Opole University of Technology

Ph.D. Thesis

Threat Modeling methods and detecting vulnerabilities in 5G networks.

Author: mgr inż. Wiktor Sędkowski

Supervisor: dr hab. inż. Rafał Stanisławski, prof. PO

Opole, 2024 r.

Abstract: This dissertation focuses on the systematization of techniques and processes related to the automation of threat modeling and vulnerability detection in modern, dynamic systems such as 5G networks. The evolution of 5G technology represents a significant shift from traditional, closed 3G and 4G telecommunications networks towards more open, Internet Protocol (IP)-focused structures. The dynamic nature of 5G architecture introduces new security threats, necessitating innovative threat modeling enhancements to the known techniques. Automating this process and modeling potential threats automatically remains a significant challenge for cybersecurity experts, service providers, and security offices. Solutions proposed in this works address the problem of automation of vulnerability prediction and threat modeling activities in complex systems like 5G networks. The study proves that it is possible to effectively model and prioritize threats in 5G networks considering their variable structure, and that adaptive computational algorithms can model and prioritize threats in real-time. Comparative analyses presented in the dissertation show that artificial intelligence can be useful for predicting vulnerabilities, generating threat related diagrams, and descriptions for threat models.

Keywords: Cybersecurity, Threat Modeling, 5G Networks, Vulnerability Detection, Automation, Artificial Intelligence, Network Security, Risk Mitigation

Metody modelowania zagrożeń i wykrywania podatności w sieciach 5G.

Rozprawa doktorska

Autor: mgr inż. Wiktor Sędkowski

Promotor: dr hab. inż. Rafał Stanisławski, prof. PO

Streszczenie: Niniejsza rozprawa koncentruje się na usystematyzowaniu technik i procesów związanych z automatyzacją procesu modelowania zagrożeń i wykrywania luk w nowoczesnych, dynamicznych systemach, takich jak sieci 5G. Ewolucja technologii 5G w kierunku bardziej otwartych struktur skoncentrowanych na protokole internetowym (IP) odróżnia ją od tradycyjnych zamkniętych sieci 3G czy 4G. Dynamiczny charakter architektury 5G wiąże się z nowymi zagrożeniami bezpieczeństwa, wymagając innowacyjnego podejścia do procesu modelowania zagrożeń, gdyż automatyzacja procesu modelowania potencjalnych zagrożeń pozostaje poważnym wyzwaniem dla dostawców usług i ekspertów ds. cyberbezpieczeństwa. Proponowane w niniejszej pracy rozwiązania automatyzują proces przewidywania podatności i modelowania zagrożeń w złożonych systemach, takich jak sieci 5G. Przeprowadzone badania wykazały, że możliwe jest skuteczne modelowanie i priorytetyzacja zagrożeń w systemach o zmiennej strukturze. Uzyskane wyniki potwierdzają, że algorytmy obliczeniowe mogą efektywnie modelować i priorytetyzować zagrożenia w czasie rzeczywistym. Analizy porównawcze przedstawione w pracy wskazują, że sztuczna inteligencja może być użyteczna w przewidywaniu podatności, generowaniu diagramów związanych z zagrożeniami oraz w tworzeniu opisów modeli zagrożeń.

Słowa kluczowe: Cyberbezpieczeństwo, modelowanie zagrożeń, sieci 5G, wykrywanie podatności, automatyzacja, sztuczna inteligencja, bezpieczeństwo sieci, ograniczanie ryzyka

Contents

List of abbreviations and symbols

5GC	5G Core Network
5G-IWF	5G Interworking Function
A	Asset
AMF	Access and Mobility Management Function
API	Application Programming Interface
AR	Augmented Reality
ARMS	Attack Risk Mitigation Services
ARPF	Authentication Credential Repository and Processing Function
AUSF	Authentication Server Function
B2B	Business to Business
BBU	Base Band Unit
BERT	Bidirectional Encoder Representations from Transformers
BS	Base Station
CDMA	Code Division Multiple Access
CKC	Cyber Kill Chain
CoMP	Coordinated Multi-Point
CPE	Control Plane Entity
CU	Central Unit
CUPS	Control and User Plane Separation
CVSS	Common Vulnerability Scoring System
D2D	Device to Device

DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
DSM	Dynamic Spectrum Management
DU	Distributed Unit
ECPRI	Enhanced Common Public Radio Interface
EPC	Evolved Packet Core
FWA	Fixed Wireless Access
GLUE	General Language Understanding Evaluation
GPT	Generative Pre-trained Transformer
GSM	Global System for Mobile Communications
GUTI	Globally Unique Temporary UE Identity
HPLMN	Home Public Land Mobile Network
HSS	Home Subscriber Server
Hz	Hertz
IIoT	Industrial Internet of Things
IMSI	International Mobile Subscriber Identity\
IoT	Internet of Things
JSON	JavaScript Object Notation
LLM	Large Language Model
LLS	Lower-Layer split
LTE	Long-Term Evolution

MBSFN	Multicast-Broadcast Single-Frequency Network-based
MEC	Multi-access Edge Computing
MIMO	Multiple-Input Multiple-Output
MME	Mobility Management Entity
MNO	Mobile Network Operator
MRPC	Microsoft Research Paraphrase Corpus
NEF	Network Exposure Function
NFV	Network Functions Virtualization
NGRAN	Next Generation Radio Access Network
NR	New Radio
NSE	Nmap Scripting Engine
NSSF	Network Slicing Selector Function
OFDMA	Orthogonal Frequency Division Multiple Access
P(x)	Probability of event X
PFD	Process Flow Diagram
PGW	Packet Data Network Gateway
PLMN	Public Land Mobile Network
RAN	Radio Access Network
RF	Radio Frequency
RRU	Radio Reciever Unit
SBA	Service-Based Architecture
SCS	Subcarrier Spacing

SDLC	Software Development Lifecycle
SDN	Software Defined Network
SEAF	Security Anchor Function
SEPP	Security Edge Protection Proxy
SGW	Serving Gateway
SIM	Subscriber Identity Module
SOA	Service-Oriented Architecture
SUCI	Subscriber Concealed Identifiers
SUPI	Subscriber Permanent Identifiers
TFD	Transfer Function Domain
TMSI	Temporary Mobile Subscriber Identity
UPF	User Plane Network Functions
VEPA	Vulnerability Exploit Pattern Analyzer
VEPG	Vulnerability Exploit Pattern Generator
VM	Virtual Machine
VPLMN	Visiting Public Land Mobile Network
XML	Extensible Markup Language
XR	Extended Reality
YAML	Yet Another Markup Language

1. Introduction

The main topic of this dissertation is the development of techniques and processes related to automation of threat modeling and vulnerability detection in modern, dynamic systems such as 5G networks. 5G technology developed and deployed in recent years, is responsible for significant changes consisting in moving away from closed 3G and 4G telecommunications networks towards Internet Protocol IP-focused structures related to Service Based Architecture.

The development of 5G technology is motivated by a diverse set of functionalities and demands. The Fixed Wireless Access (FWA) for dense metropolitan areas is one of the most significant commercial offerings. Additional applications include those that require specialised coverage, vertical solutions such as connected vehicles, manufacturing, industry 4.0, Industrial Internet of Things (IIoT), energy, and healthcare. It is widely acknowledged among experts that verticals will play a major role in the future implementation of next-generation networks. For instance, 5G technology offers significant advantages in industrial applications that require increased data rates and reduced latency, such as augmented reality (AR) and artificial intelligence (AI)-based systems. The presence of substantial bandwidth capacities will provide the uninterrupted transmission of high-resolution photos and video streaming, akin to surroundings abundant in sensors and characterised by a high density of connections. Investment strategies of Mobile Network Operators (MNOs) will be significantly influenced by these factors.

The security framework of 5G encompasses various domains, namely Network Access Security, Network Domain Security, User Domain Security, Application Domain Security, and Service-Based Architecture (SBA) Domain Security. In the 5G core network, several security functions have been implemented, including the Authentication Server Function (AUSF), Authentication Credential Repository and Processing Function (ARPF), and Security Anchor Function (SEAF). The security needs of 5G encompass several aspects such as authentication and authorization, maintaining user data and signalling data confidentiality and integrity, and safeguarding subscribers' privacy [1]. Privacy in the 5G network is ensured through the utilization of Subscriber Permanent Identifiers (SUPI), Subscriber Concealed Identifiers (SUCI), and Globally Unique Temporary UE Identity (GUTI). Ensuring optimal performance necessitates the implementation of end-to-end security monitoring across the entire 5G architecture, encompassing devices, apps, and networks. Several significant issues arise in this context, namely the security of radio interfaces, the preservation of user plane integrity, and the vulnerability to Denial of Service (DoS) attacks against both the infrastructure and end-user devices. This new architecture involves new, dynamically changing security threats. Therefore, a new way of modeling these threats must be developed, one that will allow network operators to adapt to dynamic changes in the infrastructure and ensure the security of their resources and, above all, network users. Automating the risk prioritisation process and modeling potential threats is still a big challenge for experts, companies providing cybersecurity services, and security offices.

Since their beginnings, mobile communication systems have been susceptible to security flaws. During the initial phase (1G) of mobile networks, there was a notable occurrence of illicit cloning and impersonating activities targeting mobile phones and wireless channels. During the second generation (2G), the prevalence of message spamming increased, encompassing both ubiquitous attacks and the disinformation of false information or unwished marketing content. The advent of the third generation (3G) facilitated the transfer of Internet security vulnerabilities and threats to the wireless domain through the utilisation of IP-based communication. The expansion of smart gadgets, multimedia traffic, and new services into the mobile domain related to the rollout of fourth generation (4G) of IP-based communications, resulted in a more complex and ever-changing environment. In the 5G network, and especially in its part based on cloud solutions, the security problems related to a dynamically changing environment are also the case. The constant growth of telecommunication networks both in regard to subscriber count and complexity, creates significant requirements related to automation of processes for threat modeling and detection of vulnerabilities.

The dissertation covers results of the study, which aimed to develop a methodological basis (toolbox) that will support engineers responsible for network security in making practical decisions regarding security mechanisms and their implementation.

1.1. Research problem

Threat modeling is a process designed for identification of the potential threats in the system. Main goal of the threat modeling is to identify weaknesses in the architecture of the system and to mitigate possible threats. Artificial Intelligence (AI) or to be more precise, its technical implementation can heavily extend human capabilities to create ad-hoc threat models for dynamic digital systems. Calculations performed by the machine based on the historical data and predictions generated by AI can not only reduce the cost of threat model preparation but also increase its quality. Threat model is the term used to describe a group of security features, or a set of likely attacks that could compromise the functionality of any system. Through the use of threat modeling methodology, security specialists are able to recognise security threats and create defences during the phases of design, coding, and testing [1]. Thus, it helps in one of the most crucial steps in creating a safe application, which is to analyse and predict the possible dangers that it may encounter [3]. The primary goal of threat modeling is to offer practical recommendations for reducing the related risks for all resources that are vulnerable to threats.

The main aim of this research work is to systematise the AI techniques applicable in creating or supporting the process of threat modeling. The dissertation includes examples showing how each of the proposed techniques can be applied to bring solutions useful for threat modeling purposes.

1.2. Scope of the work

Work on this dissertation was carried out as part of the "implementation doctorate" program as a cooperation between the Opole University of Technology and Nokia Solutions and Network sp. z o.o. (Nokia). The research and the results achieved are in line with the needs of Nokia in terms of developing innovative solutions related to the improvement of security related processes.

Nokia outlined its plan to achieve profitable, sustainable development by leading the B2B technology innovation space and introducing a new mission and organizational structure in 2034. The company has set out to reset, speed up, and rescale to become leaders in a world where widespread digitalization is gathering speed. Security is one of the pillars of successful digitalization of governments, webscalers[1], and businesses which depend more and more on Nokia's vital networking technologies. Nokia Technology Vision 2030 [4] states that networks are key to opening up the vast prospects in the industrial, enterprise, and consumer domains. The underlying networks will need to change as they transition from the 5G era to 6G and 5G-Advanced. Digital twins, biosensors, and fully immersive augmented and virtual reality will all place demands on the networks of the upcoming ten years. The consumer, enterprise, and industrial metaverses, in the near future will be able to fulfill their full potential thanks to these technologies. True extended reality (XR) experiences will be made possible by them, and eventually this will result in the fusion of the actual and virtual worlds and the improvement of humanity. It will take a major change in connectivity for these technologies to be successfully and quickly implemented, and during that time it is essential to include security aspects in the working scope. This dissertation scope fits into the chance of seizing to lead the way in the next phase of networking development, by introducing AI in automation of security related processes.

AI has taken over news headlines, and even while the media's enthusiasm about it may wain, the technology's impact and influence will only grow. Artificial Intelligence (AI) will develop code, optimize processors, and shrink hardware. AI will change healthcare, speed up the time to market for our products, and streamline industrial operations. Though cloud computing is by no means a novel idea, the upcoming release of cloud technologies will bring about even more disruption. A seamless cloud continuum that replaces discrete, siloed clouds will revolutionize all facets of corporate operations, including product and service delivery and relationships with partners, clients, and staff. With public, private, hybrid, multi-cloud, and cloud-enabled edge devices spread across multiple geographic zones, cloud systems are expected to get increasingly complicated. More and more things are being included in the definition of connectedness than only information exchange. The global network has already changed, and we are connecting more than just endpoints.

[1] Company, technology, or system designed to efficiently scale web applications and services to handle large volumes of traffic and data. These large tech companies like Google, Amazon, or Microsoft operate at massive scales, both in terms of users and infrastructure. They use custom-built, highly scalable systems and technologies to manage their operations effectively.

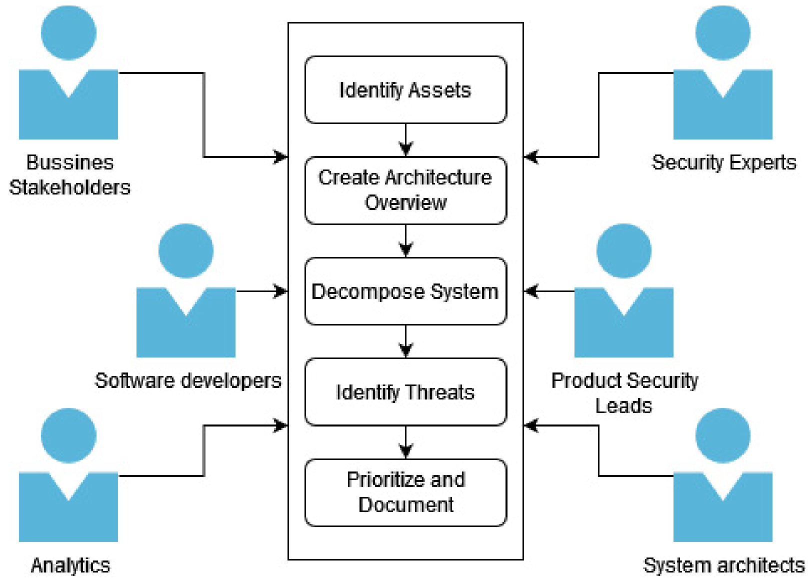

Figure 1: Threat modeling.

Source: Own

Technology companies are linking resources, clouds, and subnetworks together. This is the main reason why manual threat modeling processes are extremely hard to apply for such systems. Not only because there are so many assets that need to be considered but also due to people resources that are normally involved in the multidisciplinary team that includes business stakeholders, security experts, software developers, system architects, and many more as shown in Figure 1. According to the Ku Leuven [5] research, the duration of a threat modeling exercise is seen as challenging by the participants. Teams may be reluctant to begin threat modeling a large system or application due to the amount of time that must be invested and may lose interest if a session goes on for too long, especially if it is dominated by one or a small number of people or becomes too technical. Time is not the only constraint. Lack of security expertise is in fact the main reason why teams are not confident enough to successfully run threat modeling process. Besides expert knowledge about the systems (networking, virtualization, software, telecommunication, etc.) threat modeling methodology fundamentals should be known to team members. Teams often do not possess this specific expertise [5], therefore automation and the introduction of Artificial Intelligence in this process seem to be an interesting change and a process gap which this dissertation tries to fill. This research not only advances the understanding of the security challenges but also lays the foundation for the development of adaptive security measures essential for safeguarding the integrity and reliability of next-generation telecommunications systems.

1.3. Thesis

The study focuses on selecting methodology and developing algorithms supporting modeling and assessing vulnerabilities and threats in 5G networks. The distributed and virtualized nature of 5G networks, coupled with a vast array of interconnected devices and applications, creates fertile ground for potential security threats. Traditional security measures may prove inadequate in addressing the dynamic and multifaceted nature of these threats, necessitating innovative approaches grounded in advanced computational techniques. The variable and unpredictable nature of security threats suggests that heuristic tools based on artificial intelligence may prove particularly effective in modeling this class of problems.

Two research hypotheses were specified for this work:

- It is possible to effectively model and prioritize threats occurring in 5G networks, considering the variability of the network structure.
- It is possible to use automated, adaptive computational algorithms operating in real time to model and prioritize threats.

The development of methodology and algorithms for modeling and assessing the vulnerability of threats in 5G networks represents a crucial endeavor in ensuring the security and resilience of future telecommunications systems.

1.4. Aim of the work

In recent years, network security has become one of the most important concerns of companies and governments around the world. Each year the investments made in the cybersecurity sector are increasing, global venture funding for the cybersecurity industry's constant growth reached its peak in 2022, exceeding $23 billion. Although the investments dropped in 2023 to 8.2 billion this is still a 120% increase in comparison to 2015 and 48% in comparison to 2018 cybersecurity investments. According to the Gartner report, the end-user spending for the information security and risk management market will continue to grow at a compound annual growth rate of 8.2% through 2024 to reach $207.7 billion in the constant currency [6]. The importance of cybersecurity can also be observed by comparing the number of research papers published in recent years. Querying Google Scholar database in search for papers containing the "cybersecurity" keyword shows that this relatively narrow field is becoming more and more popular. From only 56 publications in 2010 the annual growth of research papers published per year is constantly rising. For example, in 2015, there were 259 new publications; in 2020 there were 2250 publications; and in 2023, the Google Scholar database indexed 6560 articles for the same keyword.

Cybersecurity focuses on guarding access to networks and their resources, on protecting from data theft and damage to software and hardware components. Previously mentioned major growth of interest in cybersecurity emerged from the intensification of cyberattacks and increased dependency of humans on digital systems and the internet. With the increase in different types of attacks, cybersecurity becomes a big challenge for researchers. Many categories of research are conducted in this field. Moreover, many security controls and security solutions are implemented in a variety of digital systems. The foundation for the applicability of those security mechanisms that are possible to be implemented in a system rests in the threat modeling process. Both from the cost and performance perspective, it makes little sense to implement every possible security solution into a single system. This is the primary reason for the importance of risk analysis and threat modeling. They serve a filtering function, allowing system owners to select appropriate security controls and address only the threats relevant to the specific system or solution.

Suitable identification of threats and selection of an appropriate set of countermeasures not only reduces the ability of attackers to misuse the functions of a system but also cuts the cost of the development related to patching security vulnerabilities at a later stage. The threat modeling process surveys the system from an adversary's perspective, identifying potential attacker goals. This kind of activity requires not only expert knowledge about the system itself, but also detailed information about the components and the technology in use and a wide understanding of cyber threats and related risks. Artificial Intelligence, with its dynamically growing potential in exploring big data, computing, and categorizing the information at scale, is a great fit for automation of threat modeling activities. Hence, the AI-based solutions have been widely deployed to solve problems related to cybersecurity. Recent availability of commercial AI based products is opening new opportunities for dynamic solutions in the domains of security, privacy, and threat detection. This is a great opportunity for modeling threats in complex systems such as next generation networks, for which manual threat modeling process is extremely time consuming.

The 5G networks, together with their SBA (Service-Based Architecture) using virtualization and container technologies supported by cloud-based processing platforms, can benefit a lot from AI assisted and automated threat modeling activities. A great variety of use cases for 5G systems, virtualized implementation, and cloud processing result in distinct and increased security concerns. Therefore, the automation of processes related to threat identification can improve the security posture of dynamically changing 5G and Next Generation networks virtualized deployments.

The aim of this research is not only to prove that it is possible to utilise AI capabilities to play a role in securing modern networks, but also to implement such solutions.

1.5. Methods used during research

This work focuses on using automation and machine learning methods and algorithms in order to support threat modeling and vulnerability detection processes. There are two different problems that will be solved within this thesis. First, detection of vulnerabilities with use of machine learning algorithm based on data available in publicly available exploit database. The second problem is automation of threat modeling process with use of custom algorithm, open source tools and Generative Artificial Intelligence optimized for this task.

The research is divided into two parts and is based on well established, existing methods and metrics. For the theoretical part aimed at selection of threat modeling methodology, iterative knowledge discovery and comparative methods were used. This process included the determination of quality criteria, a review of existing research, and publicly available threat-related metrics and algorithms. The comparative method applied a mix of both qualitative and quantitative research approaches. This mixed-method is usually more useful when the study being done is end-to-end, examining a problem from beginning to end, as opposed to determining whether a problem exists or identifying its cause, which perfectly fits into the research problem in question. This flexible study technique required more work than a single approach, but it was necessary to limit the consequences of improper data classification, which could lead to erroneous conclusions and misleading outcomes. Critical review, evaluation, and comparison set the groundwork for the second part of the research, that is, implementation.

In the practical part, where results of the theoretical research are implemented, a mix of quantitative and qualitative methodologies is used. Survey methodology is used in part of addressing the problem of perceived severity of vulnerabilities. A survey was performed to determine how the AI system should present the information for the user to correctly understand the severity of the security-related finding. Experiments were performed as part of the research related to the development of the software. Those included the experiments related to the evaluation of vulnerability detection capabilities and the evaluation of threat models generated automatically with the use of solutions prepared during research and described in Chapter 4. Last but not least, content analysis methodology based on the collection of qualitative data obtained from Generative Artificial Intelligence output was conducted.

1.6. Contribution

This research significantly advances the field of cybersecurity, particularly in addressing the complex challenges of threat modeling and vulnerability detection in 5G networks. The contribution encompasses both theoretical advancements and practical applications, with a strong emphasis on the integration of artificial intelligence into established security practices.

One of important contributions to security domain is the comprehensive systematization of threat modeling methodologies. Recognizing the unique vulnerabilities and attack vectors inherent in 5G networks, a thorough review and analysis of existing threat modeling methodologies is performed in this work. The identified strengths, limitations, and applicability to the networking landscape is correlated with artificial intelligence methods which can be used together with each of them. Building upon methodological systematisation deep investigation into the potential of AI to enhance threat modeling processes is conducted. Through a combination of literature review, case studies, and expert consultations, work evaluates the efficacy of various AI techniques in identifying vulnerabilities and threats. The outcome is a structured categorization that provides a clear roadmap for practitioners. This systematic approach not only clarifies the state-of-the-art but also reveals gaps and opportunities for further research.

To demonstrate the feasibility and effectiveness of proposed solutions, prototype tools were designed. The implementation and evaluation of a proof-of-concept AI-powered threat modeling tool is another significant contribution. This tool integrates automation and AI algorithms with existing threat modeling knowledge bases to automate and optimize key aspects of the threat modeling process, such as asset identification, attack surface analysis, and vulnerability prioritization. A preliminary evaluation of the tool indicates promising results in terms of accuracy, efficiency, and scalability.

Identification of future research directions at the intersection of AI and cyber security is yet another contribution. Research does not only address current challenges in threat modeling and vulnerability detection for 5G networks but also opens new avenues for further exploration. The identified promising research directions, including the development of more sophisticated AI algorithms for threat modeling, the integration of AI with other security tools and processes, set ground for other researchers in the field.

The research makes significant contributions to the field of cybersecurity by systematically analyzing existing threat modeling methodologies, rigorously evaluating the potential of AI in threat modeling, developing a novel solution for AI assisted threat modeling, and identifying future research directions. The work aims to improve the security and resilience of 5G networks, ultimately benefiting individuals and organizations.

1.7. Structure of the dissertation

This dissertation is organized into several chapters, each addressing distinct aspects of the research on threat modeling and Vulnerability Detection in 5G networks. The structure is designed to provide a comprehensive and logical flow from the introduction of the research problem to the detailed exploration of the methods, results, and contributions. Below is an overview of each chapter:

Chapter 1: Introduction sets the stage for the dissertation by presenting the research problem, the scope of the work, the thesis statement, and the aim of the work. It also outlines the methods used during the research, the contributions of the dissertation, and provides a brief overview of the structure of the dissertation along with the sources referenced.

Chapter 2: 5G Network provides a detailed overview of 5G networks, including their features and the 5G air interface. It covers key technologies such as beamforming, massive MIMO, dynamic spectrum sharing, and RAN virtualization. The core network components, including multi-access edge computing, cloud-native 5G core, user plane network functions, and CUPS, are also discussed. Additionally, the chapter delves into the 5G service-based architecture, threats targeting 5G networks, and critical elements in the 5G core.

Chapter 3: State of the Art in Threat Modeling, reviews the existing literature and methodologies in threat modeling. It introduces the basics of threat modeling and provides an overview of various methodologies, including data-centric, attack-centric, asset-centric, software-centric/system-centric threat modeling, STRIDE, VAST, CVSS, TRIKE, and the Cyber Kill Chain. The applicability of AI for threat modeling and automated threat modeling are also explored.

Chapter 4: Automating Threat Modeling Process is focusing on the automation of the threat modeling process. This chapter discusses data sources for automated threat modeling, the automation of threat diagram preparation, and the challenges in describing threats. It also examines the use of GPT for describing threats, including data collection, solution implementation attempts, and comparisons with similar tools.

Chapter 5: Automated Vulnerability Detection addresses the automation of vulnerability detection, starting with asset discovery and threat prediction. It details the processes of training data collection, developing AI models for vulnerable asset prediction, and vulnerability exploit pattern prediction. The chapter concludes with a presentation of the study results, implementation of research findings, and suggestions for further work and possible improvements.

Bibliography provides a list of all sources and references used throughout the dissertation. It includes books, journal articles, conference papers, and other relevant materials that support the research.

By following this structure, the dissertation systematically addresses the research questions, methodologies, findings, and implications of the study on automating threat modeling and vulnerability detection in 5G networks.

1.8. Sources

The main sources of information used for the research included academic literature, industry reports, public data, and empirical data collected using open-source tools. Each category of sources played a crucial role in construction of the framework for understanding the security threats inherent in 5G networks and contributed to the research objectives.

Academic literature formed the foundation of this research by offering theoretical insights and frameworks that inform the understanding of threat modeling and 5G networks use cases. Books, peer-reviewed journal articles, and conference papers were extensively reviewed to gather a comprehensive view of the current state of knowledge in the field. List of literature is available in separate chapter, however it is worth noting that in the wide range of journal articles which were consulted to explore specific aspects of 5G security, including vulnerabilities, threat vectors, and mitigation strategies, articles from high-impact journals such as IEEE Communications, Computer Networks, and IEEE Transactions were particularly valuable. These articles offered in-depth analyses of evolving security threats in 5G and provided empirical evidence supporting the need for advanced threat modeling techniques. Conference proceedings from events like IEEE Global Conferences (e.g. IEEE International Conference on Computer Communications - INFOCOM) provided the latest research findings and innovations in 5G security. These sources were instrumental in identifying emerging trends and novel approaches to threat modeling, reflecting the rapid pace of technological advancements and associated security challenges. Reports from industry leaders such as Nokia, Huawei, cybersecurity firms, and research companies such as Gartner, provided valuable data on current security practices and identified key threat vectors in 5G networks. The GSMA's Mobile Economy reports and ENISA's (European Union Agency for Cybersecurity) threat landscape reports were particularly useful for understanding the broader industry context and regulatory perspectives on 5G security.

To supplement the theoretical and industry-based insights, empirical data was collected using open-source tools, enabling practical analysis and validation of threat models. This approach was critical in bridging the gap between theoretical models and real-world scenarios. Tools such as

OpenVAS [2] and Nmap [3] were used to gather data on network traffic, identify potential vulnerabilities, and simulate attack scenarios in 5G environments. These tools allowed for hands-on exploration of threat dynamics and the effectiveness of various security measures. The use of these open-source tools facilitated a deeper understanding of the practical challenges in securing 5G networks and provided empirical support for the proposed threat models.

