-based intrusion detection system is responsible for monitoring the internal behaviour of a system to identify abnormal behaviours, which could reveal signs of malicious activity [29]. Generally, this system uses an agent installed in the target system, which will collect different kinds of data from various internal data sources, such as system events, application logs, and others. Because most systems used in industrial environments are strictly closed down and do not allow or support this kind of application, host-based IDS are not suited for this kind of environment, with some exceptions.

On the other hand, a network-based intrusion detection system passively monitors the activity in a controlled network, which makes it an ideal choice to use in an ICS/OT environment, where any disruption in the internal OT network could result in severe accidents [10]. It is responsible for capturing and analyse network traffic to detect intrusions using various data analysis techniques, usually through signatures or traffic anomalies[26]. This can be achieved through analysis of network flows, which requires processing packet header information, or by inspecting the data contents of each packet with data-packet inspection (DPS), which allows a more complete analysis of the different network protocols that are being used.

2.3.2 Detection Techniques

Another way to group intrusion detection systems is through the detection technique they use to identify possible attacks, specifically signature-based or anomaly-based methods. A signature-based intrusion detection system compares the collected activity data against a set of signatures or Indicators of Compromise (IoC), effectively detecting known malicious activity [26]. However, this detection method can not detect small changes or new intrusion behaviours, which requires complementing the monitoring system with another types of detection technique.

An anomaly-based intrusion detection system starts by creating a model that characterises a system's

normal behaviour and functioning using various analysis methods. With this base model, the system's behaviour is monitored in real-time, which triggers an alert when a significant deviation from the base model is detected, making it ideal for detecting new and unknown attacks, but with a high degree of false positives[26]. In summary, it is possible to classify these kind of systems into three sub-categories: statistical, specification, and machine learning.

Statistical-based systems detect suspicious activity by using statistical algorithms, classifying every event with a calculated risk score, and issuing an alert when this value exceeds the defined limits set by the base model. A specification-based system uses a model based on the workloads and specifications that characterise a specific system's operation and normal workings. Finally, a machine learning-based system builds a mathematical model using machine learning algorithms, using an appropriate dataset to the ICS environment, to train a model, to categorise events as malicious or not, given the normal behaviour and activity detected in the ICS environment [29].

2.4 Security Information and Event Management

Security Information and Event Management (SIEM) are security solutions that can collect, normalise, aggregate, store, and correlate different types of events to detect malicious activity in an organisation [2]. They are a fundamental pillar of the existing Security, Orchestration, Automation and Response (SOAR) system since they can manage large amounts of events, scale from small to large organisation sizes, and create alerts that will be analysed by the Security Operations Center (SOC) analysts according to defined rules and analytics.

A SIEM comprises the following main components: log management, correlation engine, and reporting system. Log management starts by collecting log data from various data sources, like firewalls, intrusion detection systems, specific applications, host endpoints, and others, which are then parsed and normalised since each data source has its own event schema representation, which requires its conversion to a common format and representation so that the SIEM can correlate different data sources events that have the same common structure and attributes. After the events have been normalised, they are stored in a database that feeds the correlation engine, events which are then analysed in real-time using different analytics like analyst-created rules, allowing it to detect complex attacks more rapidly and precisely. Once a malicious security event is detected, the reporting system will issue an alert with contextual and actionable information that will be forwarded to the responsible security personnel [2].

From the point of view of critical infrastructures, SIEMs have proven to be an essential security solution

that can be used to monitor operational networks in real-time since they have correlation capabilities that allow the analysis of different events and metrics that can be used to detect deviations in the network behaviour, that could be a result of malicious activity [25]. With the increasing focus on ICS security, multiple SIEM vendors started to add OT analysis capabilities to their systems. This trend encouraged the creation of ICS focused SIEMs built with the ICS/OT challenges in mind, providing custom utilities and functionalities that no other traditional SIEM could. An example is the SPEAR SIEM, custom-made for smart grid environments and operations [54].

For these reasons, SIEMs provide organisations with a central component that manages the organisation's security posture, providing advantages such as real-time monitoring, advanced correlation of events and analytics, incident case creation and reporting, integration with multiple security-focused tools like threat intelligence feeds, SOARs, regulatory compliance assessment, and others.

Nevertheless, SIEMs also have some limitations, including incomplete data and over time loss of capacity to process all security events produced by the organisation's activity, resulting in data gaps that could be used in correlation analysis [2]. Furthermore, most existing SIEM have basic correlational analysis, which limits the ability to detect advanced attack chains that require complex historical correlation of events. They also commonly have limited data visualisation of actionable data reproduced by the system[25]. Finally, SIEMs face some other difficult challenges, such as the scale of the organisation, including the various devices, systems, sensors, and the data they produce that must be processed and stored for future analysis [2].

To properly understand the state of current SIEM solutions, as well their capabilities and adaptability to ICS/OT environments, some publicly available SIEMs were evaluated. An analysis of the SIEM market shows that there is a vast selection of SIEM solutions from different security organizations: Splunk, IBM QRadar, Microsoft Sentinel, Crowdstrike Next-Gen SIEM, and many others. It is also possible to identify two trends: the majority of the solutions are proprietary software and they all focus on the IT side of things. Therefore, there was a prioritization on the selection of open-source solutions that can be adopted by any organization. The selected SIEMs were:

- Wazuh
- Elastic Security
- Graylog

These three SIEMs are all well-established, mature solutions used by many organizations to monitor malicious threat behaviour and protect enterprise environments. They all have great traditional SIEM capa-

bilities such as log correlation engines, alert and reporting systems, incident response utilities, compliance management, threat intelligence integration, and others. More specifically, both Wazuh and Elastic Security have dedicated agents, that unleash the most advanced features of these platforms like malware detection, active response, complete asset information (OS version, applied patches, installed software), and many other third-party integrations. Comparing the three solutions, the Elastic Security SIEM seems the most advanced and feature-rich software, with intense development and newly-added AI capabilities.

On the other hand, the dedicated agents are intended to be installed in traditional IT assets, which is not the case in ICS/OT environments, where the majority of the assets does not allow or even support this type of integration with third-party software. There is also a lack of OT-oriented features like passive asset discovery that greatly increases the visibility of the network, ICS-specific protocol analysis features, and OT threat intelligence knowledge.

2.5 Security Orchestration, Automation and Response

Another emerging security solution is the Security, Orchestration, Automation and Response (SOAR) technology that complements SIEM solutions by providing comprehensive analysis, investigation, and automatic response capabilities. These functionalities are necessary as they can save organisations a lot of time and money since the number of alerts and frequency of security events are increasing, requiring organisations to have enough specialized workers to adequately analyse, investigate, and respond to security incidents [31].

This technology consists of three different components: security orchestration and automation (SOA) unifies all external and internal tools and functions, defining the automated workflows that will be executed; security incident response platform (SIRP) allows incident response teams to define policies for different types of incident scenarios, including pre-defined tasks, which will be triggered when the conditions of that specific scenario occur; threat intelligence platform (TIP) uses known public/private threat information sources to detect, in real-time, new malicious behaviours, providing a vital defence utility against novel security threats.

Therefore, SOAR automates the security incident response process by integrating different tools with internal functions such as incident case management and threat intelligence, which feeds the detection components with the latest indicators and malicious behaviours detected in the wild. As a result, organisations that utilize SOAR can improve the overall quality and effectiveness of their internal security response team by shortening the time between detection and response to an incident, helping build knowledge

bases that can be used to detect future threats, and providing better visibility of critical OT systems and their security posture[34].

However, when comparing the utility of SOAR in IT environments to ICS/OT environments, the latter presents some challenges related to implementing and using such reactive systems in an industrial environment. Because operational environments involve physical equipment and devices, it is necessary to be very cautious in responding to a threat, especially in an automated way, since any action could disrupt the organisation's production or cause a real accident involving the organisation's employees. For this reason, applying SOAR in industrial environments is a complex and lengthy task, as it automated response capabilities require extensive knowledge, testing [65].

Finally, its also important to mention that these technologies are commonly used by traditional SOC operators, which are increasingly being requested to also handle OT threats in the organizations, task that they are not capable or prepared to operate [33]. This is largely due to a lack of expertise in the OT domain, as SOC teams typically specialize in IT-specific threats and do not possess the same level of knowledge about ICS/OT systems, protocols, and environment behaviours. As a result, there is a lack of specialized operators that have the required expertise to properly operate this kind of technologies in ICS/OT environments.

2.6 MITRE ATT&CK Framework

The MITRE ATT&CK framework is a knowledge base that defines the tactics and techniques used by real threat actors, including all the phases underlying these attacks and the behaviour that threat actors reveal throughout the process[66]. This framework consists of a set of principal components, such as tactics, which represent a tactical objective of the threat actor during an attack; techniques, which describe how an attacker can achieve the tactical objective, that can be further divided into sub-techniques, which detail more specific ways of achieving the same goal. Furthermore, it also documents the specific behaviour of certain threat actors, namely their procedures, which are specific implementations of a technique or sub-technique by the threat actor. It also presents, for each technique, a set of mechanisms that detect attempts of exploitation of the described behaviour, as well as possible mitigations that security teams should apply to prevent it.

This framework is composed, at the moment, of three different matrices, each focusing on a different domain of activity: industrial control systems, enterprise networks, and mobile devices. The ICS matrix had as its primary development motivating element the attacks against the Ukraine energy grid in

2015 and 2016 [1], as well as the goal of understanding how the ATT&CK framework, already mature in the enterprise domain, could be applied to the industrial sector, which has an entirely different set of requirements and challenges.

2.6.1 ICS Matrix

The ICS matrix overlaps with the Enterprise one, especially in the early stages of the attack chain, where there is increasing interaction between the IT and ICS/OT domains. However, the main focus of this matrix is on the actions that malicious actors perform in systems unique to these environments and that characterise the ICS/OT domain. The documented behaviours that differentiate the ICS matrix are present in more advanced phases of the attack life cycle, normally targeting devices that operate in lower levels of the Purdue model, resulting in the need to complement existing information, like the specific action that malicious actors demonstrate in this domain [66].

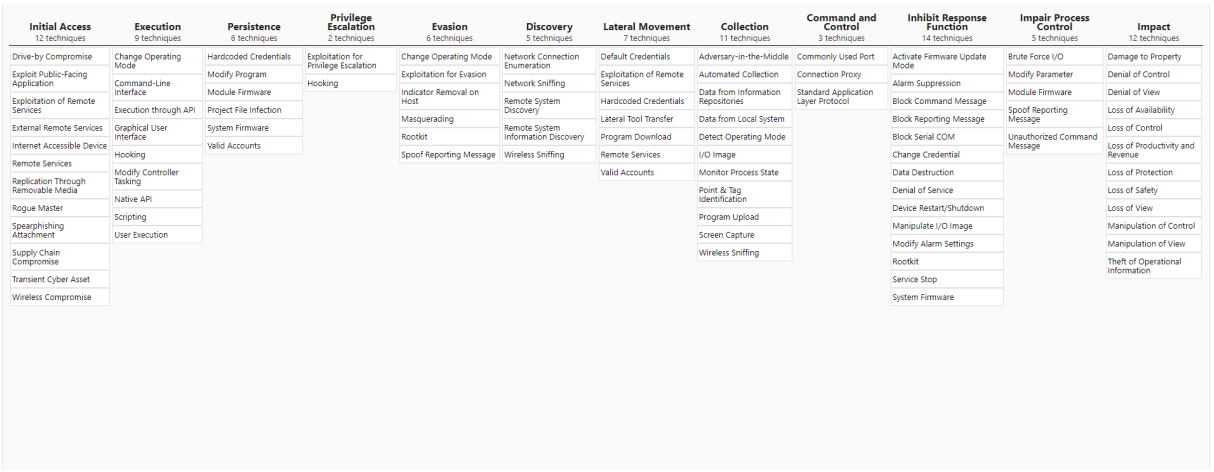


Figure 2: ATT&CK ICS matrix

Figure 2 presents, using the ATT&CK navigator tool [46], all the tactics and techniques present in the ICS matrix. By analysing this figure, one of the most relevant changes in the ICS matrix is the addition of two ICS only tactics: Inhibit Response Function and Impair Process Control. These two tactics were added to cover scenarios with ICS only assets, like a PLCs, HMIs, and others, as they are involved in the operational processes. These changes can better reflect how these assets can be compromised or used to inflict damage on the attacker's victims.

2.6.2 Use Cases

The MITRE ATT&CK framework provides a vast and exhaustive amount of information, allowing organisations to apply its matrices to various use cases.

One such use case is the assessment of defence gaps, which allows organisations to evaluate the effectiveness of their security controls, including monitoring tools, existing security policies, and mitigations. Therefore, defining and prioritising modifications or additions to existing security controls can cover the identified gaps.

Another relevant use case is the development of behavioural analytics, which in ICS/OT environments presents a valuable opportunity to create a powerful defence mechanism, one that can detect and flag suspicious activity on the network, often indicative of stealthy actions by threat actors.

Additionally, as will be shown in the results section of this dissertation, the SOC maturity evaluation case can also take a crucial role in the protection of ICS/OT environments, as it allows us to understand how effective the organisation is in detecting security incidents.

Some matrices also have specific use cases. The ICS matrix enables two use cases: failure scenario development and education resource. The first one explores the risk associated with component failures in ICS/OT environments. organisations that comprehend the different types of failures that can happen in their environment and how they might be induced, can then better protect their systems and prevent attackers from affecting the operational process. The second use case goal is to provide better OT knowledge to cybersecurity professionals, presenting how these environments work, how they are attacked, and how they can be protected.

Chapter 3

Prototype

This chapter describes the requirements the developed solution aims to meet and the justifications for the various technologies and tools used. Then, the recommended deployment architecture of the system is presented.

3.1 Requirements

As stated earlier, the IT and OT convergence, along with the adoption of IIoT, contributed to an increase in the attack surface of industrial environments, allowing a Advanced Persistent Threat (APT) to more easily gain a foothold in an industrial network, normally through the exploitation of IT services, and then pivot through the network until it reaches a position where it can interact or attack the OT devices present in the internal network. In order to ensure a secure and seamless interaction between the IT and OT domains, it is imperative that the developed solution possesses the ability to effectively monitor and detect any potential attacks on both fronts. This entails the implementation of a comprehensive security system that can identify and respond to threats, vulnerabilities, and anomalies in real-time.

The following list contains the identified requirements that the developed system must meet, based on the information gathered in the initial research for this dissertation and the ICS/OT monitoring and detection recommendations enumerated by the Cybersecurity and Infrastructure Security Agency (CISA) [7]:

- Collect and process data from the various data sources across the ICS/OT and IT domains.
- Analysis capabilities for the different ICS network protocols identified in section 2.1.3.
- Baseline capabilities that enable the detection of deviations from a given baseline, which can include the protocols used, connections between assets, operations executed, and others.
- Detect malicious activity by monitoring thresholds of defined metrics.

- Detect malicious activity based on known signatures.
- Correlation capabilities that enable the detection of malicious activity by correlating data from distinct data sources.
- Discovery and monitoring of important assets within the network.
- Promote interoperation with external organisations

3.2 Event collection, processing and storage

The process of collecting, parsing, processing, transforming and storing a log/event is a fundamental operation in a SIEM solution. Therefore, the first step to be able to develop a SIEM that will comply with the above requirements is to define and set up an event collection, processing and storage pipeline.

It is important to select tools that are stable, well-established and supported. These tools need to provide extensive support for collecting events from different data sources, processing them effectively, normalizing the information into a standard format, and storing it efficiently in optimized databases that facilitate rapid event retrieval for event analysis.

To this end, multiple data shippers and time series databases were analysed for their overall capabilities, performance, integration, and usefulness. In the case of the data shippers, the following tools were analysed: Beats, Telegraf, Fluentd and Promtail. In relation to the time series database, the following were analysed: Elasticsearch, InfluxDB and Loki.

In the end, the (E)lasticsearch and (L)ogstash from the ELK stack, alongside the Beats data shipper, were the chosen tools for the event collection, processing and storage pipeline of the Argus SIEM due to their native integration and capabilities that will be explained in the next sections.

3.2.1 Beats

Beats is the selected data shipper of this system, responsible for collecting operational data from various data sources, including network traffic captures, metrics, application logs, firewall logs, IDS logs, and other relevant data that can be captured and used in the detection of malicious behaviour in the industrial environment [11]. The selected data is then extracted from the source by using different inputs and modules.

Modules cover the log collection of multiple essential data sources such as system, firewall, and application logs, as well as cloud service logs (Zscaler, Okta) and events from supported proprietary appliance

devices like Cisco, Fortinet, CrowdStrike, and others. The most important ones for the industrial environment use case will be the Zeek and Windows events data sources. Each module has all the necessary information to collect, parse, and process the logs generated by a given application. This information includes the original log location, the ingest pipelines that dictate how the log will be parsed, what fields will be created, and their data types, which allows one to start monitoring multiple supported data sources in a matter of seconds without previous knowledge.

Similarly, inputs allow, among other things, to monitor any file or custom log in a system, supporting any custom or proprietary application not covered by modules. Additionally, they support the collection of logs received by an Hypertext Transfer Protocol (HTTP) endpoint, syslog, and others. Inputs are a manual option, requiring an additional definition of how the logs will be processed.

Furthermore, Beats comprises multiple sub-tools, each developed for different kinds of data. For the requirements of this SIEM, two sub-tools were chosen: Filebeat, responsible for collecting Zeek logs from the monitoring station, and Winlogbeat, which will be installed in every Windows machine present in the operational network [15] [22]. Beats ensures that the collected logs are delivered to the defined destination at least once without data loss.

As such, Beats incorporates many crucial and important features like flexible log collection capabilities, native support for multiple systems and cloud software, and more importantly, a vast collection of ready-to-collect data source modules and easy integration with Elasticsearch.

3.2.2 ECS Schema

The Elastic Common Schema (ECS) is an open standard specification that defines a common set of attributes for different data types, like logs and metrics, originating from various data sources [12]. It normalises event data, producing consistent logs with predictable fields. As a result, it is possible to effortlessly group and correlate data from distinct data sources, simplifying the log analysis process.

Figure 3 illustrates the benefits of using ECS when searching log events from disparate data sources containing identical data types stored in different log fields. This example showcases three event logs from distinct data sources, each providing information about the source Internet Protocol (IP) address relevant to the event.

In contrast to the raw events shown in the first scenario, where network information is scattered across three separate fields, the second case demonstrates the advantages of normalisation using a common schema. This standardised approach ensures that the same field contains the same type of information across all data sources.

The positive impact of this normalised schema is evident when constructing search queries, as it facilitates the analysis and correlation process.

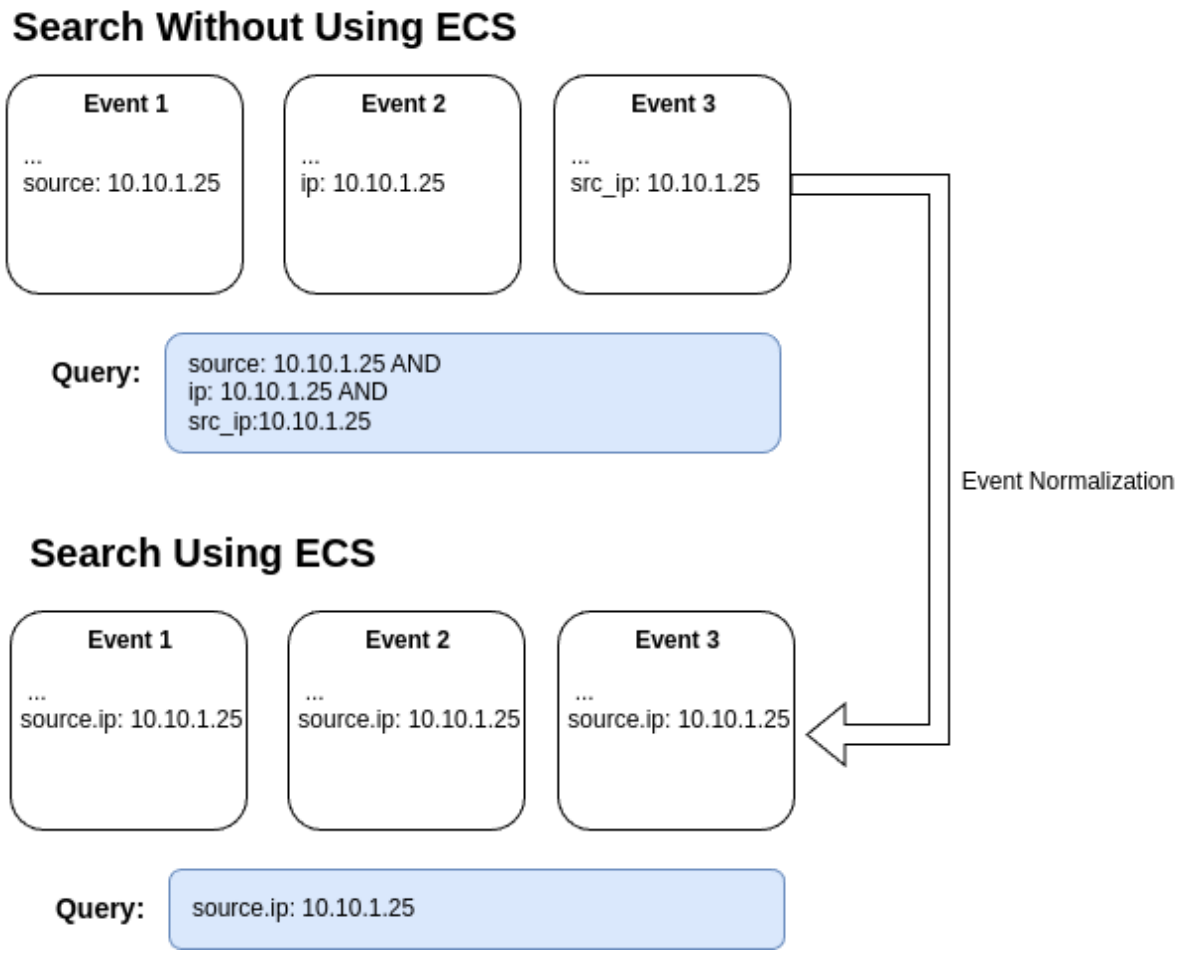


Figure 3: Event search with/without ECS

Naturally, Beats provides native support to ECS, allowing users to use other schema formats if they choose. For the above reasons, the adoption the ECS schema in the log processing pipeline was an important choice as it greatly enhances and streamlines the event analysis process.

3.2.3 Logstash

Logtash is an additional tool in the processing chain, as it has advanced log enhancement capabilities not supported by Beats, allowing it to enhance the final log that will be indexed in Elasticsearch [17]. One example is the possibility of joining data originating from different sources before the log is sent to the final destination.

The Logstash pipeline consists of inputs, filters, and outputs. Inputs allow Logstash to receive event data from multiple selected external data sources. In the case of this system, Beats is the main data

source, but it is also possible to collect data sent directly via Syslog, cloud apps and services, TCP, HTTP, and others.

Filters are responsible for processing and modifying events per the defined conditions. As such, using filters makes it possible to process unstructured log data, remove, rename, substitute, modify attributes, and many other operations.

3.2.4 Elasticsearch

Elasticsearch is the selected technology to store all the logs and metrics collected and processed by Beats/Logstash. It provides extensive advanced search and analysis capabilities for historical data and meets all current requirements for cloud application use, like clustering support, scaling, and high availability, which makes it a resilient solution, making it an adequate choice for a critical solution such as an SIEM [14].

These advanced search capabilities play a crucial role in the SIEM operation. They are responsible for executing search queries, conducting analysis, and correlating events to detect malicious activities in the monitored systems. These capabilities are essential for detecting threats effectively. Elasticsearch provides multiple query language support, making it a flexible and powerful solution for log analysis. Some query language examples are Elasticsearch DSL, SQL, EQL, Painless, and others. Each query language has unique advantages that can be leveraged in specific types of analysis. These language-specific advantages will become clear in section 4.3.1, where different system rule types take advantage of specific query languages.

3.3 Network monitoring with Zeek

As stated before, network monitoring is a fundamental element in the pursuit of securing an industrial environment because, from a network architecture perspective, there have been major changes over the years in how industrial networks are structured due to integration with IT and the adoption of IIoT. To better protect these networks, we need to understand them, specifically the different IT, ICS, and IoT communication protocols that they support. This task requires technologies that can handle massive amounts of data generated every second by industrial environments, as well as their IoT and IT integration, and at the same time, providing support for the various network protocols used in three domains. Additionally, due to the specific requirements of an industrial environment, the network monitoring solution must be passive, so they can not actively modify or block any network data flow, like an Intrusion Prevention System (IPS)

would, as it could result in operational losses or physical accidents.

With these requirements and constraints, Zeek was the network monitoring tool of choice. Zeek, formerly known as Bro, is a passive, extensible, open-source network security monitoring tool. It has the capability of creating rich and organised high-level events of the network activity, creating better visibility and comprehension of the monitored network. It was developed for high-volume traffic networks like industrial networks, it can run on commodity hardware, and it can achieve high performance by supporting clustering, which distributes the traffic analysis for the various available nodes.

Zeek has support for the large majority of the main network protocols used in modern networks. In the case of an ICS/OT network, it supports 4 of the 5 industrial network protocols enumerated in 2.1.3, and it also supports the IoT MQTT protocol and pretty much every IT-relevant protocol. For this reason, Zeek alone can satisfy the network monitoring requirements of a state-of-the-art industrial network that incorporates the ICS, IT, and IoT domains.

