POLITECNICO DI TORINO

Master Degree course in Cybersecurity

Master Degree Thesis

The Threat Risk Assessor: A Novel Proactive Approach to Cyber Threats Management

Supervisors
Prof. Marco Mellia
Eng. David Francken

Candidate
Luca Antognarelli

Academic Year 2024-2025

Acknowledgements

I would like to dedicate this section to all the people who have accompanied me throughout my studies and those who have contributed to the completion of this thesis. First of all, I would like to express my gratitude to my family: my parents and my brother, who have constantly supported me throughout my long and adventurous educational journey, starting from primary school up to these last two years of university here in Turin.
This gratitude is therefore extended to my entire family, my uncles and aunts, and my grandparents, who, together with my family, have always given me the strength to pursue and achieve my goals.

A big thanks go to David Francken and Roberto Mazzolin, who made the development of this thesis possible by assisting me and creating the internship opportunity at Starion Group, whom I sincerely thank for allowing me to have this challenging experience. In this regard, I would also like to thank Vahid Joroughi, who, together with David, supervised me during my internship, providing me with valuable advice and excellent guidance.

I would now like to thank my closest friends: Giacomo, Umberto, Amber, Hafssa, Marta and Paola, as well as my colleagues: Andrea, Riccardo, Niccolò, Fabio, Francesca, Diego, Filippo, Francesco, Paolo, Lorenzo and Simone, with whom I have shared moments of hard work, during which we have always given each other our utmost support, alternating with moments of pure joy. The opportunity to share our ideas on several occasions has made this experience truly meaningful and deeply enriching.

Two special acknowledgements go to: Luca, with whom I have shared this adventure in Turin since the beginning, when it was just a chat during a lunch break while talking about future projects and dreams, until today when we will conclude our studies; and Giorgia, with whom I had the opportunity to discover and truly appreciate Turin, as well as share moments of study and hard work.

Finally, I would like to thank Professor Marco Mellia, who guided and supported me during the writing of my thesis, providing me with hints for reflection and improvement.

Thank you to everyone who contributed significantly to my academic journey and the completion of this thesis. You know how much I wanted this moment to arrive and how hard I worked to get here. Your positive encouragement was fundamental, and for this I will always be grateful.

Desidero dedicare questa sezione a tutte le persone che mi hanno accompagnato durante il percorso di studi e coloro che hanno contribuito al compimento di questa tesi.
In primo luogo, desidero ringraziare la mia famiglia: i miei genitori e mio fratello che mi hanno costantemente sostenuto e supportato durante tutto il mio lungo e movimentato percorso di studi, partendo dalle elementari fino a questi ultimi due anni di università qui a Torino. Questo ringraziamento viene quindi estenso anche a tutta la famiglia, i miei zii ed i miei nonni, che insieme alla mia famiglia mi hanno sempre dato la forza di perseguire e raggiungere i miei obiettivi.

Un grande ringraziamento va a David Francken e Roberto Mazzolin , che hanno reso possibile lo sviluppo di questa tesi, assistendomi e creando l'opportunità di tirocinio presso Starion Group, che ringrazio sinceramente per avermi concesso di svolgere questa esperienza stimolante. A questo proposito ringrazio anche Vahid Joroughi, che insieme a David mi ha seguito durante l'attività di tirocinio, regalandomi preziosi suggerimenti e un'ottima guida.

Voglio ora ringraziare i miei amici più stretti: Giacomo, Umberto, Amber, Hafssa, Marta e Paola, oltre che i miei colleghi: Andrea, Riccardo, Niccolò, Fabio, Francesca, Diego, Filippo, Francesco, Paolo, Lorenzo e Simone, con i quali ho avuto modo di condividere momenti di duro lavoro, durante i quali ci siamo sempre scambiati il massimo supporto, alternati ad altri di pura spensieratezza. L'opportunità di condividere le nostre idee in più occasioni, attraverso confronti e discussioni, ha reso quest'esperienza realmente significativa e profondamente arricchente.

Due ringraziamenti speciali vanno a: Luca, con il quale ho condiviso questa avventura a Torino fin dagli inizi, quando era solo una chiacchera fatta durante una pausa pranzo mentre si parlava di progetti e sogni futuri, fino ad oggi che concluderemo il nostro percorso di studi; e, a Giorgia, con la quale ho avuto modo di scoprire ed apprezzare davvero Torino, oltre che condividere momenti di studio e duro lavoro.

Infine, voglio ringraziare il Professore Marco Mellia che mi ha seguito e supportato durante la stesura della tesi, fornendomi spunti di riflessione e miglioramento.

Grazie a tutti coloro che hanno contribuito in modo significativo al compimento del mio percorso accademico e alla realizzazione di questa tesi; sapete quanto desideravo l'arrivo di questo momento e quanto duro ho lavorato per arrivarci. I vostri stimoli positivi sono stati fondamentali, e per questo ve ne sarò sempre grato.

Luca

Abstract

Over the past years, the growing digitalisation and interconnection of information systems has led to significant innovations. However, the direct consequence of this situation has been a significant increase in attack surfaces and related exploitable vulnerabilities. This has led to a significant growth of attacks, characterised by an increasing level of sophistication, guided also by multi-vector strategies.
In front of these new threats, classic approaches based on static and post-event intervention, have begun to show significant limitations. This scenario has led to the creation of the Threat Risk Assessor (TRA), a technology that aims to transform risk management through the introduction of predictive and proactive methodologies.
The key characteristics of the TRA include: continuous and adaptive monitoring of system behaviour, collection and correlation of Indicators of Compromise (IoC), contextual assessment of risk situations, structured generation and management of alerts and associated reports.
Another innovative feature is the Attacker Portal, which allows the execution of controlled simulations of complete attack chains. This makes it possible to assess the resilience of infrastructures and proactively identify vulnerabilities before they can be exploited by malicious actors. The final solution proposed is a modular system that is independent of the application context and capable of adapting to different operating environments without compromising effectiveness.
In summary, the TRA represents a step towards a dynamic and proactive approach to cybersecurity, capable of adapting to evolving threats. Furthermore, the experimental validation conducted has highlighted its potential contribution to strengthening the robustness of operational infrastructures, promoting the development of increasingly secure and resilient systems.

Contents

Chapter 1

Scenario

In recent years, the world of information technology has experienced an extraordinary transformation, characterised by increasing digitisation and interconnection of systems. Just think that today, over 4.9 billion people worldwide use the Internet, with billions of devices exchanging information [32].

These digital devices are now integrated into every aspect of daily life for every individual and every sector, from healthcare to public administration, finance and defence.
This evolution has certainly brought enormous advantages in terms of efficiency and innovation, but at the same time it has introduced new and complex challenges in the field of security. Previously, systems operated mainly in isolation, while today they are part of digital ecosystems distributed across the globe.
This interconnection between heterogeneous networks and devices also brings with it an increasingly extensive attack surface and new vulnerabilities that arise daily and are becoming increasingly difficult to manage.

Looking back at the past year (i.e. 2024), this was a year of explosive growth in cyber attacks, which impacted a wide range of sectors, from large businesses and public infrastructure to healthcare facilities, costing billions. According to the report provided by Check Point Research, in the second quarter of 2024 only, there was a 30% increase on an annual basis [48]. The striking figure is not only related to the number of attacks, but also to their evolution in terms of strategies. In fact, in addition to traditional purely technological vectors, social engineering[1] has played a dominant role, demonstrating that system security cannot reside solely in technological elements but must also be intrinsic to the human operators who use them.

[1]Social Engineering [62]: it is a cyberattack technique that does not focus solely on technical aspects, but exploits psychological manipulation of people to get them to perform actions they would not otherwise commit, such as sharing passwords or installing malicious software. Some examples are: Phishing, i.e. fraudulent emails disguised as coming from trusted senders (e.g., banks), aimed at stealing credentials or installing malware; Vishing, based on telephone calls in which the criminal pretends to be, for example, an authoritative body in order to extort information from the victim.

In this context, computer security can no longer be considered as a static or post-event activity; the increasing interconnectedness of digital systems brings with it new forms of vulnerabilities. Addressing them requires the use of dynamic, proactive approaches that can continuously adapt to the changing systems scenario.
Within this context, the **Threat Risk Assessor** (TRA) was born, a technology designed to revolutionise the paradigm of risk management within systems, transforming it from passive to reactive, predictive and resilient.

The TRA aims to present itself as an advanced risk analysis and management tool, designed to constantly monitor and react automatically and dynamically in the presence of threatening situation. This is because it does not only detect a risk or an anomaly, but is able to indicate timely and targeted countermeasures, providing a report that can guide system operators in applying defensive measures to deal with risk situations. These are tailored to the severity and context of the threat detected. The logic behind the technology is based on automation, learning and orchestration mechanisms, with the ultimate goal of minimising the impact of attacks and ensuring business continuity.

An additional innovative aspect of TRA is the possibility of executing planned and controlled attacks against the systems it monitors, via the **Attacker Portal**. This functionality, technically known as 'active threat simulation' [47], allows the continuous assessment of system resilience, enabling the identification of potential structural weaknesses before they can be exploited by a malicious actor.

Furthermore, the TRA is designed to be extremely modular and context agnostic. Although it was initially conceived in an area of research and applicability related to 5G Advanced infrastructure, during the development phase it was decided to generalise its functionalities, so as to make it applicable to a wide range of use-cases, from the public sector to defence and private industry. This approach brought with it an increase in the complexity of the architecture, but at the same time significantly increased the value and scalability of the solution, making it applicable also in scenarios where until now risk management was delegated to disconnected and poorly integrated tools.

1.1 TRA Key features

This section provides an initial overview of the key features of the Threat Risk Assessor, which include:

- Continuous and adaptive monitoring of systems, with behavioural and contextual threat detection capabilities
- Active simulation of attacks to validate and constantly update security measures
- Collection and analysis of Indicators of Compromise (IoC) and their correlation with triggering events
- Centralised management interfaces, designed to offer a complete view to operators, as well as interoperability with other security tools via webhooks[2] and Application Programming Interfaces[3] (APIs) proprietary to each tool
- Modularity and portability, enabling the integration of TRA into existing infrastructures, as well as its deployment in cloud environments
- Automated response to risk events, through a reporting engine, capable of providing a list of key actions to be performed in response to the dangerous situation.

1.2 Motivations behind the TRA

The birth of the TRA is closely related to the context of the 5G infrastructure, within which it is crucial to have a component in charge of continuously monitoring the security status of the system, assessing risk levels in real time and guiding response actions. The TRA's original idea was therefore to provide adaptive security, supporting predictive prevention mechanisms. In particular, it was supposed to exploit simulated risk scenarios to optimise the countermeasures to be implemented, in line with the continuous cycle of assessment, mitigation and validation.

However, already in the early stages of the project, the idea emerged that a technology with such potential could have a much broader impact if it was removed from a single application context. Its capabilities of adaptation, active threat simulation and response orchestration, proved to be applicable to a wide variety of application domains. This led to the intention to transform the TRA from a dedicated and vertical component, to a generic and agnostic technology, capable of operating in potentially any IT system requiring dynamic security and resilience.

[2]Webhooks [65]: these are customised HTTP callbacks and are triggered by specific events. These can be configured to trigger cascading events in an automated manner.

[3]API [52]: it is a software interface used to offer services to other software.

This choice consequently also had a significant impact on the engineering and design level, as it was necessary to develop modular interfaces and abstractions that would allow the TRA to be decoupled from the specific details of the 5G infrastructure, thus enabling its integration into any type of architecture. At the same time, considering the potential critical application environments, work was done to ensure a balance between automated processes and human control, leaving the operators with the final possibility of supervising and validating the actions indicated by the TRA, allowing specific policies and risk levels to be considered for each scenario.

1.2.1 Industrial context: Starion Group

This thesis project was carried out within Starion Group, a company operating in the space, defence and critical infrastructure sectors. The company's goal is to ensure operational continuity within these environment through the integration of innovative approaches such as Model-Based System Engineering[4] (MBSE) and the use of emerging technologies, like artificial intelligence and machine learning, integrated into solutions designed to provide cybersecurity measures.
Within this context, TRA is a strategic development project for Starion Luxembourg team, enabling the assessment of emerging risks and new attack surfaces resulting from the growing interconnection of the systems that Starion Group develops.

1.3 Literature review

Now that the scanario and the TRA motivations has been introduced, it is important to provide an overview of the current cybersecurity landscape with regard to threat monitoring solutions. Nowadays, passive detection is not enough, as limiting security breaches post-event is not effective; hence the focus is on increasingly dynamic, adaptive and proactive approaches, which have as their main objectives:

- Constantly monitor system behaviour in real time
- Constantly assess the risk based on Key Performance Indicator (KPI) values
- Simulate and reproduce attacks to test the resilience and effectiveness of the defensive measures implemented
- Orchestrate countermeasures in an automated manner or by guiding security department operators.

[4]MBSE [27]: is an approach to systems engineering that replaces traditional documentation with digital models as a means of describing, analysing and reporting on the system throughout its entire life cycle.

The TRA is exactly at this stage of the transaction, and is intended to be an innovative technology precisely because it is not a simple detection solution, but an integrated platform that combines in a continuous loop all the necessary objectives for effective threat detection.
Starting from this introduction, the next subsections will analyse the technologies that stimulated the creation of the initial TRA skeleton, showing how they are in themselves extremely independent, and how it was possible to create a unique and innovative technology.

1.3.1 Breach and Attack Simulation platforms

Breach and Attack Simulation (BAS) [47] represents a category of automated systems that emulate real attacks on IT infrastructures, in a cyclic and controlled manner.
These simulations are based on real attack techniques used by cyber criminals, in order to test defences and be prepared in the event of incidents and/or attacks.
These tools typically cover a wide range of attack vectors, such as data exfiltration, lateral movement, and privilege escalation. In addition, an important feature is detailed and continuous reporting, which is extremely useful for prioritising remediation actions.
It is important to note that these tools and the simulations within them typically refer to the MITRE ATT&CK®[5] framework.
An important feature of BAS platforms is the production of detailed reports that allow potential weaknesses or vulnerabilities to be identified and classified, thus providing a concrete basis for defining remediation strategies based on the priorities of each flaw found.

Taking inspiration from tools such as *Pentera* [45], the development of TRA's Attacker Portal went in that direction. In fact it allows automated attack simulations to be carried out to assess the resilience of the system. At the same time it differs significantly in that, since it allows this approach to be applied to any type of system, as the focus is on the freedom of creation and execution of payloads designed for white-hat team[6] activities.

These tools have not only taken inspiration from, but some of them have become integral parts of the technology, through the integration of their functionalities; this is the case of *Caldera* [35], a framework developed by MITRE Corporation to perform risk assessments in an automated manner.

[5]Adversarial Tactics, Techniques, and Common Knowledge (MITRE ATT&CK®): it is a framework, released in 2013 by Mitre Corporation, designed to classify and describe cyber attacks and their techniques, tactics and procedures.