For data analysis, Python-based libraries were used to process and visualize data collected during simulations. This allowed for the identification of patterns of potential threats. In addition, Shodan[3] and OpenAI APIs[4] were used during data collection and data analysis processes. The details related to this usage are described in chapters: Automating threat modeling process and Automated vulnerability detection.

Figures, tables, and code snippets used in this dissertation are, in the vast majority, unique to this work. If not stated otherwise in the corresponding caption, the source of the figure or table is the authors own work.

[2] Open Vulnerability Assessment System is an open-source framework for performing vulnerability scanning and management. OpenVAS is designed to help identify security weaknesses in systems, networks, and applications.
[3] Network Mapper (Nmap) is a versatile open-source network scanning tool used primarily for network discovery and security auditing.

2. 5G Network

The ways in which we communicate and engage with the outside world could be completely transformed by new developments in wireless technologies and intelligent communication trends. These technologies, which include IoT, 5G, and artificial intelligence (AI), have the potential to significantly advance industries like entertainment, transportation, and medical. The cumulative trend of AI and 5G, gathered worldwide by Google Trends between 2004 and June 2024, is displayed in Figure 2. The way these new trends are created, developed, and applied will ultimately determine their impact, thus it will be critical to make sure they are applied in a way that advances society as a whole.

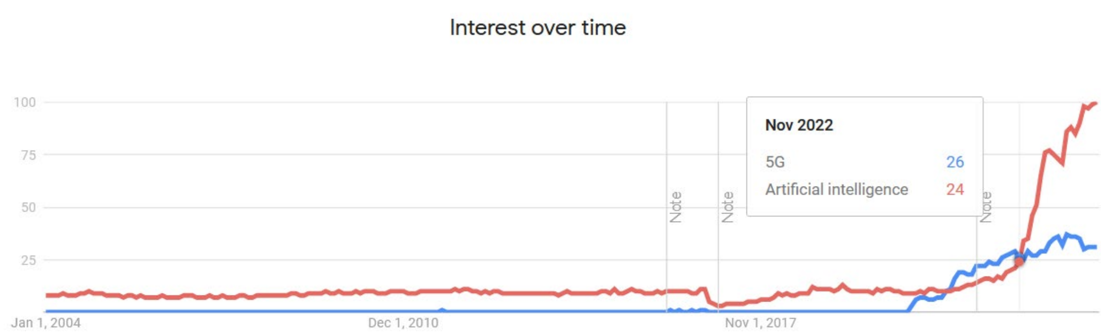

Figure 2: Worldwide interest trend for search topics between 2004 and 2024

Source: Google

The volume of data created is increasing dramatically due to the need for internet connectivity as well as the introduction of new technologies like automation, the Internet of Things (IoT), and artificial intelligence. Over the next ten years, the amount of data created will rise by several hundred zettabytes due to the exponential growth of data creation [7]. It is necessary to upgrade the existing mobile infrastructure because it was not built to handle such a large information load.

Currently, 5G's fast speed, enormous capacity, and low latency support and expand a number of applications, including video chatting, drone deliveries, cloud-connected traffic control, and mobile console gaming. The advantages and uses of 5G are infinite, ranging from international payments and emergency response to remote learning and a mobile workforce. It has the power to drastically alter people's lives, the global economy, and the workplace. Those lifestyle changes are already ongoing for several years and are commonly known as “Industry 4.0”. The terms "Industry 4.0" and "fourth industrial revolution" are interchangeable [8]and refer to a new phase in the management and structure of the industrial value chain connecting people to other people, to things, and interconnects things themselves through the use of contemporary control systems, embedded software, and Internet addresses, a concept known as the Internet of Everything (I2X). In the above-mentioned context, a thing is any item, such as a device, that may communicate with other devices via a network and establish a network.

5G can offer not only greater data transfer speeds but also reduced latency, or the amount of time it takes for a signal to go from a server to a device and back, thanks to the core network's flexibility. Because of low latency, a surgeon can perform remote surgery without worrying about a patient being harmed by a signal delay by using the 5G network to direct a surgical robot [9]. It is, in fact, essential for all types of networked robots. The "core network" enables all of 5G's essential features. This is the network's brain, providing all the functionality and architecture needed to carry out its duties. The ability for 5G's basic operations to be decentralized and deployed, for instance, in a datacentre close to the end user, is a considerable change from 4G. Businesses are now able to utilize mobile networks in ways that were not possible for them to do with earlier generations. 5G's capacity to support millions of concurrent connections—roughly one million connected devices per square kilometer [10]which is another crucial feature that will power the Internet of Things, enabling the networks connection of every traffic light, lamppost, and drain cover in a smart city. In 5G network, real-time data can be accessed by the business through the connectivity of everything, including construction materials and factory machinery. 5G-based autonomous vehicles, are now able to automatically deliver extra manufacturing components if they are being used up faster than expected. Flexible production systems facilitate human contact, and as robots become more and more integrated into businesses, with people continuing to work alongside them. This is why reliability and ultra-low latency wireless communications are necessary for machine coordination across the production line [11].

During recent years, we have seen a number of advancements in 5G technology research. Among the most noteworthy are:

- Integration with current infrastructure. To enable a more smooth and effective user experience, considerable research has been done on the integration of 5G networks with current infrastructure, such as 4G networks and Wi-Fi. As a result, new technologies that enable the effective use of several networks in a single device, like dual-connectivity and network slicing, have been developed [12].
- Standardization. Research into the creation of new and update of existing 5G standards is still ongoing, with an emphasis on features like low latency, fast connectivity, and support for novel use cases like industrial IoT and driverless cars.
- 5G for Industrial IoT. A lot of research has been done on the application of 5G technology for industrial IoT, with particular emphasis on real-time control, predictive maintenance, and smart manufacturing.
- 5G for autonomous vehicles. A lot of research has been done on the application of 5G technology for these kinds of vehicles, with particular emphasis on real-time map-pinging, vehicle-to-vehicle communications, and high-definition video streaming.
- Security and privacy. As 5G networks are used more often, worries over their potential effects on security and privacy have grown. The creation of more private and secure 5G networks, including the application of encryption, secure communications protocols, and secure device management, has thus been the subject of extensive research.

The creation of more secure and efficient networks is one of the most important areas of research for 5G technology, since these networks are anticipated to handle a lot of devices and applications. Research on the implementation and integration of 5G networks with current infrastructure, such as 4G networks and Wi-Fi, is also necessary. Creating new use cases for 5G technology and enhancing the performance and dependability of 5G networks in difficult situations, such high-density metropolitan areas or rural areas, are two further research directions [13].

2.1. 5G Features

The goal of 5G technology research and development is to achieve advanced features like massive machine-type communications, device-to-device (D2D) communications [14], and increased density of mobile broadband users to improve coverage. It also aims to support more users at faster speeds than 4G. In order to better facilitate the deployment of the Internet of Things (IoT), 5G planning also seeks to deliver greater network performance at reduced latency and lower energy consumption.

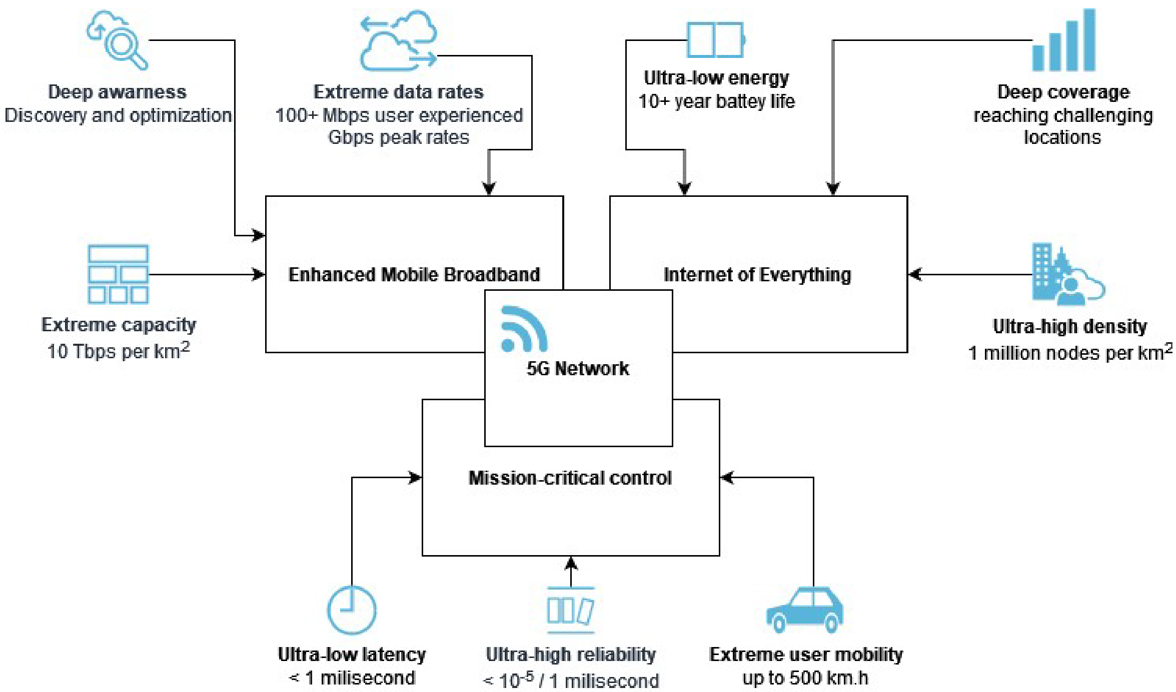

Figure 3: 5G target domains and features

More precisely, according to Warren and Dewar [15], 5G wireless systems have the following eight enhanced features:

- Data rate: 1–10Gbps connections to endpoints in the fiel.
- Low latency: 1-ms latency.
- Bandwidth: 1000× bandwidth per unit area.
- Connectivity: 10–100× number of connected devices.

- Availability: 99.999% availability.
- Coverage: 100% coverage.
- Network energy efficiency: 90% reduction of network energy usage.
- Device energy efficiency: Up to 10years of battery life for low-power devices.

Many technologies, including heterogeneous networks (HetNet), massive multiple-input multiple-output (MIMO), millimeter wave, D2D communications, software-defined networks (SDN), network functions virtualization (NFV), and networking slicing, are applied to 5G systems in order to achieve these eight advanced network performance features [16]. Not all technologies and components that make up a fifth-generation mobile network are completely new.

Table 1: Wireless technologies generations.

Source [17], [18]

Generation	**5G**	**4G / LTE**	**3G**	**2G**	**GMS**
Designed	2010	2000	1990	1980	1970
Deployed	2019	2009	2001	1991	1980
Throughput	1 Gbps	200 Mbps	2 Mbps	14.4 kbps	1.9 kbps
Core	Flatter internet protocol, 5G-Network interfacing	IP backbone	Packet network	*Public switched telephone network (PSTN)*	Public switched telephone network (PSTN)
Key technologies in radio	MIMO, mm-waves	Long-term evolution	Wideband-CDMA, CDMA-2000	*Global system for mobile communication (GSM), CDMA*	Advance mobile phone system

5G radio technology undoubtedly offers some major advancements over earlier generations of networks, however these innovations are all firmly based in technologies created for earlier network generations.

2.1. 5G Air Interface

Not all of the components that make up a fifth-generation mobile network are completely new. 5G radio technology undoubtedly offers some major advancements over earlier generations of networks, however these innovations are all firmly based in technologies created for earlier network generations. After all, by providing a strong foundation for the long-term evolution of mobile radio networks, LTE lived up to its ambitious moniker. This chapter describes the most important radio technology advances present in 5G networks.

5G New Radio (5G NR), or NR for short, is the name of the radio access technology used by 5G networks. While sharing a foundation with Orthogonal Frequency Division Multiple Access (OFDMA) technology [19], it differs from LTE radio access in a few significant ways. In Orthogonal Frequency Division Multiplexing (OFDM), NR allows for different spacings with wider subcarriers, in contrast to LTE, which has a fixed subcarrier spacing. NR supports OFDM subcarrier widths of 30, 60, 120, and 240 kHz in addition to 15 kHz. In addition to adding more frequency bands and broader channels, 5G NR's air interface improvements also include sophisticated capabilities like multi-radio connection, beamforming, and Coordinated Multi-Point (CoMP) that improve subscriber reliability and data speeds. In order to achieve bigger cell capacity and targeted communication with mobile devices, 5G base stations employ sophisticated antenna systems that make installation easier and provide even more layers for multi-user MIMO transmissions.

2.1.1. Beamforming

Antennas have changed dramatically in the last century; the sophisticated antennas of today don't resemble the basic wire structures of the past. The growing demands for radio communications efficiency, speed, and dependability have led to several inventions and creative solutions in this field [20]. Modern antennas are incredibly complex hardware components. In order to minimize energy losses and long runs of RF cables, antennas and radio heads, also known as RRUs, are located close to one another atop radio towers in almost all installations nowadays. However, numerous RF jumper wires are still used to link RRUs and antennas. These external cables are difficult to maintain, introduce losses, and are vulnerable to environmental deterioration, especially given that modern systems employ a higher number of MIMO layers. The current definition of an active antenna was established by removing these jumpers and installing RRUs inside the antenna enclosure. By using active antennas, installation may be done more easily, more cleanly, and with less wear and tear because fewer cables are required.

In order to create a range of antennas for various implementation scenarios, including antennas that vary their radiation patterns dynamically, radio engineers and antenna designers strive for the best balance in antenna radiation patterns. One method for modifying the vertical angle of an antenna radiation pattern—or, to put it more properly, controlling the antenna's gain in the vertical direction—is the electric tilt. Because it uses analog phase shifters to aim the beam downward, it is also occasionally referred to as analog beamforming. While using analog phase shifters rather than mechanical tilting may seem like a complicated solution for a straightforward problem, the phase shifters are actually rather affordable, basic components that are often driven by a stepped DC motor located inside the antenna enclosure. Usually, a unique tilt control circuit on the antenna is used by the BBU to regulate the tilt amount. Without physically touching the antenna, analog beamforming enables extremely accurate post-installation tilting of the antenna emission pattern. In 5G greater sophistication in three-dimensional beamforming is used [21].

The idea behind 3D beamforming (Figure 4 is similar to that of employing many elements to generate a beam, except this time, the elements are arranged in two dimensions, or flat arrays of antenna elements. A narrow beam can be produced in both the vertical and horizontal directions with the proper spacing between the antenna pieces. The beam produced by such a planar array of antenna elements can be guided both vertically and horizontally, illuminating only a portion of the sector inhabited by one or more target mobile devices. This is achieved by applying varying phase shifts to the antenna elements. 3D beamforming provides more independent transmissions utilizing the same frequency by allowing steering in both vertical and horizontal directions and regulating the antenna gain. This results in better data rates and cell capacities.

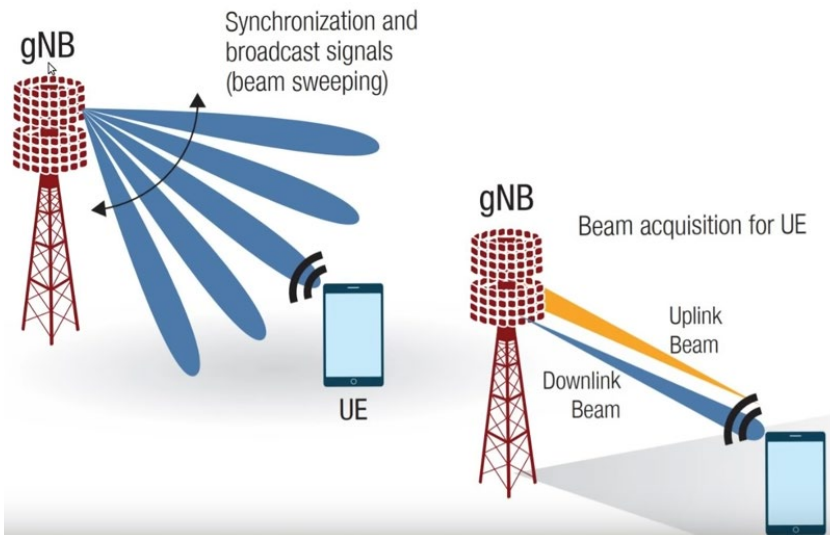

Figure 4: Beamforming in 5G.

Source: Analog Vs. Digital Vs Hybrid Beamforming for 5G.

Planar antennas with many components arranged in rows and columns are typical beamformer configurations. The optimal situation for 3D beamforming is to have each individual antenna element linked to a separate RF chain. With the help of these RF chains, every component can function as a separate antenna and send the same signal at various phase shifts. Each antenna element's phase shift is digitally controlled and applied in the appropriate RF chain. This is one of the differences between analog beamforming, which applies phase shifts using analog circuitry , while using a single RF chain to feed numerous antenna elements. The "digital beamforming" allows beams to follow specific subscribers and provides extremely accurate beam breadth and direction.

2.1.2. Massive Multiple Input Multiple Output (mMIMO)

The multiple input multiple output (MIMO) concept, which allows for several simultaneous data transmissions utilizing the same carrier frequency, is fundamental to 5G radio technology. In prior generation mobile networks, MIMO technology played a key role in increasing spectral efficiency, peak data rates for individual subscribers, and cell capacity overall. In LTE systems, 4x4 and 8x8 MIMO systems are frequently employed; however, it is not easy to increase the order of MIMO beyond these numbers. The route towards densification of spatial multiplexing encounters obstacles due to the restricted size of handheld mobile devices. More specifically, the number of antennas that can fit into a single mobile device is the limiting constraint. Antennas cannot be made indefinitely smaller, and the carrier frequency dictates the arbitrary separation between numerous antennas. In the case of single-user MIMO, this restricts the number of parallel transmissions. However, utilizing the MIMO principles, it is possible to multiplex data streams belonging to various subscribers using the same frequency when a cell serves a large number of mobile devices. This technique, known as multi-user MIMO (MU-MIMO), aids in greatly boosting cell capacity [22] and enhancing data rates for numerous customers at once.

2.1.3. Dynamic Spectrum Sharing

Making the switch to a newer radio access technology is never an easy or straightforward process. The steady deployment of 5G technology is made possible by its ability to interface with EPC (evolved packet core) of the existing 4G infrastructure. However, this doesn't mean that the transition of spectrum component can happen overnight. Repurposing spectrum or moving bands and frequencies from older technology to modern radio access technologies, can be challenging, particularly in densely populated areas, where rollout of new base stations happens gradually. In a clear and concise manner, when coexisting with earlier radio access technology, the newer technology uses a different set of frequencies. Since LTE does not use high bands, it may be possible to construct 5G NR using these. However, repurposing low and mid frequencies necessitates significantly more preparation. Since the low- and mid-band frequencies offer the ideal balance between capacity and optimal propagation, they are usually extremely valuable resources. It is usually necessary to remove certain mid- and low-band channels from the current LTE allocations in order to give 5G radios their own set of channels. Radio engineers must carefully arrange their resources for this work because it is difficult to strike the best possible balance between 5G and 4G resources [23]. Repurposing spectrum statically might lead to an imbalance when resources allocated to another radio access technology are wasted in the lack of a significant subscriber base, and users from earlier generations might not receive enough RF resources. Such undesirable situations can be avoided with the use of dynamic spectrum sharing (DSS), which provides a variable distribution of radio resources.

With use of DSS, time and frequency resources are dynamically assigned to both systems based on user demand for LTE and 5G NR services. Dynamic spectrum management (DSM) refers to this kind of dynamic resource allocation to users of current systems. DSM is a feature for early 5G network deployments [24], and as such, it is beneficial in an early NR market for allocating necessary time and frequency resources in accordance with the traffic demand of NR user equipment (UE), improving spectrum utilization. From the perspective of standardization, DSS [25] has been covered in 3GPP Rel-15 and enhanced in further releases [26].

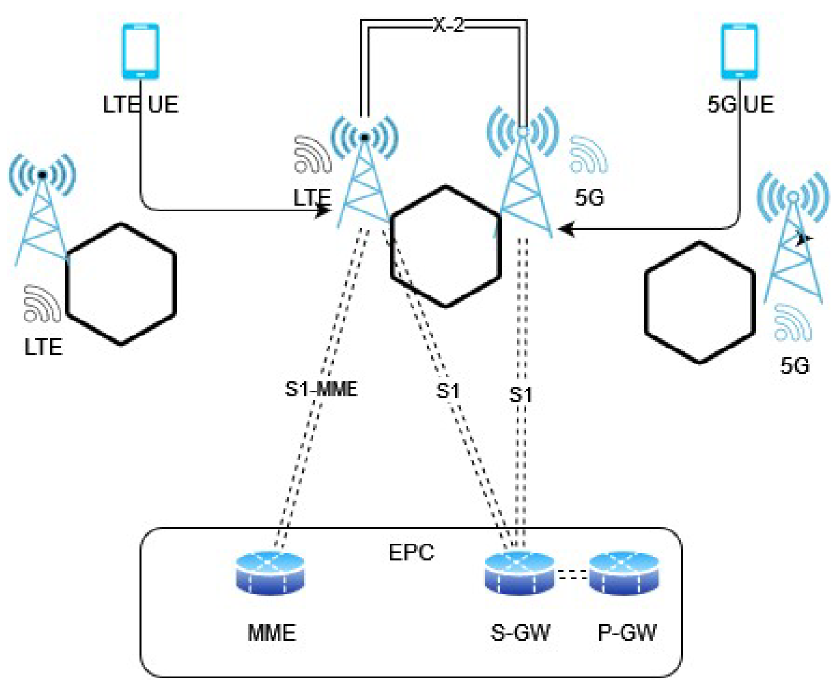

Figure 5: DSS in 5G and LTE.

Source: own

DSS is made to work with LTE UEs that are currently in use. Moreover, spectrum sharing makes sense to facilitate the 4G to 5G transition because the two technologies are anticipated to coexist for some time. Additionally, the 5G layer's design is comparable to that of the 4G layer, making DSS flexible with respect to LTE subcarrier spacing (SCS) and having a comparable time-domain structure. However, due of the overhead produced by both NR and LTE control channels, the main problems with DSS are the reduction of network capacity and the possible peak throughput by individual users [27]. The actual reduction in capacity varies depending on how the DSS is implemented and configured. Details of three specified DSS deployment options (Multicast-

Broadcast Single-Frequency Network-based, mini-slot non-MBSFN based, CRS rate matching non-MBSFN based) is out of scope of this work.

2.1.4. RAN Virtualization

The Radio Access Network (RAN) is a crucial component of the mobile network infrastructure since it is tasked with delivering wireless access to consumers through base stations (BSs). Conventionally, Base Stations (BSs) are composed of two primary elements: a Radio Unit (RU) and a Baseband Unit (BBU). The RU, or Radio Unit, is the component that handles the transmission and reception of radio frequency signals. It is responsible for the conversion of radio signals into electrical impulses and vice versa. Conversely, the BBU had the responsibility of managing the RU and handling the radio signals for the transport network. The 5G architecture divides the BBU into two units: DU (*Distributed Unit*) and CU (*Central Unit*). This gives 5G NR more flexibility in terms of infrastructure, as the DU and CU can be distributed according to the requirements of the network architecture into different hardware and/or software units [28].

Virtualization for communications service providers began with the core network and subsequently cloud technologies have been evolving at a rapid rate. Software defined networking (SDN) and network function virtualization (NFV) enabled the launch of network slices to meet the requirements of various 5G use cases in a more cost efficient way [29]. Virtualization of Radio Access Network followed. It is based on the subdivision of the RAN into upper and lower sections, which was standardized in 3GPP R15. In this standard, a higher-layer split was defined, with a clearly defined interface (F1) between two logical units: the Centralized Unit and the Distributed Unit. The Central Unit which has fewer strict processing needs, has proved more suitable for virtualization compared to the Distributed Unit and its functions that are in closer proximity to the radio. In the context of full-stack RAN virtualization, the Distributed Unit (DU) is linked to the radio using a packet interface called enhanced Common Public Radio Interface (eCPRI). There are several methods to allocate functions between the DU and the radio, which are commonly known as lower-layer split (LLS) options. An alternative proposed by the O-RAN Alliance known as the 7-2x split and other functional splits are also under consideration [30], but are not scope of this dissertation.

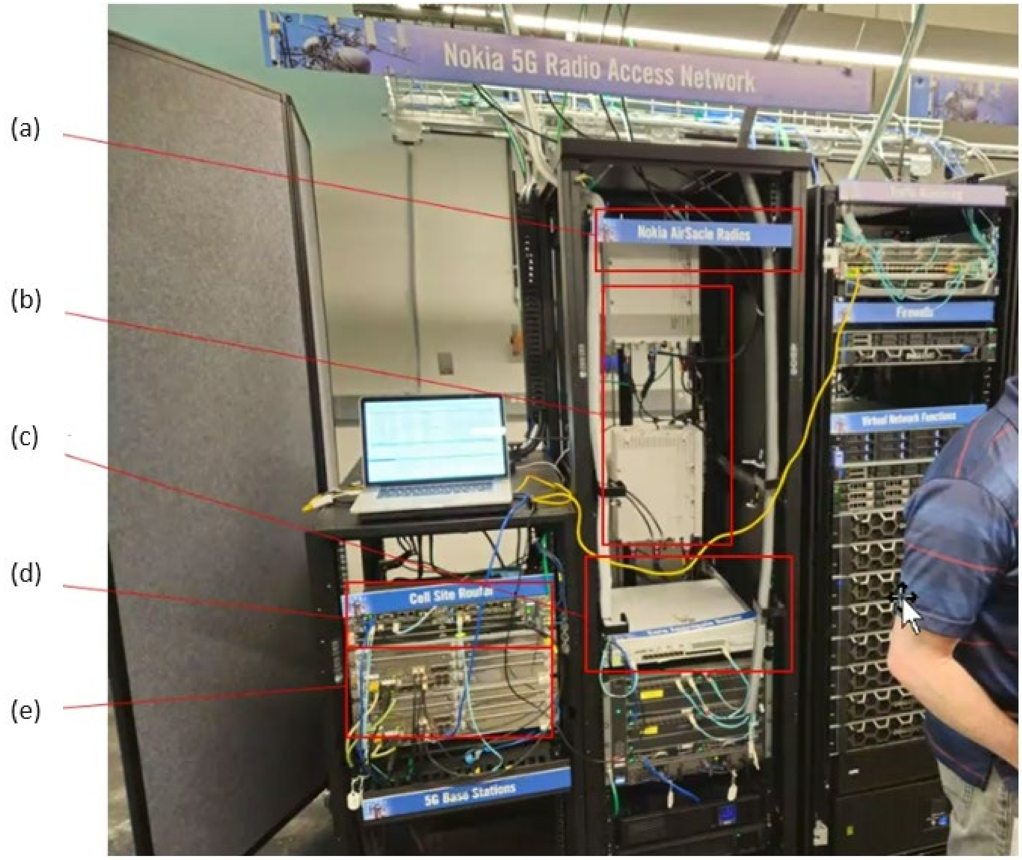

Figure 6: 5G RAN.

(a) Nokia AirScale Solution. (b) RU. (c) Edge router. (d) Service aggregation router. (e) DU (ASIK+ABIL). Source: Nokia

The primary objective of vRAN is to transition away from the conventional hardware-centric approach of RAN, and instead adopt virtualized or software-based functionalities. It separates software and hardware into distinct components. By virtualizing the RAN, operators have the ability to execute their 5G stacks on any vendor infrastructure. Due to its exceptional flexibility, adaptability, and interoperability, it allows for the implementation of complex and valuable next-generation use cases. The virtualized RAN relocates the control functions of hardware base stations to centralized servers, therefore reducing the distance between them and the network edge (as presented in Figure 6). By implementing this approach, operators have the ability to optimize their resources in order to adapt to the always fluctuating network traffic [31].

2.2. Core network

All mobile control, voice and data are managed by the mobile exchange and core data network. The core network for 5G was redesigned for better integration with dispersed servers available throughout the network, which speeds up reaction times (lower latency) and improves connectivity with cloud-based appliances. Many of the advanced features of 5G, including network function virtualization and network slicing for a range of services and applications, are managed by the core. Although changes to the core architecture were introduced the 5G core is an evolution and not a revolution allowing network operators to evolve their service to meet market demands when they need to. With the ability to create dedicated networks with specific service quality characteristics through network slicing and the ability to process application traffic at the edge of a network to create ultra-low latency experiences through Multi-access Edge Computing (MEC), the 5G architecture lays the groundwork for new technical capabilities. The 5G core system architecture is specified as a cloud-based system that leverages a service-based architecture (SBA) to provide increased scalability and flexibility [32].