Finally, Zeek has other relevant features like an advanced scripting language that can be customised and extended as needed, providing a way to analyse traffic and apply policies or actions based on the defined logic or even generate additional logs that will be analysed later. Zeek can extract files that have been transferred across the network and also supports a signature-based detection approach, just like Suricata and Snort, by using pattern-matching features to detect known malicious traffic strings, like a specific HTTP header, or find specific strings within an extracted file.

3.4 Deployment Architecture

This section presents a deployment architecture of the Argus SIEM, including the different tools and data sources that should be used to ensure the system's detection capabilities in an ICS/OT environment.

Figure 4 showcases a simplified ICS/OT network architecture based on the Purdue model, seen in section 2.1.2, and on a real environment to better showcase the possible implementation of the SIEM. This architecture contains two networks: the DMZ and OT network. The DMZ network can be accessed from an external individual by authenticating through the enterprise Virtual Private Network (VPN). This network contains some essential devices like a data historian, which stores information about the operational process, and a jump machine that provides authenticated users access to the OT network and their systems. Although a simplified version, it is a common network architecture adopted by organisations that need to provide remote access to external providers and operators that require access to the OT systems for maintenance and other purposes.

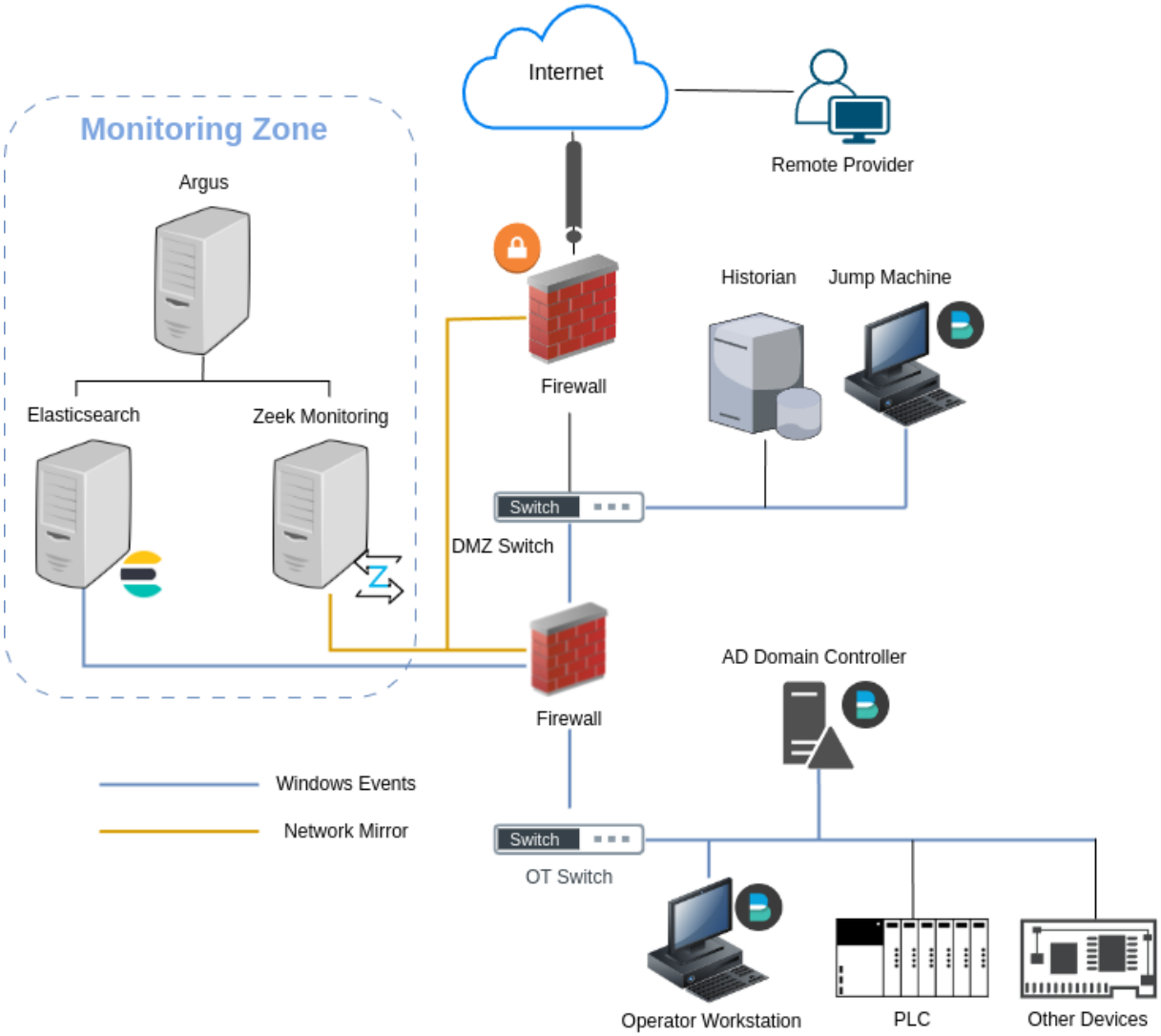


Figure 4: Argus internal deployment architecture

A local monitoring zone was created to deploy the Argus SIEM to secure such a network. In this version, the monitoring security suite is part of the internal network, but it could also be located in the cloud, like many other security solution providers. This zone comprises three machines, each running a different component of the security system: a Zeek monitoring instance, an Elasticsearch/Logstash server, and the Argus SIEM. As shown by the flows coming out of the ICS/OT network to the monitoring zone, two data sources are collected from ICS/OT network: Windows events and network traffic. The Windows events data source comes from the two individual Windows machines in the DMZ and OT networks and the Active Directory (AD) domain controller. Once collected by the agents installed in these Windows machines, these events will be forwarded to the Elasticsearch server, where they will be processed and stored. The network traffic comes from a traffic mirror established in the two firewalls that sit between the WAN/DMZ networks and the DMZ/OT networks, providing complete coverage of the internal network.

These events are then redirected to the Zeek monitoring instance.

Furthermore, additional data source integration could be added to the monitoring system to enhance the data coverage and the potential correlation of the SIEM. For example, the firewalls typically used in industrial environments are high-quality equipment with advanced capabilities that generate helpful additional logs. Additionally, it is common to find IDS running in these environments, whose logs and alerts can be collected. Finally, threat intelligence data sources can also be included as they provided IoCs of known and emergent threats.

Chapter 4

Argus

This chapter presents the developed solution, Argus SIEM, which aims to help detect intrusions in state-of-the-art ICS/OT environments. This was accomplished by developing a solution tailored to address ICS/OT requirements and challenges as its core foundation. This approach allowed the development of a set of submodules that enhances an organisation's capacity to detect and respond effectively, thereby protecting them from APT attacks. A detailed description of the detection system and the reasoning for each decision is enumerated, including the various submodules.

4.1 Overview

The Argus SIEM is a rule-based engine composed of multiple internal submodules that work together to create a unified system. It was built and developed with IT and ICS/OT environments in mind, providing detection and analysis capabilities that can help security professionals and organisations to better protect and defend their networks.

At its core, like any SIEM system, the main goal of this detection system is to analyse logs and metrics coming from different data sources within the organisation's environment to detect threat behaviours that will be reported and signalled to be reviewed by the responsible security operator that will further analyse the security events and act accordingly.

For this purpose, different submodules and utilities were built into the system, providing better visibility, detection and analysis functionalities like the asset discovery and monitoring module, strong event correlation, and post-alert analysis utility, respectively.

To provide some examples, to achieve better visibility into the organisation's network, the asset discovery and monitoring module was essential, as it displays all the monitored network information easily and comprehensively, allowing security professionals to visualize the network interactions and behaviour. To enhance the detection of this system, special care was taken to achieve one of the most important

features that modern SIEMs must have, which is an intense event correlation analysis capability. The post-alert analysis utility aggregates all available information related to the security alert, including all the events, network and host information, past occurrences, and other important contextual information.

To implement some of these features and many others, the SIEM relies on the Elasticsearch search engine to perform many data analysis operations.

4.2 System Architecture

Based on the monitoring and detection system requirements identified in section 3.1 and the event collection, processing, and storage technologies enumerated in section 3.2, a system architecture has been designed to achieve a set of capabilities to address the ICS/OT threat detection requirements.

The system architecture, illustrated in figure 5, consists of two main components: the event collection, processing, and storage pipeline and the Argus SIEM, which comprises multiple sub-modules and a web management console.

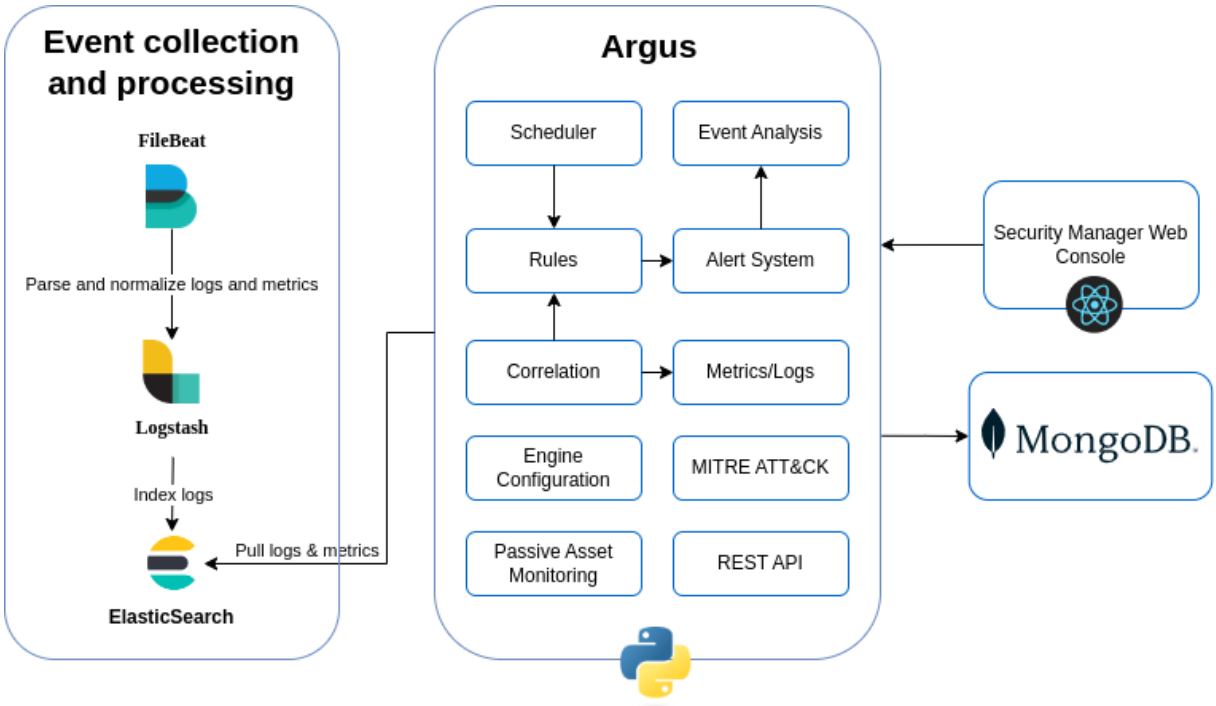


Figure 5: Argus architecture representation

The event collection, processing, and storage pipeline uses the technologies identified in section 3.2. Beats and its sub-tools are configured and installed in the target assets to enable the data collection from the pre-selected data sources. A set of pre-defined processing and normalization operations are applied

to the gathered data, which is then passed on to Logtash. Logstash receives this data, enriches it, if necessary, and then forwards it to an Elasticsearch instance, where it will be stored.

Once the data collection pipeline is established, configured, and running, the Argus SIEM engine will use the gathered logs and metrics stored in Elasticsearch to detect malicious threats in an ICS/OT environment by using different strategies and methods, but fundamentally taking advantage of the advanced search and analysis capabilities of Elasticsearch.

As stated before, Argus contains a detection rule engine in its core, so it has different types of internal rules that define the conditions that signal an event, or a set of events, as a malicious threat. The supported analysis methods include simple detection of known malware signatures and behaviours, exceeding predefined metric thresholds, or more advanced correlation analysis of events from multiple data sources within a given time frame. Argus also supports Sigma rules, which are converted to a structure supported by the engine. The engine scheduler then executes these rules according to the defined execution parameters of the rule.

Beyond the user-defined rules, created to detect specific behaviours and attacks, the baseline module allows the creation of automatic system rules based on the regular activity and behaviour patterns that exist within the ICS/OT networks at many levels: common asset communication channels, protocols, operational behaviour, among others. By creating a baseline of what the ICS/OT network activity looks like, it is possible to detect deviations that could result from malicious activity inside the network.

Once the engine rules detect a malicious threat, the alert subsystem will issue an alert or alarm, depending on the defined security threat level. The security operator can then review the alert/alarm, which includes contextual information, like the detection source, the involved assets, the events that trigger the threat, the MITRE ATT&CK associated tactics and techniques, and the pre-defined policy actions that must be executed. Upon his review, if deemed valid, the security operator can close the alert/alarm as resolved, but only after he has completed the execution of the pre-defined policy actions that was stated in the alert. Alternatively, if he sees fit, he can tune the rule's original attributes and conditions to more precisely and effectively detect the target-specific behaviour in the future.

As mentioned previously, Argus integrates with the MITRE ATT&CK framework, supporting both the ICS and the enterprise matrices, allowing rules to be mapped to the tactics and techniques defined in the framework. This integration provides useful information that can be used to show an overview of each current matrix Tactics, Techniques, and Procedures (TTP) coverage provided by the detection system.

Additionally, Argus also has an asset monitoring module, which enables passive discovery and monitoring of the internal assets distributed by the internal organisation network, allowing the cataloguing of

assets and, at the same time, providing important and useful contextual-asset information like the available data sources, network protocols used by the asset, and all the asset-specific events and the alerts/alarms that it is involved with.

Finally, it also provides a web administrative console from which the security operator can access and manage all the above system capabilities and modify them at his will.

4.3 Rule Engine

The rule engine is the heart of the SIEM, comprising various elements such as the scheduler, rules, and alert system that collaborate to fulfil the primary objective: detecting malicious threats and behaviours.

Rule engines allow for the evaluation of the logical conditions defined within a rule and based on the provided data. These conditions can be simple (matching a specific event) or with more complex logic, such as with multiple AND/OR condition statements. In the context of malware detection, these conditions describe the patterns and behaviours of malicious threats that must be detected and defended against.

After the rule evaluation, the rule engine executes a set of actions depending on the evaluation result. In this case, if the rule conditions are met, it means that a malicious threat has been successfully deployed within the organisation's environment, and so, the security responsible personnel must be warned by triggering an alert that can be quickly viewed and analysed, containing all the necessary information for post-incident analysis.

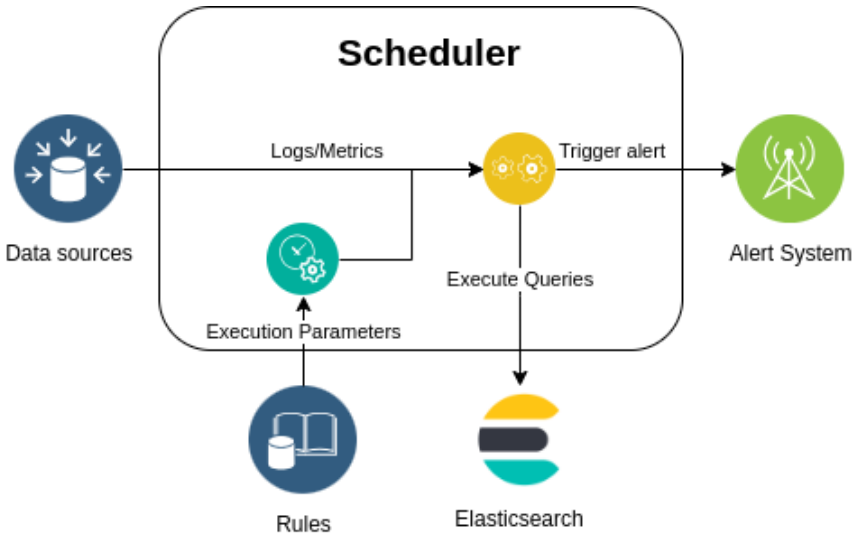


Figure 6: Rule engine execution representation

As illustrated in figure 6, three different components were created to build a rule engine: a scheduler, an alert system, and the different types of rules (explained in the next section). The scheduler is responsible for managing and ensuring the rule execution according to the execution trigger defined in the rule itself. Each rule has a set of parameters that influence the rule execution and evaluation. This can include filters, groupings, and conditions. Then, if the rule conditions are met, the alert system will trigger an alert or alarm, depending on the security event risk defined in the rule.

4.3.1 Rule definition

A rule consists of a set of attributes that describe and contextualize it and a set of parameters that define how the rule is executed and which conditions must be met to trigger an action, in this case, the generation of an alert or alarm. For the system correlation goals, two different types of rules were created, each with specific characteristics and parameters.

Table 1 lists the base rule structure, which includes the common attributes and parameters that each rule in the system has.

Aside from the self-explanatory `name` and `description` attributes, a rule has a base risk score, ranging from 1 to 10, depending on the severity of the security event underlying the rule. A higher value means a higher security severity. In the SIEM, this range is classified into 3 different severity levels: low, ranging from 1 to 4; medium, ranging from 5 to 7; and high, ranging from 8 to 10.

The `datasource` parameter defines the data sources that will be used to detect the target security event, which makes it an important correlation parameter since it allows the remaining parameters to be applied to a vast set of distinct data. On the other hand, the `timeframe` parameter sets the event's temporal analysis range, allowing the precise time selection of the events that will be analysed. For example, if the `timeframe` parameter is set to "10m", every time the rule is executed, it will retrieve all the selected events from the last 10 minutes.

The `trigger` parameter defines when a rule is executed. Two types of triggers were created: a periodic and a rule trigger. The periodic trigger sets the rule execution periodicity, which means if set to "10m", the rule will be executed every 10 minutes. On the other hand, the rule trigger determines that the rule execution is dependent on the generation of an alert/alarm from another rule in the system. This is another important correlation parameter, as it allows rule chaining, where each rule is only executed if the previous rule in the chain has successfully detected a security event.

The last base parameter is the `rule type`, and as the name implies, it defines the type of rule that

Name	Type	Description
name	attribute	Rule summary
description	attribute	Describes what the rule aims to detect
risk score	attribute	Pre-defined risk score associated with rule (1-10)
datasource	parameter	Data source of events used by the rule
timeframe	parameter	Determines the time range frame which events are subject to analysis
trigger	parameter	Type of trigger that fires the execution of the rule. Can be periodic or triggered by another rule alert
rule type	parameter	The type of rule. Can be a threshold rule, EQL, or system rule
ATT&CK tactics	attribute	Associated ATT&CK tactics with the security event that is being detected
ATT&CK techniques	attribute	Associated ATT&CK techniques tactics with the security event that is being detected
note	attribute	Internal note for operators
policy action	attribute	The action defined by the security policy of the organisation that should be executed if an alert/alarm is triggered

Table 1: Basic rule attributes

will be used to detect the target behaviour. Two rule types were created: a threshold rule and an EQL rule. Both have particular characteristics and use cases as they use distinct query languages and define different kinds of conditions, as explained below.

The rule creator can use the MITRE ATT&CK tactics and techniques attributes if he knows which TTPs are related to the monitored security event. As will be shown in later sections, this helps to keep track of the MITRE ATT&CK coverage provided by the Argus SIEM.

Finally, the `note` attribute provides organisational context to the rule. For example, if the rule detects remote logins outside business hours, the note would contain the organisation's business hours range, as well as other kind of organisation related information. The last attribute, the `policy action`, describes the action/s defined in the organisation's security policy that should be performed when an alert is issued. Following the last example, after attempting to access the system remotely, the user's network access may be blocked as a valid policy action.

Threshold rule

A threshold rule is supported by a query language called Domain Specific Language (DSL), which provides extensive search capabilities through compound and expensive queries [19]. It allows multiple types of aggregations, like calculated metrics and document field aggregations. In this way, the main use for threshold rules is to monitor different metrics values, triggering alerts when the limits defined by the rule are surpassed.

As listed on table 2, a threshold rule comprises three main parameters: `filters`, `groupings`, and `conditions`. A filter lets one choose which events from the selected data sources will be analysed. Two filters were created: a simple filter and a script filter. The simple filters allow the selection of events with an attribute equal to or different from a specified value, supporting multiple alternative values. The script filter uses the Painless script language designed for Elasticsearch, and it allows the system to select events programmatically by applying more complex logic comparisons to a given event [18]. These two filter types provide flexibility in the event-filtering process, which is essential for narrowing down particular attack behaviours.

Name	Type	Description
filters	parameter	Selects events by their field values. Can be a simple filter or a Painless script filter.
groupings	parameter	Groups events by the defined fields.
conditions	parameter	Defines with logical operators (AND/OR), the conditions that trigger an alert/alarm.

Table 2: Threshold rule paremeters

As the name implies, the `groupings` parameter groups the filtered events by the user-defined fields. For each resulting group, the defined conditions will be tested, which can lead to multiple alerts/alarms for a single rule execution run. As an example, it allows network events to be grouped by the source address, where the conditions will be tested for each source IP address.

Finally, the `conditions` parameter defines the logical conditions that should be met to trigger an alert/alarm through a set of functions that are applied to metric attribute values. These functions can include the "count" function that counts all filtered and grouped events, the max/min function that retrieves the highest/lowest attribute value seen in all events, and other metric-relevant functions. Then, for each

function, a threshold value is defined. It is possible to define a rule with a single condition or multiple conditions with logic operators (AND/OR). To trigger an alert/alarm, the final result of the logical operation must be true.

Due to its focus on metrics, threshold rules enable the detection of attacks based on metric values. For example, they are crucial to the baseline module capability of to detecting deviations in data flows occurences based on the defined baseline period. They can also be used to detect potential brute-force attacks through by monitoring failed authentication events, abnormal asset resource usage, among many other use cases.

EQL rule

The EQL rule is supported by the Event Query Language (EQL). EQL has an easy-to-read syntax, and it was created to analyse event-based time series data, with security use cases in mind. Besides detection based on IoC, the main feature of EQL is the ability to match ordered and unordered sequences of events across multiple data sources and time periods, which is fundamental to event correlation detection [13].

Unlike the threshold rule, an EQL rule only contains an additional parameter in comparison to the base attributes/parameters table 1. This is the `query` parameter that stores the full query body that will be executed in the backend.

For example, the following text represents an EQL query used in the "Windows RDP Logon Brute-force" rule created for detecting the Remote Services technique in the results section of this dissertation. This query uses an EQL sequence to detect if an ordered series of events has occurred within a 30-second time frame. The sequence of events is grouped by the source IP address(`by source.ip`) and defined within the 2 square brackets. The first one represents multiple failed authentication attempts over the network, that occurred at least 10 times (`runs=10`) in a row. The second event is a successful remote login from the same source IP address. As a result, this sequence can detect when an automatic brute force tool has successfully cracked a given user password.

```
sequence by source.ip with maxspan=30s
  [ authentication where winlog.event_id: "4625"
                    and winlog.logon.type: "Network" ] with runs=10
  [ authentication where winlog.event_id: "4624"
                    and winlog.logon.type: "Network" ]
```

This is just a simple example that uses only one data source. However, when multiple data sources are used as input, the power of the ordered and unordered sequences becomes noticeable, as it is capable

of detecting complex scenarios with multiple operations.

For all the above reasons, EQL rules support a multitude of use cases. They can detect specific known attack behaviours through the use of IoCs, and they are the right choice when the goal is to correlate different types of events to detect a complex chain of behaviours within a given time frame.

4.3.2 Sigma Rules

Creating detection rules for every single possible attack is almost impossible since this process requires extensive knowledge and time. To address this problem, Sigma rules support was added to the SIEM, allowing the bulk upload of existing publicly available rules created by experts, resulting in huge time savings.

Sigma is a popular open standard generic signature format that allows one to easily define the detection parameters for specific attacks by describing what a malicious log event looks like. It has an easy-to-read syntax due to the use of YAML, which reduces the necessary time to understand a given rule. Due to the huge variety of SIEMs in the market, each one with specific detection languages and features, it has become a necessity, especially for detection engineers, to have a generic common format that enables fast collaboration and that can be used to easily share detection parameters that can be quickly converted to specific custom SIEM logic.

