future internet

MDPI

Article

A Vulnerability Assessment of Open-Source Implementations of Fifth-Generation Core Network Functions

Filippo Dolente, Rosario Giuseppe Garroppo and Michele Pagano *

Department of Information Engineering, University of Pisa, Via Girolamo Caruso, 16, 56122 Pisa, Italy; f.dolente@studenti.unipi.it (F.D.); rosario.garroppo@unipi.it (R.G.G.)
* Correspondence: michele.pagano@unipi.it

Abstract: The paper presents an experimental security assessment within two widely used open-source 5G projects, namely Open5GS and OAI (Open-Air Interface). The examination concentrates on two network functions (NFs) that are externally exposed within the core network architecture, i.e., the Access and Mobility Management Function (AMF) and the Network Repository Function/Network Exposure Function (NRF/NEF) of the Service-Based Architecture (SBA). Focusing on the Service-Based Interface (SBI) of these exposed NFs, the analysis not only identifies potential security gaps but also underscores the crucial role of Mobile Network Operators (MNOs) in implementing robust security measures. Furthermore, given the shift towards Network Function Virtualization (NFV), this paper emphasizes the importance of secure development practices to enhance the integrity of 5G network functions. In essence, this paper underscores the significance of scrutinizing security vulnerabilities in open-source 5G projects, particularly within the core network's SBI and externally exposed NFs. The research outcomes provide valuable insights for MNOs, enabling them to establish effective security measures and promote secure development practices to safeguard the integrity of 5G network functions. Additionally, the empirical investigation aids in identifying potential vulnerabilities in open-source 5G projects, paving the way for future enhancements and standard releases.

Keywords: 5G security; Open5GS; Open-Air Interface; NF vulnerability assessment

Citation: Dolente, F.; Garroppo, R.G.; Pagano, M. A Vulnerability Assessment of Open-Source Implementations of Fifth-Generation Core Network Functions. *Future Internet* **2024**, *16*, 1. https://doi.org/10.3390/fi16010001

Academic Editors: Tao Wan and Haixin Duan

Received: 6 October 2023
Revised: 14 December 2023
Accepted: 15 December 2023
Published: 19 December 2023

1. Introduction

With the mounting surge in the number of devices connecting to mobile networks, particularly within the realm of 4G networks, there is an ever-growing strain on network capacity. This challenge has spurred the imperative for network densification and the emergence of novel technologies like 5G to accommodate the escalating number of subscribers within the network ecosystem [1].

In line with the observations presented in the Cisco Internet Annual Report [2], the forecasted volume of devices connected to IP networks is poised to surpass three times the global population by the year 2023. By 2023, there will be 3.6 network devices per capita, a substantial increase from the 2.4 network devices per capita recorded in 2018. Consequently, the global count of network devices is projected to surge to 29.3 billion by 2023, rising from 18.4 billion in 2018. This rapid expansion is underscored by the estimation that machine-to-machine (M2M) connections will account for half of all globally connected devices and connections by 2023, translating to 14.7 billion M2M connections.

This seismic trajectory in mobile technology growth coincides with the advent of the 5G mobile network, a pivotal juncture that is exerting significant stress on network capacity. The advent of 5G technology constitutes a monumental leap forward in the capabilities of mobile networks, ushering in opportunities for a plethora of new use cases that were previously inconceivable with 4G technology. This remarkable stride has the potential to stimulate substantial economic growth for businesses across various domains. The far-reaching applications of 5G encompass realms such as industrial management,

telemedicine (including remote surgical operations), IoT-equipped self-driving cars, smart city infrastructure, and much more.

In light of these compelling dynamics, the imperative to safeguard the security of the entire 5G infrastructure becomes paramount, necessitating rigorous testing and scrutiny of the novel features introduced by this groundbreaking mobile standard.

Developed under the aegis of the Third Generation Partnership Project (3GPP), the 5G standard has been meticulously designed to rectify the vulnerabilities inherent in its precursor, 4G. This new standard incorporates more rigorous constraints to fortify user traffic and enhance end-user security. Nonetheless, it is imperative to recognize that while the 3GPP imposes stringent rules governing network design and functionality, the responsibility for implementing security measures ultimately rests with 5G vendors and Mobile Network Operators (MNOs). This underscored dependence implies that network security can only be assured through the proactive engagement of MNOs in adopting and enacting suitable security protocols.

