Penetration Testing of 5G Core Network Web Technologies

Filippo Giambartolomei
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate-Mondragón, Spain
fgiambartolomei@ikerlan.es

Marc Barceló
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate-Mondragón, Spain
mbarcelo@ikerlan.es

Alessandro Brighente
Deparments of Mathematics
University of Padua
Padua, Italy
alessandro.brighente@unipd.it

Aitor Urbieta
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate-Mondragón, Spain
aurbieta@ikerlan.es

Mauro Conti
Deparments of Mathematics
University of Padua
Padua, Italy
mauro.conti@unipd.it

arXiv:2403.01871v1 [cs.CR] 4 Mar 2024

Abstract—**Thanks to technologies such as virtual network function the Fifth Generation (5G) of mobile networks dynamically allocate resources to different types of users in an on-demand fashion. Virtualization extends up to the 5G core, where software-defined networks and network slicing implement a customizable environment. These technologies can be controlled via application programming interfaces and web technologies, inheriting hence their security risks and settings. An attacker exploiting vulnerable implementations of the 5G core may gain privileged control of the network assets and disrupt its availability. However, there is currently no security assessment of the web security of the 5G core network.**

In this paper, we present the first security assessment of the 5G core from a web security perspective. We use the STRIDE threat modeling approach to define a complete list of possible threat vectors and associated attacks. Thanks to a suite of security testing tools, we cover all of these threats and test the security of the 5G core. In particular, we test the three most relevant open-source 5G core implementations, i.e., Open5GS, Free5Gc, and OpenAirInterface. Our analysis shows that all these cores are vulnerable to at least two of our identified attack vectors, demanding increased security measures in the development of future 5G core networks.

Index Terms—**5G network, 5G Core, network virtualization function, Penetration testing.**

I. INTRODUCTION

The 5G core is the heart of a 5G network. Its main roles include providing end users with secure and reliable connection and access to a set of services. Although the network core's role did not change with respect to previous generations, its implementation has been completely revolutionized. Indeed, all 5G core network functions are completely software-based and designed as cloud-native [1]. These changes were necessary to accommodate the increasing number of connected users and their different requirements (based on traffic type). Indeed, the performance that 5G networks are expected to meet, includes a remarkable data rate requirement of 20Gbps for download and 10Gbps for upload. This represents a substantial leap forward when compared to the capabilities of 4G/LTE Release 14, which provides a maximum data rate of 1Gbps for downlink and 50Mbps for uplink [2]. 2019 marked the beginning of the deployment and spreading of the fifth generation network, more commonly known as the 5G network [3].

The development of a new generation is not solely driven by the increased performance demand for communications, but also by an increased need for security and privacy guarantees. Indeed, the 5G core is responsible for establishing secure connections and providing basic services such as authentication and authorization [4]. However, although security improved compared to the previous generations, there still exist security and privacy gaps that may jeopardize 5G networks [5]. Recent examples showed that it is possible to create fake base station attacks [6], or to tamper with the management of the identifiers [7]. Therefore, a thorough assessment of the security of 5G core network implementation is a fundamental need to develop secure networks.

Despite the importance of the issue, the literature on this topic is still scarce, providing only a few examples of the analysis of the 5G core security. Holtrup et al. [8] provide a formal analysis of the security of the 5G network, considering both stand-alone and non-stand-alone implementations. However, the authors do not follow a standardized methodology for security assessment. Pell et al. present a more specific approach bearing in mind the MITRE Threat modeling framework [9]. However, such an approach is not sufficient to clearly understand the broader class of attack types and it is particularly dependent on the system implementation and/or version. Altariqi et al. [10] consider the security of the 5G core, however focusing only on its function virtualization capabilities.

In this paper, we present the first security assessment of the 5G core from a web security perspective. We leverage STRIDE [11], a well-established threat modeling methodology developed at Microsoft which is one of the de facto standard

approaches. While drawing inspiration from Altariqi et al. work [10], our paper takes a different approach by focusing on the web components of the 5G core. To assess the security of the 5G core network, we test our model on three public and well-established 5G core implementations, namely Open5GS[1], Free5GC[2], and OpenAirInterface[3]. We leverage a suite of penetration testing tools to implement a large set of attacks and identify a relevant number of attacks that can be performed on these 5G core implementations.