[6]White-hat team: it is the team in charge of conducting regular security audits of IT infrastructures, through the identification of possible vulnerabilities and/or misconfigurations

1.3.2 Dynamic Risk Assessment framework

Dynamic Risk Assessment (DRA) [42] frameworks represent a major evolution compared to traditional risk analysis, which is generally static and heavily guided by human intervention. This manual approach, although effective in the past, is inadequate for modern systems, which are constantly subject to changes and updates, and consequently to new threats.
DRAs introduce an approach based on continuous monitoring, contextual analysis and dynamic threat estimation, providing a more realistic assessment. These systems are used today in areas such as critical infrastructures, transport networks, industrial plants and cloud environments.
The concept behind this framework is an integral part of TRA's risk evaluation engine, which however has some significant differences, as it does not rely on purely statistical or predictive models, but bases its logic on established lookup tables derived from well-established and codified standards, frameworks and scenarios. This allows for a much faster reaction time than computationally heavy models.

1.3.3 Continuous Exposure Management

Continuous Exposure Management (CEM) is a recent concept, created to address the critical issues of today's IT security, characterised by the persistent exposure of system attack surfaces, as they are increasingly distributed and interconnected, and traditional vulnerability scanning systems (mainly static) have proven ineffective.

The TRA shares with CEM the idea that monitoring and security must be continuous, proactive and contextual. In fact, as will be explained in the course of this thesis, thanks to the centralised log collection system, it is possible to have a real-time view of the attack surface, allowing it to be constantly monitored and consequently providing the possibility to anticipate and prevent potential malicious actions, guiding response operations in a timely and targeted manner.
A characteristic aspect of the TRA is its ability to adapt autonomously to the evolution of the attack surface, through the automatic identification of assets and the consequent potential updating of KPIs.

Chapter 2

Test-based environment

Considering the strong interest in the telecommunication sector and its key role in today's society, the environment exploited for the first phase of technology development and validation fell into the world of mobile connectivity management, particularly, the Advanced 5G Core Network (CN).
The 5G Core Netowrk (5GC) is the heart of the 5G architecture, as it allows managing user access and mobility, establishing and controlling data sessions, distributing traffic and applying Quality of Service (QoS) policies. It is in charge of supervising the management of data within the network, in particular: controlling access to the network, routing traffic, managing the connection of and between users, and guaranteeing the quality of service. In the course of this chapter, the following sections will present the elements of the CN that are responsible for performing each of these functions, as well as those that perform secondary and auxiliary functions that are nevertheless fundamental to the execution of the system.

2.1 Core Network evolution: 3G, 4G and 5G

CN has evolved significantly over time and with the succession of generations. In particular, the transition from 3G, 4G and 5G, has seen major changes reflecting technological advances and connectivity needs, thus significantly impacting network architecture and data management within it.

2.1.1 3G CN

The 3G network introduced a division between Circuit Switching (CS) and Packet Switching (PS). The first handled voice calls, establishing a dedicated connection for each call and managing this process through the Mobile Switching Centre (MSC) [31]. This made it possible to establish a dedicated path between the sender and the receiver for the duration of the communication, thus having the line between the two devices occupied only for that conversation, preventing other users from using the same resource until the end of the communication.
Over time, this approach will prove to be inflexible and not very scalable, as it is effective for voice communication, but not so effective for data transmission.

Moving on to data transmission, next comes PS, which supported data traffic and allowed IP packets to be sent. In this section of the CN, unlike the CS, data is broken down into small packets that travel separately in the network, only to be reassembled once they reach their destination. The main components that handled the management of the PS were:

- Serving GPRS[1] Support Node (SGSN): manages user mobility and traffic between the mobile device and the CN
- Gateway GPRS Support Node (GGSN): has the same functionality as a router, it routes data between the mobile network and external networks. It also establishes the connection between the mobile device and the IP networks

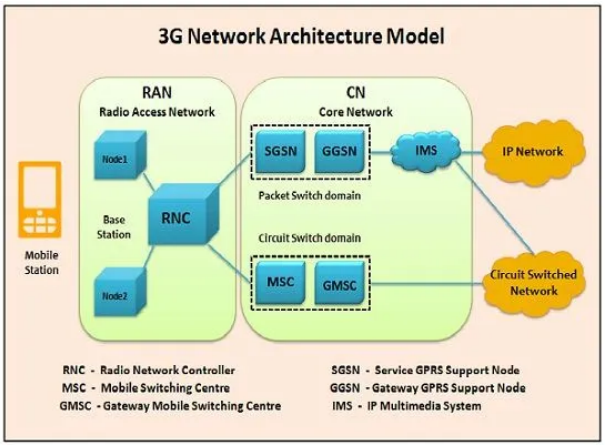

Figure 2.1: 3G Core Network Architecture (from [30])

[1]General Packet Radio Service (GPRS) [55]: it is a packet transmission service for Global System for Mobile Communications (GSM) networks, enabling data transmission between mobile devices and the network.

The architecture of the 3G core was thus a mix of CS and PS, with separate functionality for voice calls and data, representing an important transition from the past, but the real change would come with the evolution brought by 4G, which abandoned CS.

2.1.2 4G CN

As anticipated, the transition from 3G to 4G marked a revolution through the elimination of cirtuit switching, moving to the adoption of a completely IP-based network and packet-switching, with the goal of supporting faster, low-latency data traffic.
A key element introduced in this architecture is the eNodeB, which is the single access entity to the network, simplifying its management and allowing greater flexibility.

This architecture is characterised by the following components:

- Mobility Management Entity (MME): manages user mobility and sessions
- Serving GateWay (SGW): allows data exchange between the network and the access points to it
- Packet GateWay (PGW): manages the assignment of IP addresses and connects the mobile network to the Internet
- Home Subscriber Server (HSS): manages user authentication and stores user data
- Policy and Charging Rules Function (PCRF): determines the QoS rules and manages billing policies

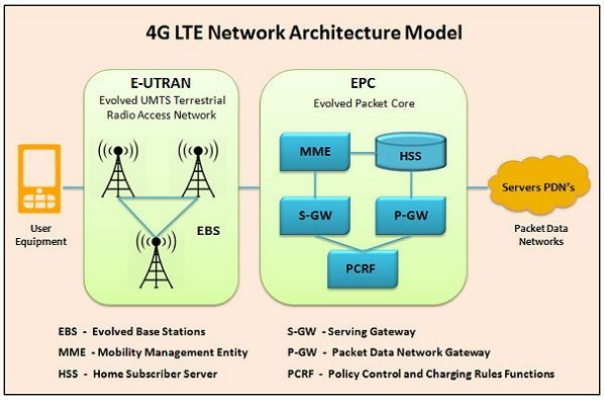

Figure 2.2: 4G Core Network Architecture (from [49])

2.1.3 5G CN

The evolution introduced by 5G marked a radical transformation in the architecture of mobile networks. Unlike previous networks, 5G has been designed from the beginning according to a service-based and fully IP-oriented approach, introducing an architecture called Service-Based Architecture (SBA).
Traditional components were replaced by virtualised modules, called Network Functions (NFs), which communicate through standardised APIs based on the HTTP/2 protocol and RESTful mechanisms, all referring to the 3GPP standard. This new service-oriented structure has made possible the adoption of cloud-native solutions such as network slicing, which enable the creation of multiple independent logical networks on a single physical infrastructure [29].
A further element to be taken into account is the possibility of integrating different access technologies, including Wi-Fi, through the Non-3GPP Interworking Function (N3IWF); this allows 5G CN to foster continuity of service in heterogeneous environments.
The functional components of the 5G Core include:

- User Plane Function (UPF): handles data forwarding and packet routing between user devices and external networks
- Network Repository Function (NRF): manages service discovery and registration of network functions in 5G
- Access and Mobility Management Function (AMF): manages connection and mobility management for users (i.e., registration, authentication, and handovers)
- AUthentication Server Function (AUSF): manages user authentication and security procedures during initial access.
- Network Slicing Selection Function (NSSF): selects and manages network slices for user sessions
- Policy Control Function (PCF): provides policy rules for managing data traffic, QoS, and charging in the network
- Session Management Function (SMF): handles session establishment, modification, and termination; it also manages user sessions and IP addresses
- Unified Data Management (UDM): stores subscription and user data, including authentication credentials and service profiles
- Unified Data Repository (UDR): central data repository for storing network-related data used by different 5GC functions
- Charging Function (CF): manages billing and charging-related processes based on user data and service usage

- User Equipment RAN SIMulator (UERANSIM): simulates the behaviour of User Equipment (UE) and Radio Access Network (RAN) interactions for testing purposes

- Non-3GPP InterWorking Function (N3IWF): enables interworking between 5G core and non-3GPP access networks (e.g., Wi-Fi)

- Non-3GPP InterWorking Edge (N3IWE): edge component for managing connections between non-3GPP networks and 5G systems

- Database: stores network and user data.

2.2 3rd Generation Partnership Project - 3GPP

The 3rd Generation Partnership Project (3GPP) is an international cooperation initiative aimed at defining, developing and maintaining technical standards for mobile communication systems [5]. It was set up with the aim of standardising the Universal Mobile Telecommunications System (UMTS), and over time has progressively expanded to include 4G, 5G and the future 6G technologies. It differs significantly from standardisation organisations since it has no independent legal personality, but acts as a technical partnership between seven regional standardisation organisations, called Organisational Partners. Each of these represents the geographical interests of each area and participates in the development of common standards; for Europe this is the European Telecommunications Standards Institute (ETSI). The partners coordinate the work through technical groups and working groups and the results of which are published under the name "Release".

As the focus for TRA development related to 5G and Advanced 5G technology, the main changes introduced for these by 3GPP are now presented.
Considering the increased demand for bandwidth and reduced latency, the first specifications for 5G were developed from Release 15, dating back to 2018 (with two late-drop in 2019) [1]. This one introduced a completely new, cloud-native CN, based on a service-based architecture and with full support for network slicing. The subsequent Release 16, dating back to 2020, extended the capabilities of 5G to support vehicular communications (V2X), optimisation of industrial networks and to provide greater energy efficiency, as well as providing functionality for precise localisation [2]. This Release thus laid the foundation for the adoption of 5G in scenarios beyond just mobile communications, including possible fields of application in Industry 4.0, advanced logistics and smart cities.
Release 18 marks the beginning of the first official phase of 5G Advanced [3], which further expands the range of use cases, introducing Non-Terrestrial Networks (NTNs) via satellite, as well as introducing ultra-high precision positioning.
Releases 19 and 20, on which the work of this thesis is based, aims to act as a bridge to 6G [4].

2.3 Open Source CN models

At the beginning of the thesis work, the basis for its development was prepared. In particular, an open-source tool called *Open5GS* was used as a use-case scenario.
This, and other tools such as *Free5GC* (which will be presented later) emerged due to the advent of 5G mobile networks, that are based on NF, and this characteristic stimulated the development of open-source solutions for modelling and implementing CN, each with its own implementation languages and peculiarities.

2.3.1 Open5GS

Open5GS is one of the most popular implementations. It is a project based on the C language [40], compatible with 3GPP standards, and is having a major impact in the academic and research fields, as well as for testing and simulations in development environments.
It implements the full spectrum of NFs envisaged by the 5G CN SBA, which are no longer monolithic, but well separated and interconnected via RESTful interfaces over the HTTP/2 protocol, favouring scalability and virtualisation. Furthermore, the architecture allows the creation of separate network slices, each with specific characteristics for each use case.

2.3.2 Free5GC

In the landscape of open-source 5G CN implementations, Free5GC represents one of the most complete solutions aligned to 3GPP standards. It was designed from the very beginning to support the SBA paradigm, acting as a reference tool for research activities, experimental validation and pre-commercial development of 5G networks.
All the NFs required by the 3GPP standard are implemented, with a special focus on the security standards imposed; in fact, it is possible to find provisions to implement the TLS protocol.
The components are realised in the Go language (Golang), and again the interaction between the NFs is via a RESTful API over the HTTP/2 protocol. Each of the functions can be executed as an independent microservice, enabling the containerisation of this technology. It is precisely for this reason, as well as the commercial aspect, that the work of this thesis focuses on this technology.
The installation process is presented below, as it requires certain technical precautions that are not indicated in any online guide, making the installation process complex. The aim of this guide is that it can be recreated in test and development environments easily.

Installation process

The installation took place in an Ubuntu 24.04.2 LTS (Noble Numbat) virtual machine, with Docker version 28.0.4 (build b8034c0) installed.

```
sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/
    keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

echo "deb␣[arch=$(dpkg␣--print-architecture)␣signed-by=/etc/apt/keyrings/
    docker.asc]␣\
https://download.docker.com/linux/ubuntu␣$(.␣/etc/os-release␣&&␣echo␣"
    $VERSION_CODENAME")␣stable" \
| sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-
    plugin docker-compose-plugin

sudo usermod -aG docker $USER

docker version

docker compose version
```

Listing 2.1: Docker installation

Next, it is essential to install a Linux kernel module called `gtp5g`, which enables packet management via Packet Forwarding Control Protocol (PFCP), necessary for 5G CN operation.

```
sudo apt install make gcc-13
git clone https://github.com/free5gc/gtp5g.git
cd gtp5g
git checkout v0.8.10
```

Listing 2.2: Kernel module download

Now, to ensure the correct installation and compatibility of all elements, the `Makefile` and `src/gtpu/pktinfo.c` files must be modified, it is fundamental to import all required libraries, this may require a manual intervention.

```
sudo make clean
sudo make
sudo make install
```

Listing 2.3: Kernel module installation

Then proceed to download the FreeGC docker-compose file from the official repository and run the script to start the system.

```
git clone https://github.com/free5gc/free5gc-compose.git
cd free5gc-compose
git checkout v3.4.2
docker compose up -d
docker compose ps -a
```

Listing 2.4: Free5FC startup through docker-compose

Chapter 3

Threat Risk Assessor - TRA

Carrying on the discourse begun in the first chapter of this thesis, the TRA was created as an advanced tool for risk analysis and response, by constantly monitoring systems and guiding the automatic activation of targeted countermeasures in the presence of threat conditions. Unlike traditional monitoring systems, the TRA is not limited to threat detection. Thanks to the Attacker Portal, the system is able to conduct controlled simulations of single threats and complete attack chains, following the active threat simulation paradigm, in order to continuously test the resilience of the monitored infrastructures over time.

The development of the TRA did not start from scratch, but benefited from an initial experimental structure from the very beginning, designed to verify and validate the design idea. This initial infrastructure was divided into two main components:

- A test-based environment, based on the containerised implementation of the 5GC provided by *Open5GS*, in which each functional element (AMF, SMF, UPF, etc.) was equipped with a *Prometheus* client, responsible for collecting the related metrics
- An initial monitoring system, composed by: the *Prometheus* server, a dashboard for visualising the metrics based on *Grafana* and an Alertmanager module for managing alerts sent through filters on thresholds set on *Grafana*. This architecture made it possible to collect the CN data, verify the correct observation-reaction cycle and set the basis for the subsequent evolution of the system.

The validation of this prototype was a fundamental step, as it enabled the birth of this thesis and the development of the TRA, as well as providing many ideas for its expansion. It is precisely from these ideas and this basis that three evolutionary versions of the system were subsequently designed and realised, culminating in the first production version adopted and validated through implementation in two projects.

In the course of this chapter, the technical evolution of the TRA, from the prototyping phase up to its operational adoption, will be set out in a clear and structured manner; the following aspects will be described in detail:

- The initial version of the system and its features
- The two subsequent versions, with particular attention to the solutions adopted for the orchestration and automation of the detection mechanisms, as well as the reasons behind the evolutionary choices between the different versions.