In addition to the shifting paradigm brought about by the introduction of Network Function Virtualization (NFV), where dedicated hardware devices are being supplanted by software-based solutions deployable on generic servers, the relevance of adhering to secure development practices takes center stage. The adoption of practices such as establishing a continuous integration/continuous delivery (CI/CD) pipeline and leveraging code testing tools becomes paramount. These measures ensure the robust security and vulnerability-free nature of the functions deployed within the network architecture, especially as the landscape transitions towards software-oriented solutions.

This paper is motivated by the need to critically examine and experimentally analyze the security vulnerabilities of two prominent open-source 5G projects, Open5GS [3] and the Open-Air Interface (OAI) [4], within this rapidly evolving landscape. The selection of these two frameworks takes into account their popularity. Indeed, they are used in different large testbeds for experimentally testing alternative deployment architectures and new services and protocols [5].

1.1. Paper Contribution

The primary goal of the paper is to conduct an experimental investigation into the security vulnerabilities present in open-source 5G projects, namely Open5GS and OAI. Specifically, our focus centers on analyzing the security landscape of the 5G core network, with a pronounced emphasis on evaluating the Service-Based Interface (SBI), the novel communication interface connecting network functions (NFs) that was introduced with the 5G standard. Our scrutiny will be directed towards two pivotal components within the Service-Based Architecture (SBA) NFs: the Access and Mobility Management Function (AMF) and the Network Repository Function/Network Exposure Function (NRF/NEF).

In summary, the paper's central contribution lies in its rigorous experimental exploration of the security vulnerabilities inherent in the AMF and NRF/NEF implementation of the above-mentioned open-source 5G projects. By focusing on the core network's SBI and scrutinizing externally exposed NFs, this analysis not only sheds light on potential security gaps but also underscores the critical role of 5G vendors and MNOs in ensuring the implementation of robust security measures. Furthermore, in the backdrop of the evolution towards NFV, the paper underscores the significance of secure development practices to bolster the integrity of 5G network functions. To provide a visual summary of our experimental tests, an accurate comparison of the main issues identified in the two implementations is also presented in the final section.

Finally, all the traffic captures and the scripts used in the experimental analysis are available in the GitHub repository [6], so that readers can not only reproduce the paper results but also check the evolution of the frameworks' releases in terms of security (or adapt the scripts to other frameworks of interest).

1.2. Organization of This Paper

The paper is organized as follows. The related work is discussed in Section 2, while Section 3 presents some background information on 5G architecture and Section 4 focuses on the security issues in 5G infrastructure, outlining the main vulnerabilities that will be analyzed in the paper. A general overview of our testing methodology and related software tools is then provided in Section 5, while Sections 6–8 present the experimental results for DoS (denial of service), replay, and API injection attacks, respectively. Finally, Section 9 concludes the paper with a recap of our main experimental findings and some general remarks on security concerns of 5G infrastructure. Furthermore, Section 9 contains a list of the abbreviations used in this manuscript.

2. Related Work

To gain a comprehensive understanding of the security threats in 5G networks, it is essential to consider recent survey papers that provide an overview of various aspects of 5G security. For example, Nencioni et al. [7] offer an analysis of 5G-MEC architecture and its security implications, which is crucial in understanding the security challenges in 5G networks. Bozorgchenani et al. [8] highlight the introduction of new security challenges in 5G due to the utilization of enabling technologies and the high degree of network heterogeneity. This is important in understanding the evolving nature of security threats in 5G networks. Additionally, Salazar et al. [9] discuss the new cybersecurity threats introduced by 5G SBA, emphasizing the ineffectiveness of previously adopted security and privacy mechanisms in the context of 5G. Mahyoub et al. [10] present a detailed 5G security analysis from the perspective of network architecture. They closely examine the critical 5G interfaces and provide a set of possible vulnerabilities and threats that would occur if no proper security mechanisms were set in place.