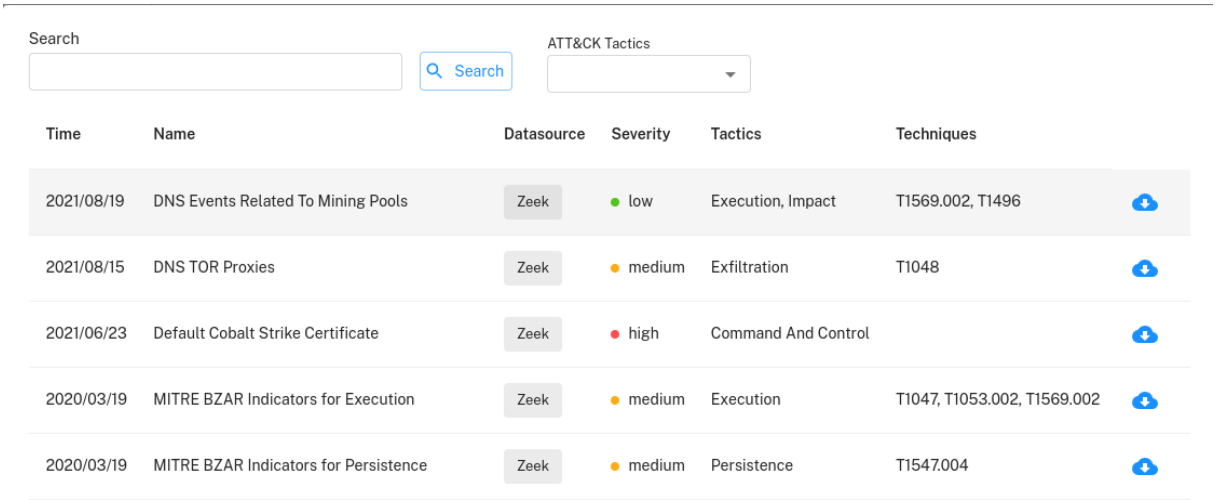


Figure 7: Sigma rule importer

As a result, Sigma rules provide a way to quickly start monitoring well-known threat behaviours. There are a number of public Sigma rule repositories, including the official Sigma repository, with huge amounts of rules covering different kinds of data sources, including Zeek and Windows events logs [59]. As shown in figure 7, it is possible to quickly import the rules located in these public repositories via the sigma rule

importer available in the web console, with the option of adding custom public or private repositories in the management settings.

4.3.3 Event Correlation

Event correlation is one of the most important capabilities that a proper SIEM needs to have. It allows the analysis of sequences of events from multiple sources, according to a set of parameters and conditions, to find correlations between multiple individual events, which when isolated do not seem much or have any value, but grouped and viewed as a whole, can represent malicious threat behaviours.

To be able to correlate events from distinct data sources, all events need to be normalized in a common and unified format. Without normalization, it would be impossible to correlate 2 events that have identical data attributes but that have different formats and structures. As seen earlier, this problem is solved in the event collection and processing pipeline, where the ECS schema is used to normalize the events with common attributes that can be grouped afterward for analysis.

With these events normalized, the next step is to use the system tools to correlate events. The Argus SIEM contains two functionalities that provide correlation capabilities: rule chaining and EQL rules.

Rule chaining, as the name implies, is the ability to create an execution sequence of multiple rules. Each rule in the chain is only executed if the previous rule output was evaluated as true, which in this case means that the previous rule has triggered an alert/alarm. This is possible due to the the flexibility implemented by the rule trigger, that was presented in the last section. This trigger states that a rule will only be executed if the rule that is dependent on issues a new alert/alarm. This brings flexibility in the correlation process, since it supports any rule type present in the system, even future ones. As a result, a security operator could create a set of rules, each one detecting a specific behaviour and dependent of one another, all working to detect a wider attack scenario that is only possible to detect when monitoring multiple indicators across many data sources.

The other correlation mechanism is the EQL rule itself, that was explained in detailed in the previous section. In summary, it is ordered and unordered sequence matching allows the correlation of multiple events, across-multiple data sources, which makes it a powerfully correlation tool.

4.4 Alert System

When the conditions of a rule are met, the SIEM triggers the creation of an alert or alarm via the alert system. This component was designed to help security operators track and analyse security incidents

identified by the SIEM by providing a simple interface that combines all the alert-related information, which is displayed in a concise way.

First, it is important to make a distinction between an alert and an alarm. The distinction comes from the original rule risk score, as an alert comprises low and medium severity levels, which correspond to the 1-4 and 5-7 risk score ranges, and an alarm corresponds to high severity incidents, which includes the highest risk score range (8-10). Additionally, the SIEM has the ability to raise the base alert risk score if the security event was detected multiple times in recent history. Alert repetition over short time frames could be a result of persistent actions of a threat actor, which requires the utmost attention of security operators, and so by incrementally increasing the alert risk score, it would eventually reach an alarm status, requiring immediate action from the security personnel. Once these security events are analysed and resolved, the SIEM resets the rule risk score to the original base score.

Furthermore, it is important to understand the different attributes that form an alert/alarm. Besides the already mentioned risk score and type, an alert/alarm can have different statuses: open, when it is first created and requires evaluation by a security operator; processing, when it is currently being analysed by someone; and resolved, when the evaluation was completed and defined policy actions performed. It also contains various types of data, including the date and time of detection, rule-related information, and all the original events that trigger the alert/alarm, which can then be analysed more thoroughly by a security operator. Finally, it also provides contextual information such as the associated ATT&CK tactics and techniques, security policy actions, and the assets that are involved in the security incident.

4.4.1 Alert Response Model

Although the SIEM does not have automated response capabilities like those available in SOAR solutions, the security operator still needs to analyse and react to the numerous alerts and alarms that are issued by the SIEM. This analysis/response process is illustrated in the model present in figure 8.

Firstly, there is a normal state that is preceded by the trigger of an alert/alarm when the conditions of some rule are met. At this moment, the security operator receives a notification in the web console of the existence of a new alert/alarm. He or some other security team member starts analysing the alert/alarm by using the different tools that will be shown in the next section to understand the root cause of the security incident and to conclude whether the security incident is valid or false-positive.

If after the analysis, the security operator comes to the conclusion that the alert/alarm successfully detected a malicious behaviour in the organisation's network, he will then perform, if they exist, the pre-defined policy actions. After that, since the original rule is working as intended, no additional actions will

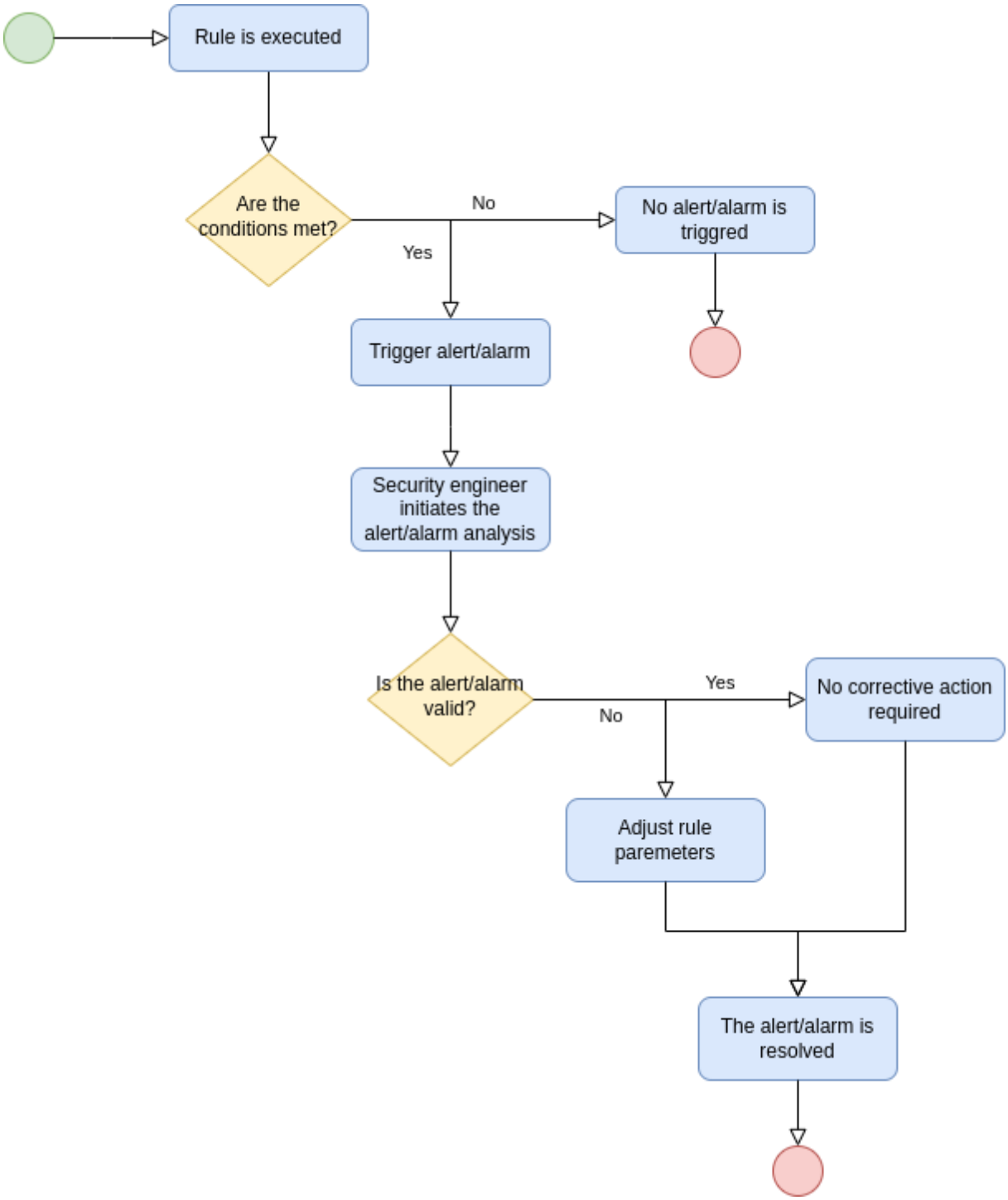


Figure 8: Alert response model

be performed, and the alert/alarm will be resolved, terminating the response process.

On the other hand, if the alert/alarm is considered a false positive, the security operator may need to re-evaluate the original rule parameters. The process of rule tuning is essential for to enhance the detection effectiveness of the SIEM because continually improving and adjusting the different system rules parameters ensures that, in the future, fewer false positives will be issued, which increases the percentage

of valid security alerts produced by the SIEM, resulting in the optimization of the security operators time, as they will spend progressively more time analysing security alerts that really require their full attention and knowledge.

4.4.2 Alert Analysis

In the web console, a security operator has access to all alert-related information about a given security incident, allowing him to assess each incident and make informed decisions. On the alert review page, it is possible to find the following alert-related data:

- **Contextual information** - This includes alert metadata, information about the rule that triggered the alert, and contextual information, such as all the assets that were involved in the incident.
- **Events** - Displays all the original events that triggered the alert. This includes the full event/log data. This helps the security engineer begin immediate the alert review process because he does not need to waste time searching for the security incident root events, which usually contain additional information that is essential to incident analysis.
- **Intelligence** - Display the ATT&CK alert-related information, including the TTPs associated with the alert, alongside with internal organisation information, such as the security policy actions that should be performed.

Additionally, security operators can also use the profiler utility, which is used in the what-if analysis.

What-if Analysis

The what-if analysis, powered by the profiler utility, provides historical information about the rule that originated the alert. It returns different outputs for each rule type available.

In the case of the threshold rule, the profiler provides a comprehensive temporal view of the behaviour and patterns of the specific monitored metrics, as seen in figure 9, allowing one to quickly identify outliers that do not fit on the normal behaviour constraints and limits, allowing the precise tuning of rule conditions thresholds, and consequently, the reduction of the the number of false positives produced by the engine.

In the case of EQL rules, the profiler works more as a historical lookup, as it presents all the moments in the past when the specific malicious behaviour has occurred. This feature is particularly valuable for newly established rules, as it provides insight into past events that went unmonitored and therefore unreported and analysed. As a result, previously unknown security incidents may be uncovered.



Figure 9: Profiler metric visualization

4.5 Asset Discovery and Monitoring

As stated in section [65], asset discovery and monitoring are fundamental to the risk assessment process of ICS/OT environments. The goal is to collect as much accurate and up-to-date asset information as possible, including the asset product model, vendor data, network information, firmware and software versions, information about the running operating system and the applied patches, and many others. This type of asset-related information is crucial to correctly execute other security-related processes, such as patch and vulnerability management.

Naturally, this type of data can be of great help for detecting certain types of attacks in IT and OT environments, such as unexpected firmware or configuration changes, both of which can be monitored to ensure that the correct firmware and pre-defined configuration options are being used, preventing manipulated firmware versions and incorrect, or even malicious, configurations being run.

Asset monitoring also provides additional context to events and alerts, for example, if an alert is triggered on a critical asset, it should have priority over alerts that were triggered in common assets, as it implies a higher impact and risk to the OT environment, and as so, requires a much faster response from security operators, before any impactful or destructive action could be performed.

It can also be used to visualize the current IT and OT assets that are present in the organisation network and discover newly added assets.

For these reasons, creating a passive asset monitoring module within the Argus SIEM was an obvious step as it provides important asset and contextual environment information of all assets present within the organisation network.

4.5.1 Passive Monitoring

Automated active asset discovery and monitoring can negatively impact an OT network by inadvertently congesting it or by disrupting an error in one of the assets that is being scanned or interacted with, which requires extensive testing and evaluation before deploying such a solution.

For this reason, the basic asset discovery and monitoring module in Argus works by passively monitoring and analysing the different available data sources, especially the network monitoring data sources like Zeek, Snort, and proprietary firewalls. With the logs and events provided by these network solutions, it is possible to map the internal IT and ICS/OT network by analysing the various communication data flows.

Because the event collection and processing pipeline uses the Beats log shipper, which in turn uses the ECS schema to normalize events, it becomes possible to collect network data flow information, which includes all the IP addresses present in the internal network. This information can be extracted using the ECS `source.ip`. `destination.ip` and `related.ips` attributes, available in every network-related log or event. Once collected, this information is used to create base assets, each one with one or multiple interface IP addresses.

Using the Zeek data source ensures that all data flows within a network are captured since the Zeek `conn.log` log file stores all general information about each flow recorded in the network, which includes the source and destination IP addresses and ports, the protocol used, request and response bytes, connection duration, and others.

Additionally, by using optional Zeek scripts, it is possible to gather additional information not mentioned above. For example, the `mac-logging` Zeek script attaches to each `conn.log` record the source and destination Media Access Control (MAC) addresses, which will then be processed and stored in the

`source.mac` and `destination.mac` ECS attributes. With the MAC address of each asset, the vendor of that particular device can be automatically retrieved using the MAC address prefix (the first three octets), known as the Organizationally Unique Identifier (OUI), in lookup databases. Furthermore, the `vlan-logging` Zeek script provides VLAN information for every connection log, which can be used to visualize and verify the segmentation of the network.

Another relevant ECS attribute group is the `host` attributes. Host attributes define and describe a computer instance, as they are present in every log collected by a Beats shipper. It contains detailed information about the host where the Beats agent is installed, including the host IP and MAC address, operating system information, host architecture, hostname, and others. For example, in a Windows system running the Winlogbeat log shipper, it is possible to collect information such as the operating system version and build (Ex: Windows Server 2019, Build: 7763.737), and even information about the Windows Active Directory domain that the machine is part of.

All of this information can be used to create a basic asset discovery and monitoring module that can be used to track the organisation's asset inventory and provide support and context for the remaining modules.

4.5.2 Usage

In the web console, a security operator can see all the available assets that are part of the internal organisation network, as shown in figure 10. For each asset, it shows basic information, such as the asset name, description, vendor, Operating System (OS), IP and MAC addresses, the available data sources, and if the asset is managed or not (has a Beats agent installed). Except for the IP and MAC address, the remaining attributes are customizable, like the asset name and description, since it is not always possible to retrieve the necessary information using passive monitoring. This customizability allows a security operator to tag each asset according to his knowledge of the internal network, which can be helpful when no dedicated asset monitoring software is used in the organisation.

Beyond the above primary attributes, each asset also contains contextual information for the SIEM, like the alerts that the asset is part of and all the events collected from or associated with that asset. In this last case, it is even possible to execute custom inline queries against the asset events if it is necessary to quickly filter and analyse a particular type of event associated with that asset.

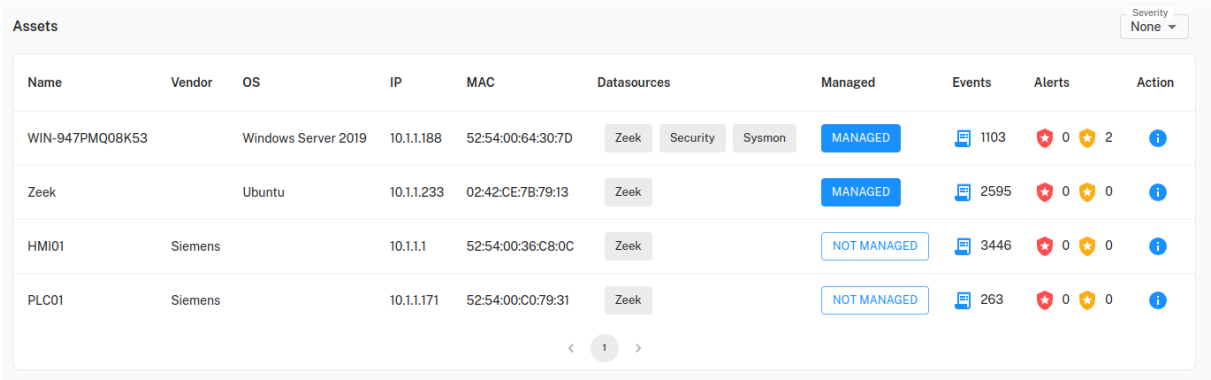


Figure 10: Asset discovery and monitoring dashboard

4.5.3 Limitations

By relying on only passive collected data, much of the essential information that dedicated active asset monitoring solutions gathered is missing, such as firmware and software versions, configuration changes, etc.

To address this, in the future, the asset discovery and monitoring module should be integrated with third-party active asset monitoring solutions since they provide more complete data about each asset that can be used to provide additional information to the SIEM.

Another solution could be the addition of new capabilities to the asset discovery and monitoring module, more precisely, the ability to extract asset information from the communications between assets using ICS protocols.

4.6 Baselining

The baseline process is a crucial mechanism to protect ICS/OT environments as it establishes the normal state and behaviour of the environment, which includes all the assets, network communications patterns, and different configurations. Essentially it defines the expected behaviour of different components of the ICS/OT environment, ensuring consistent execution and monitoring to know if the components behave the way they are supposed to. This process is perfectly compatible with the way that industrial environments work, where the operational process and all the assets and devices that support it, have a consistent operation behaviour that can be monitored.

By being able to baseline the ICS/OT environment at different levels, it becomes possible to detect deviations by comparing the current behaviour to the expected state, resulting in the identification of behavioural anomalies that do not correspond to the baseline behaviour and might be related to malicious

activities.

Despite its benefits, this detection mechanism has a important drawback: the potential for a high volume of false positives, which can compromise the accuracy and usefulness of the SIEM alerts. False-positives could be of a result of sudden, unexpected legitimate changes in the current behaviour of the environment. These changes may pertain to the operational processes, the configuration, functioning, and communication of OT assets, or the actions performed by the operational personnel. Consequently, the baseline-based detections must be extensively tested, monitored, and updated in a continuous fine-tuning process that will enhance the precision of this detection capability.

Due to these advantages, it was crucial to develop and incorporate baseline native capabilities in the Argus SIEM, allowing security operators to quickly create and deploy baseline detections that help identify abnormal behaviour in general or very specific environments.

4.6.1 Analytics

The baseline analytics is the core component of the baseline module. Analytics specify what type of data will be monitored, how to group it, and how to analyse it to build a baseline. For the selection of events that will be analysed, analytics have similar parameters to system rules, such as event data sources, event filters, and field groupings. There are also different types of analytics: general analytics, flow analytics and asset analytics, which will be shown in the next sections.

By specifying exactly the type of data that is monitored, it is possible to create general or very specific behaviour baselines that will detect any deviation in the current state compared to the defined baseline period.

One of the most important abilities of the baseline module is the ability to generate new events from every deviation detected by the SIEM, containing information related to the source analytic and the deviation data. This rich event will be stored in a custom Elasticsearch index, resulting in the creation of the baseline data source, which can be then further used by the system detection rules to correlate with other data sources events. This enables the creation of complex detection rules that take into account baseline deviations detected by the SIEM. This is specifically powerful with asset baseline analytics that can be easily correlated with other network events since the asset IP address can be used to group different events.

Due to the focus on ICS/OT environments, multiple built-in analytics were created to enable the native detection of some crucial ICS/OT-related behaviours without requiring further configuration. These pre-built analytics include general and asset analytics, providing good initial coverage for general use cases.

4.6.2 Data Flows

Data flow baseline analytics track and monitor general interactions between network devices using the `destination.ip` and `source.ip` attributes present in every network-related event. The network data flows are monitored in order to map the communication channels between network devices. This information is valuable as it allows the identification of network communication flow patterns and, consequently, the detection of abnormal network activity. The data flow visualization page, depicted in Figure 11, allows users to visualize their network by showcasing all the connected devices and the communication flows that occur between them.

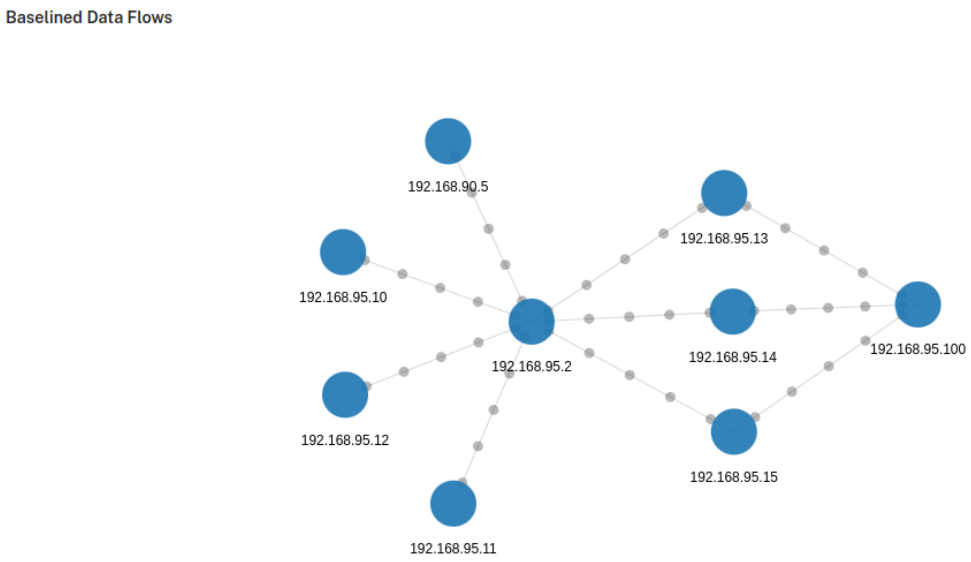


Figure 11: Baseline asset data flows

The baseline module has built-in data flow analytics to monitor common use cases. For example, it is possible to detect when two network devices communicate with each other for the first time, even though the baseline indicates that they have never interacted before, exhibiting, in this way, abnormal network behaviour. This particular detection can be useful for detecting lateral movement in a network since attackers may enumerate and interact with multiple network devices to gain further access to the network.

Another option is the ability to monitor unusual data flows between trusted assets, this means that, in the baseline, these devices legitimately communicate with each other. Because OT devices have a very consistent execution behaviour, it is possible to detect when unusual communication data flows occur. For example, if two devices constantly communicate with each other, within a certain time period, and one suddenly stops sending or responding to network requests, this may indicate a problem with that device.

On the other hand, unusually high volumes of traffic between devices may also be related to malicious activity in the network. To achieve this, once this option is enabled, it automatically creates new threshold rules using the known baseline data flows as the rule limits.

4.6.3 Assets

Asset baseline analytics builds upon data flow analytics as it allows for the monitoring of asset behaviour by automatically grouping the events for each asset listed in the asset monitoring module. Using this baseline analytic any behaviour associated with all assets can be monitored, a specific group of them, or even individually.

Similar to the data flow analytics, the baseline module has a set of built-in asset analytics to cover the most common use cases. From this set, it is important to highlight some analytics. The first one detects when a new asset is added to the network, which can help detect rogue devices that are physically placed in the IT or OT network. There is also analytics related to the network protocols used by each asset, which includes the ability to detect when a specific asset uses a new network protocol that is not part of the trusted protocols listed in the asset baseline, or even detecting when assets use known network protocols but the communication is performed using non-standard protocol ports. Both analytics can be used at the same time to detect abnormal network activity coming from a specific asset.

As will be presented later, throughout the results chapter, multiple custom asset analytics were created for the detection of threat behaviour in different use cases, showcasing the advantages of using this detection method. Some examples include the detection of malicious Windows authentication patterns and, specifically to ICS network protocols, the detection of non-common Modbus functions from OT assets.

4.7 MITRE ATT&CK Integration

One important addition to the SIEM was the integration with the MITRE ATT&CK framework. This is an essential component of the SIEM due to the increasing adoption of the ATT&CK framework in both IT and ICS environments, as well as the contextual information that can be presented, and the value that this framework brings to organisations and professionals by enabling various use cases, enumerated in section 2.6.

4.7.1 Alert Insights

By embedding ATT&CK TTP information in the rule creation process, once a new alert is generated, the security operator responsible for handling that alert will be able to see which ATT&CK tactics and techniques is the security alert event related to. This is information is valuable since it allows anyone who is reviewing the alert to know what kind of attack it is, what to look for, and its possible consequences without requiring previous knowledge about the attack. As such, this additional information enhances the alert-analysis process as it provides important contextual information about the security alert.