The contributions of our paper can be summarized as follows:

- We propose the first web-based threat model for the 5G core. We leverage the STRIDE methodology, a well-known approach to systematically derive all possible threat vectors affecting the 5G core. For each threat, we discuss how a successful attack could impact the 5G network.
- We test the web security of multiple well-known implementations of the 5G core. Being these implementations compliant with the 5G standard and used by many companies and research institutions, they represent what is closest to a real 5G core implementation. Via our testing, we point out that all of these cores are vulnerable to at least two attacks, thus requiring cautious deployment and further security measures.

The rest of the paper is organized as follows. We introduce the system and threat models in Section II. In Section III, we present the set of 5G core implementations that we tested and the set of security tools used to cover all the threats we identified in Section II. We then present the results. In Section IV we present possible countermeasures to the identified attacks and then derive the conclusions in Section V.

II. SYSTEM AND THREAT MODEL

We provide a description of the considered system and threat model. In Section II-A, we present the components of the 5G core network focusing on their use of web-based technologies. We then present in Section II-B the STRIDE-based threat model for the 5G core.

A. System Model

A native 5G core implements a service-based architecture thanks to which it is possible to independently scale different functions [12]. Evolved Packet Core (EPC) functions are instead based on a point-to-point architecture and have overlapping responsibilities. The integration of new functions happens via a message bus for control plane interactions. It implements (REST)-ful application programming interfaces (APIs) that use hyper-text transport protocol (HTTP/2) inquiries for high-level flexibility [13]. Compared to previous generations, 5G implements a totally virtualized environment, enhancing the portability and usability of networking systems

and services. In particular, the 5G core extensively uses virtualization techniques to implement Virtual Network Functions (VNF), network slices, and software-defined networking to implement a scalable and customizable network of services.

Fig. 1 shows the standard implementation of the 5G core [1]. The major components are:

- Access and Mobility Management Function (AMF): responsible for access control and mobility, and includes Network Slice Selection Function (NSSF).
- Session Management Function (SMF): manages sessions according to network policies.
- User Plane Function (UPF): responsible for packet-level operations, such as packet routing, forwarding, and inspection.
- Policy Control Function (PCF): provides a policy framework for network slicing, roaming, and mobility management.
- Unified Data Management (UDM): stores subscriber data and profiles.
- Network Function Responsibility Function (NRF): provide registration and discovery functionalities such that network functions can communicate with one another via APIs.
- Network Exposure Function (NEF): API gateway that allows external users the ability to monitor, provision, and enforce application policies for users inside the operator network.
- Authentication Server Function (AUSF): authentication server.

All these components are connected via standard interfaces (e.g., N1 and N2) defined by 3GPP. To manage and control network functions at different levels, both operators and services use API calls with a request-response mechanism passing via *reference points*. Each of these reference points can be enforced with the use of Service Based Interfaces (SBI) whose usage is based on a logically shared framework between functions. One meaningful consideration that must be done is that SBIs are mainly used inside the Control Plane, indeed, the User Data Plane does not present SBI, but it only owns the reference points.

This huge increment of virtualization within the 5G core infrastructure engenders an urgent requirement for a graphical user interface (GUI). This imperative GUI serves as an essential tool that facilitates administrators in the seamless and rapid management of subscriber connections. Through this graphical interface, administrators have the chance to oversee, control, and optimize the connectivity of subscribers, thereby enhancing the overall efficiency and efficacy of the 5G core network. Furthermore, the network requires storage capabilities to account for the different configurations and data at rest about users and operators. All these functionalities can be enabled via web-based technologies.