2.2.1. Multi-Access Edge Compute

Multi-access Edge Computing (MEC) solution was designed as an add-on to a 4G network to create a smooth transition between 4G and 5G services. Although it was existing in LTE in fact it is a key component of the 5G network infrastructure that brings computing power and resources closer to end-users [33]. MEC plays several crucial roles, including reducing latency by processing data closer to the user, improving bandwidth by offloading the core network, and supporting new services that require low latency and high bandwidth. This is especially useful for applications that require immediate response times, such as remote surgery, autonomous vehicles, augmented and virtual reality [34].

The main feature of MEC is to manage tasks more efficiently and faster, thanks to higher proximity between network elements. It also optimizes costs by processing data locally and enhances security by processing sensitive data closer to the source. Typically, the MEC platform is set up on a dedicated server within a cloud architecture, which also allows for the deployment of virtual machines or containers for applications [35]. MEC is based on a virtualized platform, hosted by the infrastructure placed at the edge of the mobile network. It can be collocated within the Radio Access Network [36], with network aggregation point or located together with Core Network functions as presented in Figure 7.

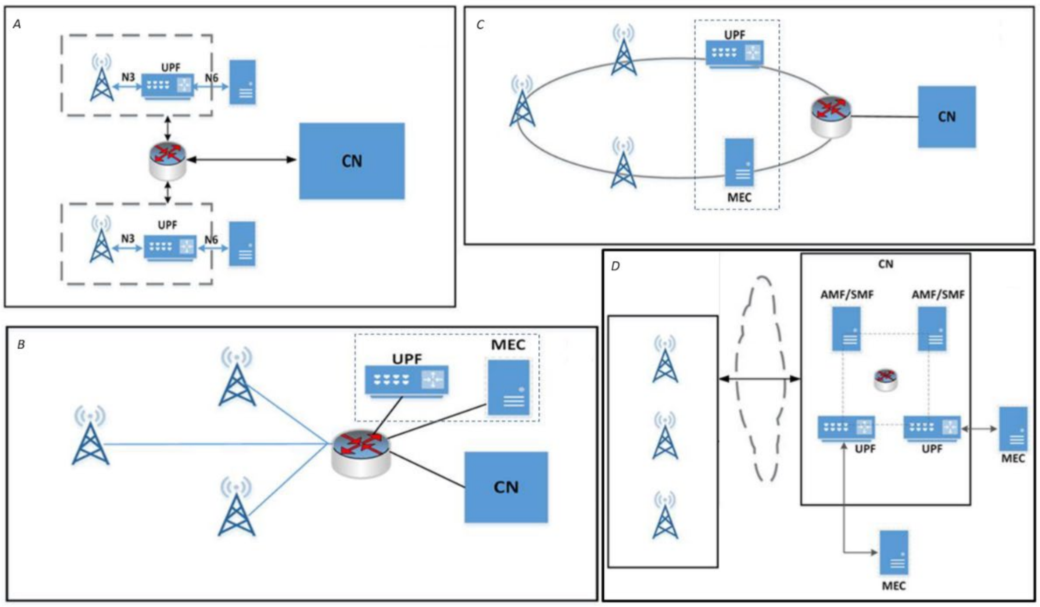

Figure 7: MEC physical deployment options.

A) MEC in local UPF at gNB, B) MEC at transmission node. C) MEC at network aggregation point with locl UPF. D) MEC within the Core Network (at the same data center).
Source: ETSI. MEC in 5G networks

Regardless of the deployment location the MEC platform is usually set up on a dedicated server. In most cases within a cloud architecture, which also allows for the deployment of virtual machines or containers for applications. When examining MEC in greater detail, general entities fall into three categories: host, system, and network-level entities.

- MEC host is an entity that contains a MEC platform and a virtualization infrastructure which provides compute, storage, and network resources, to run MEC applications.

- MEC platform is the set of fundamental features needed to execute MEC apps on a certain virtualization infrastructure and allow them to supply and receive MEC services (the MEC platform itself is capable of providing services).

- MEC applications are instantiated on the virtualization infrastructure of the MEC host based on configuration or requests validated by the MEC management.

The introduction of Multi-access Edge Computing (MEC) in 5G networks brings significant advantages, but also introduces new security risks and threats. The distributed nature of MEC, with its numerous edge nodes, expands the attack surface, making it more challenging to secure. Additionally, the reliance on third-party applications and services in the MEC ecosystem increases the potential for supply chain attacks [36] where malicious actors compromise these components to gain unauthorized access to the network or data. The dynamic nature of MEC, with its rapidly changing workloads and resource allocations, makes it difficult to implement consistent security policies and controls. This can lead to misconfigurations and gaps in security that can be exploited by attackers.

2.2.2. Cloud native 5G Core

Building and executing programs that fully utilize the benefits of the cloud computing delivery mechanism is known as cloud native development. This is a novel approach to the development, deployment, and maintenance of applications, it does not refer to the actual location of the program, which might be hosted in an edge, core, or data center cloud (either private or public). The term cloud native describes the division of applications into smaller components called microservices. In this development mode, applications are divided into microservices which may come with the application logic or routing logic needed to make the application function. When these are dismantled, it offers the flexibility and agility to scale each part separately from the others, adding an additional degree of security.

A container orchestration system, like Docker or Kubernetes, represents a new type of application architecture beyond what virtualization has historically delivered and lays the groundwork for cloud-native designs. Each virtual machine (VM) created with hardware-based virtualization has a guest operating system that offers application isolation from the host. On the other hand, because they share an OS and host kernel, containers demand less system resources. Containers can be installed on virtual machines (VMs), which is prevalent in public cloud applications, and VMs offer an additional degree of security and isolation. From threat modeling perspective the cloud nature of 5G core network is an important feature, because unlike legacy systems, a cloud-native 5G core dynamically adapts to fluctuating demands by spinning up or down instances of microservices as needed. This flexibility also extends to service deployment and customization, giving network operators opportunities to tailor their offerings to specific industries or use cases. This opens up a world of possibilities for new services and applications. This modular design allows for easier integration and faster deployment of novel services. However, it is important to note that this architectural shift also introduces unique security challenges. The distributed nature of microservices expands the attack surface, requiring vigilant attention to API security, container vulnerabilities, and potential misconfigurations in orchestration tools, chapter 2.4 describes this problem in more detail.

2.2.3. User Plane Network Functions and CUPS

The User Plane Network Functions (UPFs) and the Control and User Plane Separation (CUPS) architecture have revolutionized the way 5G data traffic is handled compared to legacy mobile networks. In previous generations, the control plane and user plane were tightly integrated[37], leading to potential service disruptions when changes or upgrades were made. It changed in LTE (3GPP Rel. 14) when CUPS concept was introduced. CUPS architecture has decoupled user and control planes, enabling the control plane to focus on signaling and decision-making while the user plane, spearheaded by the UPF, concentrates on data forwarding. This separation brings about several key advantages. Firstly, it allows for independent scaling of both planes. The control plane can adapt to increased signaling demands without affecting the user plane's data processing capacity. Secondly, UPFs can be strategically deployed closer to the network edge or even within the radio access network, minimizing latency for data-intensive applications. Thirdly, service chaining becomes possible, enabling flexible traffic steering and the implementation of advanced network services such as traffic optimization, security functions, and content caching.

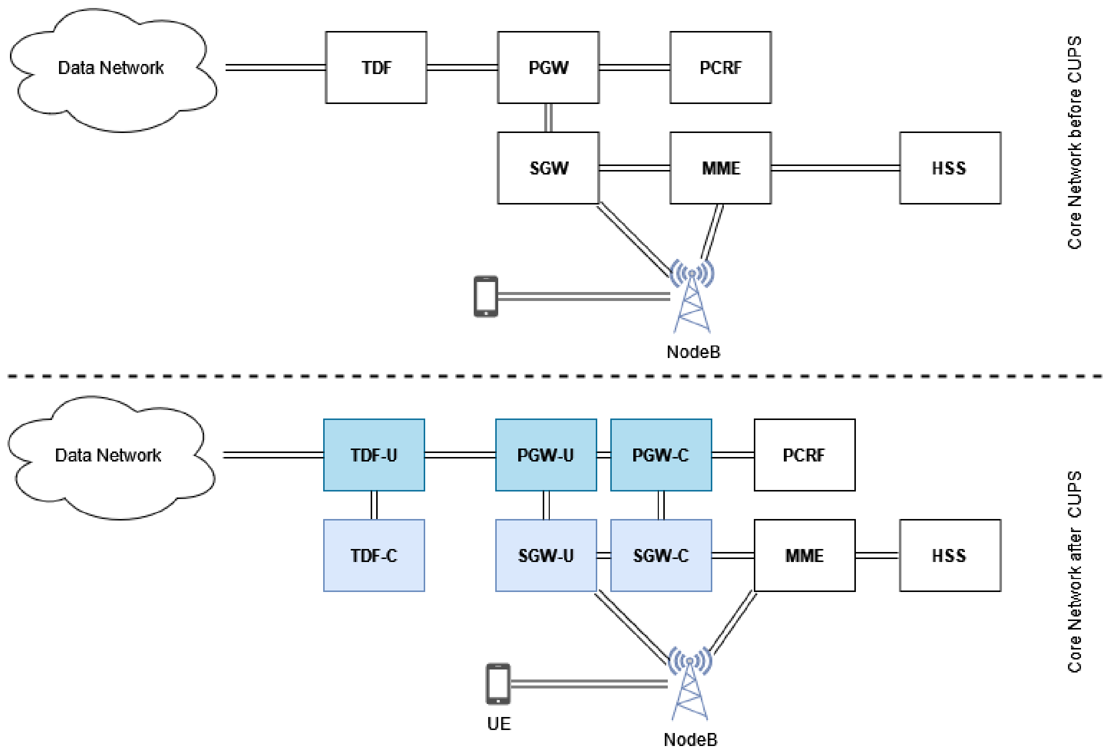

Figure 8: Control and User Plane Separation.

Source: own.

Presented in Figure 8 is the architectural difference between core network before and after the introduction of 3GPP release 14 changes (CUPS). The complete architecture of 5G network is described further in this work, but to understand the basic differences key elements need to be described. TFD/TDF-C is the network function responsible for making policy decisions in 5G TFD-C does it in the control plane and TDF-U is enforcing them in the user plane. Meanwhile, the Packet Data Network Gateway (PGW), split into PGW-C and PGW-U, takes charge of IP address allocation, session management, and policy enforcement, while also forwarding user data packets with QoS enforcement. The Serving Gateway (SGW) plays a pivotal role in mobility management, session establishment, bearer resource allocation, and data packet routing between the radio access network (RAN) and PGW-U, all through its SGW-C and SGW-U components. The PCRF, remaining solely in the control plane, dictates policy and charging rules to the TDF-C and PGW-C based on user subscriptions, network conditions, and operator-defined policies. The Mobility Management Entity (MME) of 4G evolves into the Access and Mobility Management Function (AMF) in 5G, bearing the responsibility for mobility management, authentication, security, and initial context setup for user sessions. Finally, the Home Subscriber Server (HSS) remains a cornerstone of the control plane, storing vital subscriber information and authentication credentials used by the AMF.

CUPS architecture is a fundamental enabler of network slicing, allowing a single physical network to be divided into multiple virtual networks with tailored traffic policies and quality of service. The separation of control and user planes enhances security by reducing the attack surface and protecting user data even if the control plane is compromised. On the other hand CUPS allows for flexible deployment of user plane functions closer to the edge, potentially exposing them to physical or logical attacks. In contrast to legacy networks where scaling was limited and UPF deployment was centralized, 5G with CUPS offers independent scaling, flexible UPF placement, and the possibility of service chaining and network slicing, marking a significant advancement in mobile network architecture and capabilities.

2.3. 5G Service Based Architecture

This chapter briefly describes 5G network architecture along with the key SBA concept and most important network functions. The description of system architecture is a prerequisite for effective threat modeling. For readers without knowledge about 5G networks It provides the necessary foundation by outlining the system's components, interactions and data flow. This understanding allows for accurate assessment of potential threats and vulnerabilities, defining the scope of analysis, and identifying critical assets.

2.3.1. SBA concept

The 5GC architecture is based on a framework called Service-Based Architecture (SBA), which defines the architecture parts in terms of Network Functions (NFs) instead of traditional Network Entities. Service-Based Architecture is a software design approach that structures an application as a collection of loosely coupled services. It is also known as Service-Oriented Architecture (SOA) - a distributed architectural paradigm that involves organizing an application into a set of services that are loosely connected and may be reused. Each service is a self-contained unit responsible for a specific business function and communicates with other services through well-defined interfaces, often using standard protocols like HTTP or messaging systems such as SOAP, REST [38] or gRPC [39]. The benefits of SBA, such as modularity, flexibility, scalability, and reusability, are relevant to various network domains, including telecommunications, enterprise networks, cloud infrastructures, and the Internet of Things (IoT). SBA enables the development of network solutions that can adapt to changing requirements, accommodate new technologies, and support diverse applications.

In network systems, SBA is highly applicable as it enables the modularization of network functionalities, allowing for independent development, deployment, and scaling of different components.

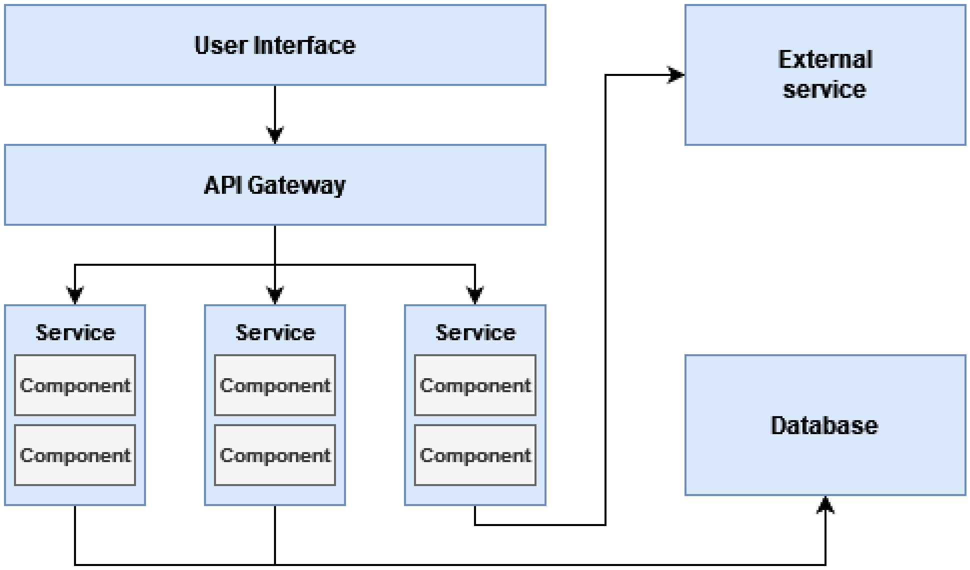

Figure 9: Service Based Architecture concept.

Source: Own.

In a telecommunication network, services like authentication, billing, and call routing can be implemented as separate services. This modularity enhances flexibility, simplifies maintenance, and promotes reusability of network components. Some common implementations of SBA in networking include the use of web services for network management and control, service-oriented architectures for service delivery platforms, and microservices for cloud-based network functions virtualization (NFV).

2.3.2. 5G architecture

Schematically, the 5G system uses the same elements as the previous generations: a User Equipment (UE), combined from a mobile device and a SIM (subscriber identity module), the Radio Access Network (NG-RAN) and the Core Network (5GC), as shown in Figure 5. The primary component of the Next Generation Radio Access Network is the gNB, which is an abbreviation for 5G NodeB. The term NodeB has been carried over from 3G and forward to denote the radio transmitter. The radio interface is denoted as NR-Uu due to similar considerations, albeit with variations: in this case, 5G is represented by NR (New Radio) and Uu is a name that has been carried over from earlier generations. The gNB can be divided into a gNB-Central Unit (gNB-CU) and one or more gNB-Distributed Units (gNB-DU), which are connected by the F1 interface.

Through the interfaces of a shared framework, a specific Network Function (NF) exposes its services to other approved NFs and/or to any consumers who are allowed to utilize these services. The usage of SBA concept guarantees the advantages of modularity and reusability. The 5G Mobile Core utilizes a microservice-like structure, which means that it follows a similar design to microservices. However, it is important to note that the 3GPP specification [40] only defines the functional blocks and does not provide a specific implementation. It is important to note that conceptualizing a group of functional blocks is distinct from the engineering choices involved in developing a system based on microservices.

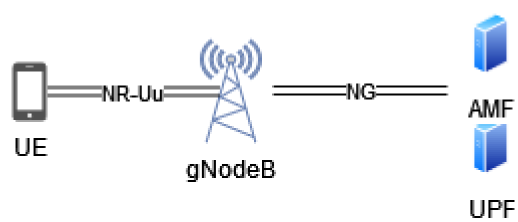

Figure 10: 5G basic overview.

Source: Own.

The 5GC is depicted above as the AMF and UPF entities. User Plane Function (UPF) is responsible for managing user data, and the Access and Mobility Management Function (AMF) is responsible for handling the signaling plane, connecting with the User Equipment (UE) and the Radio Access Network (RAN). Additional components of the 5th Generation Core Network (5GC) are outlined in chapter 2.3.3.

Due to the heterogeneous nature of the 5G cellular architecture, it is necessary to incorporate macrocells, microcells, tiny cells, and relays. The mobile small cell concept is a crucial component of the 5G wireless cellular network and includes both the mobile relay and small cell concepts [41]. It is being implemented to accommodate users with high mobility, such as those in cars and high-speed trains. Mobile small cells are strategically placed within moving vehicles to establish communication with the occupants, while the massive MIMO unit, comprising of enormous antenna arrays, is positioned outside the vehicle to establish communication with the external base station. Users of mobile small cells experience a significantly decreased signaling overhead and enjoy high data rates for data rate services, as demonstrated in [42]. As already discussed the 5G wireless cellular network design has two logical layers: a radio network and a network cloud. The radio network is composed of various components that serve distinct duties, which are mentioned in chapters 2.1.1 - 2.1.4. The NFV cloud comprises a User Plane Entity (UPE) and a Control plane entity (CPE) responsible for executing higher layer functions associated with the User and Control plane, respectively.

2.3.3. Interfaces and Network Functions

The set of functional network elements in 5G can be placed into three groups as mentioned in [43]. The group operating in Control Plane (CP) having their equivalents in the EPC consists of:

- AMF (Core Access and Mobility Management Function) responsible for overseeing the mobility-related components of the EPC's MME. Responsible for managing the connection and reachability of devices, handling mobility, ensuring access authentication and authorization, and providing location services.
- PCF (Policy Control Function) responsible for management of the policy rules that other CP functions then enforce. Roughly corresponds to the EPC’s PCRF.
- SMF (Session Management Function) responsible for managing each user session. This includes tasks such as allocating IP addresses, selecting the appropriate User Plane (UP) function, controlling aspects of Quality of Service , and managing UP routing. SMF corresponds to the EPC's Mobility Management Entity (MME) and the control-related elements of PGW.
- UDM (Unified Data Management) is responsible for overseeing user identity, which includes the creation of authentication credentials. Includes part of the functionality in the EPC’s HSS.
- AUSF (Authentication Server Function) is a server responsible for authentication. Encompasses a portion of the capabilities found in the EPC's HSS.

- EIR (Equipment Identifier Register): a means to ensure device security by maintaining a database of device identities, verifying their legitimacy, and blacklisting stolen or unauthorized devices to protect the network and support law enforcement.

Another group operating in the Control Plane (CP) but unique to 5G, ad not having a counterpart in the EPC is formed by:

- SDSF (Structured Data Storage Network Function): an auxiliary service employed for the purpose of storing structured data. An SQL Database could be utilized in a system built on microservices.
- UDSF (Unstructured Data Storage Network Function): a supplementary service utilized for the storage of unstructured data. Could be executed using a Key/Value Store within a system built on microservices.
- NEF (Network Exposure Function): a service designed for selectively revealing certain capabilities to other third-party services. This includes converting data between internal and external formats.
- NFR (NF Repository Function): a service which allow discovery and access to other available services.
- NSSF (Network Slicing Selector Function): a function designed to allow selection of specific network slice[4] to serve a particular User Equipment (UE).
- 5G-IWF (5G Interworking Function): a means to enable seamless communication and interoperability between the 5G core network (5GC) and legacy 4G/LTE networks (EPC). It translates protocols and adapts signaling messages, ensuring smooth transitions for devices and services between the two network generations.
- SEPP (Security Edge Protection Proxy): a means to provide security gateway functions, protecting the home network by filtering and validating signaling messages that are exchanged

[4] Network slicing is a function that enables the creation of numerous virtualized and independent networks on a shared physical infrastructure. Each segment or division of the network can be assigned according to the specific requirements of the application, use case, or customer.

between the home network and other external networks like roaming partners. SEPP helps to prevent unauthorized access and protect against external security threats.

NEF, NRF, and NSSF are vital elements of the 5G Core Network Architecture. They have unique but complimentary functions in facilitating dynamic service orchestration, resource management, and network slicing [44]. Collectively, they enable operators to provide a diverse array of cutting-edge services and applications, while also assuring optimal use of network resources and compliance with service-level agreements.

The last group, formed by a single component, relates to functions used in User Plane:

- UPF (User Plane Function): is responsible for routing traffic between the Radio Access Network (RAN) and the Internet. It performs the same function as the combination of the Serving Gateway (S-GW) and Packet Data Network Gateway (P-GW) in the Evolved Packet Core. Aside from packet forwarding, it is also responsible for policy enforcement, lawful intercept, traffic consumption reporting, and QoS monitoring.

The fact that 5G core network consists of so many elements make the creation of a comprehensive threat model for a 5G core network a challenge. This is not only because of the complexities of its architecture, but also because of dynamic specifics of the network. Unlike traditional network infrastructures, 5G is built upon a cloud-native foundation, which allows for the dynamic scaling and distribution of network functions across various virtualized environments. Virtual Network Functions in different software versions and from different vendors, including open source [45] solutions, can be dynamically added or removed from the network. This flexibility, while beneficial for operational efficiency and scalability, significantly complicates the task of identifying and mitigating potential security risks. Dynamic nature also introduces challenges in terms of visibility and control. With network functions spread across multiple virtualized environments, it becomes increasingly difficult to gain a holistic view of the entire system and its potential vulnerabilities. This lack of visibility can hinder the identification of potential attack paths and compromise the effectiveness of security measures. Furthermore, the dynamic nature of the cloud-native environment means that the threat landscape is constantly evolving, requiring continuous updates to the threat model to remain relevant. In this cloud-native environment, network functions are no longer confined to physical hardware but are instead instantiated as software components running on virtual machines or containers. These components can be rapidly deployed, replicated, and migrated across different nodes, making it difficult to track their precise location and interactions. Additionally, the sheer number of network functions involved in a 5G core network, each with its own set of vulnerabilities, further complexes the difficulty of threat modeling.

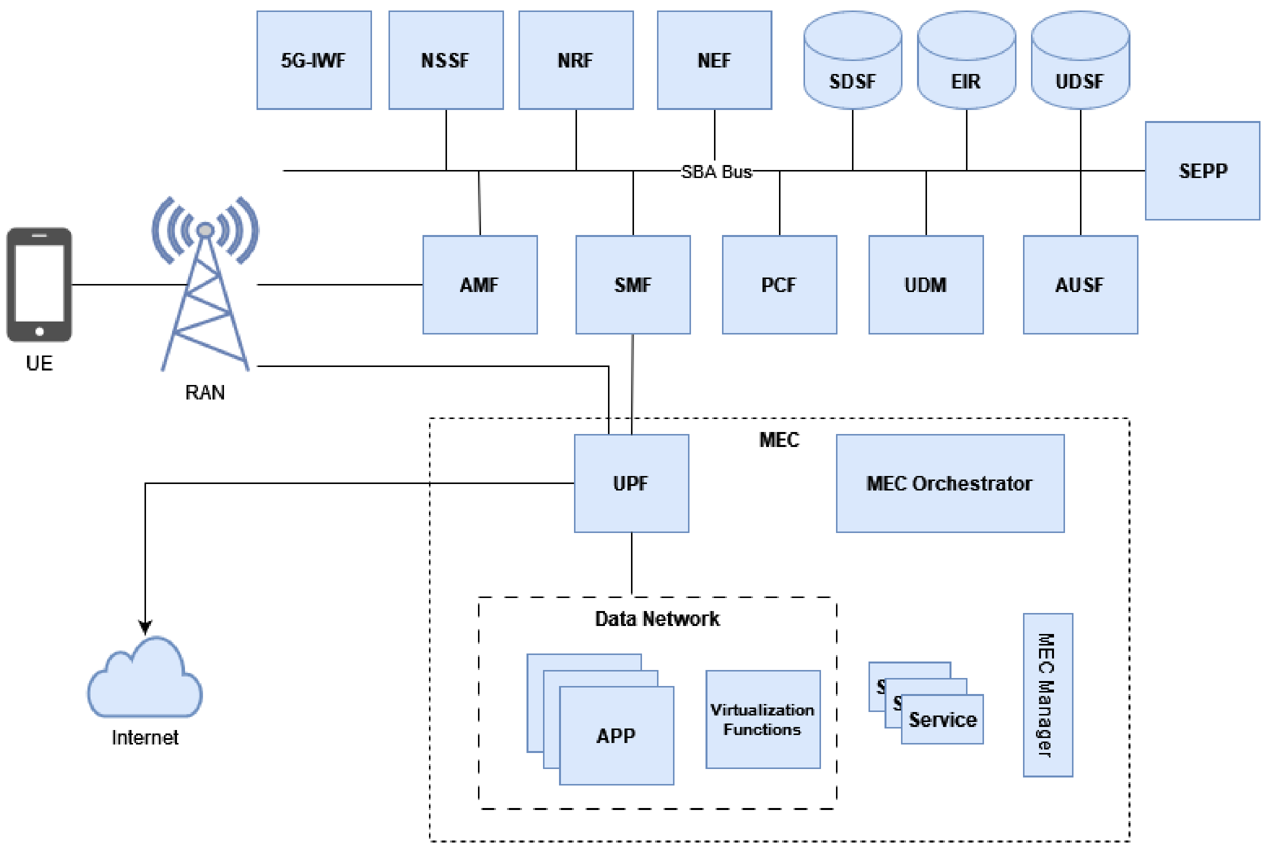

Figure 11: 5G Network basic architecture.

Source: Own

The multi-access edge cloud, described in chapter 2.2.1 adds another layer of complexity to the threat modeling process. Designed to host a wide range of third-party applications and services closer to end users [46], the MEC introduces a heterogeneous environment with diverse and often unknown security risks. The applications running on the MEC may have varying levels of security maturity and could potentially become entry points for attackers to infiltrate the network. Moreover, the proximity of the MEC to end users raises concerns about potential privacy and data protection issues. The MEC may process sensitive user data, making it a prime target for malicious actors seeking to exploit vulnerabilities. The dynamic nature of the MEC, with applications being constantly added, removed, or updated, further complicates the task of securing this environment and ensuring the confidentiality, integrity, and availability of user data.

2.4. Threats targeting 5G networks

To lay foundations for a solution capable of delivering threat identification and threat modeling functionalities in an automated manner, a manual attempt has been conducted. This chapter cover threat identification performed on a 5g network using a combination of STRIDE [47] and Bhadra [48] methodologies. Other threat modeling methods are also analyzed in this dissertation and are described in chapter 3.2.

2.4.1. Critical elements in 5G core

5G networks utilize a more decentralized and virtualized infrastructure than their predecessors, incorporating numerous interconnected devices and software components. This increased complexity can create potential vulnerabilities that malicious actors may exploit, necessitating robust security measures at each point of the network. For example, the Access and Mobility Management Function, a core component of the 5G core network, is responsible for authentication, authorization, and mobility management procedures. If compromised, attackers could gain unauthorized access to subscriber data, track users' locations, or disrupt network services through denial-of-service attacks. The distributed nature of 5G networks, with components like AMF spread across multiple locations, further complicates the security challenge, requiring robust protection mechanisms at each instance to mitigate the risk of cascading failures.