4.7.2 Learning Support

Integrating the ATT&CK knowledge base into the SIEM will also allow security operators to access the framework data within the same interface that they use to perform other tasks, such as alert monitoring, analysis and response. This offers an educational opportunity since it supports easy and quick access to all the tactics, techniques, adversary behaviours, software, and more, in both enterprise and ICS ATT&CK domains. As such, security professionals can learn about specific attacks that they probably should be defending against, improving their knowledge of IT and ICS security, which includes understanding how the different attacks work, their implications, how they can be detected and protected against, and how they are being used in the real world.

The ATT&CK knowledge base can be viewed from different pages in the web console, including the rule creation page, the alert review page, and the ATT&CK coverage page. It presents comprehensive information about the most important ATT&CK components, such as tactics, techniques, data sources, detection, mitigations, groups, etc.

4.7.3 Coverage Assessment

Another fundamental advantage of the ATT&CK framework integration is the possibility to continuously monitor the tactics and techniques coverage provided by the data sources that are being collected and the detection analytics running in the SIEM.

This capability is crucial to expand and improve the SIEM detection system's effectiveness because it identifies detection gaps, which are then prioritized based on their impact [35]. For example, in an ICS/OT environment, every technique that could lead to the disruption of OT operations should be first addressed.

For this purpose, the technique coverage page, illustrated in figure 12, was created and inspired by the MITRE ATT&CK Navigator visualization tool [46]. It supports both ICS and enterprise ATT&CK matrices

and allows a security operator to quickly visualize the techniques currently covered by the SIEM system's rules, as well those that are missing.

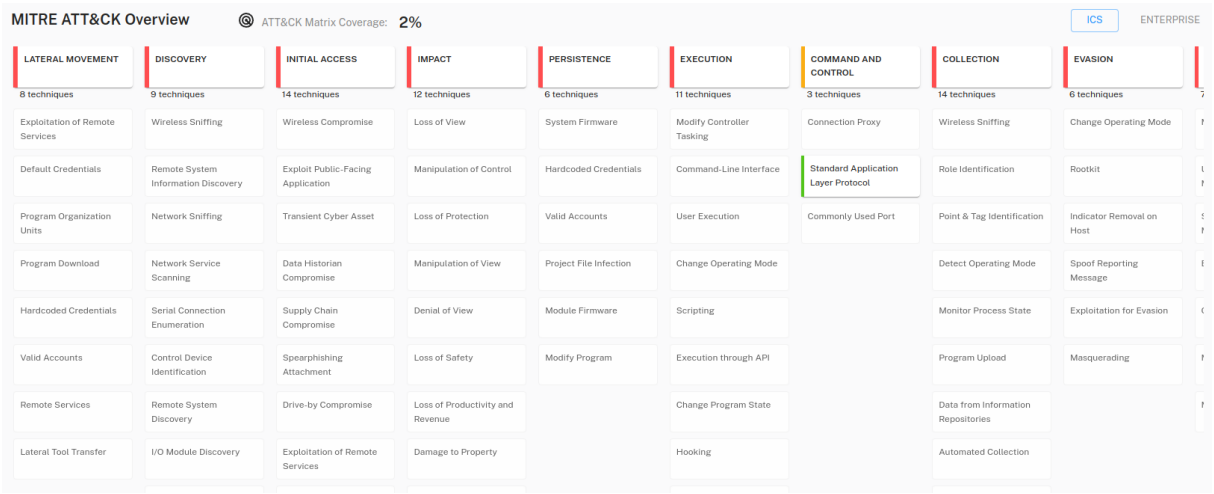


Figure 12: ATT&CK matrix rule coverage

Each technique can have multiple rules associated since many techniques have alternative detection methods that use different data sources. Because of this, it is possible to see which rules in the system cover a specific technique. It is also important to mention the difficulty in assessing the detection accuracy provided by a given rule because a single rule can completely cover the detection of a specific technique and, at the same time, have multiple rules providing partial coverage to a technique. To achieve the highest accuracy possible, it is essential the creation of rules based on the different detection data sources available for each technique.

Furthermore, once the gaps are identified, it is important to understand if the available data sources can be used to create rules that detect the technique behaviours or if it is necessary to start collecting new data from non-monitored data sources. By toggling the data source coverage option, the matrix visualization changes from a detection perspective to a data source support perspective. Depending on the number of data sources referenced in the detection component of each technique, the coverage colour can range from red (no data source available) to yellow (some of the supported data sources are available) and green (all the data sources are available). With the information presented, a security operator can identify the missing data sources that should be collected to support partial or complete detection of the tactics and techniques that he deems important.

Finally, once the necessary data sources are being collected and are available for analysis, the next step is the creation of rules based on known technique detection analytics. The rule creation page has a dedicated section for the ATT&CK metadata, allowing the selection of the tactics and techniques that

are associated to the rule. A single rule can cover multiple techniques from different tactics and across different matrices.

Chapter 5

System Evaluation

This chapter presents the evaluation process to assess the effectiveness and validate the developed system detection capabilities against IT and ICS/OT threats. The baseline module and the rule engine were tested to demonstrate their intended use, and specific use cases were selected to showcase the advantages and limitations of each detection method according to each attack characteristics. For this purpose, an ICS/OT simulation environment was deployed and used to test the detection of the different attack types. All tools, data sources, and configuration files are described and explained for reproducibility purposes.

5.1 Simulation Environment Setup

To evaluate the detection capabilities of the SIEM and since it was not possible to have a real environment with all the required hardware and devices, a simulation environment had to be set up. This platform was designed to be structurally similar to real-world ICS/OT environments. The flexibility provided by the simulation environment allowed the execution of attacks along the attack chain, providing a better view and comprehension of the possible types of attacks present at the different network zones or levels, as explained in section 2.1.2. Concerning the attacks selection, the ATT&CK framework was used to obtain a list of tactics, techniques and procedures that are relevant and impactful in an ICS/OT environment. Each evaluated TTP provides a concise explanation of its purpose and includes relevant real-world examples, accompanied by a comprehensive list of applicable detection methods. A rule/analytic is then created to test the system's detection capabilities based on the available data sources and the known detection methods.

To simulate an ICS environment, an ICS testbed was chosen, which provides, through virtualization, the fundamental ICS components and communication networks that are part of such environment. This provides great support for testing different attack vectors and behaviours, playing a crucial role in testing the baseline capabilities of the SIEM and emulating threat behaviour at the OT level.

5.1.1 ICS Test bed

Test beds can be grouped into different types, depending on their implementation characteristics [24]. These types are physical simulation, software simulation, semi-physical simulation, and virtualized test beds. Since it was impossible to access any real OT equipment, the only choice was to use virtualized test beds.

As the name implies, a virtualized test bed uses virtualization technologies to design and build cheap and reusable components that are easy and quick to deploy [24]. For this purpose, the Graphical Realism Framework for Industrial Control Simulations (GRFICS) test bed was selected. GRFICS virtualizes the OT physical process and the ICS network and devices, including an HMI, PLC, and an I/O layer [23]. The PLC uses OpenPLC, a standardized open-source PLC controller, the HMI uses the ScadaBR software to monitor and present operational measurements collected and forwarded by the PLC, and the I/O layer is represented by a chemical process simulation, whose I/O devices are controlled by the PLC. It also provides a visualization component using the Unity game engine. On the network side, these ICS devices communicate using the Modbus ICS protocol. Furthermore, there are two additional virtual machines: an engineering workstation that can access and modify the PLC configuration and a pfSense firewall that controls access between the DMZ and ICS networks.

It should be noted that completely relying on virtualized ICS environments has obvious drawbacks. The biggest limitation is that it is only capable of virtualizing open-source ICS software which is not used by most ICS devices vendors, like Siemens, Schneider, and others, that run close-source proprietary software.

5.1.2 Test environment

To complete the test bed and to better simulate a modern IT and ICS/OT environment, some changes were applied to the original GRFICS testbed environment. The most important one was the addition of an enterprise layer before the DMZ network, which could contain many of the common enterprise components of an organisation like the enterprise employee workstation, web and mail servers, among other functions. This new layer reflects the changes that the convergence of the IT and OT domains has created, opening a new attack vector to infiltrate industrial organisations and their networks.

As illustrated in figure 13, the virtualized test environment has a similar structure to the Purdue model described in section 2.1.2. It comprises 7 virtual machines distributed by the IT,DMZ, and the OT zones. The IT zone or enterprise network, contains the workstations the employees use, which in this case is represented by a single Windows enterprise machine. Normally, users within the IT network would require

VPN credentials to access the DMZ network, however, for simplification of the attack process, this option was not considered, and instead, firewall rules were applied to only allow IT devices to remotely access the DMZ jump machine, which still requires valid authentication.

The DMZ network contains a Windows jump machine, which can be accessed using the Remote Desktop Protocol (RDP) protocol with the proper user credentials. The jump machine can access the HMI, which also works as a historian and presents the operational measurements provided by the PLC.

The OT network contains a Windows operator workstation that can control the PLC in this network and view the 3D simulation running on the chemical simulation machine. Between the IT/DMZ and the DMZ/OT networks is a pfSense firewall that routes and controls access from devices within the IT network to the DMZ network, and then to the OT network. The VirtualBox hypervisor was used for environment virtualization because the original GRFICS virtual machines were shared as VirtualBox appliances.

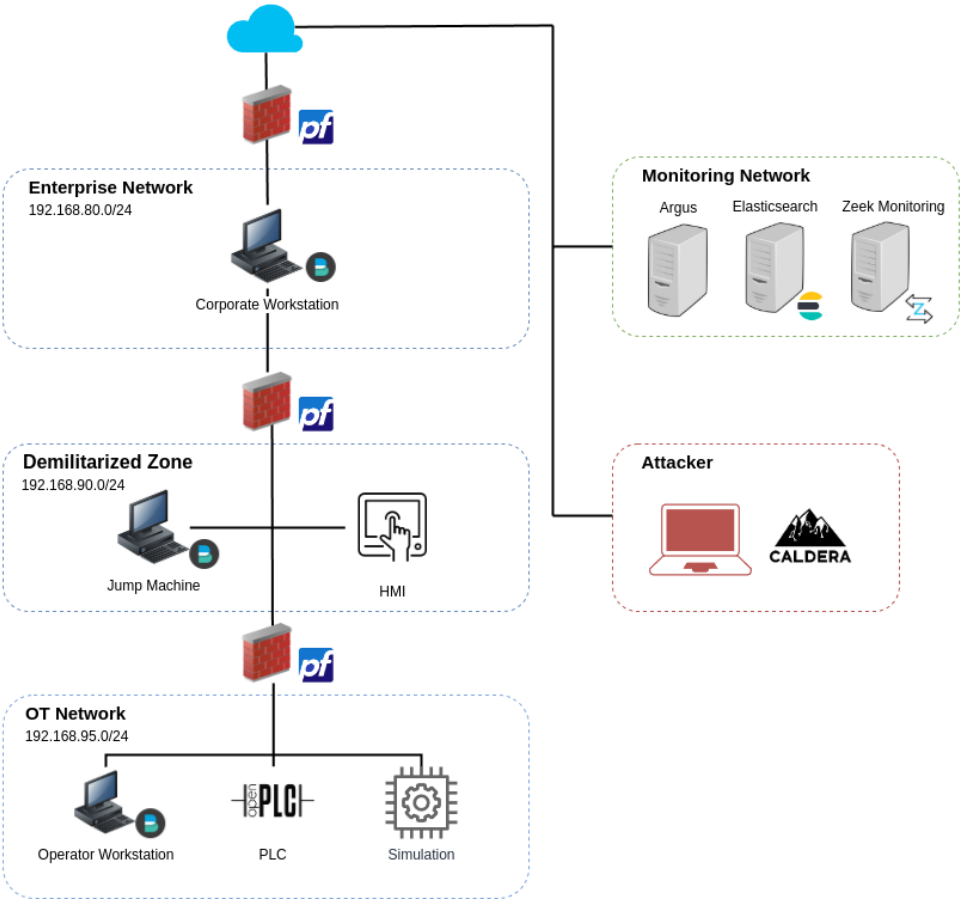


Figure 13: Test environment architecture

Moving on to the tools used in the environment, the Beats data shipper was installed on every Windows machine (Winlogbeat) to collect OS and applicational data logs. The pfSense firewall contains a third interface that works as a Switched Port Analyzer (SPAN) port, which is responsible for mirroring the network

activity in the IT, DMZ and OT networks and sending it to a destination, which, in this case, is the Zeek monitoring instance. This Zeek instance is part of the monitoring network that also includes a Logstash/Elasticsearch stack running on another VM, which receives and stores all the data source events, and a last VM which runs the Argus SIEM, identical to the deployment architecture described in section 3.4. To forward the OS and application logs collected by the Windows clients to the Elasticserach instance, where they will be stored, a secondary network interface was set up with access to the monitoring network.

It is also important to note that the main ICS protocol in this environment is the Modbus protocol, as the PLC uses it to control the various devices in the simulation, as well as to exchange operational information with the HMI and to change the PLC program configuration from the engineer workstation. For remote access, the RDP protocol is used to access the jump machine and the operator workstation, each with specific role-oriented credentials.

5.1.3 Tools configuration

As the above architecture shows, two data sources are collected and fed to the Argus SIEM: network traffic and Windows host events. The network traffic captures are forwarded and analysed by a Zeek node, which produces two out of three data elements listed in the ATT&CK framework: network connections and network traffic flows. The Windows data source provides extensive host event log information, including user and authentication information, process creation, command execution (PowerShell), and application logs.

Starting with the Zeek configuration, Zeek version 6.0.1 was installed and configured on an Ubuntu 20.04 instance. It was configured with a monitoring network interface that receives traffic from the firewall SPAN port. Additional Zeek policy scripts were added, by editing the default configuration `local.zeek` file. This includes the `mac-logging.zeek` script that enables logging of link-layer addresses, and the `json-logs.zeek` script that outputs the Zeek logs in the JSON format, required by Filebeat to properly parse the log attributes and values.

After this, the Filebeat service was installed and configured, including the Zeek module. This module contains multiple file sets, each one corresponding to a specific Zeek log. These are the most important file sets enabled: connection, dnp3, modbus, http, dns, rdp, ssh, ssl, stats, and x509.

Finally, a Caldera server instance was installed and configured [4]. Caldera is a platform that can be used to automate adversary behaviour emulation. It comes with a Command & Control (C2) server used to communicate with the various agents installed on the target systems. It is possible to run specific payloads and commands in these systems without any manual interaction, which comes in handy when

it is necessary to quickly test different exploits and behaviours. The selected Caldera plugins were Atomic (TTP payloads) and Caldera for OT (includes capabilities for ICS protocols).

Regarding the Windows hosts configuration, the first step was to install the Sysmon service and device driver from Sysinternals, which logs extensive host activity information to the Windows Event log subsystem, such as process details, network activity, file and registry modifications, and many more. The installed version was 15.10, which includes 29 different event IDs. To configure sysmon, a high-quality, publicly available configuration template was used [68]. After this, the Winlogbeat agent (version 8.10) was installed to retrieve all Windows event logs. The three available modules were enabled: Security, Powershell, and Sysmon. This ensures that all essential security-related Windows events are continuously collected and stored in Elasticsearch, ready to be analysed. The last step was to generate the Caldera Windows agent, which will be used to automatically and effortlessly launch and simulate the selected attacks, as explained in the next section.

5.2 Test Plan

The SIEM detection capabilities test plan will evaluate how the solution responds to cyber attacks originating from the IT domain and targeting the ICS/OT domain. As mentioned multiple times previously, this is crucial for modern ICS/OT environments due to ongoing IT/OT convergence. As a result, security solutions protecting ICS environments require both ICS-specific and IT detection capabilities, as explained in section 3.1. The ability to monitor and detect malicious activities in both domains ensures increased protection coverage, resulting in a more complete and in-depth security monitoring solution.

The IT attack vectors are predominant in the early stages of a security incident, namely, to gain initial access to the target infrastructure network by exploiting common IT vulnerabilities. This results in access to systems in the IT domain that could possibly have access or privileges to the OT network. Another relevant ICS ATT&CK tactic that relies on the exploitation of IT devices is the Lateral Movement tactic, which describes the different ways in which threat actors can exploit these systems to gain access to the OT network and its devices. Therefore, the test plan is based on a set of TTPs that will be simulated in the test environment to evaluate the detection capabilities of the system. The TTPs that will be selected below will cover both IT and OT domains to properly evaluate how the solution can effectively protect modern ICS/OT environments.

5.2.1 TTPs Overview

As mentioned earlier, two data sources will be used from the simulation environment: network traffic and Windows event and application logs. Based on these two data sources, which correspond to multiple MITRE ATT&CK data source elements, it is possible to detect numerous TTPs, applicable to the IT and ICS/OT domain. To identify relevant TTPs for this use case, both the MITRE ATT&CK ICS and enterprise matrices were used. Because the collected data sources can potentially detect many TTPs, a sub-selection was performed to select the most important and impactful tactics and techniques for ICS/OT environments.

To better decide which TTPs to choose, it is necessary to first understand the most common attack vectors used by real threat actors. For this, the ATT&CK framework provides the Campaign component, which groups real intrusion activities performed by APTs, including the TTPs used in each campaign. It was also crucial to analyse multiple real threat reports from multiple organisations that focus on securing ICS networks, containing very detailed information about the attacker's TTPs.

The selection process was made using the MITRE ATT&CK navigator tool, a visualization utility that can be used, among other things, to assess the detection coverage provided by multiple data sources [46]. Using this tool, 49 supported TTPs were identified. Table 3 enumerates all the selected TTPs that will be tested for detection. This sub selection was performed using the above selection method.

In this TTP selection, special care was taken to choose techniques that cover most tactics rather than focusing on a specific subset of attacks, such as testing all methods related to the Initial Access tactic. This ensures a diverse test plan that incorporates multiple common and important attack vectors against ICS/OT environments, essential for any solution that intends to secure ICS environments at various levels.

Each technique will be evaluated according to the corresponding stage of the attack test. This approach provides a better understanding of the attack scenario and how it can occur, presenting the possibilities at each stage of the attack chain.

5.2.2 Test Preparation

Besides setting up the test environment, before proceeding to the execution of the test plan, it is important to simulate the various actions from the different components and actors, which represent the baseline behaviour of the network and the operational process. This is crucial for correctly operating the behavioural detection methods based on the baseline module.

To achieve this, first, it is necessary to identify which legitimate actions can be executed within the

ATT&CK Matrix	Tactic	Technique
IT & ICS	Initial Access	Spearphishing Attachment (T0865)
	Command and Control	Standard Application Layer Protocol (T0869)
	Discovery	Remote System Discovery (T0846)
IT	Persistence	Boot or Logon Autostart Execution (T1547)
		Create or Modify System Process (T1543)
	Credential Access	OS Credential Dumping (T1003)
ICS	Lateral Movement	Remote Services (T0886)
		Valid Accounts (T0859)
		Lateral Tool Transfer (T0867)
	Collection	Point & Tag Identification (T0861)
	Inhibit Response Function	Device Restart/Shutdown (T0816)
	Impair Process Control	Unauthorized Command Message (T0855)
	Impact	Manipulation of Control (T0831)

Table 3: Selected TTPs

environment and what their conditions are, such as who performs this action and when. Besides the automatic interactions between assets that occur due to the normal execution of the operational process, these are the relevant manual actions that can be executed at any point in time:

- An operator can remotely log into the "jump machine" and "operator workstation" using the RDP protocol.
- From the jump machine, an operator can interact with the HMI via its webpage to see operational process measurements.
- From the operator workstation, an operator can access the PLC web service webpage to stop/start the process and modify the PLC program.

With these actions in mind and the test environment fully configured and working, a new baseline was recorded. This recording represents a full day of work in the test environment, where the collected activity includes all the network communications exchanged by all network assets, all Windows events logs generated by the Windows assets, and all the behavioural logs of the above-identified actions. All of this data will represent the normal behaviour of the test environment, which will be set as the baseline time frame in the baseline module configuration.

5.3 Results

This section details all the tests performed to evaluate the system's detection capabilities. As described earlier, the tactics and techniques will be introduced gradually according to the stage of the attack. The attack simulation starts with the attacker gaining a foothold in the organisation's network, then moving laterally between assets and networks until getting the capacity to access OT devices, and then finally, manipulating these devices to inflict damage to the operational process.

5.3.1 Initial Access

In this initial stage, the first attacker's goal is to find a way to get into the victim's organisation, which likely means gaining a foothold in the organisation's enterprise network. To achieve this, the attacker can use multiple initial access techniques, but without a doubt, the most popular and effective way to gain quick access to an organisation's network or one of its assets is by using a phishing attack against their employees.

Spearphishing Attachment (T0865)

The spearphishing attachment technique describes how threat actors use email attachments to gain access to the victim's machine [49]. Normally, these malicious attachments come in the form of Microsoft Office documents, PDF files, masked executable files, and others. This technique relies on the attacker's ability to convince the victim to open the sent attachment. Once executed in the victim's environment, the malicious attachment will grant the attacker system access to the victim's machine. For this to happen, the victim may be prompted to bypass application restrictions to execute the attachment payload (enabling macros on Microsoft Word), or it may exploit a specific vulnerability in the victim's operating system or client application without further interaction.

This technique is one of the most common initial access attack vectors targeting both enterprise and ICS environments due to its effectiveness and low skill barrier. Analysing various real-world examples where this technique was used, it is possible to verify that the most commonly exploited attachment type is Microsoft Word and Excel documents due to its widespread use in various organisation operations and its ability to execute compromising system commands without the victim realizing it. A concise example is the 2015 Ukraine attack, where threat actors gained access to the targeted networks by emailing employees with weaponized Microsoft (MS) Office files, specifically Word and Excel [67].

To test this technique, the Microsoft Office "Follina" vulnerability was chosen. Assigned with the CVE-

2022-30190 identifier, Follina is a remote code execution vulnerability found in the Microsoft Support Diagnostic Tool (MSDT) component. This vulnerability can be leveraged in a Word document by loading a remote template file using the `ms-msdt` URL protocol, which then calls the MSDT, executing the attacker's payload [72]. It can bypass Office-protected mode, and it does not require Office macros to be enabled, like many other attacks involving Word documents, which makes it especially dangerous since the victim only needs to open the Word document.

To detect the exploitation of this vulnerability, it is necessary to analyse the behaviour and activity of the application. Detection analysis shows that every time this vulnerability is exploited, the Microsoft Office Word parent process(`winword.exe`) creates a MSDT child process (`msdt.exe`). This is a unique behaviour, which can be used to create a detection rule. Fortunately, there are plenty of public detection rules for different security solutions that can effectively detect Follina exploitation. Among those, there is an EQL detection rule created by Elastic, which is supported by Argus [16]. This rule can detect not only the Follina vulnerability but also other malicious Word documents. It essentially monitors if the parent process of Office applications like Word, Excel, PowerPoint, and Access creates a new child process of uncommon utilities like `cmd.exe`, `powershell.exe`, and many others, including the `msdt.exe` subprocess, which strongly indicates a malicious non-common behaviour.

To validate this detection rule, the following steps were performed:

- Import the public detection rule to Argus.
- Create a Follina PoC using a publicly available Metasploit module [69].
- On the enterprise workstation, download the Word PoC document, simulating a user who had downloaded it from an email he had received.
- Open the malicious Word document, which immediately triggers a new alert.

For future reference, the payload was a PowerShell script that installed the Caldera Sandcat agent, a fully-featured cross-platform agent that will provide full-system access to the attacker. Figure 14 displays the Caldera agent page list, which contains the agent that was installed after opening the malicious Word document.

Finally, it is important to note that the tested detection rule only covers Microsoft Office application attachments, not others, like executables, PDF or zipped files. Even for Microsoft applications, it is not a silver bullet because, in the future, new vulnerabilities can be discovered affecting these applications that do not correspond to the detected behaviours, evading detection.

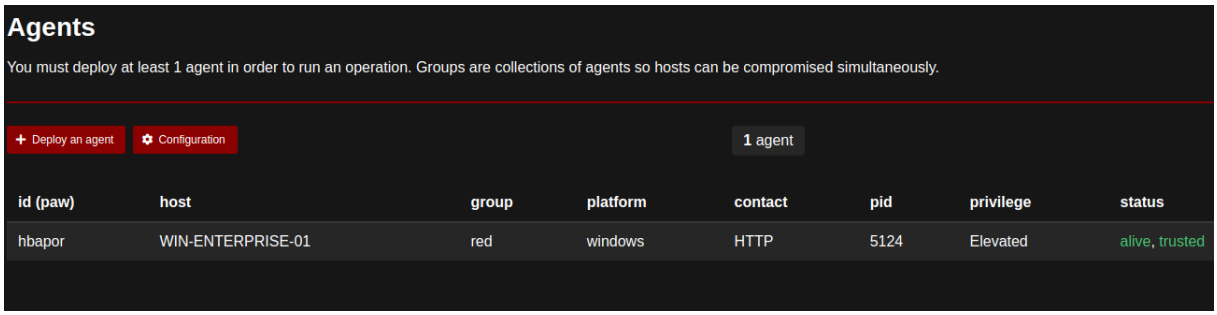


Figure 14: Caldera agents page

5.3.2 Command and Control

Once the malware created by the attacker successfully runs in the victim's system context, the first task it will perform is to establish a connection with the attacker's remote C2 server. The malware uses this channel to receive commands and instructions to be executed in the target system and then return the result to the attacker-controlled server. This tactic includes common enterprise machines like workstations and ICS-specific devices like HMIs and SCADA servers.