Considering commercial networks, there are works such as Park et al. (2021) [11] that analyze if the mobile operators provide 5G services applying techniques of reaction related to security threats. Their analysis lists the security challenges that are valid in the real 5G NSA (non-stand-alone) network and how they could be mitigated. However, the considered NSA networks are based on the Evolved Packet Core (EPC) architecture, and the identified vulnerabilities are related to the elements of the EPC architecture.

In their study on 5G architecture, Park et al. (2022) [12] propose a 5G security system to ensure security within the 5G core network. They evaluate the performance and security features in real time, focusing on interfaces used in NAS, HTTP, PFCP, and GTP protocols. The authors introduce an effective detection algorithm for PFCP-in-GTP and IPsec disable attacks, representative of 5G vulnerability attacks. Tests were conducted using a commercial 5G core network system simulator.

The formal verification of 5G network protocols has been approached through various models. Basin et al. [13] model the Authentication and Key Agreement Protocols (5G-AKA) of 5GS, while Hussain et al. [14] analyze several NAS and RRC protocols. Yang et al. [15] present a formal model of the Authentication and Key Management for Application (AKMA) service used for establishing authenticated communication between users and application functions. Akon et al. [16] recently introduced the first scalable formal model of the 5G core network's access control mechanism. The proposed 5GCVerif is an adversary-controlled framework designed for the systematic formal analysis of the access control mechanism of the 5G core network, revealing five new classes of exploitable privilege escalation vulnerabilities in the technical specifications.

Conversely, numerous fuzzers have been developed in the past to identify new vulnerabilities. The National Computing Centre (NCC) group introduces three representative tools: Fuzzowski 5GC, Frizzer2, and AFLNet. Among them, Frizzer2 and AFLNet are open-source network protocol fuzzers accessible to anyone [17]. The group examines two open-source solutions, Free5GC and Open5GS, with the latter chosen for its stability and ease of installation. While neither of these solutions is of commercial grade, they provide a reasonable test target for free.

Salazar et al. [18] propose several scenarios to simulate attack injection using the open-source simulator 5Greplay [19]. These scenarios include modifying and injecting network traffic offline, sending malformed packets against open-source 5G cores in real time, and conducting a NAS-5G Security Mode Command (SMC) replay attack. The last use case involves generating high-bandwidth traffic to validate the scalability of 5Greplay.

5Greplay can be considered a classical traffic replay tool when inputting a PCAP file. However, it distinguishes itself from classical traffic replay tools due to its ability to modify 5G protocols, such as NAS-5G and Next Generation Application Protocol (NGAP). Users can modify attributes in a completely arbitrary manner. Dumitru-Guzu et al. [20] deploy the Open5GS core as microservices orchestrated by Kubernetes to simulate two attack scenarios on the control and user plane. The simulation involves injecting traffic using the 5Greplay tool to analyze the AMF throughput response under the two attack scenarios.

3. Background on 5G System

This section is organized in two parts: the first one presents the architecture of the network, while the second one focuses on communication aspects between the NFs.

3.1. The 5G Architecture

As illustrated in Figure 1, the 5G system architecture is composed of three main parts: the user equipment (UE), the radio access network (RAN), and the 5G core network (5GC). The figure highlights the main SBIs on the control plane and the reference points involving NFs working on the user plane [21]. The presented architecture refers to the 5G stand-alone nonroaming scenario. This architecture can be used both in the case of private and public networks.

