1 2 9 0

UNIVERSIDADE Đ COIMBRA

João Pedro Menoita Henriques

AUDIT COMPLIANCE AND FORENSICS FRAMEWORKS FOR IMPROVED CRITICAL INFRASTRUCTURE PROTECTION

Tese no âmbito do Doutoramento em Engenharia Informática, Especialidade em Arquiteturas, Redes e Cibersegurança, orientada pelo Professor Doutor Paulo Simões, pelo Professor Doutor Filipe Caldeira e pelo Professor Doutor Tiago Cruz, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Agosto de 2023

1 2 UNIVERSIDADE D COIMBRA 9 0

DEPARTMENT OF INFORMATICS ENGINEERING
FACULTY OF SCIENCES AND TECHNOLOGY
UNIVERSITY OF COIMBRA

AUDIT COMPLIANCE AND FORENSICS FRAMEWORKS FOR IMPROVED CRITICAL INFRASTRUCTURES PROTECTION

João Pedro Menoita Henriques

Doctoral Program in Informatics Engineering
PhD Thesis submitted to the University of Coimbra

Advised by Prof. Dr. Paulo Simões, Prof. Dr. Filipe Caldeira and Prof. Dr. Tiago Cruz

August, 2023

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

DEPARTMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

AUDIT COMPLIANCE AND FORENSICS FRAMEWORKS FOR IMPROVED CRITICAL INFRASTRUCTURE PROTECTION

João Pedro Menoita Henriques

Doutoramento em Engenharia Informática
Tese de Doutoramento apresentada à Universidade de Coimbra

Orientado pelo Prof. Dr. Paulo Simões, pelo Prof. Dr. Filipe Caldeira e pelo Prof. Dr. Tiago Cruz

Agosto, 2023

This work was co-supported by the European Commission under the H2020 ATENA project (Advanced Tools to assEss and mitigate the criticality of ICT compoNents and their dependencies over Critical InfrAstructures), with reference 7005813, H2020-DS-2015-1. Co-support was also provided by National Funds through the FCT—Foundation for Science and Technology, I.P., and the European Social Fund, through the Regional Operational Program Centro 2020, within the scope of the projects UIDB/05583/2020 and CISUC UID/CEC/00326/2020. It was also co-funded by FEDER, in the context of the Competitiveness and Internationalization Operational Program (COMPETE 2020) of the Portugal 2020 framework, in the scope of the Smart5Grid (POCI-01-0247-FEDER-047226) project and also by the "Agenda Mobilizadora Sines Nexus" project (ref. No. 7113), supported by the Recovery and Resilience Plan (PRR) and by the European Funds Next Generation EU, following Notice No. 02/C05-i01/2022, Component 5 - Capitalization and Business Innovation - Mobilizing Agendas for Business Innovation.

Acknowledgments

The conclusion of this work is an important milestone in my life. It results from more than twenty years on a non so usual journey, marked by many moments of hard work and huge challenges under different dimensions: family, professional, academic, and research. Has been a huge privilege sharing so many moments with so many people, family, friends, colleagues and teachers.

Firstly, I would like to thank the advisory, mentorship, patience, and support from Paulo Simões, Filipe Caldeira, and Tiago Cruz.

I would like to thank my lovely wife, Sandra Henriques for trusting me, and without whom was not possible to conclude this work. Many thanks to my little daughter Maria (I'm glad that you came) and my wonderful sons, Gabriel, and Diogo for making me happy and for their encouragement and support in the past years. They are also responsible for the strength put in this hard work.

Likewise, I am also grateful to my parents Maria Clara and my father António Henriques for their full support in all the moments of my life. Since April 1, 1998, my father is always present in my thoughts.

Because my family is my blessing, so I would like to thank all my mom's family from Carpinteiro, Casal de Cinza, Guarda, such as my grandmother Clara, my grandfather Joaquim and aunts, Otilia, Elvira, Generosa, Estela and Rodrigues and their daughter Joana. Also thanks to my brother Luis Miguel and his wife Catarina and their sons: Afonso, Alice, and António. I would also like to thank all my father´s family from Pousadas, Castelo de Penalva, Penalva do Castelo, Viseu, such as my aunts Carlos and Alice, Matilde, Cecilia, and my cousins, António and Manuela, Augusta and Carlos, José Manuel and Cláudia, Américo, Paula and Eduardo, Fátima and Romeu, João and Cândida, Lena and José, Maria Zé and Zé António, and the other generation Pedro, Carlos, Elizabete, Sandra, Luis, Susana, Bruno, Gabriela, Inês, Helder, and André.

From my wife´s family thanks to their parents Mateus and Fátima (also a mother I lost recently), João and Paula, Manuel and Antónia, and their sons, Vanda, Inês, João Paulo and Fábio.

I also want to thank all my colleagues and friends at the Polytechnic of Viseu Department of Informatics and Visabeira Group and Catholic University for their support, which each, in his way, gave me.

Concluding, in this so long journey I would also like to thank all my other friends and teachers, since the old days. From my primary school and all the ones in some way contributed to my education, including teachers, sisters and professors at Outeiro de São Miguel, Escola Secundária da Sé and Polytechnic of Guarda.

In summary and sincerely, I feel too much lucky to be part of your lives, and believe me: I just wanted to contribute to see your happiness.

P.S.: Sorry if I missed someone.

Abstract

Modern societies are increasingly dependent on essential products and services provided by Critical Infrastructures (CI), many of them consisting of Industrial Automation and Control Systems (IACS) such as power plants, energy distribution networks, transportation systems and manufacturing facilities. These IACS are becoming larger and more complex, due to the increasingly sophisticated physical processes they manage and the growing amount of heterogeneous data they need to consider, generated by a increasing number of interconnected control and monitoring devices. These IACS are also heavily dependent on common Information Technologies systems whose security, management, and compliance must also be considered. This evolving scenario requires new strategies to improve the associated Critical Infrastructure Protection (CIP) frameworks.

Compliance management and cyber-security forensic analysis are rapidly emerging as key components of CIP strategies. However, the intrinsic nature of the involved IACS requires specialized, domain-specific solutions which are still not common. In this scope, the work presented in this thesis researched and proposed innovative approaches, tools and frameworks to improve forensic techniques and continuous auditing for compliance management in CIP scenarios.

This dissertation proposes to improve CIP by integrating Forensics and Compliance Auditing (FCA) capabilities into a unified framework, to help identifying and understanding past security incidents and non-conformity situations. To leverage the analytic capabilities of the proposed FCA framework, it entails a scalable model helping to detect anomalies from a large number of distinct sources producing data at an intense pace. This dissertation also researches how to streamline the analytic processes over the living data dispersed across multiple and distinct corporate databases.

Keywords: Analytics, Big Data, Cloud Computing, Compliance Auditing, Critical Infrastructures, Forensics, Security

Resumo

As sociedades modernas dependem cada vez mais de produtos e serviços essenciais fornecidos por Infraestruturas Críticas, muitas delas compostas por Sistemas de Controlo e Automação Industrial como centrais elétricas, redes de distribuição de energia, sistemas de transporte e instalações industriais. Estes sistemas estão a tornar-se maiores e mais complexos, devido à crescente sofisticação e complexidade dos processos físicos que gerem e à crescente quantidade de dados heterogéneos que necessitam de analisar, produzidos por um número crescente de dispositivos de controlo e monitorização. Esses sistemas de controlo também dependem fortemente de sistemas comuns de Tecnologias de Informação, cuja segurança, gestão e conformidade também devem ser considerados. Esse cenário requer novas estratégias para melhorar as *frameworks* relacionadas de Proteção de Infraestruturas Críticas (PIC).

A gestão de conformidade e a análise forense de ciber-segurança estão rapidamente a emergir como componentes essenciais das estratégias de PIC. No entanto, a natureza intrínseca dos sistemas envolvidos requer soluções especializadas que ainda não são comuns. Neste âmbito, o trabalho apresentado nesta tese investigou e propôs abordagens, ferramentas e *frameworks* inovadoras para melhorar as técnicas forenses e de auditoria contínua para gestão de conformidade em cenários de PIC.

Esta dissertação propõe melhorar as práticas de PIC, integrando as capacidades de Auditoria Forense e de Conformidade numa estrutura unificada para melhor identificar e compreender os incidentes de segurança passados e as situações de não conformidade. Para alavancar os recursos analíticos da abordagem proposta, foi considerado um modelo escalável que suporta a deteção de anomalias com base num grande número de diferentes fontes, que geram dados heterogéneos a um ritmo intenso. Adicionalmente, é também investigada a forma de agilizar os processos analíticos sobre dados dispersos por diferentes bases de dados corporativas.

Palavras-chave: Análise, Big Data, Computação na Nuvem, Auditoria de Conformidade, Infraestruturas Críticas, Forense, Segurança

Foreword

THE work described in this dissertation was conducted at the Laboratory of Communication and Telematics (LCT) of the Centre for Informatics and Systems of the University of Coimbra (CISUC), within the context of the ATENA European research project (H2020 ATENA, Advanced Tools to assEss and mitigate the criticality of ICT compoNents and their dependencies over Critical InfrAstructures, N. 7005813, H2020-DS-2015-1).

Among other goals, the ATENA project researched new distributed awareness capabilities for Industrial Automation and Control Systems (IACS) systems considering threats and vulnerabilities of modern Critical Infrastructures (CIs).

The team from University of Coimbra (UC) led the research and development of solutions and components for distributed anomaly detection and risk assessment. The research work described in this thesis directly contributed to such efforts, namely the analysis of the state-of-the-art, the conceptualization and definition of a proof-of-concept Forensics and Compliance Auditing framework leveraged by an analytic model highlighting security incidents and non-conformity events and then extracting evidence from large volumes of heterogeneous data. A set of use cases also contributed to the demonstration and evaluation activities of the project.

Part of the contents of this dissertation were also reflected in ATENA technical deliverables. On the other hand, ATENA helped the research work presented in this thesis by providing reference scenarios, access to relevant requirements (produced by stakeholders such as CI operators) and demonstration testbeds (made available by Israel Electric Corporation).

The work conducted in the scope of this dissertation also led to the refereed publications listed next.

Journal Papers:

- **Henriques, J.**, Caldeira, F., Cruz, T., & Simões, P. (2018). On the use of ontology data for protecting critical infrastructures. Journal of Information Warfare, 17(4), 38-55.

  Main contributions: Extended version of a previously published conference paper (ECCWS 2018), with the review of the state of the art on the use of ontology data and conceptualization of a reference architecture. Validation of a federated framework to provide inference mechanisms relying on a semantic web approach over several different databases.

- **Henriques, J.**, Caldeira, F., Cruz, T., & Simões, P. (2020). Combining K-means and XGboost Models for Anomaly Detection using Log Datasets. Electronics, 9(7), 1164. DOI: 10.3390/electronics9071164

  Main contributions: Conceptualization and validation of a novel integrated and scalable framework for efficiently detecting anomalous events on large amounts of unlabeled data logs by the combined use of the K-Means algorithm and the XGBoost system.

- Rosa, L., Cruz, T., de Freitas, M. B., Quitério, P., **Henriques, J.**, Caldeira, F., & Simões, P. (2021). Intrusion and anomaly detection for the next-generation of industrial automation and control systems. Future Generation Computer Systems, 119, 50-67. DOI: 10.1016/j.future.2021.01.033

  Main contributions: Description of the Forensics and Compliance Auditing (FCA) framework, as part of the architecture of the ATENA Intrusion and Anomaly Detection System.

- **Henriques, J.**, Caldeira, F., Cruz, T., & Simões, P. (2022). An Automated Closed-Loop Framework to Enforce Security Policies from Anomaly Detection. Computers & Security, 102949. DOI: 10.1016/j.cose.2022.102949

  Main contributions: Review of the state of the art, proposal, conceptualization and validation of a closed-loop framework aiming to intelligently streamline the continuous maintenance of security policies.

- **Henriques, J.**, Caldeira, F., Cruz, T., & Simões, P. (2023). A Forensics and Compliance Auditing Framework for Critical Infrastructure Protection. International Journal of Critical Infrastructure Protection, 100613. DOI: 10.1016/j.ijcip.2023.100613

Main contributions: Proposal and formal proposal description of a framework unifying the Forensics and Compliance Auditing capabilities and evaluation of its scalability when ingesting, storing and computing data.

Conference Papers:

- Rosa, L., Proença, J., **Henriques, J.**, Graveto, V., Cruz, T., Simões, P., & Monteiro, E. (2017, June). An evolved security architecture for distributed Industrial Automation and Control Systems. In European Conference on Cyber Warfare and Security (pp. 380-390). Academic Conferences International Limited. ISBN: 978-1- 911218-43-2

  Main contributions: Preliminary description of the FCA framework, as part of the initial design of the ATENA Intrusion and Anomaly Detection System.

- **Henriques, J.**, Caldeira, F., Cruz, T., & Simões, P. (2018). On the use of ontology data for protecting critical infrastructures. In ECCWS 2018 17th European Conference on Cyber Warfare and Security V2 (p. 208). Academic Conferences and publishing limited.

  Main contributions: Review of the state of the art on the use of ontology data and conceptualization of a reference architecture for a federated framework to provide inference mechanisms relying in semantic web approach over several different databases.

Book Chapter:

- Rosa, L., de Freitas, B., **Henriques, J.**, Quitério, P., Caldeira, F., Cruz, T., & Simões, P. (2020). Evolving the security paradigm for industrial IoT environments. In Cyber Security of Industrial Control Systems in the Future Internet Environment (pp. 69-90). IGI Global.

  Main contributions: Review of the state of the art and conceptualization of a reference architecture, and development of the tools to be part of the FCA framework.

Submitted Papers:

- **Henriques, J.**, Caldeira, F., Cruz, T., & Simões, P. (2023). A Survey on Forensics and Compliance Auditing for Critical Infrastructure Protection. *Submitted to the IEEE Access Journal, under review.*

  Main contributions: Comprehensive survey on the topic of FCA for CIP, including the proposal of a taxonomy and a reference framework architecture.

Contents

List of Figures

List of Tables

List of Acronyms

AAA Authentication Authorization and Accounting. 61, 77

AAFS American Academy of Forensic Sciences. 29

AI Artificial Intelligence. 43, 51–53, 57

AM Analytics Module. 86

API Application Programming Interface. 67, 130

C2 Command and Control. 101

CI Critical Infrastructure. 2–7, 14, 15, 17–19, 21, 22, 25, 33, 38, 41, 43, 47, 50, 54, 55, 58, 60, 61, 65, 66, 68, 69, 71, 75, 83, 85, 87–91, 126, 127, 131, 132, 143, 161, 164–166

CI/CD Continuous Integration/Continuous Delivery. 149

CIP Critical Infrastructure Protection. xvii, 2–7, 9, 14, 16, 17, 21, 25, 31, 37, 43–46, 51, 52, 57, 58, 64, 68–71, 75, 126–128, 131, 132, 143, 164, 166, 167

COBIT Control Objectives for Information and Related Technology. 19, 21

CPS Cyber Physical Systems. 15, 21, 26, 45, 52

CRUD Create, Read, Update, and Delete. 131

CS Control Systems. 21, 49

CSP Cloud Service Provider. 33–35

CVE Common Vulnerabilities and Exposures. 148

CVSS Common Vulnerability Scoring System. 148

CWE Common Weakness Enumeration. 148

DAM Data Acquisition Module. 77–80, 102

DBMS Database Management System. 52

DDoS Distributed denial-of-service. 53, 57

DHS Department of Homeland Security. 39

DL Data Lake. 62–66, 78, 82–84, 86–89, 99, 102

DPM Domain Processor Module. 80, 81, 84, 102

DRDoS Distributed Reflective Denial of Service. 40

DT Decision Tree. 147, 148, 151–153, 155–157, 160, 167

EDR Endpoint Detection and Response. 24, 67

EIC Explore, Investigate, and Correlate. 40

ETL Extract, Transform and Load. 62

EU European Union. 47, 52

FCA Forensics and Compliance Auditing. xvii, 2, 4, 6–9, 14, 15, 17, 22, 44, 51, 52, 55, 57–60, 62, 66–71, 75, 77, 83–85, 90–94, 98, 99, 103, 116, 132–134, 161, 164–168

FISMA Federal Information Security Modernization Act. 47

GDPR General Data Protection Regulation. 36, 47, 70

GRR Google Rapid Response. 40

HIDS Host Intrusion Detection System. 22

HIPAA Health Insurance Portability and Accountability Act. 47

HMI Human Machine Interface. 19

HTTP Secure HyperText Transport Protocol. 115, 116, 122, 123

IaaS Infrastructure as a Service. 48

IaC Infrastructure as Code. 148, 150

IACS Industrial Automation and Control Systems. 2–7, 14, 15, 17, 18, 21, 22, 25, 38, 42, 49, 55, 58, 60, 71, 75, 77, 91, 126, 161, 164, 166, 167

IADS Intrusion and Anomaly Detection System. 91

ICS Industrial Control Systems. 2, 17, 26, 31

ICT Information and Communication Technologies. 22, 60, 61, 69, 77, 91

IDPS Intrusion Detection and Prevention System. 22

IDS Intrusion Detection System. 18, 22, 32, 34, 42, 58, 62, 83, 84, 101

IEC International Electrotechnical Commission. 49

IED Intelligent Electronic Devices. 19

IIoT Industrial IoT. 26, 42, 51, 52, 69, 70, 167

IM Ingesting Module. 62

IoT Internet of Things. 2, 26, 42, 43, 52, 57

IP Internet Protocol. 18, 53

IPA Information-Technology Promotion Agency. 49

IPS Intrusion Prevention System. 7, 18, 22

ISMS Information Security Management Systems. 49

ISP Internet Service Provider. 40

IT Information Technology. 27, 42, 47, 54, 146, 149, 151

JSON JavaScript Object Notation. 157

KVM Kernel-based Virtual Machine. 35

LDAP Lightweight Directory Access Protocol. 62, 133

LSTM Long Short-Term Memory. 56

MAPE Monitor-Analysis-Planning-Executing. 53, 151

ML Machine Learning. 31, 34, 51, 53, 54, 57, 63, 64, 85, 99, 111, 112, 147, 150–152, 156, 160, 168

NCC FSWG NIST Cloud Computing Forensic Science Working Group. 35

NERC-CIP North American Electric Reliability Corporation Critical Infrastructure Protection. 47

NFI Netherlands Forensic Institute. 39

NIDS Network Intrusion Detection System. 22

NIST National Institute of Standards and Technology. 26, 30, 33, 36, 42, 49

OS Operating Systems. 30, 101, 102

OT Operational Technology. 77

OWASP Open Web Application Security Project. 148

OWL Web Ontology Language. 131

PaaS Platform as a Service. 68

PaC Policy as Code. 9, 147–152, 155, 157, 158, 160, 161, 167

PCAP Packet Capture Process. 79

PCIDSS Payment Card Industry Data Security Standard. 47, 148

PE Policy Engine. 147–150, 152, 155, 157, 167

PLC Programmable Logic Controller. 18, 19

PoC Proof of Concept. xv, 7, 91, 155, 157–161, 165, 166

QoE Quality of Experience. 70

QoS Quality of Service. 70

RDB Relational Database. 8, 126–132, 135, 141, 143, 165, 168

RDBMS Relational Database Management Systems. 16

RDF Resource Description Framework. 8, 40, 126–132, 135, 140, 141, 143

RTU Remote Terminal Unit. 18, 19

SaaS Software as a Service. 24, 67

SAML Security Assertion Markup Language. 63

SCADA Supervisory Acquisition and Data Control. 3, 17–19, 26, 27, 31, 32, 41, 42, 49, 52, 57, 58, 77

SDN Software Defined Networking. 30

SIEM Security Information and Event Management. 19, 22–25, 58, 67, 70, 91, 126, 127, 132

SLA Service Level Agreement. 6, 33, 164

SOAR Security Orchestration, Automation and Response. 24, 67

SQL Structured Query Language. 130, 131

SSU Shadow Security Unit. 22

TSMS Time Series Management Systems. 52

UAC User Account Control. 101, 102

US United States. 19, 21, 39

VCS Version Control System. 149, 150

ViCAP Violent Criminal Apprehension Program. 39

ViCLAS Violent Crime Linkage System. 39

VM Virtual Machine. 34–36

W3C World Wide Web Consortium. 129

XDR Extended Detection and Response. 24, 67

ZSM Zero-touch Network & Service Management. 146, 147, 150, 160

Chapter 1

Introduction

Contents

This dissertation addresses the topic of Critical Infrastructure Protection (CIP) by exploring the adoption of an unified security framework offering Forensics and Compliance Auditing (FCA) capabilities, to identify and understand past and ongoing security incidents and to assess the level of compliance with relevant standards, regulations and recommended practices.

This chapter starts by introducing the background on the topic and explaining the motivation behind the research work described in this dissertation. The research goal and objectives are presented next, followed by the identification of the main contributions produced by this research work. Finally, the structure of this document is presented.

1.1 Background and Motivation

Critical Infrastructures (CIs) provide essential products and services supporting our society and economy, including power plants, energy distribution networks, transportation systems, and manufacturing facilities. Necessarily, and due to their importance, such infrastructures constitute attractive targets for malicious actors, with cyber-attacks rapidly becoming one of the largest risks for CIs and customers alike, with potential impacts ranging from service disruption and economic losses to physical damage and human casualties [2, 3].

Protecting CIs is not an easy task, though. Industrial Control Systems (ICS), which are at the core of many CIs, have evolved towards becoming increasingly connected and distributed (encompassing a diversified array of interconnected devices, sensors, and actuators, often widely spread in the field), and producing an increasing amount of information exchanged among system components. Examples such as water-to-wire generation, microgeneration, smart grids, smart metering, oil and gas distribution, or smart water management, among others, are pushing the boundaries of the classic ICS model, leveraging Internet of Things (IoT) technologies to create a new generation of Industrial Automation and Control Systems (IACS) [4]. While the benefits of this evolution are manyfold, there is also a side effect in the form of an expanded attack surface.

As increasing connectivity exposes CI systems not only to traditional threats, but also to new ones, attacks against them are likely to grow in the near future [5]. Thus, it comes an no surprise that cyber-attacks targeting CIs such as government, power, health, communications, financial, and emergency response services constitute a serious concern for governments and organizations alike. Particularly, such attacks can be potentially disruptive for IACS, as it is the case for Supervisory Acquisition and Data Control (SCADA) systems, which have become increasingly attractive for malicious actors over the past years [6, 7], as confirmed by several publicly disclosed incidents which have impacted organizations with millions of customers, at a significant cost – for instance, and according to IBM Managed Security Services data [8], attacks targeting IACS have increased over 110% in 2016, linked with the growing connectivity of industrial systems.

The increased complexity of CIs and IACS makes it difficult to understand the nature of incidents and assess their progression and threat profile. Moreover, reacting and defending against those threats is something that is becoming increasingly difficult to manage, requiring orchestrated and collaborative distributed detection. Thus, coping with the challenges posed by the current CIP landscape implies not only to deal with the growing volume and sophistication of cyber-threats but also with the consequent increased cybersecurity analysis workloads. The latter aspect constitutes a crucial bottleneck, often involving a multi-step process where skilled security analysts go to multiple systems to retrieve data, later to be manually analyzed and correlated. Thus, leveraging security tools and systems capable of quickly and automatically correlating data from different sources with other security data can help drawing an effective big picture discerning security events leading to true, high-priority, actionable alerts to cover an entire attack surface. They help to create meaningful and valuable alerts, specifically warning over-worked, understaffed security teams of real, consequential attack activity early enough to minimize damage.

Nevertheless, solutions, tools and procedures are meaningless without strategy and guidance, mandating the definition of organization-wide Information Security Management Systems (ISMS) policies which help define and steer a coherent security posture. This is further reinforced by the need to comply with sectorial regulatory and standardization requirements, such as the ISO27K series [9]. However, many organizations neglect the establishment of proper security policies, only aligning with the least necessary requirements to comply with mandatory regulations – this means that aspects such as supply-chain security, among others, are often disregarded.

But even organizations who strive to comply and go the extra mile often fail at monitoring and assessing regulatory and/or standards compliance, due to the implicit task complexity and the skillset required to implement and maintain adequate procedures and processes, whose operational costs are far from negligible.

Therefore, in addition to other security tools, such as specialized probes, intrusion detection platforms, and firewalls, FCA tools are becoming increasingly important for security professionals. Such tools provide the means to extract relevant insights and evidence from large volumes of heterogeneous data produced from the sources within the CI, which can be leveraged both for forensics analysis purposes and to audit the enforcement of security and quality policies.

When it comes to FCA toolsets, the current perspective is somehow limited. Most tools are either too specialized, single-threaded, or limited to offline tasks, being unsuitable for massively distributed and/or complex scenarios, eventually involving microservices or cloud-native environments [10]. Moreover, a large number of the existing tools do not follow the standards for data formats or digital forensic processes, lacking evidence provenance or anti-tampering protection [11].

With the bulk of the CIP cybersecurity R&D efforts being oriented towards threat prevention, detection and mitigation, other aspects such as forensics support or compliance auditing are often disregarded. As a consequence, the current breed of FCA tools is either unsuitable for modern IACS and/or requires a significant integration effort, due to their IT-oriented[1] scope which also limits coverage of many domain-specific aspects. This is a significant gap, mainly due to two reasons:

- The best practices for incident handling always encompass a "lessons learned" stage, that is only possible by employing post-mortem trace analysis, allowing experts to reconstruct the trail conducting to the root cause and generate valid and useful evidence. This falls within the realm of forensics processes, comprising evidence collection, in the form of records and digital trails that can be legally used for criminal prosecution [12] or incident analysis.

- There is an extensive body of knowledge supporting the definition of proper procedures and guidelines for ongoing secure and safe operation, management, and maintenance of CIs. However, the effectiveness of such policies is only as good as the existing capabilities to continuously monitor their correct enforcement and application, something that becomes increasingly difficult to be manually undertaken as the infrastructure grows in size and complexity. In this context, these capabilities constitute a form of audit compliance processes that should be able to check whether adequate procedures and security policies are in place and in conformity with the standards, regulations or in-

[1] IT = Information Technology

ternal rules, in a slower process path that will check for eventual breaches or misconfigurations.

Both contexts share many similarities, in the sense that compliance auditing procedures are closely related to and often overlap with forensic processes. Moreover, forensics procedures often feed organizational policy definitions and, consequently, compliance audit procedures – an example can be illustrated by cyber-incidents caused by lack of user awareness and/or adequate training, where human errors are identified as the main cause [13]. In such situations, forensics procedures provide insights that may be leveraged not only to improve user training processes, but also to develop policies to help avoid future incident replicas.

Forensic and audit investigations can involve analyzing and correlating large amounts of past events from different data sources using different tools and techniques to accurately identify and triage incidents and determine their cause. Evidence can be retrieved from any type of file, such as log files or virtual machine images, and any other digitally acquired data. First, the data can be used to learn what the normal behavior is, and then it can also be used to uncover anomalies and identify traces of threat events, compromised security, and non-compliant policies, standards, or business rules to be extracted as evidence.

The work described in this thesis has researched and developed new strategies to improve CIP by adopting proactive approaches regarding forensics and audit compliance, capable of coping with the challenges of evolved IACS infrastructures. Gathering both forensics and compliance auditing approaches, and aggregating such disparate number of techniques and tools in a unified platform designed for CI contexts, helps reducing the complexity, effort, and costs associated with investigating and connecting individual alerts to uncover potential threats. Such a proactive approach may avoid the disruption of operations and prevent evidence from being lost or corrupted. Moreover, it will also help to deal with evolving cyber threats affecting CIs, providing valuable insights to fine tune traditional security mechanisms (intrusion detection, prevention, and mitigation) and to prepare the platforms for post-incident forensics analysis.

1.2 Research Goal and Objectives

Considering the overview presented so far, **the research efforts undertaken in the scope of this thesis aim at improving CIP by researching, develop-**

ing, and integrating FCA capabilities into an advanced unified security solution framework.

To achieve that ultimate goal, the research work was driven by the following three fundamental objectives:

- **Objective 1. To improve the forensic approach for CIs supported by IACS, making it possible to process massive amounts of heterogeneous data using scalable approaches.**

  This objective aims to improve the state of the art of CIP supported by a security architecture designed to leverage the capability to collect data from diversified and heterogeneous sources (in contrast to the current practice in the area), which may later serve to support forensics activities, such as post-mortem or root cause analysis. Moreover, that data should later be admissible as evidence in the case of a security incident. By considering new techniques not tried in the field, this framework should be able to scale as needed, enabling, for instance, the ingestion of massive amounts of data at high speed, also allowing it to cope with peaks caused by heavy system activity.

- **Objective 2. To improve policy, regulatory and standards compliance auditing mechanisms for CIP supported by IACS.**

  This objective is to enhance CI security, by leveraging a compliance auditing approach focused on assessing the application of the regulatory, legal and standardization guidelines from regulators and standardization bodies (including domain-specific entities), while also providing continuous monitoring of third-parties (e.g. supplier relationships, staffing activities, internal staffing activity), quality of service in the scope of Service Level Agreements (SLAs), and violation of business rules or organizational policies. By reusing the same data sources used to achieve the previous objective, this approach will help to identify latent security threats and risks or even reveal potential cyber-attacks, providing a chance to prevent them or at least reduce their impact.

- **Objective 3. To identify the most suitable analytical models to improve the detection of anomalies, replay of incidents and enhance the ability to collect evidence for CI forensic and audit compliance processes.**

  A cyber anomaly detection system represents the first ring of the early warning chain in case of cyber-attacks, denial of service or malware spreading. This role can be taken by both FCA and Intrusion Prevention System (IPS) mechanisms in a complementary way: while the IPS operates online with the aim of blocking threats, often using a limited time window, the FCA is a slower process, supposed to consider a broader event horizon while encompassing a more diverse set of inputs. Thus, the identification of suitable anomaly detection models, algorithms and mechanisms, also taking into account the use of distributed approaches to improve efficiency, will contribute to the robustness and resilience of the physical CI and their IACS. Attaining this objective will support the previous ones by making possible to reconstruct the chain of events to be replayed and extracted for forensics and compliance auditing purposes.

1.3 Contributions

The work developed in the scope of the thesis led to five key contributions:

- **Contribution 1. A comprehensive survey on the topic of FCA for CIP, which also produced a related taxonomy and a reference architecture for consolidated operations.** This survey reviewed the state of the art and the latest developments, methodologies, challenges, and solutions related to the topic. It focused on relevant contributions, capable of tackling the requirements imposed by massively distributed and complex IACS, in terms of handling large volumes of heterogeneous data (that can be noisy, ambiguous, and redundant) for analytic purposes, with adequate performance and reliability. The survey also produced a structured taxonomy for the field of FCA, based on the relevant topics found in the literature. Also, the collected knowledge resulted in the establishment of a reference architecture for FCA systems, proposed as a generic template for converged platforms (which was later materialized in a Proof of Concept (PoC) demonstrated in the scope of the ATENA project). These results, which are useful to introduce and guide future research works in the field, are reflected mainly in the contents of Chapter 2.

- **Contribution 2. Proposal, design and implementation of a domain-specific FCA framework for CIP scenarios**. Based on the lessons learned from the survey and its generic reference architecture, a more detailed platform for supporting FCA activities has been designed and specified. A proof-of-concept implementation of this design has also been developed and integrated into the ATENA framework, for demonstration and evaluation purposes. This contribution reflects mostly in the contents of Chapter 3

As described next, once the core FCA platform was designed and implemented, the research work focused on contributing to more specific areas: innovative ways of improving scalability, improvement of inference analysis by means of ontology data, and automation of closed-loop enforcement of security policies.

- **Contribution 3. Mechanisms for detecting anomalies using large log datasets.** As one of the application scenarios of the FCA platform, specific mechanisms were proposed for identifying anomaly events in massive log files, relying on a two-stage process. In the first stage, an unsupervised model helps separating anomalies from normal events. In the second stage, a gradient tree boosting classification supervised model, using XGBoost distributed gradient-boosted decision trees to produces the interpretable and meaningful rule set for generalizing its application to a massive number of unseen events. Given its ability to leverage parallel computing resources, this approach can scale to identify anomaly events in out-of-core datasets, making it able to deal with situations where case log sources are so massive that it becomes impossible to use more traditional approaches. Chapter 4 describes these mechanisms.

- **Contribution 4. A federated approach for supporting inference analysis on ontology data.** This approach offers semantic web inference capabilities from the existing heterogeneous data sources, maintained in multiple and natively different Relational Database (RDB) systems. It leverages the process of interactively exploring, searching, extracting, and pinpointing insights by combining data sources from disparate organizational RDBs. Such an approach avoids the duplication of information in RDB and Resource Description Framework (RDF) stores and overcomes the issues arising from the use of static data integration (such as the lack of support on transforming data and the effort required for maintaining up-to-date synchronization processes). An abstraction layer deals with the inherent complexities of resorting to different platforms, systems, technologies, and information schemas to retrieve and combine heterogeneous data. This abstraction layer can contribute to improve security by hiding the infrastructure's internal details. This approach is detailed in Chapter 5.

- **Contribution 5. An automated closed-loop framework to enforce security policies from anomaly detection**. A framework was proposed for creating workflows that automate the end-to-end process that goes from the classification of anomalies to translational policy rule generation and subsequent enforcement. This framework consists of a closed-loop that integrates the security-related classification processes that continuously improve the security policies from the new incoming data. These policies can be integrated into the artifacts devoted to their enforcement, in line with Policy as Code (PaC) principles. Such an approach contributes to reducing the efforts, time, and costs of required human labour for maintaining a system secure. This contribution is reflected mostly in Chapter 6.

1.4 Structure of the Document

The rest of this document is organised as follows.

Chapter 2 provides the background and reviews the literature related to CIP and FCA topics, identifying open challenges and proposing a taxonomy and a reference architecture for FCA systems. Based on the outcomes of the survey, Chapter 3 presents the proposed FCA framework specifically designed for CIP scenarios, describing each of its components, the proof-of-concept implementation, and evaluation results.

Afterwards, the thesis splits into three specialized contributions proposed on top of the FCA framework. Chapter 4 proposes mechanisms combining K-Means and XGBoost models for distributed anomaly detection using large log datasets. Chapter 5 presents a federated approach providing inference capabilities supported by ontologies over data from heterogeneous sources, and Chapter 6 describes an automation model to translate into security policies the from rules from decision tree models. Finally, Chapter 7 concludes the document.

Chapter 2

Forensics and Compliance Auditing for Critical Infrastructure Protection

Contents

This chapter presents a survey of the latest developments, methodologies, challenges and solutions addressing forensics and compliance auditing in the scope of Critical Infrastructure Protection (CIP), exploring the most relevant approaches, methodologies, and technologies. To the best of our knowledge, this is the first survey specifically focused on Forensics and Compliance Auditing (FCA) applied to the CIP domain. The survey focuses on relevant contributions, capable of tackling the requirements imposed by massively distributed and complex Industrial Automation and Control Systems (IACS), in terms of handling large volumes of heterogeneous data (that can be noisy, ambiguous, and redundant) for analytic purposes, with adequate performance and reliability. This effort also spawned a taxonomy in the field of FCA whose key categories denote the topics of relevance in the literature. Finally, the collected knowledge resulted in the establishment of a reference FCA architecture, proposed as a generic template for a converged platform. These results also provided valuable insights and guidance regarding the research and development efforts undertaken in the scope of this thesis.

The rest of the chapter is organised as follows. Motivation is discussed in Section 2.1. The background of CIP and IACS security are introduced in Section 2.2. Section 2.3 discusses previous works related with forensics, and Section 2.4 presents a similar analysis of works addressing compliance auditing. Section 2.5 discusses benefits and challenges of modern analytics in the era of Big Data, AI and ML, with a focus on FCA. Based on the lessons learned from the survey, Section 2.6 proposes a new taxonomy for FCA systems for CIP and Section 2.7 introduces a reference architecture for FCA systems. Section 2.8 discusses achieved results and open issues, and Section 2.9 concludes the chapter.

It should be noted that this chapter includes content that has already been submitted for publication [14].

2.1 Motivation

The broadening dependency and reliance that modern societies have on essential services provided by Critical Infrastructure (CI) is increasing the relevance of their trustworthiness. However, CIs are attractive targets for cyberattacks, due to the potential for considerable impact, not just at the economic level but also in terms of physical damage and even loss of human life.

From this perspective, forensics and compliance audit processes play an important role in ensuring CI trustworthiness, by complementing the role of traditional security mechanisms. While compliance auditing contributes to checking if security measures are in place and compliant with standards and internal policies, forensics processes assist the investigation of past security incidents. Since these two areas significantly overlap in terms of the leveraged data sources, tools and techniques, they can naturally fit within a converged FCA framework.

2.1.1 The challenge of protecting Critical Infrastructures

As already mentioned (cf. Section 1.1), CIs provide a series of essential services which are key to ensure the security, societal and economical activities of a country, thus constituting an attractive target for cyber-attackers [15]. Smart grids, water, oil, and gas distribution networks are becoming more complex due to the growing number of interconnected distributed devices, sensors, and actuators, often widely dispersed in the field, as well as the increasing amount of information exchanged among system components. They are pushing the boundaries of the classic Industrial Cyber Physical Systems (CPS) model, with an impact on IACS cybersecurity requirements, due to a substantial increase in the scale and complexity of the protected infrastructure.

This increase in terms of interconnections has a direct impact in terms of the exposed attack surface, exposing the IACS to both traditional and new threats. Network-based attacks targeting CIs are also becoming a greater concern, as state-sponsored groups have become more active. Their activities comprise unauthorized access to government and corporate networks with the main purpose of gathering information, although they can be potentially disruptive for CIs [16]. This trend is already a major concern, and is expected to further intensify in the future [7] [17].

Other IACS security threats come from their increasingly distributed nature, regarding both the physical processes under control, which have also become more widely dispersed and interconnected, and the associated control applications, which have also become increasingly distributed, for sake of scalability, elasticity, adaptability, resiliency, and fault-tolerance. Overall, this scenario makes it difficult to understand the nature of incidents and to assess their progression and threat profile. Moreover, defending against those threats is becoming increasingly difficult, requiring orchestrated and collaborative distributed detection, analysis, and reaction capabilities.

Continuously capturing live data from a running IACS system, that has an intrinsic volatile nature, presents important challenges to forensics investigators. For instance, volatile data in physical memory contains information about the current state of the system, such as process information, open network connections and encryption keys.

Another challenge comes from the amount of data to be collected, analyzed, and stored for detecting and profiling cyberattacks. According to IBM [18], the world produces over 2.5 quintillion bytes of data every day, and 80% of it is unstructured (and not analyzed). To improve decision making, enterprises are facing new challenges to collect a large amount of available data, retrieved from heterogeneous sources (including structured and unstructured data), and enriching it with the inclusion of additional contextualized data. In the specific scope of CIP, to face the tremendous growth of raw data being produced by sensors and process controllers, a Big Data approach is required to handle massive amounts of data in intensive online and offline processing flows. The growth of volume and heterogeneity of data sources, systems, workloads, and environment variability contributes to the complexity of data management. Traditional approaches, such as Relational Database Management Systems (RDBMS), might not be able to handle the deluge of industrial data they are experiencing, especially while addressing the need for improved performance, reliability, and user experience [19]. Gaining critical business insights by querying and analyzing such massive amounts of data is becoming a vital requirement [20].

2.1.2 The need for better Forensics and Compliance Auditing

Security incidents trigger a series of reactive activities, such as blocking access to and quarantining compromised systems, assessing the impact of the breach, mitigating the damage, and conducting forensics investigations to identify exploited vulnerabilities, identify the attackers, and enhance future defensive actions.

In 2020, it took an average of 207 days to identify a breach, and 280 days to contain it [21]. Such a scenario results from the current solutions demanding a multi-step process where security analysts goes to multiple systems to retrieve uncorrelated data and then correlate it manually. Moreover, the complexity, skillset, and costs required to deploy and operate those solutions present a significant number of obstacles to their adoption. Therefore, in addition to other security tools, such as specialized probes, intrusion detection platforms and firewalls, forensics tools are increasingly important for security professionals. Such tools provide the means to extract relevant insights and evidence from large volumes of heterogeneous data produced from the sources within the CI, which can be leveraged both for forensics and security analysis purposes.

Auditing compliance with applicable laws, regulations, policies, and standards processes also contributes to increase CI trustworthiness. However, such auditing processes are complex, since they need to use of a large number of tools, protocols and standards to correlate and enforce the audit compliance policies that may help to prevent future incidents.

Aggregating such tools in a unified platform can reduce the complexity, effort, and costs associated with investigating and connecting individual alerts to uncover potential threats. Such a proactive approach may avoid the disruption of operations and prevent evidence from being lost or corrupted. Moreover, it will also help deal with the evolving cyber threats affecting CIs, preparing the platforms for post-incident forensics analysis.

Due to the considerable overlap of functionalities associated with security forensics and compliance audit processes, it makes sense to consider them as unified platforms, which we generically designate as FCA frameworks – even though many tools are applied only to one of these areas, they still share most requirements and technologies.

2.2 CIP and IACS Security Landscape

In this section we provide a landscape overview of CIP, with a more detailed perspective on IACS security – since most CIs are based on some sort of industrial control frameworks. The goal is to provide the reader with a more detailed perspective on how such systems are currently managed, from a security perspective, so that the associated needs, in terms of forensics and compliance auditing, become more clear.

2.2.1 The role of IACS and SCADA Systems in CIP

As already mentioned, a large number of CIs are based on large-scale IACS or Industrial Control Systems (ICS) which, traditionally, use Supervisory Acquisition and Data Control (SCADA) systems to manage physical processes such as energy production and distribution, water and sewage treatment, traffic management and railways.

These SCADA systems can be roughly defined as a set of systems of command

and control networks that control the operational sequence of the underlying physical processes. A typical SCADA system controlling CIs generally includes a control center and several field sites [22]. These sites are often distributed over a wide geographical area. Field sites are equipped with devices such as Programmable Logic Controllers (PLCs) or Remote Terminal Units (RTUs) [23], that control the on-site machines and periodically send information about the state of the field equipment to the control center. SCADA communications use a wide range of protocols, such as DNP3, Modbus, PCOM, ProfiNet, DeviceNet, ControlNet or Common Industrial Protocol [24].

In the early days, SCADA systems did not incorporate cyber-security mechanisms, since they were significantly resource-constrained and designed to run in isolated networks. They consisted of simple I/O devices transmitting signals between master and remote terminal units. Currently, SCADA systems can communicate over Internet Protocol (IP) networks, enabling its connection to the corporate network or even directly to the Internet, to integrate SCADA data with external systems such as Enterprise Resource Planning and Business Process Management tools. This interconnection of SCADA systems with wider networks brings new threats for which they were not originally designed, making them much more vulnerable. Moreover, CIs such as smart grids and water distribution networks have become increasingly complex due to the number of interconnected distributed devices, sensors and actuators, often widely dispersed in the field, and the larger amount of information exchanged both within the control system and between the control system and external systems.

As pointed out by Ahmed et al. [22], Cornelius and Fabro [25], and Eden et al. [26], the different nature of SCADA systems also raises important challenges in the application of forensics, when compared to traditional approaches. Those classic forensics methodologies potentially interfere with the IACS operation, since they may introduce latency and cause critical processes to fail. Another challenge arises from the use of resource-constrained devices such as RTU and PLC, which often lack the storage and processing capabilities required by forensics tools. Also, SCADA logs might be not suitable for forensic investigation, as they are geared towards process management, not cybersecurity. Nonetheless, there is still a general lack of SCADA-specific forensics tools.

To prevent known and unknown attacks, including security vulnerabilities and threats, organizations are adopting a common set of defense solutions such as firewalls, antivirus, Intrusion Detection Systems (IDSs), Intrusion Prevention Systems

(IPSs), and Security Information and Event Management (SIEM) platforms [27, 28]. Eden et al. [26] provided an overall forensic taxonomy of the SCADA system incident response model and discussed the development of forensic readiness within SCADA system investigations, including the challenges faced by the SCADA forensic investigator and suggested ways in which the process may be improved. Van der Knijff [29] identified possible sources of evidence in the investigation process in CI. Some of them include engineering workstations, databases, historian, Human Machine Interface (HMI), application server, Field devices like PLC, RTU, Intelligent Electronic Devices (IED), firewall logs, web proxy cache, and ARP tables.

2.2.2 Security Frameworks

Several security frameworks incorporate a series of documented processes used to define the policies and procedures around the implementation and management of information security controls in an enterprise environment. These frameworks are a blueprint for building an information security program to manage risk and reduce vulnerabilities by applying a function for identifying, protecting, detecting, and responding to activities. These frameworks can help information security professionals to define and prioritize the tasks required to manage their organizations' security. Examples of IT security frameworks include Control Objectives for Information and Related Technology (COBIT) [30], ISO 27000 series [31], NIST Special Publications 800-53 [32], 800-171 [33], NIST Cybersecurity Framework for Improving Critical Infrastructure Cybersecurity [34] and HITRUST CSF. The HITRUST CSF represents a certifiable framework that provides a comprehensive, flexible, and efficient approach to regulatory/standards compliance and risk management [35].

NIST SP 800-53 is the standard required by United States (US) federal agencies but may also be used by any company to build a technology-specific information security plan [36]. NIST 800-171 [33] provides federal agencies with recommended security requirements for protecting the confidentiality of controlled unclassified information.

The ISO/IEC 27000 series provide key information security frameworks applicable to any industry [37, 31]. For instance, ISO/IEC 27004:2016 provides guidelines supporting organizations in assessing security performance and effectiveness indicators [38] to fulfill the requirements of ISO/IEC 27001:2013, with ISO/IEC 27005:2018 providing guidelines for information security risk management. ISO/IEC 27037:2012 provides guidelines on the handling of digital evidence, including identification, collection, acquisition, and preservation of potential digital evidence [39]. ISO/IEC 27038:2014 covers the techniques for performing digital redaction on digital documents [40]. ISO/IEC 27042:2015 provides guidance on the analysis and interpretation of digital evidence keeping continuity, validity, reproducibility, and repeatability [41]. ISO/IEC 27050 represents a group of standards (27050-1 to 27050-3) addressing the discovery of Electronically Stored Information, a term coined to refer to *forensic evidence in the form of digital data* [42].

Also within the ISO/IEC 27000 series, ISO/IEC 27041:2015 provides guidelines on how to make sure that the methodologies and processes used to investigate information security events are suitable [43]. ISO/IEC 27043:2015 includes guidance for common incident investigation techniques across numerous incident investigation scenarios utilizing digital evidence, based on idealized models [44]. ISO/IEC 27006:2015 specifies requirements and guidance providing audit and certification of information security management systems [45]. ISO/IEC TS 27008:2019 provides guidance for evaluating the implementation and operation of information security controls, including their technical assessment, following an organization's established information security requirements, including technical compliance [46]. ISO/IEC 27040:2015 provides technical recommendations on how organizations can establish an appropriate level of risk mitigation by using a tried-and-true approach to data storage security strategy, design, documentation, and implementation.

Moreover, there are other relevant standards within the ISO/IEC frameworks, such as ISO 21043-1:2018 that introduces important terms and definitions in forensic sciences [47], also providing the requirements for the forensic process with a focus on the recognition, recording, collection, transport, and storage of potential forensic items [48]. Also, ISO/IEC 30121:2015 is a framework for helping organizations to be prepared for digital investigation processes [49].

Regarding forensics education and training, the ASTM standards are worth mentioning [50]. ASTM E2678 helps promoting computer forensics by developing model courses that are compatible with other forensic science programs. ASTM E2917 provides core standards for forensic science practitioners' training, continuing education, and professional development, including training criteria for competency, training documentation and implementation, and continual professional development. ASTM E2916 takes computer forensics, image analysis, video analysis, forensic audio, and facial identification are just some of the phrases and definitions that are utilized in the study of digital and multimedia evidence.

Deciding upon the applicable regulatory or standardisation frameworks an organization must comply with must consider several factors such as the type of industry or country-specific compliance requirements. For example, US traded companies may start by complying with Sarbanes-Oxley [51] and COBIT. In case the company needs information security capabilities the option is ISO 27000 certification. NIST SP 800-53 is the standard required by US federal agencies but could also be used by any company to build a technology-specific information security plan. The HITRUST CSF integrates well with healthcare software or hardware vendors looking to provide validation of the security of their products. NIST 800-94 [52], was introduced in 2007 highlighting the challenges in the detection accuracy, extensive tuning, blindspots, and performance limits.

2.2.3 IACS Security

Although the protection of CI is a topic not necessarily dependent on technology, this survey is driven by a technological approach focused on IACS protection. IACS incorporate Control Systems (CS) designed to manage and control physical processes, constituting one of the main targets for CIP activities. These CS can be defined as manual or automatic mechanisms used to manage dynamic processes by adjusting or maintaining physical quantities such as mass, temperature, or speed. CS are classified in two distinct categories: open- and closed-loop. Open-loop CS generate their output based on input only, while in a closed-loop the output is used as a feedback mechanism together with inputs to generate new output [29]. CS are generally used for monitoring and controlling industrial and infrastructure processes and dispersed assets supported by centralized data acquisition and supervisory control, often constituing a CPS. In the scope of the so-called essential services, these CPS are vital, often being highly interconnected and mutually dependent.

2010's Stuxnet [53] and 2015's BlackEnergy [54, 55] demonstrated that the so-called *security by obscurity* approach is no longer adequate for CIs. Stuxnet was the first known malware specifically designed to target automation systems, infecting between 50,000 to 100,000 computers worldwide. BlackEnergy was directly responsible for power outages for 250,000 customers in western Ukraine. Since then, many other attacks targeting IACS were recorded (e.g. Gauss, Havex, and Shamoon [56]).

This situation has prompted the development of suitable mitigation mechanisms to deal with cyberthreats against IACS which may compromise integrity, information/control confidentiality or availability [57], such as unauthorised accesses, break-ins, penetration attempts, and other forms of abuse, to detect and secure the automation infrastructure perimeter from attacks [58].

Among these mechanisms, IDSs provide the means to monitor the infrastructure, detecting security anomalies or suspicious behaviour by resorting to signature (rule-based) [59] or anomaly detection strategies [60]. Due to their nature, IDSs often constitute one of the most relevant data sources for FCA purposes, detecting threats and recording incident-related valuable evidence for forensics analysis purposes, helping understand attacks and prevent them in the future.

While the IDS concept was borrowed from the Information and Communication Technologies (ICT) world, its deployment in IACS must obey a specific set of restrictions calling for the development of domain-specific approaches [61]. As a result, several proposals for IACS IDS have been presented over the past years, covering several levels of the automation infrastructure, from the field-level, as it is the case for the Shadow Security Unit (SSU) PLC security monitor [62], to higher levels, as it is the case for the distributed security framework for IACS presented by Rosa et al. [63]. According to its target, an IDS system can be classified as Network Intrusion Detection System (NIDS) or Host Intrusion Detection System (HIDS) [64].

IPSs are the natural counterpart for IDSs, providing active response capabilities, with Intrusion Detection and Prevention Systems (IDPSs) combining both detection and response capabilities [64]. However, it must be said that automatic reaction mechanisms are often avoided by CI operators, due to the risk of a knowledgeable attacker abusing them for its own purposes.