Standard Application Layer Protocol (T0869)

The standard application layer protocol technique states that threat actors use network protocols that can typically be found in ICS/OT environments like HTTP, RDP, DNS, Modbus, DNP3, and others when communicating with C2 servers [50]. This helps attackers to blend in with normal network traffic, which reduces the chances of detection.

Associated with this technique, four different ATT&CK enterprise sub-techniques (T1071.001-4) explore how threat actors use various typical protocols to bypass network filtering and evade detection. This includes protocols used in web browsing (HTTP/HTTPS and DNS), file transfers (File Transfer Protocol (FTP)/Server Message Block (SMB)), and electronic mail (POP/IMAP/SMTP). Analysing the most common protocols used by threat actors in real attacks in both enterprise and ICS environments shows that HTTP and DNS are the common choices to covertly establish a connection with a C2 server. Further examination of how these two protocols are used in C2 communications shows that DNS-based communication can be easily detected, unlike C2-based HTTP/HTTPS communication, which may require TLS inspection, signature matching, monitoring known malicious domains, and other methods.

As such, to detect the usage of the `Application Layer Protocol: DNS (T1071.004)` technique, it is necessary to analyse how the DNS protocol is used to communicate with a C2 server. For this purpose, the popular DNS tunnelling tool dnscat2 was used to establish a C2 channel from the enterprise

workstation. The Caldera agent has the "DNS C2" ability, which uses the dnscat2 tool. Using this ability, a reverse shell was established with the attacker's remote machine, as shown in figure 15.

```
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>
shell 3> systeminfo
shell 3> systeminfo

Host Name:                 WIN-ENTERPRISE-
OS Name:                   Microsoft Windows Server 2019 Standard Evaluation
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             POTENTIAL CACHE HIT
Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00431-10000-00000-AA235
Original Install Date:     24/12/2022, 00:17:44
System Boot Time:          20/12/2024, 00:43:42
System Manufacturer:       innotek GmbH
System Model:              VirtualBox
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: AMD64 Family 25 Model 97 Stepping 2 AuthenticAMD ~3700 Mhz
BIOS Version:              innotek GmbH VirtualBox, 01/12/2006
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             pt;Portuguese (Portugal)
Input Locale:              pt;Portuguese (Portugal)
Time Zone:                 (UTC+00:00) Dublin, Edinburgh, Lisbon, London
```

Figure 15: Dnscat2 Windows shell command execution

After some minutes of playing around with the victim's machine, by executing a few OS commands and retrieving some system information, the following patterns can be observed:

- Abnormal high volume of DNS queries originating from a single machine to a single domain over a short period of time.
- A suspiciously high number of unique Fully Qualified Domain Name (FQDN)s for the same domain.
- The query subdomain is an unusually long string, more than 20 characters, with random alphanumeric characters, ex: 2c4801be485c838cc408d900a505d03327.domain.com

Based on these indicators, a threshold rule was created, which monitors Zeek DNS looking into:

- DNS queries in which the subdomain is an alphanumeric string of over 20 characters. For this, a Painless script was used with the following script:

```
doc['dns.question.subdomain'].size() > 0 &&
```

```
doc['dns.question.subdomain'].get(0).length() > 20
```

- DNS queries originate from the same internal machine and target the same domain.
- Traffic volume when it surpasses 1000 queries in a 1-hour time frame.

To validate this rule, the following steps were performed:

- With the Caldera Sandcat agent up, rerun the "DNS C2" ability. It is necessary to modify the original ability execution command with the attacker's domain and also, in this case, to redirect the default DNS server to the attacker's machine.
- With the new session established, open a command session and execute some commands to collect system information.
- The rule triggers a new alert, as shown in figure 16.

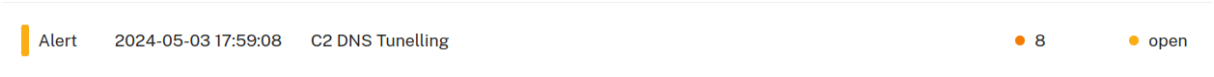


Figure 16: C2 DNS tunnelling alert

5.3.3 Persistence

After establishing a covert communication channel with the attacker's remote server, the attacker must guarantee access to the victim's machine at any time. This requires the setup of persistence mechanisms on the target machine to ensure continuous access.

Boot or Logon Autostart Execution (T1547)

Part of the enterprise ATT&CK matrix, the boot or logon autostart execution technique lists the different ways that threat actors achieve and maintain persistence in a system by automatically executing a malicious payload when a system boots or a user logs in [42]. This is a common way to maintain persistence in a victim's system and is widely used in enterprise network attacks. This technique contains 15 sub-techniques, each addressing a specific method to achieve persistence. One example is the Havex malware that writes itself to disk in a shared folder and then creates an auto-start registry key that points to the path of the malware file, which will be executed when a user logs into the Windows machine [62].

There are well-known detection indicators for identifying the use of registry run keys or startup folders to gain persistence, corresponding to the `Registry Run Keys / Startup Folder (T1547.001)` sub-technique. Multiple registry hives contain run keys that can be used for persistence. The most abused run keys are the `Run` and `RunOnce` registry keys, which, per the Windows documentation [48], are located in the following registry paths:

- HKLM\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce

Figure 17 presents an entry example that is stored in the `Run` registry key.

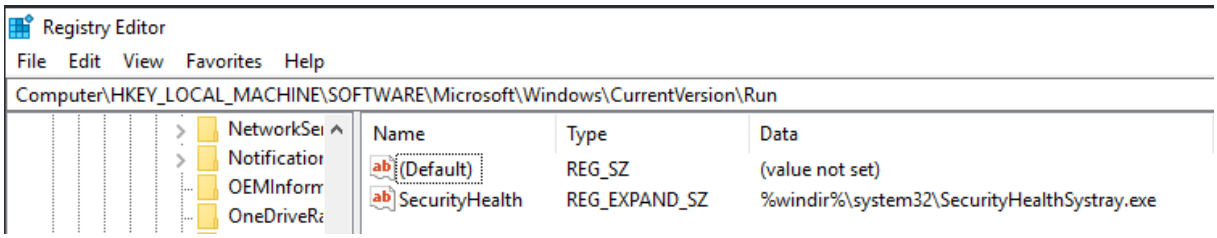


Figure 17: Registry run key contents

The AutoRun tool from the Sysinternals Windows tools suite contains many more registry keys that can be used for this purpose [32]. Monitoring all of these sources allows for greater coverage of this technique.

On the other hand, in the case of startup folders, a program will be executed if it is present either in the user or system-specific startup folders. The startup folder paths are the following:

- C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp (System Path)
- C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup (User Path)

It is important to note that these paths can be changed by the user, or in this case, the attacker, by modifying the `Startup` key in the User Shell Folders registry path, requiring additional Windows registry monitoring. Based on these indicators, it is possible to detect boot or logon autostart persistence by:

- Monitoring modifications on the above-listed registry run keys. This can be achieved using the Sysmon 12 event ID, that tracks registry event changes.

- Monitor the startup folders for new executables. This can be achieved using the Sysmon 11 event ID, which logs every new file creation/modification in a given folder. As said previously, the startup folder paths can be changed by the user, and as such, it is crucial to also monitor registry changes in the "Startup" and "Common Startup" keys of the following Windows registry path: `HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders`.

As a result, three EQL rules were created for these two use cases. The registry run keys rule is a public EQL rule, shared by Elastic [20]. It was chosen because it considers many Windows native applications, services, and common legitimate programs that interact with and change registry run keys, which could result in many false positives. For the startup folder rules, two EQL rules were created, one monitoring new shortcuts created in the default startup folders and the other monitoring the registry startup folder location keys.

To validate the registry run key rule, the following steps were performed:

- Modify the Caldera "Reg Key Run" ability by adding additional executors to create a new persistence key for each of the above run keys.
- With the Caldera Sandcat agent, run the modified "Reg Key Run" ability.
- A new alert is triggered, as shown in figure 18

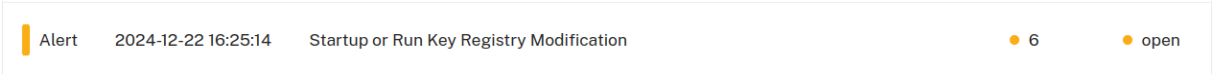


Figure 18: Startup or run key registry alert

To validate the startup folder rule, the following steps were performed:

- Execute the Caldera "Add Executable Shortcut Link to User Startup Folder" ability, which creates a new application shortcut in the current user startup folder.
- A new alert is triggered, as shown in figure 19.

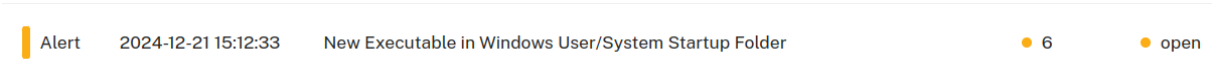


Figure 19: New Executable in windows user/system startup folder alert

Create or Modify System Process (T1543)

This technique states that threat actors will create or modify existing services, like Windows services, that will be started when the system initializes [36]. Services can be used to achieve persistence by executing a malicious payload created by the attacker. Services are a standard mechanism to ensure that essential background programs are started when the system initializes. This makes it a popular and effective persistence vector in many real-world attacks. Some examples include the Industroyer malware that modifies an existing non-critical Windows service and changes the service executable path to a malicious executable [5], and the Electrum threat actor that uses native Windows utilities to create and start malicious Windows services remotely [60].

To detect this persistence vector, specifically on Windows systems, it is critical to monitor service configuration changes. These service changes can be performed using the following methods:

- Directly create or modify Windows registry keys related to Windows services. This can be detected using the Sysmon 12 event ID. The corresponding Windows services registry path is: HKLM\System\CurrentControlSet\Services
- Using Windows utilities like `sc.exe` or the PowerShell cmdlet `New-Service` and `Set-Service` to create new Windows services or modify the configuration of existing ones. Analysing the behaviour of both Windows utilities shows that both change key values in the Windows registry tree `HKLM\System\CurrentControlSet\Services`. As such, it is unnecessary to monitor process creation based on the `sc.exe` utility or PowerShell cmdlet execution. Additionally, a new Windows event is triggered with the 7045 event ID when a new service is created.

Figure 20 shows a service entry in the `HKLM\System\CurrentControlSet\Services` registry path.

Based on these indicators, a new EQL rule was created. This rule detects modifications on the Windows registry `HKLM\System\CurrentControlSet\Services` path, specifically, changes involving the `ImagePath` key. It also takes into consideration that legitimate Windows programs frequently apply changes to these Windows registry paths, which would lead to false positives, if not correctly excluded.

To validate this new EQL rule, the following steps were performed:

- Execute the Caldera "Service Installation CMD" ability, which uses the `sc.exe` utility to create a new service.
- Execute the Caldera "Service Installation PowerShell" ability, which creates a new service using the

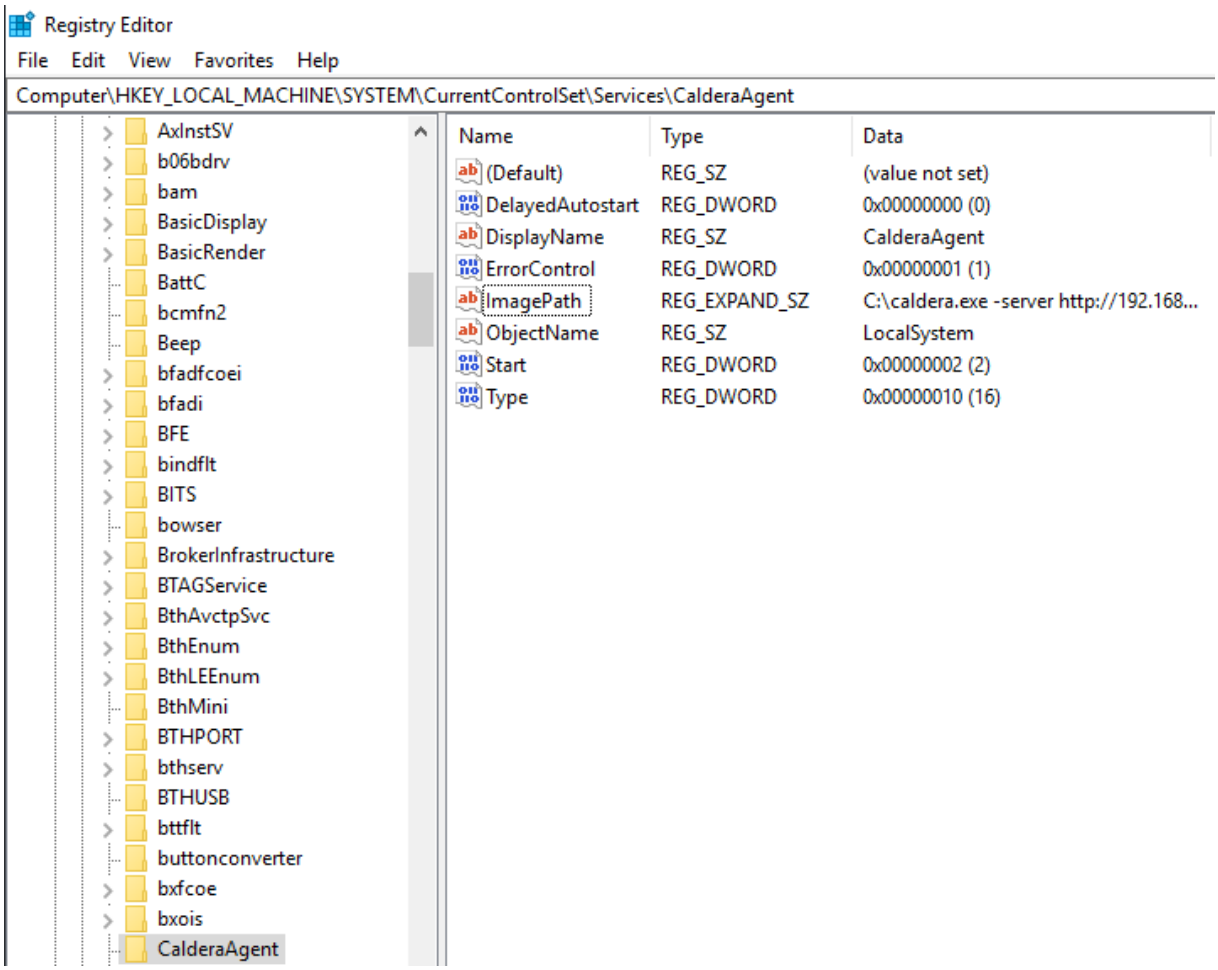


Figure 20: Windows service registry key

`New-Service` Powershell cmdlet.

- Create a new custom Caldera ability, "Windows Service Configuration Change (sc.exe)", that uses the `sc.exe` utility with the `config` parameter to modify the configuration of existing services.
- After executing each Caldera ability, each corresponding rule triggers a new alert, as shown in figure 21.

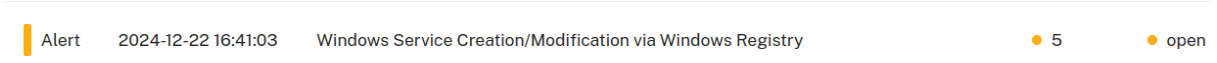


Figure 21: Windows service creation/modification via windows registry alert

5.3.4 Discovery

Now, with access to the enterprise network, the attacker's next step is to perform internal network reconnaissance to find potentially vulnerable targets that can provide further privileged access to other sections of the organisation network.

Remote System Discovery (T0846)

The remote system discovery technique exists in both ATT&CK enterprise and ICS matrices. It describes the different methods used by threat actors to find network devices present in the victim's network that could potentially be used or exploited to gain further access to the system [47].

The analysis of real-world use cases shows that threat actors commonly use native utilities to display information about known remote systems, like the Windows `net` command that can be used to list all the machines that are part of an AD domain, or the `arp` command(supported in Windows and Linux), which lists the system's arp table that contains network information about the remote systems the machine has communicated with.

Another typical action is to actively scan the internal network, running specific scan applications. Normally, this would be done using specific port scan utilities such as the widely known nmap tool, or even using custom-made tooling. This can be used in enterprise networks to find specific machines like HTTP servers, databases, domain controllers, RDP-enabled Windows machines, etc. In the case of an ICS/OT network, it is used to discover ICS-only devices like an HMI, PLC, and others. There are multiple examples of active network discovery being used by threat actors in ICS environments like the ICS-specific PIPEDREAM malware, which performs multiple TCP/UDP port scans to identify Schneider PLCs and OPC UA servers [74]. There is also the Industroyer malware that can enumerate the existing network adapters and perform a port scan in their correspondent network subnets for hosts that have the 102 port(IEC 61850 protocol) open [5]. Finally, the Havex malware includes enumeration modules that perform port scans using a list of hard-coded ports related to multiple ICS network protocols [62].

Examining these examples reveals some key detection indicators of active network scanning:

- Attackers may perform ping sweeps on a network subnet to identify active hosts.
- Attackers may perform port scans across a network subnet, targetting a single port or a short list of custom ports.
- Attackers may perform a full port scan against a single host.

Normally, IDS solutions like Suricata are the primary choice to identify active enumeration activity in a network. However, in this case, the Zeek connection log collects all the necessary information to detect these activities, including every network connection attempt between machines.

In the first case, ping sweeps are sent initially by attackers to identify active hosts on specific subnets, which then become targets of port scan operations. Ping sweeps rely on Internet Control Message Protocol (ICMP) echo requests(type 8) sent to the target hosts that respond if they are alive. It is important to note that some systems do not respond to ICMP requests by default, in particular Windows systems, due to default Windows firewall rules. As a result, a sudden spike in ICMP requests from a single host to multiple other assets within a short period may suggest enumeration activity. Consequently, a threshold rule was created to detect ping sweeps using Zeek events that:

- Match the `connection` file set (`fileset.name`).
- Match the `icmp` network protocol (`network.transport`).
- Match the "8" ICMP type (`zeek.connection.icmp.type`).
- Group events by the source IP address(`source.ip`).
- Surpasses the limit of 5 distinct ICMP requests in a 1-minute time frame.

To evaluate this rule, the engineer workstation will be used to find all live OT assets that are within the OT network `192.168.95.0/24`. To test this, the following steps were performed:

1. Edit the `Remote System Discovery - ping sweep` Caldera ability by changing the default subnet to the OT subnet (`192.168.95.0/24`).
2. Run the `Remote System Discovery - ping sweep` using the engineer workstation caldera agent.
3. The rule triggers a new alert, as shown in figure 22.



Figure 22: Ping sweep enumeration alert

In the second case, the attacker executes a port scan across multiple hosts in a subnet to discover assets that have a particular service running like FTP, SMB, RDP, Modbus over Ethernet, etc. When

performed against all subnet hosts, this kind of activity generates a lot of traffic, where each connection attempt will be stored in the connection log. To detect port scans across multiple hosts, a threshold rule was created that monitors Zeek events that:

- Match the `connection` file set.
- Group the events by the source IP address(source.ip) and the destination port(destination.port).
- Surpasses the limit of 5 distinct destination addresses in a 1-minute time frame.

To evaluate this rule, the enterprise workstation will be used to find IT assets of interest, and the operator workstation will be used to enumerate assets in the OT network. For the case of the enterprise workstation, the following steps were performed:

1. Edit the default Caldera `Network Service Discovery` ability to scan hosts from the local networks (192.168.80.0/24 and 192.168.95.0/24 subnets, in our case) targetting the RDP port (3389).
2. From the enterprise workstation, run the Caldera `Network Service Discovery` ability.
3. The rule triggers a new alert, as shown in figure 23.



Figure 23: Single port scan on multiple assets alert

Finally, in the third case, the attacker targets a single asset but tries to scan multiple ports. This is useful to find if additional services, besides the common ones, are being run on the target, as they might have vulnerabilities that can be exploited or have services that disclose sensitive information. A full port scan can be performed over all the 65535 ports, but it is common to scan the first 1000 or 10000 ports. Similarly to the previous case, this scanning behaviour is noisy and can be easily detected. The following threshold rule was created to monitor Zeek events that:

- Match the `connection` file set.
- Group the events by the source and destination IP addresses (source.ip, destination.ip)
- Surpasses the limit of 50 distinct ports in a 5-minute time frame.

To evaluate this rule, the following steps were performed:

1. From the enterprise workstation, run the `Port Scan` Caldera ability to scan the entire TCP ports space of the jump box machine (192.168.90.115).
2. The rule triggers a new alert, as shown in figure 24.

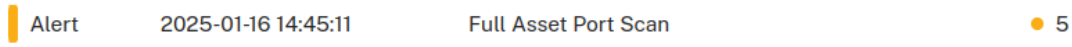


Figure 24: Full port scan on a single asset alert

From these active enumeration activities, the attacker found the jump machine in the DMZ network, which runs the RDP service. Later, once he had gained access to the operator workstation, he discovered the PLC (192.168.90.2) and the process simulation box (192.168.90.10-15). This information will be helpful in later stages of the attack.

5.3.5 Lateral Movement

Now that the attacker has established a secure and stable foothold on the IT network and discovered potentially vulnerable assets, the next step involves the use of multiple techniques to gain deeper access to the organisation network by pivoting between systems, gaining privileged access to sensitive assets.

Remote Services (T0886)

The Remote Services technique states that threat actors can abuse remote services enabled in workstations and dedicated maintenance instances, used normally by external employees or contractors [43]. Remote services like RDP and VNC provide a graphical user interface for remote users that normally is used to interact with other ICS components like HMIs. Because of its capabilities, remote services can access privileged services and devices to exchange data and files. There are multiple occurrences of remote service abuse by threat actors on ICS environments, including the TRITON malware that used the RDP protocol to pivot and blend in the network [64].

Associated with this technique is the ATT&CK enterprise password guessing technique (T1110.001), whose goal is to guess account passwords via a common password wordlist by abusing limited or even inexisting login password and account lock policies [40]. As such, an attacker could continually try to guess different user passwords until he is successful.

RDP account login attempts can be identified and detected by monitoring the Windows event ID 4625, which records every failed login attempt. It is also necessary to filter login attempts that match the Network login type. Monitoring events with these characteristics makes it possible to identify multiple failed login attempts in one or more accounts from a specific source within a short time frame.

To create a detection rule based on these indicators, it is important to first identify the common login patterns. Setting a low threshold on the number of failed attempts required to trigger an alert can lead to false positives. Figure 25 shows that the profiler tool can be used to see the patterns of login attempts in the environment. Based on this data, it is safe to place a minimum threshold of 10 failed attempts within a 10-minute time frame.

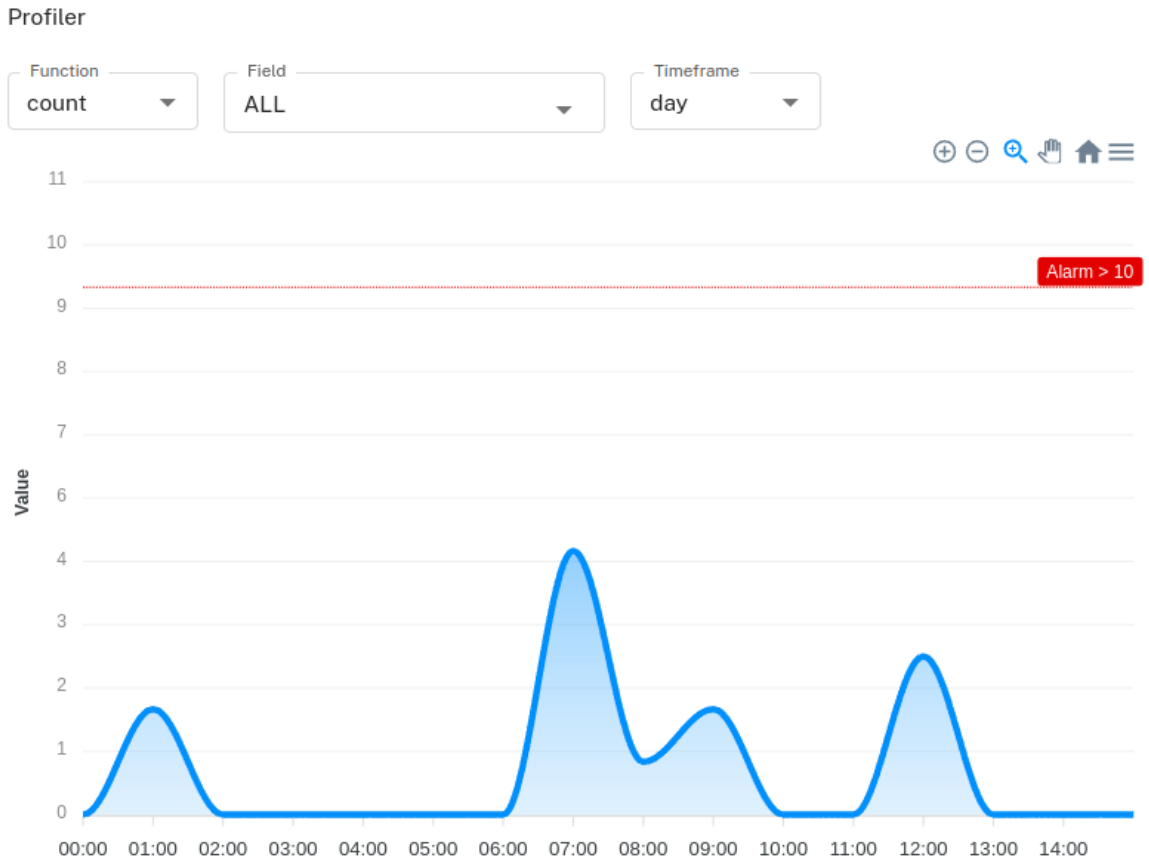


Figure 25: Windows failed authentication attempts profiling

Following that observation, a threshold rule was created to monitor Windows events that:

- Match the 4625 Event ID and the Network Login Type.
- Originates from the same IP address.
- Surpasses the maximum limit of 10 attempts in a 10-minute time frame.