In addition, for each phase, there will also be a subsection dedicated to the tools used, described both in terms of design and functionally.

3.1 TRA Version 1

The first operational version of the Threat Risk Assesor is not part of the work related to the development of this thesis, considering that Starion Luxembourg was already working on TRA's development, but is nevertheless included since it made it possible to verify the technical validity of the concept underlying the project.

The initial infrastructure, shown in Figure 3.1, was organised into two Virtual Machines (VMs), each with a complementary role. The first, which represented the heart of this version, held all the monitoring and visualisation tools: *Prometheus* for collecting metrics, *Grafana* for visualisation, *Alertmanager* for managing notifications and a *Flask* server configured to receive alerts via webhook. In addition to these, there was *Loki*, used as a log aggregation engine.
The second VM was instead intended for the network-related component, where the *Open5GS*-based 5GC and the UERANSIM radio network and terminal simulator were installed. Each component of this setup was configured to expose Prometheus-compatible metrics, made accessible through dedicated HTTP endpoints. In parallel, the generated logs were collected by *Promtail*, configured to forward them to *Loki* on the monitoring VM.
In addition, the *5Greplay* tool was installed in this VM, which was used to simulate attack scenarios will be presented in section 4.1. Using the presented tool, a DoS situation was simulated against AMF, using malformed Next Generation Application Protocol (NGAP) packets, wanting to create errors within the system to see the results reflected in the logs and metrics. Through this attack scenario, it was possible to validate the detection system, which first allowed metrics and logs to be visualised within *Grafana*'s interface, and then notified in real time via *Alertmanager* and *Flask*, allowing a first concrete validation of the designed architecture.

To conclude and summarise, even if it had some due to the use of ready-to-use tools, this version represents a milestone in the development of the system: it showed the possibility of collecting data, processing them and being able to conveniently identify critical situations and then activate alert mechanisms in a coherent and synchronised manner.

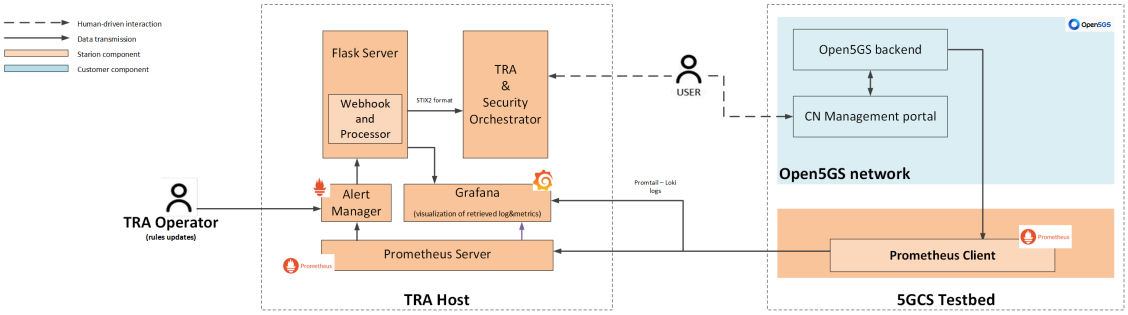

Figure 3.1: TRA Version 1 - Architecture

3.1.1 Tools

The test environment was based on two virtual machines, within which all the above-mentioned tools were orchestrated through respective *Docker Compose*[1]. This approach made it possible to keep the services separate from each other, facilitating their configuration and maintenance, and allowing for good scalability of resources.
A further aspect to be noted is the fact that the software tools implemented are all open-source. This is because in a deployment perspective it allows for important flexibility and freedom of development in terms of compatibility with the systems to be monitored, ensuring ease of integration, operational visibility and support for the security of the software itself coming from the support of the communities behind each of these tools.

Prometheus

Prometheus [43] has been adopted as a metrics collection tool due to its ability to poll HTTP endpoints exposed by the monitored NF services, and then store the resulting data in a time-series database. *Open5GS* offers native support for *Prometheus* by default, exposing metrics through RESTful endpoints for each NFs.
Through the `prometheus.yml` file, it is possible to configure the server to periodically query NFs, thus collecting KPIs, which are used to build the first layer of information for anomaly detection.

[1]Docker Compose: it is a tool for defining and managing applications based on multi-containers *Docker*, automating the processes of building, running and linking the different services provided by each [13].

In addition, *Prometheus* allows alerting rules to be defined in the `rules.yml` file, containing predefined conditions such as thresholds and sudden deviations, which can generate events in the case of abnormal behaviour.

Grafana

Grafana [26] has been used for the visual representation of the collected data; it is an open-source dashboarding tool. Data sources can be defined within it, and for this case Prometheus and *Loki* were used as data sources.
Thanks to the possibility of correlating metric and logical data by exploiting the different data sources, customised dashboards were developed. Thanks to these it has been possible to visualise, in real time, the performance of the metrics and related logs of each NF.
This made it possible, for instance, to visualise the increase in NGAP errors as a result of the attack using 5Greplay.

Alertmanager

Alertmanager [44] is a native *Prometheus* tool for receiving, classifying and managing alert notifications. Whenever a rule is violated, this tool can be used to generate events. In this case, *Alertmanager* was configured to send notifications via webhook to a Flask server, used as a prototype interface, simulating TRA logic.

Loki & Promtail

Loki [23] has been used for log analysis and management; it is a log aggregation system developed by Grafana Labs. In contrast to Prometheus, which focuses on numerical metrics, this allows textual logs to be collected and indexed in raw format.
In the test environment presented, it was necessary to forward these logs from the *Open5GS* to the monitoring VM; to do this, *Promtail* [24] was installed, an agent that reads the logs and forwards them to the *Loki* server via HTTP push.

Flask server

Flask [17] server allows the receipt of alerts via webhook and has been developed to serve as an endpoint for alerts generated by *Alertmanager*, representing a first draft of the TRA risk analysis module.

3.2 TRA Version 2

With this version, the project entered an evolving phase, aimed at increasing the flexibility and completeness of the solution. The new architecture therefore abandons the approach centred exclusively on *Prometheus*, to introduce a more articulated pipeline, centred on two macro-components: **TRA Agent** and **TRA Host**.
This has made it possible to achieve a high level of abstraction with respect to the monitored system, allowing the technology to be completely agnostic with regard to the source of the data entering the system, thus achieving the first of the major objectives that this technology aspired to during the design phases.

3.2.1 TRA Agent & TRA Host

The **TRA Agent** is represented by an ad-hoc developed application based on *Node.js* [39] and placed inside a *Docker* container. This has the task of periodically collecting both the metrics (Central Processing Unit (CPU) usage, Random Access Memory (RAM) usage, network I/O) and the structured logs provided by each monitored element, containing: the timestamp, the log level, the protocol involved and a semantic description.
This data was then made available through appropriate endpoints for graphical visualisation by means of *Grafana*, which, thanks to the Infinity plug-in [25], periodically queries these endpoints and enables the visualisation of metrics and logs.

In addition to being displayed on *Grafana*, this data is forwarded to the **TRA Host**, the logical heart of this version of TRA. This too is an application developed in *Node.js* and integrated into a *Docker* container.
It periodically receives streams of metrics and logs, which are then compared with a lookup table, in this case pre-configured, which contains rules for matching observed parameters with known behaviour: a reference value is defined for each condition and a flag indicating whether the event should generate an alert.
When an abnormal condition is detected, such as a critical threshold being exceeded, the **Lookup Agent** does not send the data to *Prometheus*, but interacts directly with *Alertmanager*, which handles the propagation of the event via an external notification sent with *Slack*, and towards the risk processing system.

The reason behind the introduction of the look-up table is double: on the one hand, it centralises the interpretation logic, and on the other hand, it allows the system to be modified when necessary without having to recompile the code, as new conditions can be directly added to the table.

At this point, the **Scoring Module** comes into play, which analyses the context and, based on the alert received, assesses the severity of the event and provides a numerical value to the identified risk. This element represents the main innovation of this development phase; for this reason, it will be presented in a specific subsection.

From a practical point of view, this has resulted in the installation of the two modules, with their respective containers, in two separate virtual machines:

- **TRA Agent**: focused on the collection and local aggregation of the monitored system's data, as well as on the controlled execution of attack scenarios
- **TRA Host**: represents the central brain of the system, responsible for risk assessment, notification and report management.

Wanting now to emphasise the peculiarities, compared to the first version, this new architecture is distinguished by three fundamental aspects:

- Functional decoupling: where previously *Prometheus* managed both polling and alerting, now the responsibilities are clearly separated, as the **Metrics Agent** takes care of collection, the **Lookup Agent** of interpretation and the *Alertmanager* of notification. This reduces the workload on each element, as well as the propagation of any errors
- Cohesion: each module deals with a single functionality, but the system as a whole operates in an extremely cohesive manner
- Flexibility in risk assessment: thanks to the use of lookup tables, it is possible to model new scenarios without modifying the entire system. it is sufficient to add new rules to update the system's behaviour with respect to new threats. In addition, the scoring module, being based on a standard, ensures that you always have a relevant response to the situation being analysed.

The described system can be observed in the Figure 3.2.

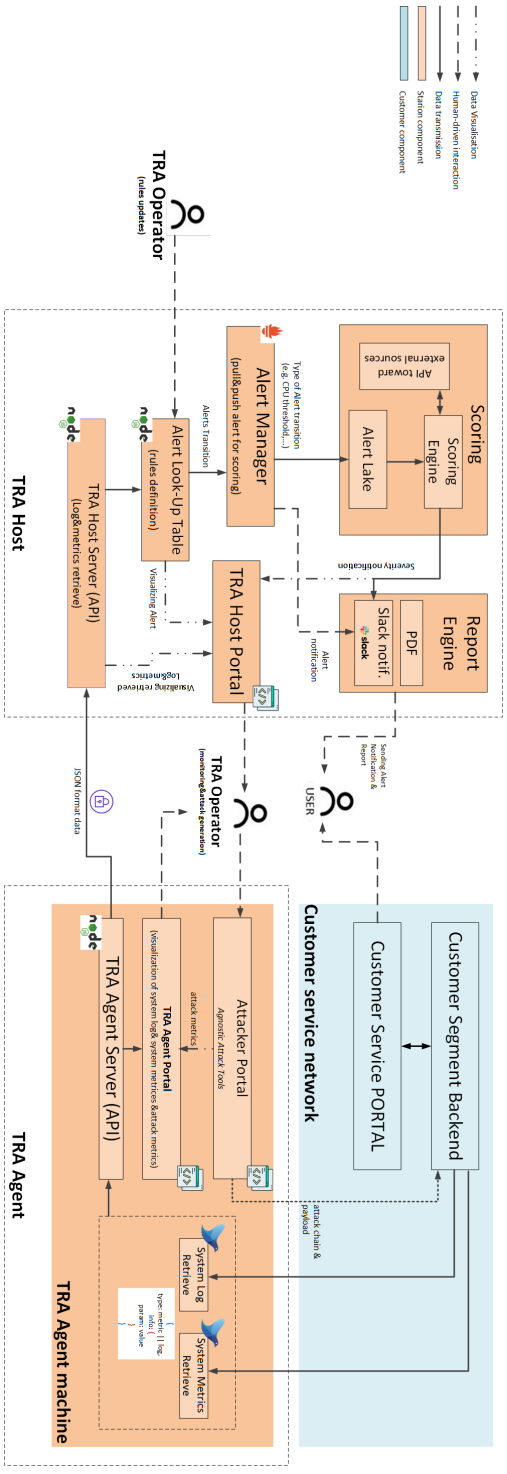

Figure 3.2: TRA Version 2 - Architecture

3.2.2 Data collection and normalisation

A critical point in the previous version was the simultaneous collection of metrics and logs from numerous elements within the monitored system, since the number of elements increased, the system implemented began to show its first criticalities in terms of flexibility and automatic adaptability for data aggregation and collection. Indeed, given the high volume of data constantly arriving, pieces of information, such as individual logs or metrics, were being lost. Such behaviour is certainly not acceptable considering the criticality of the systems for which it was designed; every piece of data is valuable and must be properly transmitted and processed.
To address this complexity, *Fluentd* [18] was introduced, a tool specifically designed for the centralised collection of heterogeneous data from different sources. This made it possible to:

- Aggregate metrics and logs from system components
- Transform the received information into a standardised JSON format, with the possibility of adding and managing additional metadata
- Send this data to the **TRA Agent Server** via the `/fluent-data` endpoint, to allow the first visualisation and processing step.

This solution guarantees consistency in format, relieving the **Agent** of the computation of formatting, allowing a simple and immediate connection of any system to the TRA. It also ensures dynamic addition or removal of components in the monitored system, which has no impact on the TRA's internal logic.
The **TRA Agent Server** receives the data processed by *Fluentd* and makes it available in real time to the **TRA Agent Portal**, a web-app dedicated to local KPI visualisation. Through this, the operator can easily and intuitively observe the status of the system, with graphs and tables broken down by log level, protocol, function and message.
In parallel, it forwards this formatted data to the **TRA Host**, via the endpoint `/monitor` exposed by the **TRA Host Server**.

3.2.3 Information flow

To fully understand how this version works, it is now necessary to explain in detail the flow of information with respect to the role of the components in the TRA architecture. The pipeline is designed to ensure consistency and cohesion between data collection, data analysis and alert activation. Each component within it plays a specific role, and this allows for an information flow that constitutes a continuous, automated and contextual cycle of detection and response.
In complex and sensitive contests like those for which TRA is designed, such as the space sector, the importance of having a well-defined data flow is a necessary condition to ensure reliability and auditability.

Moving on now to the more theoretical aspect of understanding the importance behind a pipeline, this can be seen as an ordered sequence of transformations applied to a raw data stream, up to the translation of this into a decision action. Within this flow, each step has a well-defined responsibility, following the principle of single-responsibility [61], which favours internal cohesion between the components and isolation in the event of errors. In the following subsections, the practical translation of this definition will be provided, allowing the concept to be fully assimilated.

In this version, data are generated within each component of the monitored system, where the *Fluentd* instance is located. This is responsible for collecting both metrics and logs, as shown in the Figure 3.3 below. It accesses this information at the container level, using local endpoints and log files, and then produces structured data in the form of normalised JSON objects, which contain:

- The `type` of data: metric or log
- The `parameter` observed: e.g. `cpuUsage`, `logLevel` or *description*
- The corresponding `value`.

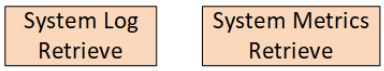

Figure 3.3: Architecture Focus - Log Retrieval

As described previously, the information collected by *Fluentd* are then periodically sent to the **TRA Agent Server** via HTTPS protocol, which collects and format them before sending to the **TRA Host Server**, as well as in parallel them are also sent to *Grafana* for purely informational purposes, allowing local system operators to check the system's performance over time and obtain a graphical representation of events.
Once the formatted data has reached the **TRA Host Server**, it is analysed using the **Lookup Table**, which associates well-known risk conditions with the observed parameters. For each input, the output generated has the following format:

- The observed value in a tuple: <`param, value`>
- The reference value: `refValue`
- A Boolean `flag` if a threshold for an alert has been activated.

When the boolean flag for an alert is set to true, an alert is generated and sent to *Alertmanager*. It receives the structured alerts, classifies them, and forwards them to the predefined recipients via the configuration file.