Components such as IDS can not provide an encompassing level of protection for the infrastructure, a situation that requires the adoption of a structured approach capable of providing collection, analysis and storage for monitoring information coming from the entire IACS infrastructure. SIEM systems, which will be next presented, constitute one of the most popular approaches to consolidate diversified and relevant information, leveraging it for analytics purposes.

2.2.4 Security Information Event Management

SIEM systems are designed to collect and correlate security log data (record of events that occurred on a computer or network device) from a wide variety of sources within organizations, including security controls, operating systems, and network infrastructure, systems and applications. Their data sources include log data and network telemetry data from flows and packets. Typically, their blocks include source device, log collection, parsing normalization, rule engine, log storage, and event monitoring [65]. Once the SIEM has the log data, data are normalized and further analysis generates alerts when suspicious activity is detected. Moreover, SIEM provides reports on the request of administrators. Some SIEM products can also act to block malicious activity, for instance by running scripts (e.g. triggering reconfiguration of firewalls and other security controls). Forensic investigations will benefit from correlating the collected data with the information from the context, including assets, users, threats, and vulnerabilities.

As stated by Gartner [66], SIEM technology provides Security Information Management, log management, analytics, compliance reporting, and Security Event Management. They provide real-time monitoring and incident management for security-related events from networks, security devices, systems, and applications.

SIEM technology typically supports three primary use cases: advanced threat detection, basic security monitoring, and forensics and incident response. Forensics and incident response contributes with dashboards and visualization capabilities, as well as workflow and documentation support to enable effective incident identification, investigation and response. Basic security monitoring includes log management, compliance reporting, and basic real-time monitoring of selected security controls. In the case of advanced threat detection, it includes real-time monitoring and reporting of user activity, data access, and application activity, incorporation of threat intelligence, business context and ad hoc query capabilities. At the most basic level, a SIEM system can be supported by rules or employ a statistical correlation engine between event log entries. Pre-processing may happen at collectors, with only part of those events being moved to a centralized management component, reducing, in this way, the volume of information being communicated and stored. Notwithstanding, this approach can discard important events too early [67].

Sun et al. [68] presented an event-linked network model to query and organize big volumes of data. In this model, events are primary units in organizing the data, whereas links represent the association among them. This model is applied in Cloud or virtual-environment analysis, as a huge quantity of involved data, such as the case of the Internet service provider with SIEM solution having a huge quantity of data at centralized locations.

In 2021, Gartner Magic Quadrant for SIEM identified the following market leaders: Exabeam, IBM, LogRhythm, Rapid7, Securonix, Splunk, Splunk, HPE, and Intel Security [66]. A survey of those solutions, including an analysis of external factors affecting the SIEM landscape in the mid and long-term, can be found in [65]. Authors concluded that SIEM systems are slowly converging with Big Data analytics tools.

2.2.5 Other Security Analytics Platforms

Active security and forensic capabilities are typically offered separately by different security systems [69]. While SIEM has pushed for the development of complementary approaches for collecting and analyzing event data to identify and respond to advanced attacks, several operators have found them to be somehow limited due to reasons such as the lack of orchestration capabilities, prompting the emergence of a new generation of security analytic technologies. Next, we introduce some of those tools and technologies.

Endpoint Detection and Response (EDR) software is complementary to SIEM, extending detection and response capabilities by acting as an additional log source. According to Gartner, EDR is "a SaaS-based, vendor-specific, security threat detection and incident response tool that natively integrates multiple security products into a cohesive security operations system that unifies all licensed components." [70] EDR solutions record and store system-level endpoint behaviors, and include several data analytics techniques to detect suspicious system behavior, provide contextual information, block malicious activity, and provide remediation suggestions to restore affected systems [71].

While the primary incident response tools for security teams are EDR platforms, emerging Extended Detection and Response (XDR) products integrate a set of security products into a cohesive security incident detection and response platform. Gartner defines them in a category aggregating and correlating telemetry from different sources to synthesize and draw conclusions to enable automated response actions. In comparison to SIEM and Security Orchestration, Automation and Response (SOAR) tools, XDR offers a higher level of integration of their products at deployment, with a focus on threat detection and incident response use cases. Moreover, while a SIEM can be be delivered in a Software as a Service (SaaS) model, most XDR products are developed using new cloud-native architectures, making them an emerging alternative or complement to existing SIEM tools. Despite such advantages, some of the SIEM use cases, such as generic log storage or compliance, are not replaced by XDR solutions [66].

The combination of Elasticsearch, Logstash, and Kibana from Elastic Stack, OpenSOC, Apache Metron, and other tools leveraged with or natively using Big Data platforms like Hadoop offers data collection, management, and analytics capabilities. Some security analytics platforms are available [72, 18, 73, 74, 75, 76], and many open-source solutions have been developed supporting a wide spectrum of security-based analysis [77, 78]. OpenSOC, for instance, was one of the open-source platforms incorporating scalable security analysis tools and providing, in many cases, an alternative to the expensive commercial SIEM-frameworks. It provides real-time security analysis and data analytics. The OpenSOC framework also integrates a great part of the Apache stack, such as Hadoop [79], Kibana [80] and Elasticsearch [81] to store, index, and enrich data sources, including network traffic and application log data. Apache Metron [78] is another example of those platforms, and the successor of OpenSOC. It also provides a full-stack software infrastructure for the analysis and detection of network intrusions, zero-day attacks, and advanced persistent threats. IBM QRadar is another security platform able to scale up in terms of performance and storage. It is designed to monitor, correlate and store large volumes of data. It includes searching capabilities over the indexed data and also provides key capabilities such as risk management, vulnerability management, incident forensics, incident response, and application. It also includes incident forensics to enable visibility to the questions who, what, when, where, and how a security incident occurred [82].

There are examples of security platforms specifically designed for CIP, as it is the case for the platform proposed by Granadillo et al. [83], where processes' events are received from multiple sources affecting a water CI to be correlated to generate security alarms accordingly, indicating the presence of a threat or an attack in the monitored systems. Another example is [61, 84], which presents an hierarchical two-level correlation architecture for electricity grids, which later evolved into a Big Data solution, presented in [63].

2.2.6 Summary

This section was not intended to provide an exhaustive overview of the field, but rather to provide an encompassing perspective about the specifics of the CIP and IACS domains from a security standpoint, providing the reader with broad knowledge about the problems, limitations and the solutions being used by CI operators. These concepts are key for understanding the next two sections, which will be devoted to discussing the functions and role of forensics and compliance auditing capabilities.

2.3 Forensics

Forensics refers to the application of science and technology to an investigation process to find out the facts in criminal or civil litigation. It comprises collecting evidence of the occurred facts, records and digital trails that can be legally used for criminal prosecution [12]. Based on this data, backward tracing can be used to reconstruct the chain of events that led to an incident, with forward tracing helping understand the repercussions of that event. Moreover, such procedures are often undertaken for reasons other than legal, such as root cause analysis of system failures or incorrect procedures, based on operational traces.

This section will start with the definition of what a Forensic Process is, followed by a description of the associated investigation processes and a definition of digital and network forensics. Next, a brief survey on digital forensics is provided, followed by a discussion of the impact of cloud computing on forensics processes, data privacy aspects, forensics readiness. Forensic schemas and interoperability formats are also discussed, together with query and visualization tools. The section closes with an overview of CPS forensics and the impact of Internet of Things (IoT) and Industrial IoT (IIoT) on the forensics domain.

2.3.1 What is a Forensic Process?

Overall, the definition of what constitutes a forensic process is mostly coherent across different literature, regulatory and/or standardisation sources. For instance, Rani and Geethakumari [85] defined computer forensics as the science allowing to identify, extract preserve, and describe the digital evidence stored in digital devices and networks that can be legally admissible in court for any cyber-crime or fraudulent act. The National Institute of Standards and Technology (NIST) [86] defined digital forensics or computer forensics as a scientific method to identify, collect, examine and analyze data, also comprising a systematic investigation process of crimes in which evidence can be retrieved from the media contents found in the associated digital device. Casey [16] defined digital forensic investigation as a complex and time-consuming activity in response to a cybersecurity incident or cybercrime that should answer these questions: what happened, when, where, how, and who is responsible.

Attacks against ICS and SCADA systems, such as Stuxnet [53], Dragonfly [87] or Flame [88], highlighted the relevance of forensic investigations for post-mortem analysis. In many cases, this has prompted operators to design and implement defense and forensic readiness strategies, encompassing actions and procedures to provide the capabilities to diagnose incidents and support the identification and prosecution of attackers. Such capabilities can also be helpful to deal with harmful events such as natural disasters or hardware malfunctions, by providing the capabilities to analyse the underlying SCADA Information Technology (IT) system [22]. These approaches gain more significance as breaches in SCADA systems may cause dangerous consequences for both human life and the infrastructure, beyond significant monetary loss or service disruption [89, 90, 91].

While most cybersecurity tools are focused on detecting and monitoring, forensic tools are focused on collecting and recording traffic and events while, at the same time, providing feedback information to the security actors. Relevant operational events are monitored and recorded using a forensic approach akin to a system black box, providing the means to investigate and retrieve evidence. Also, it should be possible to trace the attack, prepare mitigation actions, adjust countermeasures, apply damage control policies or even recover from partial or total failure.

2.3.2 The Forensic Investigation Process

According to Hunt [69], the main purpose of intrusion analysis and collection of forensically sound data is to seek answers to the following questions:

- Who is responsible for the incoming intrusion or outgoing data transfer?
- What kind of equipment and services were involved?
- Were they able to do this because of limitations of incoming or outgoing security mechanisms?

According to Whitman and Mattord [64], the forensic investigation process follows the basic methodology:

1. Preparation, including the identification of the relevant items bringing value to evidence.
2. Acquisition of evidence with preservation, without alteration or damage.
3. Assure at every step the evidence is verifiably authentic and remains unchanged since the time it was seized.

4. Evidence examination and analysis of the data without risking modification or unauthorized access.

5. Report the findings to the proper authority and take the lessons learned.

In this context, evidence may refer to a physical object or documented information about a past action that may help disclose the intent of a perpetrator [64], support an alibi [16] or provide legally admissible proof. It should be checked whether it was obtained legally as a result of a court order or by another order of an authorized institution or person.

The forensic investigation process should be able to capture evidence before processes or services on the running system overwrite useful volatile data [22]. This may be justified for a wide array for scenarios such as disputed transactions, allegations of employee misconduct, presenting legal and regulatory compliance, negligence and breach-of-contract charge avoidance, assisting law enforcement investigations, meeting disclosure requirements in civil claims, or supporting insurance claims when a loss occurs.

Digital evidence comprises the data stored or transmitted using computing means, which may be used for incident analysis and/or proof purposes. In the course of a forensic investigation, it should be assured that all available digital evidence is not only protected from deletion but also from modification without appropriate authorization [92], with all steps being recorded [93]. This is vital for integrity purposes, also protecting data from anti-forensics activities, which comprise the techniques aiming at hampering the forensics process, destroying or modifying any digital evidence [94].

For forensics applications, digital evidence integrity is a key property as its violation invalidates the admissibility of data for proof purposes. A cryptographic hash can be used to assess the integrity of the evidence, as well as the copies used along with the examinations and analysis results of compromised systems — this way, an examiner can rely on data he is working on, confident is exactly the one originally captured. A hash can be computed in the moment data is produced and used until the moment integrity is checked, allowing to detect abnormal situations, for instance, when an inconsistent data image does not accurately represent the state of the data acquisition [22].

Data provenance, which provides contextual information related to the origin of data, can support detailed explanations on how a specific state was reached, being included in evidence as a statement from the person carrying out the extraction.

It specifies the source system, the acquired artifacts to denote the chain of custody as an audit trail of all activities, and a timestamp of data extraction [95]. Several approaches have been proposed to implement provenance tracking (e.g., ES3 [96], PASS [97], SPADE [98], Story Book [99], TREC [98]). Zafar et al. [100] proposed a taxonomy of existing secure provenance schemes.

Data provenance analysis can be used to extract host events into provenance graphs that represent the entire system execution and help causal analysis of system actions. Some of the recent works focused on fidelity [101, 102, 103, 95, 104, 105], while others focused instead on efficiency [106, 105, 107, 108, 93, 109, 110, 111]. Data provenance can also be used to reduce alert fatigue [112] and identify intrusions [113, 114, 115].

The highly volatile nature of digital evidence implies that a careful integrity safeguarding approach should be followed. This chain of custody process intends to help preserving the integrity of the information, providing a non-repudiable chronological trace [116] detailing how evidence was acquired, processed/analyzed, handled, stored, and protected, to be presented as admissible evidence in court [117]. A chain of custody ensures the collected evidence is not modified along the investigation process and from the moment it was collected until it is presented [118]. Prayudi and Sn [119] provided an overview of the state of the art about challenges in the digital chain of custody. Cosic and Baca [120] presented a digital evidence management framework aiming to improve the chain of custody of digital evidence in all stages of the digital investigation process, supported by the use of SHA-2 hash function for the digital fingerprint of evidence.

2.3.3 Digital and Network Forensics

In 2008, the American Academy of Forensic Sciences (AAFS), one of the most widely recognized professional organizations for all established forensic disciplines, recognized forensic computer-related crime investigation as a legitimate area, for which a new Digital and Multimedia Sciences section was allocated [16]. This enabled the development of a common ground for the forensic science community to share knowledge and address current challenges [121].

Digital forensics deal with evidence extraction, preservation, identification, documentation, and analysis using well-defined law enforcement procedures, establishing clear lines within the chain of custody. According to McKemmish [122], digital forensics can be broadly considered as having four stages, namely: identification, preservation, analysis, and presentation. Several methods have been proposed in the literature, aiming to formally reconstruct the sequence of events executed during the incident using proven methods [123]. However, the significant growth in the volume of data and the number of evidence items coming from a wide range of sources raises new challenges when conducting digital forensic investigations.

Imaging, hashing, and carving are among the available techniques used by digital forensics investigations. Imaging consists of copying storage media to be examined as evidence. Such evidence can be compromised by modern Operating Systems (OS), due to the operations in the background on the file system, such as indexing or journal resolution [124].

Cryptographic hashing or signing is used to provide authenticity and integrity of files and other evidence. For instance, Afzaal et al. [125] presented an architecture aiming to overcome the limitations of the classic RSA algorithm to provide event integrity protection, allowing a group of n parties to participate in the digital signature process to enforce authenticity and non-repudiation. As for hashing techniques, while MD5 hashing was originally adopted by the forensics community [126], it was later superseeded by SHA-1 as a NIST federal standard, with a transition timeline towards SHA-2 or SHA-3 being announced in December 2022.

Carving refers to the forensic tools to scan unused disk blocks to find and recover deleted data. Carving uses known header and footer signatures to combine the non-used nodes into the original deleted files. Mikus [127] conducted an analysis supported by the use of carving techniques. Recent advances in carving included recovering capabilities of fragmented files with more accuracy [128].

Within the digital forensics field, network forensics is concerned with monitoring network traffic to assess anomalies and attacks. To investigate such attacks, several data sources are available, including packet filters, firewalls, intrusion detection systems, honeypots, sinkholes, surveillance and vulnerability scanning systems [69]. Software Defined Networking (SDN) was also leveraged by Bates et al. [129] to deploy capture points over the network to have a holistic view of network activity, which can be used for forensics purposes. Spiekermann and Eggendorfer [130] also discuss the challenges of executing network forensics investigations in virtual networking environments with tunneling and SDN. Nevertheless, one of the most important challenges in terms of network forensic has to do with the required data storage and computing capabilities [131]. For instance, even a moving window of some hours covering the duration of relevant real-time traffic may require a significant amount of storage from a computing cluster, something that may be aggravated in case of sustained attacks

2.3.4 A Brief Survey on Digital Forensics

There is a considerable corpus of related literature on digital forensics, whose focus is equally diverse. In this line, Casino et al. [132] reviewed several works in the field of digital forensics and identified their main topics and challenges.

Regarding methodological aspects, Sommer [133] raised awareness of the challenges involved in gathering, analyzing, and presenting digital evidence among directors, managers, and their professional advisers, with Williams [134] providing direction to those who assist in the investigation of cyber security incidents and crimes, not just for law enforcement. Van Baar et al. [135] reported benefits and performance on processing digital forensic investigations on a particular case involving collaboration between different actors.

Regarding the subject of digital forensics frameworks and other architectural developments, Verma et al. [136] proposed a digital forensic framework that uses case information, case profile data and expert knowledge for automation of the digital forensic analysis process supported by Machine Learning (ML) for finding evidence. Hunt and Slay [69] advocate the need of a new forensic analysis approach requiring the implementation of forensic engines, supported by parallel processing while providing flexibility on customizing activities for the analysis of evidential data. Ahmadi-Assalemi et al. [137] presented a federated Blockchain model that achieves forensic-readiness by establishing a digital Chain of Custody and a collaborative environment to qualify as digital witness for post-incident investigations.

Specifically on the scope of CIP, Ahmed et al. [138] highlighted that forensic analysis for ICS is still in its early development stages, due to its specialized nature, together with the prevalence of proprietary and poorly documented protocols. Nevertheless, Kilpatrick et al. [139, 140] and Chandia et al. [141] proposed an architecture allowing to capture and analyse sensor data and control actions in a SCADA network (using agents located at strategic positions within the network to capture the local traffic and forward a relevant portion of packets, called a synopsis, to a data lake). Also, Elhoseny et al. [142] proposed a conceptual framework for automated and secure forensic investigation in modern complex SCADA networks, intentionally designed to comply with green computing requirements. Eden et al. [26] suggested deploying forensic hardware instrumentation connected to field device artefacts as a wrapper implemented at physical level, in order to improve the availability and recovery of information for cases where SCADA devices have restricted physical access. Valli [143] created a framework that produces forensically verified signatures for the Snort IDS for known and published vulnerabilities of SCADA, enabling investigators to trace exploits during analysis.

There are also many works focused on identifying existing gaps and/or challenges, some which also proposing suitable solutions to address them. For instance, Huang [144] realized that the characteristics of big data complexity (e.g., volume and variety) make traditional data mining algorithms unsuitable to retrieve knowledge in forensics scenarios, something that Quick and Choo [145] also address, highlighting the challenges posed to digital forensic analysis (considering the ongoing growth in the volume of data seized and presented for analysis). These conclusions are reinforced by Koven et al. [146], who noticed a lack of suitable analysis tools for large datasets – despite the focus on email datasets, the findings are likely to be broadly applicable to other types of sources. Stelly and Roussev [147] presented the concept and prototype implementation of the first domain-specific language aimed at providing a practical and formal description of digital forensic investigations as a computation.

Regarding causality analysis for attack investigation, several works have considered provenance graphs for tracking based on audit logs. Their approach is mainly related with the sub-topics based on causality, anomalies and learning analysis. As an example, Zipperle et al. [148] surveyed the literature on provenance-based IDS and proposed a taxonomy. Alsaheel et al. [149] proposed a framework to identify and reconstruct end-to-end cyber attack stories from unmodified systems and software audit logs. Kwon et al. [150] developed a model supported by causality-based inference for audit logging. Ma et al. [151] also proposed a provenance tracing system capable of alternating between logging and unit-level taint propagation, and event processing.

Considering digital forensics performance and assessment, Ayers [152] proposed several metrics for measuring the efficacy and performance of forensic tools, such as speed, accuracy, completeness, reliability, and auditability. Roussev and Richard [153] discussed the need of distributed forensics approaches, highlighting the performance benefits inherent to distributed computing and proposing a distributed digital forensic tool to centralize data and distribute processing over multiple devices, with background preprocessing capabilities of multiple concurrent searches. Daubner et al. [154] presented research towards verification of forensic readiness in software development, with a focus on produced digital evidence.

The topic of anti-forensics techniques and prevention is also addressed in the literature and has been the subject of research [155]. As an example, Rekhis and Boudriga [123] developed and demonstrated an anti-forensics aware theoretical digital investigation approach, with Noura et al. [156] proposing a solution to prevent anti-forensics techniques targeting log availability and integrity (such as wiping and injection attacks), using encryption, fragmentation and authentication for data distribution across several storage nodes.

2.3.5 Cloud Forensics

The cloud computing paradigm, which shifts information from endpoint devices to a provider infrastructure [157], has become popular among many organizations due the potential cost and resource efficiencies it might entail, also offering several operational benefits for CI, including data redundancy, data availability and survivability when essential system components are isolated or lost [158]. Its introduction raises new and substantially different challenges for forensics, since the target environment is no longer isolated and data is no longer acquired under the investigator's control – thus, there is an evident need to go beyond traditional approaches [159]. In this scope, NIST identified 65 challenges of conducting digital investigation in cloud environments, also pinpointing existing technical gaps [160].

Forensics activities in the cloud present important challenges. Aspects such as the distribution of computing and storage (which have an impact in terms of increased attack surface), geographical storage dispersion across distinct jurisdictions with specific procedures and laws, privacy, or even the lack of norms on aspects such as Service Level Agreement (SLA) regulating the client and Cloud Service Provider (CSP), raise new complex challenges to forensics investigators [161]. Even if the CSP is compliant with the law enforcement agencies in its respective jurisdictions, cloud forensics may be a costly and time-consuming procedure [162], moreover considering how much storage may be used on the tenant storage pool.

The increased need for forensic investigations involving cloud-based scenarios has prompted the emergence of cloud forensics [163], a hybrid approach encompassing remote, virtual, network, live, and large-scale operations, geared towards the generation of digital evidence from cloud environments.

Moreover, cloud-based forensic architectures (which may still be used with private clouds) can be seen as an online solution to help remove any hardware dependency [164] [165]. This can enhance forensic experts and investigators activities with the tools and processes to be applied in the digital investigation of the collected evidence such as sorting, indexing, data recovery or bookmarking, among others. In this line, van Beek et al. [166] shared the lessons learned from providing Digital Forensics as a Service (DFaaS) implementations for almost 10 years, discussing the organizational, operational and development perspective, in a forensic and legal context. Zawoad et al. [167] presented an architecture for a secure cloud logging service, collecting information from different sources around the datacenter, both software (hypervisors) and hardware (network equipment), in order to create a complete landscape of the operations in a datacenter. Similarly, Zawoad et al. [168] proposed Secure-Logging-as-a-Service to enhance forensic investigation in the cloud ecosystem that enables the acquisition of admissible log evidence in the cloud.

Manral et al. [169] surveyed the cloud forensic literature published between January 2007 and December 2018, categorized using a five-step forensic investigation process, and included a taxonomy of existing cloud forensic solutions as well. Ruan and Carthy [170] described the need for new forensic tools or to extend the existing digital forensic tools to make them fit into Cloud frameworks, also presenting a forensic tool for OpenStack Cloud which works through a daemon running in a compute node delivering network logs and the images of instances to the dashboard. Rani and Geethakumari [85] describe a snapshot-based approach to face the dynamic nature of Cloud in which the CSP takes a snapshot of a suspected Virtual Machine (VM) when an anomaly is found by an IDS, isolating it from the network and storing it in permanent storage. A similar approach was suggested by Hibshi et al. [171], which presents a study highlighting a number of usability points that need to be taken into consideration when designing and implementing digital forensics tools, also proposing an efficient approach to forensic investigation in the cloud using VM snapshots. Yu et al. [172] presented a framework for automated detection of anomalies in a cloud environment including a module for cloud forensics with learning capabilities embedded in the management layer of the cloud infrastructure. Patrascu and Patriciu [173] claimed there should be a revision of the classic network forensic principles, and a reorganization of well-known workflows, taking in consideration tools such as ML or large scale computing.

A hypervisor-based approach has been considered for threat monitoring and forensic analysis in [174], where the hypervisor provides the means for examining VMs, by monitoring activities performed at a layer between the hardware and the virtual environment. The potential of this approach was demonstrated by Mishra et al. [175], which presented a taxonomy of hypervisor forensic tools and demonstrated how evidence that can be found in a VM, at the hypervisor and host system layers. Saibharath and Geethakumari [176] proposed a remote forensic evidence collection and pre-processing framework for cloud nodes that collects VM disk images, logs and network captures, pushed periodically into a Hadoop distributed file system. Huseinović and Ribić [177] evaluated the virtual machine memory dumps from Oracle VirtualBox and VMware VMs, with Cheng et al. [178] proposing a similar concept for a lightweight live memory forensic framework based on hardware virtualization that can build a virtualization environment on-the-fly. Also, Zhang et al. [179] and Guangqi et al. [180] proposed a KVM-based approach to acquire both data and VM meta-data, using the access and control privileges of a VM host to acquire VM-related information.

In alternative to VMs, the combination of containers and microservices can help improving isolation between components in an cloud-native application, with a reduced overhead. However, the topic of forensic investigation in containerized environments is a complex task raising new challenges [130], due to the fact that instances can be started and stopped on different systems, which results in an ongoing change in the structure of the network, as well as their shorter life span which implies that container instances may not be available anymore when a investigation process is triggered. Which such environments in mind, Sharma et al. [181] presented a deep learning approach for containerized application runtime stability analysis, and an intelligent publishing algorithm that can dynamically adjust the depth of process-level forensics published to a backend incident analysis repository. Stelly and Roussev [182] presented a scalable containerized framework for forensic computations.

Other works have proposed procedures and standards for forensics activities in the cloud. Saibharath and Geethakumari [183] developed a framework for cloud forensics in OpenStack, according to the Infrastructure-as-a-Service model and using existing forensic tools, which is able to take live snapshots, image evidence, packet captures and log evidence.

Banas [184] discussed the memory acquisition process to be placed in a Kernel-based Virtual Machine (KVM) storage and memory images in OpenStack without any CSP interaction in a self-service Cloud environment. The NIST Cloud Computing Forensic Science Working Group (NCC FSWG) [185] was established to research on Cloud forensic science challenges in the Cloud environment and to develop plans for measurements, standards and technology research to mitigate the challenges that cannot be handled with current technology and methods. Almulla et al. [162] proposed a forensic procedure based on the NIST model to examine private cloud VM snapshots, using existing digital forensic tools, being able to successfully acquire data without the need to transform the snapshot files.

There is also an emerging line of work regarding the use of Blockchain for FCA purposes, providing a tamper-resistant ledger mechanism which matches the needs for non-reputiation and chain of custody purposes. In this scope, Liang et al. [186] proposed a decentralized and trusted cloud data provenance architecture using blockchain technology. Also, Awuson-David et al. [187] and Ahmadi-Assalemi et al. [137] presented Blockchain-enabled methodologies and frameworks for keeping a chain of custody of the digital forensic log evidence from the cloud ecosystem, to ensure trustworthiness, integrity, authenticity and non-repudiation. Finally, [188] proposed a cloud forensics taxonomy and denoted the trend towards the implementation of digital provenance assurance using blockchain technology.

2.3.6 Data Privacy Protection in Digital Forensics

Privacy can be defined as the right to control who has information about someone, including activity tracking [189]. Some of the concepts raised in privacy laws intend to establish limits restricting data use or its correlation from multiple sources, often mandating anonymization or removal of personal data from records [189]. Such an example is the General Data Protection Regulation (GDPR), introduced in 2016 to bring protection to personal data [190], making it mandatory to obtain consent on the use of personal data.

The problem of balancing forensic investigation needs with privacy protection requirements is discussed by Aminnezhad et al. [191], with Dehghantanha and Franke [192] having established the foundations for the definition of privacy-respecting digital investigation as a new cross-disciplinary field of research, also reviewing the state of art in this field. Despite the large number of digital forensic models discussed in scientific literature, just a few of them are considering data privacy along the digital forensic investigation process, many of which are either tailored for specific environments or included as an independent module [136].

van Staden [193] proposed a framework to protect privacy in multi-user environments that are subject to post-incident forensics investigation, supported by profiling and filtering mechanisms. Law et al. [194] described a way to protect data privacy using encryption, proposing the introduction of simultaneous data encryption processes by email servers and indexing of related keywords, allowing an investigator to give a keyword input to the server owner, who has the encryption keys, to get back the emails that contain the keyword. Also regarding encryption-based approaches, Hou et al. [195] proposed a mechanism to protect data privacy on a third-party service provider's storage center, using homomorphic and commutative encryption, with Hou et al. [196] describing a similar solution.

As for identity or knowledge-based approaches, Shebaro and Crandall [197] used an identity-based encryption mechanism to carry out a network traffic data investigation in privacy preserving setting. Croft and Olivier [198] proposed a mechanism where data is divided into layers of sensitivity, placing less private data on lower layers, and highly private data on higher layers. In this schema, access to private information is controlled by initially restricting investigator access to the lower layers, requiring further proof to get access to higher-level information.

2.3.7 Digital Forensic Readiness and Forensics-by-Design

For many, the possibility of a security incident should be regarded as a certainty rather than a possibility [199]. In fact, when incidents happen, the priority is often restoring normal operational levels, instead of making an effort to collect and preserve as much forensics evidence as possible, eventually to be admitted to a court. The generalised approach is mostly reactive: first restore operational capacity, and then carry out investigations and seek evidence. As a result, evidence might be lost or rendered unsuitable as proof.

Forensic readiness is a concept that contributes to minimise the aforementioned problems. It suggests taking proactive actions to capture evidence even before or during an incident and before investigations are started. This helps not only to save time and money, but also to mitigate potential incidents and ensure business continuity and compliance with minimal disruption and interruption of operations. Kruger and Venter [200] provided a systematic literature review to identify topics where digital forensic readiness is included. However, as denoted by Iqbal et al. [201], digital forensic readiness for CIP is still immature, judging by the lack of published research or industry reports.

Forensic readiness comprises planning activities to collect, preserve, protect and analyze digital evidence to be effectively used [202]. It can also assist in fulfilling the increasing demand for the implementation of security practices addressing compliance with organizational policies and regulatory requirements, providing the means to deploy continuous monitoring and review processes supported by the already collected forensic data. This approach can help fill that gap, since even common standards such as the ISO 9001 series and regulatory frameworks for B2B relationships (e.g. supply chain risk management) do not account for best practices in the CI and IACS security domains.

Forensic-by-design extends the concept of Digital Forensic Readiness. Similarly to Security-by-design, it advocates the integration of forensic requirements into the system's design and development stages. Ab Rahman et al. [199] proposes a system and software engineering driven Forensic-by-design framework, with an emphasis on Cloud computing systems. Akilal and Kechadi [203] investigated the potential adoption of Forensic-by-design in cloud computing systems, with [204, 205] suggesting the application of Forensic-by-design (FbD) strategy to enhance digital forensic readiness.

Moreover, several proposals for implementing digital forensics readiness are documented in the literature. For instance, Daubner and Matulevičius [206] proposed the introduction of forensic readiness mechanisms within security risk management to refine specific requirements of forensic-ready software systems, by re-evaluating the taken security risk decisions with the aim of providing trustable data when the security measures fail. Elyas et al. [207] presented a digital forensic readiness framework through a series of expert focus groups to discuss the critical issues facing practitioners in achieving digital forensic readiness. Also, De Marco et al. [208] proposed a reference architecture for a Cloud forensic readiness system. Mouhtaropoulos et al. [209] classified forensic investigation frameworks to expose gaps in proactive forensics research and reviewed prominent information security incidents with regard to proactive forensics planning. On a more network-focused scope, Endicott-Popovsky et al. [210] proposed a framework for operationalizing network forensic readiness, with Ngobeni et al. [211] proposing a wireless forensic readiness model designed to help monitor, log, and preserve wireless network traffic for digital forensic investigations.

Considering readiness maturity assessment, Ariffin and Ahmad [212] presented five indicators for the maturity and readiness of digital forensics, with Elyas et al. [213] describing an approach to identify the factors that contribute to digital forensic readiness and how these factors work together to achieve forensic readiness in an organization. Iqbal et al. [214] presented a study on the current support for forensic readiness of CI, highlighting the involved key challenges and providing a literature review on the subject. Also, Alenezi et al. [215] presented a framework to investigate the factors that facilitate the forensic readiness of organizations.

2.3.8 Forensic Schemas and Interoperability

In a general way, interoperability is concerned with making it possible for components or systems coming from different vendors to easily communicate and interact with each other. When investigation processes require evidence exchange between investigators, the use of different tools for the reconstruction of events or analytical purposes, the absence of standardised digital evidence formats can become a serious obstacle. Thus, it is particularly important to develop information interoperability mechanisms by means of common Forensic Schemas.

A standardized approach for representing and sharing digital forensic information is also useful to help investigators collaborate when incidents involve different jurisdictions. Similar challenges were also recognized in traditional investigations of violent crime and led to the development of the US Federal Bureau of Investigation's Violent Criminal Apprehension Program (ViCAP) and Royal Canadian Mounted Police's Violent Crime Linkage System (ViCLAS) programs. These programs enabled the correlation of all the available information from unsolved violent crimes in disparate regions, trying to find links between them.

There have been several schemas proposed in the past for representing digital forensic information, but these have not been widely adopted [216] [217] [218] [219] [220]. Also, Garfinkel [221] proposed a XML schema (DFXML) for easier interoperability between forensic extraction and visualization tools, primarily developed to represent the output from tools used to analyze storage media, including file system parsers, file carvers, and hash set generators.

Casey et al. [222] conducted a review of digital forensic data schemas, including DFXML, also proposing the CybOX schema for handling forensic data. CybOX is an open-source, community-driven effort to develop a standardized representation of digital observations led by the US Department of Homeland Security (DHS) office of cyber-security and communications.

The XML-based XIRAF system was created by the Netherlands Forensic Institute (NFI) to support digital forensic analysis, storing its data using a parent–child structure within a centralized database accepting structured output and searching tools [223]. Bhoedjang et al. [224] described the second generation of this analysis system and outlined the complexity of importing different file types and analyzing and preprocessing files before storing them in databases. Van Baar et al. [135] outlined the latest iteration of this system, incorporating Cloud features, and discuss faced issues.

The Advanced Forensic Format (AFF4) has taken another approach for the representation of digital forensic information [225] [226], using the Resource Description Framework (RDF), a general purpose representational formalism for knowledge representation. Although the majority of digital forensic tools do not support AAF4, Google Rapid Response (GRR) uses the AFF4 data model to store information in a MongoDB database [227]. The AFF4 data model is flexible. However, the use of RDF requires the adoption of a shared supporting ontology. While in the community there is no consensus on such ontology to exchange digital forensic information, Casey et al. [222] addressed this gap with an ontology that could be used as a basis for community consensus.

2.3.9 Visualization and Searching Tools

When it comes to the forensics practitioner toolset, usability is a crucial aspect not to be disregarded [171]. Specifically, Osborne and Turnbull [228] pinpointed the importance and need for tools incorporating adequate visualization capabilities for digital forensic data, claiming that there is a lack of algorithms to identify relationships, normalize data, incorporate multiple data sources, and provide effective visualization methods, all of which are important to retrieve further insights from evidence. Following this same line of thought, Osborne et al. [229] highlight the importance of considering architectures incorporating familiar visualization tools and algorithms that could be able to include distinct data sources, normalizing and correlating data, later proposing a conceptual framework able to Explore, Investigate, and Correlate (EIC) [228]. Tassone et al. [230] also highlighted the importance of visualization in forensic tools, pointing out that many existing solutions where just simple layouts to search and display basic tabular data, also presenting a proof of concept including a database schema designed for third-party forensic data storage and visualization.

Irfan et al. [231] describes a virtual cloud environment incorporating visualization capabilities designed to provide visibility for all security events, allowing to follow activities of cybercriminals, reproduce crude information identifying each respective incident, and execute proactive actions. Also, Aupetit et al. [232] presented a methodology and a tool for allowing the Internet Service Provider (ISP) to assess and visualize threats from an organization's network traffic, allowing them to deal for instance with Distributed Reflective Denial of Service (DRDoS) events. Another example is provided by Setayeshfar et al. [233], which presents a graphical forensic analysis system for efficient loading, storing, processing, querying, and displaying of causal relations extracted from system events to support computer forensics.

Tools such as the Elastic Stack have been widely adopted in industry and academia as a result of their capabilities and performance in terms of log handling. There are solutions for data visualization, including graph generation capabilities for analysis purposes, supported by frameworks such as Kibana [80], Grafana [234], and Prometheus (Prometheus.io), which retrieve data stored in indexed datastores like Elasticsearch [81]. Some of the tools built on ElasticStack are SOF ELK [235] and Plaso [236], that provide rich visualization and parsing capabilities. Despite their capacity for effective forensics and provenance tracking supported by queries, they lack information about the provenance models also don't provide users with many query abilities beyond filtering. Moreover, it should be stressed that while these tools can be used for multiple use cases without the incorporation of analytic inference mechanisms, that's not typically the case in cyber-security analytics or forensics [232].

2.3.10 Forensics Constraints for the CIP domain

Homem [237] identified a series of general challenges regarding digital forensics processes, namely: the rising volume of heterogeneous digital evidence involved in investigations, the evidence-centricity of industry-standard tools, a deficiency in the availability of a highly-skilled workforce, and the great effort required by the largely manual and time-consuming activities involved in the overall process. Besides, CI operational environments add further constraints related to aspects such as complexity, systems interdependency, dependency on information and communication technologies and components provided by third parties, or the deployment of heterogeneous technologies [238].

Typically, forensic investigation can rely on live or dead evidence aquisition. While the latter is performed offline on static data after a system is shutdown, the former collects data from live systems, such as the contents of physical volatile memory, and non-volatile data, such as the data maintained in a storage system. While dead forensics corresponds to the most traditional approach, there was a increasing emphasis on live forensics processes over the past years, as it is the case for network traffic analysis. More specifically, in the case of SCADA systems, the forensic investigator cannot turn it off to capture and analyze data, because this kind of system is supposed to be continuously operational [239] – in such cases, live forensics is a suitable digital investigation methodology [240]. However, and because continuous availability of SCADA systems is a mandatory requirement, forensic investigators should strive to be minimally intrusive, in order to reduce the risks in critical operations while aiming at a rapid response time, to preserve evidence that may be overwritten by runtime processes [241].

It is known that SCADA and IT systems exhibit different behaviours and possess different characteristics, often requiring for IDS and other security mechanisms to be configured according to with the domain of operation [84]. For instance, in a SCADA system, network traffic is more deterministic than in IT networks, in the sense that a system component communicates to other system components following established patterns, frequently with bounded time restrictions. Thus, administrators may impose a set of rules for security purposes, with any non-deterministic behavior flagged as an anomaly – for instance, an IDS might be configured to consider a specific communication pattern as normal [242].

Moreover, the same restrictions regarding live network trace capture can also apply to SCADA stations and other process control or monitoring systems. Any evidence collection tool or technique must avoid imposing overheads that might degrade the system response, interfere with operational indicators or expand the vulnerable attack surface. Overall, a simple rule must be kept in mind: live (or, for that matter, any other) forensics processes must be designed to adhere to the least overhead principle, in line with the recommendations from standards such as NIST SP800-82 [243], which clearly identify the risks associated with intrusive security procedures.

2.3.11 IoT and Industrial IoT Forensics

IoT can be defined as a system of networked smart devices that can be identified, named and addressed [244]. IoT is attracting great attention not only for consumer applications but also in the IACS domain, where they are usually designated as Industrial IoT. Naturally, the introduction of these technologies has increased the amount of generated, transported and processed data, as well as the number of forensically relevant events in consequence of the increasing number of available sensor devices [245].

Considering the emergence of IIoT, organisms such as NIST have defined guidelines [246, 247] to ensure that these infrastructures rely on adequate safety, security, privacy, consistency, dependability, resiliency, reliability, interaction and coordination measures. However, it's not always possible to apply traditional information security measures based on sophisticated encryption algorithms, multi-factor authentication, antivirus programs and firewalls (among others), due to the limited computational and energy resources of some sensor nodes [248], further reinforcing the need for the deployment of proper security monitoring and forensics capabilities.

Stoyanova et al. [249] identified and discussed the main issues involved in the process of IoT-based investigations, particularly all legal, privacy and Cloud security challenges. They also provided an overview of the past and current theoretical models in the digital forensics and frameworks aiming to extract data in a privacy-preserving manner or secure the evidence integrity using decentralized blockchain-based solutions. Vendors such as Infineon, NXP, and STMicroelectronics prepared a position paper for ENISA [250], stating the IoT market failure for cyber-security and privacy, and claiming that there were "no level zero defined for the security and privacy of connected and smart devices," no legal guidelines for IoT device and service trust, and no "precautionary requirements are in place". This paper also predicts that attacks will get more risky and threatening due to the rise of IoT enabled cars, CI, and health applications. In the same line of thought, Chehri et al. [251] identified the trends, problems, and challenges of cybersecurity in smart grid CI in Big Data and Artificial Intelligence (AI).

(I)IoT scenarios require the implementation of adequate forensic and compliance auditing approaches to improve security and privacy. In that regard, Yaqoob et al. [252] investigated studies on the topic of IoT forensics by analyzing their strengths and weaknesses. The authors categorize and classify the literature by devising a taxonomy based on forensics phases, enablers, networks, sources of evidence, investigation modes, forensics models, forensics layers, forensics tools, and forensics data processing. They also enumerate a few prominent use cases of IoT forensics and present the key requirements for enabling IoT forensics, identifying and discussing open research challenges as future research directions.

2.3.12 Summary

The purpose of this section was to introduce and present a series of concepts and topics within the scope of digital forensics, with a view towards its application in the CIP domain. We started by conceptually introducing a definition of forensics activities, followed by a discussion about digital, network and cloud forensics, the latter constituting not only a challenge, but also an opportunity to implement innovative solutions tackling the issues of FCA. The implications of data privacy protection regulations in digital forensics activities were also discussed, followed by a review of the related subtopics of forensic readiness, interoperability, visualization and automation. We concluded with an overview of the current forensics constraints for the CIP domain. Table 2.1 summarizes the reviewed literature on these topics. While some topics are addressed from a more neutral perspective, it must be noted that this is due to the fact that many are still valid in the CIP domain.

Scope	**Works**
Forensics Definitions	Morioka and Sharbaf [12]
	Rani and Geethakumari [85]
Forensic Investigation Process	Casey [16]
	Whitman and Mattord [64]
Data Provenance	Hassan et al. [101]
	Ma et al. [102]
	Bates et al. [103]
	Bates et al. [95]
	Hossain et al. [104]
	Pasquier et al. [105]
	Lee et al. [106]
	Pasquier et al. [105]
	Hassan et al. [107]
	Ma et al. [108]
	Hossain et al. [93]
	Tang et al. [109]
	Liu et al. [110]
	Xu et al. [111]
	Hassan et al. [112]
	Han et al. [113]
	Wang et al. [114]
	Bates and Hassan [115]
Chain of Custody	Giova [116]
	Prayudi and Sn [119]
	Cosic and Baca [120]
	Awuson-David et al. [187]
Forensics for Smart Grid	Abdullah et al. [56]
Forensic Analysis of Intrusions	Hunt and Slay [69]
	Stelly and Roussev [147]
Evidence Definition	Whitman and Mattord [64]
Forensics Readiness	CESG [202]
	Daubner and Matulevičius [206]

	Iqbal et al. [214]
	Elyas et al. [207]
	Ariffin and Ahmad [212]
	De Marco et al. [208]
	Alenezi et al. [215]
	Mouhtaropoulos et al. [209]
	Elyas et al. [213]
	Endicott-Popovsky et al. [210]
	Ngobeni et al. [211]
	Kruger and Venter [200]
Stages of Digital Forensics	McKemmish [122]
Forensic Taxonomy in SCADA	Eden et al. [26]
Forensics for CPS	Cornelius and Fabro [25]
	Valli [143]
Surveys on CPS	Yaacoub et al. [253]
Anomaly Detection	Grubbs [254]
	Gogoi et al. [255]
	Chandola et al. [256]
	Fu et al. [257]
	Ten et al. [258]
	Yu et al. [172]
	Henriques et al. [259]
CIP in Cloud	Alcaraz et al. [158]
Forensic Green Computing	Elhoseny et al. [142]
Forensic Containerized Framework	Stelly and Roussev [182]
Hypervisor Forensics	Cheng et al. [178]
	Jackson et al. [174]
	Saibharath and Geethakumari [176]
	Huseinović and Ribić [177]
	Cheng et al. [178]
	Zhang et al. [179]
	Guangqi et al. [180]
Taxonomy of Hypervisor Forensics	Mishra et al. [175]
Forensics as a Service	van Beek et al. [166]
Data Provenance in Cloud	Liang et al. [186]
Scalable Microservice Forensics	Sharma et al. [181]
SDN and Virtual Network Forensics	Spiekermann and Eggendorfer [130]
	Bates et al. [129]

Table 2.1: Key Forensics for CIP literature

2.4 Compliance Auditing

An audit process represents a systematic, independent, formal, structured, and documented process, usually performed by a certified professional on behalf of stakeholders, aiming to verify if certain criteria match internal policies, external formal standards, and/or legal requirements [260]. Auditing practices help organizations meet such requirements, also providing due diligence, certification, and stakeholder security. Compliance auditing expertise is closely related to and frequently overlaps with forensic processes, since both often share data sources, tools, and techniques.

This section will delve into the topic of compliance auditing, with a view towards its applicability in CIP environments. Starting with an overview of the motivation and context, it will next review existing audit models and standards, concluding with a discussion about logging systems compliance for audit purposes.

2.4.1 Motivation and Context

Policy definition and enforcement are cornerstones of modern security practices. For instance, Yaacoub et al. [253] describes a series of policies encompassing aspects such as employee screening processes before recruitment, privilege suspension outside working hours, or additional activity monitoring for people in charge of sensitive tasks, which contribute to enhance the security posture of an organization.

Compliance auditing checks whether workflows are compliant with organizational policies and rules – thus, each process or transaction may be checked to confirm whether it followed the applicable rules or policies. In case rules are violated, the auditor analyses relevant data to determine causes and recommends actions to prevent future deviations or non-compliance situations. Compliance audit frameworks can also help highlighting misconfigurations – for example, they can used for monitoring access security levels for individual and group accounts and help with detailed reports measuring the security progress.

The compliance auditing process ends up with a report that includes the conclusions and additional information about requirements that have been met and non-compliance situations (if found). It can also highlight the implications and risks of non-compliance or suggest corrective actions [261].

As the surrounding environment evolves, infrastructure and service operators are often forced to adapt to an increasingly complex and constantly changing regulatory landscape. Thus, an organization aiming to implement specific regulatory or standardisation measures should depart from the identification of the entities with relevant technical and/or legal jurisdiction over its domain of activity. In this line, the GDPR [190] regulations constitute an example of a mandatory framework for privacy protection, which applies to organizations within the European Union (EU).

Besides generic or sectorial standards, CI-specific regulations may also be imposed by organizations such as North American Electric Reliability Corporation Critical Infrastructure Protection (NERC-CIP) [262], which publishes a set of security guidelines, as it is the case for Electronic Security Perimeters (CIP-005) and System Security Requirements (CIP-007).

2.4.2 Cybersecurity Audit Models

Businesses are being increasingly pressured to undergo periodic audits and inspections as part of legal and regulatory compliance certification requirements. While such certifications processes are important to reinforce trust at the B2B and B2C levels, it should not be forgotten that their ultimate aim is to ensure that adequate preventive and reactive security mechanisms are implemented, as well as proper handling of sensitive data. Ultimately, it all comes to the establishment and maintenance of suitable levels of data confidentiality, integrity and availability within an organization, which may vary accordingly to the type of applications, data to be stored or processed (e.g., the case of sensitive healthcare data), or geographical location (e.g., regional requirements for data privacy and protection).

From the industry standpoint, an organization may be required to comply with regulations such as Payment Card Industry Data Security Standard (PCIDSS) [263], Health Insurance Portability and Accountability Act (HIPAA) [264], Federal Information Security Modernization Act (FISMA) [265], GDPR, FedRamp, and SOC2. These are examples of compliance drivers prescribing the application security activities. The Institute of Internal Auditors (IIA) also provides guidance in the form of the International Professional Practices Framework Standard 2420 (Quality of Communications) [266], whose aim is to establish guidelines for objective, clear, concise, constructive, complete and timely reporting.

Three different categories of cybersecurity audits were identified by Donaldson et al. [267]. The first corresponds to threat audits targeting cyber threats, aiming to search for evidence in IT environments. The second evaluates the cybersecurity controls mapped against frameworks, regulatory requirements, standards or a specific cyberthreat. The last comprises validation assessments against cybersecurity controls measuring their effectiveness against designed and documented requirements.

The assessment of access control policies is one of the aspects typically resorting to formal reasoning mechanisms to verify application control expressed at design time (for instance with eXtensible Access Control Markup Language, XACML) to dynamically enforce authorization by externalizing access controls. Fisler et al. [268] proposed Binary Decision Diagrams and custom algorithms to check access-control policies. Ahn et al. [269] used answer set programming (ASP) and leveraged existing ASP reasoning models to conduct policy verification. Arkoudas et al. [270] proposed a Satisfiability Modulo Theory policy analysis framework.

Sabillon et al. [271] proposed an audit model for conducting cybersecurity audits in organizations and nation-states. Agrawal et al. [272] introduced an auditing framework for determining whether a database system is adhering to its data disclosure policies by allowing users to formulate audit expressions to specify the data subject to disclosure review. Kaaniche et al. [273] proposed the usage of hierarchical ID-based encryption and signature schemes. Noura et al. [156] presented a security and protection audit that can be done by using an audit management system to collect and store logs in a distributed system. Bouet and Israël [274] presented a security assessment framework including an off-line tool enabling security and vulnerability audits of information systems to be used by system architects to assess the security of the system they are designing during the planning phase. The patent "Critical function monitoring and compliance auditing system" [275] describes a system and method for monitoring, auditing, and flagging compliance issues or other user-defined exceptions. Finally, Slapničar et al. [276] analyzed the effectiveness of internal audit of cybersecurity by developing a Cybersecurity Audit Index composed of three dimensions: planning, performing and reporting.

In the scope of compliance auditing cloud computing platforms, Ullah et al. [277] proposed an architecture to build automated security compliance tools, focusing on auditing remote administration and on diagnosing port protection and clock synchronization. Also, Henze et al. [278] presented a practical approach enforcing data compliance in key-value-based Cloud storage systems. Doelitzscher [279] implemented an on-demand audit architecture for Infrastructure as a Service (IaaS) clouds, based on software agents for identifying anomalies for auditing purposes. Finally, there is also SecGuru, designed to audit Azure datacenter network policies [280].

2.4.3 Standards for Compliance Auditing

The development of cyber Information Security Management Systems (ISMS) is guided by standards such as ISO/IEC 27001 and ISO/IEC 27002. As already mentioned, these standards cover the protection of an organization from cyber-attacks [281]. Domain-specific initiatives were also launched to develop and implement IACS standards to secure SCADA environments, including the ones from NIST, which presented the Special Publication 800-82 and International Electrotechnical Commission (IEC) 62443 [282] [283].

Another example is ISO/IEC 62443-1-1 (Security for industrial automation and CS: Terminology, concepts, and models), which constitutes an ongoing effort towards the improvement of cyber-security, robustness, and resilience design. The ISO/IEC 62443 series standard elements are arranged in four groups, namely: Policies and Procedures, System, and Component Requirements. The Policies and Procedures group is focused on the policies and procedures associated with IACS security, with the Systems group addressing the requirements at the system level. Systems and Component Requirements provide information about specific and detailed requirements associated with the development of IACS products [283]. The Japanese Information-Technology Promotion Agency (IPA) also implemented the Embedded Device Security Assurance Certification Program for provisioning SCADA devices [284].