To validate this detection rule, the following steps were performed:

- Create a Caldera ability called "SMB Account Brute-force" that, given a password wordlist file, will try to authenticate with each password into a target machine and user account.
- From the Caldera agent running on the enterprise workstation, run the "SMB Account Brute-force" ability to initiate an SMB password brute-force attack against the jump machine "john" user.
- The rule triggers a new alert, as shown in figure 26.



Figure 26: Windows RDP bruteforce rule alert

Valid Accounts (T0859)

The valid accounts technique is linked to threat actors using stolen user credentials, obtained through different credential access techniques, to gain privileged access to machines that might be part of sensitive network segments like the ICS/OT network [45]. This is possible because, in an environment that uses an AD domain, valid user domain credentials can be used to pivot through systems that might have access to other privileged domain credentials, allowing the attacker to move laterally in the network. In other cases, due to password policy limitations, passwords might be reused in multiple systems across multiple accounts, making life easier for attackers. There are multiple examples of real-world attacks that leveraged the use of stolen credentials to move laterally in the victim's network by mainly using remote access services like RDP [60] [9].

Since threat actors use valid user credentials to access systems, it is difficult to signal these events as malicious because they mimic normal behaviour. Once again, behavioural analysis is a crucial method for detecting potential malicious access. Some of the behaviours that can be used to detect malicious use of valid credentials are:

- Login outside of business hours (indication of abnormal system access).
- Multiple logins across different systems by the same account within a short/medium time frame (indication of an attacker testing which machines the valid credential has access to).
- A user account logging into a system that he has never accessed before (indication of an attacker using a valid set of credentials to pivot through systems).

Based on the gathered baseline data corresponding to each of the above behaviours, it is possible to create rules that detect deviations from the normal account authentication patterns. In this way, three rules were created to provide the most comprehensive detection coverage for this technique.

The first rule, "account authentication outside business hours", detects account logins that occur outside the organisation's business hours. This assumes that employees do not normally access the monitored systems beyond their fixed schedule, which means that any access outside this hourly period is considered abnormal and should be analysed. As such, a threshold rule was created that monitors Windows events that:

- Matches the 4624 Event ID(successful login) and the Network Login Type
- Filters the timestamp for logins outside the defined business hours. In this case, for example, could be the 06h-22h period, using the following painless script:

```
doc['timestamp'].getHour() < 6 &&
doc['timestamp'].getHour() > 22
```

To validate this rule, the following steps were performed:

- Create a custom Caldera ability called "Simulate SMB Logon" that establishes a RDP session connection with the target asset using a valid set of credentials.
- From the Caldera agent running on the enterprise workstation, execute the "Simulate RDP Logon" ability to log into the "Administrator" user in the jump machine outside of the business hours period.
- The rule triggers a new alert, as shown in figure 27

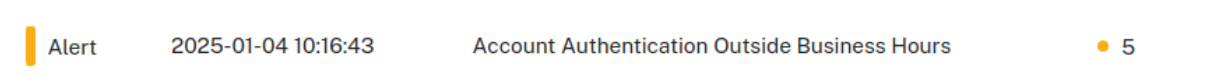


Figure 27: Account authentication outside business hours alert

The second rule, "Windows account authentication attempts in multiple systems", detects both successful and failed remote authentication attempts made by the same source IP address and targeting the same Windows account within a short time frame. This behaviour may indicate that a threat actor with access to a valid set of credentials is testing in which reachable systems he is able to authenticate to. Therefore, a threshold rule was created that monitors Windows events:

- Matches the 4624 Event ID(successful login) or the 4625 Event ID (failed login) and the Network Login Type.
- Groups the resulting events by the source IP address (`source.ip` and the Windows account username.
- Check if the number of distinct target systems is greater than 3 within a 10-minute time frame.

The validation of the second rule assumes that the organisation network contains multiple Windows machines. In the case of this test environment, only three machines are configured, requiring the test of the credential set against all three test machines. Normally, due to the vast number of assets that are present in these networks, this attack would target multiple Windows web servers, domain controllers, SQL servers, and workstations. In this regard, the following steps were performed:

- Create a custom Caldera ability called "Simulate SMB Logon on Multiple Systems" that tries to establish a RDP session connection with a list of target assets using a set of credentials.
- From the Caldera agent running in the jump machine, run the custom Caldera ability "Simulate SMB Logon on Multiple Systems." This ability requires a single account credential set, but targets the three Windows environment machines: the enterprise workstation, the jump machine itself, and the operator workstation.
- The rule triggers a new alert, as shown in figure 28.

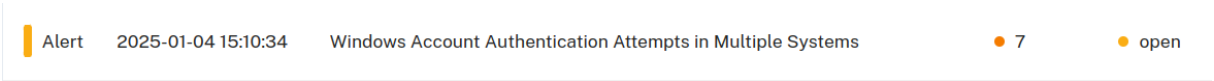


Figure 28: Account authentication attempts in multiple systems alert

The third rule, uncommon Windows authentication, uses the baseline module to detect authentication patterns that do not align with the pre-defined baseline. It aims to detect two different authentication cases: the source address has never tried to authenticate into the target Windows asset; or the provided Windows account has never authenticated remotely on this same asset.

In the first case, it is possible to detect if an asset in the network is trying to remotely authenticate into a higher-privilege machine that, per the baseline, it should not need access to. The second case allows one to detect when a threat actor reuses a valid credential set to authenticate into a higher-privileged machine, specifically in AD environments where a valid domain account may be used to access other machines that were not properly configured.

Before creating the rule, it is necessary to define the baseline that will be used in the rule logic. The first analytic, `windows.authentication.source` tracks Windows authentication events and signals which source addresses (`source.ip`) has successfully remotely logged in a given Windows asset (`host.ip`). The second analytic, `windows.authentication.account` tracks Windows authentication events and signals which local/domain accounts (`TargetUserName`) have successfully remotely authenticated on a given Windows asset (`host.ip`). Figure 29 illustrates how the information is gathered.

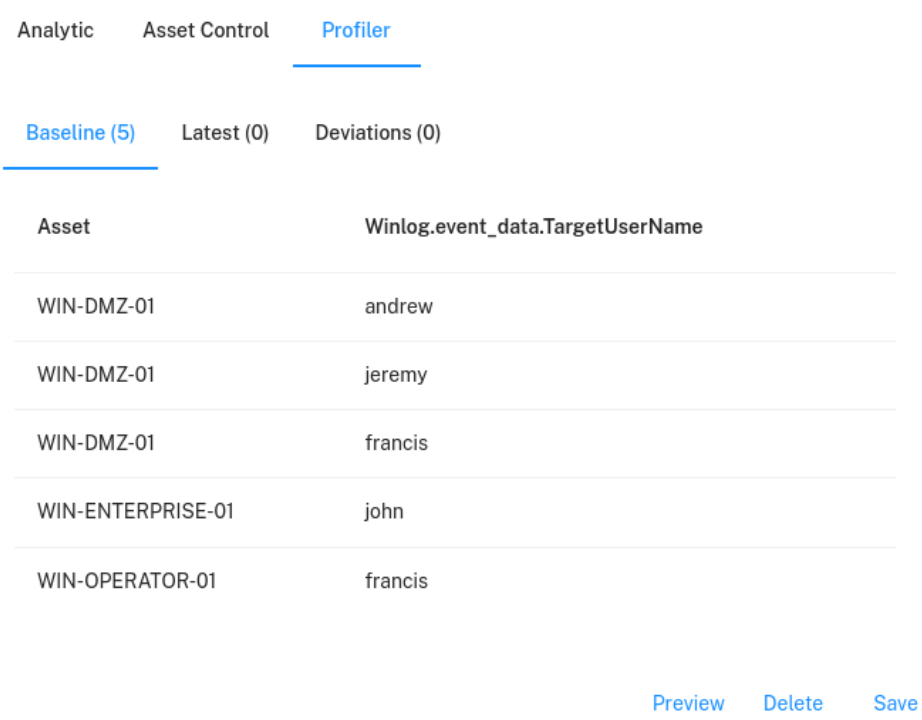


Figure 29: Windows asset authentication account profiling

With these steps completed, it is possible to create the final EQL rule that monitors for baseline events that correspond to one of the above baseline analytics: `windows.authentication.source` and `windows.authentication.account`.

Two scenarios were defined to validate the third rule and its two use cases. The first, where the source address has never been logged in a Windows system, will involve the enterprise workstation and the jump machine in the DMZ network. This example assumes that a threat actor successfully took over the enterprise machine, gathered a valid credential set, discovered the jump machine in the DMZ network, and then authenticated and gained access to this machine. It also assumes that the enterprise workstation has never interacted with the jump machine.

In the second use case, a known trusted source address logs remotely into a Windows system but using a Windows account that has never been used before for that purpose. The test scenario involves

the jump machine that normally is used to access the engineer workstation in the OT network but using a non-operator account that still provides access. This assumes that the threat actor took over the jump machine in the DMZ network, which has network access to the operator workstation in the OT network and accepts remote authentication from accounts that would not normally need to access these systems.

Therefore, for both use cases, the following steps were performed:

- Execute the Caldera ability "Simulate SMB Logon", which logins in the target Windows system using SMB protocol with the provided credentials.
- The baseline-based EQL rule triggers a new alert.

Some points about these tests are worth mentioning. The first rule assumes a strict business hours schedule, where no tasks are performed outside this period. However, for maintenance or security reasons, industrial environments may require external operators and providers to access these systems at any time, anywhere, as required.

Furthermore, the last two scenarios may lead to multiple false positives if the organisation's network constantly changes, with new devices added regularly.

For example, new assets can be legitimately added to any network segment, including the DMZ and OT network, which can be used to access other machines remotely or even be used by others. This network connection and interaction between systems will not be part of the selected baseline time frame, which causes both rules to trigger false-positive alerts.

On the other hand, it does not also consider that new local/domain accounts can be created for new employees who need to connect to other systems to perform their jobs remotely.

With these edge cases in mind, it is important to regularly fine-tune this monitoring activity to reduce the number of false positives it may generate.

Lateral Tool Transfer (T0867)

This is another technique that is part of both the enterprise and ICS ATT&CK matrices, and it shows how threat actors move tools, payloads, and utilities between overtaken machines to have access to non-native capabilities in machines deeper in the enterprise/industrial network [37]. Examples of these newly acquired capabilities include malicious payloads like executables or a Dynamic-Link Library (DLL) that will be dynamically loaded on the victim machine to provide custom functionality, for example, to execute ICS payloads and commands crafted by the attacker [60].

Analysing real-world occurrences shows that tools are moved using common remote file-sharing ser-

vices like SMB, FTP, or using remote services such as RDP and telnet [74]. Therefore, to detect lateral tool transfers, some stand-alone indicators can be used separately or, more effectively, together to detect this kind of activity:

- Monitor remote file-sharing protocols like SMB and FTP, specifically between systems that never communicated using these protocols.
- Monitor file creation of the same file by checking the file hash across multiple systems. Unfortunately, it is impossible to use this indicator as Winlogbeat does not log the file hash when creating a new file.
- Monitor processes using the above network protocols, which create and write suspicious system files like executables or DLLs (prone to false positives).

Two rules were used based on these detection indicators. The first rule, "Potential Lateral Tool Transfer through an SMB Share", tries to detect the creation of new files from network shares that contain suspicious extensions such as exe, dll, bat, ps1, etc. This is a public EQL rule shared by Elastic [21].

The second rule, Suspicious File Transfer through an SMB Share, builds on the previous rule and correlates the Windows and baseline data sources to detect suspicious file transfers. As shown in figure 30, this EQL rule detects if, within 60 seconds, the following three events occurred sequentially:

- Establishment of a network connection between two systems using the SMB protocol.
- On the SMB destination host, a new file was created (detected using the Sysmon 11 event id) by the System process (process id 4) and contains a suspicious extension. Ex: executable file (.exe) or DLL (.dll). This could include other extension files like archives (.zip, .tar, .tar.gz), etc.
- A baseline deviation on the network protocols used between systems, in this case, specific to the SMB protocol (port 445). To detect the `network.protocol.channels` baseline analytic was created over the destination port of a given communication between two different assets.

Two machines were used to validate these two EQL rule: the enterprise workstation and the jump machine. In this case, the tool that will be transferred is the Caldera agent, which was successfully implanted in the enterprise workstation through the earlier spear phishing attack. The goal is to move this executable, located on the enterprise workstation, to the jump machine to gain complete remote access to this machine. This requires a valid set of credentials that provide access to the jump machine. This credential set was gathered earlier using the remote services technique through a brute-force attack. Using

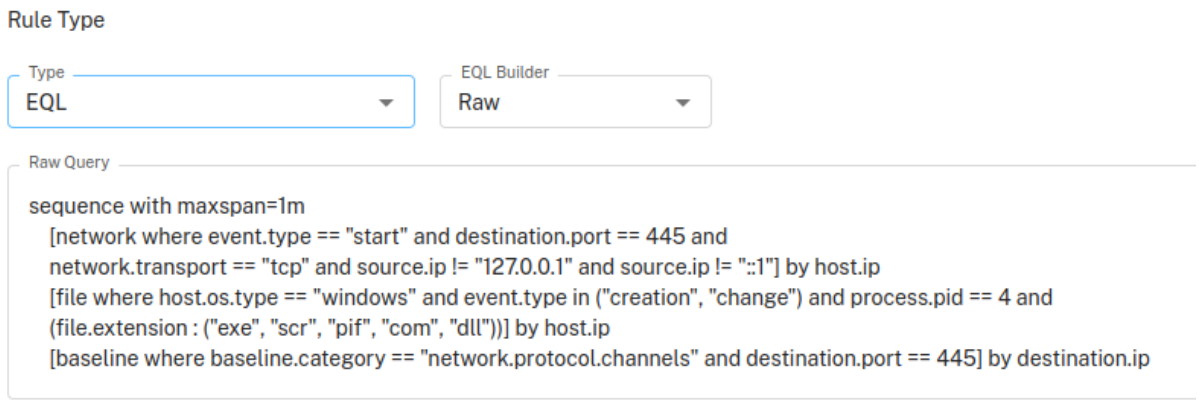


Figure 30: Suspicious file transfer EQL rule

this credential, it is possible to copy any file to the victim machine using the default Windows administrative share C$ if the credentials are from an administrative user, which is the case, or a non-default share to which the target user has write access. As such, the following steps were executed:

1. Creation of a new Caldera ability "Copy Sandcat Agent File using SMB share" that establishes a valid connection to a specific remote SMB share and copies to the share a specific file, in this case, the sandcat agent executable.

2. From the workstation machine, execute the above Caldera ability, where the destination is the jump machine, the target share is the default administrative share C$, the payload is the Windows caldera agent, and the username and password is the gathered Administrator:notsosecure credential.

3. The caldera executable file is successfully transferred to the jump machine.

4. Both rules trigger a new alert, as shown in figures: 31, 32.

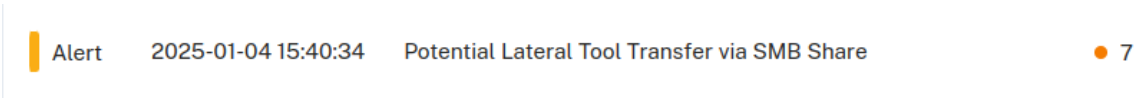


Figure 31: Potential lateral tool transfer through an SMB share alert

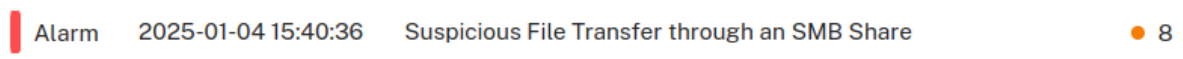


Figure 32: Suspicious file transfer through an SMB share alert

5.3.6 Credential Access

At this moment, the attacker only needs to gain access to the operator workstation present in the OT network. To achieve this, he needs a valid operator credential.

OS Credential Dumping (T1003)

Taking advantage of the procedures shown in the OS credential dumping technique, attackers can obtain sensitive credentials stored in a local machine. This fact makes this technique one of the most important techniques for attackers who need to collect privileged credentials to access further systems in a network. A real example is the ELECTRUM threat actor that targeted Ukraine's power grid in 2016 and used the popular mimikatz tool to retrieve network credentials [60]. This technique contains eight sub-techniques that explore different methods and vectors that attackers can exploit. One of these sub-techniques is the LSASS memory dumping technique (T1003.001) that can be used to extract hashed credentials from the Local Security Authority Subsystem Service (LSASS) process memory [38].

LSASS is a core process of the Windows authentication architecture, responsible for enforcing security policies, creating access tokens, and validating local or remote user authentication. As such, when a user authenticates in a Windows system, the LSASS process will cache the user account's New Technology LAN Manager (NTLM) hash in the process memory space until the system is shut down. This has many advantages as it allows Windows users to access remote resources at any time without the need to provide the user password every time, which is crucial in AD environments that do not enforce Kerberos authentication.

Because LSASS handles sensitive data, only the SYSTEM user and the local or domain administrator accounts have the privileges to dump the LSASS process memory. For that reason, attackers must first gain access to a local or domain administrator credential or exploit a local privilege escalation vulnerability.

From a defense perspective, dumping the LSASS process memory is a highly suspicious activity that should be closely monitored. There are multiple ways to dump the LSASS process memory, with varying levels of obfuscation and evasion, for example: using the task manager to dump the `lsass.exe` process memory through the `Create dump file` functionality, using legitimate built-in Windows utilities (LOLBins) like `procdump.exe`, the MiniDump function of the `comsvcs.dll` DLL, and many others. Additionally, multiple popular tools can automatically do this: mimikatz, pypykatz, LaZange, and others.

Now, concerning detection methods, a detection analysis blog post from the Splunk team shows that the majority of methods used to dump the `lsass.exe` memory have the following characteristics [70]:

- The utility chosen by the attacker will try to access the `lsass.exe` process to retrieve its memory contents. This can be detected using the sysmon event ID 10 (ProcessAccess).
- Normally, the access request permission for the `lsass` object will be one of the following: PROCESS_ALL_ACCESS(0x1fffff), 0x1010, 0x1038, 0x1438, or 0x143a.
- The utility loads specific DLLs that have API functions that aid in the dumping process, like `dbgcore.dll`, `dbghelp.dll`, and `ntdll.dll`.

On the other hand, multiple Windows applications legitimately request access to the `lsass.exe` process that must be filtered to prevent false positives [70]. Based on these indicators, an existing Sigma rule was used and converted to a supported EQL rule [55].

The purpose of this technique test is to collect the credentials of users who have access to the operator workstation in the OT network. Because real operators will use the jump machine to connect to the operator workstation, the LSASS process will cache the password hash of their account when they remotely authenticate. As mentioned previously, this attack is only possible if the attacker has access as a local/domain administrator, which is the case because, in the valid accounts technique 5.3.5, the "Administrator" password was successfully brute-forced.

To validate the EQL rule, the caldera ability `Powershell Mimikatz` was selected as it uses the very popular red-team mimikatz tool to dump the `lsass.exe` process memory. This test assumes that a real operator has already, at some point, logged into the current Windows system. Therefore, the following test steps were performed:

- Simulate a real operator called "andrew", logging remotely into the jump machine.
- From the Caldera agent running on the jump machine, execute the `Powershell Mimikatz` ability.
- A list of cached credentials is returned, including the "andrew" account NTLM hash, as shown in figure 34.
- The rule triggers a new alert, as shown in figure 33.

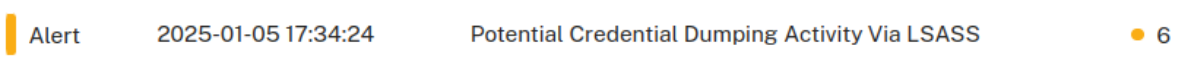


Figure 33: Credential dumping activity via LSASS alert

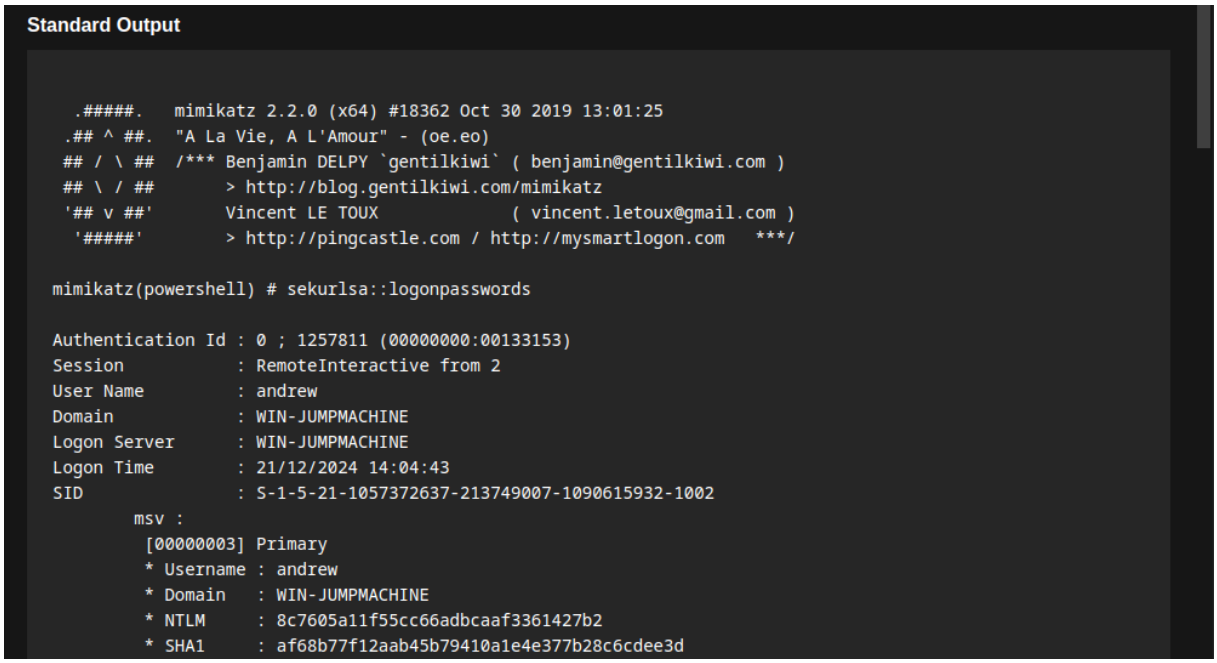


Figure 34: Jump machine LSASS dump with Caldera "Powershell Mimikatz" ability

Once the attacker has access to the NTLM hash of the victim account, he/she can use it to impersonate the victim and authenticate to other Windows systems through a pass-the-hash attack if within an AD environment. In this case, however, it is assumed that the attacker locally brute-forced the NTLM hash and found out that the real password of the "andrew" user is "andrew1980", as shown in figure 35. With this credential, it is now possible to access the operational workstation in the OT network.

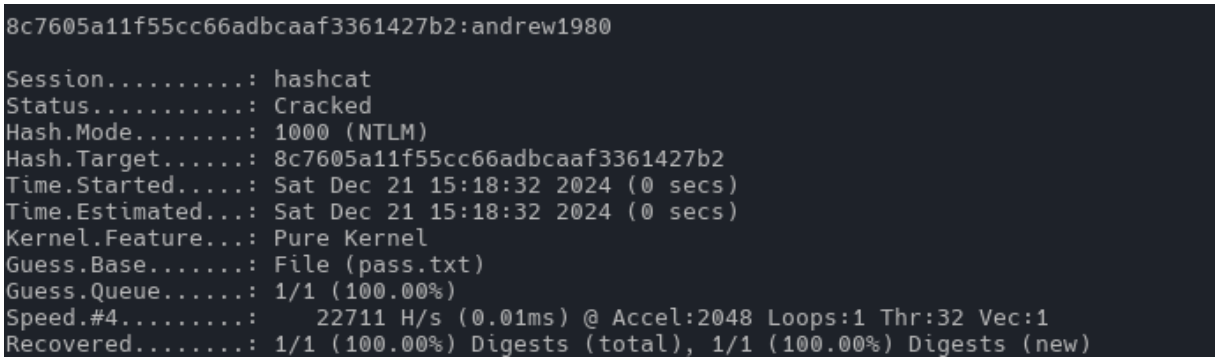


Figure 35: andrew's NTLM hash crack with hashcat

5.3.7 Collection

After gaining access to the OT network and mapping its devices, the attacker's next step is to gather enough information about how the ICS environment works and how its devices behave.

Point & Tag Identification (T0861)

The point & tag identification technique mentions that threat actors actively collect information values from the process environment to better understand the dynamics of the current operational process. This may include information like pressure level, rotations, and other common operational values used by machines and operators [41]. The continuous collection of operational data gives the attacker a better understanding of the operational process, how it works, and what the typical process behaviour is, including the range of values that are common for a given variable or set point value. In addition to providing critical operational data to the attacker, this information can also be helpful in later operations to disrupt the physical operational process. One real-world example is the Incontroller software, which can read OCP UA structures of remote devices [74].