The alert is enriched with the risk score. At this stage, the **Scoring Module**, shown in Figure 3.4, comes into play, correlating each alert with the Common Vulnerability Scoring

System (CVSS) profile associated with the element involved. This process then produces the contextualised numerical risk index. This will then stored within an **Alert Lake** (collection of alerts), from which it is subsequently extracted or grouped for be sent to the **Report Engine**, which in charge of the production of reports in PDF format. As well the display in the **TRA Host Portal** of the generated alarm, together with other context metadata.
The **TRA Host Portal** is an additional web-app that is the central interface for the control and visualisation of detected events and risks. Unlike the **Agent Portal**, this obtains data from the **TRA Host Server** and not directly from the monitored system, and includes: visualisation of active alerts and CVSS risk score.

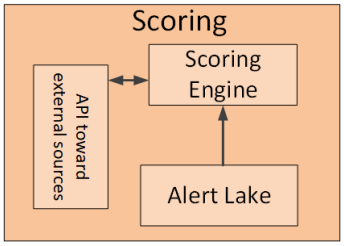

Figure 3.4: Architecture Focus - Scoring Module

3.2.4 Lookup Table

Within the Threat Risk Assessor, lookup tables (shown in Figure 3.5) play a key role, as they take care of the decision-making aspect concerning alert generation. In general, it is a simple but very effective tool that allows decision-making logic to be implemented quickly and easily.
In computer science, a lookup table is a data structure that associates each input with a predefined output [58]; considering this definition in a more abstract way, it is a discrete function that maps an input → response pair, allowing calculations to be avoided at runtime, replacing them with access to a predefined table. This type of approach is therefore typically used to optimise performance in environments where it is essential to have a high response speed and flexibility in configuration, such as TRA.

In this context, the lookup table is used to associate a given input parameter with: a threshold if it is a metric, which if exceeded triggers an alert; or an expected response, such as the log level of a log message. This then enables it to:

- Make the logic modifiable without modifying the code, in fact it is sufficient to update the table to introduce new risk conditions, or readjust the parameters in the event of false positives
- Keep the evaluation process transparent and auditable, since each TRA decision represents a hypothesis encoded in a threshold or parameter, this allows each decision taken by the system to be tracked.

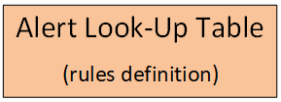

Figure 3.5: Architecture Focus - Lookup Table

3.2.5 Scoring Engine

The risk assessment process has taken shape with this release. This is based on the adoption of the CVSS as the central scoring system, which aims to provide a structured, objective and repeatable method for classifying vulnerabilities and risk situations in information systems. It is an internationally recognised standard, developed by the Forum of Incident Response and Security Teams (FIRST).
The main advantage of using the CVSS lies in the possibility of evaluating a set of risk metrics, related to both the impact and the ease of exploitation of the vulnerability, in a single score from 0 to 10.

Each component of the monitored system is enriched through a string of CVSS metadata, which represents an a priori assessment of the criticality of the element, which is then stored as part of the component's risk profile in the system. This metadata includes basic metrics such as:

- Attack Vector (AV): which access mode is used to carry out the attack (e.g. network, physical)
- Attack Complexity (AC): technical complexity required
- Privileges Required (PR): minimum privileges required to carry out the attack
- User interaction: need for user involvement
- Confidentiality, integrity, availability impact: potential impact on the three pillars of security.

When a risk event is detected by the lookup table, it is passed to the **Scoring Engine**, which enriches the information with a risk score; this data set is then sent to *Alertmanager*, which creates and manages the related alert. This allows for a dynamic notification system, each depending on the severity of the alert generated.
The final result, thus enriched with this risk number index, will then be passed to the system part that will be responsible for indicating which possible countermeasures to take; in this version, this component has not yet been developed.

CVSS Risk Profile - Example

For the purpose of providing a clear understanding, a CVSS risk profile that has been assigned to a component of the 5GC system under consideration is now presented:

```
N,H,H,N,U,N,L,H
```

Listing 3.1: CVSS risk profile example

Analysing the value of each of these attributes, the profile is the following:

- **N**: this describes the AV, so how the vulnerability can be exploited. In this case, it means that the attack can be launched over a *Network*
- **H**: this indicates the AC, indicating how difficult it is to exploit a vulnerability on such component. In the presented example it is set to *High*, due to factors like the need of specific conditions, timing and requirements
- **H**: according with the previous argumentation, this is set to *High*, since it indicates the PR to conduct the attack
- **N**: this indicates the user interaction, that in this case is not needed, since this risk profile is related to a NF
- **U**: the value describes whether the vulnerability exploitation has a cascading effect, altering the security properties of the system. In this case, it is set to *Unchanged*, since a possible exploitation does not lead to spread across different parts of the system
- **N**: this measures the impact on the confidentiality, that in this case is *Null*
- **L**: it indicates the impact on the integrity of the system, in this case it is *Low*, since possible modification over the data elaborated by the NF are unlikely to be significant or harmful in most cases
- **H**: it describes the impact on the system availability, in this case it is set to *High*, as an exploitation to this component can lead to a DoS situation.

3.2.6 Report Engine

The second version of TRA includes one of the key components for the whole process of analysing and processing of the KPIs relative to the monitored systems, the **Report Engine**.
This element has a dual function, which fragments the notifications sent into two different time instants. First, a notification is sent via a Slack channel, indicating the presence of a potential risk situation. Secondly, following system processing, a report is promptly generated containing all the key information relating to the detected risk. This is supplemented with a graph of the KPIs involved, so that the user can immediately see the trend of the event. In addition, some possible countermeasures are suggested to try to mitigate the detected risk.

Figure 3.6 shows an example of a portion of a generated report containing general information associated with the alert, i.e. the timestamp, severity, KPIs, summary with all key-risk data and some possible mitigations.

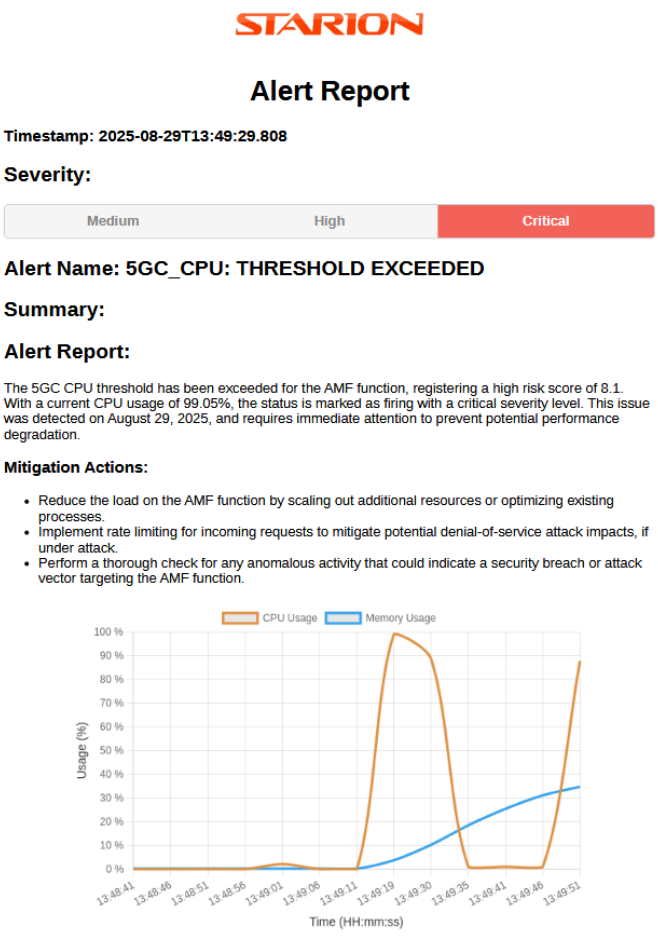

Figure 3.6: Alert Report Summary

3.2.7 Attacker Portal

The second version of TRA introduces a platform dedicated to the controlled simulation of cyber attacks, called **Attacker Portal**, shown in Figure 3.7. This is designed to offer operators a secure and flexible environment to perform targeted tests, simulating real threat situations. This is a web-app built ad-hoc, which allows an operator to:

- Load a custom *Python* script representing the attack payload, allowing complex and/or specific scenarios to be modelled
- Define the target of the attack, specifying IP address, port and protocol
- Monitor the impact of the attack in real time through the portal, where KPIs are dynamically updated. Different sets of pre-defined KPIs can be selected based on the most popular scenarios (e.g. DDoS, replay, traffic manipulation).

This functionality transforms the TRA from a simple monitoring system to an active testing platform, allowing for integrated penetration testing capabilities. This approach makes it possible to switch to an early risk management approach, identifying potential vulnerabilities and critical points throughout the system's life cycle, following the principle of security by design[2].

Figure 3.7: Architecture Focus - Attacker Portal

[2]Security by design: it is a software engineering principle according to which security aspects must be incorporated at an early stage in the design of a system, rather than being added later as an accessory component.

3.2.8 Tools

Compared to the previous version, the number of third-party tools used has decreased in favour of ad-hoc solutions, with the aim of increasing the abstraction of the TRA with respect to the monitored system.

Node.js & dockerode

Node.js was the key innovative element of this second version of TRA, allowing both monitoring and system analysis to be orchestrated. It is a runtime platform built on Chrome's V8 engine, designed to allow *JavaScript* code to be executed on the server side. It adopts an asynchronous, event-based execution model, which makes it suitable for scenarios where it is necessary to manage parallel operations [22] with low computational overhead, as is the case with TRA.

Fluentd

Fluentd is a log aggregation technology that is particularly useful in complex environments where log and metrics data are continuously generated from different sources, making their management and monitoring complex without the presence of a centralised and well-structured system. This allows data from multiple sources to be aggregated, transformed into a standardised format in real time, and forwarded to the processing server. It thus presents itself as a versatile and effective solution for the TRA context, allowing complete control of system data.

3.3 TRA Version 3 - Final Version

The third version of Threat Risk Assessor represents the final step in the completion of this thesis project, representing a significant evolution in the direction of a more scalable and robust system that is suitable for use in real systems. This version introduces advanced elements in terms of both attack and detection, with the aim of consolidating the entire architecture so that it can be considered ready for application in real environments.

This update therefore aims primarily to improve the system pipeline in terms of reliability and traceability. In fact, this version introduces two fundamental innovations in this direction:

- The adoption of *Apache Kafka* [6] for the asynchronous and resilient transport of data between **Agent** and **Host**
- The introduction of a specialised storage system, based on *MinIO* [34] for raw data and *Elasticsearch* [16] for structured post-processing data and alerts.

These lay the foundations for further consolidation of the system, which in this version includes significant advances in attack simulation capabilities and detection accuracy.
This version also brings in the possibility of simulating real threat scenarios and, at the same time, improves the ability to distinguish anomalous events from simple variations or exceptions, also adapting the analyses to the evolution of the system.
To expand the tool's capabilities in these areas, the following has been done: the integration of *MITRE Caldera* as a Command and Control (C2) server, which significantly improves the capabilities provided by the **Attacker Portal**, and the addition of an anomaly detection module, based on a machine learning technique, to improve the accuracy of the system's detections.

Following these changes, the updated architecture can be observed in Figure 3.8.

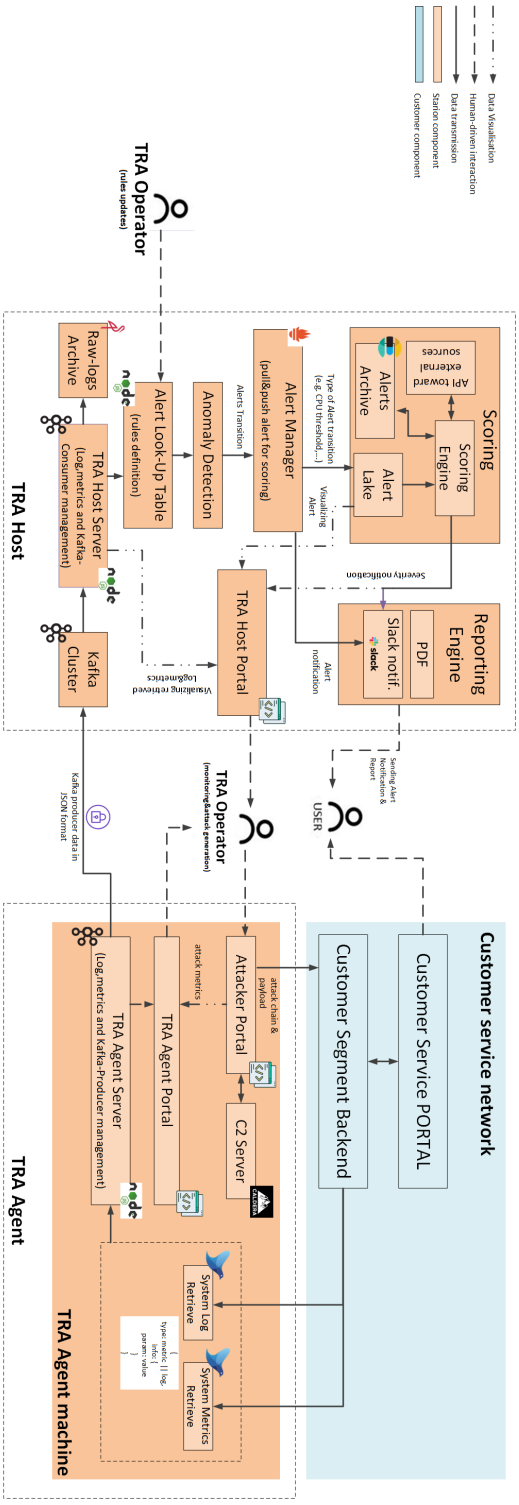

Figure 3.8: TRA Version 3 - Architecture

3.3.1 Kafka pipeline

In previous versions, data transmission between the **TRA Agent Server** and the **TRA Host Server** took place via calls to the respective APIs. This approach proved to be suitable for a laboratory environment, but unsuitable for distributed or latency-prone scenarios.
For this reason, *Apache Kafka* was introduced, a platform designed to handle large volumes of data in a reliable, scalable and asynchronous manner.
The integration of *Kafka* has led to a number of important benefits:

- Temporal independence: the producer and consumer no longer have to be active simultaneously, thanks to the presence of the Kafka-cluster the data is retained within it until the consumer picks it up to process it
- Resilience to data loss: again thanks to the cluster, the sending of each message is guaranteed and can be reprocessed in the event of an error
- Horizontal scalability: in the event of a major increase in traffic and monitored systems, new consumer modules can be integrated for parallel analysis, without modifying existing components.

From a practical point of view, the data sent by the *Fluentd* module are forwarded to the cluster by a producer implemented in *Node.js* within the TRA Agent Server. These are then consumed by the **TRA Host Server**, via a *Kafka* consumer developed in *Node.js*.
The *Kafka* cluster was realised via a container based on the Bitnami Kafka image [12], chosen for its reliability and ease of configuration.

3.3.2 Archiving system

The other significant introduction of this version was a hybrid archiving system, shown in Figure 3.9, aimed at satisfying two distinct but at the same time complementary needs: the storage of raw data for forensic and analysis purposes, and the efficient consultation of processed alerts.