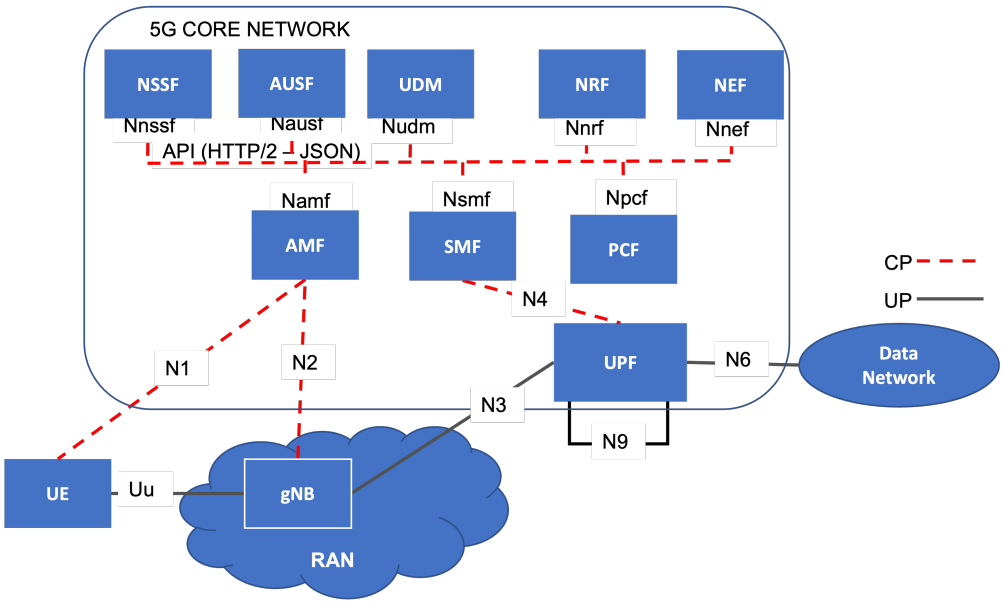

Figure 1. 5G system architecture.

The RAN is a mesh network of gNodeBs (gNBs), which facilitates wireless connectivity for 5G users to 5GC via the Uu air interface. The gNodeB engages with the UE through New Radio Radio Resource Control (NR RRC) protocols in the control plane (CP) and forwards traffic data to the user plane function (UPF) in the user plane (UP). Additionally, it establishes a relay bridge for the transmission of non-access stratum (NAS) messages between the UE and the AMF, to manage the establishment of communication sessions and to maintain continuous communications with the UE as it moves.

UE denotes the mobile equipment (ME) equipped with a Subscriber Identity Module (SIM). The ME, typically represented by smartphones, serves as a cellular device capable of establishing connections with gNB. The SIM can take the form of a physical Universal SIM

(USIM) or a virtual eSIM. It stores subscriber data identical to that found in the operator's Unified Data Management (UDM) database, including the globally unique Subscriber Permanent Identity (SUPI), cryptographic keys, and security algorithms.

The 5GC incorporates an array of NFs, each serving distinct roles. Noteworthy, on the control plane among these are the following:

- The Network Exposure Function (NEF) facilitates the exposure of NF functionality externally, fostering interaction with third-party applications.
- AMF orchestrates access control and mobility management.
- The Network Slice Selection Function (NSSF) facilitates the selection of network slices for user-requested services.
- The Session Management Function (SMF) is responsible for session setup and management in alignment with network policies.
- The Policy-Control Function (PCF) provides policy management capabilities.
- UDM stores subscriber data and profiles.
- The NF Repository Function (NRF) enables registration and discovery of NFs for interfunction communication.
- The Authentication Server Function (AUSF) is responsible for user authentication.

On the data plane, only UPF provides all the functions necessary for user packet routing and forwarding. UPF corresponds to the gateway concept in 4G, interfacing the 5G systems with the external data networks.

Of note, NEF and NRF functions stand as noteworthy additions in the 5GC compared to the 4G EPC. Although both fulfill similar roles, NEF targets external network exposure, while NRF serves internal 5GC functions.

3.2. NF Communication Procedures

The architecture of the 5GC ushers in a novel paradigm known as SBA. In this innovative architectural framework, a microservice structure is adopted, wherein each component (i.e., NF) is assigned a well-defined and specialized task to enhance modularity. These NFs communicate with one another through API requests via the SBI, facilitating the exchange of services. This architecture accommodates the utilization of NFs from diverse vendors, as the internal workings of these functions are externally exposed through interfaces standardized by the 3GPP.

The 3GPP has laid out meticulous guidelines in TS 29.501 [22] to ensure consistent and interoperable API design across the 5G ecosystem. In this context, OpenAPI version 3 [23] serves as the standardized language for defining APIs. This description format allows for comprehensive interface specifications, enabling both human and machine comprehension of APIs without necessitating extensive source code or documentation scrutiny.