Another critical element lies in the software-defined nature of 5G networks. While this approach offers flexibility and scalability, it also introduces risks if the underlying software is not adequately secured. Software vulnerabilities in the 5G core network functions, like the Session Management Function or the Policy Control Function, could be exploited to intercept and manipulate user traffic, leading to data breaches and privacy violations. Applications residing in Multi-access Edge Computing (MEC) environments are also potentially vulnerable to code injection attacks, where malicious code is inserted into the application's execution flow, compromising the integrity of data and services.

The massive data volume generated and transmitted across 5G networks poses a significant cybersecurity concern. 5G already handles unprecedented amounts of sensitive personal and operational data, making it a prime target for cybercriminals seeking valuable information. For instance, the User Plane Function (UPF), responsible for routing user traffic, is a potential target for man-in-the-middle attacks, where attackers intercept and eavesdrop on communications. Ensuring the confidentiality, integrity, and availability of this data through robust encryption, authentication, and access control mechanisms at each stage of the data flow, from the device to the core network and the application layer, is paramount. Implementing strong data protection measures like end-to-end encryption and anonymization techniques can help mitigate the risk of unauthorized access and data leakage.

The integration of various technologies, such as the Internet of Things (IoT) and edge computing, within the 5G ecosystem expands the potential attack vectors. These interconnected devices and systems often lack adequate security measures, making them vulnerable to exploitation. IoT devices connected to the 5G network could be compromised and used to launch distributed denial-of-service (DDoS) attacks, disrupting critical services and infrastructure. Similarly, edge computing nodes, which process and store data closer to the end-users, are susceptible to physical tampering and unauthorized access, leading to data breaches and operational disruptions. Securing these endpoints and ensuring secure communication between them through measures like device authentication, secure boot processes, and intrusion detection systems [49] is vital to safeguard the entire 5G network.

The increased reliance on cloud infrastructure for 5G services introduces new cybersecurity considerations. While cloud computing offers numerous benefits, it also raises concerns about data security, privacy, and regulatory compliance. The 5G core network functions deployed on cloud platforms are susceptible to misconfigurations and vulnerabilities in the cloud infrastructure itself, which could lead to unauthorized access and data exposure. Ensuring the implementation of proper security standards and adding necessary mechanisms safeguarding data during transmission and storage is crucial for maintaining the trust and integrity of 5G networks, automated threat modeling can help to detect related problems before they materialize in form of a cybersecurity incident.

2.4.2. Threat identification

Adapting and combining generic threat modeling methodology for a given area necessitates precise application. STRIDE is specifically designed for the analysis of software vulnerabilities and requires access to detailed documentation and source code. However, STRIDE is frequently utilized in the context of other distributed systems that rely solely on a high-level specification. As the target system develops and the attention turns from general threats to threats specific to a particular domain, there is a requirement for a domain-specific threat modeling framework that has its own classification system for threats, vulnerabilities, and attacks. Here Bhadra framework that encompasses a shared conceptual framework, taxonomy, and categorization for risks and assaults against mobile communications systems comes in handy.

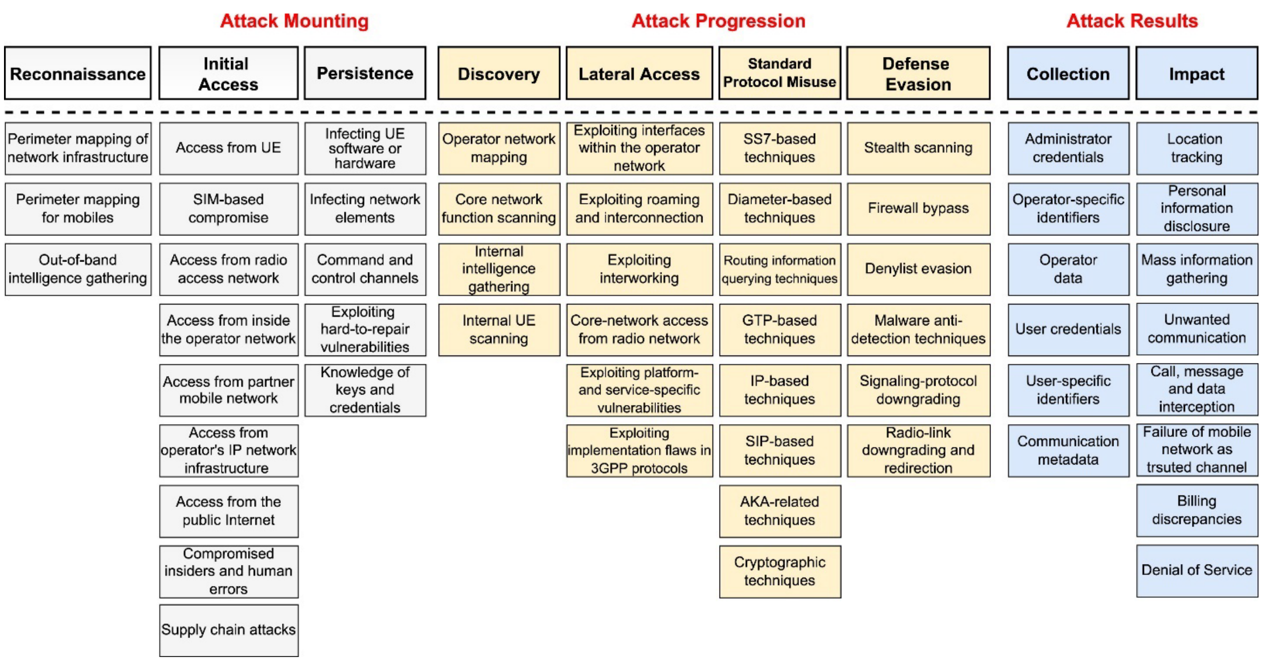

Figure 12: Bhadra threat modeling framework.

Source: Rao, et. al. "Threat modeling framework for mobile communication systems".

The framework offers a means for security experts to examine and convey incident and vulnerability information with ample context. During the process of investigation, it is beneficial for security analysts to have a comprehensive understanding of the technical aspects while also being able to deliver concise and practical information for their colleagues and management. The lexicon (Figure 12) provided by the framework is also valuable for the dissemination of threat intelligence among different enterprises. The framework was successfully used for quantitative threat identification and analyzing the security posture of a mobile operator's network in [50].

The primary objective of the Bhadra framework is to offer a cohesive conceptual framework for the analysis and communication of security risks that especially focus on or exploit the mobile operator infrastructure. Building on previous research in the field of enterprise IT, the framework specifically examines the actions of attackers at various points in the attack life cycle. The objective is to identify a level of abstraction that enables a coherent description of the assault without necessitating a comprehensive understanding of all technological intricacies. This facilitates the analysis of prevalent attack patterns and trends, as well as the exchange of information among organizations.

2.4.2.1. Threats in SBA

Software based architecture together with SDN promotes creativity in communication networks and streamlines network administration by allowing for programmability of the network. These two traits expose the network to security vulnerabilities. As an example, the SDN controller updates or alters flow rules in the data forwarding elements. The control information flow can be readily recognized, making it a conspicuous element in the network and hence a preferred target for DoS attacks. Moreover, the concentration of network control can potentially result in the controller being a bottleneck for the entire network during saturation attacks. Through the use of SBA concept, the majority of network functions can be executed as software applications. If unauthorized apps are given permission or if important Application Programming Interfaces (APIs) are made accessible to unwanted software, it can cause widespread chaos on the network. The existing SBA architecture necessitates that the data forwarding components retain traffic flow requests until the controller modifies the flow forwarding rules. Therefore, the data plane components are susceptible to saturation assaults due to the restricted resources of forwarding elements to handle unsolicited (TCP/UDP) flows [51]. Any misconfiguration of forwarding elements or conflicts between federated networks caused by many controllers might cause additional issues to the network security.

Table 2: Threats in SBA

Security Threat / Attack	Target Point/Network Element	Effected Area
Exploiting platform and service vulnerabilities	Centralized control elements	**SBA**
Internal intelligence gathering	Virtual resources	**SBA**
Hijacking attacks	Hypervisor	**SBA**
Saturation attacks	SDN controller and switches	**SBA**
Configuration attacks	SDN (virtual) switches, routers	**SBA**
Exploiting interworking	Virtual resources	**SBA**
IP based attacks	SDN controller-switch communication	**SBA**
Exploiting interfaces within operator network	SDN controller-communication	**SBA**

2.4.2.2. Threats in NFV

NFV faces fundamental security concerns related to confidentiality, integrity, authenticity, and non-repudiation. Regarding its application in mobile networks, the NFV platform implementation can potentially lack adequate security measures and separation for virtualized services. A significant concern in implementing NFV in mobile networks is the inherent volatility of VNFs, which often results in setup errors and subsequently compromises security. In addition, Virtual Network Functions are susceptible to common cyber-attacks such as spoofing and Denial of Service. NFV is susceptible to a unique range of virtualization risks, including side-channel attacks, flooding attacks, hypervisor hijacking, malware injection, attacks linked to Virtual Machine (VM) migration, and vulnerabilities specific to cloud environments. Even the VNF deployed in private cloud are not secure as this implementation of NFV is susceptible to hostile insiders, such as a malicious administrator or third party contractors. Because the cloud infrastructure is easily accessible, a malicious user or a compromised provider of VNF can disrupt the operations of the infrastructure by introducing malware or altering network traffic. Operational interference and misuse of shared resources are seen as infrastructure-level attacks on Network Function Virtualization (NFV). As a result of the widespread availability of physical infrastructure resources, an attacker can disrupt the functioning of the infrastructure by introducing malware or manipulating network traffic. During resource misuse assaults, the victim does not receive any advantage from the shared or dedicated resources. Ensuring confidence in virtualized NFV systems is a significant challenge. Typically, a reliable person is responsible for the installation and configuration of physical network devices, and there is a pre-existing level of trust in the device. Since VNFs are being retrieved dynamically from the cloud, it is necessary to implement a trust mechanism to safeguard against malicious VNFs.

Table 3: Threats in VNF

Security Threat /Attack	Target Point/Network Element	Effected Area
DoS attack	Centralized control elements	**NFV**
Hijacking attacks	SDN controller, hypervisor	**NFV**
Resource (slice) theft	Hypervisor, shared cloud resources	**NFV**
Configuration attacks	Virtual switches and routers	**NFV**

2.4.2.3. Threats in MEC

Unlike NFV systems, which are situated within the internal network of the mobile network operator and limit access and usage by end users, third-party Multi-access Edge Computing (MEC) apps can operate on the border of data network and mobile network operator's network. This creates an increased exposure to malware [53] and other types of attacks. The shared nature of cloud computing systems, it is feasible for a user to distribute harmful traffic in order to disrupt the overall system performance, consume excessive resources, or secretly access the resources of other users. Multi-access Edge Computing, same as any other cloud system, consists of various interconnected technologies that work together in an open ecosystem [52]. Service providers utilize virtualization and distributed computing to install and give more specific and practical solutions, rather than generic ones. For instance, the traffic analysis, learning-based system collects a specific quantity of packet samples and analyzes them for several recognized characteristics in order to identify patterns. Mobile terminals are protected by installing anti-malware solutions either on the device itself or by accessing them directly from the cloud. The security framework in MCC data and storage will include energy-efficient ways for verifying the integrity of data and storage services. This will be done in combination with a public proven data possession scheme and lightweight storage outsourcing that can withstand compromises.

MEC data centers, due to their open environments and the presence of various applications, APIs, data, and technologies, are susceptible to abuse, exploitation, and misuse. These vulnerabilities might manifest as asset abuse, supply chain compromise, misconfiguration, and inadequate security measures implemented. The MEC data, whether at rest or in transit, pose increased security risks because of the open and complex nature of the environment. A MEC data center may encounter many dangers once it becomes operational. An effective attack on a single application running in MEC has the potential to exploit the whole MEC ecosystem and serve as a means to launch a more extensive attack on the 5G Core network or supply chain.

Just like in any other cloud system the utilization of resources also presents a substantial risk in a virtualization environment with multiple tenants. MEC is not an exception. When a tenant excessively consumes resources, it can cause a Denial of Service (DoS) event for neighboring tenant systems. Additionally, this activity can lead to compromising or significantly reducing the functionality of the solution. Colocation attacks, such as VM/container escape or side-channel assaults, have the potential to jeopardize neighboring tenant compute workloads by depriving them of resources. The risk of compromising data confidentiality, integrity, or availability is also possible [53].

Table 4: Threats in MEC

Security Threat /Attack	**Target Point/Network Element**	**Effected Area**
DoS attack	Centralized control elements	**NFV**
Hijacking attacks	SDN controller, hypervisor	**NFV**
Resource (slice) theft	Hypervisor, shared cloud resources	**NFV**
Configuration attacks	Virtual switches and routers	**NFV**

2.4.2.4. Privacy related challenges

5G technology facilitates numerous innovative applications that have the ability to create opportunities for a wide range of vertical industries. Consequently, a substantial quantity of personal data will be transmitted across the 5G networks. The implementation of data-mining tools has made it simpler to access data privacy information, hence posing a significant risk to the data. The 5G system must have security procedures to safeguard a wide range of trusted information, including data related to both human users and machine users. This information includes but is not limited to identification, subscribed services, location/presence information, mobility patterns, network usage behavior, and regularly used applications [54]. In order to safeguard user privacy in 5G networks, it is essential to establish mutual agreements and trust models among all stakeholders, including the user, network operator, service provider, application developer, and manufacturer. These agreements should address data usage and storage. They should also specify the technology which will necessitate enhanced systems for ensuring responsibility, reducing data collection, promoting clarity, facilitating openness, and managing access permissions. Methods such as obfuscation are essential for protecting location privacy by intentionally reducing the accuracy of location information. Furthermore, algorithms that utilize location cloaking are highly effective in mitigating significant location privacy threats, including timing and boundary assaults. To mitigate IMSI catching attacks, an effective option is to employ Temporary Mobile Subscriber Identity (TMSI) that is produced randomly and periodically assigned to the User Equipment (UE). The long term IMSI is only used during the fault recovery procedure and when the TMSI has not been allocated yet. Another threat are the counterfeit base stations capable of intercepting the subscriber's IMSI.

Table 5: Privacy related threats

Security Threat /Attack	**Target Point/Network Element**	**Effected Area**
Operator Data Leak	Registers and databases	**Privacy**
User-specific identifiers leak	EIR, UDSF, SDSF	**Privacy**
Location Tracking	gNodeB, AMF	**Privacy**
Information disclosure	VNFs	**Privacy**
Data Interception	MEC, UPF, SEPP	**Privacy**

2.4.3. Threat model for 5G core

This section of the dissertation covers threat model developed for a 5G network. It highlights various potential security vulnerabilities and attack vectors within both the Control Plane and the User Plane functions.

The threats span a range of categories, from API exploitation to the abuse of network access, and from malicious flooding to sophisticated slide-channel attacks. Model is based on the threats identified with use of Bhadra framework described in chapter 2.4.2 and on ENISA “Threat Landscape for 5G Networks Report” [55].

Control Plane:

- API Exploitation.: Attackers can exploit vulnerabilities in APIs (Application Programming Interfaces) to manipulate network resources. This includes unauthorized access to network functions and data, leading to potential service disruptions and data breaches.
- Manipulation of Network Resources. This involves attackers altering network settings or resources to degrade service quality or redirect traffic, causing denial of service (DoS) or man-in-the-middle attacks.
- Abuse of Third-Party Hosted Network Function. Utilizing third-party network functions can introduce risks if these functions are compromised, potentially leading to unauthorized access and control over the network.
- Registration of Malicious Network Functions. Attackers may register malicious network functions within the network infrastructure, which can then be used to disrupt services or steal information.
- Fraud Scenarios Related to Roaming Interconnections. When users roam between different network providers (VPLMN and HPLMN), vulnerabilities can be exploited to conduct fraud, such as unauthorized service use or billing scams.
- Abuse of Network Access. Unauthorized access to the network can lead to exploitation of resources, data theft, and further infiltration into network components.

- Lateral Movement. Once inside the network, attackers can move laterally across different network functions and services, escalating privileges and expanding their control within the network.
- Abuse of User Authentication/Authorization Data. Exploiting weaknesses in user authentication processes or authorization data can lead to unauthorized access to services and sensitive information.

User Plane:

- Exploit Poor Misconfiguration. Poorly configured network components can be exploited to gain unauthorized access or to disrupt services. This includes misconfigured firewalls, routers, and other network infrastructure elements.
- Man-in-the-Middle Attack. Attackers can intercept and manipulate communications between users (UE) and the network (RAN), leading to data breaches, fraud, and service disruption.
- Exploit Poorly Designed Architecture. Inherent weaknesses in network design can be targeted, leading to vulnerabilities that can be exploited for unauthorized access, data theft, or service disruption.
- Slide-Channel Attacks. These sophisticated attacks exploit side channels in the hardware or software implementation of the network to extract sensitive information without direct interaction with the network interfaces.

Cross-Plane Threats:

- Malicious Flooding of Core Network Components. Attackers can flood core network components with excessive traffic, causing denial of service and disrupting normal network operations.

This threat model underscores the complexity and multifaceted nature of security in 5G networks. It highlights the need for robust security measures, continuous monitoring, and regular updates to safeguard against evolving threats.

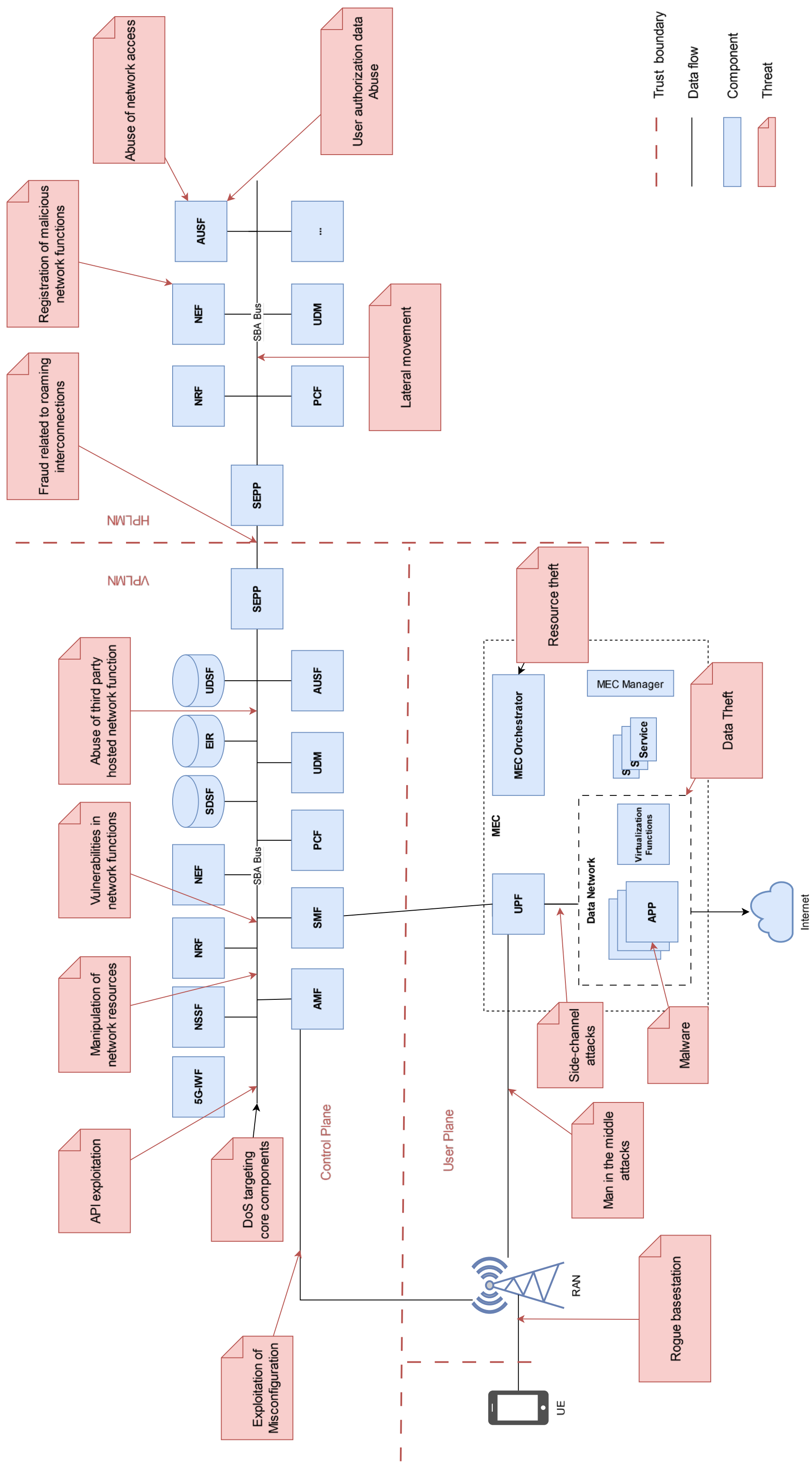

Figure 13: Example of threats targeting 5G network.

3. State-of-the-art in Threat Modeling

Threat Modeling is a process normally used in the earlier stages of the system design activity for the purpose of identifying, assessing, and addressing potential security threats to a system. Outcome of that process is a threat model that highpoints potentially exploitable weaknesses of the system. Based on the identified threats, related probabilities and risks, different mitigation techniques should be planned to secure the system from malicious activities. Threat modeling can be used not only to identify but also to mitigate various security threats in the system that is still in the design phase. One of the goals of this work is to find the AI applications that can potentially support the threat modeling approach to identify the threats in digital systems basing on already established threat modeling methodologies, i.e., The Visual, Agile, and Simple Threat (VAST), STRIDE, Common Vulnerability Scoring System (CVSS) and Cyber Kill Chain (CKC). All of them are well suited for identification of system and application-level threats.

The threat modeling process, to put it simply, involves providing honest answers to three questions [56]:

1. What elements does the system under analysis consist of and how does data flow between them? At this point, it is extremely important to understand the system, identify all its components and data flows between them. We need to consider what data we want to protect and what data may be particularly desired by an attacker.

2. What threats may occur? The next step is the essence of this process. It requires us to both know the types of threats, as well as imagination and an open mind to honestly identify all the weak points of our system. This step involves designing worst-case scenarios that may occur in relation to our solution.

3. Do I have adequate control mechanisms implemented? The last element of threat modeling is verification whether (and to what extent) the system is resistant to the threats that we have identified. We should take into account the already implemented control measures, as well as design additional ones if we have identified threats against which we are not protected (there are exceptions to this, which we write about in the following section, describing the risk analysis).

These threats, of course, may have a different nature and origin. Each of them will also have a different probability of occurrence. For example: when modeling threats for an online store, we will identify the possibility of placing malicious code (backdoor) by its programmer, but the probability of this will be low because the development process requires second-hand code reviews and we periodically perform security tests that include code analysis. Keep in mind that it will not be possible to identify a finite set of all threats related to the use of the analyzed application. However, it is worth building an exhaustive list of them, based on our best knowledge, available materials and the knowledge of other people who have knowledge of the analyzed system.

When talking about threat modeling and risk analysis, we should become familiar with a few basic concepts that we will use for the purposes of this chapter.

- Assets - everything that has value for the organization (e.g. physical device, software or data). Vulnerability - a weak point (security gap) in the system that allows a threat to occur. An example of a vulnerability in a web application may be the SQL injection error.
- Threat - a potential event resulting from the use of a vulnerability, having an undesirable impact on the system or its resources. An example of a threat may be an attacker's access to the system's database.
- Attack - using a vulnerability to materialize a threat. An example of an attack could be an attacker sending a string of characters to a web application, which will display the contents of the entire database using an SQL injection error.
- Risk - the product of the probability of a given threat materializing and its effects (e.g. on a low, medium-high scale).

Although the definition of risk itself seems complicated, we recognize the meaning of this concept instinctively. Let's imagine that the 5G core elements are located on cloud infrastructure located in Warsaw. When modeling hazards and estimating risk, we take into account the threat associated with the possibility of an earthquake. While the effect of such an earthquake itself can be classified as significant (potential destruction of the server room), the probability of its occurrence will be classified as extremely low in our geographical conditions and after analyzing the history of such disasters in recent years. Therefore, we will assess this type of risk as medium or low (depending on the existing security measures and the adopted risk assessment methodology) [56]. Risk in threat modeling can often be calculated using the formula:

$$Risk = Impact \times Probability$$

Equation 1: Risk calculation

Where Impact is the potential damage or loss caused by the threat and Probability (often used in literature as Likelihood) is the probability of the threat occurring. If multiple independent events need to happen for a threat to materialize, the probability can be calculated as:

$$P(Threat) = P(A) \times P(B) \times P(C) \ldots$$

Equation 2: Event probability

Where *P(A),P(B),P(C),…P(A), P(B), P(C)*,… are the probabilities of individual events.

Bayesian probability can be also useful in threat modeling when updating the likelihood of a threat based on new evidence:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Equation 3: Bayesian Analysis

In the above equation *P(A)* is the probability of event *A*. *P(B)* probability of event *B*. *P(A/B)* is the probability of event *A* given event *B*. *P(B/A*) is the probability of event *B* given event *A*. For complex threat models, such as the ones created for 5G networks or their sub parts a quantitative risk calculation needs to be performed. This can be calculated from following equation:

$$Risk = \sum_{i=1}^{n} (Impact_i \times Probability_i)$$

Equation 4: Quantitative Risk

Impact and Probability of i-th threat is considered separately.

3.1. Threat Modeling basics

Although, there are multiple methodologies (most used ones are analyzed in chapter 3.2) they all have similar requirements. First, information about the system needs to be collected. Identifying negative scenarios related to the use of the analyzed application is not an easy process and requires appropriate knowledge and experience. It is therefore worth using all available documentation. To model threats properly the collected documentation should include:

- solution architecture – providing knowledge of all application components (both physical and virtual),
- description of the "business" functions of the application - before starting modeling, it is extremely important to know about all processes supported by the application's use by its user,
- list of dependent systems - the process should include all places where information is obtained by the analyzed application, as well as the recipients of this data (sometimes these may be other systems, sometimes people who perform certain actions on this basis).

People with knowledge of the structure and operation of the application should also be involved in the threat modeling process. Key personnel that should be considered in this process consists of:

- developers - unless, the system is a legacy one created by programmers which we simply cannot invite to participate in such an analysis,
- the business owner of the application - i.e. the person responsible for the area in which the application is used and who has the best knowledge of the processes in which it takes part, the expected course of action, as well as various negative scenarios; it may turn out that the lack of availability of the application will seem unacceptable to the security person, but from the business side it will be completely acceptable,

- representatives of the unit responsible for security in the organization - they should provide assistance with their experience in identifying threats specific to a given organization [56].

An exhaustive examination of system architecture, business context, and artifacts such as functional specifications and user documentation enables threat analysts to uncover significant parts of the system, whether they are connected to security or not, and develop a comprehensive understanding that may currently be lacking. After selecting the participants, gathering the necessary data model of the system needs to be created. Threat analysts should construct a streamlined diagram encompassing all relevant components, including those that are nearby or connected (even if they are not within the scope of the analysis), along with their respective connections and protocols. Constructing a diagram of the system's architecture offers a concrete visual comprehension of the system. Detailed information about using DFD (data flow diagrams) for threat modeling can be found in [57].

After the completion of the component diagram, the next stage requires the inclusion of assets, controls, threat agents, and trust zones by threat analysts in order to construct the comprehensive threat model diagram. The diagram should comprehensively enumerate all plausible vulnerabilities to the system, with particular emphasis on data, dataflow, platform, and operational security. Additionally, it should detect any vulnerable or absent security measures that could result in a successful breach. After being discovered, the threat analysts should indicate the precise position of each threat. Example, high level, diagram is presented in Figure 13). Upon completion of the model, last step is the examination of the dataflow and connections, component by component, by threat analysts in order to compile a comprehensive inventory of all plausible threat scenarios. A traceability matrix is on of the tools used to document and track deficiencies in controls [58], enabling the development of a systematic approach to prioritize and address them. To approach this process systematically, several methodologies and frameworks were established.