Nevertheless, and despite these efforts from the academia and industry, there is still a lack of standards for compliance auditing techniques in Cloud domains [285].

2.4.4 Logging Systems Compliance

Logs constitute key data sources to acquire visibility and obtain insights from the operational infrastructure processes, with log analysis being recognised as vital for collecting evidence and retrieving the necessary insights to understand the behaviour of a whole system, as well as its individual components, regardless of the deployment type. For instance, Amazon suggests the use of AWS CloudTrail and CloudWatch [286] for auditing purposes, as a web API offering logs and metrics data.

Being important for administrators, developers and security operators alike (albeit for different reasons), log handling and processing often needs to comply with suitable availability, resiliency and continuous operation requirements – such systems should be sized and ready for possible high-demand situations where the overall system becomes unstable or overloaded, triggering a large number of events.

It is important to rely on a logging system to acquire and deliver information, but also to intelligently process it using insight and analytics. A logging system should provide visibility over its behavior to enable correct predictions. From a security standpoint, log analysis must be reliable and accurate, especially in circumstances involving security incidents or critical situations. Thus, using a inadequate or non-compliant logging system may have several consequences, such as hampering monitoring, diagnosis or forensics procedures, up to the point of potentially voiding the possibility of gathering legally admissible evidence.

2.4.5 Summary

This section presented the key definitions, topics, and related work about compliance auditing standards and regulations. Table 2.2 summarizes the reviewed literature.

Scope	**Works**
Enterprise Cybersecurity	Donaldson et al. [267]
Cybersecurity Audit Model	Sabillon et al. [271]
CI Security Model	Torres et al. [287]
Forensics for incident response	Kent et al. [261]
Auditing Framework	Agrawal et al. [272]
Data Compliance	Henze et al. [278]
Data Privacy Compliance	Kaaniche et al. [273]
	Union [190]
Anti-Forensic Approaches	Noura et al. [156]
Compliance Audit System Patent	Lee et al. [275]
Security Assessment Framework	Bouet and Israël [274]
Formal Reasoning Mechanisms	Fisler et al. [268]
	Ahn et al. [269]
	Arkoudas et al. [270]
Cloud-based Compliance Auditing	Henze et al. [278]
	Ullah et al. [277]
	Doelitzscher [279]
	Bjørner and Jayaraman [280]

Table 2.2: Compliance and Auditing Works

More than pinpointing specific standards, it is important to understand that compliance auditing is a multi-dimensional activity subject to legal, regulatory and standardisation requirements that often vary accordingly with the domain or scope of activity of the CI operator. Such requirements are crucial to guarantee the usefulness and validity of the audit processes, as well as the data sources supporting them.

2.5 Analytics for CIP FCA: the Road Ahead

Analytics corresponds to the set of activities focused on how to extract insights from data, correlating evidence to provide security-related capabilities to system administrators, security analysts, and network and application engineers. Analytics leverage FCA capabilities to improve CIP because they help identifying anomalies and their root cause and then extract evidence.

This section will delve into the benefits and challenges of modern analytics in the era of Big Data, AI and ML, starting with a motivation and following with a discussion about the impact of Big Data technologies on CIP. The intersection of Big Data technologies, AI and ML is also discussed, followed by the topic of Forensics Automation. The section closes with a discussion of anomaly detection techniques for log analytics.

2.5.1 Motivation and context

Modern protected infrastructures are becoming increasingly complex, a situation for which CIP is no exception, with IIoT infrastructures spreading on a massive scale, both geographically and in terms of components [288]. This has the side effect of generating considerable amounts of operational data and evidence, that cannot be properly handled by traditional analysis techniques. This poses a challenge to FCA, requiring the introduction of scalable techniques able to transport, store and process large amounts of data, thus calling for the adoption of Big Data techniques, designed to handle large amounts of data whose volume is beyond the ability of typical vertical approaches [289].

These circumstances also deem unfeasible to manually analyze large amounts of data, requiring practitioners to resort to automated techniques [290], often supported by AI-based techniques (with a particular focus on ML [251], a branch of AI geared towards automating pattern recognition or classification tasks to analyze vast amounts of data to predict or detect certain behaviors, which in the case of forensics, may consist of discovering or detecting malicious activity). These ML and information retrieval techniques have significantly improved in the last years, enabling the extraction of deeper insights from data [291, 292], with many of these analytic frameworks being able to perform effective and efficient data analysis supported by ML models implemented from a few lines of code, also supporting the automation of time-consuming tasks.

2.5.2 The Impact of Big Data Technologies on CIP

One of the most pressing issues when handling large data volumes is the implementation of efficient distributed storage and retrieval technologies. Big Data NoSQL databases address such challenges with technologies such as MongoDB, HyperTable, Cassandra, and Amazon Dynamo offering scalability and performance predictability that is suitable for storing and indexing real-time streams of big datasets [293]. Kalakanti et al. [19] evaluated different NoSQL datastores as a solution to the data and knowledge management challenges to meet the requirements of performance, reliability and scale imposed by the next generation of data historians as a central repository of SCADA systems.

The need to deal with increasingly big data volumes also calls for an increase in the required amount of computational resources, which must be balanced with the need to contain query latencies within acceptable thresholds. To address this problem, Google developed the Google File System [294], as well as MapReduce [295], that was designed to address computational challenges. Several efforts were also made to have those technologies available as open source software, resulting in tools such as Apache Hadoop and the Hadoop File System [296].

As already explained, Big Data technologies are especially suitable for CIP and particularly IIoT, where large volumes of data are produced devices from distributed CPSs, for time series analysis. Specialized Time Series Management Systems (TSMS) have been developed to overcome the limitations of general purpose Database Management System (DBMS) for times series management [297]. For instance, Jensen et al. [297] surveyed the field of TSMSs developed bt the academy and the industry, and organized them into categories. Finally, Wang et al. [298] surveyed TSMSs in industrial and IoT fields addressing the new demand such as large amount and real-time analysis of industrial data.

Big Data also poses significant challenges and stresses out privacy requirements, especially those related to privacy regulation emanated from the EU [299]. In that regard, Gartner predicted that by 2018, 50 percent of business ethics violations will be related with data [300].

2.5.3 Big Data Analytics in the Age of IA and ML

In FCA applications, handling large volumes of data is only half of the equation, with analysis being the other half. Extracting insights and patterns from evidence calls for methods other than manual analysis, thus constituting a natural fit for AI and particularly ML techniques, something that was investigated by Brighi et al. [301], that tried to bridge these technologies with the substantive and procedural rules to be observed during investigation activities.

Regarding forensics applications, Hoon et al. [302] reviewed the literature by addressing the challenges and opportunities of employing Big Data in Distributed denial-of-service (DDoS) forensics, implementing and comparing the performance of multiple supervised and unsupervised learning models, according to their efficiency and accuracy. They found that Naïve Bayes, Gradient Boosting and Distributed Random Forest are the most suitable models for DDoS detection, due to their accuracy and time taken on training.

As for network forensics, Yavanoglu and Aydos [303] reviewed the most commonly used datasets in AI and ML techniques, as primary tools for analyzing network traffic and detecting anomalies. Usman et al. [304] proposed a ML approach supported by Decision Tree algorithms to predict IP address reputation in zero-day attacks, categorized via behavioral analysis to highlight forensic issues in big datasets. Wiyono and Cahyani [305] presented classification algorithms for network forensics based on the identification of network flows that could track suspected botnet activity in the infected network.

Other tools presented by Hassan et al. [112], Setayeshfar et al. [233] implemented models based on AI to assist forensics experts in monitoring the system and detecting malicious behaviors based on known patterns – however, these tools are not designed for manual forensics tasks such as whole system provenance tracking, being often bound to a single proprietary data stream scheme.

In the scope of Compliance Auditing, Moore and Childers [306] presented a ML solution to automatically generate program affinity policies that consider program behavior and the target machine. Similarly, Quiroz et al. [307] relied on unsupervised algorithms to capture the dynamic behavior of systems and the hidden relationship between the high-level business attribute space and the low-level monitoring space. Similarly, Pelaez et al. [308] used supervised models to capture dynamic behavior. Johansen et al. [309] proposed a mechanism for expressing and enforcing security policies for shared data expressed as stateful meta-code operations defined in scripting languages interposed in the filesystem. Gheibi et al. [310] reviewed the state of the art on the use of ML in self-adaptive systems based in the traditional Monitor-Analysis-Planning-Executing (MAPE) [311] feedback loop. Weyns et al. [312] also presented an approach combining MAPE and Control Theory to produce better adaptive systems.

2.5.4 Forensics Automation

Organizations often check whether their security and forensic controls are actually in place as intended using manual assessment procedures. Forensic processes are often no different, being typically time-consuming activities dependent on humans. From this perspective, the lack of qualified human skills and resources can hamper investigation and compliance auditing processes [301].

The use of technology to implement automated processes can streamline forensic investigation tasks fed by large volumes of data. The adoption of automation is therefore seen as an effective strategy to implement forensic processes while reducing the costs and operational errors resulting from human intervention [313], also constituting an emergent field of interest in the research community.

Regarding the introduction of automated procedures, Hayes and Kyobe [314] reviewed the existing research in the field of cyber forensics, identifying current practices and associated challenges that could be tackled by the adoption of automation, as well as the relevant technology that could be leveraged to address such needs. Asquith and Horsman [313] provided an introductory discussion on robotic process automation, a form of service task automation that can improve efficiency in the field of forensics, with Moffitt et al. [315] discussing the automation of repetitive and manual rule-based tasks.

From a more practical perspective, Verma et al. [136, 316] proposed a digital forensic framework that uses case information, case profile data, and expert knowledge for automation of the digital forensic analysis process supported by ML for finding evidence. Also, Patrascu and Patriciu [317] discussed the issues threatening CI systems and proposed an automated learning framework based on ML algorithms to protect such systems that, despite not being focused on forensics applications, can be leveraged for such purpose.

Finally, recent contributions on the use of ML models supporting the automation of self-adaptive IT operations have focused on topics such as observability and AIOps [318, 319] – Notaro et al. [320] has compiled several contributions in this scope.

2.5.5 Anomaly Detection from Log Data Sources

An anomaly corresponds to an outlying observation that appears to deviate significantly from a nominal state or a statistical data distribution [254]. Anomalies can be expressed by scores or labels [256], and are often classified into three types: point anomalies, contextual anomalies, and collective anomalies contexts [255].

While anomaly detection techniques can be applied for all sorts of data sources, logs are of special importance for FCA applications, due to their almost pervasive and non-invasive nature, playing a vital role in case of a breach or incident analysis as they provide detailed information about activities. Nevertheless, the use of anomaly detection mechanisms using application and service log data for forensics and compliance auditing raises important challenges, due to factors such as the abundance of unstructured plain text contents and heterogeneous formats, redundant runtime information (which sometimes may change, as it is the case for certain IP addresses), and the existence of a significant amount of unbalanced data (a direct consequence of the prevalence of a normal operation mode). Moreover, with the increasing scale and complexity of distributed systems in the CI environment, monitoring, correlating and analysing logs is a time-consuming task that takes considerable effort, making it increasingly unfeasible to manually sort out trough evidence to detect anomalies.

Event correlation can be also categorized into different categories: temporal, spatial, or hybrid, whose combined use allows to capture both local (subsystem level) or global (IACS level) abnormalities [258]. After anomalies have been identified, is important to take forensic efforts in the analysis to determine the root causes and collect evidence, which will help to elaborate on the definition and application of countermeasures.

Some proposals have addressed the usage of log analysis as one of the input sources for anomaly detection. Chen and Li [321], for instance, proposed an improved version of an algorithm for detecting anomalies from audit data while updating the detection profile along with its execution.

Clustering techniques, such as the k-means algorithm, are often used by intrusion detection systems for classifying normal or anomalous events, having also application in the forensics analysis field. For instance, Asif-Iqbal et al. [322] correlated logs from different sources, supported by clustering techniques, to identify and remove unneeded logs. Syarif et al. [323] compared five different clustering algorithms and identified those providing the highest detection accuracy, also concluding that those algorithms were not mature enough for practical applications. Hoglund et al. [324], as well as Hajamydeen et al. [325], classified events in two different stages supported by the same clustering algorithm.

Münz et al. [326] applied the k-means clustering algorithm to feature datasets extracted from raw records, where training data are divided into clusters of time intervals for normal and anomalous traffic. Tian and Jianwen [327] improved traditional means clustering algorithm, to improve efficiency and accuracy when classifying data. Eslamnezhad and Varjani [328] proposed a detection algorithm to increase the quality of the clustering method based on a MinMax k-means algorithm, overcoming the low sensitivity to initial centers in the k-means algorithm. Ranjan and Sahoo [329] proposed a modified k-medoids clustering algorithm by presenting a new strategy to select the initial medoids, overcoming the means in anomaly intrusion detection and the dependency on initial centroids, number of clusters, and irrelevant clusters. Also, a k-nearest neighbor classifier for intrusion detection was explored by Liao and Vemuri [330].

Other authors adopted hybrid solutions for log analysis, combining the use of the k-means algorithm with other techniques for improving detection performance. They realized that despite the inherent complex structure and high computational cost, hybrid classifiers can contribute to improving accuracy. Mohammed et al. [331] proposed a clustering approach based on Fuzzy C-Means (FCM) and K-means algorithms to identify the evidential files and isolate the non-related files based on their metadata. Makanju et al. [332] took advantage of an integrated signature-based and anomaly-based approach to propose a framework based on frequent patterns. Varuna and Natesan [333] introduced a hybrid learning method integrating k-means clustering and Naive Bayes classification. Muda et al. [334] proposed k-means clustering and Naive Bayes classifiers in a hybrid learning approach by splitting instances into potential attacks and normal clusters.

Hybrid approaches have indeed proven to be quite interesting. However, in general, they still take a considerable amount of time to generate models for particular datasets, aggravated by the growth patterns normally associated with log sources in production systems. Elbasiony et al. [335] used data mining techniques to build a hybrid framework for identifying network misuse and detecting intrusions through the use of random forests algorithm to detect misuses, with k-means as the clustering algorithm for unsupervised anomaly detection. Fu et al. [257] presented an algorithm to convert free-form text messages in log files to log keys without heavily relying on application-specific knowledge. Du et al. [336] proposed the use of a Long Short-Term Memory (LSTM) to model a system to automatically learn log patterns from normal execution, and detect anomalies when log patterns deviate from the model trained from log data under normal execution. Henriques et al. [259] proposed an integrated scalable framework for efficiently detecting anomalous events on large amounts of unlabeled data logs through the use of clustering and classification methods supported by a parallel computing approach.

2.5.6 Summary

This section addressed the opportunities and challenges in the use of advanced analytics based on Big Data technologies, with AI and ML support, in the field of FCA. We surveyed the research in the field of advanced Big Data analytics taking into account the increased softwarization trend in terms of computing and network resource usage, as well as the benefits of leveraging advanced learning algorithms for improved automation. This has allowed to unveil a series of emerging development and evolution paths for FCA practices which are expected to have a profound change across the entire domain. Table 2.3 summarizes the relevant literature in Big Data for CIP.

Scope	**Works**
Automation in Forensics	Asquith and Horsman [313]
	Hayes and Kyobe [314]
	Homem [237]
	Brighi et al. [301]
	Verma et al. [316]
Big Data Properties	Demchenko et al. [288]
Big Data & AI challenges	Chehri et al. [251]
IoT Definition	Minerva et al. [244]
IoT Cyber-security and Privacy	Infineon et al. [250]
IoT-based Investigations	Stoyanova et al. [249]
IoT Forensics Analysis	Yaqoob et al. [252]
Big Data SCADA Historians	Kalakanti et al. [19]
Time Series Databases	Jensen et al. [297]
	Wang et al. [298]
Mapreduce for Big Data Analysis	Hegazy et al. [290]
Review on Cybersecurity Datasets for ML	Yavanoglu and Aydos [303]
ML Algorithms to Predict Zero-day Attacks	Usman et al. [304]
Network Forensics Classification Algorithms	Wiyono and Cahyani [305]
DDoS Forensics with ML Big Data Analytics	Hoon et al. [302]
Compliance Auditing Platforms	Ullah et al. [277]
	Doelitzscher [279]
	Bjørner and Jayaraman [280]

Table 2.3: Works related to Big Data-supported FCA

2.6 A FCA Taxonomy for CIP

To the best of our knowledge, there is no specific taxonomy in the domain of FCA for CIP in the surveyed literature. To fill this gap, we devised a taxonomy covering the scopes as well as the functional and non-functional dimensions of the FCA practice, inspired by forensic investigation and compliance practices. The proposed taxonomy is depicted in Figure 2.1, being organised along seven major dimensions, inspired by the methodology proposed by [337]. These are the following:

- **Critical Infrastructures:** this dimension characterises the scope and environment to be protected, including SCADA and IACS core systems. Moreover, specific attacks targeting CIs, SIEM, and other security platforms and systems providing protection capabilities are also considered

- **Governance:** gathers the orientations that can support the decisions in the application of FCA processes. It comprises the investigation processes, guidelines, agencies, standards and regulations, training, directives, and existing specific security frameworks.

- **Preparedness:** this dimension comprises the proactive aspects that may be considered to safeguard, support and prepare in advance the execution of FCA processes. It encompasses readiness, forensic by design, forensic frameworks, anti-forensics, and auditing frameworks.

- **Data Acquisition:** this dimension deals with the challenges of gathering digital and network forensics covering aspects such as volume, live forensics and data provenance, while safeguarding the need to protect information about evidence.

- **Evidence Identification:** covers the models, algorithms and approaches helping to identify evidence and non-compliant events. It comprises IDS, detection techniques, causality, and learning, in this last case by using approaches supported by clustering and hybrid approaches algorithms.

- **Reporting:** this dimension covers communication and interoperability-related aspects, encompassing topics such as privacy concerns, visualization and searching, interoperability, and Chain of Custody.

- **Deployment:** this encompasses non-functional aspects, which relate to platform and infrastructure-related aspects, such as cloud computing, virtualization support, scalability, automation, and quality.

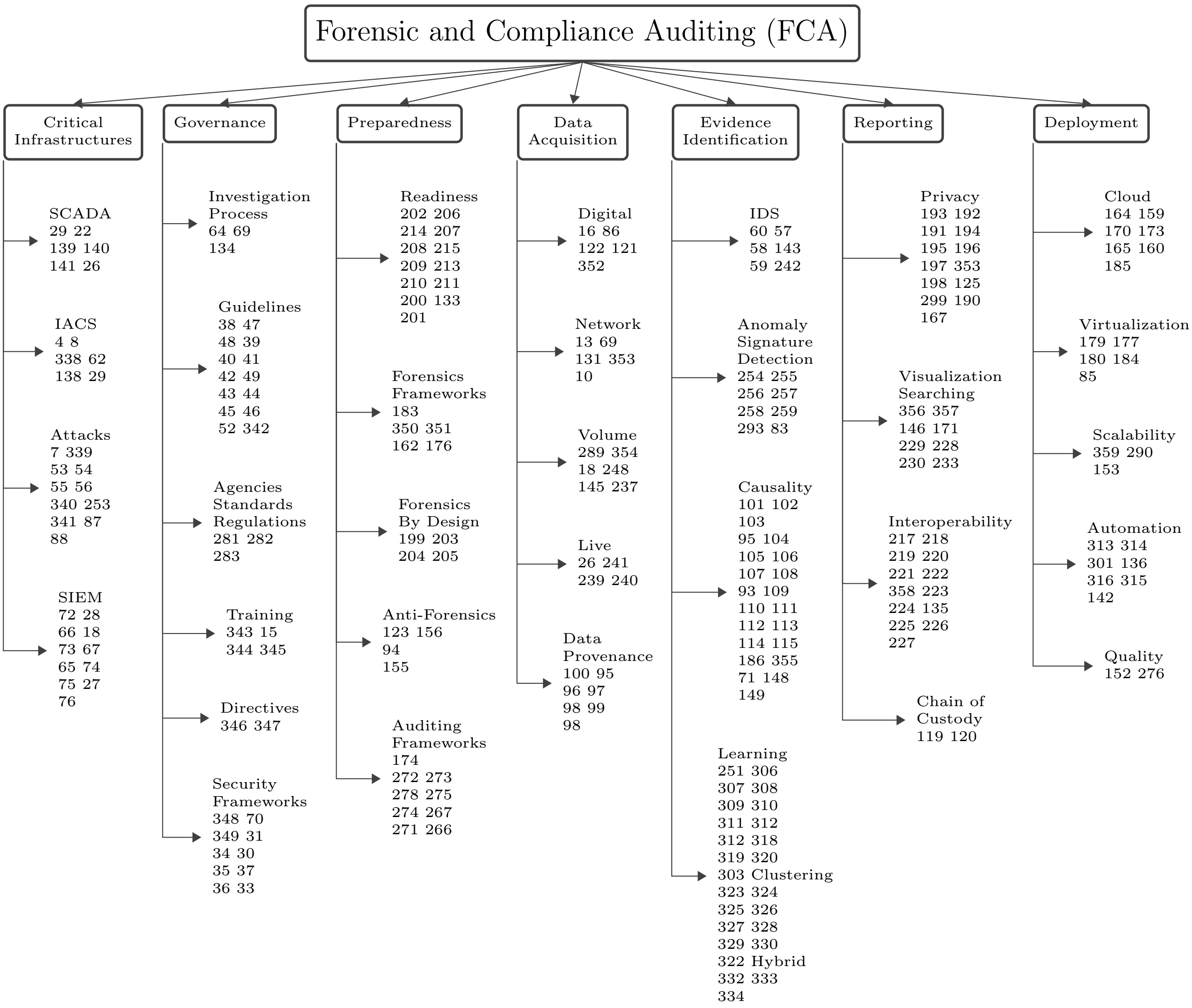

Forensic and Compliance Auditing (FCA)
Critical Infrastructures
SCADA 29 22 139 140 141 26
IACS 4 8 338 62 138 29
Attacks 7 339 53 54 55 56 340 253 341 87 88
SIEM 72 28 66 18 73 67 65 74 75 27 76
Governance
Investigation Process 64 69 134
Guidelines 38 47 48 39 40 41 42 49 43 44 45 46 52 342
Agencies Standards Regulations 281 282 283
Training 343 15 344 345
Directives 346 347
Security Frameworks 348 70 349 31 34 30 35 37 36 33
Preparedness
Readiness 202 206 214 207 208 215 209 213 210 211 200 133 201
Forensics Frameworks 183 350 351 162 176
Forensics By Design 199 203 204 205
Anti-Forensics 123 156 94 155
Auditing Frameworks 174 272 273 278 275 274 267 271 266
Data Acquisition
Digital 16 86 122 121 352
Network 13 69 131 353 10
Volume 289 354 18 248 145 237
Live 26 241 239 240
Data Provenance 100 95 96 97 98 99 98
Evidence Identification
IDS 60 57 58 143 59 242
Anomaly Signature Detection 254 255 256 257 258 259 293 83
Causality 101 102 103 95 104 105 106 107 108 93 109 110 111 112 113 114 115 186 355 71 148 149
Learning 251 306 307 308 309 310 311 312 312 318 319 320 303 Clustering 323 324 325 326 327 328 329 330 322 Hybrid 332 333 334
Reporting
Privacy 193 192 191 194 195 196 197 353 198 125 299 190 167
Visualization Searching 356 357 146 171 229 228 230 233
Interoperability 217 218 219 220 221 222 358 223 224 135 225 226 227
Chain of Custody 119 120
Deployment
Cloud 164 159 170 173 165 160 185
Virtualization 179 177 180 184 85
Scalability 359 290 153
Automation 313 314 301 136 316 315 142
Quality 152 276

Figure 2.1: Proposed FCA Taxonomy for CIP

This taxonomy aims at presenting FCA-related topics in a convenient way, using a set of criteria covering both functional and non-functional aspects while striving to provide a convenient organization for the most significant developments.

2.7 A Reference Architecture for FCA Systems

As already mentioned, even though Forensics and Compliance Auditing are different activities, both in terms of purpose and expected outcomes, there is a considerable amount of proximity between them, since they often resort to the same data sources and similar information and context extraction techniques to gather and process evidence. This hints at the possibility of building both capabilities on top of a shared reference architecture, providing data acquisition, transport and processing pipelines, as well as persistence capabilities.

In this section, we provide such a reference architecture, in order to better identify the various functional blocks typically found in FCA systems. It should be noted that this is an abstract architecture. Real-world FCA tools will usually map into subsets of this architecture.

The main functional requirements to be met by FCA solutions include identifying, extracting, preserving and presenting digital evidence. Table 2.4 highlights how the functional blocks typically required for Forensics operations and for Compliance Auditing activities largely overlap.

Functional Block	Forensics	Compliance Auditing
Data Lake	Yes	Yes
Analytics	Yes	Maybe
Business Policies & Rules	No	Yes
Real Time Search	Yes	Yes
Monitoring	No	Yes
Ingesting	Yes	Yes
Orchestration	Yes	Yes
Visualization & Dashboards	Yes	Yes
Platform as a Service	Yes	Yes
Security	Yes	Yes
Cloud-native	Yes	Yes
Scalability	Yes	Yes

Table 2.4: Functional Blocks vs. Forensics and Compliance Auditing

Figure 2.2 presents proposed reference architecture. The first stage of FCA systems includes the collection of heterogeneous data from internal and external sources to be gathered into a single logical store. That data can include a vast amount of structured and unstructured heterogeneous data from a large number of sources widely dispersed across the CI, including those from the associated IACS and the ICT infrastructures.

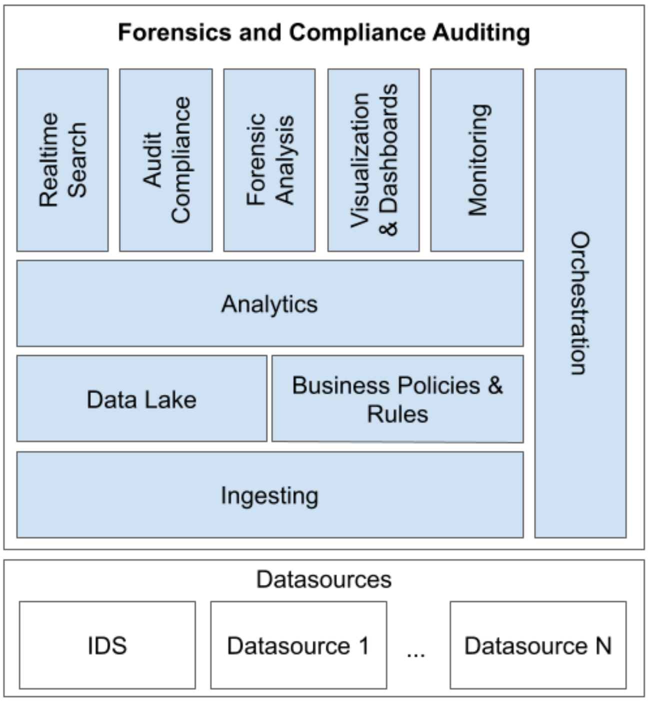


Figure 2.2: Reference Architecture for FCA Systems

The second stage incorporates forensic analysis and third-party continuous auditing capabilities for the identification of *post mortem* security events, foreseeing, tracking, and tracing possible anomalies. Such objectives can be achieved by correlating the features retrieved from a seemingly disparate class of events that usually are not considered in terms of CI. Thus, beyond the forensics activities, the auditing layer checks compliance with standards, policies and rules. An example of such verifications is the cross-check of past system logs with the registration of physical access to remote facilities, to indirectly detect unauthorised accesses.

Next, we discuss the key components of this architecture.

2.7.1 Data Sources & Data Ingestion

The Ingesting Module acts as a set of probes capturing data from a large number of heterogeneous data sources from the surrounding environment, including applications such as Authentication Authorization and Accounting (AAA), ICT security logs (e.g., anti-virus, IDSs), internal personnel activities, physical access control logs (door switches and surveillance cameras), maintenance activities (physical and logical systems), interactions with third-parties (e.g., general documents, emails) and incident logs (e.g. ICT trouble tickets).

Integration of third-party sources within the Ingesting Module is usually accomplished by using custom data adapter components. Such modules ingest data from IDSs, third-party applications or triggered alerts from monitoring processes, also including trust and reputation data, all of it being integrated using pull or push-based approaches.

The ultimate goal of the Ingesting Module is to acquire, parse, enrich and normalize incoming data (which may be structured or unstructured, depending on its nature and sources) into a common format suitable to be stored in the Data Lake (DL) and later used for analysis purposes, while ensuring consistent timestamp synchronization across several sources in order not to compromise event timelines. This means that incoming raw data needs to be handled in a streamlined way, in order to optimize its transport, storage and processing, thus implying the deployment of data processing pipelines akin to Extract, Transform and Load (ETL) workflows.

These Ingesting Module workflows, which may also include filtering, normalization, indexing, enrichment, and aggregation steps, must be capable of dealing with high volumes of heteregeneous data later to be fed into the DL, which constitutes the central repository component in the reference architecture. Persisted data from different sources (including enriched data) may be used for several purposes, such as training learning models or to feed visualization tools helping to identify threats.

2.7.2 Data Lake

The DL provides a repository to store data in different formats. This repository centralizes logs and other different sorts of data collected by the Ingesting Module (IM), to be made avaliable to FCA activities (and possibly other applications). Aditionally, the DL also persists the correlation and/or classification results of such data feeds, helping streamline higher-dimensional analytic procedures.

The DL often assumes a distributed nature, to horizontally scale in order to fit increasing volumes of data and/or to increase the performance of data searching and correlating activities. It usually provides the automation capabilities to manage how indexes and queries are distributed across the cluster to accommodate large amounts of data and transactions, including support for automated scaling. This is important since high availability, resiliency, throughput, and low latency when querying large volumes of data are important non-functional requirements for the DL.

The DL may also provide integration mechanisms to plug-in common authentication systems such as Active Directory, Lightweight Directory Access Protocol

(LDAP), and Security Assertion Markup Language (SAML).

2.7.3 Analytics

After the data is captured and stored in the DL, the Analytics Module takes the responsibility for extracting relevant insights. Supported by state-of-the-art analytic methods, this module provides the capabilities to classify threats with potential impact on the systems' integrity, confidentiality, or availability. It starts by individually identifying unusual behaviors in past events, logged in computers or networks, correlating them in order to identify the compromised systems from the chain of events. For instance, this can be used to correlate the sequence of past executed shell commands with the list of files that have changed, to discover threats. The outcomes of this component also provide an important input to trigger automated rapid response actions.

Within the Analytics Module, the use of ML techniques can help discover new behaviors and patterns to define and/or reveal the policies and business rules used to classify threats, from a vast amount and variety of data. Thus, it is expected that taking such a proactive approach to classify events in advance (before the forensic investigation has even started) may contribute to improve the readiness of forensic and compliance auditing processes. This is further reinforced by the fact that the resulting classified data will also be stored in the DL as input for further forensic analysis processes.

The nature of its role requires Analytics Module to be flexible, allowing models to be updated "on the fly" between retraining, but also to offer a good performance/efficiency balance. The latter can be achieved by decoupling the training and classification processes and running in parallel, thus reducing the time devoted to event classification while increasing the chances of automatically recognizing new threats. Improving the time spent on training can also be achieved by dividing the dataset and even the model, assigning parts to different processes. Thus, even when the training model is too large, it can be trained in the background without disturbing the live system.

Taking advantage of its scale-out properties, the reference Analytics Module architecture is designed to simultaneously train and run different models. Some of them can be used for training, while other ones can be used for classification purposes. Update or introduction of models into production after training should follow best practices, eventually pursuing a MLOps-like lifecycle management approach.

2.7.4 Forensic Analysis

Forensic analysis is a key step in the investigation process to identify the traces of malicious activity and extract evidence. Additionally, this may also encompass the establishment of a causality path between classified anomalies, oriented towards identifying the root cause and progression path of an incident.

Forensic analysis capabilities can be leveraged by using ML models in the context of the Analytics Module. These can help forensic investigators efficiently find out the relevant events from large amounts of data, coming from diversified sources. Technically, evidence can be collected with queries entailing a set of rules to be run against the events previously stored in the DL.

The adoption of a common standardized forensic schema assumes particular importance in collecting and exchanging relevant information or evidence between different entities and even jurisdictions, along the investigation chain. To ensure that evidence is legally admissible while safeguarding authenticity and integrity, schemas may adopt techniques such as cryptographic hashing.

2.7.5 Audit Compliance

The audit compliance component provides the capability to assess conformity with standard practices and defined policies, as part of an ongoing CIP strategy. Such standards may encompass regulatory requirements and/or industry guidelines that the infrastructure operator must comply with for certification, security and/or safety reasons. In case an audit trail is available, an expert can return to the source material to check the quality of the analysis and processing.

Beyond the policies resulting from the need to comply with regulatory or standardization frameworks, organizations can establish custom rules based on their own internal processes and procedures, such as corporate laws, plans, and procedures.

The Audit Compliance module takes business rules and regulatory policies to identify violations and trace the path of non-conforming events. This process assesses the compliance of the facts denoted by the ongoing events with the defined business rules and policies, providing an outcome that includes scores computed by quantifying the aspects regarding security and the level of risk. Both the Forensic Analysis and the Audit Compliance modules leverage the outcomes from correlating data at the Analytic component.

2.7.6 Visualization and Dashboards

Visualization capabilities are key for forensics activities, providing the means to display information in a manner that may evince the presence of suspicious or anomalous patterns. Such capabilities can be key to help understand and analyze specific domain datasets by applying histograms, scatter and box plots, tree maps, surface pots, parallel coordinate plots, and radar charts [356, 357].

This module is fed by the data persisted in the DL repository, which is used for analysis purposes. In a typical arrangement, dashboard panels are used to highlight a variety of indicators which may be directly generated from agent feeds, or as the result of enrichment (providing contextual information), aggregation or analytics/analysis sources. For instance, panels may provide information about the total number of received events, their variety, or a histogram depicting when events were received, just to name a few. Moreover, this data may be exported for integration with third-party tools.

Visualization and Dashboards provide operators with suitable graphical tools to explore and analyze contextual information – such tools must provide querying and summarization capabilities adequate for dealing with large volumes of data in repositories, computing metrics and applying specific functions against some attributes.

2.7.7 Business Policies and Rules

Beyond the mandatory regulatory, legal and standardization frameworks, organizations often define specific procedural or workflow rules based on their own internal processes and needs, based on corporate laws, plans or roadmaps.

A repository of CI business policies and rules may be used to support organizational-wide compliance assessment. If those events trigger some of the rules describing policies, then the associated alerts will also be triggered. Such rules can be tuned according to specific thresholds and can help prioritizing and score events. For example, a company policy may impose constraints on their employees on the use of resources, thus, any login attempt violating this rule should be reported. Physical access control is another example: alerts can be triggered when the doors in a given department or physical installation are opened out of the authorized period. Formally, those CI Business rules will assess the compliance of processes accordingly to the business norms.

2.7.8 Monitoring

A Monitoring component provides the capabilities to look at things as they happen, helping operators to identify anomalies from data. It can either trigger alerts or highlight information resulting from such a continuous assessment, matching CI audit compliance rules against persisted events in DL. Moreover, it will also check the level of trust and reputation risks to classify eventual threats and trigger alerts to the operators.

Such a Monitoring component may also offer the ability to set up automatic response rulebooks or human-supported actions, as well as triggering alerts and notifications, providing information to help the operator become an effective link of a human-in-the-loop decision chain. Necessarily, models and rules used for alerting purposes must be fine tuned to provide adequate accuracy and low false positive rates.

2.7.9 Real-Time Search

A Real-Time Search component provides high-performance query capabilities from large amounts of stored data in the DL to support the extraction of relevant FCA information. The component is able to run queries against the indexed data in the DL. Because every second counts when looking up for quick responses, the process for indexing data can be executed in advance to improve the query performance. This component also includes an interface to integrate third-party components to run lookup actions for data.

2.7.10 Orchestration

The Orchestration component offers capabilities for managing and coordinating the different FCA components. Such capabilities comprise automation, self-healing, and service discovery. For cloud-native implementations, the use of containerized services integrated within a microservice architecture may improve scalability, eventually allowing for the deployment of multiple instances of the same architectural functional block to scale capabilities, as needed.

A set of high-performance services requires a management and maintenance subsystem to coordinate the configuration settings supporting distributed FCA service synchronization and operation. While the capabilities of this framework are provided independently by different modules and components and made available to third parties through the use of an Application Programming Interface (API), the overall system can be depicted as being akin to an atomic structure. This paves the way to its possible provisioning and deployment in the form of a SaaS model.

2.7.11 A Cloud-native Platform as a Service

The implementation of a framework designed according to the reference architecture hereby described can also benefit from adopting a Cloud-native architecture in which features are decoupled in microservices designed to improve scale-out capabilities, eventually hosted in containers. Taking this approach makes it possible to have wrap-up FCA solutions supporting a large number of customers.

Providing independent external interfaces to the available functions of the cloud hosting or orchestration platform can provide instrumentation mechanisms for third parties, allowing them to tailor and deploy custom scenarios according to their needs. This approach increases the opportunity to integrate custom third-party solutions with the FCA, resorting to APIs or queuing mechanisms to enable effective integration between third-party applications and the FCA reference architecture. Such capabilities can enable the integration of specific policies and business rules, also providing the means to customize data sources, disable some components or extend their core capabilities, among other options. Moreover, this makes it possible, for instance, to customize solutions to integrate this reference architecture with solutions such as SIEM, SOAR, EDR and XDR.

2.7.12 Platform Security

Incorporating security in the FCA reference architecture allows to develop, deploy and operate each component safely, following the best practices in the field. It is also important to protect communication channels, providing secure inter-module integration. For this purpose, the adoption of Zero trust principles [360] may be a key design feature – while those principles are primarily focused on data and service protection, they can and should be expanded to include all enterprise assets and subjects.

Complementary to active protection characteristics, Authentication and Authorization mechanisms should also be properly implemented and continuously assessed to check compliance with the defined access rules.

2.7.13 Summary

This section proposed a reference architecture for FCA and its functional building blocks, according to the identified requirements, also detailing roles and interactions. Moreover, non-functional aspects comprising the implementation of Cloud-native Architecture, Platform as a Service (PaaS), and Platform Security were also addressed, in order to demonstrate the plasticity of the proposed concept in terms of deployment and operational options.

2.8 Discussion and Open Issues

This section discusses the findings of this survey and highlights the open issues and the research opportunities to be considered in the topic of FCA in the scope of CIP.

2.8.1 Discussion

When it comes to CIP, most literature references are focused on conventional cybersecurity prevention, detection and mitigation techniques. However, and given the considerable overlap of functionalities associated with security, forensics and compliance audit contexts, it makes sense to consider some proposals and technologies as candidates for application in FCA contexts.

In fact, the lack of proper FCA capabilities within CIs may not be attributed to any sort of technological obstacles, but rather to a chronic lack of readiness. For instance, this can lead to situations where forensics procedures are undertaken on an *as needed* basis, long after incidents have occurred, in an offline basis. This can restrict the forensics process in a decisive way, hampering the establishment of a clear perspective about incidents, their root causes and implications.

Moreover, and on the compliance auditing side, there is an ongoing trend requiring CI operators to comply with a growing body of standards and regulations while, at the same time, having to keep up with increasingly complex and interconnected infrastructures with a proliferation of control, sensory and endpoint devices.

The implementation of adequate FCA mechanisms can assist in the prevention as well as in the mitigation of the potential consequences of incidents or adverse events, improving the CI resiliency. In fact, it is worth noticing that forensically reconstructing past events and highlighting disrupted compliance events in CI environments can make it possible to discover potential vulnerable vectors and hidden threats whose correction can be decisive to avoid future consequences.

When it comes to FCA, proactivity is key. But operators need to understand the added value of adding such capabilities before committing to invest to adapt infrastructures, for instance to deploy and customize adaptor agents to extract the significant amounts of data living in silos (e.g. ICT systems surrounding the CI environment) into a single homogeneous coherent dataset, whose existence can help overcome the complexity arising from the use of a large number of forensic tools, protocols, and standards.

With proper collection mechanisms in place, it becomes possible to correlate data by applying models, algorithms, architectures, and solutions to effectively classify and predict behaviors and extract evidence from large amounts of data or automatically support data-driven decision-making. Moreover, results from correlation can also help enforce auditing compliance on security policies, regulations, recommendations, applicable laws, and standards processes to increase the security and trust in CIs that may help to prevent future security incidents.

Also, FCA needs to keep up with times and adapt, as the trend towards resource consolidation also reaches CIs, with the adoption of virtualization technologies within private, public or hybrid clouds. For instance, while the adoption of a cloud-native setup with containers can bring significant challenges in terms of forensics integration, it can also provide net benefits in terms of management, monitoring, and control of FCA frameworks for CIP, providing elasticity to accommodate transient requirements from analysis processes.

Another significant trend with impact in FCA processes is the emergence of IIoT and Big Data, which tend to go hand-in-hand in modern CIs, due to the considerable data handling requirements for massively distributed infrastructures. However, while such developments pose challenges to FCA solutions, it should also be noted that Big Data technologies also provide a technological basis enabling the development of sophisticated forensic data and evidence transport, processing and storage mechanisms that can take advantage of the elasticity of virtualization and cloud technologies.

All the aforementioned aspects have been considered to devise a comprehensive and easy-to-deploy FCA framework template which was designed to be neutral from a deployment standpoint and decoupled from the end-user infrastructure to be protected. This reference design gathers the capabilities to collect and continuously monitor and correlate data from diversified data sources, being able to support decision-makers and forensics practitioners alike, also enabling the definition of responsive actions from large amounts of data. This approach can help track past events to perform evidence extraction and incident root cause analysis, also allowing to detect non-compliant events in near real-time, for example, from logs collected before, during, and after incidents.

2.8.2 Open Issues

This survey also identified a series of open issues and research gaps in terms of FCA capabilities for CIP. Probably one of the most important findings of this survey has to do with realising that, in most cases, existing security tools are missing the integration means for a full-stack FCA solution. This is due to the fact that many of these tools are not embracing open standards on maintaining an effective chain of custody or plug-and-play capabilities to increase their interoperability and reduce the need for collaborative work between tool owners and end-users. Also, many of these tools lack flexible FCA capabilities, decoupled from the applications they aim to protect (e.g. applied to 5G vertical applications taking advantage of cloud-native approaches).

Other identified handicaps that equally affect SIEMs and forensics tools for CIP include: the absence of custom connectors and parsers for data source integration, incomplete data, lack of basic correlation rules, elemental storage capabilities, reliance on manual operation, basic reaction and reporting capabilities, limited data visualization, or deployment, and management complexity [65]. Other missing aspects comprise the lack of GDPR privacy compliance [361], as well as the absence of high-level security risk metrics. Also regarding metrics, there are no well-defined KPIs for FCA tools, for example to assess the Quality of Service (QoS) and Quality of Experience (QoE), reliability, availability, and resiliency.

Also, the availability of open standards, languages, and data abstractions for sharing and exchanging evidence are key to enhance FCA tools and improve their interoperability while enhancing the processes devoted to discovering forensic evidence in an automated, effective, and efficient manner. That includes, for example, the adoption of open standards for sharing evidence and keeping an effective chain of custody.

With the emergence of IIoT scenarios, the requirements to capture, transport and process of large volumes of data become more demanding. Thus, the lack of adequate computational and storage resources may impose limits on the application of FCA methodologies for gathering and analyzing data. Overcoming them is instrumental to achieve a near real-time data correlation latency from multiple physical sources, also enabling the deployment of effective alerting mechanisms for non-compliance incidents. Equally important is the lack of automated and dynamic orchestration capabilities, adaptation systems, and tools supporting FCA activities and managing their entire life cycle, which are key for implementing efficient and resource-effective FCA capabilities.

Another key concern in FCA activities is their eventual impact on performance and efficiency on systems being secured, such as in the case of collecting large amounts of data for forensic purposes and preserving data privacy [316]. For instance, in systems with specific determinism and real-time requirements, special care must be taken to avoid imposing any kind of undesirable overhead or creating potential points of failure.

2.9 Summary

This work highlights the importance of considering both forensics and compliance auditing (FCA) as high-priority topics for CIP, contributing with guidance in the design and implementation of security processes by considering policies, standards, guidelines and procedures and evidence analysis techniques. For this purpose, we surveyed the latest developments, methodologies, challenges, and solutions addressing FCA in the scope of CIP, focusing on contributions capable of tackling the requirements imposed by massively distributed and complex IACS which handle large volumes of heterogeneous, noisy, redundant and even ambiguous data, for analytic purposes.

We started by highlighting the need for addressing modern security challenges and requirements to improve the security of CI by considering FCA capabilities.

With that in mind, a survey of the the relevant literature was undertaken, focused on the intertwined topics that may stress the benefits and value brought by FCA approaches. From this survey it was also noticed the lack of specific FCA approaches and taxonomies for CIP.

The surveyed literature resulted in a taxonomy gathering the major identified categories, such as CIs governance, preparedness, data acquisition, evidence identification, reporting, and data. Together with the lessons learned from the literature analysis, this taxonomy was instrumental to identify the most relevant FCA capabilities, resulting in the identification of a series of key functional blocks later organized as part of a reference FCA architecture template, designed to provide a strong foundation to support the implementation of future solutions.

Chapter 3

A Framework for Forensics and Compliance Auditing in Critical Infrastructures

Contents

In this chapter we introduce the proposed Forensics and Compliance Auditing (FCA) framework, that was designed to provide explicit support for forensics and audit compliance activities in the specific scope of Critical Infrastructure Protection (CIP). It constitutes a kind of domain-specific augmented flight recorder that, besides persisting relevant information, also incorporates compliance auditing capabilities addressing Critical Infrastructure (CI) security policies and forensics capabilities for post-mortem event analysis, with the ability to scale by making use of a cloud-native approach. The architecture of the proposed framework enables an unified security solution in the context of distributed Industrial Automation and Control Systems (IACS), with tools that are easy to deploy, use, modify and extend. This approach allows to collect forensic evidence and ensures that future unforeseen incidents are avoided or identified to leverage classification capabilities throughout the assessment of deviations to reference behaviors. This will contribute to prevent risks from operational errors and cyber-attacks. Compliance auditing on the CI security policies also contributes to eventually minimize future security incidents. This allows to leverage the outcomes of forensic research in the benefit of auditing compliance, in which forensic tools can be reused to assess regulatory compliance.

The chapter is organized as follows. Section 3.1 describes the proposed FCA. Section 3.2 presents the implemented proof-of-concept. Section 3.3 presents an experimental performance evaluation. Finally, Section 3.4 concludes the chapter.

It should be noted that the content of this chapter has already been published, to a large extent, in [362]. Moreover, it should also be noted that all data and function symbols used in the next subsections are listed in Table 3.5, at the end of this chapter.

3.1 Proposed FCA Framework

In this section we present the proposed FCA framework. First, we describe the framework architecture. Next, we specifically discuss each key component of the framework: Data Acquisition; Domain Processor; Cyber-physical IDS; Data Lake; CI Business Rules; Monitoring; Data Visualization; Analytics; Trust and Reputation Indicators; Orchestration; Actors and Roles.

Figure 3.1 shows the architecture of the proposed platform. In practice, this framework works like a domain-specific augmented flight recorder that complements typical security solutions by providing audit compliance and forensic tools. It supports forensics activities for identifying, extracting, preserving, and highlighting digital evidence. Regarding compliance auditing, it enables the assessment of compliance with standards, business rules, and policies.

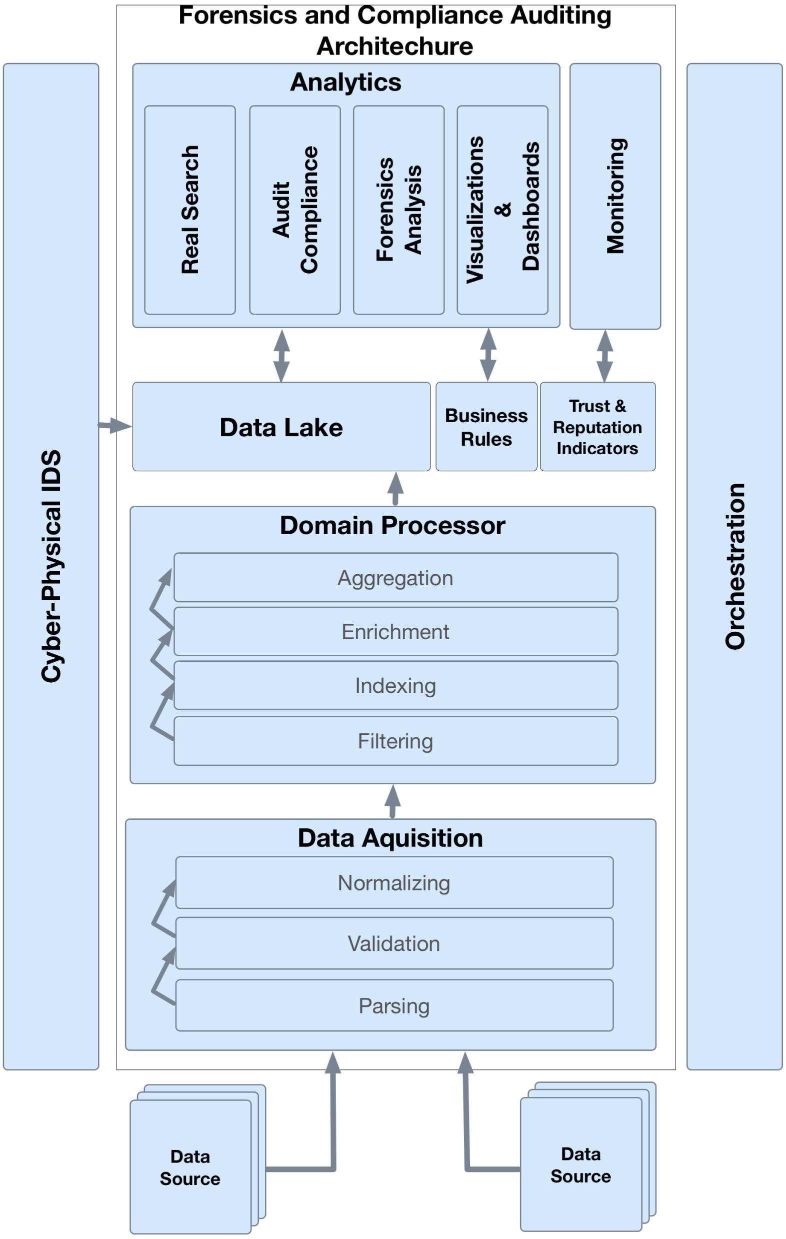


Figure 3.1: Architecture of the Proposed FCA Framework

The framework is able to dynamically scale horizontally, in order to adapt to different deployment scenarios and to different loads over time, while keeping the ability to collect, store and correlate large volumes of data from a large number of disparate and geographically dispersed sources.