All the `Nxxx` interfaces (communication between functions) are based on REST technologies: API requests in HTTP/2, encoded in JSON format [24]; see Figure 1. Authorization and access control within SBA are ensured by OAuth 2.0 [25]. As such, web technologies, particularly RESTful APIs, play a dominant role in managing control functions (e.g., mobility, access control, etc.) within the network.

This integration of APIs within the SBA framework enables a transformative shift from proprietary protocols to a dynamic, service-oriented environment. This evolution enhances the network's interaction with third-party developers and vertical industries, fostering the development of tailored, on-demand services. The quintessential role of RESTful APIs as the backbone of this architectural model is indisputable, underpinning the technical realization and integration of the 5G ecosystem.

Integral to the architecture is the integration of NFV, which shifts the network landscape away from dedicated hardware devices in favor of software-based solutions. This transition underscores the significance of adopting cloud-native technologies. Embracing this approach means that network functions like UPF and SMF can be developed using open-source software, such as [3,4]. The software–hardware disaggregation of the network components working on the Uu air interface has pushed the open-source approach in

the RAN, as shown by the Open Radio Access Network (O-RAN) Alliance [26]. This signifies the convergence of open-source and cloud-native principles in 5G deployment, distinguishing it as the pioneer in this domain.

The interaction among NFs in 5GC is characterized by a "producer–consumer" relationship. This relationship promotes dynamic collaboration, facilitating the smooth exchange of services between functions and thereby enhancing the overall network efficiency.

Within the orchestration of NFs in the 5GC, a dynamic model emerges in which each NF serves as both a "producer" and a "consumer" of services. This dynamic interaction model is based on "request" and "subscribe" interactions, providing flexibility and adaptability to the system. Figure 2 shows an example of an API call involving NRF. The example refers to the API-driven interaction for session establishment. The `SMF discovery` and `Session Establishment` are initiated by the AMF, serving as a key instance of the interaction flow. The NRF plays a pivotal role in facilitating SMF discovery, thereby exemplifying the essence of API-driven communication in SBA.

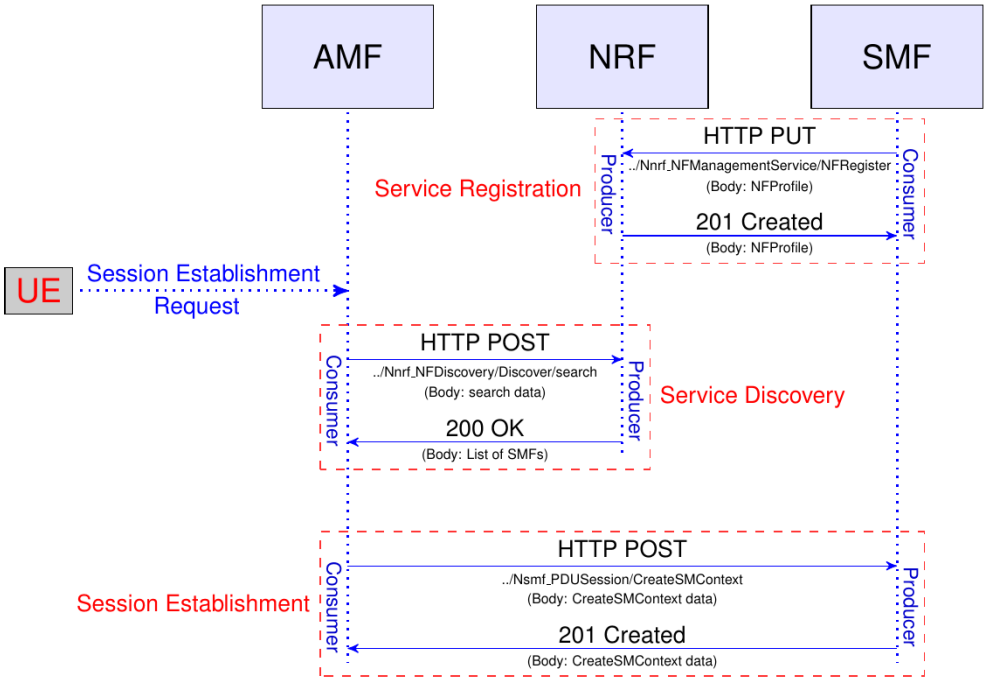

Figure 2. Example of API calls and NF interaction in SBA.