B. Threat Model

We consider a powerful attacker that has a large amount of computation power and resources to try accessing the 5G
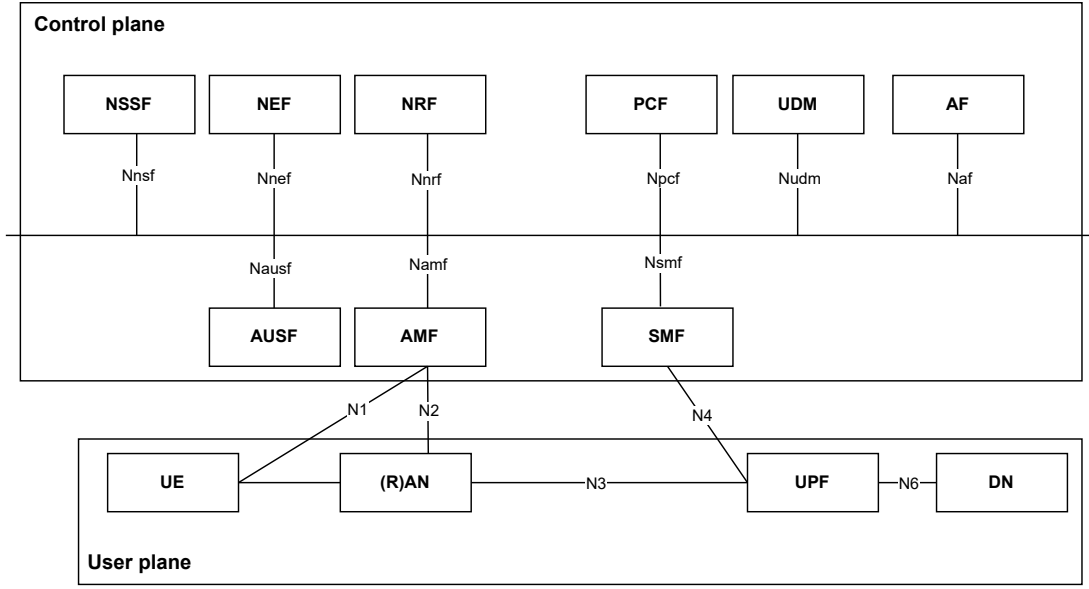
Fig. 1: Overview of the 5G Core architecture.

core. We put no restriction on the attacker's capabilities to expose a large set of possible threats. We leverage the STRIDE threat modeling approach [11] to identify vulnerabilities in the 5G core. STRIDE is an acronym that encompasses the main threats that might affect the system as visible in Tab. I, i.e.,

- **Spoofing:** a malicious user impersonates a legitimate user, deceiving the receiver that the communication is lawful.
- **Tampering:** a baleful user alters or modifies legitimate data such that they are harmful to the receiver. Commonly paired with the spoofing technique.
- **Repudiation:** A malicious actor cannot be unambiguously linked to an action/transmission that happened.
- **Information disclosure:** it is the ability to gather sensitive system information due to unprotected data flows.
- **Denial of service:** a baleful user disrupts resource availability by overwhelming a communication link with packets/requests.
- **Elevation of privilege:** the attacker gains access to information that a regular user should not be authorized to access.

Tab. I provides a list of possible attacks, a list of possible attack implementations, and the attacker's goals for each of the threats identified by STRIDE. We assume the attacker uses these attacks to find and exploit vulnerabilities in the 5G core. In the following, we test the different 5G core implementations encompassing all the threats listed by the STRIDE methodology. We note that, although not providing a full list of all possible attacks, our identified attacks cover the OWASP top 10 web application security risks [14].

III. TESTS IMPLEMENTATION AND RESULTS

A. Chosen 5G Core

Our objective of studying the web security of the 5G core requires a proper implementation of the overall standard defined by the 3GPP [1]. However, implementing it from scratch does not provide significant insights into currently available deployments and solutions. To assess whether state-of-the-art implementations of the 5G are vulnerable to the threats identified via the STRIDE methodology from a web-based standpoint, we refer to the Open5GS, Free5gc, and OpenAirInterface implementations. In other studies related to threats associated with the 5G core, Neto et al. used a similar setting, considering Magma rather than OpenAirInterface [15]. Moreover, at the current time, these three parsed cores are the most cutting-edge in the marketplace, used both in research and industry for research and development [16]–[20].

B. Tools

To test all the threats in STRIDE, we leverage a set of well-known and established security tools. We choose them considering the performance they provide and the results that can be obtained using them. In the following, we provide a brief description of the tools and their main scope:

Nmap. This tool is mainly employed to gather information about a specific network or host. Nmap, indeed, allows the user to scan one or more targets simultaneously, to obtain as much information as possible regarding services, ports, hosts, and vulnerabilities.

WhatWeb. Its purpose is primarily to identify all the technologies and features the system presents from a web perspective. It can identify entities, packages, and servers in a way similar to what nmap does for networks.