From the perspective of the operational process, the collection of operational data inside of an ICS/OT environment is one of the most common operations done by OT devices because they need to know the current process values to take action to comply with the operational process constraints and conditions. As a result, detecting malicious operational data collection can be particularly difficult since it is a legitimate operation with a high occurrence frequency.

Analysing the test environment shows that two devices, the HMI and the PLC, are constantly collecting information by sending two specific Modbus commands to their targets: READ_COILS and READ_INPUT_REGISTERS. There are two particular relationships: the HMI is collecting information from the PLC, and the PLC is collecting information from the simulation machine.

With this information, it is possible to consider the HMI and the PLC trusted devices authorized to collect and read operational data. On the other hand, according to this baseline, any data collection activity from any other OT device than the HMI and PLC should be flagged as suspicious, which may include the operator workstation or other OT assets. For these reasons, a new baseline analytic was created, which monitors deviations in assets that use the Modbus network protocol, particularly sending Modbus data collection commands such as:

- READ_COILS (code 1)
- READ_DISCRETE_INPUTS (code 2)
- READ_MULTIPLE_HOLDING_REGISTERS (code 3)
- READ_INPUT_REGISTERS (code 4)

The operator machine will be used as the attacker's foothold in the OT network to validate this rule.

This test simulates an attacker who, once with access to a asset in the OT network, will start to collect operational data to perform further malicious actions. This also considers that the attacker already has mapped the OT network and knows its structure and which devices are available. The caldera OT module will be used to perform this test. This module contains multiple Caldera abilities that can interact with OT devices through various OT network protocols, including the Modbus protocol. There are four abilities related to data collection, each utilizing a specific Modbus command:

- Modbus Read Discrete Inputs
- Modbus Read Holding Registers
- Modbus Read Coils
- Modbus Read Input Registers

.

With the test plan defined, the following steps were performed:

- From the Caldera agent running in the operator's workstation, execute the above-enumerated Caldera abilities targeting both the PLC(192.168.95.2) and the simulation devices(192.168.95.10-15).
- Because the operator workstation is not part of the baseline of trusted devices that can collect operational data, a new alert is triggered.

5.3.8 Inhibit Response Function

With enough operational information collected, the final phase of the attack involves active interaction with OT assets to achieve the attacker's goal. The inhibit response function tactic contains techniques that aim to affect and inhibit safety processes and safeguards and even interrupt operational processes, potentially leading to many side effects.

Device Restart/Shutdown (T0816)

The device restart/shutdown technique states that threat actors with access to the OT environment and its equipment can forcefully stop key OT field devices like PLCs to cause disturbances in the operational process. Start, stop, and restart are common functionalities of these devices and are often available to operators through various interfaces, either via a specific ICS network protocol function, a web GUI, or a

command line[44]. Consequently, it is important to closely monitor these state variations because sudden changes, unrelated to maintenance tasks, can indicate malicious activity in the OT network.

In this test environment, the PLC located on the OT network has a web GUI service running on port 8080 that allows the operator, among other things, to start or stop the process operation, as shown in figure 36.

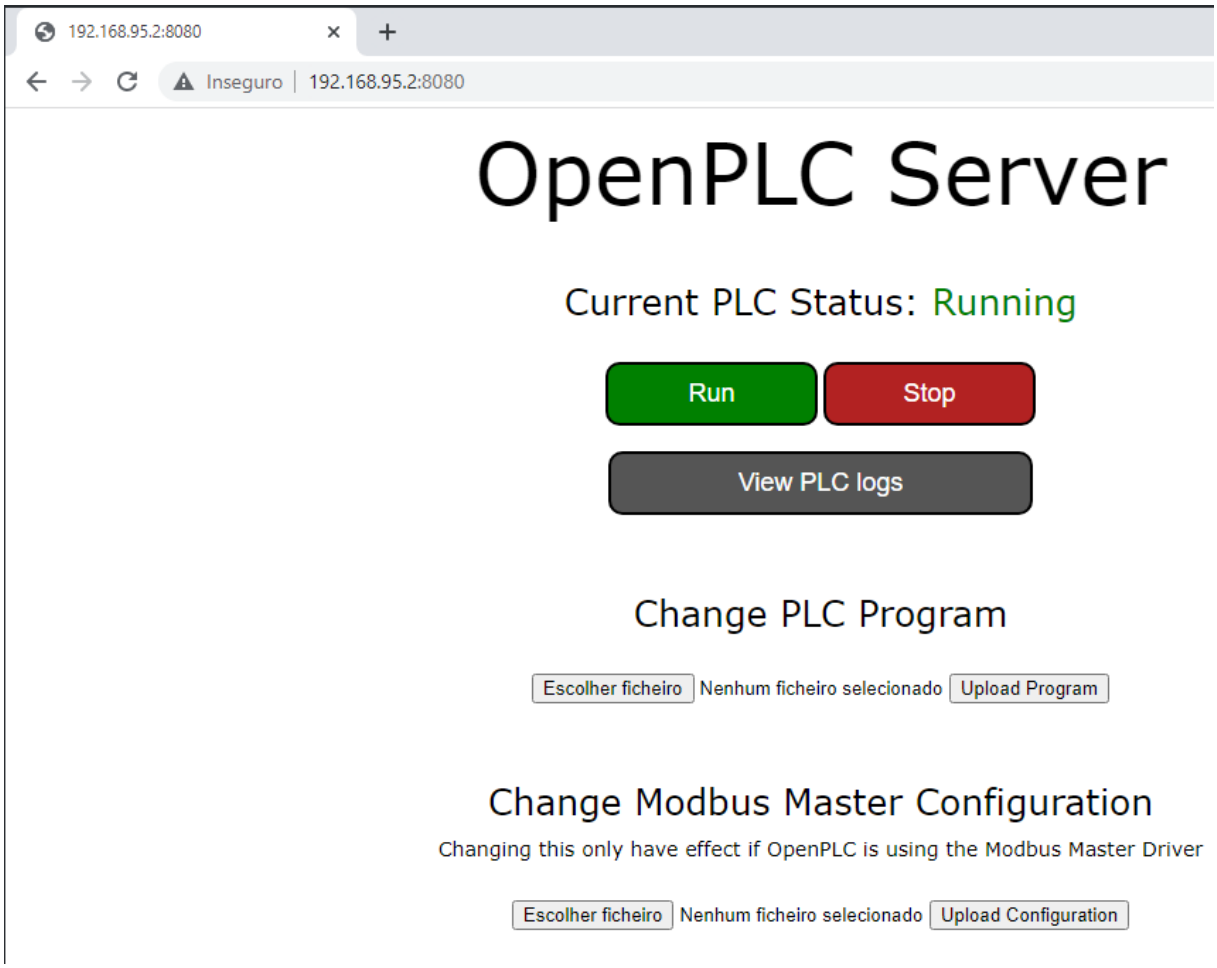


Figure 36: OpenPLC web UI interface

The start and stop functionality is available through two HTTP endpoints, `plc-ip:8000/run` and `plc-ip:8080/stop`, respectively. Since this functionality is available to operators, they will use it when necessary to complete maintenance tasks. Thus, monitoring these operations outside of the operator's business hours is crucial because otherwise false positive alerts will be issued that correspond to legitimate operations from the operator's work activity. In addition to this, the previous operational data collection technique revealed that the Modbus coil with reference number 40 controls whether the operational process is running or is stopped.

As a result, to detect state changes in the PLC, it is necessary to monitor these HTTP endpoints,

explicitly looking for stop and write Modbus coil commands.

With this information, two rules were created. The first is a threshold rule that monitors for the following Zeek events:

- Filter for only HTTP requests.
- The HTTP request destination address is the PLC (192.168.95.2), and the port is 8080.
- The HTTP request URL path matches the `/stop` endpoint.
- The request timestamp is outside of business hours (22h-06h).

The second rule is an EQL rule that monitors the following Zeek events:

- Filter for network requests using the Modbus protocol.
- Request destination is the PLC (192.168.95.2).
- Filter for the Modbus command WRITE_SINGLE_COIL or WRITE_MULTIPLE_COILS, against the coil with reference number 40.

To validate the first rule, the following steps were performed:

- Outside the business hours time frame, login in the operator workstation and send an HTTP request to the PLC stop endpoint, stopping the PLC execution.
- The rule triggers a new alert, as shown in figure 37.

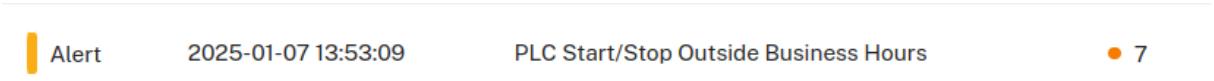


Figure 37: PLC start/stop outside business hours alert

To validate the second rule, the following steps were performed:

- From the Caldera agent running on the operator workstation, execute the "Modbus Write Single Coil" and "Modbus Write Multiple Coils" abilities.
- The rule triggers a new alert, as shown in figure 38.

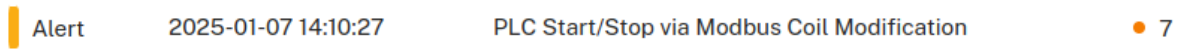


Figure 38: PLC start/stop via modbus coil modification alert

Finally, it should be noted that the first use case is based on a custom PLC management web interface, which is not representative of real-world field devices. However, many of these devices also lack strong access control mechanisms, making them easy targets for attackers.

In the second use case, using specific ICS protocol commands to start/stop OT field devices is typical behaviour in ICS/OT environments. In this case, a Modbus coil was used to control the state of the operational process. Still, there are other ICS protocols with even more specific commands to change the execution state of ICS devices, like the DNP3 stop application function code 18 and the cold restart function code 13.

5.3.9 Impair Process Control

The impair process control tactic consists of techniques that aim to affect the logic of the physical process that is being managed by ICS assets, directly influencing how the operation is conducted.

Unauthorized Command Message (T0855)

This technique shows that if threat actors have access to critical ICS assets, they can send unauthorized command messages to these devices to make them execute operations that are not defined in their role in the operational process [44]. These unauthorized operations can disrupt or even interrupt the physical process. Due to its impact, it is common to see it in real-world attacks. One such example is the Ukraine 2015 electric power incident, where the attackers sent unauthorized commands to breakers in substations, resulting in the disruption of the electrical service [67].

To identify unauthorized command messages in this test environment, it is necessary to determine which commands lead to legitimate actions by client devices and between ICS systems. For example, the only ICS asset that instructs other ICS assets to execute their operations is the PLC. The PLC sends Modbus commands, particularly the WRITE_MULTIPLE_REGISTERS command, to the simulation machine to comply with the predefined PLC program. As a result, in this environment, command messages are classified as unauthorized if they do not respect all the following conditions:

- The command message source is the PLC(192.168.95.2) and the destination is the simulation machine(192.168.95.10-15).

- The Modbus command is the WRITE_MULTIPLE_REGISTERS command (code 23).

Knowing what is expected and what to detect, the next step was to create a new baseline analytic that detects deviations in ICS assets that send Mobdus write commands to other assets, as well as their communication relationships. The Modbus write commands that will be monitored are the following:

- WRITE_SINGLE_COIL (code 5)
- WRITE_SINGLE_HOLDING_REGISTER (code 6)
- WRITE_MULTIPLE_COILS (code 15)
- WRITE_MULTIPLE_HOLDING_REGISTERS (code 16)
- READ/WRITE_MULTIPLE_REGISTERS (code 23)

As such, the baseline analytic has the following parameters:

- Filter for only network logs using the Modbus network protocol and the above-listed Modbus commands.
- Baseline the communication channel, which includes the source and destination IP addresses and the Modbus function code issued.

As shown in figure 39, by using the baseline analytic profiler utility, it is possible to verify that according to the baseline data, the PLC is the only asset that legitimately sends a write command.

After creation, the baseline analytic automatic alert mode was enabled. The operator machine will be used again as the OT network entry point from the attacker's perspective to validate this rule. The test will simulate an attacker trying to interfere directly with the simulation process, by sending unauthorized Modbus commands to the PLC or the field devices that directly influence the process. This requires that the attacker has previous knowledge of the operational process and how it usually works. This information can be gathered by collecting operational data, as shown in the point & tag identification technique description (Section 5.3.7).

To perform this test, two Caldera OT abilities were used: "Modbus Fuzz Registers" and "Modbus Fuzz Coils", and as their name implies, they use the WRITE_SINGLE_HOLDING_REGISTER and the WRITE_SINGLE_COIL Modbus commands, respectively. These two abilities can consistently write random values on a random number of coils/registers in the target ICS asset. The targets for this test are the PLC(192.168.95.2) and the six field devices (192.168.95.10-15).

Finally, for validation, the following steps were performed:

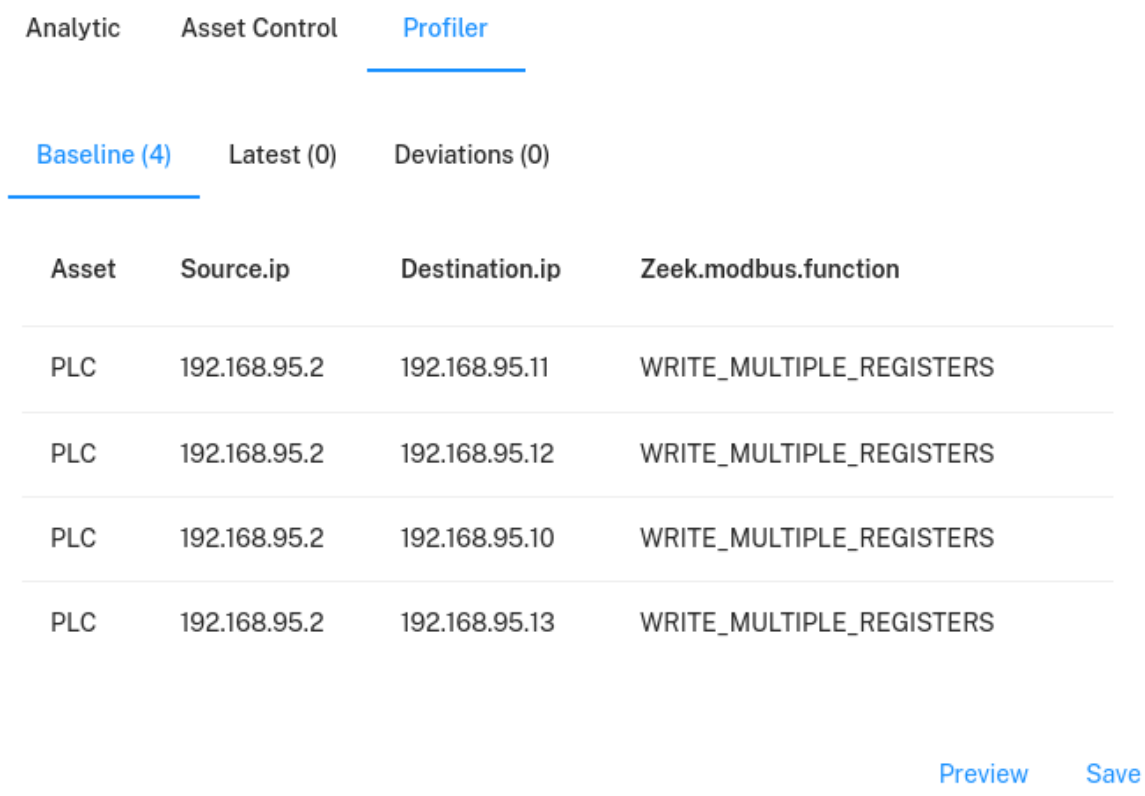


Figure 39: Modbus write command by asset profilling

1. From the Caldera agent running on the operation workstation, execute the two Caldera abilities against each of the selected targets.
2. For each target, a new alert was triggered, as shown in figure 40.



Figure 40: Modbus unauthorized write baseline analytic alert

5.3.10 Impact

Attackers use all the techniques of the Impact Tactic in the very last phase of the attack, as they all negatively impact the ICS/OT environment and its operational process. They lead to damage of property and industrial assets, interruption of operations, losses in control, safety, availability and visibility.

Manipulation of Control(T0831)

The manipulation of control technique lists the different methods threat actors use to directly interfere with the physical process by manipulating the core process parameters, highly impacting the physical process due to incorrect operation. The outcome of this manipulation can go from mild side effects to a completely unstable process that can cause physical destruction [39].

This can be achieved by spoofing command messages to ICS devices, altering the process state, or changing the operational process original parameters or set point values. Additionally, it is related to other techniques like the "Modify Program" technique that consists in modifying a controller or program behaviour like a PLC to affect the operational process, which leads to a manipulation of control. An example of this technique is the Stuxnet software that can reprogram a target PLC to ignore legitimate commands from operators and modify critical parameters of the operational process [52].

In this test environment, the operator can reprogram the PLC through a specific function in the PLC web GUI service by allowing the operator to upload a PLC program file that will be compiled and applied to the simulation process. This program contains the logic, parameters, and set point values the simulation process must obey. Analysing the reprogramming functionality available in the web service shows that it is done by uploading a program file to the `plc-ip:8080/api/upload` HTTP endpoint.

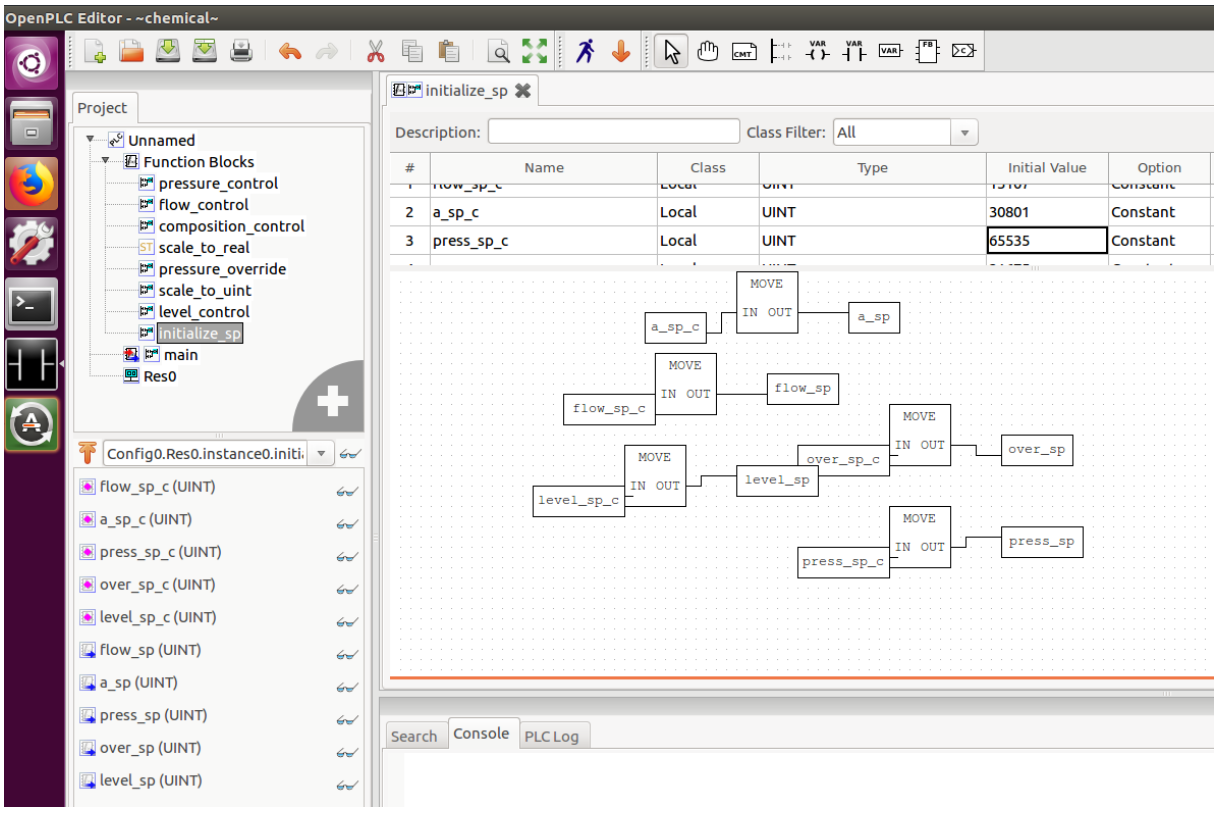


Figure 41: OpenPLC control logic parameters modification

The original PLC program can be edited in the OpenPLC editor, which is also available in the operator workstation. As a result, an attacker with access to the original PLC program can modify it in a way that disrupts the operational process.

As shown in figure 41, the existing PLC program set point values for key process parameters, such as tank pressure levels, were changed to a much higher value of 65535 instead of the original 55295.

This change will affect how the simulation process reacts to higher pressure values, allowing the process to reach pressure value ranges unsupported by the tanks, resulting in a tank explosion and plant destruction, as shown in figure 42.



Figure 42: Operational process physical destruction

To detect when the PLC is reprogrammed, a new EQL rule was created that monitors Zeek events that:

- Are HTTP-only requests.
- The HTTP request destination address is the PLC (192.168.95.2), and the port is 8080.
- The HTTP request method is POST.
- The HTTP request URL path matches the `/api/upload` endpoint.

To validate this rule, the following steps were performed:

- Login to the operator workstation.
- Open the original PLC program in the OpenPLC editor.
- Modify the tank pressure level set point (press_sp_c) to 65535.
- Open the PLC web GUI page and upload the malicious program.
- The rule triggers a new alert, as shown in figure 43.

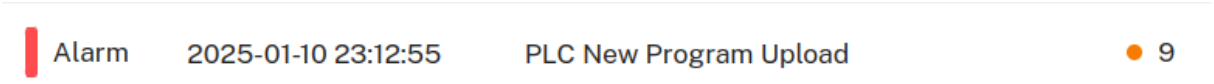


Figure 43: PLC new program upload modification alert

It is important to note that although the program upload functionality is available to the operator, it is a very sensitive operation that should constantly be closely monitored, as mistakes made when changing the original PLC program can lead to catastrophic consequences, as mentioned above. Additionally, one could set the detection rule time restriction to periods outside of business hours to completely mitigate potential false positives generated by legitimate actions by the operator.

5.3.11 Other Use Cases

This section presents very specific use cases not mapped by the ICS ATT&CK matrix but relevant enough to be monitored in ICS/OT environments, as they can indicate abnormal or suspicious activity inside that network that might be related to an ongoing intrusion.

Never seen assets

ICS/OT networks are known to be very static, characterized by a rigid and predictable architecture, with a consistent set of connected devices. Changes in this type of network occur occasionally due to the replacement/addition of equipment due to new requirements of the physical process or just for maintenance reasons. Consequently, it is essential to actively monitor these sensitive networks to know when new assets or devices have been added to the network, as this provides the necessary information to evaluate if a given asset was adequately authorized to access and be part of the network. As a result, this use case aims to detect when new unauthorized devices or assets are added to the ICS/OT network. This

is very important since placing unauthorized rogue devices in an operational network provides a fast and direct line of communication with sensitive ICS assets that directly control and execute the operational process. This use case is possible if the organisation lacks or has ineffective physical access controls to personnel.

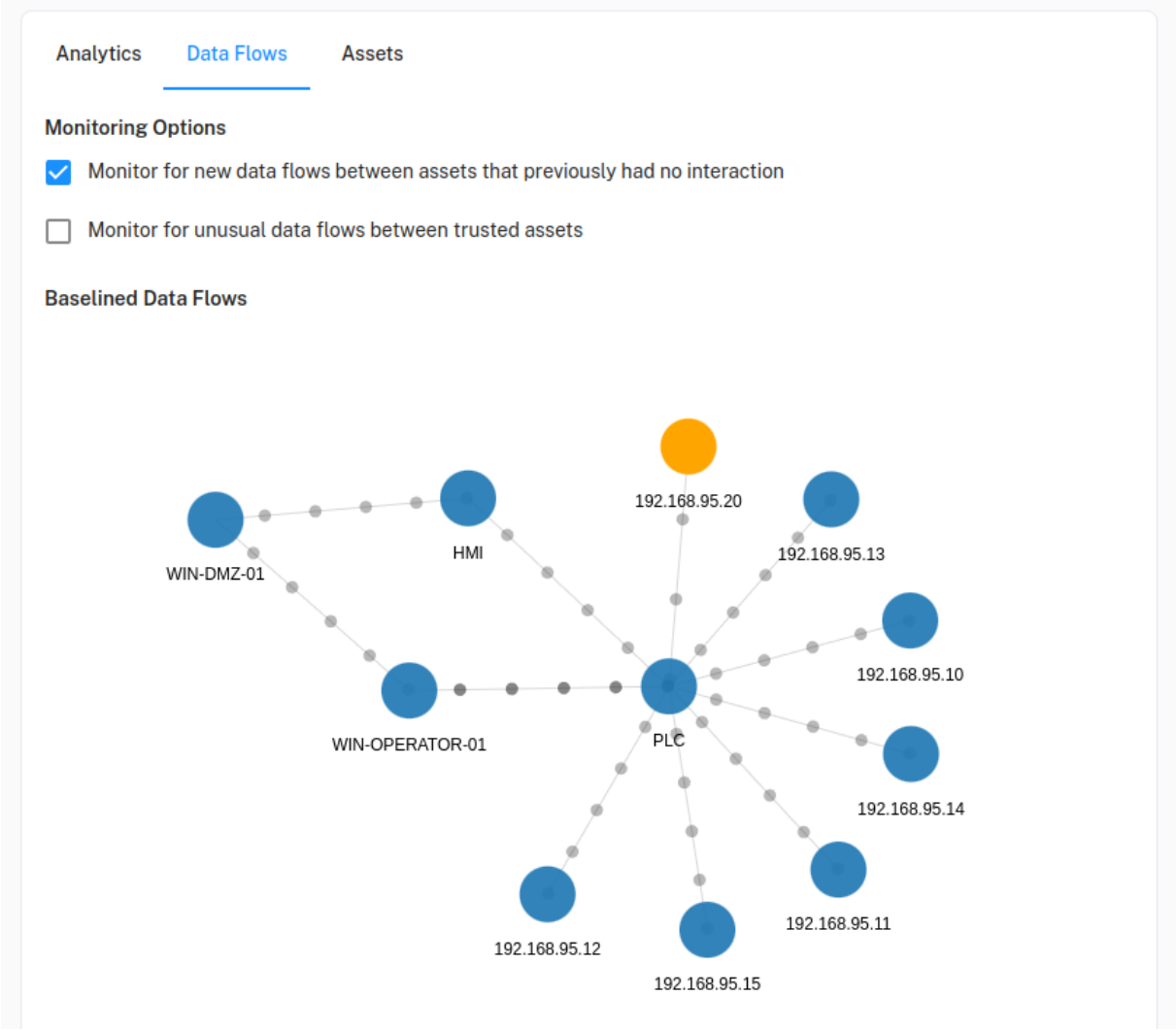


Figure 44: Unknown asset discovery alert

As mentioned earlier, the asset monitoring module tracks all devices in all available networks or specific monitored subnets. The baseline module has a native asset monitoring rule that, once enabled, will monitor network-related data sources like Zeek for new assets added to the selected networks, triggering a new alert every time a new unknown asset appears in the network.