4. Security in 5G Infrastructure

The adoption of 5G technology brings opportunities across automotive, industrial, and business sectors, fostering IoT integration and productivity gains. However, this integration exposes 5G networks to unique security vulnerabilities. Addressing these risks is crucial for network safeguarding and threat prevention. An assault on a 5G network carries more profound consequences than on 4G or earlier networks. This arises from the intricate 5G network structure and its pivotal role in critical industries.

The 5G security framework encompasses defense against multiple threats: radio access, signaling planes, user planes, impersonation, confidentiality breaches, replay attacks, downgrades, man-in-the-middle intrusions, and interoperator security issues. These concerns were meticulously considered during the 5G design and deployment [27].

This section presents a concise overview of the security mechanisms in 5G systems. It includes a brief description of the protection mechanisms adopted in 5G, highlighting their improvements over 4G, as well as an outline of the major vulnerabilities associated with core network elements.

4.1. Fifth-Generation Protection Mechanisms

The security foundations of 5G systems are shaped by the security requirements outlined in [28]. This document underscores the five pillars of information security: confidentiality, integrity, availability, authenticity, and nonrepudiation. The security protocols established in 5G are designed to safeguard networks and users against diverse security threats, including cyberattacks, data breaches, and unauthorized access.

Notably, 3GPP security specifications are mandatory for vendors, yet optional for 5G service providers. This flexibility in implementation has led to varying security levels across different 5G networks. This approach caters to the varied regulations of different countries, some of which opt not to mandate certain security features for national telecommunications providers to preserve privacy [29].

General Security Requirements

One of the most important functions considered in the general security architecture is related to the authentication and authorization of UEs. In particular, subscriber and serving network authentication ensure authorized access and secure communication. Serving networks verify the SUPI authenticity, while UEs authenticate the serving network identifier to counter rogue networks or man-in-the-middle attacks. Furthermore, the threat of bidding down attacks necessitates robust security mechanisms. These attacks manipulate negotiation between UE and network entities, deceiving both into believing the other lacks specific security features. Preventive measures involve mutual authentication, secure negotiation of security parameters, and detecting and rejecting misleading messages.

The security architecture considers the UE requirements. The most important is the guarantee of confidentiality and integrity of user data and signaling data. UEs must encrypt user data and signaling messages to preserve confidentiality and integrity. Ciphering activation by gNB indication, support for different data rates, and integrity protection mitigate tampering and replay threats.

The same security services are required on the gNBs and AMF. gNBs encrypt user data and signaling messages based on security policies and requirements. Versatility in applying confidentiality and integrity protection balances security needs. AMFs encrypt and protect NAS signaling messages, enhancing confidentiality and integrity. The security policy-driven application of protection mechanisms renders confidentiality protection optional.

The security requirements considered for the SBA are mainly related to the interaction among the different NFs. Among the NFs, the NRF represents the most critical one from the security perspective because it enables the registration and discovery of NFs for interfunction communications. Given this role, the NRF must support confidentiality, integrity, and replay protection in the *discovery* and *registration* processes between the NF Service Consumer and NF Service Producer. The NRF must authorize discovery and registration requests, allowing only authorized entities to discover and register with NFs. The process should hide the topology of NFs within one trust domain from entities in different domains, preventing unauthorized access and disclosure of network topology. Furthermore, Mutual authentication between the NF Service Consumer and NF Service Producer is crucial for secure communication.

The other critical element is the NEF, which exposes NF functionality externally for fostering interaction with third-party applications. The NEF must ensure **integrity, replay,** and **confidentiality protection** for communication with application functions (AFs). **Mutual authentication** between the NEF and AFs is necessary to prevent unauthorized access and ensure secure communication.

Each NF must validate all incoming messages. Messages failing validation based on protocol specification and network state should be rejected or discarded. This ensures the processing of valid messages and the discarding of malicious ones.