Nikto. This web application scanner is mainly employed to get basic information about servers, checking for outdated

Threat	Possible attacks	Possible implementations	Attacker's goal
Spoofing	Bruteforce attack, Dictionary attack.	Exploits the absence of/weak encryption, to hijack the traffic and modify it according to necessity, leading to identity theft or MiTM attack.	Intercept the traffic with the purpose of impersonation, sharing false information or obtaining sensitive information.
Tampering	Bruteforce attack, Dictionary attack, SQL injection, NoSQL injection.	Usually coupled with the spoofing technique, allows to modify the data traffic to deceive the user or obtain useful information like user credentials.	Make the victim unaware that the data traffic has been modified.
Repudiation	JWT Robustness, Clickjacking.	Possibility to modify content pretending to be another user, i.e., Identity Theft.	Falsify data, and alter transmission, with the intent of deceiving or defrauding others without being caught.
Information disclosure	Permission leakage, SQL injection, NoSQL injection, Directory traversal.	The malicious actor may gain unauthorized access to a list of usernames, passwords, and even data configurations.	Acquire, alter, and utilize data to which the attacker should not have legitimate access.
Denial of service	DoS and DDoS attack.	As an illustrative instance, a malicious actor could render the Access and Mobility Management Function (AMF) unavailable, thereby making unsuitable the users' administration.	Capability to render a service inoperable, causing it to crash, ceasing functioning, or potentially slowing down its operational efficiency.
Elevation of privileges	JWT Robustness, Bruteforce attack, Dictionary attack.	The potential to assume the identity of an administrator, to gain more privileges and thereby undertaking actions beyond those initially authorized.	Seeking to elevate his privileges, thereby enabling him to perform actions that were, until that moment, beyond his authorized capabilities.

TABLE I: Potential threats for STRIDE modeling.

system versions, unusual headers, and everything that can be exploited against the target.

Dirbuster and Dirb. Both the mentioned web content scanners are employed to brute force a specific web page, to discover hidden directories and hidden files. The attacker can use these tools to identify all the directories available to the web architecture.

Burpsuite. This penetration testing tool is one of the most employed by the community. It provides a large variety of techniques implementable for several attacks.

Nessus. This tool is a powerful vulnerability scanner. This allows to inspect the objective from a general perspective but also delving into its details providing specific scan templates, focusing only on specific ports, services or login pages.

Hping3. It is mainly employed for the creation of ICMP, TCP and UDP packets. This network tool has the purpose of crafting, following some specifications, the most effective packet; such as a DoS attack.

SlowLoris. Also known as Low bandwidth DoS tool allows to exploit a Denial of Service (DoS) against a targeted web server. This tool employs the "low and slow" technique, opening connections with the server and keeping them open.

Hulk. HTTP Unbearable Load King DDoS is a tool used to assemble and deliver unique pattern packets.

JohnTheRipper. This tool has the leading purpose of cracking passwords.

C. Results

Permissions leakage, SQL and NoSQL injections. Open5gs and OpenAirInterface are both affected by this exposure, even if in a distinct fashion. The first, as a result, allows an attacker to remotely access the MongoDB core's database without any kind of authentication, granting to look, add, modify, and even remove accounts. While, for the second core, the approach was a little disparate, especially since OpenAirInterface does not own an unstructured database. Therefore, at implementation time, the latter holds two users: the "root" which possesses all the privileges, and the "user" which owns the same permissions as the administrator, and it should not. Even if accessing a different kind of DB, the weakness exploitation is pretty much the same as the Open5gs. In both cases, the outflow of privileges leads to injections: whether SQL or NoSQL.

In addition, in both cases, we discovered the Principle of Least Privilege [21] (PoLP) was not respected, and neither was the absence of user input sanitization. In the matter of Free5gc, this core was not vulnerable to information leakage and therefore to injection of any type, mainly because it does not enable external users to remotely access its database.

Dictionary and Bruteforce attacks. These attacks cannot be tested in the OpenAirInterface because of the absence of the web interface, and therefore these results as not implementable. In such a way, as visible from the table in Tab. II, this offense has succeeded for the Open5gs, whilst it did not succeed for the Free5gc. The latter indeed, presents hard-coded credentials, preserving the login details inside a stored file. Even if this is not a good practice, it did not allow to brute force the credentials. Moreover, it possesses NIA2: an integrity protection algorithm that does not allow users to read or modify the content of the packets without a specific key.

DoS and DDoS. In the matter of Open5gs, this Core showed a little delay during the opening of the page, but nothing remarkable intended to be considered as vulnerability. Similarly, OpenAirInterface did not seem to be affected by this kind of attack, it indeed kept regularly working, even if for this specific instance the attack has been implemented against some of its containers. Nevertheless, Free5gc appeared exposed unlike the other two. When testing the latter, it revealed a significant delay during the opening of the login page. This isolated scenario turns out to be vulnerable to DDoS attack, exploited using the `Hping3` tool in flooding mode, randomizing the source IPs of each request. Although the web server did not crush, this offense may be deemed as effective also considering a large-scale uptake.