MinIO was chosen for its lightness and was therefore used to archive logs and metrics in raw format, received directly from the Kafka-consumer, prior to their processing.
This archive therefore provides a verifiable forensic basis in the event of a security incident and keeps audit logs separate from the actual risk events, enabling retrospective reconstruction[3] of events in the case of malfunctions or errors in the pipeline.

At the same time, the processed and enriched data, in particular those containing alert and scoring information, are sent to *Elasticsearch*, a full-text indexing system that allows events to be quickly retrieved from specific fields (e.g. the status of an alert), as well as simple integration with visualisation tools such as *Kibana* [15].

Figure 3.9: Architecture Focus - Archiving

[3]Retrospective Reconstruction: it is a forensic technique used in the IT field that consists of the post-incident analysis of a sequence of events, in order to identify the origin, propagation and impact of a security incident or malfunction

3.3.3 Attacker Portal evolution

In this version of TRA, the functionality and capabilities of the **Attacker Portal** have been significantly enhanced thanks to the integration of *Caldera* as the backend of the **Attacker Portal**, which remains completely transparent to the user, who continues to interact with the ad hoc interface. The operator can now leverage complete predefined attack chains, as well as create new ones, to create complete attack scenarios relevant to the system to be targeted.
Thanks to this integration, it has been possible to automate complex attacks that required a series of interdependent operations: in previous versions, any lateral movement and privilege escalation actions were the responsibility of the operator, who had to define ad hoc scripts each time. Now, these operations, like many others (e.g., data exfiltration), are automatically included in the selected attack scenarios. It is also possible to monitor the status of each phase through the **Attacker Portal**.
Thanks to the community behind *Caldera* and its open-source nature, the available chains are based on real TTPs, in line with the MITRE ATT&CK® framework, thus increasing the realism and relevance of the simulation. This not only allows the system's resilience to more complex attack sequences to be tested, but also maximises the response capabilities of the detection, scoring and reporting system.
This evolved structure is shown in Figure 3.10.

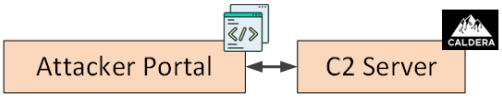

Figure 3.10: Architecture Focus - Attacker Portal with C2

From a practical point of view, this has resulted in the creation of a new feature called `Attack operation` within the **Attacker Portal**, which does not replace the previous attack deployment feature, but increases its possibilities. It is now possible to connect targets to the C2 via a script, and then select the attack scenario to recreate, which includes all the TTPs necessary to execute a complete and realistic attack. At this point, simply start the simulation using the appropriate button and observe its status through the interface that appears below, which will indicate the operation being performed step by step, with also a reference to the identification code within MITRE ATT&CK®.

3.3.4 Anomaly Detection

On the other side of the architecture, in the **TRA Host**, the **Anomaly Detection** (shown in Figure 3.11) component has been added, positioned between the **Lookup Table** and the **Alert Manager**, with the aim of detecting which events are truly anomalous and which are exceptions, thus reducing the generation of insignificant alerts and increasing the reliability of the system. This proved to be crucial as a significant number of false positives were detected during the simulations.
To contain this type of event, it was decided to adopt a classification module that, each time a threshold configured in the Lookup Table is exceeded, analyses the consistency of the data with the historical and expected behaviour of the system. If the data is not consistent with the system history, an alert is generated; otherwise, it is ignored. Specifically, the model adopted is Isolation Forest [57].

Figure 3.11: Architecture Focus - Anomaly Detection

During the experimental phase, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [53] was also evaluated, which is often used for detecting irregular clusters. However, it showed some limitations for the specific context of TRA:

- Sensitivity to configuration parameters ϵ (eps): the optimal choice of these is complex and not adaptive, especially in dynamic environments. This represents the search radius around a point to determine whether there are other points belonging to the same density, which, if configured inappropriately, could result in high noise[4]. Therefore, a value that works well in one situation may not be valid in another, and this would lead to the need for many different models, which is not an optimal solution for this application
- Difficulty in handling data with variable density: DBSCAN tends to merge clusters with similar densities, confusing local anomalies with normal behaviour. Certain components may have standard patterns that would be exceptions for others, and the intersection of these cannot exist correctly in a single model.

On the other hand, *Isolation Forest* showed better generalisation and adaptability. This is an unsupervised machine learning algorithm, designed specifically for detecting anomalies in multidimensional datasets. The logic behind this is based on the assumption that anomalies are rare and easily separable data, requiring few operations to be isolated

[4]Noise: in this context, noise refers to points in the dataset that do not belong to any significant cluster and are therefore considered anomalies, even though they are not. These can therefore lead to errors in assessment during the decision-making process.

from regular points. To perform this separation, the algorithm constructs a forest of binary trees, generated by randomly dividing the data along the features [41]. The average number of partitions required to isolate a point becomes an indirect measure of its anomaly: the lower this depth, the greater the probability that the data is an outlier.

From a practical point of view, for the application of TRA, the model is trained on a dataset that includes the set of KPIs, together with categorical data, transformed using One-Hot Encoding[5]. The trained model is then loaded on disk and saved together with the encoder used, so that it can be used to make predictions in the event of anomalies.

Each time a threshold in the **Lookup Table** is exceeded, the data necessary for analysis is sent to the model, which will return a binary anomaly indicator: -1 in case of anomaly, $+1$ in case of regularity.
To ensure that the model adapts to the evolution of the system over time, it is periodically retrained with new data collected since the last training session. This further improves the robustness of the system with respect to possible false positives due to changes in the monitored systems.

The results of the training of this model and the related tests are presented in the section 5.3 of the chapter dedicated to results.

[5]One-Hot Encoding: it is an encoding technique used to transform categorical variables into a numerical form so that they can be used in machine learning models [19]. In this case, they represent the system and the component under analysis. Each category is then represented by a binary vector in which a value of 1 indicates the presence of that category, while all others are set to 0. In this way, it is possible to associate numerical values with specific elements represented by a string value.

3.3.5 Tools

The evolution of the TRA architecture involved the integration of three new fundamental tools to improve the reliability, scalability and observability of the entire system. Each of these tools was selected to fulfil specific tasks within the TRA pipeline and will now be presented individually.

Apache Kafka

Apache Kafka is an open-source platform initially developed by *LinkedIn* and subsequently donated to the *Apache Software Foundation* [51]. It is designed for real-time management of data streams, ensuring resilience, scalability and persistence.
It is based on a distributed publish-subscribe architecture, in which producers send data to partitioned topics, while consumers subscribe to these topics to receive messages. Inside these functionalities, a characteristic aspect is native persistence: messages are written to disk and can be stored for a defined time (or even indefinitely), allowing the data to be reprocessed in the event of pipeline errors.
Kafka provides various APIs to interact with its features and these are divided into four main categories:

- Producer API: allow messages (events) to be sent to Kafka-topics from client applications
- Consumer API: allow applications to read messages from one or more Kafka-topics
- Streams API: enable real-time processing of data streams directly within *Kafka*
- Connector API: provide predefined connectors for integration between *Kafka* and external systems such as databases.

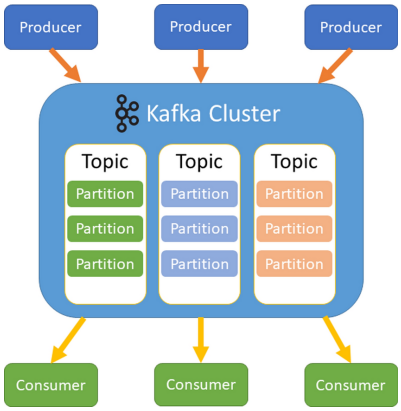

Figure 3.12: Apache Kafka Architecture (from [10])

MinIO

MinIO is an object-torage platform designed for use in cloud-native, containerised and distributed infrastructure environments. It enables the storage, replication and retrieval of structured and unstructured files in the form of objects. These are identified by: bucket, key and related metadata.
In the TRA, it is used as an archive of logs and metadata in raw format, acting as a forensic persistence system of the original data before any processing or transformation, allowing complete reconstruction of events. In addition, it includes access management and object versioning functionalities, fundamental functionalities in a dynamic architecture such as the TRA architecture that interfaces with heterogeneous systems.

Elasticsearch

Elasticsearch is a search and analysis engine based on the open-source Apache Lucene framework. it is designed for indexing and searching large amounts of data, supporting complex queries based on: full-text, aggregations, Boolean filters and statistical aggregations.
Documents are organised in indexes, distributed over one or more shards, which can be replicated to ensure fault tolerance. These documents are formatted in JSON format, which allows them to be indexed internally as a structure composed of analysable fields. Within the TRA it is used as a storage and query system for alerts and processed data, offering fast response times in correlating events and visualising risk patterns.

Caldera

MITRE Caldera is an open-source platform designed to automate cyber attack simulations and test IT system defence measures. It allows to recreate complete attack chains based on TTPs used by real-world adversaries, enabling security teams to effectively assess the resilience of their infrastructure against real threats. *Caldera* includes a series of predefined scenarios that comprise a specific sequence of TTPs aimed at automating red teaming operations.
In the context of TRA, this solution is used to leverage C2 server functionality for attack simulation, through the implementation of the respective APIs [7] to expand the capabilities and effectiveness of the Attacker Portal.

Chapter 4

Real-Case Scenarios

Now that the architecture and functionality of the Threat Risk Assessor have been described, this chapter presents two real-world use cases in which the platform has been used to monitor systems in critical environments. The underlying objective of these applications is to demonstrate the effectiveness, versatility and adaptability of TRA in realistic and extremely diverse scenarios.

The first scenario concerns the integration of TRA within a Core Network 5G Advanced, based on *Free5GC* in which it has taken on a central role in continuous risk assessment, controlled attack simulation and constant system monitoring.
The second scenario focuses on Earth Observation (EO) using satellite technology, with particular attention to the detection and monitoring of natural disasters such as floods. In this scenario, TRA has been adopted to monitor the image processing system in order to validate the processing chain, guarantee data integrity and prevent tampering, thus ensuring that the information transmitted is reliable and verified.

The following sections will present the scenarios that has been considered and created for the validation of the TRA, describing the reasons behind their realization and providing all the information necessary for a complete understanding.

4.1 5G Threat Landscape

With the aim of validating the reliability and effectiveness of the risk detection and assessment mechanisms, it was decided to conduct a reverse engineering process applied to the *Free5GC* to uncover vulnerabilities to be exploited to trigger alarms in the TRA.
Before delving into this process, a preliminary analysis on this architecture is presented. It started with the analysis of the report entitled "ENISA threat landscape for 5G networks", prepared by the European Network and Information Security Agency (ENISA) [33], that contains the challenges, vulnerabilities and responsibilities associated with the security of 5G networks. This is an extremely important aspect, since this technology represents a significant technological evolution, which consequently brings a significant expansion of the attack surface.

4.1.1 Virtualisation and risk

The 5GC, as anticipated, compared to previous generations is based on an SBA, in which the NFs interact each other, guaranteeing great flexibility and scalability, but at the same time introducing many attack surfaces related to virtualisation, orchestration and API management.
The vulnerabilities present, as demonstrated in a previous thesis [20] work carried out, are often strongly linked to the implementation and configuration of the functionalities:

- Functions such as AMF and UPF are often vulnerable and with incomplete implementations, with no strong control over authentication, status management and message validation
- The use of protocols such as HTTP/2 increases exposure to known vulnerabilities such as the injection of malicious commands
- The orchestration of components via Virtual NFs (VNFs) can introduce configuration and versioning errors that create time-windows in which vulnerabilities can be exploited
- Weak or absent encryption, communication between elements does not always adopt authenticated and encrypted channels.

Another key aspect of CN is network slicing, which brings with it several inherent risks, such as:

- Insecure instantiation: with frequent use of weak passwords and unauthorised access while using NFs
- Insufficient isolation: slices share physical resources and if there is no clear separation, there is a risk that an attacker will compromise one slice and propagate to the others.

4.1.2 Security testing and evaluation in open-source 5GCs

To identify which indicators of compromise could be useful to include in the TRA analysis to validate the technology, two study and test phases has been conducted: the first, based on papers and articles found online, while the second through a process of reverse engineering the *Free5GC* architecture.
The tests were conducted in virtualised environments based on *Docker*, reproducing CNs in stand-alone mode and implementing virtualised simulation of radio access network (RAN) and user equipment (UE) through UERANSIM [50]. The testbeds therefore included the functions: AMF, SMF, UDM, NRF, NEF and SBI interfaces exposed on isolated *Docker* virtual networks within the test environment.

During the first phase, analysis activities were based on papers found online, to identify and understand the vulnerabilities at the protocol and API management level. The results that were considered significant for these studies are presented next, and these will lay the foundation for the preparation of this first validation use-case.

Replay Attack

One of the most relevant attack vectors for 5GC is the replay of NAS messages, especially during the security mode command procedure [11], which takes place before encryption. To carry out this type of attack, targeting the AMF, the *5Greplay* tool is used to modify and resend packets from a legitimate session. The final objective is to check whether or not the AMF is able to distinguish duplicate and/or delayed messages, thus avoiding the registration of fake terminals. The results presented in the papers [46] [20]show that:

- *Open5GS*: accepts modified messages, generating duplicate registrations for the same UE/gNB, affecting the stability and performance of the system, eventually even causing the state to be reset
- *Free5GC*: which is the reference example case for the subsequent test phase, ignores modified messages after the first attempt, but since it does not return warning logs relating to this situation, the detection is impossible.

DoS and DDoS attacks

Providing services to the public, a critical aspect for 5G networks is the ability to resilience to abnormal loads that could be derived from DoS, or Distributed DoS (DDoS) attacks, especially towards the most exposed control functions, such as: AMF, SMF and NRF.
The techniques used in the case studied made use of tools that simulated both volumetric (MHDDoS) and resource exhaustion (HULK) attacks [21]. Furthermore, for attacks against the Open5GS-based CN, the *5Greplay* tool was reused, injecting malformed packets [46].

The results obtained were as follows:

- *Open5GS*: AMF proved to be the most vulnerable component, since the inability to handle malformed NGAP packets was also demonstrated during the tests, leading to an overflow of the decoding queue, which consequently led to a crash of the component
- *Free5GC*: proved to be resilient in handling malformed packets, however, it nevertheless suffers significant slowdowns in the routing of requests towards the other components, leading to a significant increase in waiting times for replies. A behaviour that is considered unacceptable and unsafe for critical scenarios.

API injection and parameter manipulation

In the context of SBA architectures, network functions communicate via RESTful APIs, which are thus exposed to the typical vulnerabilities of web applications, such as injection and parameter manipulation.
For this scenario, malformed requests were made to the NFs' REST endpoints, with the aim of carrying out SQL and NoSQL injection attacks, as well as DoS through NULL parameters or malformed JSON requests. The results [21] presented are as follows:

- *Open5GS*: several endpoints fail to validate requests, leading to the crash of the service
- *Free5GC*: is resilient, as it validates requests before passing them to functions.