3.2. Methodologies Overview

Common distinctions for approaches used for threat modeling activities are their main focus. The approaches include:

- asset-centric threat modeling,
- attack-centric threat modeling,
- software-centric or system-centric threat modeling,
- data centric threat modeling.

All of them can be used to produce threat models, but depending on business needs, scope, available data and resource different approaches will yield more useful results.

3.2.1. Data-centric threat modeling

In contrast to Software/System-centric threat modeling, instead of concentrating on specific hosts, operating systems, or programs, data-centric threat modeling concentrates on particular categories of data within a system [59]. The system and data of interest must be correctly located, described, and specified in the initial phase of the data-centric threat modeling process. The features of components used for storing, transferring, executing, entering, and outputting data within the system need to be carefully reviewed to identify all related threats. The process should include the data flows between authorized locations, security goals (aligned with business needs and regulatory expectations), and processes related to data lifecycle.

3.2.2. Attack-centric threat modeling

Examining system dangers from an attacker's point of view is the main difference between asset and attack centric approach. The goal of the attack-centric threat modeling approach is to determine whether attacks, given a variety of reported abuse cases, vulnerabilities, and other factors, can be successfully carried against a system [60]. Additionally, this approach makes an effort to investigate the intention, sources, and relative identification of the attacker or group connected to the attacker, as these can reveal the strategy and resources of the attacker.

3.2.1. Asset-centric threat modeling

Asset-centric approaches to threat modeling put an emphasis on the system's assets. Analyzing the business impact or information loss of the targeted assets is involved. Asset centric threat modeling can be expanded to include the detection of security holes in the system environment in addition to determining the goals and intentions of the attacker. It may be also used to identify potential threat scenarios not concerned unsafe coding or design practices [61].

3.2.2. Software-centric / System-centric threat modeling

This kind of threat modeling approach focuses on the software used in the system being threat modeled. Its focus point is the design model of the system under concern. The important step in this approach is to understand the full complexity of the system being threat modeled. To achieve it, visibility of software components and ideally the source code of the application should be available so that all threats to the system can be identified. Software-centric threat modeling results are best if deep enough technical information is available to the specialists who are responsible for the threat modeling exercise. Personnel involved in threat modeling of a system should have deep technical knowledge and a fair understanding of the system components and its technology stack.

This work concentrates on software-centric and asset centric threat modeling approaches. This is mainly due to the fact that organizations such as network operators and telecoms (operating in cloud infrastructure) have a clear understanding of their business objectives [62]. But also, because assets have huge value for any organization operating in cloud infrastructure, and even though different enterprises and industries have different asset structures, assets are in vast majority of cases the foundation for services and products provided by an enterprise [63]. Separate factor considered while selecting the approach comes from the fact that automation of processes related to data and attack-centric threat modeling is more complex as expert knowledge and in many cases prediction skills are necessary to model threats which may materialize in the future.

3.2.3. STRIDE

STRIDE is a methodology used to determine possible threats used as a part of the secure software development lifecycle (SDLC). It takes into account all components of the system and is designed to model the complete system. With the use of data flow diagrams [64] (DFDs), STRIDE identifies system entities, events and particular segments or borders of the system. STRIDE groups the threats using predefined set of categories which names form the methodology abbreviation, as shown in the table below.

Table 6: STRIDE threat categories

	Category	**Description**
S	Spoofing	Masquerading of a legitimate user, process or system element
T	Tampering	Modification/editing of legitimate information
R	Repudiation	Denying or disowning a certain action executed in the system
I	Information disclosure	Data breach or unauthorized access to confidential information
D	Denial of Service	Disruption of service for legitimate user
E	Elevation of Privilege	Gaining access to functions which should be protected by security controls

Because modern digital systems are architecturally complex solutions, built from multiple components, STRIDE is one of the best options for threat modeling. Mainly due to the fact that STRIDE is the most mature methodology. STRIDE addresses threats that could be exploited by the attacker to compromise the whole system. It aims to find all of the interdependencies between system components. Research conducted in the past identified STRIDE as an effective approach towards ensuring the system security at the component level and has shown that the entire system security can be ensured by addressing vulnerabilities of each system component separately [47].

3.2.4. VAST

The Visual, Agile, and Simple Threat (VAST) Modeling methodology is supported by ThreatModeler[5], an automated threat-modeling tool. The VAST methodology similarly to STRIDE aims to embed the threat modeling process as the inseparable element of the SDLC. The difference is that VAST tries to enforce it throughout the whole enterprise. Its scalability and usability allow it to be applied to large systems and even organizations. The methodology “provides a unique and straightforward way to visualize system/application architectures and does not require extensive security expertise, thus making it accessible to a wider audience” (Agarwal, 2016). VAST implementation requires producing two different types of models: Application Threat Model (ATM) and Operational Threat Model (OTM). Application threat models are created using process flow diagrams (PFDs). Process flow diagrams describe the functions of application components and the communication between them in the same way that developers and architects describe the construction of applications in the design phase. Operational threat models are intended for infrastructure teams. Although more similar to traditional DFDs than application threat models, data flow information is presented from the perspective of the attacker rather than the data flow.

3.2.5. CVSS

Common Vulnerability Scoring System (CVSS) is a threat scoring system used for assigning quantitative values to existing vulnerabilities. It was developed by the National Institute of Standards and Technology as a methodology designed to support the security teams to assess the threat-related risk, possible impacts and to identify applicable countermeasures. Due to the fact that the vulnerability impact and exploitation probability can vary from system to system, CVSS includes functions allowing to adjust the calculated vulnerability scores based on the configuration of the system in question. Although in the past CVSS was used mainly for supporting Risk Assessment [65], recent changes in methodology documentation (version 4.0 of CVSS) emphasize that CVSS is intended to measure the severity of a vulnerability and not to assess any

---

[5] ThreatModeler is a threat modeling platform that automates the process of identifying and mitigating security risks during the software development lifecycle. It is a commercial software available at https://threatmodeler.com/.

risks directly [66]. The CVSS v4.0 Specification Document also states that the CVSS Base Score should be accompanied by more contextual analysis of the environment. Therefore, the CVSS scoring values should be used only as a part of the comprehensive risk assessment or threat modeling system. Similar conclusions were drawn from the research [65] in which authors, using the CVSS as the fundamental system of measurement linked with additional low-level metrics, proposed and discussed a method for modeling the risk and shaping better decisions related to cybersecurity threats.

3.2.6. TRIKE

Trike was created as a security audit framework which focuses on management perspective. When creating a model using this methodology, we strive to achieve four goals: Assure stakeholders or other persons responsible for business decisions that the risks associated with each component of the modeled system are acceptable. Be able to determine that stakeholders are confident and accept the risk. Communicate what threat mitigation measures we have implemented or identified. Enable stakeholders to understand and reduce the risks our decisions may generate in other areas. In addition to being focused on business goals, TRIKE is also distinguished by its threat assessment module. When assessing threats, we use a list of all components/resources of the modeled system, taking into account connections with external resources. For each asset, the risk of attack is rated on a five-point scale for each CRUDXF action.

Table 7: CRUDEFX actions in TRIKE

C	Create	Create a new object. For example, create new entry in EIR registry of network equipment.
R	Read	View or use the contents of an object. For example, select values from a database, or compare user TMSI identifier against expected value.
U	Update	Changing the content of an object. For example alter AMF configuration in live network.
D	Delete	Delete or destroy an object. For example, delete an instance of network element to save resources when network is underutilized.
X	eXecute	Execute. For example, run a program, call a specific function in a library, or run a script written in an interpreted language.

F	conFigure	Object configuration. For example, set file permissions, change header bits in a TCP packet, or set configuration registers in a hardware device.

Actors who can perform the above-mentioned actions are assessed separately. Using a five-point scale based on trust resulting from the user's role in the system, values are assigned to each of the defined actors. A trusted administrator with full privileges would be assigned a rating of one, while an anonymous user with the most restricted access should be assigned a rating of 5. In this methodology, each attack falls into one of two attack types: escalation of privileges or denial of service. This makes the analysis less complicated than using the previously discussed STRIDE. TRIKE, combining risk analysis, threat modeling and emphasizing business aspects, has become a popular tool in many corporations. It is still used today, but not as widely as a decade ago. Spreadsheets and tools supporting work with it are publicly available, but these are quite old solutions and the project website has not been updated for many years. Another problem is the lack of documentation for version 2.0 of TRIKE.

3.2.7. Cyber Kill Chain

Apart from the above well-established threat modeling methodologies and techniques, in the recent years the Intelligence Driven Defense (IDD) methodology (based on Cyber Kill Chain practices) became a popular method for ensuring that defensive capabilities are implemented in the systems that should be protected from adversaries targeting them with malicious intent. Although CKC at first glance seems to have more in common with Attack Trees [67], it is in fact yet another threat modeling methodology in which threats are mapped to the predefined stages of a compromise and bounded directly to actions describing the common methods that attackers use.

The process of modeling threats using CKC methodology breaks the attack into seven stages. Each of these include the opportunity for a breach which should be detected, prevented or mitigated by the security controls. Defenders can measure the performance as well as the effectiveness of controls already implemented in the system and plan further security investments based on the findings. The key element in CKC methodology is the Courses of Action Matrix presented in Table 8. For each potential threat and its phase, countering controls should be put in place and they should be able to address the actions of detect, deny, disrupt, degrade, deceive, and destroy as defined in USA Department of Defense information operations doctrine [68]. For example, in case of a threat in the form of ransomware, CKC should identify Network Intrusion Detection System (NIDS) as the control countering the threat in Weaponization/Delivery phase by detecting it. The antivirus system would disrupt the same threat in the Installation phase and its exploitation phase would be denied by applying proper security patches. Courses of Action Matrix aims to find the best fitting controls, so that they can be prioritized and implemented in accordance with the system needs.

Table 8: CKC action matrix

<table>
<tr><th>Phase</th><th colspan="6">Action</th></tr>
<tr><td>Reconnaissance</td><td rowspan="7">Detect</td><td rowspan="7">Deny</td><td rowspan="7">Disrupt</td><td rowspan="7">Degrade</td><td rowspan="7">Deceive</td><td rowspan="7">Destroy</td></tr>
<tr><td>Weaponization</td></tr>
<tr><td>Delivery</td></tr>
<tr><td>Exploitation</td></tr>
<tr><td>Installation</td></tr>
<tr><td>Command and control</td></tr>
<tr><td>Actions on Objectives</td></tr>
</table>

3.3. Applicability of AI for threat modeling

The concept of using AI techniques in the cybersecurity field is not new, e.g. malware prevention and detection supported by AI/ML related techniques are successfully used [69],[70] in multiple commercially available antivirus products. Feasibility of AI techniques in other cybersecurity areas gained more attention with the increased availability of computing power and toolboxes supporting the implementation of deep learning algorithms. Proposals for AI application areas are discussed in [71], [72], and [73]. For some of them a detailed performance analysis of the applicable techniques was checked [74]. However the systematic comparison of AI techniques applicability in threat modeling field was not yet conducted.

Detailed AI learning algorithms description is not in the scope of this work, but for the completeness, a brief introduction to the most important techniques and categories is needed. AI as a branch of the computer science aims to help automate the decision making processes which normally require human intelligence. General AI which is commonly referred to as strong AI theoretically should be able to understand and perform any logical task. In contrast, narrow AI doesn't require full cognitive capabilities, therefore it can be used to perform pre-learned or reasoning based tasks. Expert systems are a great example of narrow AI implementation. Where it comes to learning algorithms, in this research we focus on two types used to train the AI:

- Unsupervised learning, which is using unlabeled training datasets, is commonly used to cluster data or estimate the concentration of elements.
- Supervised learning, requiring large, labeled set of data for the training process. This type of machine learning is used mainly as a regression or classification mechanism.

From the threat modeling point of view interesting AI Methods are also Deep Learning (DL) and Biocomputational methods. DL is a subset of Machine Learning that aims to mimic human brain capabilities and its neural structure used for processing signals [75]. DL uses artificially constructed neural networks and algorithms such as: CNNs (convolutional neural networks), FFNs (feed-forward networks), RBMs (restricted Boltzmann machines), etc. together with large datasets. Biocomputational algorithms are relatively new approach based on the principles of biological evolution and solutions developed by nature [76].

In the last years, the interest in the application of biocomputational algorithms has grown and the following techniques (applicable in the cybersecurity) gained attention: genetic algorithms (GA), evolution strategies (ES), artificial immune systems (AIS), ant colony and particle swarm optimization (ACO and PSO). Table 4 contains the applicable areas related to the cybersecurity and the AI/ML techniques which can be used to address challenges in those.

Table 9: AI/ML applicability for cybersecurity area

AI/ML technique	Cybersecurity area	Applicability in threat modeling
General AI	ALL	Full
Unsupervised learning	Clustering of Threats Countering APT Counter Cyberespionage DDoS prevention Network Activity Monitoring Traffic Analysis Virus detection	Partial
Supervised learning	Anomaly Detection Countering APT Cryptojacking Counter Cyberespionage Data protection DdoS prevention DNS Monitoring Fraud Detection Intrusion Detection Network Activity Monitoring Phishing detection Spam Classification Virus Detection	Full
Deep Learning	ALL	Full
Biocomputational methods	Anomaly Detection Clustering of Threats Data protection DdoS prevention Fraud Detection Intrusion Detection Network Activity Monitoring Traffic Analysis Virus detection	Partial

The applicability of AI technique in the threat modeling as well as its usefulness for certain cybersecurity areas of interests, originate mainly from the capabilities of AI technique to cluster the data and its effectiveness for problems where amount of training data is limited. Hence unsupervised learning techniques are not recommended for Traffic Analysis or Fraud Detection. When considering the detailed usefulness of AI types in threat modeling recommendation presented in Table 10 can be followed.

Table 10: Possible usage of AI in threat modeling

Method	Unsupervised Learning	Supervised Learning	Deep Learning	Biocomputational
CVSS	+	+	+	+
STRIDE	-	+	+	+
VAST	-	+	+	+
CKC	-	+	+	+

The above table presents the findings of the research based on the basic assumptions of AI technique and the requirements of each threat modeling approach, described in chapter II. General AI (strong AI) as a theoretical form was removed from this analysis.

Deep Learning methods consist of many different types of algorithms, each of them to a certain extent can be utilized by all of the studied threat modeling techniques. Neural Networks can be applied to solve most of the problems faced in threat modeling starting from data clustering (proven in 126), up to generating graphs useful in VAST method. Biocomputational methods can also be used in all threat modeling techniques is scope, because they group many different algorithms ranging from evolution strategies (theoretically fit to support VAST) to particle swarm optimization (applicable for CVSS TM). CCK and VAST by design require the expert knowledge, therefore Unsupervised Learning should not achieve much success in this approach. Nevertheless ULs high capability of solving issues related to data clustering will be useful in both STRIDE and CVSS where grouping of threat data is one of the key elements.

3.4. Automated threat modeling

Commercial software offers opportunities to automate the process, and in many cases the cost of licensing will be much lower than the cost of hiring more security specialists. However, it will not fully reduce the need to devote work time to developers, architects and testers, who will be forced to support the tool's operators by answering numerous questions related to the modeling process. The challenge to full automation is the complexity of the systems and the limited capabilities of pattern recognition software. AI can help to solve this issue as proposed in chapter 4.4.2, but even without AI there are two other automation options. It is possible to use+ commercial solutions or implement solutions that will help us describe the modeled system in a computer-understandable way [56]. Computers understand code, so in order for them to understand the threats and be able to help us generate a model, we must provide them with the appropriate instructions. It can be done in two ways. The first is to provide the model code, with use of solutions such as Threagile or PyTM. Both of them are capable of creating models automatically based on the code we provide ourselves.

```
from pytm.pytm import TM, Server, Datastore, Boundary, Actor, Dataflow

tm = TM("5G AMF Threat Model")
tm.description = "Simplified threat model for AMF in 5G core"

# Boundaries
UE_RAN = Boundary("UE/RAN")  # User Equipment / Radio Access Network
Core = Boundary("5G Core")

# Actors
UE = Actor("User Equipment (UE)")
UE.inBoundary = UE_RAN

Attacker = Actor("External Attacker")
Attacker.inBoundary = UE_RAN

# Datastores
UE_DB = Datastore("UE Subscription Data")
UE_DB.inBoundary = Core

# Servers (Network Functions)
AMF = Server("Access and Mobility Management Function (AMF)")
AMF.inBoundary = Core

# Dataflows (Important for AMF)
registration_request = Dataflow(UE, AMF, "Registration Request (NAS)")
```

```
registration_response = Dataflow(AMF, UE, "Registration Response (NAS)")
location_update = Dataflow(UE, AMF, "Location Update (NAS)")
context_transfer = Dataflow(AMF, AMF, "Context Transfer (N11)")  # Simplified

# Threats
# - Attacker intercepts/modifies NAS messages
# - Attacker performs Denial of Service (DoS) on AMF
# - Insider in Core misuses UE data
# ...
```

Code Listing 1: PytTM code example for AMF network element.

In same manner PyTM can be used to model any other element of 5G network including a specific application residing in MEC:

```
from pytm.pytm import TM, Server, Datastore, Boundary, Actor, Dataflow

tm = TM("5G MEC Application Threat Model")
tm.description = "Threat model for an app running in a 5G MEC environment"

# Boundaries
MEC_Host = Boundary("MEC Host") # Where the application runs
MEC_Network = Boundary("MEC Network") # Connects to other MEC nodes, 5G Core, etc.
Internet = Boundary("Internet") # For external access (if applicable)

# Actors
MEC_App_User = Actor("MEC Application User (UE or other device)")
MEC_App_User.inBoundary = MEC_Network # Could be UE/RAN or inside MEC Network
External_Attacker = Actor("External Attacker")
External_Attacker.inBoundary = Internet

# Datastores
App_Data = Datastore("Application Data")
App_Data.inBoundary = MEC_Host

# Servers
MEC_App = Server("MEC Application")
MEC_App.inBoundary = MEC_Host

# Dataflows (Examples)
user_interaction = Dataflow(MEC_App_User, MEC_App, "User Interaction (API calls, etc.)")
data_storage = Dataflow(MEC_App, App_Data, "Data Storage/Retrieval")
external_access = Dataflow(MEC_App, Internet, "External Access (if applicable)")

# Threats
# MEC Host Threats
# - Unauthorized access to MEC host (physically or remotely)
# - Resource exhaustion attacks (CPU, memory, etc.) on MEC host
# - Data exfiltration from App_Data
# - Compromise of MEC_App code/dependencies
# MEC Network Threats
```

```
# - Interception/modification of dataflows within MEC network
# - Lateral movement within MEC network to other nodes/applications
# - Denial of Service (DoS) attacks targeting MEC network infrastructure
# External Access Threats (if applicable)
# - Typical web application attacks (SQL injection, XSS, etc.) if app is exposed
# - DDoS attacks from the Internet
```

Code Listing 2: PytTM code example showing a threat model for application deployed in MEC.

Other useful tool is Threatspec. The application allows developers and security engineers to work together in the same code repository and create threat specifications in parallel with the code. Special tags recognized by the tool are used to generate reports and data flow diagrams based on the source code. This has an additional advantage in the form of forced analysis of the code in terms of security, and it also focuses programmers' attention on aspects related to the threats specified by security experts. The most important element of Threatspec is the parser, which processes source code files and collects the annotations found in these files, supporting many different programming and scripting languages [56].

Below is a representation of how threat modeling with Threatspec could be implemented in 5G code base.

```
/*
@component External:User
@component 5G:Core:AMF
@component 5G:Core:SDSF

@connects #user to #AMF
@connects #AMF to #SDSF with real_connect

*/
int main(int argc, char **argv)
{
  MYSQL *con = mysql_init(NULL);
  if (con == NULL)
  {
        fprintf(stderr, "%s\n", mysql_error(con));
        exit(1);
}
/*
# Threats
@threat Authentication Info Disclosure (#auth_info_disclosure):
  description: An attacker can obtain information about existing users to the system
# Exposures
@exposes #db to #auth_info_disclosure with hardcoded credentials
*/
```

```
 if (mysql_real_connect(con, host, "root", "root_passwd",
           NULL, 0, NULL, 0) == NULL)
 {
...
 }
 mysql_close(con);
 exit(0);
}
```

Code Listing 3: Threatdpec implementation for 5G

Based on the information provided in the source code, tool will be able to generate a threat model automatically both in form of a markdown report file and in graphical format presented in Figure 14.

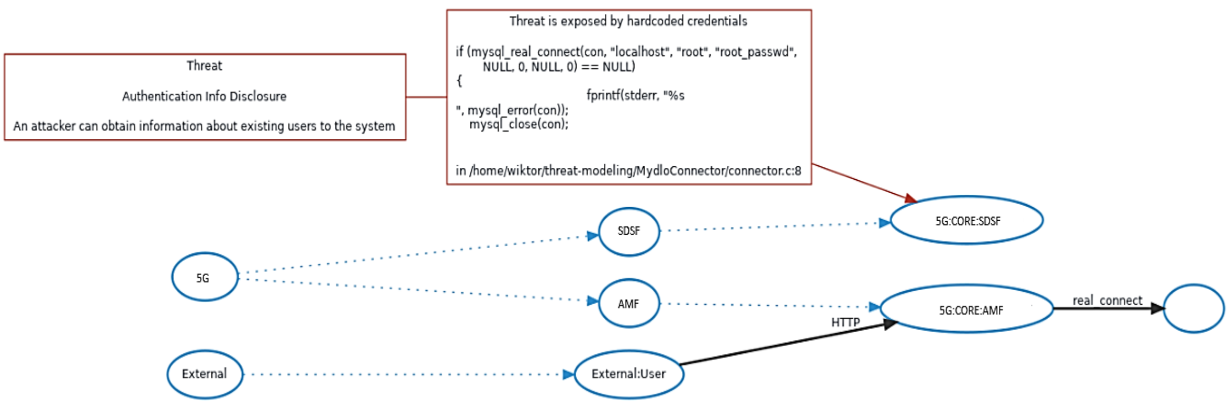

Figure 14: Threatspec generated model

4. Automating threat modeling process

Due to the dynamic changes occurring in cloud infrastructure and the increasing complexity of modern system architectures, it is difficult to manually maintain a comprehensive threat model for a system operating in the cloud. Cloud workloads are dynamic, being launched and disbanded based on user needs. They are periodically updated and restructured, making manual processing and regeneration of related threat models extremely challenging without at least partial automation. Even in less dynamic environments, models can quickly become outdated [78], leading to attempts to automate the threat modeling process [79],[80],[81]. However, automation in this area is not an easy task because each business operating in the cloud has its own virtual infrastructure and development environment, which may differ significantly from others. As a result, the threat modeling methodology needs to be flexible and adaptable to be useful in any situation. Some businesses might own specialized or even self-made workloads that must be considered while creating a useful threat model.

To aid in the threat modeling process, many tools have been created over the years, and nowadays the tool suite available for experts responsible for threat modeling tasks is quite impressive. Although the tools have a similar overall objective, their features and work processes differ. The selection of a tool will, therefore, depend on the particular use case.

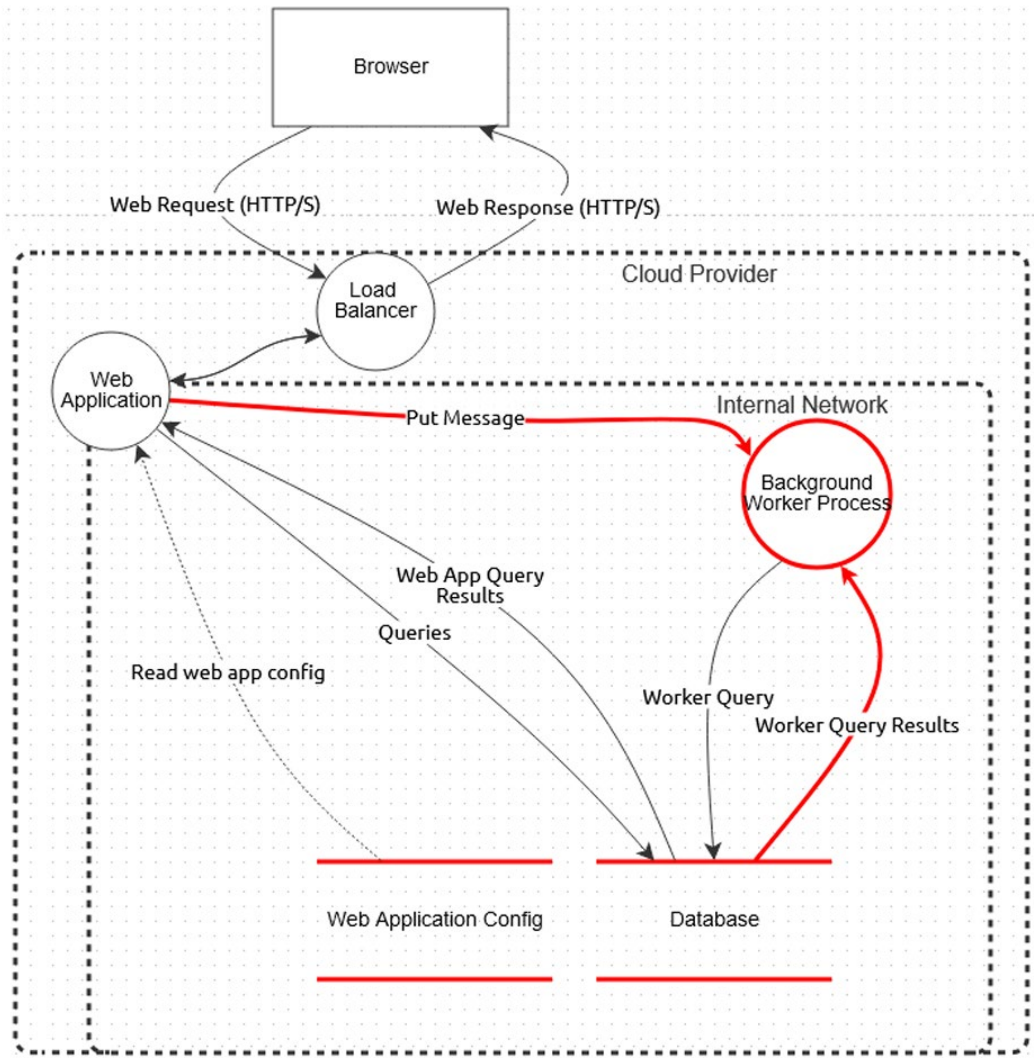

Figure 15. Diagram based modeling in OWASP Threat Dragon.

Source: Threat Dragon.

The most popular tools used in connection with the current techniques and methodologies have been the subject of several distinct studies. For example the most commonly used products available on the market are evaluated in [82]. Authors compare six threat modeling tools under a predefined set of criteria, focusing on threat modeling Tool, OWASP Threat Dragon, Open Weakness and Vulnerability Modeler (OVVL), OWASP pytm, Threagile, and IriusRisk. Other work [83] focuses on open-source tools which may be used in a variety of fields. Authors of this research proved that the simplicity of modeling allows the user of the tool to obtain meaningful security information in a semi-automatic way. Diagram based modeling in OWASP Threat Dragon The most challenging part related to using the tools for automated threat modeling is related to provisioning of the data related to assets. This is true both for diagram-based tools such as OWASP Threat Dragon presented in Figure 15 and for text-based tools e.g. OWASP Pytm or Threagile. Diagram-based tools provide users an intuitive way of providing initial data. This happens with use of user-friendly interface, designed to draw system diagrams or stencils. Then users are able to apply rules from the library to model potential threats.

The process of diagram drawing is performed manually, based on data gathered during system review. For large (in architecture terms) systems it's an extremely time-consuming task, which if not done properly results in a faulty threat model. Similarly, for text-based tools, users need to provide the data, but instead of graphical representation they are expressed in a structured language, often in form of comments embedded in the source code or additional JSON/XML files. Although a text-based technique does not require the diagram to be manually drawn, it still requires a lot of user attention and work investment to produce a viable threat model. This work is shifted mainly to developers who are asked to include specific annotations in application source code. Commercial products available on the market offer integrations with external data sources, but it is still a fact that threat modeling is largely a manual task [84]. Although recent advances introduce a certain level of automation of threat modeling and threat analysis, they are mostly focused on automated threat elicitation and automated threat prioritization bound with risk assessment. In next chapters, a novel method of including additional sources of data to automate the process of system description is proposed.