The platform collects data from multiple sources, gathered into an unified view to enhance FCA capabilities such as digital evidence extraction. Data sources include, for instance, service and device logs, physical process logs, network traces, and Authentication Authorization and Accounting (AAA) sessions or physical access control systems, both for forensics and compliance auditing purposes. It is able to cope with high-throughput flows of structured and unstructured heterogeneous data, acquired from both internal (e.g. IACS) and external sources (e.g. corporate Information and Communication Technologies (ICT) systems) – therefore enabling a broader perspective.

The Analytics component systematically correlates collected data, to detect anomalies and non-compliance events. Those findings will help highlighting the relevant facts and improve the post-mortem identification of security events to be extracted as evidence. The adoption of a continuous auditing approach, against a set of predefined policies, helps detect and mitigate possible threats.

The FCA works in a hybrid fashion: cyber-physical system anomalies are handled by the corresponding domain-specific IDS, which by their turn feed the FCA with the analysis outcomes. For Operational Technology (OT) systems above IEC62443 [283] layer 2 (comprising Area Supervisory Control components such as Supervisory Acquisition and Data Control (SCADA) servers or historian databases) and IT systems, the FCA is able to receive direct event feeds (from probes and log sources), later to be normalised and preprocessed. Anomaly detection in the scope of the FCA framework takes advantage of both the post-processed CPS IDS data as well as from the OT and IT feeds.

Next, we discuss each individual module of the FCA framework.

3.1.1 Data Acquisition

The Data Acquisition Module (DAM) is able to ingest a large amount of data, for instance from application logs, AAA services, ICT security tools (e.g., anti-virus), internal personnel activities, physical access control logs (door switches and surveillance cameras), maintenance activities (physical and logical systems), interactions with third-parties (e.g., general documents, emails) and incident logs (e.g. ICT trouble tickets). This module can eventually be decoupled across separate gateway/edge instances to optimise load distribution.

The integration of sources from third parties is implemented by customizing a data adapter provided by an interface. This way, the DAM manages the incoming data from heterogeneous sources $S_j^{(\Gamma)} \in S$, with their own schemas Γ. This results in discrete events $E^{(S_j)}$, collected from probes P_{E_i} assigned to sources S_i. Those events contain a set of independent attributes or features, where i is the i^{th} event and k denotes the total number of features in a discrete event $E_i^{(S_j)}$.

The incoming flow of raw data goes through a pipeline producing events described relying in a common schema – and later persisted into a Data Lake (DL) for further analysis. The set of tuples $(a,v)_k \in \{1..n\}$ are composing parts of events E_i, as denoted by:

$$E_i = \bigcup_{i=1}^{n} (a,v)_k \tag{3.1}$$

A tuple includes n attributes a and their corresponding values v, and is addressed by an index or key k, according to:

$$\{(a,v)\}_{k\in\{1..n\}} \in E_i^{(S_j)}, a \in A, v \in T \tag{3.2}$$

A value v of any type $t \in T$, scalar or not, is assigned to an attribute $a \in A$, where A represents the set of all existing attributes. Common attributes include, for example, the provenance and time when the events were created.

This module also provides an interface to add new data sources S_{i+1}. A parsing function $P()$ converts the incoming data events E_i into a discrete event N_i, according to:

$$P : E_i \rightarrow N_i \tag{3.3}$$

The adoption of a common schema for normalized events N_i facilitates the integration processes between the internal components and external third party applications. This schema can be either followed or extended by third parties.

The DAM takes the role on managing the data ingestion $I()$, processing the received events E_i from the different sources S_j to be parsed $P()$, validated $V()$ and normalized $N()$, according to:

$$I = [N \circ V \circ P]() \tag{3.4}$$

This ingestion function $I()$ of a given event E_i results in a normalized event $N_i^{(1)}$, such as:

$$N_i^{(1)} = I(E_i) \tag{3.5}$$

The parsing function $P()$ extracts the features as attributes and values in a set of tuples $(a,v)_k)$. These attributes and values come from a raw event $E_i^{(S_j)}$, received from system S_j, and are transformed into an event $E_i^{(\Gamma_{S_i})}$, following the native schema of the corresponding data source Γ_{S_i}:

$$P : E_i^{(S_j)} \rightarrow E_i^{(\Gamma_{S_i})} \tag{3.6}$$

The validation function $V()$ checks whether an event $E_i^{(\Gamma_{S_i})}$ follows the schema Γ_{S_i} associated with its source system S_i.

$$v_i = V(E_i^{(\Gamma_{S_i})}, L), v_i \in \{0,1\} \tag{3.7}$$

In case this check fails, that event is discarded and recorded as a validation failure V, for further analysis, as denoted by:

$$V = \bigcup_{i=1}^{N} v_i \tag{3.8}$$

The normalization function $N()$ maps an event $E_i^{(\Gamma_{S_i})}$ in schema Γ_{S_i} to an event $N_i^{(\Gamma_{S_i})}$ following the general schema Γ of the framework, according to:

$$N : E_i^{(\Gamma_{S_i})} \rightarrow N_i^{(\Gamma)} \tag{3.9}$$

Raw events E_i are subject to a set of functions, including parsing $R()$, validation $V()$ and normalization $N()$ into a common format $N_i^{(1)}$. Finally, the events collected at the first stage $N_i^{(1)}$ are collected as part of the set of normalized events N_{DA} in DAM, according to:

$$N_{DA} = \bigcup_{i=1}^{N} N_i^{(1)} \tag{3.10}$$

To keep overall performance, probes receiving data are decoupled from the rest of the components in the framework. Decoupling components from data sources makes it easier to support the assigned workloads, even if the data sources are delivered at a high pace (e.g. Packet Capture Process (PCAP) or data logs from a large number of endpoints). To that aim, the incoming events are stacked in a queue Q, waiting for the availability of component C_i.

The performance and availability of the framework is impacted by overloaded components. Decoupling the intertwined asynchronous communication between the core functions can be achieved by queuing the underlying events $E_i^{(\Gamma)}$. This offers the opportunity to shift in time the future workloads. This queue function $Q()$ interfaces the coupled components according to:

$$Q_{C_{i-1},C_i} = Q(C_{i-1},C_i), E_i^{(\Gamma)} \in Q_{C_{i-1},C_i} \tag{3.11}$$

Since different components along the ingestion stream along may have different ingestion rates, queuing becomes necessary.

The rate of received data Δ results from counting events ξ for a given period of time Δt, according to:

$$\xi_{\Delta t} = \frac{(\xi_{T_1} - \xi_{T_0})}{\Delta t} \tag{3.12}$$

The output rate $\theta_{\Delta t}$ represents the difference between ingested (ξ) and consumed rates (θ) in a certain period of time Δt, yielding:

$$\theta_{\Delta t} = \frac{(\theta_{T_1} - \theta_{T_0})}{\Delta t} \tag{3.13}$$

3.1.2 Domain Processor

The Domain Processor Module (DPM) transforms and loads ingested events. This module gathers the functions for filtering $F()$, indexing $I()$, enriching $E()$, and aggregating $A()$. At this stage, all the events' metadata can be extracted into a keyword index for later support of queries. Therefore, the second stage of the normalization process over events $N_i^{(1)}$ to $N_i^{(2)}$ takes place in this module. The domain processor module may be decoupled across distributed gateway/edge instances to optimise load distribution, either as a separate entity, or consolidated together with Data Acquisition module instances.

As previously stated, this module receives a flow of normalized events $N_i^{(1)}$, resulted from the pipeline (as a sequence of stages) provided by the DAM. Thus, the processing function $R()$ applies the sequence of the above functions $\{A,E,I,F\}$, as denoted by:

$$R = [A \circ E \circ I \circ F]() \tag{3.14}$$

The processing function $R()$ receives as input events of the first stage $N_i^{(1)}$ to output the second stage events $N_i^{(2)}$, according to:

$$R : N_i^{(1)} \rightarrow N_i^{(2)} \tag{3.15}$$

For the sake of simplicity, from now on N_i will denote the result of the second normalization process $N_i^{(2)}$.

Next, we discuss each of the four functions that compose the DPM.

Filtering provides filtering capabilities $F()$, supported by rules R_i, to check if events $E_i^{(j)}$ in domain $\{0,1\}$ should be accepted to be forwarded to the next component, as denoted by:

$$F : E_i^{(j)} \rightarrow \{0,1\} \tag{3.16}$$

A filter F denotes a logical disjunction of n Boolean rules R_i, according to:

$$F = R_1 \vee R_2 \vee \cdots \vee R_n = \bigvee_{i=1}^{n} R_i \tag{3.17}$$

A rule R_i may either denote a conjunction of positive or negative disjunctions of specific attribute levels, as defined by:

$$R_i = \bigwedge_{a_k \in A_{R_i}} S_k \tag{3.18}$$

$A_{R_i} \subset A$, denotes a subset of attributes in rule R_i. $\bigvee$ and $\bigwedge$ represent the Boolean algebra operators OR and AND. The S_k refers to a logical statement about the k^{th} attribute $a_k \in A_{R_i}$. Thus, S_k is composed by two distinct parts, according to:

$$S_k = n_k \bigvee_j l_k^j \tag{3.19}$$

The first part is the disjunction of level values with l_k^j the j^{th} level of the attribute a_k. The second part is the parameter $n_k \in [1, \neg]$, which allows negating (logical operator NOT) the disjunction when set to $\neg$. The user enters specific rules specifying the levels l_k^j and the parameters n_k.

$$R_i = \bigwedge_{a_k \in A_{R_i}} (n_k \bigvee_j l_k^j) \tag{3.20}$$

The filtering function $F()$ can target the values of those attributes through the use of rules. Thus, a set of disjunction rules allows to discard some of those events. In case either attributes or values do not match the logical definition of the filter, F is TRUE. In that case the event is blocked.

Indexing supports the analysis of data in real-time (e.g., by means of visualization tools, along with associated analysis capabilities). Function $I()$ indexes the events being ingested to improve the performance of answering to queries $q_i \in Q$. It takes as input a set of attributes $\{a\} \subset N_i$ and data in $N \subseteq DL$ as A stored in DL to return a subset of k tuples $\{(a,v)_i\}$, as denoted by:

$$I = I(A,N), (a,v)_i \in I \tag{3.21}$$

Enrichment relies in the $E()$ function to add contextual features (attribute and value $(a,v)_i$) to the events being ingested (e.g. associate geolocation data to an IP-address), to increase their usefulness. This way, the correlation of data outside of the framework will be avoided, enhancing the usage of the collected data to a larger number of use cases. It takes as input a set of n existing attributes $\{a_i\}_{i=1}^n$ as A to produce m new features $\{(a,v)_i\}_{i=1}^m$ as C^m by extracting the values v contained in external enrichment sources S_E, through the use of function $M_e()$, according to:

$$C^m = \{(a,v)_i\}_{i=1}^m \tag{3.22}$$

and

$$C^m = M_e(A, S_E), (a,v)_i \in A \tag{3.23}$$

The enrichment function $E()$ adds new m attributes to the set of existing attributes of the normalized event N_i, as denoted by:

$$E : N_i^{(n)} \to N_i^{(n+m)} \tag{3.24}$$

The resulting event is denoted by $N_i^{(n+m)}$, gathering $C^{(m)}$ features from an enrichment source S_E according to:

$$N_i^{(n+m)} = E(N_i^n, C^m, S_E) \tag{3.25}$$

An example of an enrichment function is adding the geographical location to the event, based on the already existing IP address attribute. Other examples are the inference of User and Asset elements or even the related DNS information (e.g.

retrieved from a "Whois" command).

Finally, *aggregation* corresponds to the $G()$ function, that summarizes the data from events N_i, according to a given set of grouping attributes $H \subset A$ applied to a set of operations functions $k_i() \in K$ (e.g. sums, minimum, maximum or average). The resulting events G are persisted into the DL for further analysis $G \subset DL$. The aggregation function $G()$ is denoted by:

$$G = G(N, H, K), G_i \in N \tag{3.26}$$

with $G_i \in G$ and $\{(a_a, v)_i\}_{i=1..n}$ as the set of summarized attributes and $\{(a_s, v)_j\}_{j=1..m}$ as the set of summarising functions as attributes and their resulted values from functions $k_i() \in K$, according to:

$$G_i = \{(a, v)_i\}_{i=1..n} \cup \{(a, v)_j\}_{j=1..m} \tag{3.27}$$

This allows, for instance, to create indicators such the number of events per second (grouped per category).

3.1.3 Cyber-physical IDS

The Cyber-physical IDS is an external module that identifies threats at CI level, by means of fast path processing. This would typically be the Intrusion Detection System (IDS) systems already deployed at the Critical Infrastructure that feed the FCA framework.

Similarly to the other data sources producing events $N_i \in N$, this module contributes with discrete events $N_i \in N_{IDS}$ as result of the application of the classification function $S()$, to be persisted into the central data lake, according to:

$$N_{IDS} = S(N), N_{IDS} \subset DL \tag{3.28}$$

Where $N_{IDS} \subset DL$ will be used for further analysis.

3.1.4 Data Lake (DL)

The DL is the central storage to the framework. It relies on the $\psi()$ function for gathering and persisting the normalized events $N_{DP} \in DL$, as denoted by:

$$N_{DP} = \psi(N), N_{DP} \subset DL \tag{3.29}$$

Where events N result both from normalized events resulting from the processing in the DPM and from events received directly from the IDS:

$$N = \{N_{IDS}, N_{DP}\} \tag{3.30}$$

Other sources that feed the Data lake include the intermediary functions, such as generated monitoring alerts Y, trust and reputation data T, and aggregation data G.

Thus, the DL maintains the collected data from all existing different sources producing events, including the enriched data, according to the following data sources:

$$DL = \{N, Y, T, G\} \tag{3.31}$$

3.1.5 CI Business Rules

The Critical Infrastructure's Business Rules $R_i \in R$ can be used to describe the policies $R_i \subseteq P$ supposedly adopted by the target organization – so that the FCA can assess the effective compliance with those rules, by scoring incoming events $N_i \in N_{DP}$ against rules $_i \in R$. As a result of this evaluation $Ev(R_i, N) \in \{0,1\}$, alerts A_{R_i} can be generated according to adjustable thresholds.

For example, an organization can specify a set of policies $\{p\}_i \in P$ to enforce the access control to resources from users within their own organizational unit, and report any login attempt violating that policy. Another example is the physical access control to facilities. In that regard alerts can be triggered in case doors of a given department are opened out of the authorized time range.

3.1.6 Analytics

The Analytics Module provides the capabilities to classify events N_{DL} and detected anomalies from events in DL, adding a new attribute for this purpose (a_{n+1}, k). Event-contained features provide the data inputs to classify $K()$ anomalies K or get the knowledge that can help future investigations.

The function $K()$ is used to classify all the events in the data lake, flagging them as normal or anomalous, based on the insights resulting from the correlation of the N_{DL} normalized events in the DL.

For example, by applying a classification learning model $ML^{(k)}$ to a normalized event N_i results in a binary classification $k \in \{0,1\}$ a new feature (a_{n+1}, k) to be included in a event $k_i \in K$. Function $K()$ provides the intelligence to classify event threats, accordingly to:

$$k_i = K(N_i, ML^k), k_i \in K, \forall N_i \in N_{DL} \tag{3.32}$$

In this classification example, $k_i \in \{0,1\}$, $\{0\}$ corresponds to an event marked as non-abnormal, while $\{1\}$ corresponds to an anomaly. Similarly to the enrichment process, a new feature (a_{n+1}, k) in an event N_i containing the result from such classification k is introduced. To this effect, the features $\{a_i\}_{i=1}^{n}$ of second stage events $N_i^{(2)}$ are updated to include an additional attribute $n \leftarrow n+1$, resulting in a third stage event $N_i^{(3)}$.

The events classified as anomalies in $N_b \in B$ can help to understand the chain of events (e.g. since the moment an attack has started until it has finished). These events are chronologically ordered according to their creation timestamp a_t. The reconstruction of the path of events t_b results from the set of sequenced events classified as anomalies following a numerical order $N_1, N_2, N_3 \in N_{DL}$ in the sequence $N_1 \rightarrow N_2 \rightarrow N_3$. That implies that the absolute times when they are produced are consistently ordered so that $t_1 < t_2 < t_3$.

This component relies on models enabled by Machine Learning (ML) algorithms to automate the classification of anomalous events. These classification algorithms help to identify and mitigate the impact of potential threats to the integrity, confidentiality or availability of the CI. Each classification function $K()$ applies the set of learning models $ML_j \in ML$, where j denotes the j^{th} learning model. These models distil and extract the insights from the ingested normalized events. Selected features contained in the normalized event N_i will be used as inputs to the function $K()$, for classification purposes.

In this section we describe these ML algorithms from a generic and abstract point a view, since the framework is agnostic regarding which specific algorithms are used. Nevertheless, we have explored specific ML approaches in the scope of FCA, including for instance the combination of K-means with XGBoost [259] and the usage of decision-trees for policy and audit compliance purposes [363].

The framework includes two different sets of learning models, one for training purposes $ML^{(t)}$ and another for classification $ML^{(k)}$, as denoted by:

$$ML = \{ML^{(t)}, ML^{(k)}\} \tag{3.33}$$

Models are gradually trained according to more recent data to be used for classification purpose. To that aim, a deployment function D is used to transfer the learning model ML_i from the training set $ML_i^{(t)}$ to the classification set $ML_i^{(k)}$, accordingly to:

$$D : ML_i^{(t)} \rightarrow ML_i^{(k)} \tag{3.34}$$

Forensic Analytics

The Forensics Analytics component includes the functions for supporting the investigation process of examiners on extracting evidence for helping them to reconstruct the path and timeline of events to identify and bookmark anomalies and non-conforming situations previously classified by the Analytics Module (AM).

Moreover, this component will provide the control mechanisms over the information being exchanged, read, and processed by different entities along the investigation chain. Thus, in that regard, it will be important to record examiner activities on handling digital forensic evidence along the investigation chain. The adoption of a standardized approach on sharing evidence will help investigators to collaborate in the identification of the root cause of events (e.g. criminals for crimes committed in different jurisdictions). Evidence structure follows a common Forensic Schema Γ_A along the investigation chain. Proof of evidence provenance is achieved by including a specific attribute in events $a_p \in N_i$.

This component provides support to query data from $q_i \in Q$ gathering a set of n rules:

$$q_i = \{R_j\}_{i=j}^{n} \tag{3.35}$$

The forensic analysis function $Z()$ finds out the events $Z \subset N_{DL}$ in DL to be extracted and digitally signed with a hash H. This cryptographic hashing H will help to trace the events to their sources and check their authenticity and integrity, accordingly to:

$$Z = Z(N_{DL}, Q, H), Z_i \in Z, Z_i \in N_{DL} \tag{3.36}$$

Audit Compliance

The Audit Compliance component is in charge of checking security, company policies or adherence to standard practices as a way to improve overall protection.

Such an assessment can result from laws emanated by external regulations or standards the CI operator has chosen or needs to follow. The CIs can also dictate their internal regulatory rules and decides when specific business policies (R_c), comprising corporate laws, policies, plans, and procedures, should be followed.

This component offers continuous assessment and compliance check capabilities. It ensures due diligence, certification, and stakeholder security by checking whether the CI meets the regulatory requirements and follows the industry guidance. In that regard, the audit compliance function $W()$ checks whether compliance rules R_c are being violated or not against the DL events N_{DL}. Such an assessment will help to identify the non-compliance rules $W \subset R_c$, as denoted by:

$$W = W(N_{DL}, R_c), r_i \in W \tag{3.37}$$

This component provides the means, for instance, for computing the scores quantifying the risk level. Therefore, by identifying the non-conformance rules will be possible to score the security risk level function $R_c()$ from rules R_c:

$$R_l : W \to r_l, r_l \in \mathbb{N} \tag{3.38}$$

The auditor can trigger auditing tasks upon request or enable and schedule them to be executed at regular intervals. This component will automatically notify the auditors Ac with the auditing results, according with the specific roles Rl assigned in the platform.

Data Visualization

In the final stage of forensics and compliance auditing activities the Data Visualization component provides visualization capabilities $V()$, reporting, summarizing and displaying the collected data in a suitable, transparent and simple to use way. The function $V()$ fits the visualization data $v_i \in V$ from events N_{DL} in the DL through the use of a set of queries $q_i \in Q$ and defining the summarizing attributes A_s and functions $f_i() \in F$, according to:

$$V = V(N_{DL}, Q, A_s, F), V \subset N_{DL} \tag{3.39}$$

Dashboard panels are the visualization artifacts highlighting the information in V. The panels V present the information in different ways (e.g. histogram with events being received according to their type of requests, logical representation of the infrastructure). That information is ready to be exported and support further analysis by third-party tools.

Real Time Search

The Real Time Search Component relies in function $J()$ for querying data from the DL with low latency to achieve near real-time results. The function $J()$ runs in parallel with other functions along the ingesting pipeline. It returns the set of events $J \subset DL$ for a given period Δt, according to a set of queries $q_i \in Q$ and the indexing data I, as denoted by:

$$J = J(I, Q, \Delta t), J \subset DL \tag{3.40}$$

The indexing function $I()$ offers the capabilities to function $J()$ to achieve results in near real-time. The collected information is also made available to actors through user interfaces and third-parties, by means of integration components (e.g. to detect intruders before they have access to resources).

3.1.7 Trust and Reputation Indicators

This component integrates the work of Caldeira et al. [364] for computing sets of trust and reputation indicators $K_{\Delta t}$ for a period of time Δt. The function $K_{\Delta t}$ relies in the trust function $T()$ to compute those indicators in a given period from events N_{DL} in DL, as defined by:

$$K_{\Delta t} = T(N_{\Delta t}) \tag{3.41}$$

The trust function $T()$ includes the reference levels of risk (Rl_i) and the current measured risk levels (Ml_i), in order to detect deviations and highlight threats.

Therefore, $K_{\Delta t}$ takes function $T()$ for averaging the measured risk level Ml_i and the received risk alert level Rl_i, computed for n events $N_{\Delta t}$, according to:

$$T(N_{\Delta t}) = 1 - \frac{\sum_{i \in N_{\Delta t}} \Phi(Ml_i, Rl_i)}{n} \tag{3.42}$$

Finally, the function $R_t()$ computes the risk level between CIs for each event N_i. This function considers current events (N_i) and past events stored in the data lake (N_{DL}), with $T_{N_i}^{(i,j,s)}$ denoting the cascade risk level between critical infrastructures i and j for service s and for the i^{th} event N_i, as defined by:

$$T_{N_i}^{(i,j,s)} = R_t(N_i, N_{DL}, T_{N_i}^{(i,j,s)}) \tag{3.43}$$

and

$$R_t(N_i, N, T_{N_i}^{(i,j,s)}) = \frac{(N-1)\phi T_{N_i}^{(i,j,s)} + K_{N_i}}{(N-1)\phi + 1} \tag{3.44}$$

The function Φ is a discrete function and value k applies penalties to higher differences between levels of risk (Rl_i) and the current measured risk levels (Ml_i), as denoted by:

$$\Phi(Ml_i, Rl_i) = |\frac{(Ml_i - Rl_i)}{4}|^k, k \in R^+ \tag{3.45}$$

The concept of aging is implemented by applying more weight ϕ to recent events. Trust and reputation can be regarded as a set of continuous indicators to be persistent $t_i \in T$, according to:

$$T = Tr(N_{\Delta t}, T^{(i,j,s)}), t_i \in T \tag{3.46}$$

3.1.8 Business Rule Compliance Monitoring

The Monitoring Module continuously monitors data to assess the compliance of observed events with predefined rulesets, to assure the CI remains compliant with adopted user and business rules. It also allows operators to identify non-compliant events $N_i \in M$, by checking the rules $r_{mi} \in R_m$. The monitoring function $M()$ checks whether user and business rules R_m are being met by evaluating them against the events in DL N_{DL} for a given period of time Δt. The function $M()$ assesses the CI business rules $r_i \in R$ supporting auditing compliance activities from events in Data Lake $N_i \in DL$, and trust and reputation risk alert levels Rl_t to classify eventual threats and triggering alerts to the operators, according to:

$$M = M(N_{DL}, \Delta t, R_m), N_i \in M \tag{3.47}$$

The set M gathers the events $N_i \in M$ deemed as non-compliant with rules R_m, which trigger non-compliance event alarms.

3.1.9 Orchestration

The Orchestration Module provides the means for coordinating and automating the management of the various framework modules.

3.1.10 Actors and Roles

Departing from a conventional IDS role, the FCA constitutes a system that persists relevant data, providing the means for forensics experts to search and identify relevant events, also enabling continuous auditing of organisational processes and procedures – for these purposes, knowledge extraction mechanisms allow for the definition of rules that can be used for both (semi-)automated forensics analysis and for auditing compliance purposes. Thus, it can be considered that operators and security analysts are the main actors of the framework.

Forensic and auditing experts are security analysts searching for facts to be extracted as evidence. They investigate the trail of events to identify the root cause of relevant events and involved systems and criminals. Along their investigation, they run *ad hoc* queries in the available data in the data lake. At the end, collected evidence are reported to stakeholders to help them to organize and take responsive actions.

Operators can define the level of criticality associated with non-compliance events for each rule. Those criticality levels are evaluated either from events being monitored or from triggered alerts. CI operators rely on the Data Visualization Component capabilities and make use of user interfaces depicting summarized data (e.g. graphically displayed in dashboards).

Permission levels granted to each actor are role-based. To increase granularity, each actor is associated with a set of role(s) for each platform module, and each module role is associated with a set of (allowed) functions.

The security analyst perceives the level of threat from the incoming events highlighted by the Monitoring Module. These actors take their investigations activities by introducing *ad hoc* queries to understand the chain of events.

The actions and decisions taken by these actors are logged as self-accountable mechanisms of the framework. Therefore, all these actions are potential inputs to resolve future conflicts such as data privacy violation allegations and studying examiner investigative styles for learning and training purposes.

3.2 Proof-of-concept Implementation

A Proof of Concept (PoC) of the proposed FCA framework has been implemented and deployed as part of the Intrusion and Anomaly Detection System (IADS) of the ATENA H2020 project [63].

The ATENA project combines new anomaly detection algorithms and risk assessment methodologies within a distributed environment, to provide a suite of integrated market-ready ICT networked components and advanced tools embedding innovative algorithms [365]. It provided the reference scenarios, use cases and testbeds needed to validate our research, allowing to collect valuable data from realistic testbeds in order to evaluate the prototype frameworks throughout experimental use cases scenarios.

Within the IADS, the FCA platform provides the mechanisms to persist a broad spectrum of digital pieces of evidence obtained from multiple data sources, for forensics analysis and policy conformity checks. The implementation of the FCA was supported by an Elasticsearch stack (Elasticsearch, Logstash, and Kibana) as an efficient manual and semi-automatic searching tool to cope with the complexity and massive amount of available data.

Within ATENA, the IADS module is responsible for monitoring the underlying CI environment using distributed probes. Moreover, the distributed probes will generate events for suspicious activity, which will be processed before being sent to the upper layers of the ATENA architecture to be classified [1].

Figure 3.2 illustrates the ATENA cyber-security architecture with their modules. Beyond the FCA, it includes: different types of probes, from conventional network and host probes to IACS field-specific probes; a Domain Processor per scope, backed by a Message Queuing system; a distributed Security Information and Event Management (SIEM), for support of streaming and batch processing; and a Data Lake, where all the data is stored [1].

3.3 Evaluation and Discussion

In this section we evaluate and discuss the FCA framework. We start this evaluation with a general description of the experimental setup, followed by the analysis of the results obtained for data ingestion and query handling performance evaluation. Next, we present two Use Cases that highlight the potential of the proposed framework for detecting cyberattacks: the Password Cracking Use Case, and the

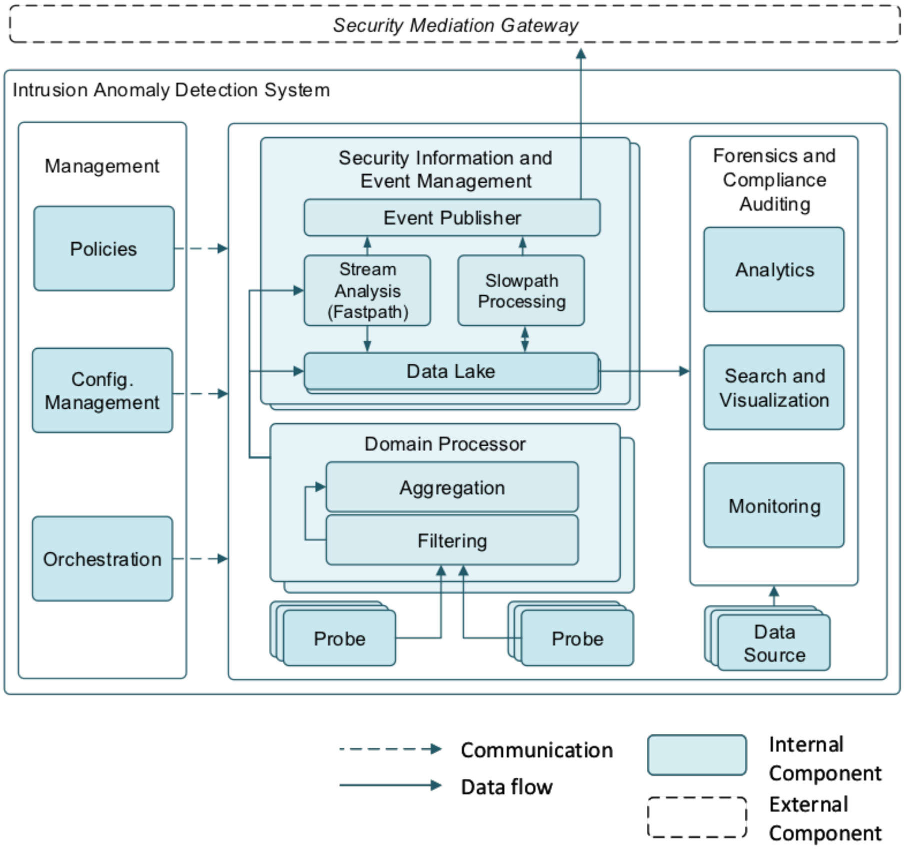


Figure 3.2: ATENA Cyber-Security Architecture [1]

User Account Control Bypass Use Case.

It should be noted that the proposed FCA framework is intended to be agnostic regarding which specific analytics algorithms are used. In fact, each specific deployment scenario will probably call for different algorithms. For this reason, this evaluation focuses more on performance than accuracy. Nonetheless, as further discussed in the next chapters, we have also assessed the accuracy of specific ML algorithms in the scope of FCA, such as Decision Trees [363] and K-Means with XGBoost [259].

3.3.1 Experimental Setup

As already mentioned, the implemented PoC uses the Elasticsearch open-source full-text search engine technology for storage and query capabilities. Additionally, the FCA platform forms a cluster composed of a set of nodes deployed as Docker containers.

The infrastructure for the experiments used a VMware Hypervisor to host a single

Virtual Machine (VM) with 8 vCPUs (Intel Xeon Gold 5120 CPU @ 2.20GHz), 64GB of RAM and 48G of SSD storage. This VM supported the Elasticsearch nodes, deployed as Docker containers from Elasticsearch 6.2.4 images and the clients requesting ingestion and computation capabilities from these nodes.

The Elastic Search nodes were constrained in terms of CPU and memory (through the use of mem_limit: 2G, memswap_limit: 2G cpus: 0.5). The ingestion producers generate events to be ingested into a given node with a bulk operation. They were deployed as containers from custom Docker images (python:3.8-slim-buster). The containers were limited to 50 percent in terms of CPU and 2GB of RAM (Docker settings: –memory 2G, –memory-swap 2G and –cpus 0.5). Similarly, also the computation clients were deployed as containers from custom Docker Python images taking the responsibility on querying a given node, in this case constrained with 25% percent in terms of CPU and 2GB of RAM.

The experiments use synthetic events, generated by the clients, with a payload of 160 bytes, formed by two fields: "any" and "timestamp", the latter corresponding to the moment the event was created. Events are ingested by the FCA platform through a unique bulk request from the client. Listing 3.1 depicts the client python source code feeding the ingestion of events into the "river" index and "tweet" type of the Elasticsearch cluster. It receives the identification of the block for events and the number of events to be generated in a loop and stored in an array. Finally, they are pushed into a single endpoint of the cluster, in a single bulk operation, and the time to ingest them is recorded.

Listing 3.1: Client Source Code

```
from datetime import datetime
from elasticsearch import Elasticsearch
from elasticsearch import helpers
import time
import sys

start_time = time.time()
block=int(sys.argv[1])
records=int(sys.argv[2])
es = Elasticsearch("http://172.27.221.159:9204/")
actions = [
  {
    "_index": "river",
    "_type": "tweet",
    "_id": block+j,
    "_source": {
        "any":"data" + str(j),
        "timestamp": datetime.now()}
  }
  for j in range(0, records)
]
helpers.bulk(es, actions)
elapsed_time = time.time() - start_time
```

The purpose of this setup was to assess the performance of the framework, by measuring its latency on ingestion and computing operations. To this purpose, the FCA framework was evaluated with different combinations of Elasticsearch nodes (3, 5 and 20 nodes) and shards (1 and 5 primary shards, and 1 to 5 replica shards). Different combinations of feeding clients (also designated as producers) were also used. Within the Elasticsearch terminology, a shard is a logical block that stores data and indexes. Shards can be replicated by different nodes along the cluster, with each one being classified as a primary or replica. A primary shard is the block assigned with the role of the primary storage for data, while replicas maintain a copy. Replicas can be added or removed anytime, to scale out/in computing queries (by default, Elasticsearch would create one primary shard and one replica for every index).

To assess the FCA ingestion performance, 10 producers generate 10 Million events to be ingested evenly, distributed by all cluster nodes. Regarding FCA computation performance, a client evenly queries the cluster nodes with a simple operation (counting the 10 Million ingested events), using the DSL Query language (based on JSON). Our analysis used mostly the maximum time it took to complete the operations, even though minimum and average time are also reported.

3.3.2 Ingestion Performance

A first set of experiments measured the time it takes to ingest 10 Million events when using eighteen distinct setups. First, we adjusted the number of nodes (N) to 3, 5 or 10. Moreover, we also adjusted the number of primary shards (PS) (1, 3, 5 and 10) and replicas (R) (1, 3 and 5).

Table 3.1 summarizes the obtained results, for each setup, when ingesting 10 Million events. Figure 3.3 depicts the maximum ingestion time (in milliseconds) when running the 10 clients for different cluster sizes and under different settings of primary shards and replicas. It should be noted that, for this specific experiment, there were no significant differences observed between maximum, average and minimum times.

Table 3.1: Time to ingest 10 Million events (ms)

N	PS	R	Minimum	Maximum	Average
3	1	1	207 384	217 280	212 591
3	3	1	177 157	180 524	179 492
3	5	1	183 895	188 718	186 804
3	5	2	176 575	181 951	180 244
3	5	3	183 568	186 511	185 543
3	10	1	252 972	255 232	254 256
5	1	1	2 636 090	2 792 717	2 751 700
5	3	1	1 319 386	1 383 388	1 350 092
5	5	1	1 334 385	1 409 409	1 369 947
5	5	2	1 283 605	1 362 303	1 329 752
5	5	3	1 359 796	1 419 684	1 385 052
5	10	1	1 208 413	1 225 788	1 219 440
10	1	1	2 590 074	2 706 697	2 680 579
10	3	1	863 380	891 132	875 575
10	5	1	871 322	901 407	886 429
10	5	2	863 380	881 599	873 379
10	5	3	864 986	910 689	891 984
10	10	1	856 180	893 589	874 900

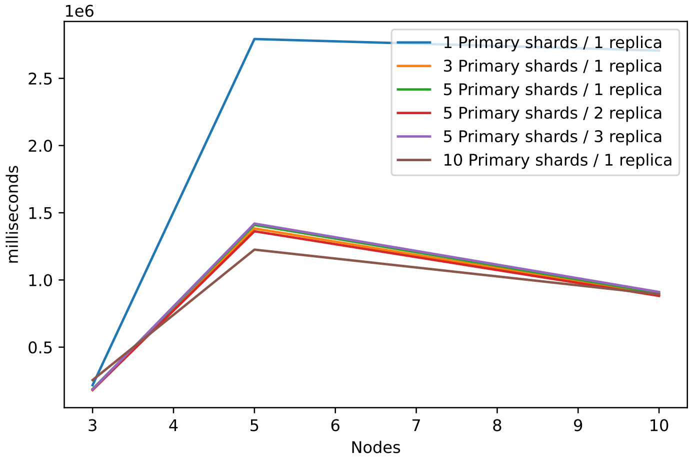


Figure 3.3: Max Ingestion Time (ms)

The achieved results highlight the scalability of the cluster. Most notably, it can be observed that the injection latency decreases as the number of nodes increases from 5 to 10. For all different settings, including the primary shards (3, 5, and 10) and replicas (1, 2, and 3), results denote an improvement in the ingestion performance when the cluster size increases from 5 to 10 nodes. On the other hand, the cluster reveals a poor performance scenario when ingesting before the 5 node threshold is reached. Maybe it can be explained with the trade-off between the size and overhead caused by management activities of the cluster.

3.3.3 Computational Performance

A second set of experiments measured the framework performance when computing the 10 Million events previously ingested along the first experiment. For this purpose, 10 different clients are running the same query simultaneously, distributed evenly across the available nodes. The query requires checking the different ID of all events (cf. Listing 3.2).

Listing 3.2: Query Counting Records

```
{"size":0,
    "aggs":{
        "DistinctWords":{
            "cardinality":{
                "field":"id"
    }   }   }
}
```

The eighteen aforementioned setups were also used for this experiment, therefore considering different cluster sizes, different numbers of primary shards and different numbers of replicas. Table 3.2 summarizes the observed results.

Figures 3.4, 3.5, and 3.6 depict the maximum, minimum and average computation time for each setup. Unlike the previous experiment, larger variations between maximum, minimum and average times were observed in a few situations, namely when using one primary shard and one replica that keeps a low minimum computation time for different cluster sizes.

These results highlight the scalability of the cluster, except in the case of 1 and 10 primary shards. All other settings (3 and 5 primary shards and 1, 2, and 3 replicas) denote an effective improvement in ingestion performance when increasing the size of the cluster: from 3 to 5 nodes, and from 5 to 10 nodes. This also holds when different settings are considered for primary shards (3 and 5) and replicas (1, 2, and 3).

Table 3.2: Time to compute 10 Million events (ms)

N	PS	R	Minimum	Maximum	Average
3	1	1	6	18	11
3	3	1	8	108	43
3	5	1	9	182	40
3	5	2	6	60	15
3	5	3	9	27	15
3	10	1	16	119	41
5	1	1	6	19	11
5	3	1	8	62	19
5	5	1	8	31	16
5	5	2	7	63	17
5	5	3	9	35	17
5	10	1	86	493	274
10	1	1	14	212	122
10	3	1	9	14	12
10	5	1	6	25	15
10	5	2	10	28	16
10	5	3	9	14	11
10	10	1	116	681	366

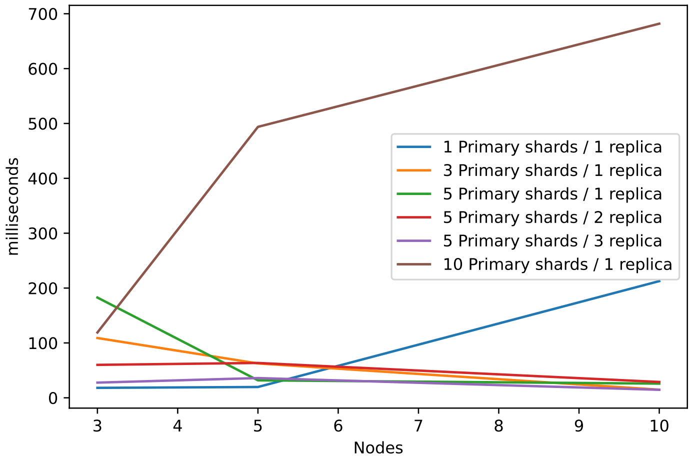


Figure 3.4: Maximum Computation Time (ms)

The results also denote that defining a cluster with one primary shard is a bad option when the objective is to scale it in a distributed environment. Another observation is that the computation performance improves when the number of shards (primary and replica) is higher than the number of nodes.

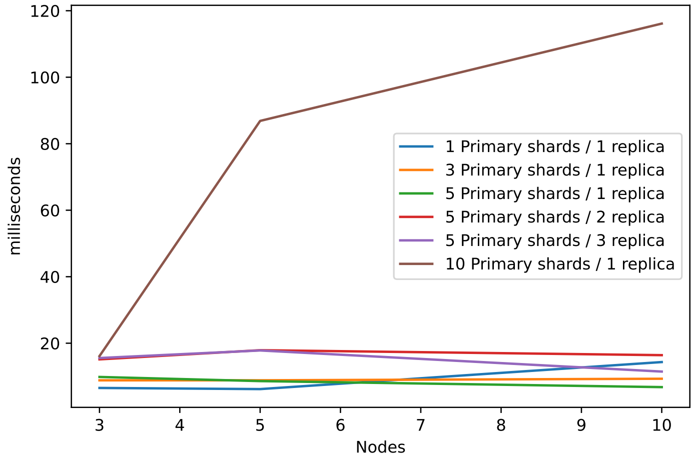


Figure 3.5: Minimum Computation Time (ms)

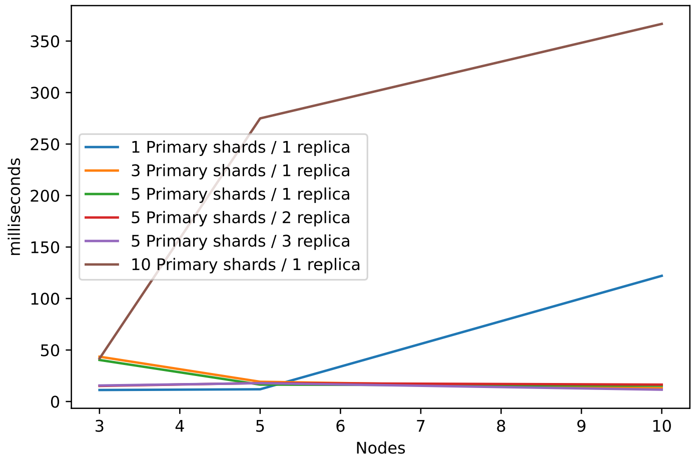


Figure 3.6: Average Computation Time (ms)

3.3.4 The Password Cracking Use Case

Many attacks start with an unauthorised entry into the network, and then evolve to scouting and exploitation of weaknesses in other nodes of the network, using horizontal movement[366]. Quite often, such attacks involve a progression chain, with multiple stages and different sources. Independent ML models might recognise such attacks in different sources, albeit in a stateless way, lacking the overall perspective about the attack progression and the relationship between multiple stages.

In this section we show how the FCA framework can be used to improve the overall detection process, by recognising the different stages of those attacks through the analysis of different sources. More specifically, this is done by combining the classification of heterogeneous data from different sources to recognise multistage attacks and the relationship between them.

A rule R_i recognizing an attack at stage i can be applied along n stages, with $i \in \{t_0, t_1, ..., t_n\}$. Optionally, an additional rule $R_{(i-1,i)}$ may be used to check if there is a connection between stages $i-1$ and i. This way, it is possible for instance to flag a "password crack" attack at stage S_1, preceded by a "recognition" attack at stage S_0 within a predefined time window (for instance 7 days: $t_{(0,1)} = (t_1 - t_0) <= 7$).

According to the FCA framework, rules can target the classified events $E_i \in E$ in the DL. Moreover, each ML model $ML_i^{(S_j)} \in ML$ can be attached to a distinct source $j \in \{S_0, S_1, ..., S_n\}$. Each of those rules R_j can be defined as a tuple encompassing three sub-rules $(R_{i-1}, R_i, R_{(i-1,i)})$. The first sub-rule R_{i-1} evaluates an event attribute $a_j \in E_i$ in a first source S_0, with name "attack“ and value "true“. The second sub-rule R_i also checks a previous classified attack in a source S_0. Finally, a third sub-rule $R_{(i-1,i)}$ evaluates the precedence between those two events, for instance checking if they (E_{i-1} and E_i) differ less than than 7 days.

To demonstrate this Use Case, we used the TON_IoT datasets [367], which comprise heterogeneous data sources collected from telemetry datasets of IoT services, Windows and Linux-based datasets, as well as datasets of network traffic. Events are labelled as normal or as generated under the different kind of attacks, including: Scanning, Password cracking, Denial of Service (DoS), Distributed DoS, Ransomware, Backdoor, Injection, Cross-site Scripting (XSS), Password cracking and Man-In-The-Middle (MITM).

The Scanning Attack (and also Reconnaissance and Probing Attacks) corresponds to the first stage of the cyber kill chain model. This kind of attacks usually tries to discover active IP addresses and open ports in the targeted network, in order to prepare the following stages of attack.

In our first application scenario, the capabilities of the FCA Framework are leveraged by using Elasticsearch technology. First to ingesting the different TON_IoT datasets to different indexes, and later to apply the rules aiming to recognise the various stages of attacks supported by their classified events. In this scenario, it is possible for instance to check if a password cracking attack occurred (within the "Modbus" TON_IoT dataset), and if it was preceded by a scanning attack (in the "Windows 7" TON_IoT dataset). According to data contained in those datasets, a password cracking attack occurred on April 27, and it was possible to realise it was preceded by a scanning attack that occurred two days before.

A first rule R_i checks if events are classified as "password" attacks. For that purpose, an Elasticsearch query Q_0 (type:"password" and date>="27-Apr-19") was defined, targeting the “modbus” dataset, stored in a specific "modbus" Elasticsearch index, as defined in Listing 3.3 query, which returned 4665 results.

Listing 3.3: Password Cracking Elasticsearch Query to Modbus index R_i

```
GET /modbus/_search
{
  "query":{
    "bool":{
      "must":[
        {"match":{"type":"password"}}
      ],
      "filter": [
        {"range":{"date":{"gte":"27-Apr-19"}}}
      ]
    }
  }
}
```

A second Elasticsearch query Q_1 (type :"scanning" and @timestamp $>=$ "2019-04-25") materialises the rule $R_{(i-1)}$, which checks if current the "Password Cracking" was preceded by a "Scanning attack". This query checks another source (“windows 7” dataset collected into a specific Elasticsearch index "windows7"), as defined in Listing 3.4, which returned 71 results.

Listing 3.4: Scanning Phase Elasticsearch Query to Windows 7 index $R_{(i-1)}$

```
GET /windows7/_search
{
  "query":{
    "bool":{
      "must":[
        {"match": { "type":"scanning"}}
      ],
      "filter":[
        {"range":{"@timestamp":
            {"gte":"2019-04-25"}}}
      ]
    }
  }
}
```

Finally, the rule $R_{(i-1,i)}$ has been implicitly materialised in the previous Elasticsearch queries, namely as part of the condition @timestamp $<=$ "2019-04-25" in $R_{(i-1)}$ and date$>=$"27-Apr-19“ in R_i.

All the Elasticsearch queries filtering the type and date of the attack, as well as the number of results along the two stages, are summarised in Table 3.3.

Table 3.3: Sequence of Stage Queries to recognize a Password Cracking Attack

Stage	**(1) Type**	**(2) @timestamp**	**Results**
1	password	27-Apr-19	4665
2	scanning	2019-04-25	71

3.3.5 The User Account Control Bypass Attack Use Case

Currently, forensic activities focus mostly on servers and network security. Usual approaches include IDS, firewalls and proxies, while the regulatory compliance focuses on analyzing logs from assets. On the other hand, attacks on desktop endpoints are usually underestimated. Nevertheless, it is important to realise that a significant number of recent exploits attacks start at desktop endpoints.

Actions taken by attackers in the Windows endpoints Operating Systems (OS) leave back a set of logs, which raise the chances of detection. In this Section, we describe how the proposed framework can be used to detect such attacks by recognising the stages of an User Account Control (UAC) Bypass attack by means of analysing logged actions at the endpoints.

The UAC is a Windows OS capability that allows programs to run at administrator level. When the programs introduce changes, the user is informed and prompted for confirmation to elevate their privileges. Despite this, in some circumstances, certain Windows programs are allowed to elevate privileges without authorisation by the UAC.

In this experimental work, we started by mimicking the usual steps of this kind of attack by injecting the usually generated logs into the endpoint OS, in order to later determine if it is possible to detect such attacks. According to MITRE ATT&CK [368], a UAC Bypass attack comprises the following steps: USB compromise, persistence, theft of credentials, and their reuse by installing a backdoor in a local account. Finally, the attacker may want clear the logs.

In a first stage S_0, the attacker tries to compromise an endpoint OS by exploiting its weaknesses – for instance through the use of virtual keyboard USB devices or Powershell scripts. As a result, the attacker may compromise the endpoint system by installing a Command and Control (C2) agent. To that aim, a PowerShell C2 can be launched from a virtual keyboard USB device, a corresponding DriverFrameworkd-

UserMode (2001) event log being recorded. Other event logs can also be recorded at the system level, in the case a Powershell script is executed (4100) or a new service (System 7045) is installed.

In the second stage S_1, the backdoor is installed, which may trigger flagging events at the Windows endpoint OS, such as new services (System 7045), scheduling tasks (System 4698), or registry modification (System 4657).

In the third stage S_2, the attacker steals the credentials to later reuse them. This can trigger system windows event logs such as successful login (System 4624) or failed login (Security 4625).

In the fourth stage S_3, the backdoor is installed through the use of a local account. This is achieved by including a new local user into the Administrators group in a critical system. As a consequence, the following Windows security events may be triggered: adding new users (Security 4720) or adding the user to the local group (Security 4732).

Finally, in the last stage S_4, the attacker may try to clear its tracks from endpoints and critical systems. Those actions can be logged as a new service (System 7045), scheduling a new task (System 4698), or even trying to modify the registry (System 4657).

For each one of the previous actions, a log event is recorded into the Windows endpoint OS, by the use of a Powershell script. Next, the WindowsLogBeat agent installed in the host pushes those logs into the Elasticsearch. There, those logs are parsed by a grok regular expression in the Logstash component (DAM), processed along the pipeline by the DPM and, finally, stored into Elasticsearch (DL).

The operator can define a rule R_j as a tuple of stage rules $(R_i,R_{i-1},R_{(i,i-1)})$, supported by Elastic queries. These rules try to recognise the typical sequence of stages R_j of an UAC Bypass attack, with $j \in \{0,1,2,3,4\}$. Stage queries aim to recognise a specific stage of the attack chain and the (possible) connection to the previous stage actions.

An example of this query, filtering the event_id (<EVENT_ID>) and @timestamp (<TIMESTAMP>) fields, is provided in Listing 3.5. Table 3.4 summarises obtained results.