4.2 Free5GC reverse engineering and exploit development

Once the phase of analysing the results of the previous studies was completed, and considering the need to validate not only the TRA, but also the functionality of the Attacker Portal, it was decided to create two scenarios useful for validating both aspects of the developed technology. The TRA analyses metrics and logs, while the Attacker Portal allows the creation of customised payloads, or complete attack-chains, and the execution of attacks.
Starting from this objective, an initial test phase was carried out on what could be the most potentially interesting targets for an attacker, and considering the functionality of each of these elements, the choice fell on two key elements of the *Free5GC* architecture, namely the AMF and the WebUI. The first element has already been presented in the previous subsection 2.1.3, while the second deals with the management of users registered to the CN, with particular reference to IMSI[1] profiles, authentication keys and network parameters; thus playing a critical role in the configuration and provisioning of users, and thus representing an important target for malicious actors.
In order to find and develop appropriate exploits, the work performed will be based on the reverse engineering of the two elements of interest, first statically and then dynamically. These will be executed only within an isolated and virtualised environment, set up to simulate the compromise scenarios required for validation.

4.2.1 Analysis and exploit development for WebUI

Considering the centrality of the WebUI, it represents a highly critical point of attack. The analysis of this component was conducted in two parallel phases: on one side, a static study of the source code was carried out, with particular attention to session management, RESTful API calls and interaction with the database (*MongoDB*). While on the other side, a dynamic phase was undertaken, through the analysis, and manipulation, of the generated requests and outputs, to observe the behaviour of the WebUI in the presence of malicious input.
Since access to this interface is restricted to accounts with administrative permissions only, the main focus is precisely on the authentication mechanism. Following the code analysis, it emerged that administrative access is granted through the generation of a JSON Web Token (JWT), the release of which takes place following a correct login. The process is then first analysed in a normal scenario, through a login with an authorised user, the token is then released and analysed.
The JWT received was malformed, with an invalid signature with respect to the configuration parameters entered in the token's meta-data; this check was carried out through

[1] International Mobile Subscriber Identity (IMSI) [56]: it is a unique numeric identifier assigned to each user within a cellular network and serves to identify each subscriber within the CN system. It therefore allows authentication during access to the network, as well as associating user data in the database (e.g. authentication keys) with the device requesting them.

the official JWT standard platform [28], which allows verification of the token's format validity, the result is shown in the figure 4.1.
The signature is truncated during the creation process, and despite this, the WebUI accepts the token as valid and allows access to the reserved API, without verifying the digital signature.