4.1. Data Sources for automated threat modeling

In this work a few sources of information helpful in the preparation of model description/diagram were evaluated. As the target was to assess the automation possibilities for threat modeling of 5G related applications deployable in cloud infrastructure, following data sources were selected:

- Kubectl output,
- Docker Compose YAML file,
- DockerFile,
- Nmap output.

If the application deployed in the cloud uses Kubernetes[6] (an open-source system for automating deployment, scaling, and management of containerized applications also known as k8s) it is possible to automatically generate information, useful for threat modeling purposes, running administrator command line tool – kubectl. In case kubectl is not available or k8s is not in use, useful information related to containerized applications can be gathered from Docker Compose files. Docker Compose is a tool designed to help developers in defining and running multi-container Docker applications. The YAML configuration files, utilized by Docker Compose, configure all application's services and allow to start the complete application ecosystem with a single command. YAML file contains basic information about software images used by services, storage systems and networking.


```
version: '3'

services:
  db:
    container_name: db
    image: "mongo:${MONGODB_VERSION}"
    command: "mongod --bind_ip 0.0.0.0 --port 27017"
    networks:
      open5gs:
        aliases:
```


[6] Kubernetes itself is a Greek word meaning pilot, which reflects the system's role in managing and steering containerized applications across clusters. The project was originally developed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF).

```
        - db.open5gs.org
  volumes:
    - db_data:/data/db
    - db_config:/data/configdb
  ports:
    - "0.0.0.0:27017:27017/tcp"

nrf:
  container_name: nrf
  image: "nrf:${OPEN5GS_VERSION}"
  build:
    context: ../../images/nrf
    args:
      - OPEN5GS_VERSION=${OPEN5GS_VERSION}
      - UBUNTU_VERSION=${UBUNTU_VERSION}
  command: "-c /etc/open5gs/custom/nrf.yaml"
  networks:
    open5gs:
      aliases:
        - nrf.open5gs.org
  configs:
    - source: nrf_config
      target: /etc/open5gs/custom/nrf.yaml

ausf:
  container_name: ausf
  image: "ausf:${OPEN5GS_VERSION}"
  build:
    context: ../../images/ausf
    args:
      - OPEN5GS_VERSION=${OPEN5GS_VERSION}
      - UBUNTU_VERSION=${UBUNTU_VERSION}
  command: "-c /etc/open5gs/custom/ausf.yaml"
  networks:
    open5gs:
      aliases:
        - ausf.open5gs.org
  configs:
    - source: ausf_config
      target: /etc/open5gs/custom/ausf.yaml
  depends_on:
    - nrf

udm:
  container_name: udm
  image: "udm:${OPEN5GS_VERSION}"
  build:
    context: ../../images/udm
    args:
      - OPEN5GS_VERSION=${OPEN5GS_VERSION}
      - UBUNTU_VERSION=${UBUNTU_VERSION}
  command: "-c /etc/open5gs/custom/udm.yaml"
  networks:
```

```
    open5gs:
      aliases:
        - udm.open5gs.org
  configs:
    - source: udm_config
      target: /etc/open5gs/custom/udm.yaml
  depends_on:
    - nrf

udr:
  container_name: udr
  image: "udr:${OPEN5GS_VERSION}"
  build:
    context: ../../images/udr
    args:
      - OPEN5GS_VERSION=${OPEN5GS_VERSION}
      - UBUNTU_VERSION=${UBUNTU_VERSION}
  command: "-c /etc/open5gs/custom/udr.yaml"
  restart: unless-stopped
  networks:
    open5gs:
      aliases:
        - udr.open5gs.org
  configs:
    - source: udr_config
      target: /etc/open5gs/custom/udr.yaml
  depends_on:
    - db
    - nrf

nssf:
  container_name: nssf
  image: "nssf:${OPEN5GS_VERSION}"
  build:
    context: ../../images/nssf
    args:
      - OPEN5GS_VERSION=${OPEN5GS_VERSION}
      - UBUNTU_VERSION=${UBUNTU_VERSION}
  command: "-c /etc/open5gs/custom/nssf.yaml"
  networks:
    open5gs:
      aliases:
        - nssf.open5gs.org
  configs:
    - source: nssf_config
      target: /etc/open5gs/custom/nssf.yaml
  depends_on:
    - nrf

bsf:
  container_name: bsf
  image: "bsf:${OPEN5GS_VERSION}"
  build:
```

```
      context: ../../images/bsf
      args:
        - OPEN5GS_VERSION=${OPEN5GS_VERSION}
        - UBUNTU_VERSION=${UBUNTU_VERSION}
    command: "-c /open5gs/config/bsf.yaml"
    networks:
      open5gs:
        aliases:
          - bsf.open5gs.org
    configs:
      - source: bsf_config
        target: /open5gs/config/bsf.yaml
    depends_on:
      - nrf

  pcf:
    container_name: pcf
    image: "pcf:${OPEN5GS_VERSION}"
    build:
      context: ../../images/pcf
      args:
        - OPEN5GS_VERSION=${OPEN5GS_VERSION}
        - UBUNTU_VERSION=${UBUNTU_VERSION}
    command: "-c /etc/open5gs/custom/pcf.yaml"
    restart: unless-stopped
    networks:
      open5gs:
        aliases:
          - pcf.open5gs.org
    configs:
      - source: pcf_config
        target: /etc/open5gs/custom/pcf.yaml
    depends_on:
      - db
      - nrf

  amf:
    container_name: amf
    image: "amf:${OPEN5GS_VERSION}"
    build:
      context: ../../images/amf
      args:
        - OPEN5GS_VERSION=${OPEN5GS_VERSION}
        - UBUNTU_VERSION=${UBUNTU_VERSION}
    command: "-c /etc/open5gs/custom/amf.yaml"
    networks:
      open5gs:
        aliases:
          - amf.open5gs.org
    configs:
      - source: amf_config
        target: /etc/open5gs/custom/amf.yaml
    depends_on:
```

```yaml
      - nrf
    ports:
      - "0.0.0.0:38412:38412/sctp"

  smf:
    container_name: smf
    image: "smf:${OPEN5GS_VERSION}"
    build:
      context: ../../images/smf
      args:
        - OPEN5GS_VERSION=${OPEN5GS_VERSION}
        - UBUNTU_VERSION=${UBUNTU_VERSION}
    command: "-c /etc/open5gs/custom/smf.yaml"
    networks:
      open5gs:
        aliases:
          - smf.open5gs.org
    configs:
      - source: smf_config
        target: /etc/open5gs/custom/smf.yaml
    depends_on:
      - nrf
      - upf

  upf:
    container_name: upf
    image: "upf:${OPEN5GS_VERSION}"
    build:
      context: ../../images/upf
      args:
        - OPEN5GS_VERSION=${OPEN5GS_VERSION}
        - UBUNTU_VERSION=${UBUNTU_VERSION}
    command: "-c /etc/open5gs/custom/upf.yaml"
    networks:
      open5gs:
        aliases:
          - upf.open5gs.org
    extra_hosts:
      docker-host.external-ip: ${DOCKER_HOST_IP}
    configs:
      - source: upf_config
        target: /etc/open5gs/custom/upf.yaml
    ports:
      - "0.0.0.0:2152:2152/udp"
    privileged: true
    cap_add:
      - NET_ADMIN

networks:
  open5gs:
    name: open5gs
    driver: bridge
    driver_opts:
```

```
      com.docker.network.bridge.name: br-ogs
    ipam:
      config:
        - subnet: 10.33.33.0/24

volumes:
  db_data:
    name: open5gs_db_data
    labels:
      org.open5gs.mongodb_version: ${MONGODB_VERSION}
  db_config:
    name: open5gs_db_config
    labels:
      org.open5gs.mongodb_version: ${MONGODB_VERSION}

configs:
  nrf_config:
    file: ../../configs/basic/nrf.yaml
  ausf_config:
    file: ../../configs/basic/ausf.yaml
  udm_config:
    file: ../../configs/basic/udm.yaml
  udr_config:
    file: ../../configs/basic/udr.yaml
  nssf_config:
    file: ../../configs/basic/nssf.yaml
  bsf_config:
    file: ../../configs/basic/bsf.yaml
  pcf_config:
    file: ../../configs/basic/pcf.yaml
  amf_config:
    file: ../../configs/basic/amf.yaml
  smf_config:
    file: ../../configs/basic/smf.yaml
  upf_config:
    file: ../../configs/basic/upf.yaml
```

Code Listing 4: Example Docker Compose YAML file listing.

More details about particular service operating within the application can be found in container DockerFiles. In the above example the configs section specifies path to those docker configuration files. The yaml files are text documents that contain all the parameters a user would normally need to call on the command line to assemble the application or service image. That's why analyzing DockerFiles as part of the threat modeling process brings vast amount of important (from security perspective) information, such as:

- base image name and version,
- additional packages and dependencies,

- list open ports and additional networking information,
- environmental variables with their values,
- information about privileges and user accounts,
- source code and files copied to containers,
- volume mounts,
- container interfaces (services container interacts with),
- security controls or their lack.

Additionally information related to misconfiguration or errors in configuration of dockerfiles (which happen fairly often [85]) can be collected and used.

Table 11: Analyzed data sources for threat modeling

Data source	Asset information						Post deployment	Pre deployment
	Operating system info	Software details	Configuration details	Packages and dependencies	Storage information	Networking and interfaces		
Nmap	●	◌	◌			◌	●	
Kubectl	●	●	◌			◌	●	◌
Docker-compose	●	◌	◌		◌	◌	●	●
DockerFile	●	●	◌	◌	●		●	●

● – fully covered, ◌ - partially covered.

Other source of data analyzed in this research was scanner output from Nmap (Network Mapper)[7] tool used frequently by cybersecurity engineers. Information related to open ports, service names and versions, vulnerabilities, operating system fingerprints, security capabilities (e.g. Web Application Firewall) can be obtained from Nmap. Extra scripts available from NSE (Nmap Scripting Engine[8]) database can enhance this information even further by detecting specific vulnerabilities, misconfigurations, or potential security issues on the target service. Since Nmap usage related to automation of threat modeling and assessment is not novel [86] it is not covered in this work.

Analysis of the data available in listed above sources is presented in Table 11, it includes a comparison of data sources made to evaluate their usefulness for threat modeling purposes.

The "Post deployment" and "Pre deployment" columns represent the capabilities of collecting the data in scenarios where the application/system is not yet deployed. Nmap requires a working instance of the service to scan it, whereas kubectl information could be partially collected even from pods/container images that are not started. Docker configuration files can be accessed at any time and do not require an instantiated workload.

[7] https://nmap.org, accessed: 29.08.2024
[8] NSE is one of the most powerful and flexible features of Nmap. It allows users to write scripts to automate a wide variety of networking tasks, including network discovery, vulnerability detection, security auditing,

4.2. Threat Diagram automation

With use of python, it is possible to create a simple script with diagram creation automation capabilities. It is possible to use if to fully automate the process of generating Threat Dragon diagrams (figure 2) and avoid manual interaction as much as possible. OWASP Threat Dragon can save and load the diagrams from JSON format, schema for that format is documented [87], so it is pretty straightforward to map the Docker Compose YAML and DockerFile fields onto specific elements. Research proved that is is possible to automatically generate diagrams containing:

- Boundaries: trust boundaries indicate where trust levels change. On the data flow diagram, it is a location where data changes its level of trust. Network information defined in DockerFile can be used as main factor to define trust boundaries.
- Flows: data flows representation of how application interacts with external systems and clients and how internal components interact. Exposed ports and interfaces can be mapped to generate at least partial flows on the diagram.
- Processes: containers themselves can be mapped to assets in form of processes on the diagrams.
- Stores: data stores are entities responsible for short- and long-term storage of information. Volumes and database containers can be mapped as stores.

For simple systems, the diagram generation process could be fully automated. However, for complex solutions that included more than 5 Docker containers, automation for threat diagram generation was only partial, requiring some manual work to complete it. Often, the data flows were not identified properly, and in some cases, the script was unable to distinguish processes from stores correctly. Nevertheless, automation still saved a significant amount of time, which was measured using 5 randomly selected small projects and 5 large projects collected from GitHub. Time necessary for creating the diagram manually using the UI interface of Threat Dragon tool and time needed to generate it with automation script was measured. Because the script was not able to generate the diagram without mistakes, the time needed to fix the issues was measured. Results are presented in Table 12.

Table 12: Generating threat diagrams from Docker information

Project Size	**Average time [s]** Manual creation	**Maximum time [s]** Manual creation	**Average time [s]** Automatic creation	**Average fix time [s]** Automatic creation
Small < 5 containers	384	480	1	36
Large >= 5 containers	1860	2700	1	576

The results are not bias free, as while fixing the diagrams (mostly by providing missing flows) basic knowledge about the modeled system was already in place. Therefore, in a real-world scenario this step could require more effort.

Figure 16 shows the example diagram generated with use of automated script is presented. It was based on docker-compose and dockerfiles obtained from docker sample Atsea Sample Shop application available on GitHub [89]. Automatically generated diagram correctly displays all processes, boundaries and database store. Flows needed a manual fix, which was limited to connecting the processes correctly.

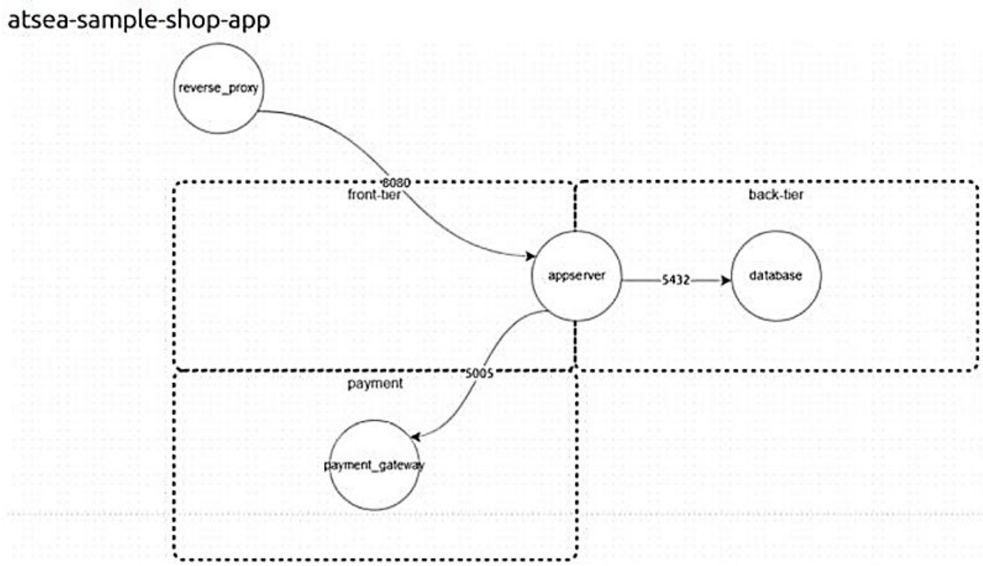

Figure 16: Sample Diagram generated automatically and imported to Threat Dragon tool

As demonstrated above it is possible to enhance the automation of the threat modeling task using simple automation scripts and publicly available solutions. The diagram is only partial input for complete threat model, apart from visualization part also description of threats needs to be generated.

4.3. Threat Description problems

To investigate what is the best method of presenting data related to a particular vulnerability a survey asking cybersecurity experts to grade the vulnerability rated with Common Vulnerability Scoring System (CVSS) system as 8.9 was conducted. Details on the exercise, which was in scope of this research are published in “2022 IEEE International Carnahan Conference on Security Technology (ICCST)” [90]. Aim of the work was to find the most bias free method for presenting descriptions related to security threats.

Using Survey Monkey, a tool that enables a person to develop a survey for use over the internet a short questionnaire containing 6 questions, information about the potential threat of attacking the system, was prepared . All notifications of possible risks were presented in English, but each was created using language with varying degrees of technical complexity. Participants were asked to rate the scale of each threat presented (1 - lowest, 10 - highest). All of the participants did not possess the information that in all of the 6 questions they were in fact grading the same vulnerability, described using a different method. The vulnerability we have used for the purpose of this research was based on the CVE-2021-28242[x], which is an SQL injection type of issue, with a base score of 8.8 assigned according to CVSS.

The following descriptions were used in the survey:

- SQL Injection type of vulnerability.
- Vulnerability with CVSS: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H.
- Vulnerability in user-facing component of PHP application allows remote attackers to obtain sensitive database information by injecting SQL commands.
- Vulnerability with CVSS score of 8.8.
- SQL Injection in the "evoadm.php" component of b2evolution v7.2.2-stable allows remote attackers to obtain sensitive database information by injecting SQL commands into the "cf_name" parameter when creating a new filter under the "Collections" tab.
- Nmap NSE script scan result:

```
PORT STATE SERVICE
80/tcp open http syn-ack
| http-sql-injection:
| Possible sqli for queries:
| http://foo.pl/form/page.php?param=13'%20OR%20sqlspider
| Possible sqli for forms:
| Form at path: /forms/f1.html, form's action: a1/check1.php. Fields that might be vulnerable:
| …
```

All of the above questions were in a random order presented to security experts participating in cybersecurity focused groups on social media (Facebook and LinkedIn).

Depending on the method the vulnerability was presented in, different scores were given by the experts participating in the survey. The CVSS used as a reference was designed to measure technical severity [91] and is used to classify the severity of known and disclosed vulnerabilities. The CVSS standard is widely used by many different government security organizations dealing with risks and threats. The information gathered from CVSS databases is reused by numerous commercial products, including intelligence systems, compliance assessment tools and vulnerability scanners [92], hence its selection as reference value.

From the responses collected from 20 participants the quantitative values were extracted and compared, results are presented in table.

Table 13: Perceived severity of vulnerability survey results.

Vulnerability presented as:	***Language used***	***Averaged response***
SQL Injection type of vulnerability	Basic	7.4
Vulnerability with CVSS: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	Highly technical information	6.8
Vulnerability in user-facing component of PHP application allows remote attackers to obtain sensitive database information by injecting SQL commands.	Technical	8.1
Vulnerability with CVSS score of 8.8	Basic technical	8.2
SQL Injection in the "evoadm.php" component of b2evolution v7.2.2-stable allows remote attackers to obtain sensitive database information by injecting SQL commands into the "cf_name" parameter when creating a new filter under the "Collections" tab.	Detailed technical	7.4

Vulnerability presented as:	***Language used***	***Averaged response***
Nmap NSE script scan result: PORT STATE SERVICE 80/tcp open http syn-ack | http-sql-injection: | Possible sqli for queries: (…)	Highly technical information	5.6

Important conclusion from this survey is that there is a high correlation between the end score with the information detail provided to the expert. The description in the form “Vulnerability in the user facing component of PHP application allows remote attackers to obtain sensitive database information by injecting SQL commands.” received the most adequate score in comparison with CVSS score. When more specific information was added to the description i.e: “SQL Injection in the ‘evoadm.php’ component of b2evolution v7.2.2-stable allows remote attackers to obtain sensitive database information by injecting SQL commands into the cf_name parameter when creating a new filter under the Collections tab” the score was on average 10% lower, which is a significant difference. This most probably comes from the fact that experts were assuming that vulnerability is less severe as the exploitability field is reduced to a certain version of the software, parameter and component availability. In fact, both descriptions are valid for the same vulnerability and should be graded the same way.

Considering all of the above it is safe to assume that AI-based tools interacting with humans or producing text-based reporting, should either use a quantitative approach while preparing severity information or provide descriptions using technical language without providing too much details.

4.4. Describing threats with GPT

As discussed before, the threat identification process is resource absorbing not only because of the dynamic nature of modern applications but also because experts from many fields and experiences should be typically involved in this step to make sure that all important threats are identified. To simply this step, this research proposes using a combination of a network scanner (with a built-in vulnerability detection module) and knowledge base delivered by ChatGPT - an artificial intelligence (AI) chatbot that uses natural language processing to create humanlike responses. The reasoning behind using GPT is that it was already proven that it might be trained on a corpus of AI research articles and used to provide summaries of major findings from different investigations. Several research demonstrates that ChatGPT is useful for tasks like literature searches [93] and reviews [94]. Since it could help researchers find regions of agreement and disagreement among studies and highlight areas that require additional research, it could as well be applied in cybersecurity field to find applicable and not applicable threats related to service of a system and highlight the most important areas. However, it is important to note that some observers have pointed out occasions in which ChatGPT produced convincing responses that contained inaccuracies and misrepresentations [95]. Similar observations were made during this research.

Usage of LLM (Large Language Models) in cyber threat detection was researched by Ferrag [96]. Team responsible for research used two generative elements: SecurityBERT and FalconLLM to show the exceptional potential of LLM in cybersecurity. Solution proposed in [96] SecurityBERT operates as a cyber threat detection mechanism, while FalconLLM is an incident response and recovery system.

The advancement of ChatGPT has been shaped by both theoretical and empirical research. Although empirical studies are more frequently mentioned in literature, the theoretical work has been pivotal in progressing the fields of natural language processing and deep learning. This theoretical foundation includes the creation of mathematical and computational models that allow ChatGPT to mimic human language. Bahrini et al. examined[97] the applications, opportunities, and threats of ChatGPT in 10 main domains, providing detailed examples of potential implementation. They have also conducted an experimental study, checking the effectiveness and comparing the performances of GPT-3.5 and GPT-4. Generative AI usage supporting threat hunting in 6G-enabled IoT networks was researched by Ferrag, Debbah and Al-Hawawreh [98]. Worth noting is the fact that authors recognize that, generative AI systems has tendency to generate false positives. This makes the automated identification of threats difficult because AI can generate alerts when no threat really exists. This can result in useless testing and resource consumption.

This work focuses on comparison of the applicability of pure ChatGPT, STRIDE-GPT and a custom design of pre-prompted ChatGPT technique for identification of threats. Pre-prompting technique was used to minimize the issue related to inability of GPT models to completely comprehend the context and meaning of the text. It was proven for example by Borji [99] that ChatGPT cannot do well in tasks that involve common sense reasoning or logical reasoning that is not covered in the training data. For threat identification process critical thinking, decision making, and problem solving are all critical tasks. They rely significantly on reasoning, which means that lack of awareness and the ability to reason about the relationships between concepts, can be the cause of generating false information. With use of additional information in form of pre-prompt, its possible to make the GPT model more aware of context based on patterns provided.

4.4.1. Data collection

To prove that its possible to use GPT for the above mentioned task, input data was collected from live hosts on the internet. For each of the preselected host Nmap was launched with following parameters:

```
nmap -sV --script vulners ${line} -oA scans/${line}
```

With those parameters the tool does not only perform a basic port scan - a technique used to identify open ports and services available on a network host, but also scans the host for detailed information about the services versions (with '-sV') and, for each identified service performs a query in the Common Vulnerabilities and Exposures (CVE) database (with the –script vulners). CVE database started to collect data in 1999 and every day new weaknesses identified by security experts are being added and at the time of writing 166978 unique vulnerabilities exist in the database. The -oA parameter simply helped to save the data in different formats for further usage.

The collection was performed for 1374 hosts preselected using Shodan, a commonly used tool designed for monitoring and surveillance of IoT devices connected to the internet. Nmap scan reports were used instead of using Shodan provided reports both because Shodan's results are excessively numerous, potentially incomplete, challenging to follow, and require efficient interpretation [19] and because the information needed to be combined with fresh CVE entries. Also, in real life scenario the modelled hosts would be in vast majority running in an internal 5G network, hence Shodan would not have information about them.

Out of the 1374 collected files 860 contained data useful for further analysis. The most common reason for useless files was that the host was not reachable during the scan, or all ports were closed/filtered on it, therefore it was not possible to collect any information from the server. During further analysis it occurred that GPT model has problems parsing large Nmap report output, and we had to remove additional 330 files which were larger than 30kB. Those in vast majority contained scans from honeypots and severely outdated systems for which vulners script was reporting hundreds of vulnerabilities. From the remaining files random selection of 200 reports was made and those were used in further study.

4.4.2. Solution Implementation attempt

The solution with pre-prompt technique used a python script developed for this special purpose.

```
import requests

# Replace 'your_api_key_here' with your actual OpenAI API key.
API_KEY = 'sk-DNI{redacted}yPz'
API_ENDPOINT = 'https://api.openai.com/v1/chat/completions'

# The preprompt is hardcoded and always precedes the variable question.
PREPROMPT = """
Here is an Nmap report, based on it please identify the most probable threats in each of STRIDE categories. Take
into the scope only the services running on the open ports. Do not report threats not related to the services
identified. In case service is unknown/custom port, try to predict what service is running on the open port.
If service has many critical vulnerabilities (score > 8.0) it should be prioritized.
```

For each STRIDE Category define one threat and describe it using one sentence. Also please define which of the threats you have identified should be prioritized to be addressed.

Please provide output in json format, use STRIDE categories and the priority element as keys like this:

```
  "Spoofing": "{identified_threat}",
  "Tampering": "{identified_threat}",
  "Repudiation": "{identified_threat}",
  "Information Disclosure": "{identified_threat}",
  "Denial of Service": "{identified_threat}",
  "Elevation of Privilege": "{identified_threat}",
  "Priority": "{STRIDE_category: identified_threat}"
"""


SCAN = {loaded file}

def ask_chatgpt(question,system):
    headers = {
        "Authorization": f"Bearer {API_KEY}",
        "Content-Type": "application/json",
    }

    payload = {
        "model": "gpt-4-1106-preview
        "messages":[
{
            "role": "system",
            "content": "You are a senior cybersecurity researcher answering with technical language. You understand
how STRIDE threat modeling works."
},
{
            "role": "user",
            "content": f"{PREPROMPT}\n The Nmap report is: {question}"
}],
        "max_tokens": 1500,
    }

    response = requests.post(API_ENDPOINT, json=payload, headers=headers)

    if response.status_code == 200:
.
        response_data = response.json()
        chat_response = response_data['choices'][0]['message']['content'].strip()
        return chat_response
    else:
        return f"Error: {response.status_code}, {response.text}"

user_question = SCAN
response = ask_chatgpt(user_question,"")
```

```
context = "Act as a cybersecurity expert. You know how to apply STRIDE methodology in threat modeling and identification. Try to identify threats based on the input given and on the knowledge an expert level security specialist would posses. In case there is a vulnerability identifier involved (CVE) you can consult the cvedetails database e.g. for CVE-2021-21112 get the details from https://www.cvedetails.com/cve/CVE-2021-21112/"
response = ask_chatgpt(user_question,context)
print("\n\n AI Response: \n", response , "\n")
```

Code Listing 5: Python Script for threat description

OpenAI recommends assigning an identity to the chatbot as a primary step, this modifies the chatbot's responses to resemble those of an individual with the same identity. In the absence of this directive, the chatbot might imitate the user or adopt a snarky tone, which would be unsuitable in the context of providing valuable expert type of answers. This identity can be set by modifying the system role in query as per example below:

```
payload = {
    "model": "gpt-4-1106-preview",
    "messages":[
{
      "role": "system",
      "content": "You are a senior cybersecurity researcher answering with technical language. You understand how STRIDE threat modeling works."
},
{
      "role": "user",
      "content": "Question"
}]
```

Code Listing 6: GPT identity creation

The script was run multiple times asking pre-prompted GPT-4-1106 to deliver output for a selection of 200 reports. Then during research manual check of the threats it identified was conducted. In most cases, the script successfully generated threats related to the actual services mentioned in the scans. Of course, some of the findings were vague and generic but were still applicable and could be considered as meaningful information from a threat modeling point of view.

Only for 15 out of 200 reports LLM delivered threat identifications which were totally not applicable. Some of the examples with explanations:

- LLM output: FTP (port 21) does not provide sufficient logging, allowing malicious activities without traceability.

Comment: FTP service on port 21 will provide by default all logs. LLM did not possess any information about configuration of that service or logging services.

- LLM output: An exploit in a database service could allow an attacker to gain unauthorized administrative access.

  Comment: There was no database service present in the scan. Only port 80,443 and 8080 were open on the target.

- LLM output: FTP does not natively support strong logging mechanisms to prove the occurrence of a transaction.

  Comment: Native support for strong logging mechanisms in FTP seems to be random information LLM generated.