Directory traversal. For this particular attack, only Free5gc was vulnerable, indeed, the examined folder discovered without knowing its specific path was `/etc/passwd` enabling in this circumstance the leakage of a large amount of critical information that can be exploited by malicious actors.

Clickjacking. About OpenAirInterface, here the attack is not feasible, due to the absence of a web interface. Clickjacking seemed to be a feasible attack, taking into account that either Open5gs and Free5gc enable `IFRAME`. This was not sufficient, indeed, we did not reach the expected result, mainly due to the presence of the Cross-Origin Resource Sharing policy (CORS) which forestalls to access web resources through different sources than the ones employed by the website.

JSON Web Token. Open5gs turned out to be vulnerable to this attack because it uses a known passphrase. Therefore, once obtained the key, it is possible to change the content of the JWT's payload and then recreate the proper signature through the previously obtained key, making it valid. Regarding Free5gc, it does not rely on cookies, or hard-coding the credentials, thus it does not save the logged user's session. Whilst, for the OpenAirInterface, this category of offense is not even applicable, because of the absence of a web interface.

Table II presents a summary of the attacks tested against the different 5G core implementations, together with the effectiveness outcome. We notice that not all cores are vulnerable to the same attacks, and for some of them, attacks were not implementable at all.

Attacks	Open5Gs	Free5Gc	OpenAirInterface
Implementable attacks	9	9	4
Successful attacks	5	2	2
Database permission leakage	yes	no	yes
SQL injection	no	no	yes
NoSQL injection	yes	no	no
Dictionary attack	yes	no	not implementable
Bruteforce attack	yes	no	not implementable
DoS and DDoS	no	yes	no
Directory traversal	no	yes	not implementable
Clickjacking	no	no	not implementable
Json Web Token robustness	yes	no	not implementable

TABLE II: Comparison Table between tested attacks over the three mentioned 5G Core.

IV. COUNTERMEASURES

This section involves all the possible expedients and countermeasures to avoid the attacks formerly mentioned and exploited. The presented precautions are some of the most commonly employed, nevertheless not unique. Starting from the former section introduced earlier, the following list presents some possible tricks to avert *Permission leakage, SQL and NoSQL injection:*

- The first method is to apply the principle of least privilege (PoLP) in which each user possesses the minimum levels of privileges.
- The second regards the compulsoriness of the authentication for remote connections.
- The third approach, instead, is inherent to the input sanitization: which means removing unsolicited characters from the input, before processing them.

The concept that stands behind each of these tricks is to not supply more permissions to users than the ones needed.

Taking a step forward, pursuing with the *Dictionary and Bruteforce attacks*, the best way to prevent both, as visible from the failure achieved with Free5gc, is to attach some encryption algorithm. In such a way, the packets can be captured but not read or modified.

In addition to these scenarios, the tested *DoS and DDoS* exhibit several countermeasures even if there is no assurance of being 100% safe:

- The firewall is the first methodology that can be applied with its editable rules.
- Intrusion Detection System (IDS) is another significant system to avert DoS or DDoS, primarily employed to monitor the network, trying to understand if an attack is ongoing.
- Introducing a connection timeout or even limiting incoming data rate (SlowLoris).
- Extending the backlog of pending connections (SlowLoris).

Now, moving on to the fourth attack, known as *Directory Traversal*. This vulnerability hinges on two crucial concepts

that should be considered to prevent exploitation. It's important to note that Free5gc, unfortunately, lacks of input sanitization, as evidenced by the exploit ../../. Therefore, it is essential to thoroughly analyze and carefully process any input characters to safeguard against this vulnerability. [22]. Moreover, another significant expedient that we can employ to avoid the Directory traversal is the already mentioned PoLP.

Concerning the *Clickjacking* attack, we will not delve too much into details mainly because this attempt wasn't successful and because it was already presented with an effective countermeasure such as the CORS policy.

Taking a step forward to the last selected attack, there are two main possibilities to avoid its misuse. To avert the exploitation of the *Json Web Token*, one possible way is to avail of an uncommon password; the latter indeed, should not be present in any of the known wordlists. Moreover, another potential countermeasure should be to decrease the IAT (issued at) that identifies how long the JWT is alive. The concept is to employ an adequate amount of "time to live" based on the usage of the token.