Listing 3.5: Query Windows Logs Events

```
GET /windowslogs/_search
{
  "query":{
    "bool":{
      "must":[
        {"match": { "type":"windowseventlog"}}
      ],
      "filter":[
        {event_id:<EVENT_ID>},
         {"range":{"@timestamp":
            {"gte":"<TIMESTAMP>"}}}
      ]
    }
  }
}
```

Table 3.4: Sequence of Stage Queries to recognize a UAC Bypass Attack

Stage	(1) Event_Id	(2) Timestamp	Results
1	2001	2019-04-24T01:42:58	1
2	4698	Stage 1 Timestamp	1
3	4624	Stage 2 Timestamp	1
4	4720	Stage 3 Timestamp	1
5	7045	Stage 4 Timestamp	1

3.4 Summary

The proposed framework intends to provide a foundation to build up improved solutions addressing the current and future requirements in the field of FCA. In this chapter, we described its functional blocks, as well as their internal, input, and output communications.

The framework is able to manage a significant amount of heterogeneous data moving under intense flows, from the moment it is ingested until insights are extracted, by relying on distributed computing resources to achieve a high-performing system providing results near real-time. The key components of the framework were explored in terms of their ability to scale to specifically cope with the volume and speed at which data is produced.

The experimental evaluation, which encompassed several experiments to evaluate the performance of a cluster under different settings for ingestion and computing workloads, demonstrated the suitability of the proposed framework to cope with FCA requirements at scale. Moreover, this experimental evaluation also showed how the proposed framework can be useful to detect typical chains of attacks.

In the next three chapters we will further explore the framework, in the scope of more specialized techniques.

Table 3.5: Data and Function Symbols

Module	**Name**	**Type**	**Description**
Data Acquisition	E_i	Data	Event
	a_i	Data	Event Attribute
	$\xi_{\Delta t}$	Data	Counting events
	S_i	Data	Data Source
	$\theta_{\Delta t}$	Data	Output rate
	I()	Function	Data ingestion
	N()	Function	Normalization
	Q()	Function	Queue
	P()	Function	Parsing
	V()	Function	Validation
Domain Processor	C^m	Data	Enriched event
	S_E	Data	Enrichment source
	H	Data	Grouping Attributes
	K	Data	Operations
	Q_i	Data	Queries
	R_i	Data	Rule
	S_k	Data	Rule statement
	G	Data	Summarizing data
	E()	Function	Enrichment
	M()	Function	Enriching from sources
	F()	Function	Filtering
	I()	Function	Indexing
	$k_i()$	Function	Operation
	G()	Function	Summarizing
CI Business Rules	P	Data	Policies
	A_{R_i}	Function	Alert from a rule
	Ev()	Function	Auditing evaluation
Data Lake	DL	Data	Data Lake
	N	Data	Normalized events
	$\psi()$	Function	Persist
Analytics	K	Data	Anomalies
	ML_i	Data	ML model
	K()	Function	Classification anomalies
	D()	Function	Deployment
Forensics Analytics	q_i	Data	Query
	Γ_A	Data	Forensic schema
	Z	Data	Forensic analysed event
	Z()	Function	Forensic analysis
	H	Function	Hash
Audit Compliance	W	Data	Non-compliance rules
	W()	Function	Audit compliance
Data Visualization	V()	Function	Visualization
Real time Search	Δt	Data	Period of time
	J()	Function	Real Time search
Trust & Reputation	K	Data	indicators
	Ml_i	Function	Measured risk level
	Rl_i	Data	Risk level alert
	$Rt()$	Function	Risk level
	$\Phi()$	Function	Risk level penalty
	T()	Function	Trust
Business Rule	R_c	Data	Business policies
	M	Data	Non compliant rules
	M()	Function	Monitoring

Chapter 4

An Anomaly Detection Framework for Large Log Datasets

Contents

Hosts and network systems typically record their detailed activity in log files with specific formats, which are valuable sources for anomaly detection systems. The growing number of hosts per organization and the growing complexity of infrastructures result in an increasingly massive amount of recorded logs available – requiring simpler and cheaper anomaly detection methods. While classic log management applications based on manual or preset rule-based analysis still hold value, they do not scale well with the large volumes of data that are currently available, often being limited in terms of exploratory analysis: they fail to detect anomalies not predefined in the rules (i.e., based on prior knowledge) and/or require considerable operator expertise to reach their full potential. This opens the way for the introduction of new approaches, which are less dependent on prior knowledge and human-guided workflows, being able to extract knowledge from large volumes of log data in a scalable and (semi)automated way. Moreover, taking advantage of available computational resources may contribute to achieve performance and accuracy for identifying anomalies and retrieving forensic and compliance auditing evidence.

Over the past years, several automated log analysis methods for anomaly detection have been proposed. However, most of them are not suitable to the scale needed for identifying unknown anomalies from the growing high-rate amount of logs being produced and their inherent complexity. To address such challenges, novel integrated anomaly detection methods employing parallel processing capabilities for improving detection accuracy and efficiency over massive amounts of log records were researched. These methods combine the k-means clustering algorithm [369] and the gradient tree boosting classification algorithm [370] to leverage the filtering capabilities over normal events, in order to concentrate the efforts on the remaining anomaly candidates. Such an approach may greatly contribute to reducing the involved computational complexity.

The characteristics of abnormal system behaviors were obtained by extracting 14 statistical features containing numerical and categorical attributes from the logs. Then, the k-means clustering algorithm was employed to separate anomalous from normal events into two highly coherent clusters. The previous binary clustered data serve as labeled input to produce a gradient tree boosting algorithm implemented by the XGBoost system [371]. Its role is to produce a set of simple rules with the rationale for generalizing the classification of anomalies of a large number of unseen events in a distributed computing environment. K-means, XGBoost and Dask [372] provide the tools for building scalable clustering and classification solutions to find out the candidate events for forensic and compliance auditing analysis.

This chapter is organized as follows. Section 4.1 discusses background concepts and related work. Section 4.2 describes the proposed approach. Section 4.3 presents the validation work and discusses the achieved results, and Section 4.4 concludes the chapter.

It should be noted that the content of this chapter has already been published, to a large extent, in [259].

4.1 Background and Related Work

This section starts by providing the reader with the key base concepts related with the scope of the proposed approach. Next, related work is discussed (Section 4.1.2). Finally, the algorithms and tools that were adopted in this work are presented, namely k-means (Section 4.1.3), decision trees (Section 4.1.4), gradient tree boosting on XGBoost (Section 4.1.5) and Dask (Section 4.1.6).

4.1.1 Base Concepts

By definition, an anomaly is an outlying observation that appears to deviate markedly from other members [254]. Anomalies are typically classified into three types: point anomalies, contextual anomalies and collective anomalies. A point anomaly in data significantly deviates from the average or normal distribution of the rest of the data [255]. A contextual anomaly is identified as anomalous behavior constrained to a specific context, and normal according to other contexts [255]. Collection of data instances may reveal collective anomalies while anomalous behavior may not be depicted when analyzed individually [373]. Time series data include a significant amount of chronologically ordered sequence data sample values retrieved at different instants. Their features include high-dimensionality, dynamicity, high levels of noise, and complexity. Consequently, in the data mining research area, time series data mining was classified as one of the ten most challenging problems [374].

Anomaly detection for application log data faces important challenges due to the inherent unstructured plain text contents, the redundant runtime information and the existence of a significant amount of unbalanced data. Application logs are unstructured and stored as plain text, and their format varies significantly between applications. This lack of structure presents important barriers to data analysis.

Moreover, runtime information, such as server IP addresses, may change during execution. Additionally, application log data are designed to record all changes to an application and hence contain data that are significantly unbalanced in comparison to non-anomalous execution. The size and unbalanced nature of log data thus complicate the anomaly detection process.

4.1.2 Related Work

Various anomaly detection methods have been proposed for applying clustering algorithms to detect unknown abnormal behaviors or potential security attacks.

Some of those proposals have addressed the usage of log analysis as one of the input sources for anomaly detection. Chen and Li [321], for instance, proposed an improved version of the DBSCAN algorithm for detecting anomalies from audit data while updating the detection profile along its execution. Syarif et al. [323] compared five different clustering algorithms and identified those providing the highest detection accuracy. However, they also concluded that those algorithms are not mature enough for practical applications. Hoglund et al. [324], as well as Lichodzijewski et al. [375], constructed host-based anomaly detection systems that applied a self-organizing maps algorithm to evaluate if a user behavior pattern is abnormal.

Clustering techniques, such as the k-means algorithm, are often used by intrusion detection systems for classifying normal or anomalous events. Münz et al. [326] applied the k-means clustering algorithm to feature datasets extracted from raw records, where training data are divided into clusters of time intervals for normal and anomalous traffic. Li and Wang [327] improved a clustering algorithm supported by a traditional means clustering algorithm, in order to achieve efficiency and accuracy when classifying data. Eslamnezhad and Varjani [328] proposed a new detection algorithm to increase the quality of the clustering method based on a MinMax k-means algorithm, overcoming the low sensitivity to initial centers in the k-means algorithm. Ranjan and Sahoo [329] proposed a modified k-medoids clustering algorithm for intrusion detection. The algorithm takes a new approach in selecting the initial medoids, overcoming the means in anomaly intrusion detection and the dependency on initial centroids, number of clusters and irrelevant clusters.

Other authors have used hybrid solutions for log analysis, combining the use of the k-means algorithm with other techniques for improving detection performance. They realized that despite the inherent complex structure and high computational cost, hybrid classifiers can contribute to improving accuracy. Tokanju et al. [332], for instance, took advantage of an integrated signature-based and anomaly-based approach to propose a framework based on frequent patterns. Asif-Iqbal et al. [322] correlated different logs from different sources, supported by clustering techniques, to identify and remove unneeded logs. Hajamydeen et al. [325] classified events in two different stages supported by the same clustering algorithm. Initially, it uses a filtering process to identify the abnormal events, and then it applies it for detecting anomalies. Varuna and Natesan [333] introduced a new hybrid learning method integrating k-means clustering and Naive Bayes classification. Muda et al. [334] proposed k-means clustering and Naive Bayes classifiers in a hybrid learning approach, by using the KDD Cup'99 benchmark dataset for validation. In their approach, instances are separated into potential attacks and normal clusters. Subsequently, they are further classified into more specific categories. Elbasiony et al. [335] used data mining techniques to build a hybrid framework for identifying network misuse and detecting intrusions. They used the random forests algorithm to detect misuses, with k-means as the clustering algorithm for unsupervised anomaly detection. The hybrid approach is achieved by combining the random forests algorithm with the weighted k-means algorithm.

Some research focused on detecting which outliers constitute an anomaly when applying clustering methods [376, 330]. Liao and Vemuri [330] computed the membership of data points to a given cluster, using Euclidean distances. Breunig et al. [377] stated that some detection proposals weight data points as outliers.

Hybrid approaches have indeed proven quite interesting. However, in general, proposed solutions still take considerable amounts of time to generate models for particular datasets, aggravated by the growth patterns normally associated with log sources in production systems. This situation calls for alternative strategies that are able to improve speed (as well as accuracy and efficiency) by taking advantage of innovative algorithmic approaches together with improved parallelism.

This work focuses on scalability and interpretability, since the aim is to use it in the forensics and audit compliance contexts already discussed in previous Chapters. The goal is to be able to sift through data to select candidates for a more detailed analysis or inspection.

Similarly to other works, a hybrid approach for identifying anomalies for log analysis was adopted. However, unlike other works, there is a focus on speed, agility and interpretability. The proposed approach allows training and classifying out-of-core datasets in scenarios involving the computation of very large datasets with limited computing resources, parallelizing their processing by distributing them across the available nodes. Therefore, it is supported by clustering and classification algorithms that are able to scale and produce interpretable results. The presented method works in two stages: first, it starts with the unlabelled dataset, implementing a binary anomalous event classifier through the use of unsupervised learning algorithms; the second stage produces a set of simple rules by considering the previously classified data through the use of supervised learning algorithms. It combines the k-means algorithm for clustering anomalies and gradient tree boosting to produce a simple set of interpretable rules to be parallelized in a distributed environment on classifying a large amount of data. Next, the techniques adopted by the proposed approach. will be presented, in more detail.

4.1.3 K-Means

K-means remains one of the most popular clustering methods and one of the most relevant algorithms in data mining [369]. The main advantage of k-means is its simplicity. By starting with a set of randomly chosen initial centers, one procedure assigns each input point to its nearest center and then recomputes the centers given the point assignment [378].

Scaling k-means to massive data is relatively easy, due to its simple iterative nature. Given a set of cluster centers, each point can independently decide which center is closest to it, and given an assignment of points to clusters, computing the optimum center can be performed by simply averaging the points. Indeed, parallel implementations of k-means are readily available [378].

From a theoretical standpoint, k-means is not a good clustering algorithm in terms of efficiency or quality. Thus, the running time can grow exponentially in the worst case [379, 380] and even though the final solution is locally optimal, it can be very far away from the global optimum (even under repeated random initializations). Nevertheless, in practice, the speed and simplicity of k-means are attractive. Therefore, recent work has focused on improving its initialization procedure performance in terms of quality and convergence [378].

4.1.4 Decision Trees

Decision trees is a popular supervised machine learning method that produces regression or classification models in the form of a tree structure containing decisions as nodes, resulting in a set of leaves containing the solution. Decision trees are suitable to be applied to any data without much effort when compared with algorithms such as neural networks. Trees are built top-down from the root node and involve recursive binary splitting. In neural networks, the initial dataset is partitioned into smaller subsets according to their features, while an associated decision tree is incrementally built. Such a splitting process is driven by a greedy algorithm evaluating the best solution at each of those steps and evaluating the maximum loss reduction from the cost function in order to make a split on features. To regulate the complexity of a given model and increase the performance of a given tree, pruning processes are available. Notwithstanding, decision tree learning does not generally provide the best performance in terms of prediction. Some approaches exist in learning decision forests, including bagging [381], random forests [382] and boosted trees [383].

Tree boosting overcomes the above performance problem by the use of an additive model that iteratively builds decision trees to learn decision forests by applying a greedy algorithm (boosting) on top of a decision tree base learner [383, 384, 385]. Tree boosting is regarded as one the most effective off-the-shelf nonlinear learning methods for a wide range of application problems [384]. It is also highly effective and widely used for achieving state-of-the-art results on many machine learning challenges hosted by the Machine Learning (ML) site Kaggle [386].

Regularized greedy forest is an algorithm that can handle general loss functions with a wider range of applicability. It directly learns decision forests by taking advantage of the tree structure itself, while other methods employ specific loss functions, such as exponential loss function in the case of the Adaboost algorithm [384].

4.1.5 XGBoost

XGBoost is a scalable system that implements gradient tree boosting and the regularized model so as to prevent overfitting, and simplifies the objective function—for parallelization of the regularized greedy forest algorithm [384]. It is suitable for the development of parallel computing solutions applicable to larger datasets or faster training. Besides processors and memory, it uses disk space to handle data that do not fit into the main memory. To enable out-of-core computation, the data are divided into multiple blocks [371]. The system includes cache access patterns, data compression, and sharding. Its performance relies on a tree learning algorithm, which is able to handle sparse data, and on a weighted quantile sketch procedure. This procedure enables handling instance weights in approximate tree learning and is able to solve real-world scale problems using a minimal amount of resources. Besides the penalty from regularizing the objective function, two techniques prevent overfitting: shrinkage, introduced by Friedman [387], and feature subsampling retrieved from random forests to speed up computations. XGBoost works well in practice and has won several ML competitions, such as Kaggle [386], running faster than other popular solutions on a single machine and scaling in distributed or out-of-core settings. It can be easily interpreted, given the tools it provides for finding the important features from the XGBoost model.

4.1.6 Dask

The Dask parallel computing framework leverages the existing Python ecosystem, including relevant libraries such as “numpy” or “pandas”. Dask capabilities are supported by executing graphs to be run by the scheduler component, potentially scaling execution to millions of nodes. Those features are suitable to be applied to out-of-core scenarios (not fitting in memory) on a single machine [372].

Dask is a Python specification representing the computation of directed acyclic graphs of tasks with data dependencies to encode parallel task graph schedules. It extends the easy to adopt NumPy library for leveraging parallel computation over modern hardware. It allows scaling large datasets by using disks that extend the physical memory as out-of-core and parallelize and linearly speedup the code by taking advantage of several cores. The main objective is to parallelize the existing Python software stack without triggering a full rewrite. A Dask cluster includes a central scheduler and several distributed workers. It starts up a XGBoost scheduler and a XGBoost worker within each of the Dask workers sharing the same physical processes and memory spaces.

Dask enables parallel and out-of-core computation by including collections such as arrays, bags and dataframes. It couples blocked algorithms with dynamic and memory-aware task scheduling to achieve a parallel and out-of-core popular NumPy clone [372]. Sharing distributed processes with multiple systems allows usaging of specialized services easily and avoiding large monolithic frameworks.

Dask is often compared with other distributed ML libraries, such as H2O [388] or Spark’s Machine Learning Library (MLLib) [389]. XGBoost is available in Dask to provide users with a fully featured and efficient solution. The Dask parallel computing approach can handle problems that are more complicated than the map-reduce problem at a lower cost and complexity when compared to solutions such as MLLib, given that most of the problems can be resolved in a single machine. Any function is able to be parallelized by the use of delayed function decorators.

Additionally, Dask is substantially lightweight when compared to Spark.

4.2 Proposed Approach

Motivated by the related work, an integrated method with filtering mechanisms to improve detection accuracy and efficiency in scenarios involving large amounts of logs is proposed. This method is supported by the k-means clustering and the gradient tree boosting classification algorithms, as implemented by the XGBoost system. To overcome the limitations of existing anomaly detection methods that spend a significant amount of time building the models for the whole dataset, three different tools for improving detection accuracy and efficiency were built.

This section starts with a formal presentation of the algorithm of the model, followed by a discussion of the three compounding tools used for implementing the proposed approach.

4.2.1 Description of the Algorithm

The proposed approach is formalized in Algorithm 1, which describes how to combine k-means and XGBoost. The algorithm is implemented as a function that takes as input a set of events E and returns the identification of the anomaly $anomalycluster$, the classified events $ypred^1$, total classified events $totalevents$ and the total of those events classified as anomalies $totalanomalies$.

It starts by initializing the cluster S and activating the client connection C to the cluster S. Then the distributed array G is prepared from the received events in E. The next step is to initialize the k-means model Km for binary classification in the cluster $(k = 2)$ from the distributed array G to separate events into two distinct clusters in Y. Then, the XGBoost model X is initialized with the previously predicted events Y being provided as an input for training in the cluster through the use of the client connection C. The final prediction $ypred$ is achieved from the XGBoost model X. In the next stage, each of those predictions $(i \in ypred)$ is classified according to the cluster they belong to in $ypred^1$. Such a classification will be determined by evaluating the total number of events in clusters $k1$ and $k2$, so as to decide which corresponds to the anomaly cluster. To that aim, 0.5 was considered as the threshold to classify events as belonging to clusters 1 or 2 $(ypred_i > 0.5)$.

After all events have been classified, the cluster including the fewer number of events $(k1 > k2)$ will correspond to the anomaly cluster, and such decision will stored

Algorithm 1 Proposed Algorithm

INPUT: E, Event Set

```
$S \leftarrow \textsc{Cluster}()$
$C \leftarrow \textsc{Client}(S)$
$G \leftarrow \textsc{DistributedArray}(E)$
$k \leftarrow 2$
$km \leftarrow \textsc{KMeans}(C,k)$
$km.\textsc{Train}(G)$
$Y \leftarrow km.\textsc{Predict}(G)$
$X \leftarrow \textsc{XGBoost}(X)$
$X.\textsc{train}(Y,Y)$
$ypred \leftarrow X.\textsc{predict}(G)$
for all $i \in ypred$ do
    if $ypred_i > 0.5$ then
        $ypred_i^1 \leftarrow 1$
        $k2 \leftarrow k2 + 1$
    else
        $ypred_i^1 \leftarrow 0$
        $k1 \leftarrow k1 + 1$
    end if
end for
if $k1 > k2$ then
    $anomalycluster \leftarrow 1$
    $totalanomalies \leftarrow k2$
else
    $anomalycluster \leftarrow 0$
    $totalanomalies \leftarrow k1$
end if
$totalevents \leftarrow k1 + k2$
```

OUTPUT: $ypred^1$, Cluster Predictions
OUTPUT: *anomalycluster*, Identification of the anomaly cluster
OUTPUT: *totalevents*, Total number of events
OUTPUT: *totalanomalies*, Total number o anomalies

in *anomalycluster*.

4.2.2 Tools

The framework encompasses three tools that may be independently combined in a cooperative way for normalizing raw data and for producing a model able to achieve evidence for forensic and compliance auditing analysis. The "fca_normalization" tool is used to normalize the raw data, “fca_model” produces the model and “fca_analysis” provides the pieces of evidence for forensic and compliance auditing analysis.

The normalization tool takes as input Secure HyperText Transport Protocol (HTTP) raw data logs and normalizes data into a new file. Optionally, the encoded features may also be specified. In case encoding is not provided or in the case of missing feature values, the tool automatically applies an encoding label. The tool can be invoked, for example, by using the following command:

```
python fca_normalization
-in NASA_access_log_Jul95
-in_encoding in_encoding.data
-out logs_NASA.csv
-out_encoding out_encoding.data
```

In this example, “fca_normalization” receives the raw HTTP access log data file “NASA_access_log_Jul95” along with the optional encoding file “encoding.data”. The output normalized file is saved as “logs_NASA.csv”. Finally, the tool optionally defines the encoding table in the "out_encoding.data" file.

The modeling tool takes as input the previously normalized data and builds the XGBoost classification model by making use of the gradient tree boosting algorithm after applying the k-means clustering algorithm. In the example invocation provided next, the input file “logs_NASA.csv” contains the HTTP raw log data and the output model is saved as “fca_xgboost.pkl”.

```
python fca_model
-in logs_NASA.csv
-out fca_xgboost.pkl
```

The forensic and compliance auditing analysis tool takes as input the model and the normalized events in order to identify the anomalies. In the invocation example provided next, the input model in read from ‘fca_xgboost.pkl’, and the normalized data is read from ‘logs_NASA.csv’. The final output containing the anomaly events is saved on ‘outlier_events.csv’.

```
python fca_analysis
-in_model fca_xgboost.pkl
-in_data logs_NASA.csv
-out outlier_events.csv
```

Table 4.1 summarizes the inputs and outputs for each tool.

Table 4.1: Forensics and Compliance Auditing (FCA) Tools

Tool	**Input**	**Output**
Normalization	HTTP raw logs data, encoding	Normalized data, encoding
Modelling	Normalized data	Model
Analysis	Model, normalized data	Anomaly events

4.3 Discussion and Evaluation

This section addresses the validation of the proposed framework. It starts with a discussion about feature extraction, followed by a description of the initial application of the k-means clustering algorithm for dividing the dataset into two different clusters, based on the extracted features. Finally, it is demonstrated how to use the previous clustered data for training a scalable gradient tree boosting implemented by the XGBoost system.

For the sake of readability throughout this section, a set of well-known, publicly available datasets [390] were extensively used as a reference. These datasets consist of traces containing two months' worth of all HTTP requests to the NASA Kennedy Space Center WWW server, involving 1,871,988 logged events. This dataset was selected because it is probably the largest log-based dataset publicly available, allowing us to assess the scalability characteristics of the proposed approach.

4.3.1 Feature Extraction and Data Exploration

To capture the characteristics of the system behaviors, 14 features were extracted, containing both numeric and categorical attributes from the raw log records. The original features in the raw HTTP log records are “IP”, “Date”, “Request”, “Response” and “length”. By making use of regular expressions, the most relevant time-related components were extracted from the "date" feature, including “Day”, “Month”, “Year”, “Hour”, “Minute” and “Second”. From the “Request” field, the “operation”, “page” and “method” features were extracted. Then, “Month” names were encoded. Therefore “Year”, “Month” and “Day” were composed in the temporary “date” feature in order to retrieve the day of the week (“weekday”) and “weekend” features. Next, “Request” and other temporary features were removed from the dataset. Finally, categorical features such as “IP”, “page”, “operation”, “method” and “Response” were encoded, and the dataset was saved in a file.

By exploring the dataset a series of first insights can be obtained. Figure 4.1 depicts the covariance of the most representative features, including “length”, “Hour”, "operation", "method" and "Response". This figure shows an interesting covariance between length and other features.

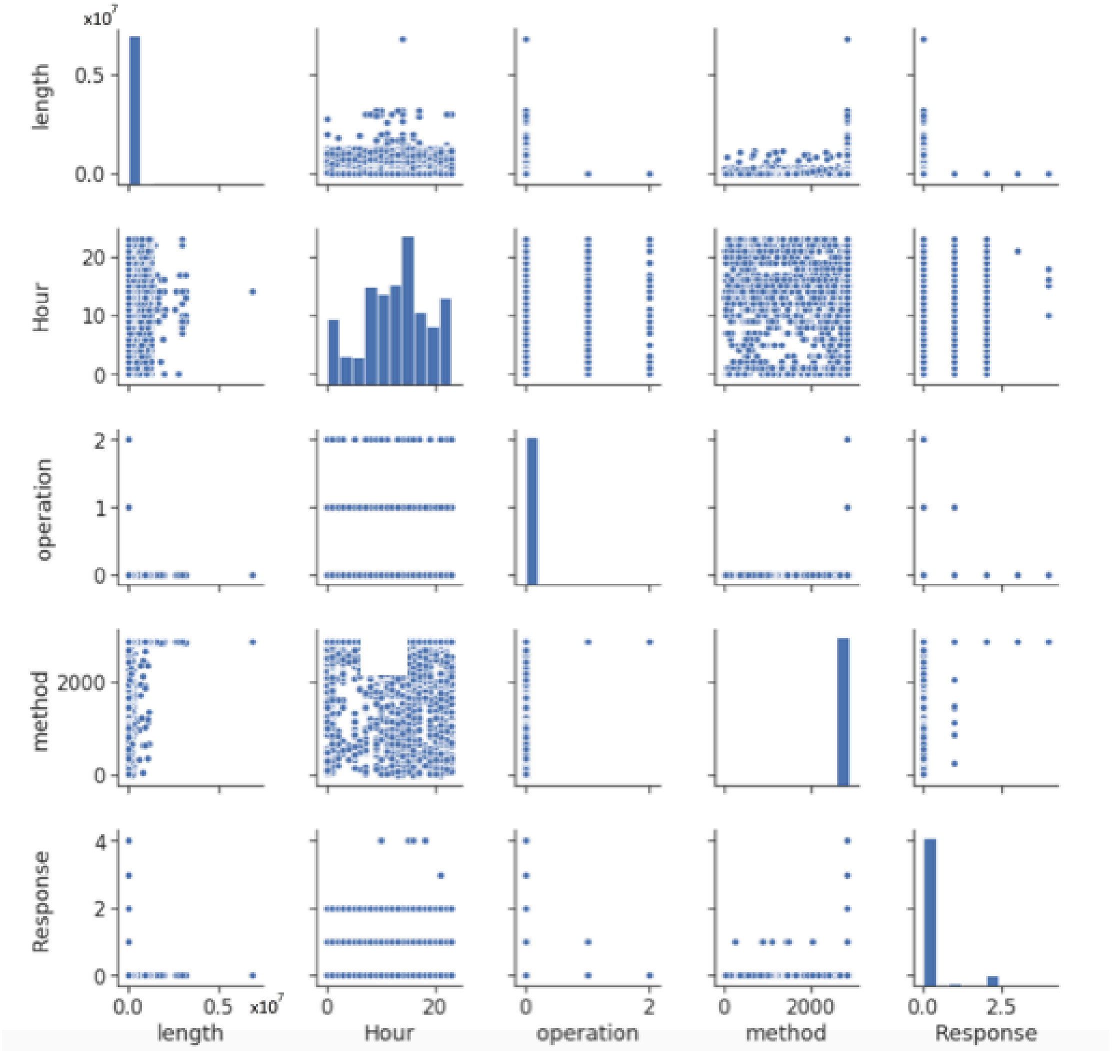


Figure 4.1: Feature covariance

Figure 4.2 provides a three-dimensional analysis of the number of events that occurred along the day (from 0 to 24 h) and along each weekday (0 to 6), where days 5 and 6 correspond to Saturday and Sunday, respectively.

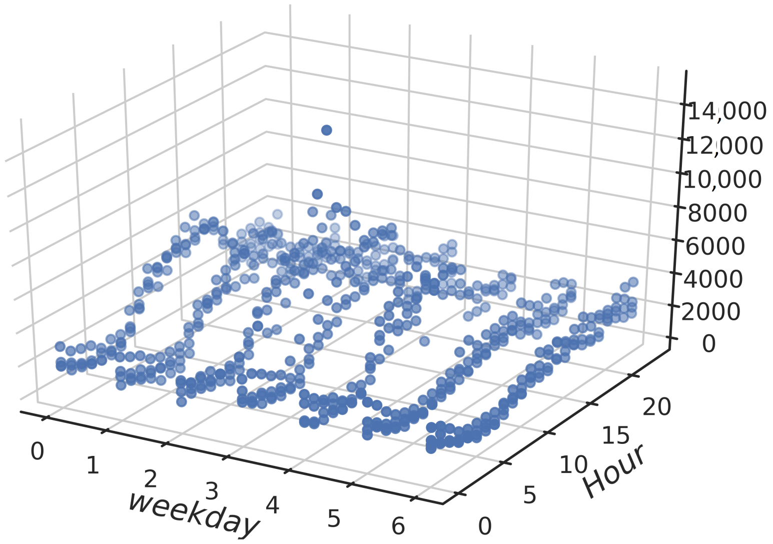


Figure 4.2: HTTP (Hypertext Transfer Protocol) events over time

4.3.2 Clustering

Based on the extracted features, a k-means clustering algorithm was employed for grouping log events into two different clusters. The larger cluster gathers the normal events, while the smaller holds the deviations from normal behavior. Therefore, the latter cluster should correspond to the set grouping the anomaly events. In addition, sparse clusters are possibly caused by anomalous activities, which can be labeled as anomaly candidates for further analysis.

The proposed approach model takes advantage of the initialization k-means|| algorithm (largely inspired by k-means++) to obtain a nearly optimal solution after a logarithmic number of steps. In practice, a constant number of passes suffices [378].

After training this model with 90% of the total number of records and using just the remaining 10% for testing, the model produces a normal cluster containing 185,897 events while the anomaly cluster includes 1301 events, corresponding to 0.06% of the total number of events in the normal cluster. The computed centroids for the two clusters, separating the normal and anomaly events, are the following:

```
[[4.41534608e+04, 0.00000000e+00, 8.12115012e+05, 1.14495884e+01,
0.00000000e+00, 0.00000000e+00, 1.26692042e+01, 2.96762857e+01,
2.94179871e+01, 0.00000000e+00, 1.75010380e+04, 2.84859910e+03,
2.82322741e+00, 2.23245109e-01]

[4.27877328e+04, 1.63161125e-01, 1.53047043e+04, 1.24323538e+01,
0.00000000e+00, 0.00000000e+00, 1.26856431e+01, 2.95910303e+01,
2.94991093e+01, 2.22648078e-03, 1.47567972e+04, 2.84883391e+03,
2.68136168e+00, 1.93607622e-01]]
```

4.3.3 Classification

Classification results from the application of the gradient tree boosting algorithm implemented by the XGBoost system, which is the second and final stage of the proposed model. The resulting tree can be linearized into decision rules, where the outcome is the content of the leaf node, and the conditions along the path form a conjunction in the if clause.

The results of this stage were validated by comparing if the number of events classified as anomalies is equal to the number of events belonging to the anomaly cluster. This condition was verified for 1301 events. The predict function for XGBoost outputs probabilities by default and not actual class labels. To calculate accuracy they were converted to 0 and 1 labels, where a 0.5 probability corresponds to the threshold. XGBoost is able to correctly classify all the test data according to the k-means clustering algorithm. Figure 4.3 depicts the importance of the XGBoost features, according to the F-score metric.

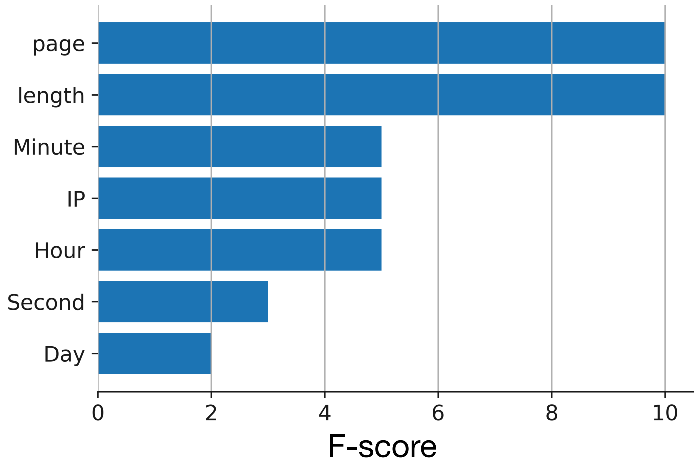


Figure 4.3: Feature importance

This classification model produces a set of rules providing the rationale for generalizing to unseen events, as shown in Figure 4.4. The leaf values depicted in the figure are converted into probabilities by applying the logistic function.

Figure 4.5 depicts the covariance of “length” and “page”, which are the two most

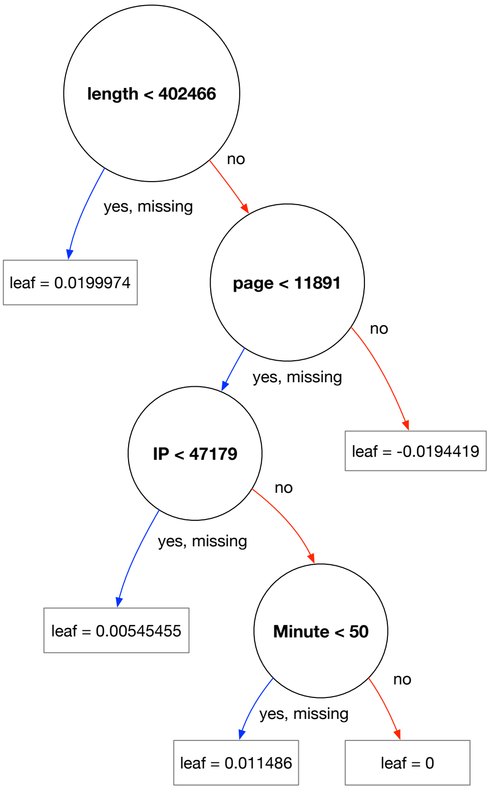


Figure 4.4: Decision tree

important features computed by the final model. The events tagged as anomalous are highlighted in red color.

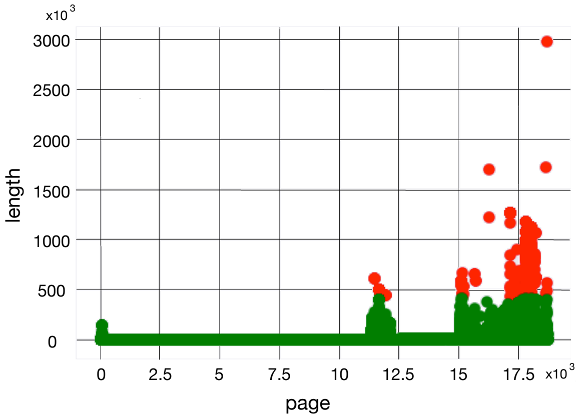


Figure 4.5: Page and length covariance

4.3.4 Parallelization

The proposed approach makes use of the k-means algorithm and the XGBoost system, which are designed to scale in a distributed environment supported by available parallel computing capabilities. This comes in line with a Big Data scenario.

Parallel computing capabilities are provided by the Python “Dask” library. More specifically, the “dataframe” component is able to manage out-of-core datasets along the execution pipeline, since the features are extracted until the clustering and classification models are implemented. Figure 4.6 provides an example of the kind of graphs Dask is able to produce when reading and splitting a dataset. The Dask libraries “dask_ml” and “dask_xgboost” provide the implementation of popular machine learning algorithms, such as k-means and XGBoost, which support the framework models.

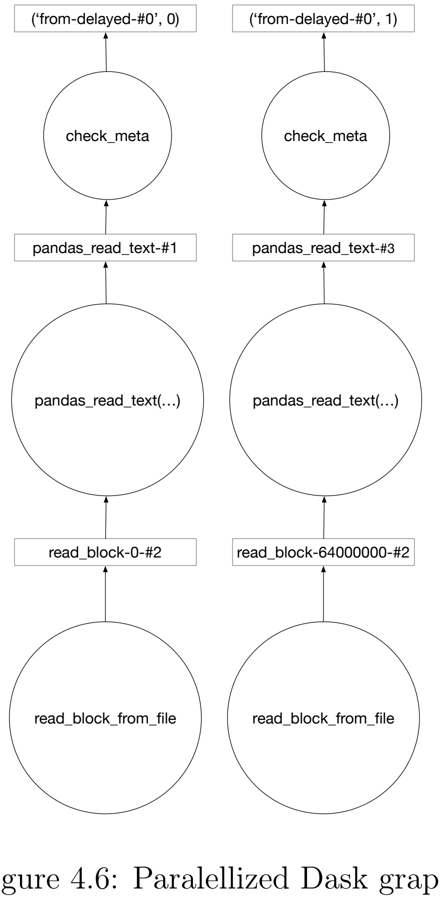


Figure 4.6: Paralellized Dask graphs

Experiments involved a simple cluster formed by just two workers in a single node with two cores while the total available memory was 13.66 GB.

To study the framework model’s ability to scale in order to cope with large datasets in a reasonable time, two experiments were performed using the parsed

NASA HTTP logs dataset. Due to constrained laboratory resources, those experiments were limited to the use of two cores in a single node. As a setup configuration, the Dask chunk size was set to 50,000 events. The model's ability to scale was assessed by comparing its performance under different configurations. To determine the model performance, running time (in milliseconds) was considered throughout the training and predict steps for both k-means and XGBoost stages in accordance with the model topology.

The first experiment aimed at determining the parallel approach performance, compared with the sequential approach—considering non-Dask sklearn as the sequential approach and Dask as the parallel approach. As a setup configuration, the Dask framework included a single worker and two threads. The running time was measured along the four steps previously defined for the two stages. Those sequential steps include the train (1) and predict (2) steps for the k-means stage, followed by the train (3) and predict (4) steps for XGBoost stage. The running time for each framework, along those running steps, is provided in Figure 4.7. The achieved results show that the Dask framework outperforms the non-Dask sklearn framework, especially in the case of the training steps.

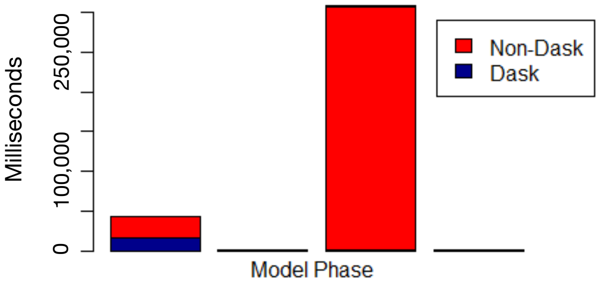


Figure 4.7: Sequential (non-Dask) vs. parallel (Dask) comparison

The second experiment evaluated the parallelization capability of the Dask framework under different configurations, such as the number of workers and threads per worker, by measuring the aggregated running time along the topological steps. Figure 4.8 compares the performance for one and two running workers, while increasing the number of threads per worker from one to ten. Measurements showed that one worker outperforms two workers. Increasing the number of workers did not improve performance, while increasing the number of threads contributes to improved performance until a given threshold is reached. Finally, it was also possible to depict higher performance running over an even number of threads in comparison to the odd ones—due to the less optimal parallelization gains that occur when splitting an odd number of threads into two cores.

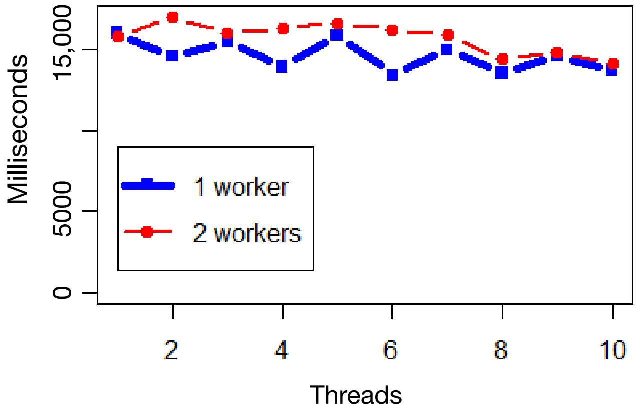


Figure 4.8: Dask parallel comparison

4.3.5 Discussion

The presented framework method relies on two stages. The clustering model is the output of the first stage and serves as the input for the classification stage. Therefore, this approach allows starting from initial unlabelled data for obtaining interpretable meaningful rules with the rationale for classifying unseen events. Those rules are simple to understand, interpret and visualize, requiring relatively little data preparation effort. Additionally, the described algorithms can easily handle heterogeneous data containing different features produced by different sources. Although the initial nature of the problem what was addressed is not about classification, this approach may be adapted to different scenarios where labeled data are not available. This way, it becomes possible to convert an unsupervised into a supervised learning scenario and take advantage of the use of classification algorithms.

The decision to select the k-means algorithm and XGBoost system, both supported by the Dask library for parallel processing, was driven by requirements in terms of scaling and interpretability when working with limited resources. This decision enabled the application of this framework to larger datasets in order to highlight the anomalous events. Given the inherent nature of the problem being addressed through the use of the unsupervised learning approach, it is not trivial to evaluate the framework model's accuracy in the scope of this chapter. An alternative option would be to compare the achieved results with those provided in the existing literature. However, to the best of the author's knowledge, there are no anomaly detection research works addressing the NASA HTTP logs.

The obtained results highlight the obviously normal events in highly coherent clusters, with a minor subset of events being classified as anomalies for further forensic and compliance auditing analysis. The model interpretability is indirectly validated by the produced decision rule set already provided in Figure 4.4, which implicitly shows how the model identifies classes. Figure 4.7 also shows the performance of the parallel approach compared with the sequential approach, and Figure 4.8 highlights the parallelization capabilities of the Dask library in processing out-of-core datasets.

Designing the framework with independent tools makes it possible to reuse them over different scenarios. For example, the same modeling tool can be combined with a different normalization tool for processing a different data source. Additionally, these framework tools can be applied to the context of the aforementioned ATENA project in order to identify anomaly events from massive logs. This approach can be independently applied to different datasets in a first stage, allowing to correlate them as heterogeneous sources in a second stage.

The achieved results demonstrate the capability of the proposed method in terms of finding a set of interpretable rules that can parallelized and applied in scale.

4.4 Summary

In this chapter a parallel computing approach for identifying anomaly events in massive log files was proposed. In its first stage, a k-means algorithm is used to separate anomalies from normal events. In the second stage, a gradient tree boosting classification model, implemented using the XGBoost system, produces the interpretable meaningful rationale rule set for generalizing its application to a massive number of unseen events. This approach is suitable for application in the context of out-of-core datasets in cases where log sources are so massive that it becomes impossible to use more traditional approaches.

The proposed method was presented, and the achieved results demonstrated its applicability to producing simple and interpretable rules for highlighting anomalies in application log data to scale and in a distributed environment. Such an approach makes it suitable to be applied in the fields of forensics and audit compliance.

Chapter 5

Using a Federated Ontology for Handling Heterogeneous Data Sources

Contents

THE benefits of enlarging the scope of information sources for Security Information and Event Management (SIEM) applications, forensic analysis, and compliance audit operations are rather evident, since the result would enable more powerful, all-inclusive approaches to cybersecurity awareness. For example, monitoring of abnormal activity within the Industrial Automation and Control Systems (IACS) specific domain might be leveraged by the correlation of different data sources, such as mail filtering logs (monitoring phishing and malware attacks, which target the employees of the Critical Infrastructure (CI)) and information about employee functions residing in Human Resources information systems. Another example would be the correlation of data from physical access control systems and staff check clocks with activity logs of IACS operators. In general, this strategy of associating core security information already fed into SIEM systems with peripheral-awareness data would result in richer security analysis processes that enable the detection of inconsistencies, malpractices, and intrusion clues, which would otherwise go unnoticed.

However, achieving tight integration of all those peripheral data sources into the already-existing SIEM frameworks is costly and often impractical. This would require considerable investments in data conversion and adaptation to the SIEM data flows. Moreover, the maintenance costs would also be considerable, since even minor adjustments on the corporate information systems would require explicit adaptations on the SIEM side.

A more plausible option is, therefore, the adoption of loosely coupled integration strategies, such as resorting to Semantic Web approaches for automating the processing and interpretation of large amounts of information available from both local databases and Internet repositories. This reasoning process, applied over a large quantity of available data with knowledge inferred from a combination of axioms, properties, and rules (with different levels of hierarchies or categorisations and deriving conclusions, for example) can be explicitly expressed by ontologies.

It should be noted that most data are still not directly available in Semantic Web formats. This is the case with data maintained in Relational Databases (RDBs)). Nonetheless, mapping data from RDB to Semantic-Web-enabled Resource Description Frameworks (Resource Description Frameworks (RDFs)) has been the focus of a large body of previous research, leading to the implementation of many generic mapping tools and their applications, on several specific domains. Those tools are natural candidates to be adapted to the field of Critical Infrastructure Protection (CIP) so that security-related ontology data currently stored in heterogeneous databases can be taken into consideration – despite the considerable challenges involved, such as the migration from existent systems to the semantic level [391].

A detailed discussion of the main motivations and driving research efforts in mapping RDB to RDF can be found in [392]. Although most models can perform inference from native ontology data stores, data still reside mostly in RDBs, which are broadly used within organisations. Moreover, the growing number of datasets published on the Web brings opportunities for extensive data availability and challenges related to the process of querying data in a semantically heterogeneous and distributed environment. The structured query approach fails on the linked data because the Web's scale makes it impractical for users to know in advance the structure of datasets [393].

This chapter introduces an approach considering inference capabilities from Semantic Web, supported by common schemas, for creating a set of independent databases, each deployed with its own domain-specific schema. This kind of reasoning is suitable for application in the context of CIP, and, therefore, it can leverage current SIEM capabilities –mainly in what relates to forensic and compliance audit processes, but also for intrusion detection purposes. This large amount of living heterogeneous data that still resides in the organisational RDBs will, in this way, become available to the Critical Infrastructure's SIEM and enable new, valuable insights into available configuration and monitoring data.

After discussing some of the key previous work and trends in the area, this chapter takes a practical approach by presenting the implementation of a federated query architecture for retrieving a set of compliance auditing rules that might be useful, for instance, in assessing CI security levels. To leverage inference capabilities, it maps the living data currently available on RDBs into RDFs formats. In this way, it can substantially enlarge the data available to the SIEM by taking advantage of the large amount of heterogeneous data of production-RDB systems. Such an approach provides an abstraction mechanism for keeping data consumers away from low-level details while leveraging the security concerns of the underlying infrastructures by hiding the internal deployment aspects, such as the identification of the involved machines and their RDB schemas.

The ontology-based approach of this work considers the available information currently stored in RDB and, as its main goal, makes it accessible through simple interfaces that collect queried data from multiple natively different data repositories within the organisation. Each available RDB maintains different information instances, deployed on specific schemas and technologies. Such an approach is suitable for combining data from two different worlds, such as the case of RDB and Semantic Web data, which is natively maintained in RDF stores and made available through an interface layer encapsulating the details of the gathering process to retrieve the data from multiple RDBs.

The work hereby presented has already been published in [394][395], and the chapter follows a similar structure. Starting with Section 5.1, which provides a background discussion focused on the domain problem and related work, it will next present an analysis of the applicability of ontology data in the context of CIP (in Section 5.2), followed by Section 5.3, where the proposed architecture will be presented, as well as its implementation. Finally, this chapter concludes with a final summary, in Section 5.4

5.1 Background

This section briefly introduces the reader to the key concepts and tools used in the proposed data integration approach: RDF; RDB, and RDF mapping; SPARQL; Direct Mapping of Relational Data to RDF; and the D2RQ platform.

5.1.1 Resource Description Framework (RDF)

An ontology is a formal specification of concepts in a domain of discourse, which includes classes and properties [396]. An ontology, together with a set of individual instances of classes, constitutes a knowledge base [397].

RDF [398] is a language that can be used to encode knowledge into web pages to make them understandable for electronic agents searching for information. This is one of the main goals for using ontologies [399, 396]. RDF aims at representing information that may be used for inference purposes over the Web. The RDF syntax core structure consists of a set of triples with a subject, a predicate, and an object. A set of triples is called an RDF graph. An RDF graph may be visualised as a directed-arc diagram, in which each triple is represented as a node-arc-node link. RDF is a data format based on a Web-scalable architecture for identification and interpretation of terms [400].

5.1.2 Mapping from RDF to RDB

As already mentioned, the mapping of large amounts of data from RDB to RDF has been the focus of intense research work in multiple domains and has led to the implementation of a set of generic mapping tools, as well as domain specific applications. RDF has provided an integration platform for data gathered from multiple sources, primarily from RDB. This is one of the main motivations driving research efforts (using various approaches) on mapping RDB to RDF [401].

SPARQL [402] can be used to express queries across diverse data sources, whether for data natively stored as RDF or for data viewed as RDF via some sort of middleware. SPARQL is a World Wide Web Consortium (W3C) recommendation for querying multiple RDF graphs. The SPARQL specifications define the syntax and semantics to proceed with queries across diverse natively stored RDF data sources. Using the latest stable release (SPARQL 1.1), SPARQL federated queries allow merging multiple results retrieved from multiple RDF sources. The syntax and semantics of SPARQL 1.1 Federated Query extension allow distributed queries over different SPARQL endpoints. Moreover, the SERVICE clause extends SPARQL 1.1 to support queries that merge data distributed across the Web. A single query is, therefore, able to return related data (for example, contacts to be applied to user John Doe) from multiple distinct SPARQL endpoints.

An important feature of RDF and SPARQL is that they can use different datasets from different locations, federating them together. They offer a middleware, which can use multiple data sources as if they were one. Moreover, it is simple to add and remove data sources. This feature significantly reduces the development costs as compared to typical data warehouse projects [403].

Listing 5.1 provides an example of a query through different SPARQL 1.1 endpoints. The query returns John's contacts from two distinct SPARQL endpoints: www.site1.com and www.site2.com.

Listing 5.1: Query example through different SPARQL 1.1 endpoints

```
SELECT ?contact1 WHERE {
    SERVICE <http://www.site1.com/sparql>{
        SELECT ?contact1 WHERE {
              ?me foaf:nick "John".
              ?me foaf:knows ?f .
              ?f foaf:name ?contact1
        }
    }
    SERVICE <http://www. site2.com/sparql>{
        SELECT ?contact2  WHERE {
              ?me foaf:nick "␣John␣".
              ?me foaf:knows ?f .
              ?f foaf:name ?contact2
        }
    }
    FILTER (?contact1 = ?contact2)
}
```

5.1.3 Direct mapping of relational data to RDF

Relational databases allow the use of tools, such as Structured Query Language (SQL), for accessing and managing the databases. Several strategies already exist to map relational data to RDF. Typically, the goal is to describe the RDB contents using an RDF graph, allowing queries submitted to the RDF schema to indirectly retrieve the data stored in relational databases. A direct mapping process enables a simple transformation and can be used for materialising RDF graphs or for defining virtual graphs, which can be queried via SPARQL or traversed by an RDF graph Application Programming Interface (API). A mapping document is an RDF document containing triples maps with instructions on how to convert relational database content into RDF graphs.