- LLM output: The MySQL service on port 3306 being 'unauthorized' gives an opportunity for attackers to attempt extracting sensitive database information if any security vulnerability.

  Comment: The MySQL 'unauthorized' message in Nmap scan informs that MySQL required authorization when scanner tried accessing it. It’s exactly opposite to what LLM reported.

4.4.3. Comparison with similar tools

When working on the topic, a similar concept called STRIDE-GPT was found. This project is similar to what was designed for the purpose of this study. STRIDE GPT is a threat modeling tool powered by AI, utilizing OpenAI's GPT models to create threat models and attack trees tailored to specific applications using the STRIDE approach. It gives users the possibility to input details about the application, including its type, authentication techniques, and whether it faces the internet or handles sensitive data. Based on this input, the GPT model generates relevant outputs. Main difference between STRIDE-GPT and proposed in research solution is that it requires text-based input from the human, describing the system in question where as the design mentioned in chapter 4.4.2 uses Nmap reports as input, minimizing human interaction.

When comparing the solutions for simplification reasons 4 “experts” were asked to identify only one threat per STRIDE category. Comparison was performed for:

- Chat GPT 4.0 (April 2023 update),
- Pre-prompted GPT 4.0 (gpt-4-1106-preview),
- STRIDE-GPT [15],
- Human with experience in threat modeling.

This study was prepared on a small sample of 10 randomly selected reports. Table 2 provides details about prompts used during exercise.

Table 14: Prompts used for result collection

Security Expert	Prompt
Human	Here is an Nmap report, based on it please identify the most probable threats in each of STRIDE categories. For each STRIDE Category define one threat and describe it in few words. Please define which of the threats you have identified should be prioritized to be addressed. {REPORT}
GPT 4.0	Here is an Nmap report, based on it please identify the most probable threats in each of STRIDE categories. For each STRIDE Category define one threat and describe it using one sentence. Also please define which of the threats you have identified should be prioritized to be addressed. {REPORT}
STRIDE-GPT	Here is an Nmap report, based on it please identify the most probable threats in each of STRIDE categories. For each STRIDE Category, define one threat and describe it using one sentence. Also, please emphasis in bold in the improvement suggestions section the action for one of the threats you have identified that should be prioritized to be addressed. {REPORT}
Pre-prompted GPT-4-1106	Here is an Nmap report, based on it please identify the most probable threats in each of STRIDE categories. For each STRIDE Category define one threat and describe it using one sentence. Also please define which of the threats you have identified should be prioritized to be addressed. {REPORT}

The AI generated responses compared to human provided answers in most of the cases were relatively close. For example, if human identified threat in elevation of privilege category was “Elevation of privileges after gaining access via phpMyAdmin” STRIDE-GPT provided answer: “By leveraging a vulnerability in phpMyAdmin service, an attacker might be able to elevate privileges gaining more control over the system.”. The pre-prompted GPT model answered: “If any known vulnerabilities are present in the version of phpMyAdmin found on the server, an attacker could exploit these to gain elevated privileges.”. Whereas the unmodified GPT-4.0 identified other services as the main threat in elevation of privileges category.

The overall similarity in answers provided by chatbots and human are presented in Figure 17.

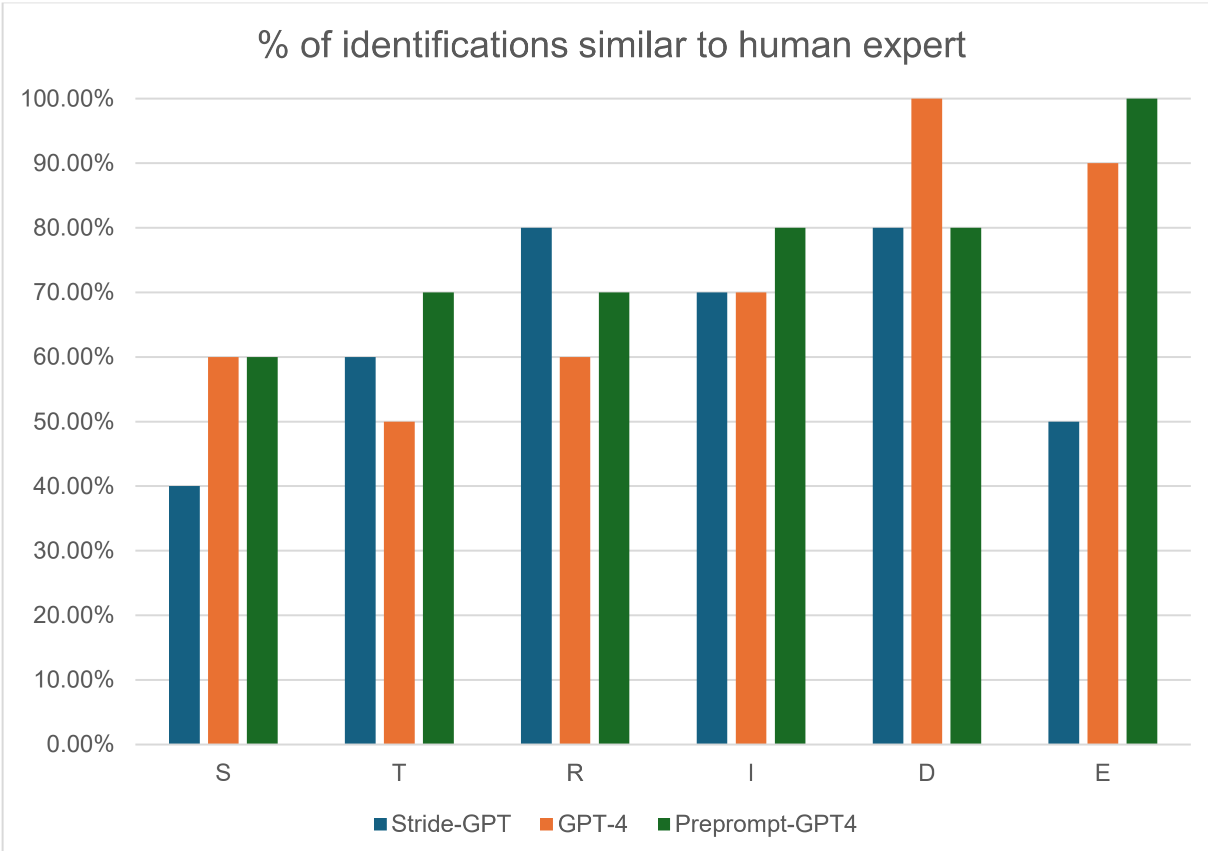

Figure 17: LLM Threat identification similarity to human identification

The results are quite impressive, especially that the complete match would not be expected even when comparing answers provided by 2 human experts. Each expert's background and experiences shape their perception of threats. For example, someone who has dealt with a lot of cyber-attacks targeting web applications may prioritize web server related attacks over attacks on management interfaces such as SSH. Trends in the industry and the way media report on certain types of threats can also influence expert perception, potentially leading to overemphasis on certain types of vulnerabilities.

Table 15: Comparison of threat modeling answers provided by LLMs and human expert

Scan	Answer	
Nmap scan report for [redacted] Host is up (0.055s latency). Not shown: 996 filtered ports PORT STATE SERVICE VERSION 21/tcp open ftp Pure-FTPd 80/tcp open http Apache httpd |_http-server-header: Apache |_https-redirect: ERROR: Script execution failed (use -d to debug) 443/tcp open ssl/http Apache httpd |_http-server-header: Apache 8080/tcp open http Apache httpd |_http-server-header: Apache Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .	**Spoofing:** It could be possible to spoof the requests made towards the ftp **Tampering:** FTP transfers could be tampered by attacker with Man-in-the-Middle position **Repudiation:** Logs tampering related threat **Information Disclosure:** data theft because of unencrypted protocols use **Denial of Service (DoS):** Attacker could launch a DoS attack targeting webserver **Elevation of Privilege:** Attacker could potentially exploit webserver to gain remote code execution (all ports) **Priotrity**: Elevation of Privilege	Human
	Based on the Nmap scan report for [redacted], the following are the most probable threats in each of the STRIDE categories: **Spoofing:** There's a risk of identity spoofing through the FTP service (Pure-FTPd), where an attacker could impersonate a legitimate user to gain unauthorized access. **Tampering:** The open HTTP and HTTPS ports with Apache HTTPD indicate a risk of web content or HTTP/HTTPS requests being tampered with by attackers. **Repudiation:** Without proper logging and monitoring, it could be difficult to track and prove unauthorized actions on web and FTP services, making it challenging to hold perpetrators accountable. **Information Disclosure:** The Apache HTTPD server, if not properly configured or updated, might be vulnerable to exploits that could lead to unauthorized access to sensitive information. **Denial of Service (DoS):** The Apache HTTPD server could be susceptible to DoS attacks, potentially disrupting service availability. **Elevation of Privilege:** Vulnerabilities in the Apache HTTPD server could potentially be exploited to gain unauthorized higher-level privileges. **Priority for Addressing:** The highest priority threats to address would be Information Disclosure and Denial of Service (DoS).	GPT-4.0

	Spoofing: An attacker could leverage the lack of strong authentication to impersonate legitimate users. **Tampering**: An attacker might exploit the open HTTP port to inject or alter the data being transmitted. **Repudiation:** Without proper logging, an attacker can deny malicious activity such as modifying data served on port 80. **Information Disclosure:** An attacker could exploit unencrypted FTP or HTTP traffic to gain access to confidential data being transmitted. **Denial of Service:** An attacker could overwhelm the web server on ports 80, 443, or 8080 with traffic, resulting in a DoS condition. **Elevation of Privilege**: Absence of privileged access management could allow an attacker to exploit vulnerabilities and gain higher-level privileges. **Priority**: Ensure Secure FTP (SFTP) or FTP over SSL (FTPS) is implemented to prevent **information disclosure** threats on the FTP service.	STRIDE-GPT
	"**Spoofing**": "An attacker could exploit vulnerabilities in the Pure-FTPd service to impersonate the FTP server or conduct man-in-the-middle attacks if secure authentication mechanisms are not enforced.", "**Tampering**": "Due to Apache's web server presence, an attacker might exploit any known vulnerabilities to tamper with the web pages or the data served to clients.", "**Repudiation**": "Without proper access logs and audit trails, an attacker could deny the actions performed on the web server, such as unauthorized file uploads or changes.", "**Information Disclosure**": "Unencrypted protocols, misconfigurations or unpatched vulnerabilities in Apache could lead to sensitive information disclosure.", "**Denial of Service**": "Any known critical vulnerabilities in the services running could be exploited to perform a Denial of Service (DoS) attack, potentially making the server unresponsive.", "**Elevation of Privilege**": "Exploiting a vulnerability in Pure-FTPd or Apache could allow an attacker to gain unauthorized elevated privileges on the server.", "**Priority**": "Information Disclosure: Misconfigurations or unpatched vulnerabilities in Apache could lead to sensitive information disclosure, like directory listings or server-side source code."	Pre-prompted GPT-4-1106

Overall Stride-GPT identified 63.33% of the threats in the same way as human, whereas GPT-4 did it with 71.67% similarity and preprompted-GPT4 with 76.67%. The fact that STRIDE-GPT seems to struggle can be attributed to the fact that this tool is designed to work in a different way. It models the threats based on human input, consisting of textual description of the system in question along with other settings which should be selected from drop down lists by the person using the tool.

5. Automated vulnerability detection

In modern networks including 5G core networks, and cloud providers infrastructure the number of hosts is often counted in hundreds of thousands. This number comes not only from the computing power utilized by services delivered by the network but also from the virtualization and containerization of the services operating within the network. Large enterprise environments and cloud infrastructure are often targeted by cyber criminals by attacks which differ in scope, size and technical details varying from DDoS [100] to a complete compromise of infrastructure and data theft or its misuse for launching further stealth attacks [101]. Methods used to prevent cyberattacks, such as security audits, configuration checks are costly both where it comes of time and resources. Some comparison of vulnerability scans were made in the past and shown that there is an significant amount of time needed to perform a vulnerability scan of a single asset. Comparison of performance [102] shown that scanning test systems hosting web application took from 23 minutes to over an hour and a half, depending on the tool which was used. The differences mainly come from the fact that the evaluated scanners varied in the types and number of security checks that were implemented. Researchers have tried to activate the needed plugins (used for performing both active and passive vulnerability checks) to align the scope, but of course identical set of checks could not be set for investigated scanners. Time is not the only resource, computing power and network throughput also need to be considered. And considering that automated vulnerability scanners are not effective in identifying all security flaws and generate certain number of false positive findings, additional resource which is required it the expert knowledge of personnel operating the tools.

5.1. Asset discovery

In 5G networks just like in any other cloud infrastructure systems it might be useful to implement solutions that automatically discover assets that are potentially vulnerable and concentrate the processes granting additional level of prevention (such as vulnerability scanning or audits) on those which were flagged as important. This way resources can be used in a more efficient way and can be assigned, with a priority to the assets which pose a larger risk of being vulnerable to cyberattack. The main problem is that constant scanning every single asset in a network which is hosting thousands, or hundreds of thousands virtualized, or containerized systems/applications makes the process extremely resource hungry. An efficient approach would be to find out at first which of the assets should be prioritized, and assign more resources there, leaving offline assets or assets with low threat probability level to be scanned at later stage, or in a less frequent manner.

$$A \ni (A_v \cup A_l \cup A_o)$$

Equation 5: Assets of a system. A_v - potentially vulnerable assets, A_l- assets with low probability of vulnerability, A_o - offline assets

In this research a basic performance comparison between opensource vulnerability scanner Greeenbone Security Assistant (GSA) and Network Mapper (Nmap), in terms of scan time was conducted. Do note the fact that both of those well known in the industry tools are designed for different purposes. Nmap is a free and open-source utility for asset discovery and basic security auditing. By design Nmap uses raw IP packets in order to discover hosts operating in the network and perform service discovery activities (determine what services or applications are running on the hosts). Nmap is believed to be one of the most popular security tools, mainly because of its portability, ease of use and a powerful Nmap Scripting Engine (NSE) allowing programmers to write their own scripts. Nmap Scripting Engine easily turns this tool into a basic or (depending on the script capabilities) powerful vulnerability scanner.

The performance comparison was focused only on the time and throughput needed to perform scanning of assets, results available in Table 16 show a significant difference both in terms of throughput and time.

Table 16: Nmap and GSA performance comparison

Target	Time Nmap [s]	Time GSA [s]	Throughput Nmap [MB]	Throughput GSA [MB]
Target A	15	840	0.5	36.1
Target B	120	2280	0.7	84.2
Target C	10	720	0.2	19.8

Throughput was measured with use of *iftop* tool and calculated as a sum of send and received traffic exchanged between scanner and target system.

Greenbone Security Assistant scans were launched using “full and fast” scan config.

Nmap scans were launched with -sV and -sC parameters which are commanding the program to run additional probes for determining exact version of service running on the host (-sV) and launching default NSE engine scripts (-sC). The default NSE scripts are non-intrusive scripts that are responsible mostly for detection of common vulnerabilities and misconfigurations.

Differences both in time and throughput needed for completing a task relate to the configuration of 3 different assets, varying in services hosted. Target A was hosting a default Apache http server installation with static web pages and ssh service. Target B was serving a complex php application with multiple webforms, ssh service, DNS server and an ftp server. Target C was hosting ssh and SMB services.

The observations prove that Nmap scan is on average 20 times faster than vulnerability scan with Greenbone scanner. It also significantly smaller amount of traffic in the network. GSA generates up to 100 times more traffic than Nmap. Of course comparing the output of the tools is not the subject of this research, as the aim was to tackle the problem of fast identification of assets that require additional attention from the security perspective. Nmap seems to be a great source of initial information and because of the performance mentioned above and its default NSE scripts which are a great base for threat prediction solution.

5.2. Threat prediction

In this research a concept based on using information obtained from scan reports to filter network workloads based on the probability of being vulnerable, is proposed. An experienced security expert can decide whenever an asset is worth investing resources for further analysis within several minutes, based on the output from Nmap scan. This filtering should ideally happen with use of expert knowledge but taking into account that the solution should be an automated one, usage of an AI/ML system which will filter the reports labelling them according to a binary system (potentially vulnerable or not) is proposed.

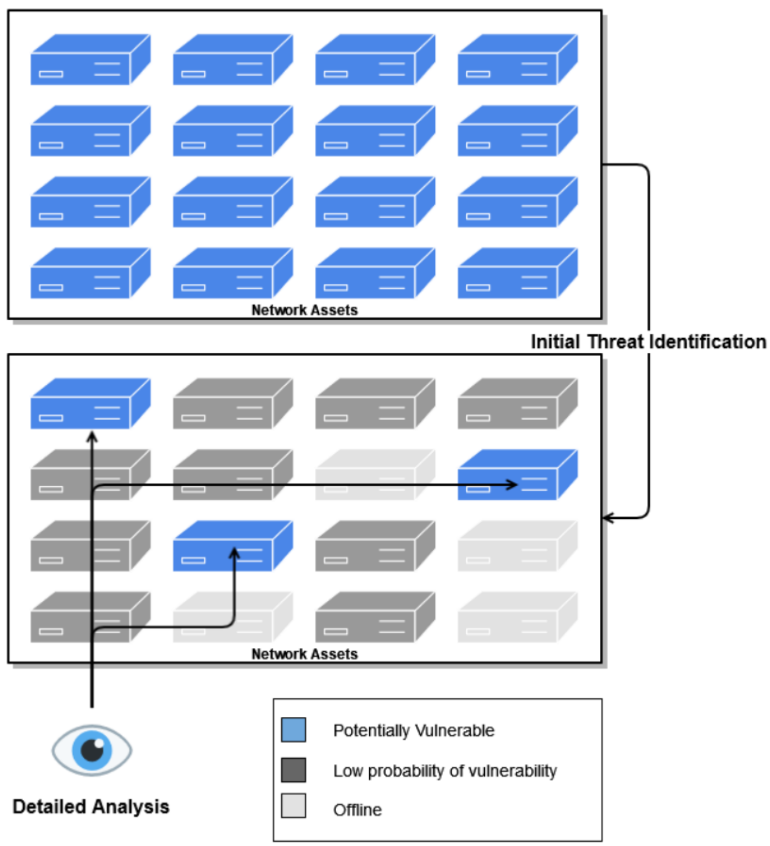

Figure 18: Asset filtering

Proposed solution, depicted in Figure 18, tries to mimic this behaviour by using a pretrained language model based on the transformer architecture. All network elements (services or workloads) can be scanned with Nmap with relatively short amount of time, and then those which are potentially vulnerable can be subjected to additional, detailed auditing.

5.2.1. Training data collection

Data was collected from the hosts operating in internet using a script launching Nmap in a sequential way. Although Nmap is capable of scanning multiple hosts in parallel, by dividing the target IP space into groups and then scanning one group at a time, decision was made to run scan jobs in a sequence instead. This choice made the collection process significantly longer but limited the possibility of scan detection [103] and blocking scanner probes.

```
for ip in {list}
 do
   echo “scanning: ${ip}”
   nmap -sC -sV ${ip} -oA scans/${ip}
 done
```

Code Listing 7: Data collection script

All the collected reports were manually labelled by a security expert with penetration testing experience. The training data was prepared by security experts labelling each of collected report either as potentially vulnerable or not. From the 28437 files obtained in total 2469 were labelled by security experts as potentially vulnerable and 25968 were labelled as not vulnerable. The data was divided into two subsets: larger subset containing 80% of the collected files used for model training and the remaining 20%, smaller subset reserved for model testing purposes.

5.2.1. AI Model for Vulnerable Assets prediction

After initial removal of not relevant meta-data information from the Nmap reports (scan length, date, etc.), token embeddings of the reports using the DistilBERT tokenizer were created. The training set was then used as input for the training of the ML model with the following training parameters:

- learning_rate=2e-5,
- per_device_train_batch_size=1,
- per_device_eval_batch_size=1,
- num_train_epochs=5,

- weight_decay=0.01.

After training for 5 epochs, the model achieved a training loss of 0.066.

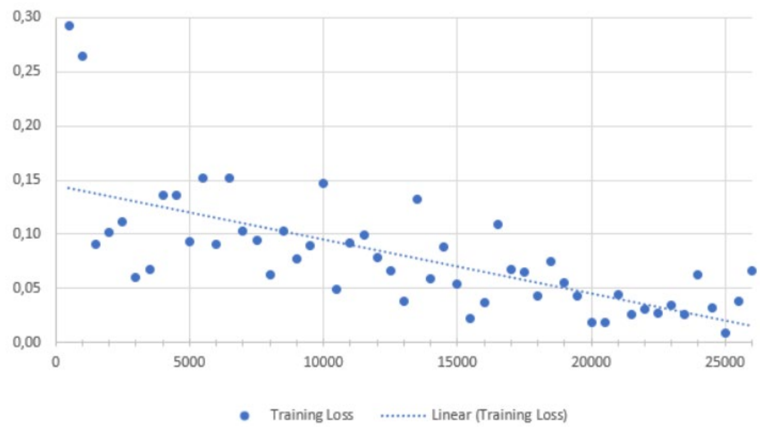

Figure 19: Training Loss in each consecutive training step

The model was then tested on the evaluation set using the General Language Understanding Evaluation (GLUE) and Microsoft Research Paraphrase Corpus (MRPC) benchmarks which are standard benchmarks for analyzing natural language understanding systems [104]. In the end an overall F1-score of 0.99 was achieved.

- 'accuracy': 0.9923195084485407,
- 'f1': 0.9902912621359223.

The model was then exported and run with a totally different data set, obtained from different data sources. It contained 5420 reports, also previously manually labeled as potentially vulnerable or not vulnerable. The details of evaluation are listed in notebook below. It loads the pre-trained PyTorch model, processes and cleans text data, tokenizes it, makes predictions, and evaluates the model's performance using the F1 score.

```
import torch
model = torch.load("model.pth")
import glob, os
def getFileList(folder, ext):
   #os.chdir(folder)
   return glob.glob(folder+"*."+ext)
def loadFiles(filelist):
   files=[]
   for each in filelist:
```

```
        with open(each) as f:
            lines = [line for line in f.readlines()]
        files.append(".".join(lines))
    return files
vulnerable = loadFiles(getFileList("sorted/vulnerable/", "gnmap"))
nonvulnerable = loadFiles(getFileList("sorted/notvulnerable/", "gnmap"))
import re
def clean_text(text):
    text=re.sub(r"^#.*\n|# Nmap done.*$", "", text, flags=re.MULTILINE)
    text = re.sub(r"(?!\b(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\b)[^\w\s\.]", " ", text, flags=re.MULTILINE)
    return text
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
import random
torch.cuda.empty_cache()
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

vulnerable = [clean_text(each) for each in vulnerable]
nonvulnerable = [clean_text(each) for each in nonvulnerable]


vulnerable = list(filter(lambda a: a != ".\n", vulnerable))
nonvulnerable = list(filter(lambda a: a != ".\n", nonvulnerable))

sentences = vulnerable + nonvulnerable

input_ids = [tokenizer.encode(sentence, truncation=True) for sentence in sentences]

# Pad the input sequences to the same length
max_len = max([len(seq) for seq in input_ids])
input_ids = [seq + [0] * (max_len - len(seq)) for seq in input_ids]

# Convert the input sequences to PyTorch tensors
input_ids = torch.tensor(input_ids, dtype=torch.long, device=device)

# Use the BERT model to make predictions
predictions=[]
for each in input_ids:
    with torch.no_grad():
        logits = model(torch.unsqueeze(each,0))[0]
        pred = torch.argmax(logits, dim=1)
        predictions.append(pred.item())
# Print the predictions
print(predictions[0])
v = torch.ones(len(vulnerable))
nv = torch.zeros(len(nonvulnerable))
target  = torch.cat((v,nv))
from torchmetrics.classification import BinaryF1Score
preds = torch.tensor(predictions)
f1 = BinaryF1Score()
print("F1 Score: " +str(f1(preds, target).item()))
```

Code Listing 8: Code used to evaluate the vulnerable asset prediction model

On a completely separate data set the model achieved an F1-Score (harmonic mean of the precision and recall) equal to 0.872 on the test set. Overall, the model was making correct decisions in 87% of the cases which is a satisfying result and proves that AI based system combined with Nmap tool, can be used to identify potentially vulnerable assets with good accuracy.

Research proves that this approach can be used in cloud environments for fast identification of targets which should be prioritized for full vulnerability scan.

5.2.2. AI model for Vulnerability Exploit pattern prediction

A similar model was created and described in detail in “Vulnerability Exploit Pattern Generation and Analysis for Proactive Security Risk Mitigation for 5G Networks” paper [105] which was published during work on this dissertation. It presents a proactive intelligent mechanism to detect possible variants of known vulnerability exploits being attempted on any component of wireless networks. Vulnerability Exploit Pattern Analyzer can prevent possible zero-day attacks by learning from the available known exploits from published databases.

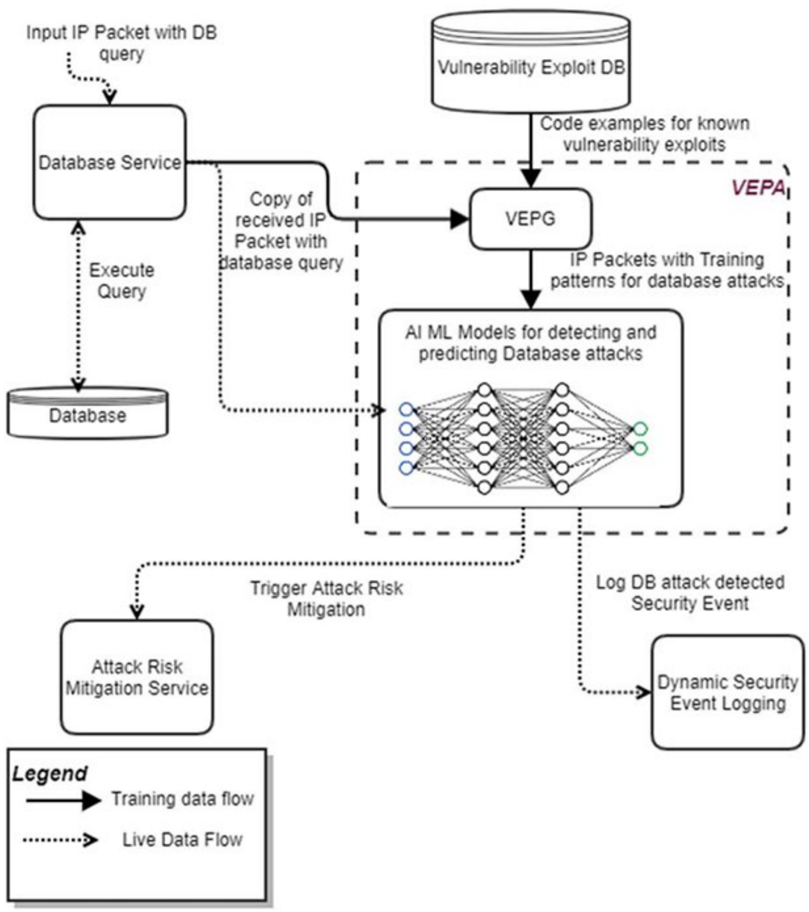

Figure 20: AI/ML based vulnerability exploit pattern generation

Source: Sedkowski et. al, " Vulnerability Exploit Pattern Generation and Analysis for Proactive Security Risk Mitigation for 5G Networks".

Figure 20 depicts an overview of the suggested approach at the system level. In the given example the Vulnerability Exploit Pattern Analyzer[9] (VEPA) is capable of receiving copies of IP packets that

[9] VEPA is a solution currently under development.

are received by database services. It reads various exploits from an Exploit DB and trains AI/ML models using patterns generated by the Vulnerability Exploit Pattern Generator (VEPG).