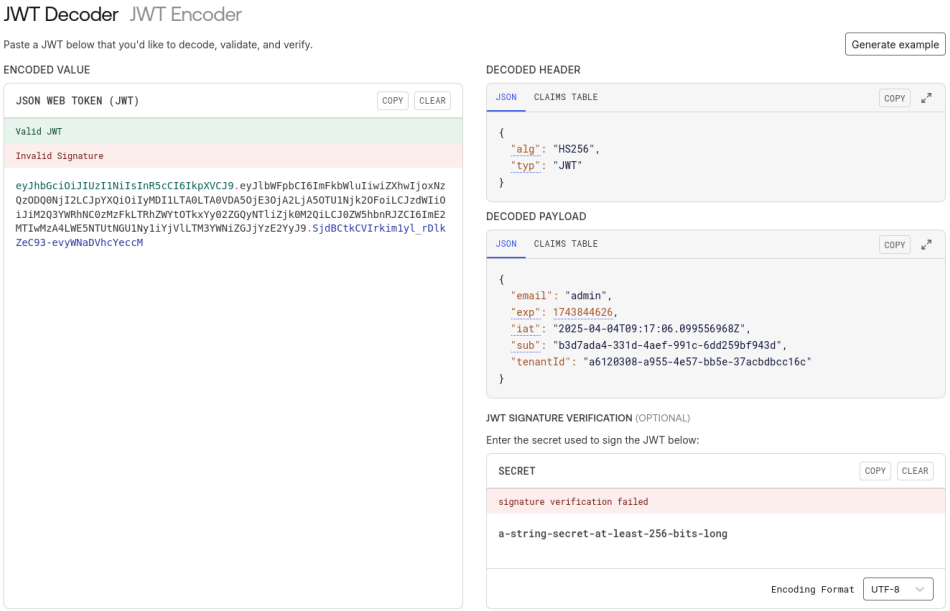

Figure 4.1: Result of JWT format validity check

To confirm and validate this vulnerability, a crafted token was created, i.e. with a correct header and payload, but with an arbitrary signature, which should therefore be recognised as invalid. However, this was recognised as valid and this token allowed privileged operations such as adding and deleting users, without the need to perform any real authentication.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFkbWluIiwiZXhwIjoxNz
QzODQ3MzcwLCJpYXQiOiIyMDI1LTA0LTA0VDEwOjAyOjUwLjM1ODk5NzQ4NFoiLCJzdWIiO
iJiM2Q3YWRhNC0zMzFkLTRhZWYtOTkxYy02ZGQyNTliZjk0M2QiLCJ0ZW5hbnRJZCI6ImE2
MTIwMzA4LWE5NTUtNGU1Ny1iYjVlLTM3YWNiZGJjYzE2YyJ9.F70z5nDL_USmKXJngPgB4C
pGTXBaxPvFxHkeTDStwmo
```

Figure 4.2: Original JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFkbWluIiwiZXhwIjoxNz
QzODQ3MzcwLCJpYXQiOiIyMDI1LTA0LTA0VDEwOjAyOjUwLjM1ODk5NzQ4NFoiLCJzdWIiO
iJiM2Q3YWRhNC0zMzFkLTRhZWYtOTkxYy02ZGQyNTliZjk0M2QiLCJ0ZW5hbnRJZCI6ImE2
MTIwMzA4LWE5NTUtNGU1Ny1iYjVlLTM3YWNiZGJjYzE2YyJ9.F70z5nDL_USmKXJngPgB4C
pGTXBaxPvFxHkeTDStwmp

Figure 4.3: Crafted JWT

The results obtained and shown in the figures 4.2 & 4.3, allowed the development of a script that generates crafted tokens until a valid one is obtained. Once the JWT considered valid has been obtained, the script allows false IMSIs to be registered, to be deleted, and the list of all users and their information to be obtained, completely bypassing the system's verification process. This exploit will be fundamental in validating the TRA log analysis system, the result of which will be presented in the next chapter of this thesis.

4.2.2 DDoS attack scenario development

The other scenario that was built to validate the TRA analysis metrics functionality was a DDoS attack. During the analysis of the CN components, the absence of load balancing mechanisms was noted. This suggested the opportunity to design an attack scenario simulating abnormal load conditions.
For this scenario, a virtual botnet was developed, consisting of several *Docker* instances, dynamically started and managed through a docker-compose file, which allows its behaviour to be managed.
As mentioned above, the main objective of this scenario is AMF, so as to realise a scenario in which it becomes impossible for UEs to authenticate themselves in the network and obtain access to mobile services. Also for this scenario, the results of its execution will be shown in the next chapter, which will validate the effectiveness of the TRA detection technology.

Botnet Simulation

In order to recreate a DDoS scenario, a botnet was created based on the replication of *Docker* images, orchestrated and managed through a script and a docker-compose file.
The basic *Docker* image is based on `python:3.9-slim` [14], to occupy little disk space and have a good number of packages installed, thus creating a minimal Python environment that can be used across multiple replicas.
This choice was made because it allows multiple instances of *Slowloris* [66] clients to be instantiated. This software is used to simulate DoS attacks, since it allows you to saturate the resources of a web server without the need to generate large amounts of traffic, by opening a large number of HTTP connections and sending incomplete or fragmented requests slowly, as the name suggests. In this way, it prevents the target from processing legitimate incoming requests, rendering the services offered by the server unusable.

Once this basic image had been created, we moved on to developing the docker-compose orchestration file, in which the number of replicas to be instantiated is set and an array of arguments is created using the `command` directive. These will then be passed to the *Slowloris* images; in particular for this simulation the following setup has been used:

- $A.B.C.D$: the target IP address
- `-p` X: the port of the target service for the attack
- `-s` Y: the number of simultaneous connections to instantiate
- `-sleeptime` Z: the waiting time between connection creation attempts
- `-ua`: activation of random user-agent[2] selection mode.

4.3 Satellite image processing pipeline attack-scenario

In the context of the second application scenario, focused on the analysis of satellite images for monitoring extreme natural events, it was deemed necessary to validate the resilience of the image processing pipeline with respect to possible compromises or exfiltrations. In particular, the integrity and confidentiality of temporary data generated during the intermediate stages of processing are critical for two reasons: a malicious actor could compromise the integrity of the processed data, leading to inconsistent results; or they could exfiltrate sensitive information, compromising the overall reliability of the system, as well as causing potential economic damage if this information were reused and resold on the market.

For this reason, the advanced features offered by the latest version of the **Attacker Portal** were exploited, in particular those relating to the `Attack-Operation` functionality, which allows complete attack chains to be distributed.
After a technical analysis conducted with the team of engineers responsible for developing the platform, it was decided to simulate a data exfiltration attack, with the aim of extracting temporary files produced during image processing.

The simulation was conducted by building a custom payload to establish a connection with the **TRA's C2 Server**, in order to simulate a compromise as realistically as possible. The payload was injected into the target environment and activated, connecting it to the server. Once the connected host was viewed through the **Attacker Portal**, the attack chain was activated.

[2] user-agent [64]: it is a string sent by HTTP clients. It identifies the browser, operating system and/or application making the request to the server.

This was modelled following the TTPs used by attackers in the data exfiltration scenarios presented in the MITRE ATT&CK® framework, replicated using *Caldera* and then implemented. Specifically, the following TTPs were used:

1. Collection – T1074.001 - Create stagin directory [37]: a dedicated directory is created to organise the files that will be extracted in subsequent stages. This will facilitate transfer to the C2 channels
2. Collection - T1005 - Find files [36]: the file system is scanned to identify temporary files and directories containing sensitive data generated during the processing process. This process is guided by the definition of `facts` [8], i.e. values such as file name extensions, which must be checked during this process
3. Collection - T1074.001 - Stage sensitive files [37]: once the target files have been identified, they are copied in the stagin directory previously created
4. Collection – T1074.001 - Compress staged directory [37]: the copied files are then compressed into .zip archives to facilitate transfer and reduce the transmission footprint that could trigger alarms in Intrusion Detection Systems (IDSs)
5. Exfiltration – T1041 - Exfiltration Over C2 Channel [38]: the compromised and obfuscated data was then transferred through the C2 channel to the server, thus completing the attack.

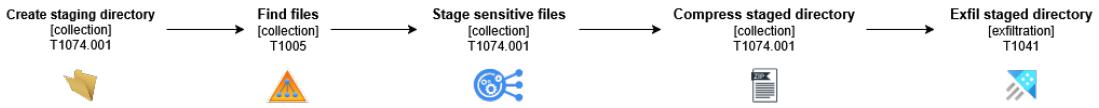

Figure 4.4: Attack Chain

This simulation had two purposes: on the one hand, to assess the effectiveness of TRA detection in a data exfiltration scenario, and on the other hand, to validate the effectiveness and versatility of the features offered by the **Attacker Portal**.
The next chapter will present the results of this simulation, in order to demonstrate the validity of the TRA platform even in contexts where the focus of security lies more in the pipeline than in the overall architecture.

Chapter 5

Results

This chapter is dedicated to the presentation and analysis of the results obtained during the TRA testing and production phases. The aim is to provide a clear and documented overview of the platform's overall performance and effectiveness in both controlled and real scenarios.

The first part will examine the data collected during testing, conducted in a controlled environment specifically designed to verify the behaviour of the TRA under measurable and reproducible conditions. These tests made it possible to evaluate each element individually, verifying that the thresholds and correlation logic were consistent with the expected results.
During this first phase, as in the subsequent ones, in addition to the outputs, structured logs played a key role, allowing each event to be recorded and analysed and its evolution within the processing process to be tracked, as well as documenting each of these events. This traceability made it possible to identify any critical points and anomalies, providing input for resolution and updating operations.
The test environment also allowed simulated attack campaigns to be carried out, in which the **Attacker Portal** and related modules were used to deploy attacks in order to validate the system's ability to detect malicious behaviour in a timely and accurate manner.

The second part of the chapter will focus on the results obtained in real operational contexts, deriving from the two application scenarios described in the previous chapter, respectively:

- Integration of TRA in an advanced 5GCN: the platform monitored the system in real time in order to identify behavioural deviations and potential threats related to anomalous flows and vulnerabilities
- Application of TRA to a satellite analysis system for the detection of natural disasters: the system was responsible for validating image processing procedures even in the face of targeted attacks, through the simulation of a temporary information exfiltration attack.

The importance of the results obtained lies above all in the empirical demonstration factor: this is not merely theoretical or laboratory validation, but tests conducted on actual operational systems, with the constraints and complexities of a production environment. This confirms the effectiveness of the TRA integrated approach, which combines continuous monitoring, anomaly analysis and active simulation within a single operating cycle.

5.1 5GC Test Environment: DDoS Scenario and False Positive Mitigation

The first set of tests focused on the *Free5GC* test environment, with the aim of validating the TRA's detection and reporting capabilities under operational stress conditions, testing its entire chain of action, including the use of the **Atatcker Portal**.
This process was divided into two distinct phases: the first conducted without the use of the Anomaly Detection module dedicated to reducing false positives; the second, in which the module was activated to evaluate the benefits of its use.
In order to obtain reproducible and measurable conditions, it was decided to create a DDoS scenario. This choice was also motivated by the critical nature of this type of attack in mobile networks, where service saturation can seriously compromise their proper functioning.

Shifting the focus to the practical aspect, the *Docker* image described in subsection 4.2.2 was set up specifically to generate high-intensity traffic packets to the designated target in order to implement the test.
This was designed to be easily replicable and is orchestrated through a docker-compose file, launched through the **Attacker Portal**. In this way, it was possible to dynamically set the number of instances launched, until a sufficient volume of traffic was reached to saturate the target system's resources and cause the disruption.

The component identified as the main target was the AMF, as it is a critical node in user connection management. Once the attack was launched, it had the expected outcome, as the AMF progressively increased resource usage until reaching CPU saturation and high RAM usage. Following this, a connection attempt was made by a UE, which was unsuccessful, confirming the complete success of the attack.

5.1.1 Test Result

Throughout the test, the TRA constantly monitored the situation, collecting and analysing performance data. Thanks to this and the logging system, it was possible to accurately track the impact of the attack at every stage of the process, tracking the thresholds exceeded as shown in the Figure 5.4, the logs generated and the consequences on the AMF. The analysis algorithm successfully detected the risk status, assigning a score consistent with the severity of the threat and informing the operator by generating a report sent via email, as well as a brief summary notification on Slack.
Based on this information, the operator was able to promptly implement the most appropriate defensive measures, blocking the attack and restoring the system to full operation. Once mitigation was complete and the system stabilised, the TRA detected the change in status and produced an additional summary report containing a summary of the attack on the system, providing a fundamental documentary basis for auditing and continuous improvement of the system.

It is now possible to analyse the most significant differences that emerged during the two phases of experimentation.

Result Analysis: Phase-1

The forced alteration of the AMF's performance had a propagating effect on the normal workflow of the system, also leading to anomaly alerts on other components not directly involved. This produced a series of false alarms that could have caused confusion in the operators' analysis, consequently slowing down response times. The distribution of performance measurements shown in Figure 5.1, associated with the number of alerts detected shown in Figure 5.2, clearly shows a high degree of heterogeneity and highlights how many of these are not actually relevant as they do not deviate too much from the median value consider as normal behaviour, representing only partial performance spikes.

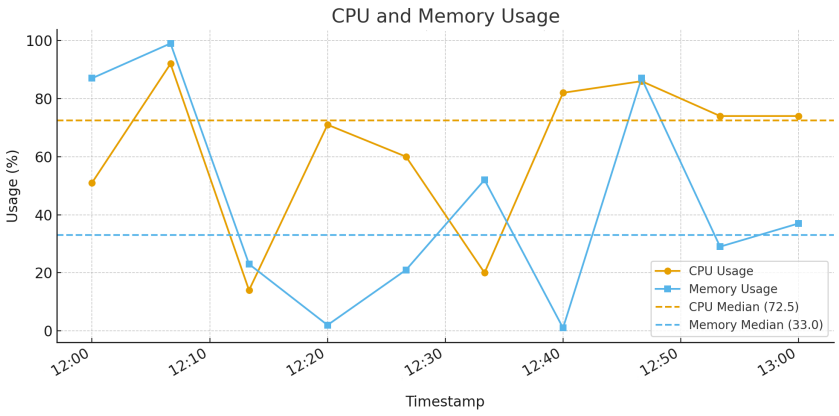

Figure 5.1: Performance graph

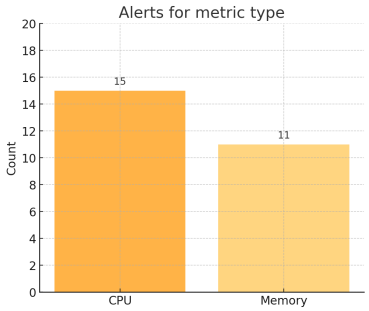

Figure 5.2: Detected alerts

Result Analysis: Phase-2

The introduction of the **Anomaly Detection** module has made it possible to filter out a large amount of the noise generated, significantly reducing false positives and consequently ensuring greater consistency between alerts and actual risk conditions. To reach this conclusion, data collected during testing without this module was used to obtain the results presented in the Figure 5.3. It can be seen that only a minority of spurious events are still reported, allowing operators to focus on truly significant indicators. This demonstrates how the adoption of this module has led to a significant improvement in the overall reliability of the TRA and the quality of the detection process.

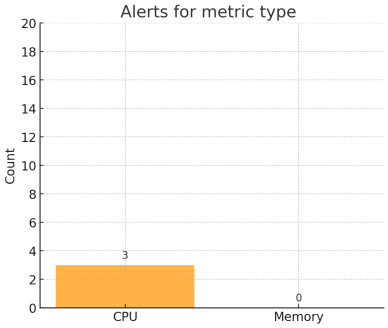

Figure 5.3: Detected alerts - Anomaly Detection

5.2 Operational Environment Validation: Production 5GC and Disaster Detection Systems

After the preliminary validation phase in a controlled environment, the tests were extended to real-world scenarios, referring to the implementations described in chapter 4: on the one hand, the integration of TRA within a real 5GC, and on the other, its application to a satellite image processing pipeline.
This phase was crucial as it allowed us to measure the effectiveness of the platform in contexts characterised by the complexity of real production systems, which are very different from laboratory conditions.
The results obtained are presented in the following subsections, each dedicated to a specific case.

5.2.1 5GC Production Environment: DDoS

The first set of results concerns the detection of a DDoS attack on the 5GC, presented in subsection 4.2.2. In this scenario, the TRA constantly monitored traffic flows and system resources, successfully detecting the risk situation.

In the Figure 5.4, it is possible to observe the saturation of the target (AMF), which was promptly identified, leading to the generation of an alert. In this way, the system initiated the process of notification and reporting of the event.
In line with what was observed in the laboratory tests, the TRA automatically generated two reports: one at the time of detection and one after the resolution phase, when the countermeasures brought the system back to stable operating conditions.

To avoid redundancy, these reports are not reproduced in full in this section as they are very similar to those already presented in the controlled tests. However, it is important to emphasise that this result represents tangible proof of the reliability of the TRA even in real-world contexts, confirming the system's ability not only to monitor technical parameters but also to contextualise information, leading to the production of clear and actionable alerts for the operator.

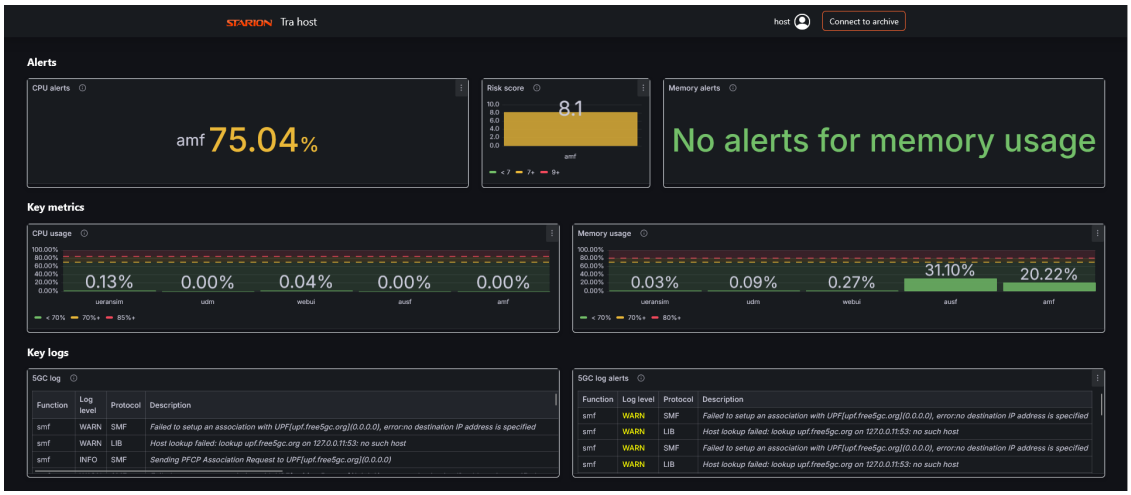

Figure 5.4: DDoS Attack Detection - 5GC Production Environment

5.2.2 5GC Production Environment: Custom Payload

The aim of this test was to verify the TRA's ability to manage compromise scenarios through the use of customised payloads, exploiting and validating the functionality of the **Attacker Portal**. In this case, the payload used is the one presented in section 4.2.1, in order to test and validate the log analysis functionality and its ability to identify IoCs.

Following the execution of the attack, Figure 5.5 shows how the log filtering and analysis interface allows messages considered as alarm triggers to be immediately highlighted. In particular, the receipt of a sequence of logs relating to the massive deletion of IMSIs from the network was recognised as anomalous behaviour. This correlation triggered the alerting mechanism, which promptly sent a notification to Slack, ensuring that operators immediately received a warning of possible system tampering.

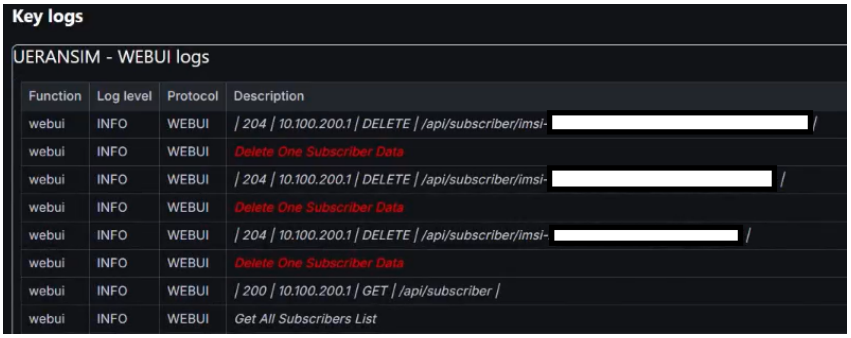

Figure 5.5: Custom Payload deploy - 5GC Production Environment

5.2.3 Disaster Detection Systems: Data Exfiltration

The third demonstration represented the most advanced stage of the experiment, with the aim of validating the integration of the `Attack Operation` features of the **Attacker Portal**. In this scenario, an entire attack chain was executed, leading to the creation of a realistic simulation of data exfiltration from the target system.
The attack was orchestrated in several stages, starting with the preparation and sending of the payload, through to the collection, compression and transmission of data via the C2 server, as presented in section 4.3.

As can be seen in the Figure 5.6, each step in the attack chain was executed, upon completion when the target system data was exfiltrated and the TRA generated a specific alert about the attack. The alert, the related risk score and contextual details were notified through appropriate external communication channels, as in this case the TRA was part of a more complex system.

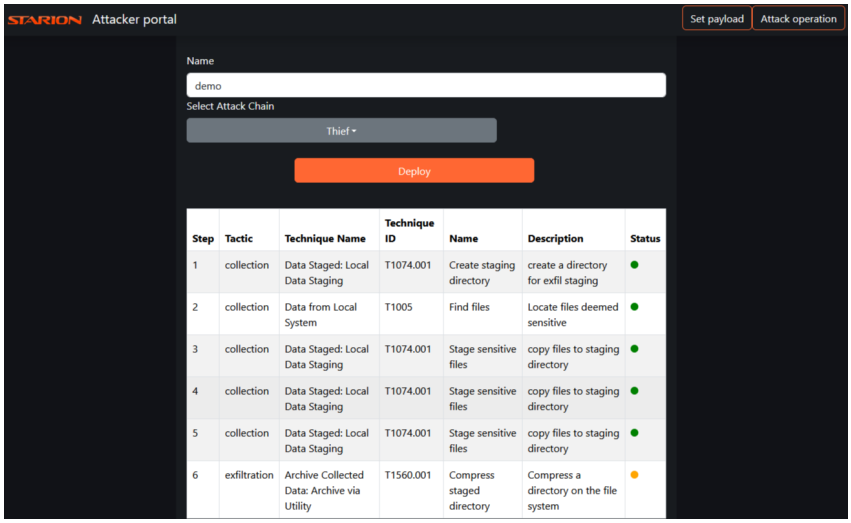

Figure 5.6: Attack Operation deploy - Data Exfiltration

These results therefore have a dual significance: on the one hand, they demonstrate the TRA's ability to orchestrate complex attack scenarios and monitor their evolution in real time; on the other hand, they highlight how the combination of realistic simulations and detection capabilities is a fundamental tool for identifying vulnerabilities in systems in advance. Furthermore, the availability of details of each phase of the attack has enabled a complete reconstruction of the sequence, which is useful for both auditing and potential forensic analysis.

5.3 Isolation Forest model training

This section will present the results obtained following the training of the ML model based on *Isolation Forest*, as well as some key tests that demonstrate its effectiveness.