5.1.4 The D2RQ platform

The D2RQ (Data to RDF Query) Platform [404] allows users to access relational databases as virtual, read-only RDF graphs while automatically producing the corresponding mappings. It allows users to create customised mappings from RDB through an integrated environment with multiple options for accessing relational data, including RDF dumps, Jena and Sesame API based access, and SPARQL endpoints on D2RQ Server [405]. It offers RDF-based access to the content of RDB, without requiring its replication into RDF stores. D2RQ, therefore, allows querying nonRDF databases using SPARQL or accessing contents of databases over the Web. It also allows the creation of custom content dumps from relational databases into RDF stores.

The D2RQ Platform includes components such as a Mapping Language, an Engine, and a D2R (Data to RDF) Server. The D2RQ Engine is a plug-in for the Jena Semantic Web toolkit, which uses mappings for rewriting the Jena API calls to SQL queries against the database and for redirecting query results up to the higher layers of the framework. The D2R Server is an HTTP server which provides linked data views, HTML views for debugging, and a SPARQL protocol endpoint providing an interface to query the database. The D2RQ platform supports databases such as MySQL, SQL Server, Oracle, PostgreSQL, HSQLDB, and InterbaseFirebird. Some limitations of D2RQ include the integration of multiple databases or other data sources and its read-only nature: it lacks Create, Read, Update, and Delete (CRUD) operations. Finally, it does not support inference mechanisms and does not include named graphs [404].

The D2RQ Mapping Language enables defining relationships between RDB schemas and RDF schema vocabularies (classes and properties) or Web Ontology Language (OWL) ontologies written in Turtle syntax [406]. The mapping properties define a virtual RDF graph, which contains information from the database schema. The mapping process between D2RQ and RDB entities includes the RDF class node to RDB tables and RDF predicates to RDB column names [404].

The same D2RQ server can be configured to access multiple databases. Therefore, a single SPARQL query can request data from multiple databases at once, which is not possible with a standard SQL query.

5.2 Applicability of Ontology Data for CIP

Current approaches on the use of ontologies in the context of CIP are mostly related to the assessment of interdependencies between CIs, such as the works of Castorini et al. [407] and Blackwell et al. [408]. Similarly, an ontology for classifying vulnerabilities in the scope of decision support tools has been proposed by Choraś et al. [409].

Other approaches worth mentioning include SPLENDID, DARQ, SemaPlorer, and FedX. SPLENDID [410] is a query optimisation strategy for federating SPARQL endpoints based on statistical data. DARQ [411] provides transparent query access to multiple SPARQL services using one single RDF graph, even when data has a distributed nature and is spread over the Web. This approach includes a service description language that enables a query engine to decompose a query into subqueries, where each of them can be answered by an individual service. SemaPlorer [412] also provides a federated query architecture allowing it to interactively explore and visualise semantically heterogeneous distributed semantic datasets in real time, through a conceptual layer on top of Amazon's Elastic Computing Cloud (EC2).

FedX [413] proposes novel joint processing and grouping techniques for minimising the number of remote requests. It also develops a practical framework that enables efficient SPARQL queries supported by federation layers for efficient query processing on heterogeneous distributed Linked Open Data sources.

Beyond D2RQ, other RDF middleware applications exist, such as TopQuadrant's TopBraid Live, OpenLink Software's Virtuoso Sponger, and Triplr project. These offer dynamic creation and integration. They also allow users to merge several RDF triples in a single SPARQL endpoint from sources such as relational databases, spreadsheets, HTML documents, and other formats.

As already mentioned, one possible application of ontology data in this scope is the use of heterogeneous sources available in organisational RDBs for leveraging inference capabilities. This application is especially interesting in the specific areas of forensic analysis and compliance audit processes, which, by nature, need to be supported by substantial amounts of heterogeneous data.

A possible practical application of this approach, in the scope of forensic analysis and compliance audit processes, consists of the collection and mapping into Semantic Web rules of the data residing in the multiple and heterogeneous relational databases of the CI organisation - so they can be combined with the knowledge already available at the SIEM systems. We have explored this path in the scope of the H2020 ATENA research project [414, 63] and our Forensics and Compliance Auditing (FCA) framework, so that peripheral data sources, processed through domain-specific business rules, may also feed the FCA analytics layer.

5.3 Proposed Approach to the Use of Ontology Data

This section describes the proposed approach to the use of ontology data in the context of CIP applications. First, the proposed reference architecture is introduced, followed by a discussion of technical aspects and implementation details. In a simplified view, the proposed solution consists of a web service that can receive several SPARQL requests from data consumers (such as the forensics and compliance auditing tools mentioned in the previous sections). Afterwards, each one of those requests is forwarded into different databases deployed using different schemas.

5.3.1 Proposed Reference Architecture

The proposed reference architecture, depicted in Figure 5.1, consists of a set of components such as a federated layer, mapping brokers, and databases. Several data consumers (clients) may send distinct sets of SPARQL queries to the federated interface layer, which delivers each query to all the brokers. The broker's main role is to transform the incoming SPARQL queries into native relational database queries. Through an inverse flow, the broker retrieves the data subset from the database to be gathered into a full data set at the federated layer which is then forwarded to the involved client(s).

Although this reference architecture may suggest its applicability to the context of federated database queries, it may be extended to use different kind of data sources, such as logs or Lightweight Directory Access Protocol (LDAP) distributed directory information services (among others) in order to provide compliance audit and forensic capabilities that can be applied to the context of our FCA framework.

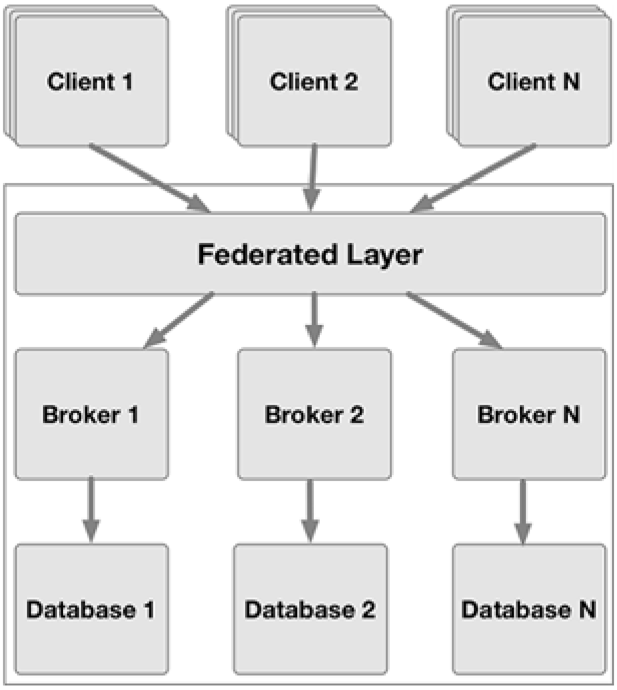


Figure 5.1: Proposed reference architecture

5.3.2 Use-case scenario

Next, a simple compliance audit scenario is presented, to demonstrate the applicability of this approach for evaluating unauthorised accesses to the assets of an international company.

The challenge is to build a common schema for the management of human and asset resources spread over different platforms, because of specific requirements imposed by national governments. A single interface, capable of answering queries merging all the data in the organisation in a single dataset, should be provided. Such an approach would help overcome barriers by approaching different native data sources spread across different locations in an organisation.

5.3.3 Implementation Aspects

This implementation starts by modelling a simple ontology for the FCA processes, which encompasses the norms, policies, and legal or regulatory guidelines that are being applied. The ontology will allow users to infer new knowledge (for example, to identify possible unauthorised or incompatible access to the assets of a large organisation). This example implements a federated query web service for evaluating whether employees have the required roles when they access those assets. An intermediary layer translates the requests arrived to the web service into queries for the internal schemas of the involved databases.

The interface layer is implemented as a web service, while the mapping brokers are implemented as D2R Server endpoints. Each endpoint is assigned to different relational databases. Figure 5.2 provides an overview of the implementation of the described architecture, depicting how requests flow from a submitted query to the web service, which implements a federated query solution to dispatch the incoming requests to the indexed list of database servers – with each of them mapped by a specific D2RQ component. For sake of simplicity, the figure includes just two different databases with different schemas (a Microsoft MSSQL database and a MySQL database), but there are no limits to the number or type of involved databases.

The use case hereby described involves a client requesting the contents of the 'Roles' database entity. The objective is to gather and combine – without requiring the end user to be aware of low-level details – information dispersed across different tables and different databases which use different schemas. After the request query to retrieve the existing contents from the 'Rules' entity has reached the database instances, each delivers its contents to a SPARQL endpoint through a D2R server assigned to each involved database. The D2RQ Mapping Language is used for the mapping process. This central web service allows clients to directly query existing

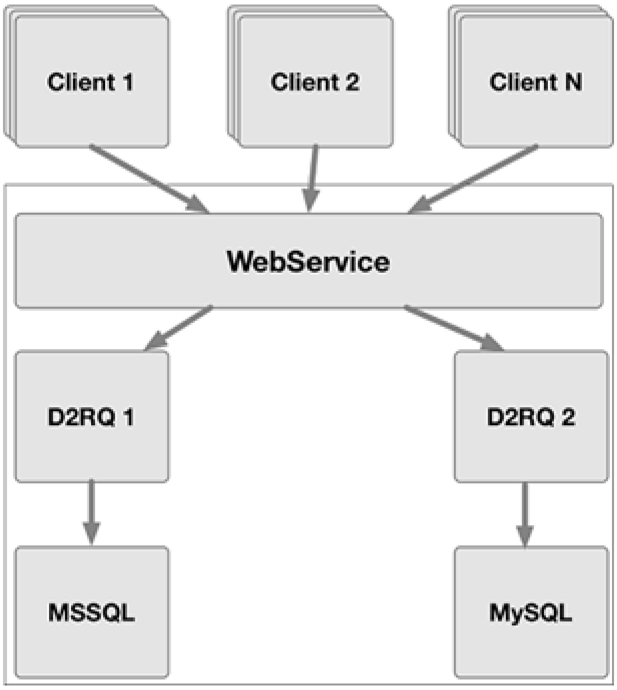


Figure 5.2: Architecture implementation

entities, to retrieve available content from each existing database, and to merge and deliver them to the querying clients.

Required tools and technologies include Visual Studio as development environment, C# as programming language, ASPX.NET for implementing the web service, classic RDBs such as MSSQL and MySQL, and the RDF and SPARQL languages describing their semantics.

Next, we discuss some details for each step involved in the implementation and deployment of this specific use case. First, a simple ontology is presented. Next, some relevant implementation steps are discussed, such as deploying the database server, generating mapping, configuring the mapping between the database server and the ontology, activating D2R servers with the corresponding mappings, and describing the web service.

Create the Ontology: In this step, a simple ontology is explored in the domain of compliance audit to support the previously presented use-case scenario, which has the main purpose of answering the following question: 'Who is able to access the assets, for maintenance purposes, in a large company spread out through different countries and businesses?'

The ontology, built within Protégé, includes classes for 'Asset', 'Employee', 'Organization', and 'Role'. The corresponding instances are 'Computer', 'John' and

'Francis', 'PowerPlantA', and 'MaintainsIT'. The ontology does not include any hierarchy of concepts. Table 5.1, summarises the relationship among class instances, their types and property assertions.

Instance	Type	Property Assertions
John	Employee	isEmployedBy:PowerPlantA Number: '1002' Name: 'John'
Francis	Employee	isEmployedBy: PowerPlantA hasRole:MaintainsIT Number: '1001' Name: 'Francis'
MaintainsIT	Role	maintains:Computer isMaintainedBy: Francis Name: 'Francis
PowerPlantA	Organization	hasEmployees:John hasEmployees: Francis hasAssets: Computer Name: 'PowerPlantA'
Computer	Asset	isRoledBy: Francis belongsTo:PowerPlantA Number: '10000001' Name: 'DELL'

Table 5.1: Classes instances

'John' and 'Francis' are instances of 'Employee'. Both have the property 'isEmployedBy' assigned with the value 'PowerPlantA'. The employee is assigned roles granting the access to the assets, enabling the building of a query to assess the regulatory rules and policies. It also has as an inverse property 'hasRole' as 'MaintainsIT'. Additionally, they have data properties '1' and '2' for the 'Number', and 'Francis' and 'John' for 'Name'. Notwithstanding, the difference between 'Francis' and 'John' instances is that the 'Francis' does not include the property 'hasRole' as 'MaintainsIT'. Therefore, they will be considered two employees for the organisation, but just one of them is able to maintain the assets. 'PowerPlantA' is an instance of the 'Organization' type and includes the property 'hasEmployees' for 'Francis' and 'John' instances. Therefore, this organisation has two employees. 'Computer' is an instance of the 'Asset' type and its properties are 'isRoledBy' of the 'MaintainsIT' instance, whose value is 'Francis' and which includes a 'Number' and a 'Name'.

Listing 5.2 provides the full contents of the above ontology, in turtle language, located at the 'data.ttl' file.

Listing 5.2: Ontology definition

```
#filename: data.ttl
@prefix FCA: <http://www.semanticweb.org/FCA#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix owl: <http://www.w3.org/2002/07/owl#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>

#################################################################
#    Object Properties
#################################################################

###  http://www.semanticweb.org/FCA#belongsTo
FCA:belongsTo rdf:type owl:ObjectProperty ;
owl:inverseOf FCA:hasAssets ;
rdfs:domain FCA:Asset ;
rdfs:range FCA:Organization .

###  http://www.semanticweb.org/FCA#hasAssets
FCA:hasAssets rdf:type owl:ObjectProperty ;
rdfs:domain FCA:Organization ;
rdfs:range FCA:Asset .

###  http://www.semanticweb.org/FCA#hasEmployees
FCA:hasEmployees rdf:type owl:ObjectProperty ;
owl:inverseOf FCA:isEmployedBy ;
rdfs:domain FCA:Organization ;
rdfs:range FCA:Employee .

###  http://www.semanticweb.org/FCA#hasRole
FCA:hasRole rdf:type owl:ObjectProperty ;
owl:inverseOf FCA:isRoledBy ;
rdfs:domain FCA:Employee ;
rdfs:range FCA:Role .

###  http://www.semanticweb.org/FCA#isEmployedBy
FCA:isEmployedBy rdf:type owl:ObjectProperty ;
rdfs:domain FCA:Employee .

###  http://www.semanticweb.org/FCA#isRoledBy
FCA:isRoledBy rdf:type owl:ObjectProperty ;
owl:inverseOf FCA:isRoledBy ;
rdfs:domain FCA:Role ;
```

```
rdfs:range FCA:Employee .

#################################################################
#    Data properties
#################################################################

###  http://www.semanticweb.org/FCA#Name
FCA:Name rdf:type owl:DatatypeProperty ;
rdfs:domain FCA:Asset ,
FCA:Employee ,
FCA:Organization ,
FCA:Role .

###  http://www.semanticweb.org/FCA#Number
FCA:Number rdf:type owl:DatatypeProperty ;
rdfs:domain FCA:Asset .

#################################################################
#    Classes
#################################################################

###  http://www.semanticweb.org/FCA#Asset
FCA:Asset rdf:type owl:Class .

###  http://www.semanticweb.org/FCA#Employee
FCA:Employee rdf:type owl:Class .

###  http://www.semanticweb.org/FCA#Organization
FCA:Organization rdf:type owl:Class .

###  http://www.semanticweb.org/FCA#Role
FCA:Role rdf:type owl:Class .

#################################################################
#    Individuals
#################################################################

###  http://www.semanticweb.org/FCA#Computer
FCA:Computer rdf:type owl:NamedIndividual ,
FCA:Asset ;
FCA:belongsTo FCA:PowerPlantA ;
FCA:Name "DELL"^^xsd:string ;
FCA:Number "10000001"^^xsd:int .
```

```
###  http://www.semanticweb.org/FCA#Francis
FCA:Francis rdf:type owl:NamedIndividual ,
FCA:Employee ;
FCA:hasRole FCA:MaintainsIT ;
FCA:isEmployedBy FCA:PowerPlantA ;
FCA:Name "Francis"^^xsd:string ;
FCA:Number "1001"^^xsd:int .

###  http://www.semanticweb.org/FCA#John
FCA:John rdf:type owl:NamedIndividual ,
FCA:Employee ;
FCA:isEmployedBy FCA:PowerPlantA ;
FCA:Name "John"^^xsd:string ;
FCA:Number "1002"^^xsd:int .

###  http://www.semanticweb.org/FCA#MaintainsIT
FCA:MaintainsIT rdf:type owl:NamedIndividual ,
FCA:Role ;
FCA:isRoledBy FCA:Francis ;
FCA:Name "MaintainsIT"^^xsd:string .

###  http://www.semanticweb.org/FCA#PowerPlantA
FCA:PowerPlantA rdf:type owl:NamedIndividual ,
FCA:Organization ;
FCA:hasAssets FCA:Computer ;
FCA:hasEmployees FCA:Francis ,
FCA:John ;
FCA:Name "PowerPlantA"^^xsd:string .
```

Deploying the database server: This step involves the creation of the table objects for MySQL and MSSQL databases, as well as the commands for populating them. For the sake of demonstration, the MSSQL database table schemas and contents are different from the ones used in the MSSQL database. At the end, these two databases should maintain different data over distinct schemas, which will become federated at the upper level of the web service. The applied commands were the following:

```
generate-mapping -u root -p password01pt -o ssfile_MYSQL.ttl
-d com.microsoft. sqlserver.jdbc.SQLServerDriver
jdbc:sqlserver://host_mysql;databaseName=BD_mssqlDB

generate-mapping -u sa -p password02pt -o ssfile_SQLServer.ttl
```

```
-d com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc:sqlserver://host_mssql;databaseName=BD_mysqlDB
```

Prepare mapping: The mapping process between database and RDF schemas is mapped through the 'ssfile_SQLServer.ttl', whose contents include the mapping between the MSSQL server and RDF schemas – the 'ssfile_MYSQL.ttl' file plays the same role, but for the MySQL schema. The initial section of these files includes a set of prefixes (several were removed from the next listing for clarity), with the map:database component providing a way for retrieving information from the database server. These files were manually updated to allow the correct mapping between RDF and the database schemas. This mapping is supported by RDF d2rq:ClassMap and d2rq:PropertyBridgefor classes and properties, respectively. Listing 5.3 includes the contents for mapping the class 'Employee' and table 'Employee' from the MSSQL server:

Listing 5.3: Mapping between RDF and database schemas

```
@prefix map: <#> .
@prefix db: <> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .

map:database a d2rq:Database;
        d2rq:jdbcDriver "com.microsoft.sqlserver.jdbc.SQLServerDriver";
        d2rq:jdbcDSN "jdbc:sqlserver://localhost;databaseName=BD_joaohenriques";
        d2rq:username "joaohenriques";
        d2rq:password "password1";
        .

# Table CREATE TABLE dbo.Employee (Number INT, Name VARCHAR(100))

map:dbo_Employee a d2rq:ClassMap;
        d2rq:dataStorage map:database;
        d2rq:uriPattern "dbo/Employee/@@dbo.Employee.Number@@";
        d2rq:class vocab:dbo_Employee;
        d2rq:classDefinitionLabel "dbo.Employee";
        .
map:dbo_Employee__label a d2rq:PropertyBridge;
```

```
        d2rq:belongsToClassMap map:dbo_Employee;
        d2rq:property rdfs:label;
        d2rq:pattern "Employee␣#@@dbo.Employee␣@@";
        .
map:dbo_Employee_Number a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:dbo_Employee;
        d2rq:property vocab:dbo_Employee_Number;
        d2rq:propertyDefinitionLabel "Employee␣Number";
        d2rq:column "dbo.Employee.Number";
        d2rq:datatype xsd:integer;
        .
map:dbo_Employee_Name a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:dbo_Employee;
        d2rq:property vocab:dbo_Employee_Name;
        d2rq:propertyDefinitionLabel "Employee␣Name";
        d2rq:column "dbo.Employee.Name";
        d2rq:datatype xsd:string;
```

Activate D2R servers: The next step deploys the D2R server, in order to map the contents from RDB to RDF according to the mapping file. The following command activates the MSSQL and MYSQL servers respectively:

```
d2r-server -p 2021 ssfile_SQLSERVER.ttl

d2r-server -p 2020 ssfile_MYSQL.ttl
```

Activate web service: The web service provides the main functions performing the federation mechanism and retrieving the information from the SPARQL endpoints. The web service provides an interface and a federated query layer and offers query services that allow end users to perform the intended inference operations while remaining abstracted from low-level details. Each submitted query is forwarded to multiple RDBs through a DR2Q component. The results are later merged into a single result set. The endpoints are configured at server level, and take into consideration the fact that the end user does not need to know the number or the location of such existing endpoint servers. The web service endpoint is located at

`http://host_webservice:17129/WebService1.asmx?op=SemanticWEB.`

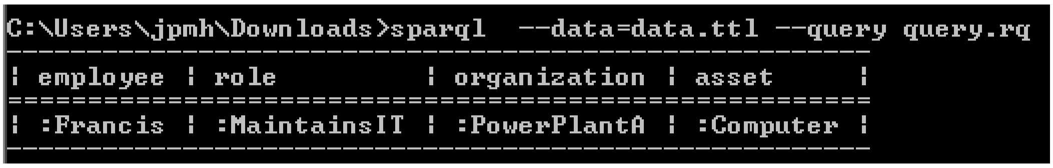
Query the ontology: The final step is to query the knowledge base. The SPARQL query in Listing 5.4 requests the knowledge base for assessing which users are authorised to execute the maintenance of the assets in a given organisation. This query is forwarded from the Web service to all the federated SPARQL endpoints assigned to different databases, and finally translated into the internal schema of those databases. The query filters the organisation ‘PowerPlantA’ for the asset ‘Computer’, where just some of the employees having the role ‘MaintainIT’ are authorised to perform its maintenance.

Listing 5.4: SPARQL query for assessing authorised users

```
PREFIX : <http://www.semanticweb.org/FCA#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT *
WHERE{
        ?employee rdf:type owl:NamedIndividual.
        ?employee :hasRole ?role.
        ?organization rdf:type :Organization.
        ?organization :hasEmployees ?employee.
        ?asset rdf:type owl:NamedIndividual.
        ?role :isRoledBy ?employee.
        FILTER(?organization = :PowerPlantA)
        FILTER(?asset = :Computer )
        FILTER(?role = :MaintainsIT )
 }
```

Figure 5.3 illustrates the use of the Apache Jena SPARQL command ‘sparql –data=data.ttl –query query.rq’ and the corresponding output. The query contents are located in the ‘query.rq’ file, which was used against data located at the ‘data.ttl’ file. According to the knowledge base, just ‘Francis’ is able to execute the ‘Computer’ maintenance.



```
C:\Users\jpmh\Downloads>sparql  --data=data.ttl --query query.rq
-------------------------------------------------------
| employee | role         | organization | asset      |
=======================================================
| :Francis | :MaintainsIT | :PowerPlantA | :Computer |
-------------------------------------------------------
```

Figure 5.3: SPARQL command

5.4 Summary

This chapter proposes an approach for leveraging inference capabilities in the use of heterogeneous data currently maintained in multiple, natively different RDB systems. This approach aims at contributing to CIP by supporting activities such as forensic analysis and compliance audit procedures. It provides Semantic Web reasoning capabilities through an interface able to answer to federated queries. The process of interactively exploring, searching, extracting, pinpointing, and combining insights can use and combine data sourced from disparate organisational RDBs. Thus, this approach avoids the duplication of information in RDB and RDF stores, and overcomes the issues arising from the use of static data integration (such as the lack of support for transformations of data and the effort required for maintaining up-to-date synchronisation processes). The proposed web service includes an abstraction layer that deals with inherent complexities of resorting to different platforms, systems, technologies, and information schemas to retrieve and to combine heterogeneous data. This abstraction layer also improves security by hiding the infrastructure's internal details.

Although the approach taken by the proposed federated architecture is similar to the one of SPARQL 1.1, it does not require previous knowledge about the existence and location of SPARQL endpoints. The benefits of this approach come from the inclusion of an abstraction layer, which provides direct access to operational data that live in different organisational RDBs. Details such as the involved database servers and differences between schemas can be kept away from users. Moreover, it is flexible enough for leveraging the exploration of additional data sources that might be easily added in the future. The proposed framework also provides a data fusion solution for gathering multiple data items – representing the same real-world object – into a single, consistent, and clean representation.

This work arises from the need to improve and facilitate the usage of the huge amounts of data living in the RDBs of CI operators. It also explored Semantic Web inference tools, and is aimed at the practical objective of federating queries against a knowledge base containing the ontology and data for assessing employee authorisations for asset maintenance in a large organisation that uses multiple different RDBs. This practical approach suggests a future path for the improvement of CIP by using inference capabilities for forensic and compliance audit purposes and leveraging the use of heterogeneous ontology data living in RDBs and in other heterogeneous kinds of data sources.

Chapter 6

An Automated Closed-Loop Framework to Enforce Security Policies from Anomaly Detection

Contents

Due to the growing complexity and scale of Information Technology (IT) systems, there is an increasing need to automate and streamline routine maintenance and security management procedures, to reduce costs and improve productivity. As a result, approaches such as the ETSI Zero-touch Network & Service Management (ZSM) [415] are becoming increasingly popular.

Such approaches enable greater consistency and uniformity and contribute to significantly enhancing the efficiency of operations and maintenance activities. Moreover, they may result in cost savings and a significant reduction in human errors. Similar approaches also occur in software development practices, with the widespread adoption of agile techniques for reducing the time allocated to the software development cycle and IT operations, leading to the well-known concept of DevOps [416]. The addition of security management as a third pillar, complementing development and operations, characterizes the emerging field of DevSecOps.

In the scope of DevSecOps methodologies, policies are a fundamental instrument to accelerate the application of best practices, since they potentially enable the automated adaptation of source code and operations to cope with new threats, changes in the network topology, new services, etc. Policies can express the desired system behavior in high-level general terms, and be later translated into specific lower-level rules applicable to the configuration of each specific component of the system.

Once security holes are found, the design, implementation and application of specific security policies require significant efforts from operators and developers. They need to design the policies and translate them to rules, code or other artifacts. This burden increases even more in the case of large organizations. Also, the verification of policies by humans is time-consuming, and the required time significantly increases with the complexity of the infrastructure. This is aggravated by the fact that rules may not exist *a priori*, being created and evolved as data becomes available [417].

Frequently, policies are enforced by directly embedding them in source code. Many existing policies or Access Control Lists are set by the use of options in user interfaces, which is not an easily repeatable or versionable task. This is inefficient and makes it difficult to keep up-to-date inventories, also hampering automated testing. Moreover, Access Control Lists usually lack support for auditing or checking if policies are being violated.

Translating policies expressed in natural language into formalized documents, in formats understandable by both humans and machines, can be challenging. Such formalized documents provide guidance and enhance readability, testability and reportability. However, such documents are still high-level, lacking the specific mappings into the configurations and tools used in the target domain, making it difficult to directly convert them into actionable actions.

A possible approach to overcome this problem is to take the concept of putting code in a high-level language to manage and automate the enforcement of policies, known as Policy as Code (PaC). This is a relatively new concept that helps decoupling the enforcement decisions from business logic policies. Describing policy logic directly in code, rather than depending on a natural language, may help documenting the reasons for those policies, by extending them using comments. PaC may help converting configuration policies into readable formats easily editable, auditable and reproducible by IT managers. Further, they can be translated into intermediate languages recognized by Policy Engine (PE). PaC offers the opportunity to have policies incrementally refined and versioned, to support automating activities. Similar to code, it is possible to include in PaC the programming constructs that determine decisions, helping to automate the enforcement of policies. Moreover, PaC may be reviewed and checked by automated tests, reducing the need of human-based testing operations. The PaC concept can be applied to different domains, such as security, software development and IT operations rules and processes.

In general, Machine Learning (ML) may help generating source code [418, 419, 420, 421]. Specifically in domain of anomaly detection, Decker et al. described a real-time evolving solution based on a fuzzy rule-based classification model for log-based anomaly detection [417]. Henriques et al. also highlighted how ML models can generate sets of rules at scale from unknown data [422]. Overall, these works inspired the approach we propose in the chapter.

It is evident that the evolving threat landscape requires the introduction of new approaches for deployment, monitoring and assessment of security policies. Inspired by ZSM Zero-touch principles and the aforementioned works, we propose a continuous automated closed-loop relying on three stages. Firstly, to extract the Decision Tree (DT) from ML models to identify the anomalies. Secondly, translating them to policies. Thirdly, enforcing them along with the different system components. This continuous closed-loop makes it possible update policies along time with the most recent data. The ML model produces DTs that identify the anomalies to be translated to PaC in a language recognized by the PE. This way, it is possible to reduce the human effort associated to defining and writing the policies to be enforced.

The rest of this chapter, whose contents have already been published in [363], is organized as follows. Section 6.1 introduces the key concepts and technologies for automating policies through the use of source code (including a literature review on PaC and PE tools) and Section 6.2 discusses related work. Section 6.3 presents the proposed closed-loop framework, and Section 6.4 describes a proof-of-concept implementation. Section 6.5 discusses validation, and Section 6.6 concludes the chapter.

6.1 Automating Policies as Code

This section introduces concepts and technologies related with automating policies through the use of source code, starting with base concepts such as policies, PaC, DT, followed by a discussion of several technologies that support PEs.

In this framework, policies specify the conditions under which particular activities should be allowed, to enable logic-based enforcement decisions. Policies include conditions such as rules providing fine-grained control and governing activities for a specific domain (e.g., network security policies; periods under which deployments are allowed), representing the conduct to be evaluated

Policies may cover a large number of use cases. For example, to follow the best practices of data security according to the Payment Card Industry Data Security Standard (PCIDSS) [423], or to enforce the best secure coding practices, such as the Open Web Application Security Project (OWASP) [424], the Computer Emergency Response Team (CERT) C Secure Coding Standard [425], the Common Weakness Enumeration (CWE) [426] and the Common Vulnerabilities and Exposures (CVE) [427] recommendations, the Defense Information Systems Agency's National Vulnerability Database (DISA NVD) [428] and the Common Vulnerability Scoring System (CVSS) [429].

The PaC concept was inspired by Donald Knuth's notion of literate programming [430], driven by the need to document programs to non-technical people. PaC also takes the best practices from Infrastructure as Code (IaC) on the automatic configuration of system dependencies [431]. Conceptually speaking, IaC relies on scripted workflows that are used to configure software systems and cloud instances at scale, in a secure manner. However, despite the evident potential of IaC for security purposes, recent literature reviews [431] found no works specifically addressing security applications.

PaC should be learnable and writable by humans with no programming skills, including those responsible for implementing, updating and auditing them. PaC machine-readable language can be applied programmatically to improve efficiency along with the development and deployment cycles. PaC allows to automatically audit the deployed systems and check their compliance, to detect gaps and quickly apply fixes. This efficiency results from the use of libraries of policies as templates for new applications and infrastructure environments. PaC also reduces the number of errors, because code and deployments can be tested before being run, decreasing implementation/deployment risks and costs. With a test sandbox, IT managers can also check policy changes against the entire policy stack, to ensure (i) modifications do not break the existing rules and (ii) there are no situations not covered by any rules.

Moreover, PaC leverages the application of consistent and accountable processes over time. Since policies are encoded in text files, it is possible to manage their lifecycle by using a Version Control System (VCS) such as git, taking advantage of features such as history, diffs, pull requests, and a central location for storing policies across platforms and applications. The VCS contributes to reusing code and helps to define modular policies that can be aggregated into comprehensive PE, to test policies on an isolated test or development environment before deploying them to production systems. The policies maintained by VCS can integrate the existing Continuous Integration/Continuous Delivery (CI/CD) development pipelines to automate approval, to ensure software compliance and to enable a tight feedback loop between developers and reviewers.

PaC can help documenting policies (which become self-documented), controls and best practices. It can be used to define the security policies to be enforced, including firewall rules, applications, resources or data access controls, data encryption rules, and code provenance restrictions. Thus, PaC also helps Software Bill of Materials assessment and tracking, in the scope of software supply chain risk management.

Enforcing policies is as important as defining and documenting them. Similarly to software compilers, PE translate PaC into implementations (e.g., network security configuration, autorization control policies or Kubernetes cluster parameters) in different environments. PE provide the capability to systematically check if a rule is broken. A PE includes the mechanisms to automatically check logical inconsistencies, syntax errors, and missing dependencies. The PE takes decisions by evaluating inputs against policies and data. PE should be generic enough to be applied to different scenarios, combining context-specific data with the higher-level policies, to enforce them according to each specific context.

PaC and PE can be used in IaC platforms to enforce infrastructure provisioning and deployment policies such as container cluster parameters and constraints in workload placement. IaC software might query the PE to take decisions before provisioning (e.g. depending on the type of node, storage, network dependencies, and application being targeted) – thus, they also help restricting access to infrastructure and enforcing rationalization policies.

Several tools are available for implementing PE. Kyverno [432], for instance, is designed specifically for Kubernetes, managing policies as Kubernetes resources which can be generated, validated and mutated. Pulumi CrossGuard [433] works with cloud management tools for AWS, Azure, Google Cloud and Kubernetes. The Open Policy Agent [434] is open-source and includes a high-level declarative language for writing PaC.

Azure PaC [435] is one of the few PaC software tools currently available for cloud environments. It can be used to define policies affecting firewall rules, application, resource or data access limits, data encryption rules, or code provenance constraints (among others), which are stored on a VCS and tested upon change.

Sentinel [436] is a policy language and a framework designed to be integrated into applications, providing an automated test framework enabling continuous integration. HashiCorp Consul [437], Nomad [438], Terraform [439], and Vault [440] rely on Sentinel functionalities.

A recent example of a standard built upon a closed-loop management approach is ETSIs ZSM [415, 439], an end-to-end reference architecture that uses feedback-driven processes to achieve intelligent automated and management functionalities.

6.2 Related Work

This section discusses previous work addressing automated and dynamic policy-based approaches somehow related with the scope of our proposal.

Moore et al. [306] presented a ML solution to automatically generate program affinity policies that consider program behavior and the target machine. Similarly, Quiroz et al. [307] relied on unsupervised algorithms to capture the dynamic behavior of systems and the hidden relationship between the high-level business attribute space, and the low-level monitoring space. Similarly, Pelaez et al. [308] used supervised models to capture the dynamic behavior.

Johansen et al. [309] proposed a mechanism for expressing and enforcing security policies for shared data expressed as stateful meta-code operations defined in scripting languages interposed in the filesystem.

Gheibi et al. [310] reviewed the state of the art on the use of ML in self-adaptive systems based in the traditional Monitor-Analysis-Planning-Executing (MAPE) [311] feedback loop. Weyns et al [312] presented an approach combining MAPE and Control Theory to produce better adaptive systems.

Finally, the more recently contributions on use of ML models supporting the automation of self-adaptive IT operations has emerged a new field (AIOps) [318, 319, 319] while their contributions have been organized in a taxonomy by Notaro et al. [320].

Our proposal suggests going a step further in the AIOps automation approach, by extending it to the security field (AISecOps). As explained next, it introduces a translation stage integrated within a closed feedback loop pipeline for simultaneously filling the gap and leveraging the benefits of decoupling ML model training and the security policies to be enforced.

6.3 Proposed Approach

This section presents the proposed closed-loop framework that allows to create a workflow for automating the end-to-end process that goes from the classification of anomalies to translational policy rule generation and subsequent enforcement.

As illustrated in Figure 6.1, the proposed continuous closed-loop model S^n relies on a three-stage loop which is applied along n iterations, as formulated in (6.1).

$$S^n = \{S_1^n, S_2^n, S_3^n\} \tag{6.1}$$

The adoption of a closed-loop helps reducing the security risks arising from organizations with outdated security rules. The continuous workflow keeps deployed rules updated, by taking into account the most recent monitoring data to adjust the notion of anomaly, and to automatically adjust deployed rules based on the retrained ML models (more specifically DTs, in the case of this proposal) generated in this way.

The first stage (S_1), automatically takes into consideration new incoming data to classify security anomalies. A DT model fits the data to classify the anomalies. At the second stage (S_2), the previously generated DTs are translated into PaC rules

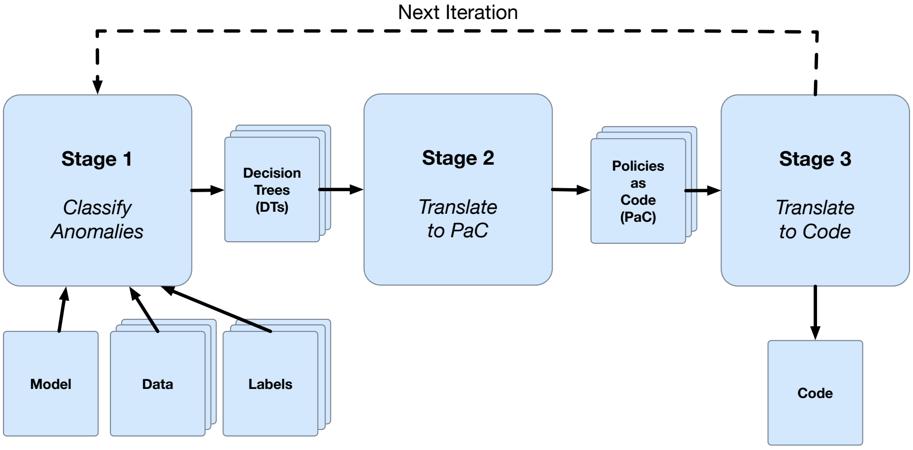


Figure 6.1: Proposed Closed-Loop Approach

in a format recognized by the PE. These rules bring together the conditional logic and the granular controls. Finally, at the third stage (S_3), the produced PaC is enforced by PE. The next cycle may be triggered periodically or based on specific events which, by their nature, might require rule adjustments. Next, is discussed each stage in more detail.

6.3.1 First Stage

The first stage (S_1) takes as input: the DT ML family of algorithms $M_{DT}()$ (e.g. Random Forest, XGBoost); input data D_S organized according to the schema S; and optional labels J (e.g., in case of supervised learning models) to obtain the DTs as T_S, according to 6.2.

$$S_1 : (M_{DT}, D_S, J) \rightarrow T_S \tag{6.2}$$

The realization of this first stage can be based, for instance, on the unsupervised learning model proposed in [422]. This model identifies the DTs classifying the anomaly R_a, and non anomaly R_n events from unlabeled data. as denoted in Algorithm 2. Therefore, the overall list of DTs T_S combines the R_a and R_n to be included as input for the second stage, according to 6.3.

$$T_S = R_a \bigcup R_n. \tag{6.3}$$

It should be noted that the framework does not propose to separate the rules and then to gather them again. Instead, the union presented in (6.3) denotes the ability of the framework to integrate classification models. This is achieved by integrating the resulting rules from an unsupervised learning model into the framework Decision Tree (T_S) set. In this case, we highlight that T_S can plug a binary classification model by integrating the anomalies (R_a) and non-anomalies (R_n) rules into the T_S set.

Algorithm 2 Unsupervised Learning Model

```
INPUT: $D_S$, Data
  $clusters \leftarrow 2$
  $K \leftarrow \text{KMeans}(clusters)$
  $Y \leftarrow K.\text{Train}(D_S)$
  $X \leftarrow \text{XGBoost}(D_S, Y)$
  $X.\text{train}()$
  $ypred \leftarrow X.\text{Predict}(D_S)$
  $R_1, R_2 \leftarrow X.\text{DecisionTrees}()$
  for all $i \in ypred$ do
      if $ypred_i > 0.5$ then
          $ypred_i^1 \leftarrow 1$
          $k2 \leftarrow k2 + 1$
      else
          $ypred_i^1 \leftarrow 0$
          $k1 \leftarrow k1 + 1$
      end if
  end for
  if $k1 > k2$ then
      $R_a \leftarrow R_2$
      $R_n \leftarrow R_1$
  else
      $R_a \leftarrow R_1$
      $R_n \leftarrow R_2$
  end if

OUTPUT: $R_a$, Anomaly Decision Trees
OUTPUT: $R_n$, Non Anomaly Decision Trees
```

6.3.2 Second Stage

The second stage represents the key contribution of the proposed framework. A mapping function $S_2()$ receives as input the DTs T_S produced by the first stage and outputs policies P_S, according to 6.4.

$$S_2 : T_S \rightarrow P_S \tag{6.4}$$

Each policy P_S is defined by a set of rules, as per 6.5.

$$P = \{R_i\}_{i=1}^{n} \tag{6.5}$$

Each policy has associated an identification, a name, a description, and a level of enforcement $P^{(i,n,m)}$. It denotes a logical disjunction of n Boolean rules R_i, as described in 6.6.

$$P^{(i,n,m)} = R_1 \vee R_2 \vee \cdots \vee R_n = \bigvee_{i=1}^{n} R_i \tag{6.6}$$

According to the circumstances, a rule R_i denotes the conjunction of either positive or negative disjunctions of specific attribute levels, as denoted by (6.7).

$$R_i = \bigwedge_k S_k \tag{6.7}$$

The policies P target the domain data D_S (including the events $e_k \in D_S$) expressed using the schema S, according to 6.8. The schema S is a set of features $a_k \in S$.

$$D_S = \bigcup_{k=1}^{n} e_k \tag{6.8}$$

A set of logical operators (e.g. AND, OR, NOT) helps defining complex rules R_i, and $\bigvee$ and $\bigwedge$ represent the Boolean algebra operators OR and AND. Using these operators, it is possible to construct other operators, such as "CONTAINS", "IN", "IS", or "MATCHES". Moreover, l_k^j refers to one of the logical parts of a statement S_k about the j^{th} attribute. Thus, the statement is composed of two distinct parts 6.9.

$$S_k = n_k \bigvee_j l_k^j \tag{6.9}$$

The first part is the disjunction of level values with l_k^j the j^{th} level of the attribute a_k. The second part is the parameter $n_k \in [1, \neg]$, which allows negating (logical operator NOT) the disjunction when set to $\neg$. The user enters specific rules specifying the levels l_k^j and the parameters n_k, as expressed in 6.10.

$$R_i = \bigwedge_k (n_k \bigvee_j l_k^j) \tag{6.10}$$

A policy P will be checked by function $X()$ with data D_S and a set of rules or a policy P. This check produces a Boolean classification telling whether the Policy is being met or not 6.11.

$$X : (P, D_S) \rightarrow K \tag{6.11}$$

It should be noticed that the model can have different levels of enforcement $L = \{l_w, l_s, l_m\}$. At (default) mandatory level l_m, the policy must be complied, regardless of the circumstances and can not be overridden. In the warning level l_w, the failure of policies is allowed and just produces a warning to the user. The intermediary soft level l_s applies to policies that can be overridden to support the configuration of exceptions. Therefore, the enforcement levels $l \in L$ are also input to function $X()$, as described in 6.12.

$$X : (P, D_S, l) \rightarrow K, l \in L \tag{6.12}$$

6.3.3 Third Stage

In the third and final stage (S_3), the PE translates the policy P_S resulted from the previous stage (6.4) into native code C_p, expressed in a programming language p to be deployed for enforcement purposes 6.13.

$$S_3 : P_S \rightarrow C_p \tag{6.13}$$

6.4 Proof-of-Concept Implementation

This section presents a Proof of Concept (PoC) implementation of the proposed approach, which demonstrates its practical feasibility by producing PaC rules from the identification of anomalies to be enforced by a PE.

This PoC was developed for spam detection use case scenarios, in email systems. According to these scenarios, an IT manager dictates a high-level rule to block suspect (spam) messages. Nevertheless, the objective is not to require the IT manager to specifically express how messages are classified as spam.

6.4.1 First Stage

First, a DT classification model $M_{DT}()$ fits the incoming data. In the PoC, for instance, we used a labeled dataset of emails [441] (originally created from [442]) to train the model, obtaining DTs from the anomaly classification process.

Function $M_{DT}()$ is used to train a ML model with the email dataset as input data D_S and corresponding labels J in schema S, to obtain T_S (cf. Equation 6.2). The resulting DTs T_S provide the logical steps for classifying anomalous emails (label 0) and non-anomalous emails (label 1), as illustrated in Listing 6.1.

Listing 6.1: Decision Trees for Email Classification

```
|--- feature_13 <= 0.50
|   |--- feature_916 <= 0.50
|   |   |--- feature_92 <= 0.50
|   |   |   |--- feature_37 <= 0.50
|   |   |   |   |--- feature_418 <= 0.50
|   |   |   |   |   |--- feature_36 <= 0.50
|   |   |   |   |   |   |--- feature_81 <= 0.50
|   |   |   |   |   |   |   |--- feature_104 <= 0.50
|   |   |   |   |   |   |   |   |--- feature_68 <= 0.50
|   |   |   |   |   |   |   |   |   |--- feature_107 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1139 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1139 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_107 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_535 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_535 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
```

6.4.2 Second Stage

Next, Sentinel [436] is used as the domain-agnostic policy language. A mapping function was implemented to translate the previous DTs T_S into Sentinel policies P_S, therefore filling the role of the S_2 function from (6.4). These Sentinel policies are sets of rules defined with key-value pairs, with the main rule with a test.

Listing 6.2 shows the Sentinel policy to classify class 0 (spam email), while Listing 6.3 represents the Sentinel policy to classify class 1 (regular email).

Listing 6.2: Sentinel Policy for class 0 (spam email)

```
main = rule {(feature_13 <= 0.50 and feature_916 <= 0.50 and
    feature_92 <= 0.50 and  feature_37 <= 0.50 and
    feature_418 <= 0.50 and feature_36 <= 0.50 and
    feature_81 <= 0.50 and feature_104 <= 0.50 and
    feature_68 <= 0.50 and  feature_107 <= 0.50 and
    feature_1139 > 0.50)
    or
    (feature_13 <= 0.50 and feature_916 <= 0.50 and
    feature_92 <= 0.50 and  feature_37 <= 0.50 and
    feature_418 <= 0.50 and feature_36 <= 0.50 and
    feature_81 <= 0.50 and feature_104 <= 0.50 and
    feature_68 <= 0.50 and  feature_107 > 0.50 and
    feature_535 <= 0.50)}
```

Listing 6.3: Sentinel Policy for class 1 (regular email)

```
main = rule {(feature_13 <= 0.50 and feature_916 <= 0.50 and
    feature_92 <= 0.50 and feature_37 <= 0.50 and
    feature_418 <= 0.50 and feature_36 <= 0.50 and
    feature_81 <= 0.50 and feature_104 <= 0.50 and
    feature_68 <= 0.50 and  feature_107 <= 0.50 and
    feature_1139 <= 0.50)
    or
    (feature_13 <= 0.50 and feature_916 <= 0.50 and
    feature_92 <= 0.50 and feature_37 <= 0.50 and
    feature_418 <= 0.50 and feature_36 <= 0.50 and
    feature_81 <= 0.50 and feature_104 <= 0.50 and
    feature_68 <= 0.50 and feature_107 > 0.50 and
    feature_535 > 0.50)}
```

In this PoC an instance of the sklearn [443] *DecisionTreeClassifier* algorithm was created, initialized with "maximum depth" set to 20. The dataset fit to this model was split with 80% for training and 20% for tests. Each word in the email dataset corresponds to a distinct feature. The function *export_ text()* provided the rules from the DTs resulting from the training stage.

6.4.3 Third Stage

Finally, the previously produced PaC P_S is translated to a language C_p recognized by the PE, according to the function S_3 referred in (6.13).

A test folder was created for the policy to be run, and a file with that policy defined in JavaScript Object Notation (JSON) format is stored in that folder. Since Sentinel allows to define one policy per class (anomalies and non-anomalies), two policies were created. Finally, policies were moved to a Github repository to streamline the PoC with versioning, continuous deployment and pull request capabilities.

For real-use scenarios, the PoC can be integrated into CI/CD tool-chains. Within a continuous integration pipeline, for example, it is possible to run a specific command translating a Sentinel PaC into an artifact containing the email rules that the email server understands.

6.5 Validation

The validation of the proposed framework is not straightforward, because its potential benefits result mainly from the operational gains obtained over time, in terms of cost of keeping rules updated and (indirect) accuracy improvements – which are not easy to measure.

To fully assess the performance of the proposed framework would require access to datasets whose rules had evolved over a significant period of time (so that new types of cyberattacks or new types of spam email would start appearing only after some time), so that it could measure the improvements brought by the automated adjustment of the rules over time, and also the ability to preserve (or even increase) the system accuracy.

Since datasets with the aforementioned characteristics are not available, a different but still relevant experiment was devised. Starting with a publicly available dataset with spam email [441] (created from [442]), we performed the following experiment:

- First, the dataset is split in six different blocks with similar sizes (block 0, block 1, block 2...). These blocks emulate the emails received during six consecutive periods (e.g., one week).

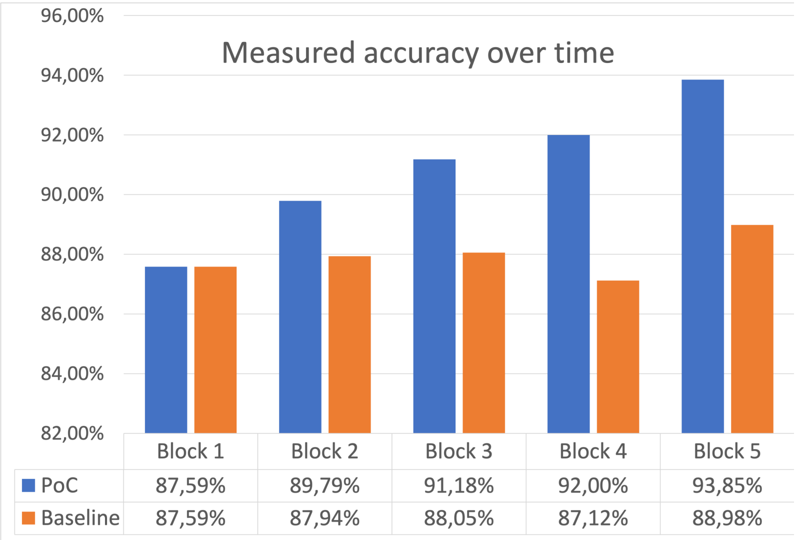
- Block 0 was used to train both the platform and a baseline system. This would be similar, for instance, to the initial training of the system with the emails from the previous week.

- Afterwards, the accuracy of the trained system was tested with block 1 as input – this could represent, for instance, the first week of emails with the framework running.

- Next, the PoC performed an automatic readjustment, based on the original training and on the updates induced by the inputs from block 1 (i.e. the first week). This corresponds to the first automatic readjustment of the rules. The baseline system used for comparison kept using the original training data.

- Then, the process was repeated for the next blocks, so that the PoC kept automatically refining the rules. This could correspond, keeping the analogy, to having 5 weeks of operation with weekly updates.