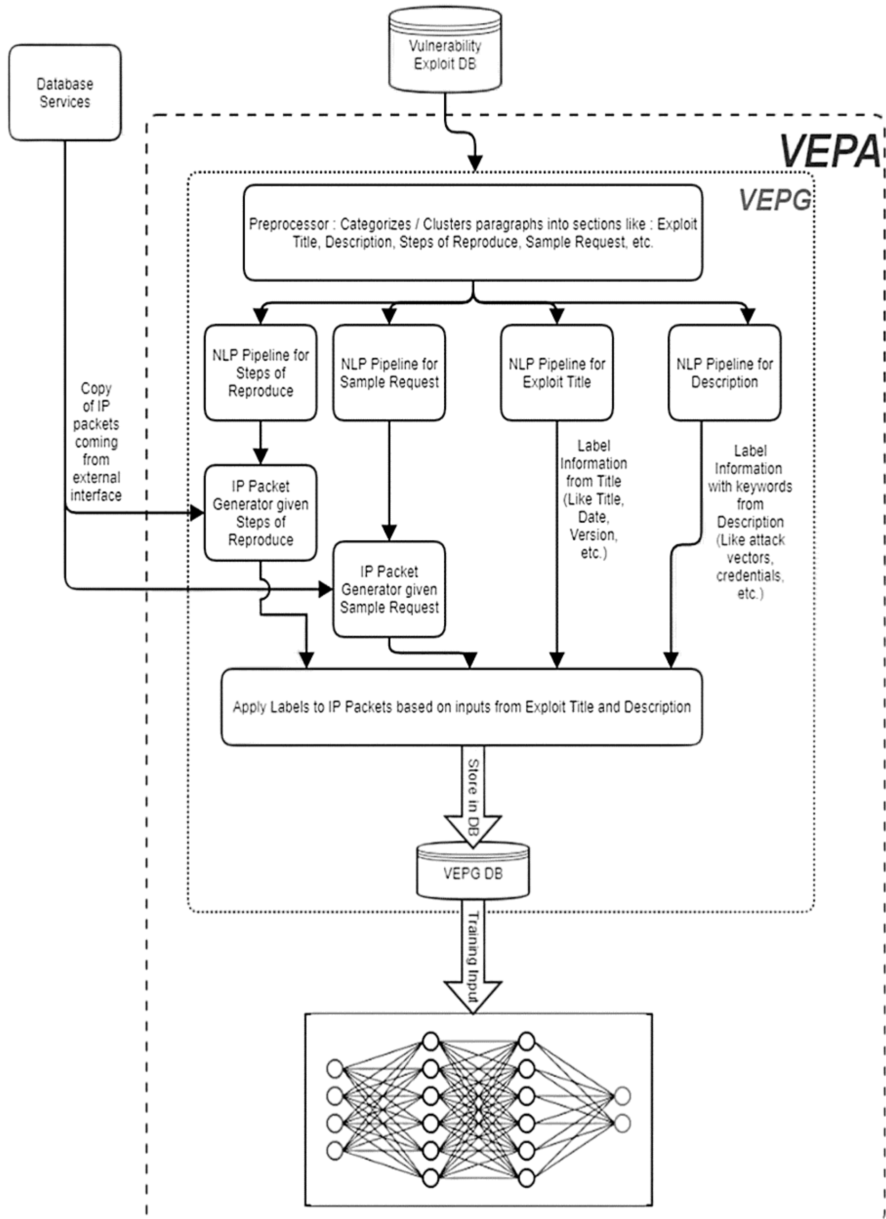

Figure 21: VEPG architecture.

Source: Sedkowski et. al, " Vulnerability Exploit Pattern Generation and Analysis for Proactive Security Risk Mitigation for 5G Networks".

These trained models are then used to detect and predict database-related vulnerability attacks using known exploits. VEPA can also detect or predict possible exploit attempts and activate Attack Risk Mitigation Services (ARMS) to take preventive measures. Additionally, it updates dynamic security event logs with detected or predicted attacks. Database services are being used as an illustrative example in this context. To address exploits associated with operating system and other software vulnerabilities, it is possible to duplicate VEPA concept training the engine for other types of exploits, applying separate instances in other parts of the network.

Training can be conducted in either an online or offline method. VEPA application includes learned models that are relevant to the particular task, it can then be deployed to detect or forecast online attacks. At the centre of VEPA training process lies the VEPG (Vulnerability Exploit Pattern Generator),a component of VEPA that is capable of producing patterns for documented vulnerability exploits. Figure 21 provides a visual representation of the specifics of VEPG. As described in [105], the VEPG can be implemented to read the exploits available, generate patterns with different possible variants of the same exploit, and feed these patterns to an AI/ML model. The patterns can be generated in the form of IP packets. Other embodiments may use other formats like SQL queries, RPC, OR csv formats, etc. Appropriate pre-processing can be implemented, as required. In some embodiments, VEPG can be implemented offline, and the training inputs can be integrated in online system in the form of a DB. VEPG can provide a mix of "good" and "bad" labelled packets to AI/ML models for training. Such mix can be generated using copy of the real world IP packets. AI/ML models get trained to detect such attacks, which are derived from known vulnerabilities. In some other implementations, an offline pre-trained AI/ML model may be integrated in the live system. During run-time, when IP packets are received by the DB Service, it can send a copy to AI/ML models as well. These copied IP packets can be used by AI/ML models to train, as well as derive inferences. Note that this solution works on post-decrypted packets, and hence, the DB VEPA can be integrated along with relevant DB services, or, inside containers where DB is encapsulated. For example, patterns generated by VEPG may be close to some other patterns received by the DB services. AI/ML models can train to detect such patterns as well and identify them as potential attacks. Inferences derived from AI/ML models can be logged in a dynamic security event logging system. Also, attacks detected / predicted by AI/ML models can be fed to an Attack Risk Mitigation Service (ARMS), which can take preventive measures to protect the system from such attackers. Preventive measures like blocking certain IPs at firewall, or, blocking certain services at the input, or, black-listing certain MAC addresses, etc. can be considered.

During research related to VEPA command injection vulnerability attack using exploit for CVE-2020-7209 was simulated [106]. The exploit contains the Exploit Title, Description, and the source code necessary to abuse particular vulnerability or misconfiguration. Basic text parsing was applied to separate these details and extract the code into a python file, which can be executed with many permutations and combinations of command line inputs. In this exploit scenario, the host, port, and cmd were the only command line arguments. The experiment generated several packets by utilizing different combinations of extracted data. 23000 packets were generated, tokenization implemented to produce an attention mask, then classification of the packets as either GOOD (benign – not originating from the exploit) or BAD (attack related to exploit launched) packets was conducted. The labeling process involved the utilization of expert knowledge regarding remote command injections. The labeled packets were utilized to train a TensorFlow model designed for sequence classification. By running training over 3 Epochs witch a training-test split of 80%-20% packets, an accuracy of above 99% was attained.

The experiment detailed in [105] proved that generating packets with different patterns for a given exploit and then training AI models using these generated packets can form AI solutions which can be used for detecting exploits or their variants, targeting vulnerabilities in the real networks.

5.3. Conclusions

This dissertation has presented an overview of topics related to threat modeling and vulnerability detection in modern systems.

Firstly, in Chapter 2, a comprehensive analysis of the 5G network with focus on the novel elements was conducted. New features and network functions were analysed from a security point of view. The increased complexity, virtualization and decentralized infrastructure broadens the attack surface and can create potential vulnerabilities that malicious actors may exploit. Threat analysis conducted during research shows the necessity of implementation of robust security measures at different points of the network, and the need of periodic and automatic threat modeling. Different threats presented throughout the chapter highlight various potential security vulnerabilities and attack vectors within both the Control Plane and the User Plane functions of 5G network.

Chapter 3 presents the state-of-the-art in threat modeling approaches. It forms a systematic review of available tools and methodologies, showing that a combination of those can be successfully used to automate the threat modeling process. Important conclusion is that it is possible to combine the available tools with AI/ML techniques to achieve even further automation. Methods such as STRIDE, VAST, CKK and CVSS can be successfully combined with Supervised Learning, to train AI models and use them for threat analysis and vulnerability detection.

The automation of the threat modeling process is described in Chapter 4, which outlines tools and data sources useful for that task. Research shows that threat diagram generation, and automated analysis based on input gathered from Kubectl, Docker and Nmap, can be performed with minimal human interaction. Measurements performed during research have shown a significant reduction in terms of time needed for threat diagram preparation in automated and manual way. The process was approximately 500% quicker using simple automation scripts and publicly available tools. Apart from diagram preparation, the threat model needs to include a description of the threats in question. During research, a survey was created to find the best (bias free) way descriptions should be generated, the conclusion was that there is a high correlation between the perceived severity of the threat with the information detail provided in the description. Therefore, automated tools should either use a quantitative approach while preparing severity information or provide descriptions using technical language without including too many details. With this in mind, LLM was used to generate descriptions based on data and pre-prompts created for the purpose of the research. The study focused on the comparison of output delivered by 3 different GPT based tools. The achieved result proves that LLMs can be useful for generating descriptions for threat models. Research described in Chapter 4 shows that preprompted-GPT4 76.67% of the time was delivering results similar to those of a human expert.

Chapter 5 concentrates on automated vulnerability detection and shows that AI based solutions achieve remarkable results in this field. Firstly, the model trained for identification of network assets that are potentially vulnerable was making correct decisions in 87% of the cases. This is a satisfying result and proves that AI based system combined with the Nmap tool, can be used to identify potentially vulnerable assets with good accuracy. Secondly, a similar model acting as a proactive intelligent mechanism to detect possible variants of known vulnerability exploits, achieved an accuracy of above 99% in detection of packets containing variants of exploit code.

This dissertation highlights the growing importance of threat modeling and vulnerability detection in securing modern systems, particularly within the context of the evolving 5G network landscape. The research demonstrates that leveraging both traditional methods and advanced AI/ML techniques can significantly enhance the efficiency and effectiveness of security processes. The work highlights the importance of ongoing adjustment and creativity in identifying and addressing potential dangers. Use of automated technologies, in conjunction with artificial intelligence, can decrease the need for human involvement and improve the reaction to new risks. The integration of evolving networks and systems is crucial for sustaining strong security postures.

Bibliography

[1] Ferrag M.,et al., Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes, *Journal of Network and Computer Applications*, Volume 101,2018, pp. 55-82.
[2] Swiderski F., Snyder W., Threat modeling, *Microsoft Press Redmond*, WA, USA, 2004.
[3] Desmet L., Jacobs B., Piessens F., JoosenW., Threat modelling for web services based web applications, *Conference on Communications and Multimedia Security*, 2004, pp. 161–174.
[4] Nokia, Technology Vision 2030. [Online]. Available: https://www.nokia.com/innovation/technology-vision-2030/ Accessed: 8/29/2024.
[5] Sion L., Verreydt S., Yskout K., Threat modeling in Dutch organizations, *KU Leuven*, 2023.
[6] Gartner Research, Forecast Analysis: Information Security and Risk Management Worldwide, *Gartner*, 2024. [Online]. Available: ihttps://www.gartner.com/document/4741831. Accessed: 8/29/2024.
[7] Reinsel D., Gantz J., Ryudning J., The Digitization of the World from Edgt to Core, *IDC White Paper*, Seagate, 2018.
[8] Rao S., Prasad R., The Impact Of 5G Network On Industry 4.0*, Wireless Personal Communications: An International Journal*, Volume 100, Issue 1, 2018, pp. 145 – 159.
[9] Pandav K., et al., Leveraging 5G technology for robotic surgery and cancer care, *Cancer Reports (Hoboken)*. 2022. [Online]. Available: https://doi.org://10.1002/cnr2.1595
[10] Qingling L., Sarfraz S.. Shubin W., An Overview of Key Technologies and Challenges of 6G. 2023. [Online]. Available: https://doi.org/10.1007/978-3-030-62460-6_28
[11] Mahmood A., et al., Factory 5G: A Review of Industry-Centric Features and Deployment Options*, IEEE Industrial Electronics Magazine*, vol. 16, no. 2, 2022, pp. 24-34.
[12] Saliba D., WIFI Integration with LTE in the Roadmap of 5G Networks, Signal and Image Processing*, Ecole nationale supérieure Mines-Télécom Atlantique*, 2020.
[13] Indrasen S., Shubham T., Niraj S., 5G and Beyond Wireless Networks: Technology, Network Deployments, and Materials for Antenna Design, *CRC Press*, 2024.
[14] Tehrani M., Uysal M., Yanikomeroglu H., Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions, *IEEE Communications Magazine*, vol. 52, 2014, pp. 86-92.
[15] Warren D., Dewar C., Understanding 5G: Perspectives on future technological advancements in mobile*, GSMA Intelligence*, 2014.
[16] Fang D., Qian Y., Hu R., 5G Wireless Network Security and Privacy, *Wiley-IEEE Press*, 2023
[17] Mohit P., Leeladhar M., Design, developments, and applications of 5G antennas: A review*, International Journal of Microwave and Wireless Technologies*, vol 15, 2022, pp. 156–182.

[18] GALAZZO R., The History of Cell Phones and their Cellular Generations, 2020. [Online]. Available: https://www.cengn.ca/information-centre/innovation/timeline-from-1g-to-5g-a-brief-history-on-cell-phones/

[19] DING Z., ET AL., A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends, *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, 2017, pp. 2181-2195.

[20] YESMIN T., AGASTI S. CHAKRABARTI K., 5G Security and Privacy Issues: A Perspective View, *ICT with Intelligent Applications*, vol. 1, 2022, pp. 89-98.

[21] LAXMIKANT S., ET AL., Comparative Study of Different Beamforming Techniques for 5G: A Review*, Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems*, vol 145. Springer, Singapore, 2021.

[22] KURVE A., Multi-user MIMO systems: The future in the making, *IEEE Potentials*, vol 28, 2010, pp. 37 - 42.

[23] NOKIA, Dynamic spectrum sharing could be the 5G solution wireless operators are looking for*, Nokia Publications*, 2021. [Online]. Available: https://www.nokia.com/thought-leadership/articles/dynamic-spectrum-sharing-could-be-5g-solution-wireless-operators-looking-for/

[24] SAHA R. & CIOFFI, J.M., Dynamic Spectrum Sharing for 5G NR and 4G LTE Coexistence -A Comprehensive Review, *IEEE Open Journal of the Communications Society*, 2024.

[25] 3GPP. Radio Access Network specification. [Online]. Available: https://www.3gpp.org/ftp/TSG_RAN/TSG_RAN/TSGR_84/Docs/RP-191042.zip

[26] 3GPP. Working Item: Core part: NR DSS. [Online]. Available: https://portal.3gpp.org/desktopmodules/WorkItem/WorkItemDetails.aspx?workitemId=860143

[27] PATIL A., ET AL., A Comprehensive Survey on Spectrum Sharing Techniques for 5G/B5G Intelligent Wireless Networks: Opportunities, Challenges and Future Research Directions. [Online]. Available: https://doi.org/10.48550/arXiv.2211.08956

[28] PARVINI M., Spectrum Sharing Schemes From 4G to 5G and Beyond: Protocol Flow, Regulation, Ecosystem, Economic, *IEEE Open Journal of the Communications Society*, vol 4, 2023, pp. 464-517.

[29] LEI ZHANG, Radio Access Network Slicing and Virtualization for 5G Vertical Industries, *Wiley*, 2021.

[30] AZARIAH, W., ET AL., A Survey on Open Radio Access Networks: Challenges, Research Directions, and Open Source Approaches. *Sensors*, vol 24, 2024, pp 1038-1050.

[31] ZENG J., ET AL., 5G virtualized radio access network approach based on NO Stack framework, *2017 IEEE International Conference on Communications (ICC),* Paris, France, 2017, pp. 1-5,

[32] SAMSUNG, 5G Core Vision, Samsung 5G Core Vol.1, Technical report, 2019.

[33] ABBAS N., ZHANG Y., TAHERKORDI A. , SKEIE T., Mobile Edge Computing: A Survey, *IEEE Internet of Things Journal*, vol. 5, no. 1, 2018, pp. 450-465.

[34] ALNAAS M., LAIAS E., HANASIH A, ALHODAIRY O., Upgrading to 5G Networks: Existing Challenges and Potential Solutions, *International Journal of Computer Sciences and Engineering*, vol 11, 2023, Pp. 5-12.

[35] ETSI ISG, Multi-access Edge Computing (MEC): Framework and Reference Architecture, Specification v 3.1.1, 2022.

[36] NOWAK T., et al., Verticals in 5G MEC use cases and security challenges, *IEEE Access*, vol. 9, 2021, pp. 87251-87298.

[37] MOHAMMADKHAN A., RAMAKRISHNAN K., JAIN V., CleanG—Improving the Architecture and Protocols for Future Cellular Networks With NFV*, IEEE/ACM Transactions on Networking*, vol. 28, no. 6, 2020, pp. 2559-2572.

[38] MALIK S, KIM D., A comparison of RESTful vs. SOAP web services in actuator networks, *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan, Italy, 2017, pp. 753-755.

[39] VILALTA R., ET AL., GRPC-based SDN control and telemetry for soft-failure detection of spectral/spacial superchannels,*45th European Conference on Optical Communication (ECOC 2019)*, Dublin, Ireland, 2019, pp. 1-4.

[40] 3GPP, 5G; System Architecture for the 5G System, 3GPP TS 23.501 version 15.2.0 Release 15, 2016.

[41] GUPTA A., JHA R., A Survey of 5G Network: Architecture and Emerging Technologies, *IEEE Access*, vol. 3, 2015, pp. 1206-1232.

[42] WANG C., ET AL., Cellular architecture and key technologies for 5G wireless communication networks, *IEEE Communications Magazine*, vol. 52, 2014, pp. 122-130.

[43] PETERSON L., SUNAY O., 5G Mobile Networks: A Systems Approach (Synthesis Lectures on Network Systems), *Morgan & Claypool Publishers*, 2020.

[44] ZHANG C., WU B., CHEN Y., TAN M., Research on the Application of Intelligent 5G Network Slicing Technology, *2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT)*, Shenzhen, China, 2023, pp. 298-301.

[45] SANCHEZ L., ET AL., Exploring the Potential of a 5G NSA Network Deployed with Open Source and USRP B210 Equipment, *2023 IEEE Seventh Ecuador Technical Chapters Meeting (ECTM)*, Ambato, Ecuador, 2023, pp. 1-6.

[46] YOU X., ET AL*.,* Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts, *Sci. China Inf. Sci.,* 2021, pp. 64-87.

[47] KHAN R., MCLAUGHLIN K., LAVERTY D., SEZER S., STRIDE-based threat modeling for cyber-physical systems, *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Turin, Italy, 2017, pp 101-106.

[48] RAO S., CHEN H., AURA T., Threat modeling framework for mobile communication systems, *Computers & Security*, vol. 125, 2023.

[49] LI B., ET AL., DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems, *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, 2021, pp. 5615-5624.

[50] CHEN H., RAO S., Adversarial Trends in Mobile Communication Systems: From Attack Patterns to Potential Defenses Strategies, Secure IT Systems*, NordSec*, 2021.

[51] AHMAD I., ET AL., Overview of 5G Security Challenges and Solutions, *IEEE Communications Standards Magazine*, vol. 2, no. 1, 2018, pp. 36-43.

[52] LU Y., ET AL., Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics, *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, 2020, pp. 2134-2143.

[53] KIM ZY., PARK J., LEE J., Security Threats in 5G Edge Computing Environments, *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), 2020, pp. 905-907.

[54] KUMAR T., ET AL., User Privacy, Identity and Trust in 5G, A Comprehensive Guide to 5G Security, *Willey*, 2018, pp. 267-279.

[55] ENISA, 5G Threat landscape for 5G Networks*, European Union Agency for Cybersecurity*, 2020.

[56] SĘDKOWSKI W., BASA Ł., Modelowanie zagrożeń i analiza ryzyka aplikacji, Wstęp do bezpieczeństwa IT, *Securitum*, Kraków, 2023, pp 491-580.

[57] JACOB P., ET AL., An Analytical approach on DFD to UML model transformation techniques, *2016 International Conference on Information Science (ICIS)*, Kochi, India, 2016, pp. 12-17.

[58] THOMMAZO A., MALIMPENSA G., OLIVEIRA T., OLIVATTO G., Requirements Traceability Matrix: Automatic Generation and Visualization, *2012 26th Brazilian Symposium on Software Engineering*, Natal, Brazil, 2012, pp. 101-110.

[59] SOUPPAYA M., SCARFONE K., Guide to Data-Centric System Threat Modeling (NIST Special Publication 800-154). *Gaithersburg: National Institute of Standards and Technology*, 2016.

[60] UCEDA-VELEZ T., MORANA M., Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis, *John Wiley & Sons*, 2015.

[61] NWEKE L., WOLTHUSEN S., A Review of Asset Centric Threat Modelling Approaches, *International Journal of Advanced Computer Science and Applications,* vol. 11, 2020, pp 1-6.

[62] PWC, The global telecom industry seeks growth in the face of rising demands, 2023. [Online]. Available: https://www.pwc.com/gx/en/industries/tmt/telecom-outlook-perspectives.html

[63] SHI W., Analyzing enterprise asset structure and profitability using cloud computing and strategic management accounting, *PLoS ONE*, vol. 16, 2021.

[64] SHOSTACK A., Experiences threat modeling at Microsoft®, MODSEC@MoDELS, 2008.

[65] AKSU M, ET AL., A Quantitative CVSS-Based Cyber Security Risk Assessment Methodology For IT Systems*, International Carnahan Conference on Security Technology*, Madrid, 2017.

[66] FIRST, User Guide. [Online]. Available: https://www.first.org/cvss/v3-1/cvss-v31-user-guide_r1.pdf

[67] SAINI V., DUAN Q., PARUCHURI V., Threat Modeling Using Attack Trees*., Consortium for Computing Sciences in Colleges*, 2008.

[68] HUTCHINS E., CLOPPERT M., AMIN R., Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*, Lockheed Martin Corporation*, 2011.

[69] OZDEMIR M., SOGUKPINAR I., An Android Malware Detection Architecture based on Ensemble Learning., *Transactions on Machine Learning and Artificial Intelligence*, vol. 2, 2014.

[70] ZHANG B., YIN J., WANG S., Research on virus detection technique based on ensemble neural network and SVM*, International Conference on Intelligent Computing*, Huangshan, 2012.

[71] HAIDER N., BAIG M., IMRAN M., Artificial Intelligence and Machine Learning in 5G Network Security: Opportunities, advantages, and future research trends, *Cryptography and Security*, 2020, arXiv:2007.04490.

[72] LIU Y., ET AL., A Secure Federated Learning Framework for 5G Networks, *IEEE Wireless Communications*, vol. 27, no. 4, 2020, pp. 24-31.

[73] MOHAMMED R., VINAYAKUMAR R., SOMAN P., A short review on Applications of Deep learning for Cyber security, *Computer Science*. 2018.

[74] SHAUKAT K., ET AL., Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity, *Energies*, 2020.

[75] LUONG N., ET AL., Applications of Deep Reinforcement Learning in Communications and Networking: A Survey, *IEEE Communications Surveys & Tutorials*, vol. 21, 2019, pp. 3133-3174.

[76] DARWISH A., Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications, *Future Computing and Informatics Journal*, vol 3, 2018.

[77] SCHAAD A., BINDER D., Ml-supported identification and prioritization of threats in the ovvl threat modelling tool, Data and Applications Security and Privacy XXXIV, *Springer International Publishing*, Regensburg, 2020.

[78] AIER S., GLEICHAUF B., SAAT J., WINTER R., Complexity levels of representing dynamics in ea planning, *Advances in Enterprise Engineering III*, pp. 55–69., Springer, Berlin, 2009.

[79] BARANKOVA I., MIKHAILOVA U., AFANASEVA M., Minimizing information security risks based on security threat modeling, *Journal of Physics: Conference Series*, Volume 1441, 2019.

[80] XU D., ET AL., Automated security test generation with formal threat models, *IEEE Trans Dependable Secure Computing*, 2019, pp. 526–540.

[81] VÄLJA M., HEIDING F., FRANKE U., Automating threat modeling using an ontology framework. *Cybersecur 3*, vol 19, 2020.

[82] SHI Z., GRAFF K., STAROBINSKI D., MATYUNIN N., Threat Modeling Tools: A Taxonomy, *IEEE Security & Privacy*, vol. 20, no. 4, 2022, pp. 29-39.

[83] GRANATA D, RAK M., Systematic analysis of automated threat modelling techniques: Comparison of open-source tools, *Software Quality Journal*, vol 23, 2023 pp. 1-37.

[84] BRAITERMAN Z., ET. AL., Threat modeling manifesto. [Online]. Available: http://www.threatmodelingmanifesto.org/ Accessed 26.07.2023.

[85] REIS D., ET AL., Developing Docker and Docker-Compose Specifications: A Developers' Survey, *IEEE Access*, vol. 10, 2022, pp. 2318-2329.

[86] VUGRIN D., ET AL., Cyber threat modeling and validation: port scanning and detection, *Proceedings of the 7th Symposium on Hot Topics in the Science of Security,* Association for Computing Machinery, New York, NY, USA, 2020, pp. 66-72.

[87] OWASP, Threat Model Schema. [Online]. Available: https://owasp.org/www-project-threat dragon/docs-2/schema/ Accessed 26.07.2023

[88] TRUTH S., Create a Threat Model, Security Innovation, 2011. [Online]. Available: https://blog.securityinnovation.com/blog/2011/02/create-a-threat-model-step-1.html Accessed: 27.07.2023

[89] DOCKER SAMPLES. [Online] Available: https://github.com/dockersamples/atsea-sample-shop-app Accessed 27.07.2023

[90] SĘDKOWSKI W. BIERCZYŃSKI K., Perceived severity of vulnerability in cybersecurity: cross linguistic variegation, *IEEE International Carnahan Conference on Security Technology (ICCST)*, Valeč u Hrotovic, Czech Republic, 2022.
[91] SPRING J., Towards improving CVSS, *Software Engineering Institute Carnegie Mellon University*, 2018.
[92] JOHNSON P., Can the common vulnerability scoring system be trusted? A Bayesian analysis, *IEEE Trans. Depend. Secur. Comput.,* 2017
[93] BISWAS S., ChatGPT and the future of medical writing, Radiology, vol. 307, 2023.
[94] DOWLING M., LUCEY B., ChatGPT for (Finance) research: the bananarama conjecture, *Finance Research Letters*, Volume 53, 2023.
[95] FARHAT F., SOHAIL S., MADSEN D., How trustworthy is ChatGPT? The case of bibliometric analyses, *Cogent Engeniering*, Vol 10, 2023.
[96] FERRAG M., ET AL., Revolutionizing Cyber Threat Detection with Large Language Models, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2306.14263 Accessed 29.08.2024.
[97] BAHRINI A., ET AL., ChatGPT: Applications, Opportunities, and Threats*, Systems and Information Engineering Design Symposium (SIEDS),* Charlottesville, VA, USA, 2023, pp. 274-279.
[98] FERRAG M., DEBBAH M., AL-HAWAWREH M., Generative AI for Cyber Threat-Hunting in 6G-enabled IoT Networks, *IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)*, Bangalore, India, 2023, pp. 16-25.
[99] BORJI A., A Categorical Archive of ChatGPT Failures, Preprint, 2023. [Online]. Available: https://doi.org/10.21203/rs.3.rs-2895792/v1 Accessed 29.08.2024.
[100] SMITH-PERRONE J., SIMS J., Securing cloud, SDN and large data network environments from emerging DDoS attacks, *International Conference on Cloud Computing, Data Science & Engineering - Confluenc*e, 2017, pp. 466-469.
[101] CHATTERJEE M., ET AL., Cloud: A Platform to Launch Stealth Attacks, *IEEE Annual Computers, Software, and Applications Conference (COMPSAC*), 2020, pp. 1558-1563.
[102] MANSOUR A., ET AL., Performance-based comparative assessment of open source web vulnerability scanners., *Security and Communication Networks*, 2017.
[103] LIAO S., ET AL., A Comprehensive Detection Approach of Nmap: Principles, Rules and Experiments, *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2020, pp. 64-71.
[104] DOLAN W., BROCKETT C., Automatically Constructing a Corpus of Sentential Paraphrases, *International Joint Conference on Natural Language Processing*, 2005.
[105] SĘDKOWSKI W., BHATT R. P., FERNANDES C. , RANGANATH K., Vulnerability Exploit Pattern Generation and Analysis for proactive security risk mitigation for 5G networks*, IEEE Conference on Computer Communications Workshops*, Hoboken, NJ, USA, 2023.
[106] NIST, CVE-2020-7209 Details. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2020-7209 Accessed 29.08.2024.

Lists of Tables and Figures

List of Tables

List of Figures