5.3.1 Model training result

The model was trained using a dataset generated during a period of constant monitoring on the systems monitored by the TRA.
The data used included values relating to the KPIs of the various components of each system and the related categorical variables. To represent them correctly, the One-Hot Encoding method was applied, which allowed each component of the system to be encoded as a unique and independent binary variable, avoiding the need to sort the data a priori.
In this way, the model can analyse the data uniformly, allowing the detection of anomalies not only in the numerical variables but also in their distribution across the various categories.

The first training phase conducted on the key computational metrics of the 5GC elements, namely CPU and RAM, showed that the model is able to differentiate between normal and abnormal behaviour with sufficient reliability. This result can be seen in Figure 5.7, where the horizontal axis represents CPU usage [%] and the vertical axis represents RAM usage [%]. The green dots correspond to behaviours considered normal, while the red dots identify anomalies.
The results show that most regular behaviours are concentrated in a condition of low memory and CPU usage, consistent with typical system operation. On the other hand, anomalies are mainly distributed in areas where high memory usage and/or CPU load can be observed.

By associating these results with behavioural analyses, it is easy to verify any performance peaks or actual anomalies in the monitored system, which could therefore represent a malfunction or a risk situation.

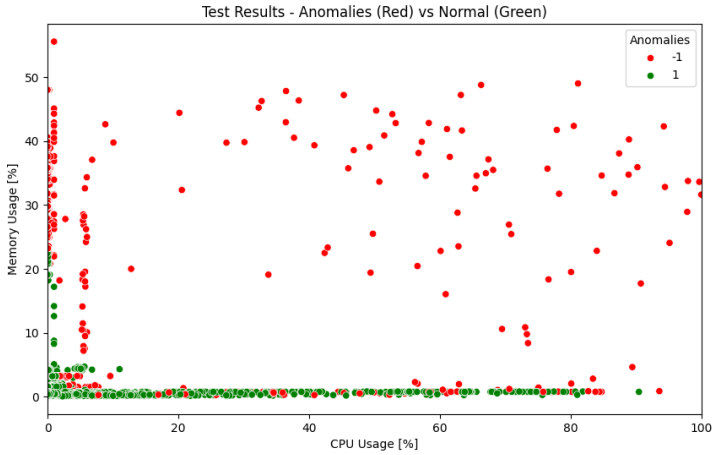

Figure 5.7: Anomaly Detection model training result - 5GCN

The model was subsequently trained on a second system with different operating characteristics. The results, shown in Figures 5.8, demonstrate how the algorithm adapted to these behavioural patterns as well, continuing to distinguish between normal and abnormal conditions. In particular, in this case, normal behaviour is observed near extreme CPU values.

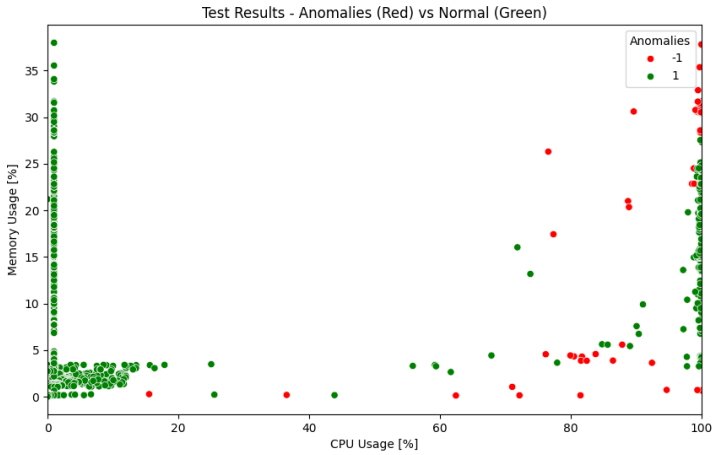

Figure 5.8: Anomaly Detection model training result - Satellite image processing

5.3.2 Model training validation test

Following the model training phase, tests were carried out to validate its effectiveness. To do this, data extrapolated from previous simulations that had triggered false positives were injected into the system. These were then associated with different components to verify the correct association of the model with the behavioural patterns of each of these. The results can be seen in Figures 5.9 & 5.10, where it can be observed how the highlighted point changed its status depending on the associated component. In the first case, the values were associated with the `AMF` component of the 5GC, while in the second case, they were associated with the `database` component of the same system, which clearly has a different behavioural pattern from the component examined previously.

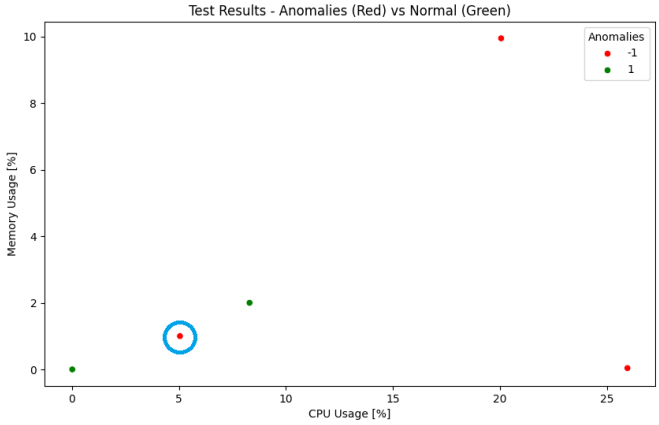

Figure 5.9: Model training test result - database

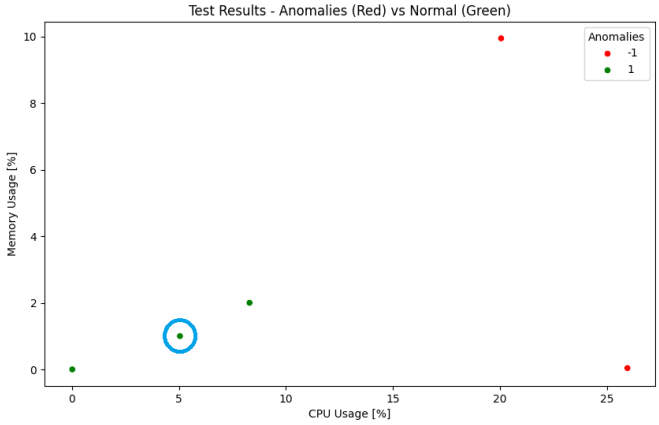

Figure 5.10: Model training test result - AMF

However, it is important to note that some false positives still remain. This reflects one of the main challenges of anomaly detection systems, the balancing of sensitivity. A highly sensitive model is capable of detecting even the slightest anomalous behaviour, but tends to classify as anomalies even behaviours that are actually part of normal operational variability, such as occasional performance peaks.

Chapter 6

Conclusions

The main objective of this thesis was to create an innovative solution to strengthen the security of computer systems, and this was achieved through the design, implementation and validation of the Threat Risk Assessor. During the course of the work, the TRA evolved from a small prototype to a functioning and validated platform, demonstrating that it had laid the foundations for the development of a solution capable of addressing contemporary risk scenarios.

The development process was divided into several phases, starting with an analysis of the state of the art and the definition of the theoretical framework of reference, then moving on to the design of the system architecture and its implementation, and finally ending with experimental validation in both controlled and real scenarios.
This step-by-step approach not only confirmed the technical feasibility of the TRA, but also highlighted its innovative contribution compared to traditional approaches. In particular, it was demonstrated that the TRA is not limited to passive monitoring, but represents an ecosystem capable of integrating several key functionalities in a continuous cycle, such as:

- Collection and correlation of metrics and logs from heterogeneous systems
- Detection of anomalies through ad-hoc algorithms, look-up tables and **Anomaly Detection** module
- Orchestration of simulated attacks through the **Attacker Portal**, with the possibility of building and distributing complete attack chains
- Production of real-time reports and notifications, useful for supporting the decision-making process by security operators of monitored systems
- Data archiving to enable incident reconstruction and related forensic analysis.

6.1 Project development and progress

From the earliest stages of conception, TRA was designed as a modular and dynamic system, so that new features could be added progressively in response to emerging operational and technical requirements. Its evolution can be summarised in the following key phases:

- Initial prototyping: TRA focused primarily on collecting and visualising data from monitored systems, without extensive correlation or simulation capabilities
- Data pipeline consolidation: the introduction of *Fluentd* and *Kafka* made it possible to efficiently manage the acquisition and management of data from a large number of components and systems, eliminating data loss and significantly reducing processing latency
- Integration of the **Attacker Portal**: its introduction, together with the `Attack Operation` functionality, represented a turning point, allowing realistic attacks to be orchestrated and transforming the TRA into a proactive security platform
- Introduction of the **Anomaly Detection** module: during the initial tests, the problem of excessive alert generation emerged, often not directly related to real risk conditions. This led to the need to introduce a module based on the Isolation Forest machine learning technique, which is constantly refined thanks to new datasets collected over time. This has significantly reduced the noise caused by false positives, increasing the reliability of the entire platform.

6.2 Contributions and added value

The validation of the TRA took place in two complementary phases: one in a controlled test environment, useful for verifying the correctness of the operating flow and the stability of the system under reproducible conditions; and one in real scenarios, aimed at demonstrating its effectiveness in complex operational contexts.
In both phases, the platform monitored the entire evolution of events, correlating data and generating alerts and associated risk scores, correctly leading to the creation of timely and useful reports for operators.

The results collected allow us to draw the following general conclusions about the innovative contributions of TRA:

- Integrated approach: the platform not only detects threats, but also contextualises and documents them. It also allows them to be detected promptly through a proactive approach of threat simulation
- Adaptability: tests have demonstrated TRA's ability to operate effectively in different contexts
- Transparency and traceability: the centrality of logs and reports ensures a high level of accountability
- Decision support: thanks to the notification system, risk scoring and event correlation, the TRA provides operators with timely information that allows them to take prompt action on the systems.

6.3 Limitations and considerations

Despite the positive results that emerged during the trials, it is worth noting some limitations that characterise the current implementation of TRA. These aspects do not compromise the validity of the work carried out, but highlight areas that require further investigation and which also represent ideas for future developments.

The first element concerns the scalability of the system. Although tests have demonstrated the TRA's ability to effectively detect and manage targeted attack scenarios, its behaviour still needs to be validated in contexts characterised by extremely high traffic volumes, such as large-scale mobile networks or distributed systems with a high density of devices.
In these cases, the workload on the monitoring components could generate bottlenecks and slow down the timeliness of the analyses, making it necessary to focus on optimising performance.

Another aspect to consider is the reduction of false positives. The introduction of the **Anomaly Detection** module has significantly improved the quality of detections, but its effectiveness is closely linked to the quality and quantity of available data, implying a continuous process of maintenance and adaptation.

6.4 General conclusion

In conclusion, the work carried out has demonstrated in a clear and empirical manner the effectiveness of TRA as an innovative solution for system security. Through a process of theoretical analysis, architectural design, controlled experimentation and validation, it was possible to verify the validity of the design choices and the achievement of the initial objectives.

TRA has proven to be a platform capable of offering concrete tools for monitoring, detecting and simulating threats. This work therefore aims to make a significant contribution to the cybersecurity field, highlighting how a proactive and integrated approach is essential for building resilient and secure infrastructures capable of effectively addressing the challenges posed by the current scenario, with its constantly evolving threats.

Chapter 7

Next Steps

The research and development carried out to this point has demonstrated how TRA can represent an innovative approach to risk management within IT systems.
Although already rich in functionality, the current architecture is only the basis for a technology that can evolve further. For this reason, it is possible to outline a series of next steps that define the direction of TRA development, both technologically and methodologically.

7.1 Automated attacks

A key step will be to define an automated process of targeted attacks, designed as an integral part of red teaming and resilience assessment procedures. Through this integration, the system will be able to perform automated tests, generating threat scenarios tailored to each system, recreating those possible in the real world. The results of these simulations will be used to produce detailed reports documenting the system's responsiveness and possible areas for improvement. This will enable the establishment of a cycle aimed at proactively addressing possible threats to the monitored system.

7.2 Automatic response engine

Another important element that is planned to be implemented is an automatic incident response engine, designed to perform defensive actions in a semi-autonomous manner. The intended logic is not to replace the human operator, but to reduce reaction times by preparing a set of operations ready to be applied automatically following approval by the operator, who will remain the final element of control.
For example, the system will be able to activate temporary firewalling rules, isolate a compromised node or automatically reconfigure traffic routing policies.
This human-in-the-loop approach represents an optimal compromise between response speed and decision-making reliability.

7.3 Evolution of the Anomaly Detection module

With regard to the Anomaly Detection module, the next steps will involve evaluating and testing additional ML models, including not only algorithms based on classic approaches (e.g. *Isolation Forest*, *Random Forest* [60] and *One-Class Classification* [59]), but also Deep Learning models capable of detecting complex patterns and non-linear temporal dynamics.
These tests will require a benchmarking phase aimed at comparing the effectiveness, training times and scalability of the different approaches, with the aim of selecting the most suitable solution for the operating context.

Another key aspect for the evolution of this module would be to find a solution to reduce false positives. From a practical point of view, these cause an increase in the workload for operators and can reduce the reliability of the system. To reduce them, strategies such as the use of adaptive thresholds and the integration of contextual information can be evaluated. In particular, unlike static thresholds, adaptive thresholds are dynamically updated based on historical trends and the current load on the systems, preventing any variations from being detected as anomalies.
On the other hand, the integration of contextual information enriches the model with additional data in order to improve its ability to distinguish legitimate variations from real situations.

7.4 Additional possible developments

Some additional developments aimed at improving and innovating the solution could be as follows:

- Integration with external threat intelligence systems: connect the TRA to real-time feeds from verified databases, thereby enriching the information context and improving prevention capabilities
- Regulatory compliance support: add functionality for the automatic generation of reports that comply with standards and regulations (e.g. General Data Protection Regulation (GDPR) [54] and NIS2 [9]), thus facilitating auditing activities
- Integration with digital twins: create a digital-twin of the monitored system, on which to test hypothetical scenarios and assess the impact, as well testing counter-measures, without affecting the real system
- Creation of a module capable of creating, following a system scan, the corresponding threat model based on STRIDE[1].

[1]STRIDE: it is a widely used threat modelling framework proposed by Microsoft [63], which classifies threats into six categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service ed Elevation of Privilege.

Acronyms List

3GPP 3rd Generation Partnership Project.

5GC 5G Core netowrk.

AC Attack Complexity.

AMF Access and Mobility Management Function.

API Application Programming Interface.

AUSF AUthentication Server Function.

AV Attack Vector.

BAS Breach and Attack Simulation.

C2 Command and Control.

CEM Continuous Exposure Management.

CF Charging Function.

CN Core Network.

CPU Central Processing Unit.

CS Circuit Switching.

CVSS Common Vulnerability Scoring System.

DBSCAN Density-Based Spatial Clustering of Applications with Noise.

DDoS Distributed Denial of Service.

DoS Denial of Service.

DRA Dynamic Risk Assessment.

ENISA European Network and Information Security Agency.

EO Earth Observation.

ETSI European Telecommunications Standards Institute.

FIRST Forum of Incident Response and Security Teams.

GDPR General Data Protection Regulation.

GGSN Gateway GPRS Support Node.

GPRS General Packet Radio Service.

GSM Global System for Mobile communications.

HSS Home Subscriber Server.

IDS Intrusion Detection System.

IMSI International Mobile Subscriber Identity.

IoC Indicator of Compromise.

JWT JSON Web Token.

KPI Key Performance Indicator.

MBSE Model-Based System Engineering.

MITRE ATT&CK® MITRE Adversarial Tactics, Techniques, and Common Knowledge.

MME Mobility Management Entity.

MSC Mobile Switching Centre.

N3IWE Non-3GPP InterWorking Edge.

N3IWF Non-3GPP Interworking Function.

NF Netowrk Function.

NGAP Next Generation Application Protocol.

NRF Network Repository Function.

NSSF Network Slicing Selection Function.

NTN Non-Terrestrial Network.

PCF Policy Control Function.

PCRF Policy and Charging Rules Function.

PGW Packet GateWay.

PR Privileges Required.

PS Packet Switching.

QoS Quality of Service.

RAM Random Access Memory.

RAN Radio Access Network.

SBA Service-Based Architecture.

SGSN Serving GPRS Support Node.

SGW Serving GateWay.

SMF Session Management Function.

TRA Threat Risk Assessor.

TTP Technique, Tactic and Procedure.

UDM Unified Data Management.

UDR Unified Data Repository.

UE User Equipment.

UERANSIM User Equipment RAN SIMulator.

UMTS Universal Mobile Telecommunications System.

UPF User Plane Function.

VM Virtual Machine.

VNF Virtual Netowrk Function.

Bibliography

[1] 3gpp - release 15.
[2] 3gpp - release 16.
[3] 3gpp - release 18.
[4] 3gpp - release 20.
[5] About 3gpp.
[6] Kafka.
[7] Caldera documentation - app.api.v2 package.
[8] Caldera documentation - basic usage - facts.
[9] European Commission. Nis2 directive: securing network and information systems.
[10] Dev Cookies. Kafka architecture overview.
[11] Devopedia. 5g security.
[12] Docker hub - bitnami/kafka.
[13] Docker compose.
[14] Docker hb - python:3.9-slim.
[15] Elasticsearch - kibana.
[16] Elasticsearch.
[17] Flask.
[18] Fluentd.
[19] Geeks For Geeks. One hot encoding in machine learning, 2025.
[20] Filippo Giambartolomei. Penetration testing applied to 5g core network. *University of Padova - Thesis*, 2023.
[21] Filippo Giambartolomei, Marc Barcelo, Alessandro Brighente, Aitor Urbieta, and Mauro Conti. Penetration testing of 5g core network web technologies. *IEEE International Conference on Communications (ICC)*, 2023.
[22] Muly Gottlieb. Using parallel processing in node.js and limitations.
[23] Grafana - loki.
[24] Grafana - promtail.
[25] Grafana - infinity.
[26] Grafana.
[27] Starion Group. Model-based system engineering.
[28] Json web tokens - jwt.io.
[29] John Burke Kinza Yasar. What is network slicing?, 2024.
[30] Sarp Koksal. 3g core network architecture.

[31] Sarp Koksal. Evolution of core network (3g vs. 4g vs. 5g).
[32] Jannik Lindner. Worldmetrics.org report 2025, 2025.
[33] Marco Barros Lourenco, Louis Marinos, Lampros Patseas, and EU Agency for Cybersecurity. Enisa threat landscape for 5g networks. *ENISA*, 2020.
[34] Minio.
[35] Caldera.
[36] MITRE. Data from local system.
[37] MITRE. Data staged: Local data staging.
[38] MITRE. Exfiltration over c2 channel.
[39] Node.js.
[40] Open5gs.
[41] Conor O'Sullivan. Isolation trees, 2024.
[42] Konstantinos Rantos Pavlos Cheimonidis. Dynamic risk assessment in cybersecurity: A systematic literature review, 2023.
[43] Prometheus.
[44] Prometheus - alertmanager.
[45] Pentera.
[46] Zujany Salazar, Huu Nghia Nguyen, Wissam Mallouli, Ana R. Cavalli, and Edgardo Montes de Oca. 5greplay: a 5g network traffic fuzzer - application to attack injection. *ARES 2021: The 16th International Conference on Availability, Reliability and Security*, pages 1–8, 2021.
[47] PICUS security. Breach and attack simulation for cyber resilience.
[48] Check Point Team. Check point research reports highest increase of global cyber attacks seen in last two years – a 30
[49] TechDifferences. Difference between 3g and 4g technology.
[50] Ueransim.
[51] Wikipedia. Apache kafka.
[52] Wikipedia. Api.
[53] Wikipedia. Dbscan.
[54] Wikipedia. General data protection regulation.
[55] Wikipedia. Gprs.
[56] Wikipedia. International mobile subscriber identity.
[57] Wikipedia. Isolation forest.
[58] Wikipedia. Lookup table.
[59] Wikipedia. One-class classification.
[60] Wikipedia. Random forest.
[61] Wikipedia. Single-responsibility principle.
[62] Wikipedia. Social engineering (security).
[63] Wikipedia. Stride model.
[64] Wikipedia. User agent.
[65] Wikipedia. Webhook.
[66] Gokberk Yaltirakli. Slowloris.