The accuracy obtained in each of these steps is presented in Figure 6.2. Overall, these results are in line with was expected. For the baseline system, the accuracy remained stable (with slight natural fluctuations), around 87%-89%. When using the proposed approach, the system kept improving accuracy over time, since the data from the previous period was used to further refine the models. It should be noted, however, that in real world operations are expected results to be slightly different: while baseline (i.e. static) systems are expected to degrade their accuracy over time (due to the appearance of new types of spam or cyberattacks not present in the original training data), this approach is expected to preserve accuracy over time, adjusting to those changes.



	Block 1	Block 2	Block 3	Block 4	Block 5
🟦 PoC	87,59%	89,79%	91,18%	92,00%	93,85%
🟧 Baseline	87,59%	87,94%	88,05%	87,12%	88,98%

Figure 6.2: Measured accuracy over time for PoC and baseline systems

6.6 Summary

In this chapter we proposed a closed-loop framework aiming to reduce the evolving security risks organizations are exposed to, by streamlining the routine maintenance and management of security policies.

This work was inspired by the ideas of translating policies to code that are present in several works [417, 418, 419, 420, 421], also aligning with the Zero-touch concept of the ETSI ZSM framework. It supports a closed-loop with the intelligence and automation of the tasks of monitoring and detecting the ongoing threats, to produce the security policies to be enforced.

The presented PoC, based on a simple but representative use case, shows how this approach can be applied in practice, to streamline the security operations associated with keeping spam email filters up-to-date. The first stage classifies spam emails as anomalies, extracting the DTs that identify spam messages as anomalies. Next, policy rules are generated, by means of translating those DTs into PaC. Finally, those PaC can be used by email servers to block new spam emails.

This process is cyclic, and can be triggered at regular time intervals or based on specific events. Emails classified by users (as spam or not spam) are used to progressively update applied policies. Automating these process reduces the operators' burden by streamlining routine maintenance and security management procedures.

The adopted policy engine in the proposed framework enables decoupling policies from the applications that will enforce them. Moreover, it may be integrated with other tools, for instance to identify threats and take automatic responses on stopping attacks in progress or introducing defensive actions.

The proposed framework helps automating repetitive operation tasks related with updating and enforcing policy rules. This potentially improves productivity and reduces the continuous effort of maintaining the systems' security up-to-date. Moreover, the time required to apply new security rules is shortened, reducing the time the systems are exposed to outdated policies.

Translating DTs into PaC contributes to the readability of those policy rules by human operators, while not requiring specific programming skills. The presented PoC can be generalized to fit other anomaly detection scenarios requiring frequent updates. The framework can also be applied to automatically update and enforce forensics and compliance auditing mechanisms.

Despite the potential benefits of the proposed framework, it should be noted that some drawbacks may arise. First, relying on an automatic enforcement from newly generated policies, generated from ML models, in some cases may result in a significant number of false positives. This may be attenuated by prior validation by humans before enforcing those policies, at the cost of some degradation in the process streamlining levels. Second, despite the benefits brought by PaC, some compromises apply regarding performance and flexibility. Performance can be compromised because, typically, PaC does not support unsafe operations (such as direct memory access) or operations (such as sub-process execution). In terms of flexibility, PaC may result in a limited offer in terms of programming languages.

The presented PoC demonstrates how it can be applied in practice. Beyond the PoC scenario, the framework can be applied to a wide range of other use cases. In practice, any security monitoring scenario with evolving threats and evolving systems, where the criteria to identify anomalies need to evolve over time, can benefit from this framework. General policy based management scenarios, in dynamic environments, may also benefit from the proposed approach, since it enables the streamlining of access policies updates without requiring formal specification of those policy updates and/or their manual translation into code.

With this chapter, we conclude the set of three distinct technical contributions for enhanced data-driven Forensics and Compliance Auditing (FCA) operations in complex scenarios such as Industrial Automation and Control Systems (IACS)-based Critical Infrastructures (CIs). Together with the proposed general FCA framework, they constitute the key contributions brought up by the research work presented in this thesis. The next chapter will conclude the thesis, synthesizing the conducted work, drawing the main conclusions and discussing the options for future work.

Chapter 7

Conclusions and Future Work

Contents

THIS thesis focused on designing a Forensics and Compliance Auditing (FCA) framework for Critical Infrastructure Protection (CIP), as well as on exploring a series of approaches devised to tackle related challenges in terms of knowledge extraction, policy assessment and creation and information organization and retrieval.

This chapter presents the conclusions from the conducted work and the open challenges to be addressed in the future, being structured as follows. Section 7.1 synthesizes this dissertation, while Section 7.2 presents the resulting contributions of this work. Finally, Section 7.3 suggests future directions to address some challenges.

7.1 Synthesis of the Dissertation

The work presented in this dissertation addressed the design, research and development of innovative strategies to improve CIP by adopting a proactive stance that distinguishes itself from the more conventional threat detection and/or mitigation approaches. This is achieved by focusing on building FCA capabilities capable of coping with the challenges of evolved Industrial Automation and Control Systems (IACS) infrastructures and dealing with large data volumes coming from diversified sources and recording system events (including IACS, endpoint hosts, network devices, and servers located at different levels of the Critical Infrastructure (CI)) and operators' actions, to enhance the chance of detecting non-compliance situations, identifying security incidents and extracting evidence.

The first two chapters provided the introductory context for this dissertation. Chapter 1 presented the thesis topics, motivation, objectives and contributions, while Chapter 2 surveyed and explored the FCA scope with relevance to the CIP domain. The latter addressed aspects such as quality Service Level Agreements (SLAs), business rules, and organization policies, also resulting in a template proposal for a FCA architecture (Contribution 1), presented in Chapter 2 and later implemented and validated as a PoC in the scope of the ATENA project.

Chapter 3 presented the FCA solution developed from the aforementioned architectural template (Contribution 2). The main rationale for this framework was to converge both forensics and compliance auditing approaches, aggregating a disparate number of techniques and tools in a unified platform designed for CI contexts. Moreover, it also helps dealing with evolving cyber threats affecting CIs, providing valuable insights to fine-tune traditional security mechanisms (intrusion detection, prevention, and mitigation) and prepare for post-incident forensics analysis.

Complementary to the FCA design, the thesis workplan also encompassed the research and evaluation of analytic models capable of horizontally scaling to improve analytic task capacity and performance, providing the opportunity to improve correlation efficiency for both forensics and compliance auditing tasks, which were discussed in the following chapters.

Chapter 4 presents a solution that combines K-Means and XGBoost models for anomaly detection over large log datasets (Contribution 3), helping create analytical models that are able to identify the security events and compromised areas through pattern analysis deviations from large volumes of data. This solution, which takes advantage of parallel computing resources for increased efficiency, was evaluated and demonstrated within the scope of the Proof of Concept (PoC).

Next, an ontology-based federated approach was proposed providing web semantic inference capabilities over the data living in heterogeneous sources (Contribution 4), as presented in Chapter 5. This approach leverages the process of interactively exploring, searching, extracting, and pinpointing insights by combining data sources from distinct organizational Relational Databases (RDBs).

Finally, an automated closed-loop framework to enforce security policies from anomaly detection models was presented in Chapter 6. The aim of this solution is to generate and enforce security policies from decision Tree Models (Contribution 5). Thus, it bridges the gap between anomaly detection to countermeasure policy definition, by creating workflows that automate the process that goes from the anomaly classification to translational policy rule generation.

When integrated within the scope of the proposed FCA architecture, the contributions of this thesis constitute a comprehensive approach improving CI security by considering forensic and auditing compliance approaches to assist in the investigation of past attacks and non-compliant activity, complementing existing intrusion detection and mitigation capabilities. It is expected that the proposed FCA framework may lead to a reduction in the number of security alerts while requiring less intervention from humans, by automating actions and providing stronger prior validation capabilities – this will help security operators to focus more on analyzing incidents and less on analyzing raw alert streams, often without context.

7.2 Contributions

The objectives presented in Section 1.2 have led to the following contributions:

- **Contribution 1. A comprehensive survey on the topic of FCA for CIP, which also produced a related taxonomy and a reference architecture for consolidated operations.** This survey reviewed the state of the art and the latest developments, methodologies, challenges, and solutions related to the topic. It focused on relevant contributions, capable of tackling the requirements imposed by massively distributed and complex IACS, in terms of handling large volumes of heterogeneous data (that can be noisy, ambiguous, and redundant) for analytic purposes, with adequate performance and reliability. The survey also produced a structured taxonomy for the field of FCA, based on relevant literature, thus contributing to Objectives 1 and 2.

- **Contribution 2. Proposal, design and implementation of a domain-specific FCA framework for CIP scenarios**, proposed as a generic template for converged platforms (which was later materialized in a PoC demonstrated in the scope of the ATENA project) and documented in Chapter 3. Starting from the FCA reference architecture template presented in Chapter 2 and designed to address the objectives stated in Section 1.2, a PoC was later implemented and evaluated, as described in Chapter 3 and subsequent chapters. The evaluation and validation of this contribution included the demonstration of use case scenarios within the context of the ATENA project and the experimental work on scalability. Overall, this contribution corresponds to Objectives 1 and 2.

- **Contribution 3. Mechanisms for detecting anomalies using large log datasets.** In Chapter 4, this thesis presented a solution incorporating novel anomaly detection methods employing distributed processing capabilities to improve the detection efficiency over massive amounts of log records. The solution combined the k-means clustering and the gradient tree boosting classification algorithms to leverage the filtering capabilities over normal events, to concentrate the efforts on the remaining anomaly candidates. This approach, which contributes to reducing the involved computational complexity (and, thus, also contributing to fulfill Objective 1), was validated with specific experimental work and demonstrated as part of the FCA framework PoC, as presented in Contribution 2. This contribution was guided by Objective 3 in the identification of the most suitable analytical models to improve the detection of anomalies and replay of incidents and enhance the ability to collect the evidence for CI forensic and audit compliance processes.

- **Contribution 4. A federated framework to support inference analysis on ontology data.** Chapter 5 presented a solution providing federated capabilities to support querying corporate data from multiple heterogeneous sources, by relying on semantic web mechanisms. This way, it becomes possible to make this data available, at reasonable costs, in a format that is suitable for security management purposes – especially those related to FCA analysis. The proposed inference mechanisms demonstrated the capabilities helping to improve forensic and compliance auditing approaches and identifying the threats from anomalies, which correspond to Objectives 1, 2 and 3.

- **Contribution 5. An automated closed-loop framework to enforce security policies from anomaly detection.** To enforce security policies from anomaly detection, this thesis proposed an automated closed-loop process with three stages, in Chapter 6. The first stage focuses on obtaining the Decision Tree (DT)s to classify anomalies (aligned with Objective 3), with the second stage translating DTs into security Policy as Code (PaC) based on languages recognized by the Policy Engines (PEs). In the last stage, the translated security policies feed the PE, which will enforce them. The feasibility of this framework was also demonstrated by means of an example that covers the three stages of the closed-loop process. Maintaining updated policies can help improving the compliance auditing processes, as stated in Objective 2.

Table 7.1 maps the thesis contributions to the objectives listed in Section 1.2.

	Objective 1	Objective 2	Objective 3
Contribution 1	X	X	
Contribution 2	X	X	
Contribution 3	X		X
Contribution 4	X	X	X
Contribution 5		X	X

Table 7.1: Mapping of Contributions vs. Objectives

7.3 Future Work

This thesis was focused on the design of an FCA framework for CIP, targeting infrastructures such has Industrial IoT (IIoT) environments or evolved IACS, whose intrinsic nature and scale pose significant challenges for any sort of process supporting cybersecurity analytics, forensics or compliance auditing tasks. While this work constitutes a contribution towards optimizing FCA procedures for these domains, it is expected that many challenges will subsist, for years to come. Thus, this section will glimpse on some of these challenges, also unveiling future research directions.

The formal definition of an architecture such as the FCA framework (cf. Chapter 3) can leverage the development of new tools and even help validate commercial solutions. To streamline the integration of other frameworks (e.g. response actions-enabled frameworks), the FCA reference architecture can be extended by adopting the Lambda Architecture concept [444] with a central data hub supporting event streaming (e.g. supported by Apache Kafka [445] or RabbitMQ [446] technologies). Regarding scalability and manageability, another significant step to be taken will be the shift of the FCA framework to an orchestrated environment (e.g. Kubernetes), in order to leverage the availability, resiliency, scalability, performance, integration, and availability benefits of a cloud-native setup. Also, tighter integration of the proposed federated ontology framework [395, 394] into the FCA framework (cf. Chapter 5) can support inference mechanisms over a larger number of heterogeneous sources maintaining corporate data living in RDBs.

Moreover, it is suggested that the investigation of self-adaptive automation approaches to continuously update compliance and auditing policies in cloud-native environments would provide a significant added value for the FCA framework, for example by retraining Machine Learning (ML) models classifying anomaly events as proposed by our work [363] (cf. Chapter 6). This approach can help reducing the dependency on humans for security operations, consequently reducing inherent costs and errors while helping sustain adequate system performance and availability.

Finally, another potential route to explore has to do with the identification of threat traces in more complex scenarios, by combining already classified threats associated to distinct data sources, in a transfer learning style. For such purpose, it is suggested that distinct data sources coupled with several models could be evaluated in parallel, using the capabilities provided by the FCA Analytic Component [259] (cf. Chapter 4). Each one of the ML models could be independently trained, fitting the data of specific sources in a distributed manner and running cross analysis, by considering the already learned specific historical patterns, for example, helping to identify and trace specific threats from anomalies to historical patterns (e.g. from logs from routers, firewalls).

References

[1] ATENA. D4.1 Requirements and Reference Architecture for the Cyber-physical IDS. https://www.atena-h2020.eu/project-documentation/, July 2017. visited on 2017-07-01.

[2] Abdul Rehman Javed, Waqas Ahmed, Mamoun Alazab, Zunera Jalil, Kashif Kifayat, and Thippa Reddy Gadekallu. A comprehensive survey on computer forensics: State-of-the-art, tools, techniques, challenges, and future directions. *IEEE Access*, 2022.

[3] David Maimon and Eric R Louderback. Cyber-dependent crimes: An interdisciplinary review. *Annual Review of Criminology*, 2(1):191–216, 2019.

[4] ATENA. D2.1 state of art. https://www.atena-h2020.eu/wp-content/uploads/ATENA/Publishable%20material/deliverables/D2.1/D2.1_State%20of%20Art.pdf.zip, October 2016. visited on 2016-12-01.

[5] Defense Use Case. Analysis of the Cyber Attack on the Ukrainian Power Grid. https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf, 2016. visited on 2017-03-01.

[6] Ronald L Krutz. *Securing SCADA systems*. John Wiley & Sons, 2005.

[7] Linda Martin. 5th Annual edition of Cyber Defense Magazine - 2017 Predictions. https://www.tradepub.com/free-offer/cyber-defense-magazine--2017-predictions, 2017. visited on 2017-04-01.

[8] IBM. IBM Managed Security Services - United States. https://www.ibm.com/security/services/managed-security-services, 2017. visited on 2016-07-02.

[9] ISO/IEC. Iso/iec 27036-1 information technology — security techniques — information security for supplier relationships. `https://www.iso.org/standard/59648.html`, May 2014. visited on 2017-01-01.

[10] Raymond A Hansen, Kathryn Seigfried-Spellar, Seunghee Lee, Siddarth Chowdhury, Niveah Abraham, John Springer, Baijian Yang, and Marcus Rogers. File toolkit for selective analysis & reconstruction (filetsar) for large-scale networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3059–3065. IEEE, 2018.

[11] Emmanuel S Pilli, Ramesh C Joshi, and Rajdeep Niyogi. Network forensic frameworks: Survey and research challenges. *digital investigation*, 7(1-2):14–27, 2010.

[12] Emi Morioka and Mehrdad Sharbaf. Cloud Computing: Digital Forensic Solutions. In *International Conference on Information Technology-New Generations*, 12, pages 589–594, Las Vegas, April 2015.

[13] Ibrahim Ghafir, Vaclav Prenosil, Jakub Svoboda, and Mohammad Hammoudeh. A survey on network security monitoring systems. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 77–82. IEEE, 2016.

[14] João Henriques, Filipe Caldeira, Tiago J. Cruz, and Paulo Simões. A survey on forensics and compliance auditing for critical infrastructure protection (submitted, under revision). *IEEE Access*, 2023. ISSN 2169-3536.

[15] Nabin Chowdhury and Vasileios Gkioulos. Cyber security training for critical infrastructure protection: A literature review. *Computer Science Review*, 40: 100361, 2021.

[16] Eoghan Casey. *Digital evidence and computer crime: Forensic science, computers, and the internet.* Academic press, 2011.

[17] The New York Times. Biden signs an executive order aimed at protecting critical American infrastructure from cyberattacks. `https://www.nytimes.com/2021/07/28/us/politics/cyber-security-biden-executive-order.html`, 2021. visited on 2021-10-19.

[18] IBM. Sibm security intelligence with big data. `http://www-03.ibm.com/security/solution/intelligence-big-data`, 2016. visited on 2016-12-01.

[19] Arun Kumar Kalakanti, Vinay Sudhakaran, Varsha Raveendran, and Nisha Menon. A comprehensive evaluation of nosql datastores in the context of historians and sensor data analysis. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 1797–1806. IEEE, 2015.

[20] Alberto Fernández, Sara del Río, Victoria López, Abdullah Bawakid, María J del Jesus, José M Benítez, and Francisco Herrera. Big data with cloud computing: an insight on the computing environment, mapreduce, and programming frameworks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5):380–409, 2014.

[21] Chaossearch. The threat hunter's handbook: Using log analytics to find and neutralize hidden threats in your environment white paper. Technical report, National Institute of Standards and Technology, 2020. URL `https://www.chaossearch.io/hubfs/ChaosSearch%20Threat%20Hunters%20Handbook.pdf`.

[22] Irfan Ahmed, Sebastian Obermeier, Martin Naedele, and Golden Richard. Scada systems: Challenges for forensic investigators. *Computer*, 45(12):44–51, 2012.

[23] Keith Stouffer and Joe Falco. *Guide to supervisory control and data acquisition (SCADA) and industrial control systems security.* National institute of standards and technology, 2006.

[24] Luis Rosa, Miguel Freitas, Sergey Mazo, Edmundo Monteiro, Tiago Cruz, and Paulo Simões. A Comprehensive Security Analysis of a SCADA Protocol: From OSINT to Mitigation. *IEEE Access*, 7:42156–42168, 2019. 10.1109/ACCESS.2019.2906926.

[25] Eric Cornelius and Mark Fabro. Recommended practice: Creating cyber forensics plans for control systems. Technical report, Idaho National Laboratory (INL), 2008. URL `https://www.cisa.gov/sites/default/files/recommended_practices/Forensics_RP.pdf`.

[26] Peter Eden, Pete Burnap, Andrew Blyth, Kevin Jones, Hugh Soulsby, and Yulia Cherdantseva. A forensic taxonomy of scada systems and approach to incident response. In *3rd International Symposium for ICS & SCADA Cyber Security Research 2015*, pages 42–51. BCS Learning &Development Ltd, 2015.

[27] Faheem Ullah, Matthew Edwards, Rajiv Ramdhany, Ruzanna Chitchyan, M Ali Babar, and Awais Rashid. Data exfiltration: A review of external attack vectors and countermeasures. *Journal of Network and Computer Applications*, 101:18–54, 2018.

[28] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[29] R.M. Van der Knijff. Control systems/scada forensics, what's the difference? *Digital Investigation*, 11(3):160 – 174, 2014. ISSN 1742-2876. https://doi.org/10.1016/j.diin.2014.06.007. URL `http://www.sciencedirect.com/science/article/pii/S1742287614000814`. Special Issue: Embedded Forensics.

[30] Gail Ridley, Judy Young, and Peter Carroll. Cobit and its utilization: A framework from the literature. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, pages 8–pp. IEEE, 2004.

[31] Georg Disterer. Iso/iec 27000, 27001 and 27002 for information security management. *Journal of Information Security*, 4(2), 2013.

[32] NIST. Security and privacy controls for information systems and organizations sp 800-53 rev. 5. Technical report, National Institute of Standards and Technology, https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final, 2020.

[33] NIST. Protecting controlled unclassified information in nonfederal systems and organizations sp 800-171 rev. 2. Technical report, National Institute of Standards and Technology, 2020a. URL `https://csrc.nist.gov/publications/detail/sp/800-172/archive/2020-07-06`.

[34] Critical Infrastructure Cybersecurity. Framework for improving critical infrastructure cybersecurity. *Framework*, 1(11), 2014.

[35] HITRUS. Hitrust csf framework, 2021. URL `https://hitrustalliance.net/product-tool/hitrust-csf/`. visited on 2022-12-15.

[36] NIST. Sp 800-53 rev. 5 security and privacy controls for information systems and organizations. Technical report, National Institute of Standards and Technology, 2020. URL `https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final`.

[37] ISO. Iso/iec 27000:2018 information technology — security techniques — information security management systems — overview and vocabulary, 2018. URL `https://www.iso.org/standard/73906.html`. visited on 2022-12-15.

[38] ISO. Iso/iec 27004:2016 information technology — security techniques — information security management — monitoring, measurement, analysis and evaluation, 2016. URL `https://www.iso.org/standard/62313.html`. visited on 2022-12-15.

[39] ISO. Iso/iec 27037:2012 information technology — security techniques — guidelines for identification, collection, acquisition and preservation of digital evidence, 2012. URL `https://www.iso.org/standard/44381.html`. visited on 2022-12-15.

[40] ISO. Iso/iec 27038:2014 information technology — security techniques — specification for digital redaction, 2014. URL `https://www.iso.org/standard/44382.html`. visited on 2022-12-15.

[41] ISO. Iso/iec 27042:2015 information technology — security techniques — guidelines for the analysis and interpretation of digital evidence, 2015. URL `https://www.iso.org/standard/44406.html`. visited on 2022-12-15.

[42] ISO. Iso/iec 27050-1:2019 information technology — electronic discovery — part 1: Overview and concepts, 2019. URL `https://www.iso.org/standard/78647.html`. visited on 2022-12-15.

[43] ISO. Iso/iec 27041:2015 information technology — security techniques — guidance on assuring suitability and adequacy of incident investigative method, 2015. URL `https://www.iso.org/standard/44405.html`. visited on 2022-12-15.

[44] ISO. Iso/iec 27043:2015 information technology — security techniques — incident investigation principles and processes, 2015. URL `https://www.iso.org/standard/44407.html`. visited on 2022-12-15.

[45] ISO. Iso/iec 27006:2015 information technology — security techniques — requirements for bodies providing audit and certification of information security management systems, 2015. URL `https://www.iso.org/standard/62313.html`. visited on 2022-12-15.

[46] ISO. Iso/iec ts 27008:2019 information technology — security techniques — guidelines for the assessment of information security controls, 2019. URL `https://www.iso.org/standard/67397.html`. visited on 2022-12-15.

[47] ISO. Iso 21043-1:2018 forensic sciences — part 1: Terms and definitions, 2018. URL `https://www.iso.org/standard/69732.html`. visited on 2022-12-15.

[48] ISO. Iso 21043-2:2018 forensic sciences — part 2: Recognition, recording, collecting, transport and storage of items, 2018. URL `https://www.iso.org/standard/72041.html`. visited on 2022-12-15.

[49] ISO. Iso/iec 30121:2015 information technology — governance of digital forensic risk framework, 2015. URL `https://www.iso.org/standard/53241.html`. visited on 2022-12-15.

[50] ASTM International. ASTM Standards and publications. `https://www.astm.org/products-services/standards-and-publications/standards.html`, 2016. visited on 2023-08-04.

[51] Robert R Moeller. *Sarbanes-Oxley internal controls: effective auditing with AS5, CobiT, and ITIL*. John Wiley & Sons, 2008.

[52] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (idps) sp 800-94. *NIST special publication*, 800(2007):94, 2007.

[53] David Kushner. The real story of stuxnet. *ieee Spectrum*, 50(3):48–53, 2013.

[54] ICS-CERT. Cyber-Attack Against Ukrainian Critical Infrastructure. `https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01`, Feb 2016. visited on 2016-12-01.

[55] Rafiullah Khan, Peter Maynard, Kieran McLaughlin, David Laverty, and Sakir Sezer. Threat Analysis of BlackEnergy Malware for Synchrophasor based Real-time Control and Monitoring in Smart Grid. In *4th Int'l Symposium ICS & SCADA Cyber Security Research. BCS*, pages 53–63, 2016.

[56] Haris Iskandar Mohd Abdullah, Zul-Azri Ibrahim, Fiza Abdul Rahim, Hafizuddin Shahril Fadzil, Saiful Amin Sharul Nizam, and Muhammad Zulhusni Mustaffa. Digital forensics investigation procedures of smart grid environment. *International Journal of Computing and Digital System*, 2021.

[57] Herve Debar. An introduction to intrusion-detection systems. *Proceedings of Connect*, 2000, 2000.

[58] Vir V Phoha. *Internet security dictionary.* Springer Science & Business Media, 2007.

[59] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (idps). *NIST special publication*, 800(2007):94, 2007.

[60] Wei Wang, Sylvain Gombault, and Thomas Guyet. Towards fast detecting intrusions: using key attributes of network traffic. In *Internet Monitoring and Protection, 2008. ICIMP'08. The Third International Conference on*, pages 86–91. IEEE, 2008.

[61] Tiago Cruz, Luis Rosa, Jorge Proença, Leandros Maglaras, Matthieu Aubigny, Leonid Lev, Jianmin Jiang, and Paulo Simões. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Transactions on Industrial Informatics*, 12(6):2236–2246, 2016. 10.1109/TII.2016.2599841.

[62] T. Cruz, J. Barrigas, J. Proença, A. Graziano, S. Panzieri, L. Lev, and P. Simões. Improving network security monitoring for industrial control systems. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 878–881, 2015.

[63] Luis Rosa, Tiago Cruz, Miguel Borges de Freitas, Pedro Quitério, João Henriques, Filipe Caldeira, Edmundo Monteiro, and Paulo Simões. Intrusion and anomaly detection for the next-generation of industrial automation and control systems. *Future Generation Computer Systems*, 119:50–67, 2021. ISSN 0167-739X. https://doi.org/10.1016/j.future.2021.01.033. URL `https://www.sciencedirect.com/science/article/pii/S0167739X21000431`.

[64] Michael E Whitman and Herbert J Mattord. *Principles of information security.* Cengage learning, 2011.

[65] G. Granadillo, S. González-Z., and R. Diaz. Security information and event management (siem): Analysis, trends, and usage in critical infrastructures. *Sensors*, 21(14):4759, 2021.

[66] Gartner. Magic Quadrant for Security Information and Event Management. https://www.gartner.com/doc/reprints?id=1-26Q3T88Y&ct=210706&st=sb, August 2021. visited on 2021-11-26.

[67] Searchsecurity. Security information and event management (siem). http://searchsecurity.techtarget.com/definition/security-information-and-event-management-SIEM, Jun 2017.

[68] Yunchuan Sun, Hongli Yan, Junsheng Zhang, Ye Xia, Shenling Wang, Rongfang Bie, and Yingjie Tian. Organizing and Querying the Big Sensing Data with Event-Linked Network in the Internet of Things. In *International Journal of Distributed Sensor Networks*, 2014. 10.1155/2014/218521.

[69] R. Hunt and J. Slay. Achieving critical infrastructure protection through the interaction of computer security and network forensics. In *2010 Eighth International Conference on Privacy, Security and Trust*, pages 23–30, Aug 2010. 10.1109/PST.2010.5593243.

[70] Gartner. Magic Quadrant for Endpoint Protection Platforms. https://www.gartner.com/doc/reprints?id=1-27NCOQKK&ct=211014&st=sb, May 2021. visited on 2021-11-29.

[71] Wajih Ul Hassan, Adam Bates, and Daniel Marino. Tactical provenance analysis for endpoint detection and response systems. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1172–1189. IEEE, 2020.

[72] Securonix. Securonix security analytics platform. http://www.securonix.com/security-intelligence, 2016. visited on 2016-12-01.

[73] RSA. Rsa security analytics. http://www.emc.com/collateral/data-sheet/security-analytics-overview-ds.pdf, 2016. visited on 2016-12-01.

[74] LogRhythm. Logrhythm security analytics. https://logrhythm.com/products/security-analytics, 2016. visited on 2016-12-01.

[75] Pravail. Pravail security analytics. https://www.pravail.com, 2016. visited on 2016-12-01.

[76] Alienvault. Alienvault: A integrated solution with real-time threat intelligence. http://www.alienvault.com, 2016. visited on 2016-12-01.

[77] Cisco. OpenSOC: Big Data Security Analytics Framework. http://opensoc.github.io, December 2016. visited on 2016-12-01.

[78] Apache Metron. Apache metron: Real-time big data security. http://metron.incubator.apache.org, December 2016. visited on 2016-12-01.

[79] Chuck Lam. *Hadoop in Action.* Manning Publications, 12 2010. ISBN 9781935182191. URL http://amazon.de/o/ASIN/1935182196/.

[80] Kibana. Kibana: Explore and Visualize Your Data. https://www.elastic.co/products/kibana, 2016. visited on 2016-12-01.

[81] Elastic. Elasticsearch: Search and Analyze Data in Real Time. https://www.elastic.co/products/elasticsearch, 2016. visited on 2016-12-01.

[82] IBM Corporation. Cognitive security white paper. http://cognitivesecuritywhitepaper.mybluemix.net/?cm_mc_uid=54600499979414852593580&cm_mc_sid_50200000=1485336800, February 2016. visited on 2016-12-01.

[83] Gustavo Granadillo, Rodrigo Diaz, Juan Caubet, and Ignasi Garcia-Milà. Clap: A cross-layer analytic platform for the correlation of cyber and physical security events affecting water critical infrastructures. *Journal of Cybersecurity and Privacy*, 1(2):365–386, 2021.

[84] Chiara Foglietta, Dario Masucci, Cosimo Palazzo, Riccardo Santini, Stefano Panzieri, Luis Rosa, Tiago Cruz, and Leonid Lev. From detecting cyber-attacks to mitigating risk within a hybrid environment. *IEEE Systems Journal*, 13(1):424–435, 2019. 10.1109/JSYST.2018.2824252.

[85] D. R. Rani and G. Geethakumari. An efficient approach to forensic investigation in cloud using vm snapshots. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–5, Jan 2015. 10.1109/PERVASIVE.2015.7087206.

[86] NIST. Guide to Integrating Forensics Techniques into Incident Response. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf, 2017. visited on 2017-06-01.

[87] Joel T Langill. Defending against the dragonfly cyber security attacks. *Retrieved*, 11:2015, 2014.

[88] Max Fillinger and Marc Stevens. Reverse-engineering of the cryptanalytic attack used in the flame super-malware. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 586–611. Springer, 2015.

[89] Dacfey Dzung, Martin Naedele, Thomas P Von Hoff, and Mario Crevatin. Security for industrial communication systems. *Proceedings of the IEEE*, 93 (6):1152–1177, 2005.

[90] Friedrich Köster, Michael Klaas, HanhQuyen Nguyen, Markus Braendle, Sebastian Obermeier, and Walter Brenner. Collaborative Security Assessments in Embedded Systems Development. In *International Conference on Security and Cryptography (SECRYPT 2009)*, 2009.

[91] Elias Levy. Crossover: online pests plaguing the off line world. *IEEE Security & Privacy*, 99(6):71–73, 2003.

[92] K. Sindhu and B. Meshram. Digital Forensic Investigation Tools and Procedures. In *International Journal of Computer Network and Information Security*, IJCNIS, April 2012. http://dx.doi.org/10.5815/ijcnis.2012.04.05.

[93] Md Nahid Hossain, Junao Wang, Ofir Weisse, R Sekar, Daniel Genkin, Boyuan He, Scott D Stoller, Gan Fang, Frank Piessens, and Evan Downing. {Dependence-Preserving} data compaction for scalable forensic analysis. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1723–1740, 2018.

[94] Amit Kumar, Gurinder Singh, Ankush Kansal, and Kulbir Singh. Digital image forensic approach to counter the jpeg anti-forensic attacks. *IEEE Access*, 9:4364–4375, 2021. 10.1109/ACCESS.2020.3048246.

[95] Adam Bates, Wajih Ul Hassan, Kevin Butler, Alin Dobra, Bradley Reaves, Patrick Cable, Thomas Moyer, and Nabil Schear. Transparent web service auditing via network provenance functions. In *Proceedings of the 26th International Conference on World Wide Web*, pages 887–895, 2017.

[96] Rajendra Bose and James Frew. Composing lineage metadata with xml for custom satellite-derived data products. In *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004.*, pages 275–284. IEEE, 2004.

[97] Kiran-Kumar Muniswamy-Reddy, David A Holland, Uri Braun, and Margo I Seltzer. Provenance-aware storage systems. In *Usenix annual technical conference, general track*, pages 43–56, 2006.

[98] Ashish Gehani and Dawood Tariq. Spade: Support for provenance auditing in distributed environments. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 101–120. Springer, 2012.

[99] Richard P Spillane, Russell Sears, Chaitanya Yalamanchili, Sachin Gaikwad, Manjunath Chinni, and Erez Zadok. Story book: An efficient extensible provenance framework. In *Workshop on the Theory and Practice of Provenance*, 2009.

[100] Faheem Zafar, Abid Khan, Saba Suhail, Idrees Ahmed, Khizar Hameed, Hayat Mohammad Khan, Farhana Jabeen, and Adeel Anjum. Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes. *Journal of network and computer applications*, 94:50–68, 2017.

[101] Wajih Ul Hassan, Mohammad Ali Noureddine, Pubali Datta, and Adam Bates. Omegalog: High-fidelity attack investigation via transparent multi-layer log analysis. In *Network and Distributed System Security Symposium*, 2020.

[102] Shiqing Ma, Juan Zhai, Fei Wang, Kyu Hyung Lee, Xiangyu Zhang, and Dongyan Xu. {MPI}: Multiple perspective attack investigation with semantic aware execution partitioning. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1111–1128, 2017.

[103] Adam Bates, Dave Jing Tian, Kevin RB Butler, and Thomas Moyer. Trustworthy {Whole-System} provenance for the linux kernel. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 319–334, 2015.

[104] Md Nahid Hossain, Sadegh M Milajerdi, Junao Wang, Birhanu Eshete, Rigel Gjomemo, R Sekar, Scott Stoller, and VN Venkatakrishnan. {SLEUTH}: Real-time attack scenario reconstruction from {COTS} audit data. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 487–504, 2017.

[105] Thomas Pasquier, Xueyuan Han, Thomas Moyer, Adam Bates, Olivier Hermant, David Eyers, Jean Bacon, and Margo Seltzer. Runtime analysis of

whole-system provenance. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1601–1616, 2018.

[106] Kyu Hyung Lee, Xiangyu Zhang, and Dongyan Xu. Loggc: garbage collecting audit log. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1005–1016, 2013.

[107] Wajih Ul Hassan, Lemay Aguse, Nuraini Aguse, Adam Bates, and Thomas Moyer. Towards scalable cluster auditing through grammatical inference over provenance graphs. In *Network and Distributed Systems Security Symposium*, 2018.

[108] Shiqing Ma, Juan Zhai, Yonghwi Kwon, Kyu Hyung Lee, Xiangyu Zhang, Gabriela Ciocarlie, Ashish Gehani, Vinod Yegneswaran, Dongyan Xu, and Somesh Jha. Kernel-supported cost-effective audit logging for causality tracking. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 241–254, 2018.

[109] Yutao Tang, Ding Li, Zhichun Li, Mu Zhang, Kangkook Jee, Xusheng Xiao, Zhenyu Wu, Junghwan Rhee, Fengyuan Xu, and Qun Li. Nodemerge: Template based efficient data reduction for big-data causality analysis. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1324–1337, 2018.

[110] Yushan Liu, Mu Zhang, Ding Li, Kangkook Jee, Zhichun Li, Zhenyu Wu, Junghwan Rhee, and Prateek Mittal. Towards a timely causality analysis for enterprise security. In *NDSS*, 2018.

[111] Zhang Xu, Zhenyu Wu, Zhichun Li, Kangkook Jee, Junghwan Rhee, Xusheng Xiao, Fengyuan Xu, Haining Wang, and Guofei Jiang. High fidelity data reduction for big data security dependency analyses. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 504–516, 2016.

[112] Wajih Ul Hassan, Shengjian Guo, Ding Li, Zhengzhang Chen, Kangkook Jee, Zhichun Li, and Adam Bates. Nodoze: Combatting threat alert fatigue with automated provenance triage. In *Network and Distributed Systems Security Symposium*, 2019.

[113] Xueyuan Han, Thomas Pasquier, Adam Bates, James Mickens, and Margo Seltzer. Unicorn: Runtime provenance-based detector for advanced persistent threats. *arXiv preprint arXiv:2001.01525*, 2020.

[114] Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, Xiao Yu, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, and Carl A Gunter. You are what you do: Hunting stealthy malware via data provenance analysis. In *NDSS*, 2020.

[115] Adam Bates and Wajih Ul Hassan. Can data provenance put an end to the data breach? *IEEE Security & Privacy*, 17(4):88–93, 2019.

[116] Giuliano Giova. Improving chain of custody in forensic investigation of electronic digital systems. *International Journal of Computer Science and Network Security*, 11(1):1–9, 2011.

[117] John R Vacca. *Computer Forensics: Computer Crime Scene Investigation (Networking Series)(Networking Series)*. Charles River Media, Inc., 2005.

[118] Max M Houck and Jay A Siegel. *Fundamentals of forensic science*. Academic Press, 2009.

[119] Yudi Prayudi and Azhari Sn. Digital chain of custody: State of the art. *International Journal of Computer Applications*, 114(5), 2015.

[120] Jasmin Cosic and Miroslav Baca. A framework to (im) prove" chain of custody" in digital investigation process. In *Central European Conference on Information and Intelligent Systems*, page 435. Faculty of Organization and Informatics Varazdin, 2010.

[121] Seymour Bosworth and Michel E Kabay. *Computer security handbook*. John Wiley & Sons, 2002.

[122] Rodney McKemmish. *What is forensic computing?* Australian Institute of Criminology Canberra, 1999.

[123] Slim Rekhis and Noureddine Boudriga. A system for formal digital forensic investigation aware of anti-forensic attacks. *IEEE transactions on information forensics and security*, 7(2):635–650, 2011.

[124] George Forman, Kave Eshghi, and Stephane Chiocchetti. Finding similar files in large document repositories. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 394–400. ACM, 2005.

[125] Muhammad Afzaal, Cesario Di Sarno, Luigi Coppolino, Salvatore D'Antonio, and Luigi Romano. A resilient architecture for forensic storage of events in critical infrastructures. In *2012 IEEE 14th international symposium on high-assurance systems engineering*, pages 48–55. IEEE, 2012.

[126] Ronald Rivest. The MD5 message-digest algorithm. Technical report, MIT Laboratory for Computer Science, 1992. URL `https://dl.acm.org/doi/pdf/10.17487/RFC1321`.

[127] Nicholas Mikus. An analysis of disc carving techniques. Master's thesis, Monterey, California. Naval Postgraduate School, 2005. visited on 2017-03-01.

[128] Anandabrata Pal and Nasir Memon. The evolution of file carving. *IEEE Signal Processing Magazine*, 26(2):59–71, 2009.

[129] Adam Bates, Kevin Butler, Andreas Haeberlen, Micah Sherr, and Wenchao Zhou. Let sdn be your eyes: Secure forensics in data center networks. In *Proceedings of the NDSS workshop on security of emerging network technologies (SENT'14)*, pages 1–7, 2014.

[130] Daniel Spiekermann and Tobias Eggendorfer. Challenges of network forensic investigation in virtual networks. *Journal of Cyber Security and Mobility*, pages 15–46, 2016.

[131] D. McPherson, R. Dobbins, Hollyman, M., C. Labovitzh, and J. Nazario. Worldwide infrastructure security report. `www.arbornetworks.com/report`, January 2010. visited on 2016-12-01.

[132] Fran Casino, Thomas K Dasaklis, Georgios P Spathoulas, Marios Anagnostopoulos, Amrita Ghosal, Istvan Borocz, Agusti Solanas, Mauro Conti, and Constantinos Patsakis. Research trends, challenges, and emerging topics in digital forensics: A review of reviews. *IEEE Access*, 10:25464–25493, 2022.

[133] Peter Sommer. Digital evidence. *Digital Investigations and E-Disclosure: A Guide to Forensic Readiness for Organizations, Security Advisers and Lawyers, The Information Assurance Advisory Council (IAAC),*, 2012.

[134] J Williams. Acpo good practice guide for digital evidence. *Metropolitan Police Service, Association of chief police officers, GB*, pages 1556–6013, 2012.

[135] RB Van Baar, HMA van Beek, and EJ van Eijk. Digital forensics as a service: A game changer. *Digital Investigation*, 11:S54–S62, 2014.

[136] Robin Verma, Jayaprakash Govindaraj Dr, Saheb Chhabra, and Gaurav Gupta. Df 2.0: An automated, privacy preserving, and efficient digital forensic framework that leverages machine learning for evidence prediction and privacy evaluation. *Journal of Digital Forensics, Security and Law*, 14(2):3, 2019.

[137] Gabriela Ahmadi-Assalemi, Haider M Al-Khateeb, Gregory Epiphaniou, Jon Cosson, Hamid Jahankhani, and Prashant Pillai. Federated blockchain-based tracking and liability attribution framework for employees and cyber-physical objects in a smart workplace. In *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, pages 1–9. IEEE, 2019.

[138] Irfan Ahmed, Sebastian Obermeier, Sneha Sudhakaran, and Vassil Roussev. Programmable logic controller forensics. *IEEE Security & Privacy*, 15(6):18–24, 2017.

[139] Tim Kilpatrick, Jesus Gonzalez, Rodrigo Chandia, Mauricio Papa, and Sujeet Shenoi. An architecture for SCADA network forensics. *Advances in digital forensics II*, pages 273–285, 2006.

[140] Tim Kilpatrick, Jesús Gonzalez, Rodrigo Chandia, Mauricio Papa, and Sujeet Shenoi. Forensic analysis of SCADA systems and networks. *International Journal of Security and Networks*, 3(2):95–102, 2008.

[141] Rodrigo Chandia, Jesus Gonzalez, Tim Kilpatrick, Mauricio Papa, and Sujeet Shenoi. Security strategies for SCADA networks. In *International Conference on Critical Infrastructure Protection*, pages 117–131. Springer, 2007.

[142] Mohamed Elhoseny, Hosny Abbas, Aboul Ella Hassanien, Khan Muhammad, and Arun Kumar Sangaiah. Secure automated forensic investigation for sustainable critical infrastructures compliant with green computing requirements. *IEEE Transactions on Sustainable Computing*, 5(2):174–191, 2020. 10.1109/TSUSC.2017.2782737.

[143] Craig Valli. *SCADA forensics with Snort IDS*. CSREA Press, 2009.

[144] Jih-Jeng Huang. Two Steps Genetic Programming for Big Data - Perspective of Distributed and High-Dimensional Data. In *2015 IEEE International Congress on Big Data (BigData Congress)*, pages 753–756. IEEE, 2015.

[145] Darren Quick and Kim-Kwang Raymond Choo. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation*, 11(4):273–294, 2014.

[146] Jay Koven, Enrico Bertini, Luke Dubois, and Nasir Memon. Invest: Intelligent visual email search and triage. *Digital Investigation*, 18:S138–S148, 2016.

[147] Christopher Stelly and Vassil Roussev. Nugget: A digital forensics language. *Digital Investigation*, 24:S38–S47, 2018. ISSN 1742-2876. https://doi.org/10.1016/j.diin.2018.01.006. URL `https://www.sciencedirect.com/science/article/pii/S1742287618300380`.

[148] Michael Zipperle, Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. Provenance-based intrusion detection systems: A survey. *ACM Computing Surveys*, 55(7):1–36, 2022.

[149] Abdulellah Alsaheel, Yuhong Nan, Shiqing Ma, Le Yu, Gregory Walkup, Z Berkay Celik, Xiangyu Zhang, and Dongyan Xu. Atlas: A sequence-based learning approach for attack investigation. In *USENIX Security Symposium*, pages 3005–3022, 2021.

[150] Yonghwi Kwon, Fei Wang, Weihang Wang, Kyu Hyung Lee, Wen-Chuan Lee, Shiqing Ma, Xiangyu Zhang, Dongyan Xu, Somesh Jha, and Gabriela F Ciocarlie. MCI: modeling-based causality inference in audit logging for attack investigation. In *NDSS*, volume 2, page 4, 2018.

[151] Shiqing Ma, Xiangyu Zhang, and Dongyan Xu. Protracer: Towards practical provenance tracing by alternating between logging and tainting. In *NDSS*, volume 2, page 4, 2016.

[152] Daniel Ayers. A second generation computer forensic analysis system. *digital investigation*, 6:S34–S42, 2009.

[153] Vassil Roussev and Golden Richard. Breaking the performance wall: The case for distributed digital forensics. In *Proceedings of the 2004 digital forensics research workshop*, volume 94, 2004.

[154] Lukas Daubner, Martin Macak, Barbora Buhnova, and Tomas Pitner. Towards verifiable evidence generation in forensic-ready systems. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2264–2269. IEEE, 2020.

[155] Ryan Harris. Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. *Digital Investigation*, 3:44–49, 2006. ISSN 1742-2876. https://doi.org/10.1016/j.diin.2006.06.005. URL `https://www.sciencedirect.com/science/article/pii/S1742287606000673`. The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[156] Hassan N Noura, Ola Salman, Ali Chehab, and Raphaël Couturier. Distlog: A distributed logging scheme for iot forensics. *Ad Hoc Networks*, 98:102061, 2020.

[157] Sameera Almulla, Youssef Iraqi, and Andrew Jones. A state-of-the-art review of cloud forensics. *Journal of Digital Forensics, Security and Law*, 9(4):2, 2014.

[158] Cristina Alcaraz, Isaac Agudo, David Nunez, and Javier Lopez. Managing incidents in smart grids a la cloud. In *2011 IEEE Third International Conference on Cloud Computing Technology and Science*, pages 527–531. IEEE, 2011.

[159] Martini and Choo. Cloud forensic technical challenges and solutions: A snapshot. *IEEE Cloud Computing*, 1(4):20–25, 2014. ISSN 2325-6095. 10.1109/MCC.2014.69.

[160] NIST Cloud Computing Forensic Science Working Group. Nist cloud computing forensic science challenges. Technical report, National Institute of Standards and Technology, 2014. URL `https://csrc.nist.gov/CSRC/media/Publications/nistir/8006/draft/documents/draft_nistir_8006.pdf`.

[161] Lori M Kaufman. Data security in the world of cloud computing. *IEEE Security & Privacy*, 7(4):61–64, 2009.

[162] Sameera Almulla, Youssef Iraqi, and Andrew Jones. Feasibility of Digital Forensic Examination and Analysis of a Cloud Based Storage Snapshot. *Journal of Digital Information Management*, 15(1), 2017.

[163] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie. Cloud Forensics. In *7th IFIP Advances in Digital Forensics VII, G. Peterson and S. Shenoi*, volume 361, pages 35–46, 2011.

[164] Haider Abbas, Christer Magnusson, Louise Yngstrom, and Ahmed Hemani. Addressing dynamic issues in information security management. *Information Management & Computer Security*, 19(1):5–24, 03 2011. ISSN 0968-5227. 10.1108/09685221111115836.

[165] M. P. Mohite and S. B. Ardhapurkar. Design and Implementation of a Cloud Based Computer Forensic Tool. In *2015 Fifth International Conference on Communication Systems and Network Technologies*, pages 1005–1009, April 2015. 10.1109/CSNT.2015.180.

[166] H.M.A. van Beek, J. van den Bos, A. Boztas, E.J. van Eijk, R. Schramp, and M. Ugen. Digital forensics as a service: Stepping up the game. *Forensic Science International: Digital Investigation*, 35:301021, 2020. ISSN 2666-2817. https://doi.org/10.1016/j.fsidi.2020.301021. URL `https://www.sciencedirect.com/science/article/pii/S2666281720300706`.

[167] Shams Zawoad, Amit Kumar Dutta, and Ragib Hasan. SecLaaS: Secure Logging-as-a-Service for Cloud Forensics. *CoRR*, abs/1302.6267, 2013. URL `http://arxiv.org/abs/1302.6267`.

[168] Shams Zawoad, Amit Kumar Dutta, and Ragib Hasan. Towards building forensics enabled cloud through secure logging-as-a-service. *IEEE Transactions on Dependable and Secure Computing*, 13(2):148–162, 2015.

[169] Bharat Manral, Gaurav Somani, Kim-Kwang Raymond Choo, Mauro Conti, and Manoj Singh Gaur. A systematic survey on cloud forensics challenges, solutions, and future directions. *ACM Comput. Surv.*, 52(6), November 2019. ISSN 0360-0300. 10.1145/3361216. URL `https://doi.org/10.1145/3361216`.

[170] Keyun Ruan and Joe Carthy. Cloud computing reference architecture and its forensic implications: A preliminary analysis. In *Digital Forensics and Cyber Crime*, pages 1–21. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[171] H. Hibshi, T. Vidas, and L. Cranor. Usability of forensics tools: A user study. In *2011 Sixth International Conference on IT Security Incident Management and IT Forensics*, pages 81–91, May 2011. 10.1109/IMF.2011.19.

[172] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, and Y. Pan. Stochastic Load Balancing for Virtual Resource Management in Datacenters. *IEEE Transactions on Cloud Computing*, PP(99):1–1, 2016. ISSN 2168-7161. 10.1109/TCC.2016.2525984.

[173] Alecsandru Patrascu and Victor-Valeriu Patriciu. c. *Journal of Control Engineering and Applied Informatics*, 16(1):80–88, 2014.

[174] C. Jackson, R. Agrawal, J. Walker, and W. Grosky. Scenario-based design for a cloud forensics portal. In *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–6, April 2015. 10.1109/THS.2015.7225260.

[175] Anand Kumar Mishra, Mahesh Govil, and Emmanuel Pilli. A taxonomy of hypervisor forensic tools. In *IFIP International Conference on Digital Forensics*, pages 181–199. Springer, 2020.

[176] S. Saibharath and G. Geethakumari. Cloud forensics: Evidence collection and preliminary analysis. In *2015 IEEE International Advance Computing Conference (IACC)*, pages 464–467, June 2015. 10.1109/IADCC.2015.7154751.

[177] A. Huseinović and S. Ribić. Virtual machine memory forensics. In *2013 21st Telecommunications Forum Telfor (TELFOR)*, pages 940–942, Nov 2013. 10.1109/TELFOR.2013.6716386.

[178] Yingxin Cheng, Xiao Fu, Xiaojiang Du, Bin Luo, and Mohsen Guizani. A lightweight live memory forensic approach based on hardware virtualization. *Information Sciences*, 379:23–41, 2017.

[179] Shuhui Zhang, Lianhai Wang, and Xiaohui Han. A kvm virtual machine memory forensics method based on vmcs. In *Computational Intelligence and Security (CIS), 2014 Tenth International Conference on*, pages 657–661. IEEE, 2014.

[180] Liu Guangqi, Wang Lianhai, Zhang Shuhui, Xu Shujiang, and Zhang Lei. Memory dump and forensic analysis based on virtual machine. In *Mechatron-*

ics and Automation (ICMA), 2014 IEEE International Conference on, pages 1773–1777. IEEE, 2014.