It is important to note that in industrial environments that adopt IoT technologies and assets, new devices may be added regularly, which may cause multiple false positives. As such, it is essential to segregate these devices into specific IoT networks that are excluded from the monitored list.

To validate this baseline rule, the following steps were performed:

1. Enable the "New Asset" baseline module option, which monitors the DMZ and OT network.
2. Add a new device to the OT network. The new device received the 192.168.95.20 IP address.
3. The rule triggers a new alert that indicates the presence of an unknown asset.

Beyond the generation of an alert, the data flow graph also displays unknown assets that have open alerts, as presented in figure 44, providing informative visibility into the network.

Uncommon asset communication interactions

In another use case, sometimes it is impossible to detect whether the PLC was reprogrammed or if some action has modified the behaviour of some field devices, altering the original operational process execution. Because the activity in ICS environments is consistent with almost fixed patterns of communication and actions, it becomes possible to baseline the overall ICS execution behaviour, allowing the detection of highly suspicious deviations from the normal operational course of action. This is important because threat actors can make subtle changes in the operational process to not alarm field operators and maintain continuous access to the ICS/OT environment. This can then result in consistent operational defects or sudden loss in availability.

To prevent these actions, it is necessary to detect deviations in the ICS/OT network activity at different levels, such as:

- Communication flows between assets that have never interacted before.
- New network protocols are used between trusted systems.
- Unusual communication flows between trusted assets much lower/higher than the recorded baseline.
- Use of new network protocol functions from trusted assets, as shown before in the unauthorized command message technique.

Individual deviations of the above indicators alone can not be fully trusted to detect suspicious network activity, but by combining them, detecting deviations in most of them within the same time frame can be a strong indicator of an intrusion in the ICS/OT network.

Consequently, a combination of baseline analytics is required to monitor these behaviours. Two out of the three baseline analytics that will be used in this case have already been created and used in the earlier tests:

- `network.protocol` - Monitors the network protocols used by each asset.
- `network.protocol.modbus.functions` - Monitors the network protocol functions used by each asset, such as the Modbus functions that a given asset typically executes. This can include any other ICS protocol.
- `network.communications` - Monitor communications interactions between assets, mapping who communicates with whom.

Furthermore, the baseline module can also automatically create threshold rules based on the observed communications flows between assets. This allows the detection of abnormal communication patterns caused by the network activity produced by the attacker's operations, such as: internal reconnaissance, exfiltration and interaction with other ICS assets.

Finally, a new EQL rule was created based on these indicators. It monitors the baseline data source for multiple deviations in a single asset that must match at least two above indicators in a given time frame.

One final point is that this case is entirely based on consistent and predictable operational process activity. As a result, after any major change in the operational process, a new baseline must be recorded to represent, effectively, the new way the process works. If left unchanged, the number of false positives will likely increase.

Chapter 6

Conclusions and future work

This final chapter presents the main conclusions obtained from the completion of the dissertation, as well as proposals for future work to secure modern ICS/OT environments.

6.1 Conclusions

The main objective of the work developed throughout this dissertation was to create a capable and adaptable security solution to protect and detect attacks in both IT and ICS/OT environments.

To achieve this objective, the first step was to study and understand how ICS/OT environments work, their evolution in recent times, how they now integrate with the enterprise side of organisations, and, ultimately, how they can be protected and secured. Following this, research was carried out to identify security systems and their supporting technologies capable of monitoring and detecting malicious behaviour in enterprise and industrial environments, as well as supporting frameworks like MITRE ATT&CK. Based on research knowledge and with the requirements identified in section 3.1, the final decision was the development of the Argus SIEM, integrated with a log and event collection pipeline supported by the Beats data shipper and the Elasticsearch search engine, which will deal mainly with system and applications logs coming from sources like Windows workstations, and network events collected by the powerful Zeek security monitor tool.

Through the development of the Argus SIEM and its different components, multiple challenges were overcome and many of the earlier technology-based decisions were reinforced. The choice of Elasticsearch as a search engine, log aggregation, and storage solution was a crucial decision for the project success due to its powerful and flexible search and log analysis capabilities that the Argus SIEM relies upon. Elasticsearch is also scalable and performant, which is crucial for industrial environments that produce high volumes of events.

The Beats data shipper proved to be an ideal solution due to its seamless integration with Elasticsearch

and its robust data collection capabilities. It effortlessly captures data from a diverse range of relevant data sources, supporting various technologies and products commonly used in enterprise environments without any additional configuration. Furthermore, it offers the ability to retrieve any custom information from target systems.

Finally, concerning technology-based decisions, the Zeek network security monitoring tool revealed itself as a core component of the system, being not only one of the most important and used data sources for the detection rules and analytics but at the same time, feeding other Argus subcomponents, namely the asset monitoring and baseline module, to provide better understanding of an organisation network. As a result, Zeek is one of the essential tools that can be used to secure ICS/OT environments due to its network event capabilities, especially when it is only possible to collect network data from an industrial environment.

Concerning the Argus SIEM itself, the system compatibility with standard signature formats like Sigma was vital since it allows the operators to easily import external and community-shared rules into the detection system without any effort. Creating multiple rule types also provided enough flexibility to cover different use cases and ways to analyse events. The event correlation capabilities are also a fundamental aspect of the SIEM as they provide different ways to correlate events from multiple sources, giving security personnel the ability to create detection rules capable of detecting complex attack scenarios. The alert system and the event analysis tools also provide a quick way to fully analyse the logs that caused the alert to be issued with additional contextual and organisational information, enhancing the analysis quality and efficiency.

Furthermore, the creation of the asset monitoring and baseline modules greatly enhanced the system information coverage, quality, contextualization, and detection capabilities. In the case of the asset monitoring module, the asset information collected and presented by the module allowed security personnel to know exactly what assets were present on the organisation network, which events data sources are collected for each one, and visualize previous security issues recorded by the detection system. The baseline module proved to be one of the most important supporting modules, improving the base detection capabilities of the SIEM as its analytics can be used to model common and custom activity behaviours to detect baseline deviations, which is an essential detection analysis mechanism when dealing with ICS/OT environments.

Finally, the created test environment allowed, in conjunction with an elaborate test plan, a comprehensive evaluation of the Argus SIEM detection capabilities. Through the results section, multiple techniques and scenarios were tested showcasing the Argus SIEM ability to detect threat activity at each level of the

Purdue model, proving its usefulness in protecting IT and ICS/OT environments.

6.2 Prospects for future work

Safeguarding ICS/OT networks and their assets will become more difficult due to the ongoing evolution of industrial environments and the integration of new technologies in these settings. As a result, security solutions that help protect these environments must continue to improve their capabilities to deal with the increasing complexity.

In terms of system integrations, future versions of Argus could incorporate threat intelligence data sources like the MISP platform, providing the latest threat data collected in the wild that can be used to detect known threat activity automatically. Another option would be the integration of the current passive asset monitoring module with dedicated industrial asset inventory systems that provide much more detailed and complete asset information.

In the case of system functionality, enhance system rules by providing a wider selection of rule logic to better cover each use case. New rule types that use new search languages, for example, the brand-new Elasticsearch Query Language (ES|QL), could be added. The existing analysis tools, namely the Profiler, could be improved to provide better insights about system and network activity.

Furthermore, in later iterations, new ICS protocol-aware modules could be created to provide advanced analysis abilities to monitor how assets behave when using these specific protocols.

Finally, one of the most apparent goals would be the addition of Artificial Intelligence (AI) capabilities to the core of Argus SIEM. This would be especially important in the case of the baseline module, where AI could significantly improve how network activity is modelled and analysed, increasing alert accuracy, among other things. The adoption of AI is already being implemented by the most notorious security system providers, such as Fortinet, CheckPoint, Trend Micro, and others.

Bibliography

[1] O. Alexander, M. Belisle, and J. Steele. Mitre att&ck for industrial control systems: Design and philosophy. *The MITRE Corporation: Bedford, MA, USA*, pages 1–14, 2020. URL https://attack.mitre.org/docs/ATTACK_for_ICS_Philosophy_March_2020.pdf.

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

[3] J. v. Brocke, A. Hevner, and A. Maedche. *Introduction to Design Science Research*, pages 1–13. Springer, 09 2020. ISBN 978-3-030-46780-7. doi: 10.1007/978-3-030-46781-4_1.

[4] Caldera. Github - mitre/caldera: Automated adversary emulation platform. URL https://github.com/mitre/caldera. [Accessed 21-03-2024].

[5] A. Cherepanov. Win32/industroyer a new threat for industrial control systems, 2017. URL https://web-assets.esetstatic.com/wls/2017/06/Win32_Industroyer.pdf.

[6] E. Chew, M. Swanson, K. M. Stine, N. Bartol, A. Brown, and W. Robinson. Performance measurement guide for information security. Technical report, National Institute of Standards and Technology, 2008. URL https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-55r1.pdf.

[7] CISA. Industrial control systems cybersecurity initiative: Considerations for ics/ot monitoring technologies with an emphasis on detection and information sharing. URL https://www.cisa.gov/sites/default/files/publications/ICS-Monitoring-Technology-Considerations-Final-v2_508c.pdf.

[8] K. Curtis. A dnp3 protocol primer. URL https://www.dnp.org/Portals/0/AboutUs/DNP3%20Primer%20Rev%20A.pdf. [Accessed 11-09-2023].

[9] Dragos. Trisis malware analysis of safety system targeted malware. *Dragos Inc.: Hanover, MD, USA*, 2017. URL https://www.dragos.com/wp-content/uploads/TRISIS-01.pdf.

[10] D. P. Duggan. Penetration testing of industrial control systems, 2005. URL https://energy.sandia.gov/wp-content/gallery/uploads/sand_2005_2846p.pdf.

[11] Elastic. Beats: Data shippers for elasticsearch | elastic, . URL https://www.elastic.co/beats. [Accessed 16-09-2023].

[12] Elastic. Getting started | elastic common schema (ecs) reference [8.10] | elastic, . URL https://www.elastic.co/guide/en/ecs/current/ecs-getting-started.html. [Accessed 17-09-2023].

[13] Elastic. Eql search | elasticsearch guide [8.10] | elastic, . URL https://www.elastic.co/guide/en/elasticsearch/reference/current/eql.html. [Accessed 28-09-2023].

[14] Elastic. Elasticsearch features list | elastic, . URL https://www.elastic.co/elasticsearch/features. [Accessed 16-09-2023].

[15] Elastic. Filebeat: Lightweight log analysis and elasticsearch | elastic, . URL https://www.elastic.co/beats/filebeat. [Accessed 16-09-2023].

[16] Elastic. initial access suspicious ms office child process, . URL https://github.com/elastic/detection-rules/blob/main/rules/windows/initial_access_suspicious_ms_office_child_process.toml. [Accessed 05-04-2024].

[17] Elastic. Logstash: Collect, parse, transform logs | elastic, . URL https://www.elastic.co/logstash. [Accessed 16-09-2023].

[18] Elastic. Painless scripting language | elasticsearch guide [8.10] | elastic, . URL https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-scripting-painless.html. [Accessed 28-09-2023].

[19] Elastic. Query dsl | elasticsearch guide [8.10] | elastic, . URL https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html. [Accessed 28-09-2023].

[20] Elastic. Startup or run key registry modification, . URL https://github.com/elastic/detection-rules/blob/main/rules/windows/persistence_run_key_and_startup_broad.toml. [Accessed 12-04-2024].

[21] Elastic. detection-rules/rules/windows/lateral_movement_executable_tool_transfer_smb.toml, . URL https://github.com/elastic/detection-rules/blob/main/rules/windows/lateral_movement_executable_tool_transfer_smb.toml. [Accessed 16-04-2024].

[22] Elastic. Winlogbeat overview | winlogbeat reference [8.9] | elastic, . URL https://www.elastic.co/guide/en/beats/winlogbeat/current/_winlogbeat_overview.html. [Accessed 16-09-2023].

[23] D. Formby, M. Rad, and R. Beyah. Lowering the barriers to industrial control system security with GRFICS. In *2018 USENIX Workshop on Advances in Security Education (ASE 18)*, Baltimore, MD, Aug. 2018. USENIX Association. URL https://www.usenix.org/conference/ase18/presentation/formby.

[24] Y. Geng, Y. Wang, W. Liu, Q. Wei, K. Liu, and H. Wu. A survey of industrial control system testbeds. *IOP Conference Series: Materials Science and Engineering*, 569(4):042030, jul 2019. doi: 10.1088/1757-899X/569/4/042030. URL https://dx.doi.org/10.1088/1757-899X/569/4/042030.

[25] G. González-Granadillo, S. González-Zarzosa, and R. Diaz. Security information and event management (siem): Analysis, trends, and usage in critical infrastructures. *Sensors*, 21:8–10,20, 7 2021. ISSN 14248220. doi: 10.3390/s21144759.

[26] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun. A survey of intrusion detection on industrial control systems. *International Journal of Distributed Sensor Networks*, 14:2–44, 8 2018. ISSN 15501477. doi: 10.1177/1550147718794615.

[27] P. Institute. The state of iot/ot cybersecurity in the enterprise. Technical report, Ponemon Institute, 2021.

[28] ISA. Quick start guide: An overview of isa/iec 62443 standards, 2023. URL https://gca.isa.org/hubfs/ISAGCA%20Quick%20Start%20Guide%20FINAL.pdf.

[29] M. Kaouk, J.-M. Flaus, M.-L. Potet, and R. Groz. A review of intrusion detection systems for industrial control systems. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1699–1704, 2019.

[30] B. Leander, A. Čaušević, and H. Hansson. Applicability of the iec 62443 standard in industry 4.0 / iiot. In *ACM International Conference Proceeding Series*. Association for Computing Machinery, 8 2019. ISBN 9781450371643. doi: 10.1145/3339252.3341481.

[31] M. Lee, J. Jang-Jaccard, and J. Kwak. Novel architecture of security orchestration, automation and response in internet of blended environment. *Computers, Materials and Continua*, 73:199–223, 2022. ISSN 15462226. doi: 10.32604/cmc.2022.028495.

[32] Microsoft. Sysinternals. URL https://learn.microsoft.com/en-us/sysinternals/. [Accessed 12-04-2024].

[33] Microsoft. Ot threat monitoring in enterprise socs, 2023. URL https://learn.microsoft.com/en-us/azure/defender-for-iot/organizations/concept-sentinel-integration. [Accessed 27-03-2024].

[34] A. W. Mir and R. K. Ramachandran. Implementation of security orchestration, automation and response (soar) in smart grid-based scada systems. In S. S. Dash, B. K. Panigrahi, and S. Das, editors, *Sixth International Conference on Intelligent Computing and Applications*, pages 157–169, Singapore, 2021. Springer Singapore. ISBN 978-981-16-1335-7.

[35] MITRE. Getting started with att&ck: Assessments and engineering, . URL https://medium.com/mitre-attack/getting-started-with-attack-assessment-cc0b01769cb4. [Accessed 28-03-2024].

[36] MITRE. Create or modify system process: Windows service, sub-technique t1543.003 - enterprise | mitre att&ck, . URL https://attack.mitre.org/techniques/T1543/003/. [Accessed 12-04-2024].

[37] MITRE. Lateral tool transfer, technique t0867 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0867/. [Accessed 16-04-2024].

[38] MITRE. Os credential dumping: Lsass memory, sub-technique t1003.001 - enterprise | mitre att&ck, . URL https://attack.mitre.org/techniques/T1003/001/. [Accessed 21-04-2024].

[39] MITRE. Manipulation of control, technique t0831 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0831/. [Accessed 02-05-2024].

[40] MITRE. Brute force: Password guessing, sub-technique t1110.001 - enterprise | mitre att&ck, . URL https://attack.mitre.org/techniques/T1110/001/. [Accessed 18-04-2024].

[41] MITRE. Point & tag identification, technique t0861 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0861/. [Accessed 26-04-2024].

[42] MITRE. Boot or logon autostart execution: Registry run keys / startup folder, sub-technique t1547.001 - enterprise | mitre att&ck, . URL https://attack.mitre.org/techniques/T1547/001/. [Accessed 12-04-2024].

[43] MITRE. Remote services, technique t0886 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0886/. [Accessed 15-04-2024].

[44] MITRE. Unauthorized command message, technique t0855 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0855/. [Accessed 28-04-2024].

[45] MITRE. Valid accounts, technique t0859 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0859/. [Accessed 15-04-2024].

[46] MITRE. Attacknavigator, . URL https://mitre-attack.github.io/attack-navigator/. [Accessed 28-03-2024].

[47] MITRE. Remote system discovery, technique t0846 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0846/. [Accessed 08-04-2024].

[48] MITRE. Run and runonce registry keys - win32 apps | microsoft learn, . URL https://learn.microsoft.com/en-us/windows/win32/setupapi/run-and-runonce-registry-keys. [Accessed 12-04-2024].

[49] MITRE. Spearphishing attachment, technique t0865 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0865/. [Accessed 04-04-2024].

[50] MITRE. Standard application layer protocol, technique t0869 - ics | mitre att&ck, . URL https://attack.mitre.org/techniques/T0869/. [Accessed 07-04-2024].

[51] NCCIC and ICS-CERT. Recommended practice: Improving industrial control system cybersecurity with defense-in-depth strategies industrial control systems cyber emergency response team. Technical report, Department of Homeland Security, 2016.

[52] L. M. Nicolas Falliere and E. Chien. W32.stuxnet dossier, 2011. URL https://www.wired.com/images_blogs/threatlevel/2011/02/Symantec-Stuxnet-Update-Feb-2011.pdf.

[53] K.-H. Niemann. Differentiation of the it security standard series iso 27000 and iec 62443 a view of automation systems in the manufacturing and process industries. White paper, ABB AG, Heidelberg, Germany, 2021. URL https://doi.org/10.25968/opus-1973.

[54] P. Radoglou-Grammatikis, P. Sarigiannidis, E. Iturbe, E. Rios, S. Martinez, A. Sarigiannidis, G. Eftathopoulos, Y. Spyridis, A. Sesis, N. Vakakis, D. Tzovaras, E. Kafetzakis, I. Giannoulakis, M. Tzifas, A. Giannakoulias, M. Angelopoulos, and F. Ramos. Spear siem: A security information and event management system for the smart grid. *Computer Networks*, 193, 7 2021. ISSN 13891286. doi: 10.1016/j.comnet.2021.108008.

[55] M. H. Samir Bousseaden. proc_access_win_lsass_memdump.yml. URL https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_access/proc_access_win_lsass_memdump.yml. [Accessed 21-04-2024].

[56] H. Santos, A. Oliveira, L. Soares, A. Satis, and A. Santos. Information security assessment and certification within supply chains. In *The 16th International Conference on Availability, Reliability and Security (ARES 21), Vienna, Austria, August 17 - 20, 2021*, pages 1–6. ACM, 8 2021. ISBN 9781450390514. doi: 10.1145/3465481.3470078. URL https://dl.acm.org/doi/10.1145/3465481.3470078.

[57] H. M. D. Santos. *Cybersecurity : A Practical Engineering Approach*, pages 13–19. Chapman and Hall/CRC, 4 2022. ISBN 9780429286742. doi: 10.1201/9780429286742.

[58] SecAware. About the iso27k standards. URL https://www.iso27001security.com/html/iso27000.html. [Accessed 11-09-2023].

[59] SigmaHQ. Sigmahq/sigma: Main sigma rule repository. URL https://github.com/SigmaHQ/sigma/. [Accessed 26-03-2024].

[60] J. Slowik. Anatomy of an attack: Detecting and defeating crashoverride. Technical report, Dragos, Inc., 2018. URL https://www.dragos.com/wp-content/uploads/CRASHOVERRIDE2018.pdf.

[61] J. Slowik. Evolution of ics attacks and the prospects for future disruptive events. *Threat Intelligence Centre Dragos Inc*, page 1, 2019.

[62] J. Slowik. The baffling berserk bear: A decade's activity targeting critical infrastructure. Technical report, Virus Bulletin, 2021. URL https://vblocalhost.com/uploads/VB2021-Slowik.pdf.

[63] Statseeker. The it/ot convergence - bridging the gap, 2015. URL https://www.isssource.com/wp-content/uploads/2015/11/111115statseeker-IT-OT-Convergence-White-Paper.pdf.

[64] D. K. Z. Steve Miller, Nathan Brubaker and D. Caban. Triton actor ttp profile, custom attack tools, detections, and att&ck mapping. Technical report, FireEye, 4 2019. URL https://www.mandiant.com/resources/blog/triton-actor-ttp-profile-custom-attack-tools-detections. [Accessed 18-04-2024].

[65] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, and S. Lightman. *Draft NIST SP 800-82r3, Guide to Operational Technology (OT) Security*, pages 3,7–15,45, 129, 132. National Institute of Standards and Technology, 2022. doi: 10.6028/NIST.SP.800-82r3.ipd. URL https://doi.org/10.6028/NIST.SP.800-82r3.ipd.

[66] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018.

[67] J. STYCZYNSKI and N. BEACH-WESTMORELAND. When the lights went out - a comprehensive review of the 2015 attacks on ukrainian critical infrastructure. Technical report, Booz Allen Hamilton, 2019. URL https://www.boozallen.com/content/dam/boozallen/documents/2016/09/ukraine-report-when-the-lights-went-out.pdf.

[68] SwiftOnSecurity. Github - swiftonsecurity/sysmon-config: Sysmon configuration file template with default high-quality event tracing. URL https://github.com/SwiftOnSecurity/sysmon-config. [Accessed 03-03-2024].

[69] R. Sébastien. metasploit-framework/modules/exploits/windows/fileformat/word_msdtjs_rce.rbat. URL https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/fileformat/word_msdtjs_rce.rb. [Accessed 05-04-2024].

[70] S. T. R. Team. You bet your lsass: Hunting lsass access | splunk. URL https://www.splunk.com/en_us/blog/security/you-bet-your-lsass-hunting-lsass-access.html. [Accessed 21-04-2024].

[71] A. Volkova, M. Niedermeier, R. Basmadjian, and H. D. Meer. Security challenges in control network protocols: A survey. *IEEE Communications Surveys and Tutorials*, 21:619–639, 1 2019. ISSN 1553877X. doi: 10.1109/COMST.2018.2872114.

[72] S. Westfall. Threat brief: Cve-2022-30190 – msdt code execution vulnerability. URL https://unit42.paloaltonetworks.com/cve-2022-30190-msdt-code-execution-vulnerability/. [Accessed 04-04-2024].

[73] WireShark. S7comm - wireshark wiki. URL https://wiki.wireshark.org/S7comm. [Accessed 11-09-2023].

[74] J. Wylie. Analyzing pipedream: Challenges in testing an ics attack toolkit, 2022. URL https://media.defcon.org/DEF%20CON%2030/DEF%20CON%2030%20presentations/Jimmy%20Wylie%20-%20Analyzing%20PIPEDREAM%20Challenges%20in%20testing%20an%20ICS%20attack%20toolkit.pdf.

[75] C.-S. Yeh, S.-L. Chen, and I.-C. Li. Implementation of mqtt protocol based network architecture for smart factory. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 235(13):2132–2142, 2021. doi: 10.1177/09544054211014488. URL https://doi.org/10.1177/09544054211014488.

[76] Zscaler. What is the purdue model for ics security. URL https://www.zscaler.com/resources/security-terms-glossary/what-is-purdue-model-ics-security. [Accessed 11-09-2023].

This work is supported by the European Regional Development Fund (ERDF), through the Portugal 2020 Operational Competitiveness and Internationalization Programme (POCI), with reference ADM.IN/CALG-01/2022.