V. CONCLUSIONS

Form Table II) we observe that among the proven attacks, there were no offenses that affected every Core in the same manner. This mainly arises from the fact that, even if considering the same standard core, every single implementation exhibits a small degree of changes that characterize the structure it deploys. The purpose of this paper is to give insights regarding which one of the analyzed three open sources Core is the best in terms of security from a web technology perspective. As already mentioned there is a consideration that must be done, mainly due to the inability to test these components in the same manner. Indeed, for OpenAirInterface we were not able to prove some of the already mentioned exploits attempted with the other cores. This leads to the conclusion that the Free5Gc is the most secure of the three analyzed Cores and over the tested attacks. However, this is an important point because each of the leveraged exploits presents a severity that must be taken into account; since not all the flaws found have the same criticality.

REFERENCES

[1] "5G; System architecture for the 5G System (5GS)," ETSI, TS 123 501 v17.7.0, Jan. 2023.

[2] A. Padmanabhan, "5G ue data rate," https://devopedia.org/5g-ue-data-rate, October 2023.

[3] H. Williams, "A timeline of 5G development: From 1979 to now," https://www.techadvisor.com/article/724833/a-timeline-of-5g-development-from-1979-to-now.html, April 2020.

[4] Q. Tang, O. Ermis, C. D. Nguyen, A. De Oliveira, and A. Hirtzig, "A systematic analysis of 5g networks with a focus on 5g core security," *IEEE Access*, vol. 10, pp. 18 298–18 319, 2022.

[5] S. Sullivan, A. Brighente, S. A. P. Kumar, and M. Conti, "5G security challenges and solutions: A review by osi layers," *IEEE Access*, 2021.

[6] A. Lotto, V. Singh, B. Ramasubramanian, A. Brighente, M. Conti, and R. Poovendran, "Baron: Base-station authentication through core network for mobility management in 5G networks," in *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2023, pp. 133–144.

[7] M. Chlosta, D. Rupprecht, C. Pöpper, and T. Holz, "5G suci-catchers: Still catching them all?" in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021, pp. 359–364.

[8] G. Holtrup, W. Lacube, D. P. David, A. Mermoud, G. Bovet, and V. Lenders, "5G system security analysis," *Cyber Defence Campus*, August 2021.

[9] R. Pell, S. Moschoyiannis, E. Panaousis, and R. Heartfield, "Towards dynamic threat modelling in 5G core networks based on mitre att&ck," October 2021.

[10] B. I. M. Altariqi, "5G core and (nfvi) network functions virtualization infrastructure penetration testing," 2020.

[11] A. Shostack, "Experiences threat modeling at microsoft." *MODSEC@ MoDELS*, vol. 2008, p. 35, 2008.

[12] G. Brown, "Service-based architecture for 5g core networks," *Huawei White Paper*, vol. 1, 2017.

[13] G. Mayer, "Restful apis for the 5g service based architecture," *Journal of ICT Standardization*, pp. 101–116, 2018.

[14] OWASP, "Top 10 web application security risks," https://owasp.org/www-project-top-ten/, 2021.

[15] F. J. de Souza Neto, E. Amatucci, N. A. Nassif, and P. A. M. Farias, "Analysis for comparison of framework for 5G core implementation," 2021.

[16] Y. Liu, Q. Li, Q. Cao, Z. Huang, Y. Li, and Y. Fan, "Evaluation of free5gc forwarding performance on private and public clouds," in *2022 IEEE Cloud Summit*. IEEE, 2022, pp. 9–16.

[17] T. Kim, J. Kim, H. Ko, S. Seo, Y. Jcon, H. Jeong, S. Lee, and S. Pack, "An implementation study of network data analytic function in 5g," in *2022 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2022, pp. 1–3.

[18] L. Mamushiane, A. Lysko, H. Kobo, and J. Mwangama, "Deploying a stable 5g sa testbed using srsran and open5gs: Ue integration and troubleshooting towards network slicing," in *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE, 2023, pp. 1–10.

[19] O.-M. Dumitru-Guzu and C. Vlădeanu, "Analysis of potential threats in nextgen 5g core," in *2022 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, 2022, pp. 1–4.

[20] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "Openairinterface: A flexible platform for 5g research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.

[21] CloudFlare, "What is the principle of least privilege?" https://www.cloudflare.com/learning/access-management/principle-of-least-privilege/, 2017.

[22] PortSwigger, "What is directory traversal and hot to prevent it," https://portswigger.net/web-security/file-path-traversal, 2018.