[181] Prakhar Sharma, Phillip Porras, Steven Cheung, James Carpenter, and Vinod Yegneswaran. Scalable microservice forensics and stability assessment using variational autoencoders. *arXiv preprint arXiv:2104.13193*, 2021.

[182] Christopher Stelly and Vassil Roussev. Scarf: A container-based approach to cloud-scale digital forensic processing. *Digital Investigation*, 22:S39–S47, 2017.

[183] S Saibharath and G Geethakumari. Design and Implementation of a forensic framework for Cloud in OpenStack cloud platform. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, pages 645–650. IEEE, 2014.

[184] Matúš Banas. *Cloud forensic framework for iaas with support for volatile memory.* PhD thesis, MS thesis, School of Computing, National College of Ireland, Dublin, Ireland, 2015.

[185] NIST. NIST Cloud Computing Forensic Science Challenges. `http://csrc.nist.gov/publications/drafts/nistir-8006/draft_nistir_8006.pdf`, note = visited on 2016-11-01, November 2016.

[186] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 468–477. IEEE, 2017.

[187] Kenny Awuson-David, Tawfik Al-Hadhrami, Mamoun Alazab, Nazaraf Shah, and Andrii Shalaginov. BCFL logging: an approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem. *Future Generation Computer Systems*, 122:1–13, 2021. ISSN 0167-739X. https://doi.org/10.1016/j.future.2021.03.001. URL `https://www.sciencedirect.com/science/article/pii/S0167739X21000807`.

[188] Prasad Purnaye and Vrushali Kulkarni. A comprehensive study of cloud forensics. *Archives of Computational Methods in Engineering*, 29(1):33–46, 2022.

[189] Patrick O'Callaghan. *Refining privacy in tort law.* Springer Science & Business Media, 2012.

[190] European Union. General data protection regulation. http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679,, 2016. visited on 2021-09-21.

[191] Asou Aminnezhad, Ali Dehghantanha, and Mohd Taufik Abdullah. A survey on privacy issues in digital forensics. *International Journal of Cyber-Security and Digital Forensics*, 1(4):311–324, 2012.

[192] Ali Dehghantanha and Katrin Franke. Privacy-respecting digital investigation. In *2014 Twelfth Annual International Conference on Privacy, Security and Trust*, pages 129–138. IEEE, 2014.

[193] Wynand van Staden. Protecting third party privacy in digital forensic investigations. In *IFIP International Conference on Digital Forensics*, pages 19–31. Springer, 2013.

[194] Frank YW Law, Patrick PF Chan, Siu-Ming Yiu, Kam-Pui Chow, Michael YK Kwan, KS Hayson, and Pierre KY Lai. Protecting digital data privacy in computer forensic examination. In *2011 Sixth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, pages 1–6. IEEE, 2011.

[195] Shuhui Hou, Tetsutaro Uehara, SM Yiu, Lucas CK Hui, and Kam-Pui Chow. Privacy preserving multiple keyword search for confidential investigation of remote forensics. In *2011 Third International Conference on Multimedia Information Networking and Security*, pages 595–599. IEEE, 2011.

[196] Shuhui Hou, Tetsutaro Uehara, Siu-Ming Yiu, Lucas CK Hui, and Kam-Pui Chow. Privacy preserving confidential forensic investigation for shared or remote servers. In *2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 378–383. IEEE, 2011.

[197] Bilal Shebaro and Jedidiah R Crandall. Privacy-preserving network flow recording. *digital investigation*, 8:S90–S100, 2011.

[198] Neil J Croft and Martin S Olivier. Sequenced release of privacy-accurate information in a forensic investigation. *Digital Investigation*, 7(1-2):95–101, 2010.

[199] Nurul Hidayah Ab Rahman, William Bradley Glisson, Yanjiang Yang, and Kim-Kwang Raymond Choo. Forensic-by-design framework for cyber-physical cloud systems. *IEEE Cloud Computing*, 3(1):50–59, 2016. 10.1109/MCC.2016.5.

[200] Jaco-Louis Kruger and Hein Venter. State of the art in digital forensics for the internet of things. In *International Conference on Cyber Warfare and Security*, pages 588–596. Academic Conferences International Limited, 2019.

[201] Asif Iqbal, Mathias Ekstedt, and Hanan Alobaidli. *Digital Forensic Readiness in Critical Infrastructures: A Case of Substation Automation in the Power Sector*, pages 117–129. Springer, 01 2018. ISBN 978-3-319-73696-9. 10.1007/978-3-319-73697-6_9.

[202] CESG. CESG Good Practice Guide No. 18. `http://www.nationalarchives.gov.uk/documents/information-management/forensicreadiness.pdf`, 2009. visited on 2021-10-14.

[203] Abdellah Akilal and M-Tahar Kechadi. An improved forensic-by-design framework for cloud computing with systems engineering standard compliance. *Forensic Science International: Digital Investigation*, 40:301315, 2022. ISSN 2666-2817. https://doi.org/10.1016/j.fsidi.2021.301315. URL `https://www.sciencedirect.com/science/article/pii/S2666281721002407`.

[204] George Grispos, William Bradley Glisson, and Kim-Kwang Raymond Choo. Medical cyber-physical systems development: A forensics-driven approach. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 108–113, 2017. 10.1109/CHASE.2017.68.

[205] Nhien-An Le-Khac, Daniel Jacobs, John Nijhoff, Karsten Bertens, and Kim-Kwang Raymond Choo. Smart vehicle forensics: Challenges and case study. *Future Generation Computer Systems*, 109:500–510, 2020. ISSN 0167-739X. https://doi.org/10.1016/j.future.2018.05.081. URL `https://www.sciencedirect.com/science/article/pii/S0167739X17322422`.

[206] Lukas Daubner and Raimundas Matulevičius. Risk-oriented design approach for forensic-ready software systems. *arXiv preprint arXiv:2106.10336*, 2021.

[207] Mohamed Elyas, Atif Ahmad, Sean B Maynard, and Andrew Lonie. Digital forensic readiness: Expert perspectives on a theoretical framework. *Computers & Security*, 52:70–89, 2015.

[208] Lucia De Marco, M-Tahar Kechadi, and Filomena Ferrucci. Cloud forensic readiness: Foundations. In *International Conference on Digital Forensics and Cyber Crime*, pages 237–244. Springer, 2013.

[209] Antonis Mouhtaropoulos, Panagiotis Dimotikalis, and Chang-Tsun Li. Applying a digital forensic readiness framework: Three case studies. In *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, pages 217–223. IEEE, 2013.

[210] Barbara Endicott-Popovsky, Deborah A Frincke, and Carol A Taylor. A theoretical framework for organizational network forensic readiness. *J. Comput.*, 2(3):1–11, 2007.

[211] Sipho Ngobeni, Hein Venter, and Ivan Burke. A forensic readiness model for wireless networks. In *IFIP International Conference on Digital Forensics*, pages 107–117. Springer, 2010.

[212] Khairul Akram Zainol Ariffin and Faris Hanif Ahmad. Indicators for maturity and readiness for digital forensic investigation in era of industrial revolution 4.0. *Computers & Security*, 105:102237, 2021.

[213] Mohamed Elyas, Sean B Maynard, Atif Ahmad, and Andrew Lonie. Towards a systemic framework for digital forensic readiness. *Journal of Computer Information Systems*, 54(3):97–105, 2014.

[214] Asif Iqbal, Mathias Ekstedt, and Hanan Alobaidli. Digital forensic readiness in critical infrastructures: A case of substation automation in the power sector. In *International Conference on Digital Forensics and Cyber Crime*, pages 117–129. Springer, 2017.

[215] Ahmed Alenezi, Hany F Atlam, and Gary B Wills. Experts reviews of a cloud forensic readiness framework for organizations. *Journal of Cloud Computing*, 8(1):1–14, 2019.

[216] Philip Turner. Unification of digital evidence from disparate sources (digital evidence bags). *Digital Investigation*, 2(3):223 – 228, 2005. ISSN

1742-2876. http://dx.doi.org/10.1016/j.diin.2005.07.001. URL http://www.sciencedirect.com/science/article/pii/S1742287605000575.

[217] Philip Turner. Selective and intelligent imaging using digital evidence bags. *Digital Investigation*, 3:59 – 64, 2006. ISSN 1742-2876. http://dx.doi.org/10.1016/j.diin.2006.06.003. URL http://www.sciencedirect.com/science/article/pii/S174228760600065X. The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[218] R Eaglin and JP Craiger. Data Sharing and the Digital Evidence Markup Language. In *1st Annual GJXDM Users Conference, Atlanta, GA.(not peer reviewed)*, 2005.

[219] Sang Su Lee, Tae-Sik Park, Sang-Uk Shin, Sung-Kyong Un, and Do-Won Hong. A new forensic image format for high capacity disk storage. In *Information Security and Assurance, 2008. ISA 2008. International Conference on*, pages 399–402. IEEE, 2008.

[220] Brian Neil Levine and Marc Liberatore. Dex: Digital evidence provenance supporting reproducibility and comparison. *Digital Investigation*, 6:S48 – S56, 2009. ISSN 1742-2876. http://dx.doi.org/10.1016/j.diin.2009.06.011. URL http://www.sciencedirect.com/science/article/pii/S1742287609000395. The Proceedings of the Ninth Annual DFRWS Conference.

[221] Simson Garfinkel. Digital forensics xml and the dfxml toolset. *Digital Investigation*, 8(3–4):161–174, 2012.

[222] Eoghan Casey, Greg Back, and Sean Barnum. Leveraging cybox™ to standardize representation and exchange of digital forensic information. *Digital Investigation*, 12:S102–S110, 2015.

[223] Wouter Alink, RAF Bhoedjang, Peter A Boncz, and Arjen P de Vries. Xiraf–xml-based indexing and querying for digital forensics. *digital investigation*, 3: 50–58, 2006.

[224] Raoul AF Bhoedjang, Alex R van Ballegooij, Harm MA van Beek, John C van Schie, Feike W Dillema, Ruud B van Baar, Floris A Ouwendijk, and Micha Streppel. Engineering an online computer forensic service. *Digital Investigation*, 9(2):96–108, 2012.

[225] Bradley Schatz. *Digital evidence: representation and assurance.* PhD thesis, Queensland University of Technology, 2007.

[226] Michael Cohen, Simson Garfinkel, and Bradley Schatz. Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary in formation and forensic workflow. *digital investigation*, 6: S57–S68, 2009.

[227] Andreas Moser and Michael I. Cohen. Hunting in the enterprise: Forensic triage and incident response. *Digital Investigation*, 10(2):89 – 98, 2013. ISSN 1742-2876. http://dx.doi.org/10.1016/j.diin.2013.03.003. URL `http://www.sciencedirect.com/science/article/pii/S1742287613000285`. Triage in Digital Forensics.

[228] Grant Osborne and Benjamin Turnbull. Enhancing computer forensics investigation through visualisation and data exploitation. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on*, pages 1012–1017. IEEE, 2009.

[229] Grant Osborne, Benjamin Turnbull, and Jill Slay. The" explore, investigate and correlate'(eic) conceptual framework for digital forensics information visualisation. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 629–634. IEEE, 2010.

[230] Christopher FR Tassone, Ben Martini, and Kim-Kwang Raymond Choo. Visualizing digital forensic datasets: A proof of concept. *Journal of forensic sciences*, 2017.

[231] Muhammad Irfan, Haider Abbas, Yunchuan Sun, Anam Sajid, and Maruf Pasha. A framework for cloud forensics evidence collection and analysis using security information and event management. *Security and Communication Networks*, 9(16):3790–3807, 2016. 10.1002/sec.1538. URL `http://dx.doi.org/10.1002/sec.1538`.

[232] Michael Aupetit, Yury Zhauniarovich, Giorgos Vasiliadis, Marc Dacier, and Yazan Boshmaf. Visualization of actionable knowledge to mitigate drdos attacks. In *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*, pages 1–8. IEEE, 2016.

[233] Omid Setayeshfar, Christian Adkins, Matthew Jones, Kyu Hyung Lee, and Prashant Doshi. Graalf: Supporting graphical analysis of audit logs for forensics. *Software Impacts*, 8:100068, 2021.

[234] Grafana. Grafana 3.1.0 released. `http://grafana.org/blog/2016/07/12/grafana-3-1-released.html`, 2016. visited on 2016-12-01.

[235] Lewes Technology Consulting. Lewes technology consulting. `https://github.com/philhagen/sof-elk/blob/master/VM_README.md`,, 2019. visited on 2021-07-21.

[236] Plaso. plaso. `https://plaso.readthedocs.io/en/latest/`,, 2019. visited on 2021-07-21.

[237] Irvin Homem. *Advancing Automation in Digital Forensic Investigations*. PhD thesis, Department of Computer and Systems Sciences, Stockholm University, 2018.

[238] Cristina Alcaraz and Sherali Zeadally. Critical infrastructure protection: Requirements and challenges for the 21st century. *International Journal of Critical Infrastructure Protection*, 8:53–66, 2015. ISSN 1874-5482. https://doi.org/10.1016/j.ijcip.2014.12.002. URL `https://www.sciencedirect.com/science/article/pii/S1874548214000791`.

[239] Martin Naedele. Addressing IT security for critical control systems. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pages 115–115. IEEE, 2007.

[240] Frank Adelstein. Live forensics: diagnosing your system without killing it first. *Communications of the ACM*, 49(2):63–66, 2006.

[241] Pedro Taveras. Scada live forensics: real time data acquisition process to detect, prevent or evaluate critical situations. *European Scientific Journal*, 9 (21), 2013.

[242] Hadeli Hadeli, Ragnar Schierholz, Markus Braendle, and Cristian Tuduce. Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. In *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on*, pages 1–8. IEEE, 2009.

[243] Keith Stouffer, Michael Pease, CheeYee Tang, Timothy Zimmerman, Victoria Pillitteri, and Suzanne Lightman. NIST SP800-82 R3 (draft) guide to operational technology (OT) security, apr 2022. URL `https://doi.org/10.6028%2Fnist.sp.800-82r3.ipd`.

[244] Roberto Minerva, Abyi Biru, and Domenico Rotondi. Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative, Torino, Italy*, 2015.

[245] Kazuhiko Isoyama, Yuji Kobayashi, Tadashi Sato, Koji Kida, Makiko Yoshida, and Hiroki Tagato. A scalable complex event processing system and evaluations of its performance. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 123–126, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1315-5. 10.1145/2335484.2335498. URL `http://doi.acm.org/10.1145/2335484.2335498`.

[246] Zakarya Drias, Ahmed Serhrouchni, and Olivier Vogel. Analysis of cyber security for industrial control systems. In *2015 international conference on cyber security of smart cities, industrial control system and communications (SSIC)*, pages 1–8. IEEE, 2015.

[247] László Monostori, Botond Kádár, Thomas Bauernhansl, Shinsuke Kondoh, Soundar Kumara, Gunther Reinhart, Olaf Sauer, Gunther Schuh, Wilfried Sihn, and Kenichi Ueda. Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2):621–641, 2016.

[248] Keke Gai, Longfei Qiu, Min Chen, Hui Zhao, and Meikang Qiu. SA-EAST: security-aware efficient data transmission for ITS in mobile heterogeneous cloud computing. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(2):60, 2017.

[249] Maria Stoyanova, Yannis Nikoloudakis, Spyridon Panagiotakis, Evangelos Pallis, and Evangelos K Markakis. A survey on the internet of things (iot) forensics: challenges, approaches, and open issues. *IEEE Communications Surveys & Tutorials*, 22(2):1191–1221, 2020.

[250] Infineon, NXP, STMicroelectronics, and ENISA. Common Position On Cybersecurity. `https://www.enisa.europa.eu/publications/enisa-position-papers-and-opinions/infineon-nxp-st-enisa-position-on-cybersecurity`, Dec 2016.

[251] Abdellah Chehri, Issouf Fofana, and Xiaomin Yang. Security risk modeling in smart grid critical infrastructures in the era of big data and artificial intelligence. *Sustainability*, 13(6):3196, 2021.

[252] Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Arif Ahmed, S.M. Ahsan Kazmi, and Choong Seon Hong. Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges. *Future Generation Computer Systems*, 92:265–275, 2019. ISSN 0167-739X. https://doi.org/10.1016/j.future.2018.09.058. URL `https://www.sciencedirect.com/science/article/pii/S0167739X18315644`.

[253] Jean-Paul A Yaacoub, Ola Salman, Hassan N Noura, Nesrine Kaaniche, Ali Chehab, and Mohamad Malli. Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and Microsystems*, 77:103201, 2020.

[254] Frank E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[255] Prasanta Gogoi, DK Bhattacharyya, Bhogeswar Borah, and Jugal K Kalita. A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, 54(4):570–588, 2011.

[256] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[257] Qiang Fu, Jian-Guang Lou, Yi Wang, and Jiang Li. Execution anomaly detection in distributed systems through unstructured log analysis. In *2009 ninth IEEE international conference on data mining*, pages 149–158. IEEE, 2009.

[258] Chee-Wooi Ten, Govindarasu Manimaran, and Chen-Ching Liu. Cybersecurity for critical infrastructures: Attack and defense modeling. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(4):853–865, 2010.

[259] João Henriques, Filipe Caldeira, Tiago Cruz, and Paulo Simões. Combining k-means and xgboost models for anomaly detection using log datasets. *Electronics*, 9(7):1164, 2020.

[260] ISO 27001 Security. ISO27k Toolkit, ISMS Auditing Guideline, Version 2, 2017 . `http://www.iso27001security.com/html/27007.html`, 2017. visited on 2017-06-01.

[261] Karen Kent, Suzanne Chevalier, Tim Grance, and Hung Dang. Guide to integrating forensic techniques into incident response. *NIST Special Publication*, 10:800–86, 2006. visited on 2017-06-01.

[262] North American Electric Reliability Corporation. NERC Cyber Security Standards. `https://www.nerc.com/pa/Stand/Pages/Cyber-Security-Permanent.aspx`, 2016. visited on 2023-08-04.

[263] Edward A Morse and Vasant Raval. Pci dss: Payment card industry data security standards in context. *Computer Law & Security Review*, 24(6):540–554, 2008.

[264] Hope Benefield, Glenn Ashkanazi, and Ronald H Rozensky. Communication and records: Hippa issues when working in health care settings. *Professional Psychology: Research and Practice*, 37(3):273, 2006.

[265] Elaine Hulitt and Rayford B Vaughn. Information system security compliance to fisma standard: a quantitative measure. *Telecommunication Systems*, 45 (2):139–152, 2010.

[266] Institute of Internal Auditors. International professional practices framework - implementation guide 2420 / quality of communications. Technical report, Institute of Internal Auditors. Research Foundation, 2013. URL `https://www.theiia.org/en/content/guidance/recommended/implementation/2420-quality-of-communications/`.

[267] Scott Donaldson, Stanley Siegel, Chris K Williams, and Abdul Aslam. *Enterprise cybersecurity: how to build a successful cyberdefense program against advanced threats.* Apress, 2015.

[268] Kathi Fisler, Shriram Krishnamurthi, Leo A Meyerovich, and Michael Carl Tschantz. Verification and change-impact analysis of access-control policies. In *Proceedings of the 27th international conference on Software engineering*, pages 196–205, 2005.

[269] Gail-Joon Ahn, Hongxin Hu, Joohyung Lee, and Yunsong Meng. Representing and reasoning about web access control policies. In *2010 IEEE 34th Annual Computer Software and Applications Conference*, pages 137–146. IEEE, 2010.

[270] Konstantine Arkoudas, Ritu Chadha, and Jason Chiang. Sophisticated access control via smt and logical frameworks. *ACM Transactions on Information and System Security (TISSEC)*, 16(4):1–31, 2014.

[271] Regner Sabillon, Jordi Serra-Ruiz, Victor Cavaller, and Jeimy Cano. A comprehensive cybersecurity audit model to improve cybersecurity assurance: The cybersecurity audit model (csam). In *2017 International Conference on Information Systems and Computer Science (INCISCOS)*, pages 253–259. IEEE, 2017.

[272] Rakesh Agrawal, Roberto Bayardo, Christos Faloutsos, Jerry Kiernan, Ralf Rantzau, and Ramakrishnan Srikant. Auditing compliance with a hippocratic database. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 516–527. VLDB Endowment, 2004. ISBN 0-12-088469-0. URL `http://dl.acm.org/citation.cfm?id=1316689.1316735`.

[273] Nesrine Kaaniche, Maryline Laurent, and Claire Levallois-Barth. Id-based user-centric data usage auditing scheme for distributed environments. *Frontiers in Blockchain*, 3:17, 2020.

[274] Mathieu Bouet and Maurice Israël. Inspire ontology handler: automatically building and managing a knowledge base for critical information infrastructure protection. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 694–697. IEEE, 2011.

[275] M. Lee, B. Hatfax, and J. Wingad. Critical function monitoring and compliance auditing system. `https://www.google.com/patents/US20070136814`, June 14 2007. US Patent App. 11/299,049.

[276] Sergeja Slapničar, Tina Vuko, Marko Čular, and Matej Drašček. Effectiveness of cybersecurity audit. *International Journal of Accounting Information Systems*, 44:100548, 2022.

[277] Kazi Wali Ullah, Abu Shohel Ahmed, and Jukka Ylitalo. Towards building an automated security compliance tool for the cloud. In *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pages 1587–1593. IEEE, 2013.

[278] M. Henze, R. Matzutt, J. Hiller, E. Mühmer, J. H. Ziegeldorf, J. v. d. Giet, and K. Wehrle. Practical Data Compliance for Cloud Storage. In *2017 IEEE International Conference on Cloud Engineering (IC2E)*, pages 252–258, April 2017. 10.1109/IC2E.2017.32.

[279] Frank Doelitzscher. *Security audit compliance for cloud computing*. PhD thesis, Plymouth University, 2014.

[280] Nikolaj Bjørner and Karthick Jayaraman. Checking cloud contracts in microsoft azure. In *International Conference on Distributed Computing and Internet Technology*, pages 21–32. Springer, 2015.

[281] IEC. IEC 62443-2-1: Industrial communication networks – Network and system security Part 2-1: Establishing an industrial automation and control system security program. `https://shop.austrian-standards.at/Preview.action?preview=&dokkey=334211&selectedLocale=en`, 2009.

[282] P. Mell and T. Grance. The NIST definition of cloud computing (NIST SP 800-145). `http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf`, 2011. visited on 2016-12-01.

[283] IEC. IEC 62443 - IEC Technical Specification - Industrial communication networks - Network and system security - Part 1-1: Terminology, concepts and models. `https://webstore.iec.ch/preview/info_iec62443-1-1%7Bed1.0%7Den.pdf`, 2017. visited on 2017-03-01.

[284] ISA SECURE. Establishment of ISASecure Japanese Scheme and Publication of ISASecure Embedded Device Security Assurance Certification Program Specifications in Japan. `http://www.isasecure.org/en-US/News-Events/Establishment-of-ISASecure-Japanese-Scheme-and-Pub`, April 2013. visited on 2016-12-01.

[285] Suryadipta Majumdar, Taous Madi, Yushun Wang, Yosr Jarraya, Makan Pourzandi, Lingyu Wang, and Mourad Debbabi. Security compliance auditing of identity and access management in the cloud: Application to openstack. In *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 58–65. IEEE, 2015.

[286] Amazon AWS. Security at scale: Logging in aws. `https://d1.awsstatic.com/whitepapers/compliance/AWS_Security_at_Scale`

_Logging_in_AWS_Whitepaper.pdf?did=wp_card&trk=wp_card, note = visited on 2022-01-17, 2022.

[287] Jose Manuel Torres, Finn Olav Sveen, and Jose Maria Sarriegi. Security strategy analysis for critical information infrastructures. In *International Workshop on Critical Information Infrastructures Security*, pages 247–257. Springer, 2008.

[288] Y Demchenko, P Grosso, C de Laat, and P Membrey. Addressing big data issues in Scientific Data Infrastructure. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 48–55, 2013. 10.1109/CTS.2013.6567203.

[289] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. *Big data: The next frontier for innovation, competition, and productivity.* McKinsey Global Institute, 2011.

[290] Osman Hegazy, Soha Safwat, and Malak El Bakry. A mapreduce fuzzy techniques of big data classification. In *SAI Computing Conference (SAI), 2016*, pages 118–128. IEEE, 2016.

[291] Alpaydin E. *Introduction to Machine Learning. 2nd ed.* MIT Press, Cambridge, MA, 2010.

[292] Hall MA Witten IH, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems.* Morgan Kaufmann, Amsterdam, 2011.

[293] R. Ranjan. Streaming big data processing in datacenter clouds. *IEEE Cloud Computing*, 1(1):78–83, May 2014. ISSN 2325-6095. 10.1109/MCC.2014.22.

[294] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 29–43, 2003.

[295] Jeffrey Dean and Sanjay Ghemawat. MapReduce: a flexible data processing tool. *Communications of the ACM*, 53(1):72–77, 2010.

[296] T White. Hadoop: the definitive guide: the definitive guide:"o'reilly media, inc.", 2009.

[297] Søren Kejser Jensen, Torben Bach Pedersen, and Christian Thomsen. Time series management systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(11):2581–2600, 2017.

[298] Miao-qiong Wang, Kai Wei, and Chun-yu Jiang. Survey of time series data processing in industrial internet. In *2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, pages 736–741. IEEE, 2019.

[299] EU. Protection of personal data. `http://ec.europa.eu/justice/data-protection/`,, 2017. visited on 2017-06-01.

[300] Iamwire. Big Data: 17 Predictions Everyone Should Read. `http://www.iamwire.com/2016/11/big-data-17-predictions-everyone-should-read/145040`, 2016. visited on 2016-12-01.

[301] Raffaella Brighi, Michele Ferrazzano, and Leonardo Summa. Legal issues in ai forensics: understanding the importance of humanware. *on Applications of AI to Forensics 2020 (AI2Forensics 2020)*, page 13, 2020.

[302] Kian Son Hoon, Kheng Cher Yeo, Sami Azam, Bharanidharan Shunmugam, and Friso De Boer. Critical review of machine learning approaches to apply big data analytics in ddos forensics. In *2018 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–5. IEEE, 2018.

[303] Ozlem Yavanoglu and Murat Aydos. A review on cyber security datasets for machine learning algorithms. In *2017 IEEE international conference on big data (big data)*, pages 2186–2193. IEEE, 2017.

[304] Nighat Usman, Saeeda Usman, Fazlullah Khan, Mian Ahmad Jan, Ahthasham Sajid, Mamoun Alazab, and Paul Watters. Intelligent dynamic malware detection using machine learning in ip reputation for forensics data analytics. *Future Generation Computer Systems*, 118:124–141, 2021.

[305] Rizky Tri Wiyono and Niken Dwi Wahyu Cahyani. Performance analysis of decision tree c4. 5 as a classification technique to conduct network forensics for botnet activities in internet of things. In *2020 International Conference on Data Science and Its Applications (ICoDSA)*, pages 1–5. IEEE, 2020.

[306] Ryan W. Moore and Bruce R. Childers. Automatic generation of program affinity policies using machine learning. In Ranjit Jhala and Koen De Bosschere, editors, *Compiler Construction*, pages 184–203, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37051-9.

[307] Andres Quiroz, Manish Parashar, Nathan Gnanasambandam, and Naveen Sharma. Autonomic policy adaptation using decentralized online clustering. In *Proceedings of the 7th international conference on Autonomic computing*, pages 151–160, 2010.

[308] Alejandro Pelaez, Andres Quiroz, and Manish Parashar. Dynamic adaptation of policies using machine learning. In *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 501–510, 2016. 10.1109/CCGrid.2016.64.

[309] Håvard D. Johansen, Eleanor Birrell, Robbert van Renesse, Fred B. Schneider, Magnus Stenhaug, and Dag Johansen. Enforcing privacy policies with meta-code. In *Proceedings of the 6th Asia-Pacific Workshop on Systems*, APSys '15, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335546. 10.1145/2797022.2797040. URL https://doi.org/10.1145/2797022.2797040.

[310] Omid Gheibi, Danny Weyns, and Federico Quin. Applying machine learning in self-adaptive systems: A systematic literature review. *ACM Trans. Auton. Adapt. Syst.*, 15(3), aug 2021. ISSN 1556-4665. 10.1145/3469440. URL https://doi.org/10.1145/3469440.

[311] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[312] Danny Weyns, Bradley Schmerl, Masako Kishida, Alberto Leva, Marin Litoiu, Necmiye Ozay, Colin Paterson, and Kenji Tei. Towards better adaptive systems by combining mape, control theory, and machine learning. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pages 217–223. IEEE, 2021.

[313] Alisha Asquith and Graeme Horsman. Let the robots do it!–taking a look at robotic process automation and its potential application in digital forensics. *Forensic Science International: Reports*, 1:100007, 2019.

[314] Dean Hayes and Michael Kyobe. The adoption of automation in cyber forensics. In *2020 Conference on Information Communications Technology and Society (ICTAS)*, pages 1–6. IEEE, 2020.

[315] Kevin C Moffitt, Andrea M Rozario, and Miklos A Vasarhelyi. Robotic process automation for auditing. *Journal of emerging technologies in accounting*, 15 (1):1–10, 2018.

[316] Robin Verma, Jayaprakash Govindaraj, and Gaurav Gupta. Df 2.0: Designing an automated, privacy preserving, and efficient digital forensic framework. In *Annual ADFSL Conference on Digital Forensics, Security and Law*, 2018.

[317] Alecsandru Patrascu and Victor-Valeriu Patriciu. Cyber protection of critical infrastructures using supervised learning. In *2015 20th International Conference on Control Systems and Computer Science*, pages 461–468. IEEE, 2015.

[318] Marin Litoiu, Ian Watts, and Joe Wigglesworth. The 13th cascon workshop on cloud computing: Engineering aiops. In *Proceedings of the 31st Annual International Conference on Computer Science and Software Engineering*, CASCON '21, page 280–281, USA, 2021. IBM Corp.

[319] IBM. Ibm pak for aiops, 2022. URL `https://www.ibm.com/cloud/cloud-pak-for-watson-aiop`. visited on 2022-09-01.

[320] Paolo Notaro, Jorge Cardoso, and Michael Gerndt. A systematic mapping study in aiops. In Hakim Hacid, Fatma Outay, Hye-young Paik, Amira Alloum, Marinella Petrocchi, Mohamed Reda Bouadjenek, Amin Beheshti, Xumin Liu, and Abderrahmane Maaradji, editors, *Service-Oriented Computing – ICSOC 2020 Workshops*, pages 110–123, Cham, 2021. Springer International Publishing. ISBN 978-3-030-76352-7.

[321] Zhenguo Chen and Yong Fei Li. Anomaly detection based on enhanced dbscan algorithm. *Procedia Engineering*, 15:178–182, 2011.

[322] H Asif-Iqbal, Nur Izura Udzir, Ramlan Mahmod, and Abdul Azim Abd Ghani. Filtering events using clustering in heterogeneous security logs. *Information Technology Journal*, 10(4):798–806, 2011.

[323] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. Unsupervised clustering approach for network anomaly detection. In *International Conference on Networked Digital Technologies*, pages 135–145. Springer, 2012.

[324] Albert J Hoglund, Kimmo Hatonen, and Antti S Sorvari. A computer host-based user anomaly detection system using the self-organizing map. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 5, pages 411–416. IEEE, 2000.

[325] Asif Iqbal Hajamydeen, Nur Izura Udzir, Ramlan Mahmod, and ABDUL AZIM ABDUL GHANI. An unsupervised heterogeneous log-based framework for anomaly detection. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(3):1117–1134, 2016.

[326] Gerhard Münz, Sa Li, and Georg Carle. Traffic anomaly detection using k-means clustering. In *GI/ITG Workshop MMBnet*, pages 13–14, 2007.

[327] Li Tian and Wang Jianwen. Research on network intrusion detection system based on improved k-means clustering algorithm. In *Computer Science-Technology and Applications, 2009. IFCSTA'09. International Forum on*, volume 1, pages 76–79. IEEE, 2009.

[328] Mohsen Eslamnezhad and Ali Yazdian Varjani. Intrusion detection based on minmax k-means clustering. In *Telecommunications (IST), 2014 7th International Symposium on*, pages 804–808. IEEE, 2014.

[329] Ravi Ranjan and G Sahoo. A new clustering approach for anomaly intrusion detection. *arXiv preprint arXiv:1404.2772*, 2014.

[330] Yihua Liao and V Rao Vemuri. Use of k-nearest neighbor classifier for intrusion detection. *Computers & security*, 21(5):439–448, 2002.

[331] Hussam Mohammed, Nathan Clarke, and Fudong Li. Evidence identification in heterogeneous data using clustering. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450364485. 10.1145/3230833.3233271. URL `https://doi.org/10.1145/3230833.3233271`.

[332] Adetokunbo Makanju, A Nur Zincir-Heywood, and Evangelos E Milios. Investigating event log analysis with minimum apriori information. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 962–968. IEEE, 2013.

[333] S Varuna and P Natesan. An integration of k-means clustering and naïve bayes classifier for intrusion detection. In *Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on*, pages 1–5. IEEE, 2015.

[334] Z Muda, W Yassin, MN Sulaiman, and N Udzir. K-means clustering and naive bayes classification for intrusion detection. *Journal of IT in Asia*, 4(1):13–25, 2016.

[335] Reda M Elbasiony, Elsayed A Sallam, Tarek E Eltobely, and Mahmoud M Fahmy. A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Engineering Journal*, 4(4):753–762, 2013.

[336] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1285–1298, 2017.

[337] Robert C Nickerson, Upkar Varshney, and Jan Muntermann. A method for taxonomy development and its application in information systems. *European Journal of Information Systems*, 22(3):336–359, 2013.

[338] Paulo Simões, Tiago Cruz, Jorge Gomes, and Edmundo Monteiro. On the use of Honeypots for detecting cyber attacks on industrial control networks. In *Proc. 12th Eur. Conf. Inform. Warfare Secur. ECIW 2013*, 2013.

[339] IRM. Amateurs attack technology. professional hackers target people. *Website article*, 2015.

[340] Yi Yang, Hai-Qing Xu, Lei Gao, Yu-Bo Yuan, Kieran McLaughlin, and Sakir Sezer. Multidimensional Intrusion Detection System for IEC 61850-Based SCADA Networks. *IEEE Transactions on Power Delivery*, 32(2):1068–1078, 2017.

[341] Luis Rosa, Tiago Cruz, Paulo Simões, Edmundo Monteiro, and L. Lev. Attacking SCADA systems: a practical perspective. In *IFIP/IEEE International Symposium on Integrated Network Management*, May 2017.

[342] Corinne Curt and Jean-Marc Tacnet. Resilience of critical infrastructures: Review and analysis of current approaches. *Risk analysis*, 38(11):2441–2458, 2018.

[343] Nabin Chowdhury and Vasileios Gkioulos. Cyber security training for critical infrastructure protection: A literature review. *Computer Science Review*, 40, 2021. ISSN 1574-0137. https://doi.org/10.1016/j.cosrev.2021.100361. URL `https://www.sciencedirect.com/science/article/pii/S1574013721000010`.

[344] Pawel Gromek. Strategic training and exercises for critical infrastructure protection and resilience: A transition from lessons learned to effective curricula. *International Journal of Disaster Risk Reduction*, 65:102647, 2021. ISSN 2212-4209. https://doi.org/10.1016/j.ijdrr.2021.102647. URL `https://www.sciencedirect.com/science/article/pii/S2212420921006087`.

[345] Luca Galbusera, Monica Cardarilli, Marina Gómez Lara, and Georgios Giannopoulos. Game-based training in critical infrastructure protection and resilience. *International Journal of Disaster Risk Reduction*, 78:103109, 2022. ISSN 2212-4209. https://doi.org/10.1016/j.ijdrr.2022.103109. URL `https://www.sciencedirect.com/science/article/pii/S2212420922003284`.

[346] THE WHITE HOUSE. NATIONAL SECURITY PRESIDENTIAL DIRECTIVE INSPD-54. `https://irp.fas.org/offdocs/nspd/nspd-54.pdf`, 2008. visited on 2021-10-19.

[347] EU Council. Council Directive 2008/114/EC. `https://eur-lex.europa.eu/eli/dir/2008/114/oj`, 2008. visited on 2023-05-09.

[348] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. Cyber-physical systems security—a survey. *IEEE Internet of Things Journal*, 4(6):1802–1831, 2017.

[349] ENISA. Communication network dependencies for ICS/SCADA Systems. `https://www.enisa.europa.eu/publications/ics-scada-dependencies/at_download/fullReport`, December 2016. visited on 2017-03-01.

[350] R. Morabito, J. Kjällman, and M. Komu. Hypervisors vs. lightweight virtualization: A performance comparison. In *2015 IEEE International Conference on Cloud Engineering*, pages 386–393, March 2015. 10.1109/IC2E.2015.74.

[351] Jorge Proença, Tiago Cruz, Edmundo Monteiro, and Paulo Simões. How to use software–defined networking to improve security—A survey. In *Proc. 14th Eur. Conf. Cyber Warfare Security (ECCWS)*, page 220, 2015.

[352] Nickson M Karie and Hein S Venter. Taxonomy of challenges for digital forensics. *Journal of forensic sciences*, 60(4):885–893, 2015.

[353] Hong Guo, Bo Jin, and Daoli Huang. Research and review on computer forensics. In *International Conference on Forensics in Telecommunications, Information, and Multimedia*, pages 224–233. Springer, 2010.

[354] Big Data. for better or worse: 90% of world's data generated over last two years. `https://www.sciencedaily.com/releases/2013/05/130522085217.htm`, 2013. visited on 2016-12-01.

[355] Yulai Xie, Dan Feng, Zhipeng Tan, and Junzhe Zhou. Unifying intrusion detection and forensic analysis via provenance awareness. *Future Generation Computer Systems*, 61:26–36, 2016.

[356] Colin Ware. *Information visualization: perception for design.* Morgan Kaufmann, 2019.

[357] Datavizcatalogue. Data visualization catalogue. `https://datavizcatalogue.com/`, 2022. visited on 2021-12-01.

[358] William Bradley Glisson, Tim Storer, and Joe Buchanan-Wollaston. An empirical comparison of data recovered from mobile forensic toolkits. *Digital Investigation*, 10(1):44–55, 2013.

[359] Sean Owen Sandy Ryza, Uri Laserson and Josh Wills. *Advanced Analytics with Spark, PATTERNS FOR LEARNING FROM DATA AT SCALE.* O'REILY, 2015.

[360] Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. Zero trust architecture. Technical report, National Institute of Standards and Technology, https://www.netskope.com/lp-everything-you-need-to-know-about-zero-trust-implementation, 2020.

[361] Florian Menges, Tobias Latzo, Manfred Vielberth, Sabine Sobola, Henrich C Pöhls, Benjamin Taubmann, Johannes Köstler, Alexander Puchta, Felix Freil-

ing, and Hans P Reiser. Towards GDPR-compliant data processing in modern SIEM systems. *Computers & Security*, 103:102165, 2021.

[362] João Henriques, Filipe Caldeira, Tiago Cruz, and Paulo Simões. A forensics and compliance auditing framework for critical infrastructure protection. *International Journal of Critical Infrastructure Protection*, 42:100613, 2023. ISSN 1874-5482. https://doi.org/10.1016/j.ijcip.2023.100613. URL `https://www.sciencedirect.com/science/article/pii/S1874548223000264`.

[363] João Henriques, Filipe Caldeira, Tiago Cruz, and Paulo Simões. An automated closed-loop framework to enforce security policies from anomaly detection. *Computers & Security*, page 102949, 2022.

[364] F. Caldeira, Schaberreiter T., Monteiro E. Varrette S., Simões P., Bouvry P., and Khadraoui D. Trust based interdependency weighting for on-line risk monitoring in interdependent critical infrastructures. *International Journal of Secure Software Engineering*, 2013.

[365] Luis Rosa, Miguel Borges de Freitas, João Henriques, Pedro Quitério, Filipe Caldeira, Tiago Cruz, and Paulo Simões. Evolving the security paradigm for industrial iot environments. In *Cyber Security of Industrial Control Systems in the Future Internet Environment*, pages 69–90. IGI Global, 2020.

[366] Nipun Jaswal. *Hands-On Network Forensics: Investigate network attacks and find evidence using common network forensic tools.* Packt Publishing Ltd, 2019.

[367] Nour Moustafa. A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. *Sustainable Cities and Society*, 72:102994, 2021.

[368] MITRE. Bypass user account control, 2017. URL `https://attack.mitre.org/techniques/T1548/002/`. visited on 2017-08-01.

[369] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, and S Yu Philip. Top 10 algorithms in data mining. *Knowledge and information systems*, 14 (1):1–37, 2008.

[370] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[371] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[372] Matthew Rocklin et al. Dask: Parallel computation with blocked algorithms and task scheduling. In *Proceedings of the 14th python in science conference*, volume 130, page 136. SciPy Austin, TX, 2015.

[373] Yu Zheng, Huichu Zhang, and Yong Yu. Detecting collective anomalies from multiple spatio-temporal datasets across different domains. In *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, pages 1–10, 2015.

[374] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.

[375] Peter Lichodzijewski, A Nur Zincir-Heywood, and Malcolm I Heywood. Host-based intrusion detection using self-organizing maps. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1714–1719. IEEE, 2002.

[376] Karlton Sequeira and Mohammed Zaki. Admit: anomaly-based data mining for intrusions. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 386–395. ACM, 2002.

[377] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[378] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.

[379] Andrea Vattani. K-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

[380] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 144–153. ACM, 2006.

[381] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[382] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[383] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors. *The annals of statistics*, 28(2):337–407, 2000.

[384] Rie Johnson and Tong Zhang. Learning nonlinear functions using regularized greedy forest. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):942–954, 2014.

[385] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, and Stuart Bowers. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9. ACM, 2014.

[386] Kaggle. Kaggle. `www.Kaggle.com`, 2018. visited on 2018-07-16.

[387] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[388] H2o framework for machine learning, 2020. URL `http://docs.h2o.ai/h2o/latest-stable/h2o-docs/index.html`. visited on 2020-02-15.

[389] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, and Sean Owen. Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1):1235–1241, 2016.

[390] NASA. NASA HTTP. `http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html`, 2018. visited on 2018-07-01.

[391] Pedro Sernadela, Lorena González-Castro, and José Luís Oliveira. SCALEUS: Semantic Web Services Integration for Biomedical Applications. *Journal of medical systems*, 41(4):54, 2017.

[392] Satya S Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau Jr, Sören Auer, Juan Sequeda, and Ahmed Ezzat. A survey of current approaches for mapping of relational databases to rdf. *W3C RDB2RDF Incubator Group Report*, pages 113–130, 2009.

[393] André Freitas, Edward Curry, Joao Gabriel Oliveira, and Sean O'Riain. Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. *IEEE Internet Computing*, 16(1):24–33, 2011.

[394] João Henriques, Filipe Caldeira, Tiago J. Cruz, and Paulo Simões. On the use of ontology data for protecting critical infrastructures. In *Proceedings of the 17th European Conference on Cyber Warfare and Security*, pages 208–216, 2018.

[395] João Henriques, Filipe Caldeira, Tiago J. Cruz, and Paulo Simões. On the use of ontology data for protecting critical infrastructures. *Journal of Information Warfare*, 17(4):38–55, 2018.

[396] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[397] Natalya F Noy and Deborah L McGuinness. Ontology development 101: A guide to creating your first ontology, 2001.

[398] D Brickley and R Guha. Resource description framework (rdf) schema specification (proposed recommendation), w3c (world wide web consortium)(1999). *See http://www. w3. org/TR/1999/PR-rdf-schema-19990303*, 3:12, 1999.

[399] Mark A Musen. Dimensions of knowledge sharing and reuse. *Computers and biomedical research*, 25(5):435–467, 1992.

[400] W3. RDF. `https://www.w3.org/RDF/a`, 2017. visited on 2017-02-01.

[401] Andy Seaborne, Axel Polleres, Lee Feigenbaum, and Gregory Todd Williams. Sparql 1.1 federated query. *W3C recommendation*, 2013.

[402] W3. SPARQL. `https://www.w3.org/TR/sparql11-query`, 2013. visited on 2016-12-01.

[403] Bob DuCharme. *Learning SPARQL: querying and updating with SPARQL 1.1.* " O'Reilly Media, Inc.", 2013.

[404] D2RQ. SPARQL. `<http://d2rq.org`, 2012. visited on 2017-08-31.

[405] C Bizer and R Cyganiak. D2RQ: Lessons learned, Position paper for the W3C, Workshop on RDF Access to Relational Databases, Cambridge, MA, US, 2007.

[406] W3. SPARQL. https://www.w3.org/TR/turtle, 2014. visited on 2016-12-01.

[407] Elisa Castorini, Paolo Palazzari, Alberto Tofani, and Paolo Servillo. Ontological framework to model critical infrastructures and their interdependencies. In *2010 Complexity in Engineering*, pages 91–93. IEEE, 2010.

[408] Joshua Blackwell, William J Tolone, Seok-Won Lee, Wei-Ning Xiang, and Lydia Marsh. An ontology-based approach to blind spot revelation in critical infrastructure protection planning. In *International Workshop on Critical Information Infrastructures Security*, pages 352–359. Springer, 2008.

[409] Michał Choraś, Rafał Kozik, Adam Flizikowski, and Witold Hołubowicz. Ontology applied in decision support system for critical infrastructures protection. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 671–680. Springer, 2010.

[410] Olaf Görlitz and Steffen Staab. Splendid: Sparql endpoint federation exploiting void descriptions. In *Proceedings of the Second International Conference on Consuming Linked Data-Volume 782*, pages 13–24. CEUR-WS. org, 2011.

[411] Bastian Quilitz and Ulf Leser. Querying distributed rdf data sources with sparql. In *European semantic web conference*, pages 524–538. Springer, 2008.

[412] Simon Schenk, Carsten Saathoff, Steffen Staab, and Ansgar Scherp. Semaplorer—interactive semantic exploration of data and media based on a federated cloud infrastructure. *Journal of Web Semantics*, 7(4):298–304, 2009.

[413] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. Fedx: Optimization techniques for federated query processing on linked data. In *International semantic web conference*, pages 601–616. Springer, 2011.

[414] H2020 atena project website, 2018. URL https://www.atena-h2020.eu/. visited on 2017-04-24.

[415] GSZSM ETSI. Zero-touch network and service management (zsm); reference architecture. Technical report, Tech. Rep, 2019. URL https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf.

[416] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A software architect's perspective.* Addison-Wesley Professional, 2015.

[417] Leticia Decker, Daniel Leite, Luca Giommi, and Daniele Bonacorsi. Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2020.

[418] Othmane Hireche, Chafika Benzaïd, and Tarik Taleb. Deep data plane programming and ai for zero-trust self-driven networking in beyond 5g. *Computer Networks*, 203:108668, 2022.

[419] Vijayaraghavan Murali, Letao Qi, Swarat Chaudhuri, and Chris Jermaine. Neural sketch learning for conditional program generation. *arXiv preprint arXiv:1703.05698*, 2017.

[420] Mohammad Riftadi, Jorik Oostenbrink, and Fernando Kuipers. Gp4p4: Enabling self-programming networks. *arXiv preprint arXiv:1910.00967*, 2019.

[421] Yuan Yuan and Wolfgang Banzhaf. Arja: Automated repair of java programs via multi-objective genetic programming. *IEEE Transactions on software engineering*, 46(10):1040–1067, 2018.

[422] João Henriques, Filipe Caldeira, Tiago Cruz, and Paulo Simões. Combining k-means and xgboost models for anomaly detection using log datasets. *Electronics*, 9(7), 2020. ISSN 2079-9292. 10.3390/electronics9071164. URL `https://www.mdpi.com/2079-9292/9/7/1164`.

[423] Payment Card Industry Security Standards Council. Payment Card Industry Data Security Standard - Requirements and Testing Procedures, v4.0, March 2022.

[424] OWASP. Owasp, 2022. URL `https://www.owasp.org`. visited on 2022-03-01.

[425] Robert C Seacord. *The CERT C secure coding standard.* Pearson Education, 2008.

[426] CWE. Common weakness enumeration, 2022. URL `https://cwe.mitre.org`. visited on 2022-03-01.

[427] CVE. Common vulnerabilities and exposures, 2022. URL `https://cve.mitre.org`. visited on 2022-03-01.

[428] NIST. National vulnerability database, 2022. URL `https://nvd.nist.gov/`. visited on 2022-03-01.

[429] NIST. Vulnerability metrics, 2022. URL `https://nvd.nist.gov/vuln-metrics/cvss`. visited on 2022-03-01.

[430] Donald Ervin Knuth. Literate programming. *The computer journal*, 27(2): 97–111, 1984.

[431] Akond Rahman, Rezvan Mahdavi-Hezaveh, and Laurie Williams. A systematic mapping study of infrastructure as code research. *Information and Software Technology*, 108:65–77, 2019. ISSN 0950-5849. https://doi.org/10.1016/j.infsof.2018.12.004. URL `https://www.sciencedirect.com/science/article/pii/S0950584918302507`.

[432] Kyverno, 2022. URL `https://kyverno.io/`. visited on 2022-04-10.

[433] Crossguard, 2022. URL `https://www.pulumi.com/crossguard/`. visited on 2022-04-10.

[434] Open policy agent, 2022. URL `https://www.openpolicyagent.org/`.

[435] Microsoft. Design azure policy as code workflows, 2022. URL `https://docs.microsoft.com/en-us/azure/governance/policy/concepts/policy-as-code`. visited on 2022-04-05.

[436] Sentinel, 2022. URL `https://www.hashicorp.com/sentinel`. visited on 2022-04-10.

[437] Consul. Sentinel in consul, 2022. URL `https://www.consul.io`. visited on 2022-04-01.

[438] Nomad. Nomad, 2022. URL `https://www.nomadproject.io`. visited on 2022-04-01.

[439] Madhusanka Liyanage. A survey on zero touch network and service management (zsm) for 5g and beyond networks. *Journal of Network and Computer Applications*, 203:103362, 2022. ISSN 1084-8045. 10.1016/j.jnca.2022.103362.

[440] Vault Project. Vault, 2022. URL `https://www.vaultproject.io/docs/enterprise/sentinel`. visited on 2022-04-01.

[441] Spam emails dataset, 2022. URL `https://www.kaggle.com/code/joaohenriques/prediction-spam-emails/data`. visited on 2022-04-10.

[442] William W. Cohen. Ernron email dataset, 2022. URL `https://www.cs.cmu.edu/~enron/`. visited on 2022-08-19.

[443] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[444] James Warren and Nathan Marz. *Big Data: Principles and best practices of scalable realtime data systems.* Simon and Schuster, 2015.

[445] Apache. Apache kafka: A high-throughput distributed messaging system. `http://kafka.apache.org`, 2016. visited on 2016-12-01.

[446] RabbitMQ. RabbitMQ Service Model. `http://www.rabbitmq.com/releases/rabbitmq-dotnet-client/v2.0.0/rabbitmq-dotnet-client-2.0.0-wcf-service-model.pdf`, March 2017. visited on 2017-03-01.