The extracted NF requirements highlight the need to validate every received message, aligning with zero-trust principles. Given the 5G vast and perimeter-less ecosystem, reliance on trustworthiness to third parties or other NFs is insufficient. Attacks like API injection,

Distributed DoS (DDoS) attacks, and replay attacks are potential threats; therefore, in our experimental testbeds, we will verify their impact on AMF and NRF/NEF in two popular 5G open-source projects.

The 5G landscape incorporates APIs and exposes the core to third parties, with the Common API Framework (CAPIF) introduced in Release 15 and expanded in Release 16. CAPIF establishes a consistent northbound API framework for diverse 3GPP functions. This framework encompasses the API Exposing Function, API Publishing Function, and API Management Function.

Any SIM card contains an International Mobile Subscriber Identity (IMSI) uniquely identifying a subscriber in the operator's network. The IMSI, comprising 15/16 digits, was sent in clear text during NAS procedures in the past mobile generation. This feature has been exploited to perform the tracking of users' location and violate the users' privacy.

In 4G, the generation of a Globally Unique Temporary ID (GUTI) avoids one sending IMSI in the clear. However, the development of IMSI catcher techniques allows to overcome the barrier added by the temporary identifier [30]. In 5G, to contrast this issue, privacy starts preauthentication via encrypting the SUPI using the home network public key in the USIM, generating the Subscription Concealed Identifier (SUCI). This approach mitigates the exposure of subscriber identity during transmission [27]. However, it does not completely prevent all tracking and linking. In scenarios where a user's identity is already known, an attacker can still probe users for that identity, as exemplified by the SUCI-Catcher attack [31]. Furthermore, the recent analysis of public mobile networks shows that the SUCI mechanism is not deployed [32,33].

Two types of SUPI are defined: the Network Specific Identifier (NSI) and the IMSI [34]. In the first case, the unique identifier is composed of a variable-length username and domain name (e.g., `userXXX@serviceprovidername.com`). In the second case, the SUPI consists of a 3-digit Mobile Country Code (MCC), a 2–3-digit Mobile Network Code (MNC), and a 10–11-digit Mobile Subscriber Identification Number (MSIN). The SUCI always contains the SUPI type (e.g., IMSI, NSI), Home Network Identifier, Routing Indicator, Protection Scheme, Home Network Public Key ID, and Protection Scheme Output.

The Home Network Identifier contains the domain name of the NSI SUPI, while consisting of MCC and MNC in the case of the IMSI. The Protection Scheme Output completes the SUPI by reporting the username part of NSI SUPI, or the mobile subscriber identification number (MSIN) of the IMSI.

Although not considered in our testbeds, it is worth mentioning that other than mobility, roaming is a distinctive feature of mobile networks. The fifth-generation system introduces Security Edge Proxy Protection (SEPP) for roaming interconnection, which ensures secure communication between home network and serving network. Two Internet Protocol Exchanges (IPXs) are established between the Visited and the Home Public Land Mobile Networks (VPLMN and HPLMN, respectively), and SEPP implements application layer security, preventing IPX providers from reading sensitive data. It detects unauthorized message modifications and performs source validation, message format verification, and topology hiding [27]. Communication between SEPPs uses HTTP/2 protocol via well-defined API requests, so our testing methodology (see, in particular, Section 5.4) could be easily extended to SEPP.

Finally, for non-3GPP access networks, 3GPP has considered two access scenarios: trusted and untrusted. In the trusted scenario, traffic is exchanged with the trusted non-3GPP access using the Trusted Non-3GPP Gateway Function (TNGF). In the case of untrusted non-3GPP networks, the Non-3GPP Interworking Function (N3IWF) acts as a security gateway, enabling the UE to access the 5G core over untrusted non-3GPP networks through IPsec tunnels for both UP and CP traffic [35].

4.2. Vulnerabilities in 5G System

The presented security mechanisms address critical aspects of the 5G system. However, the 5G system complexity and novel use cases have exposed it to vulnerabilities not previously encountered.

Vulnerabilities span the RAN, Core, and UE, but we will focus on vulnerabilities in the 5G core network. Replay attacks on N1 and N2 interfaces, API vulnerabilities on the Nnef interface and Nnrf interface, and stability tests against DDoS attacks are primary concerns.

SBA, reliant on APIs, introduces vulnerabilities due to poor developer implementation. Common API web server issues are exploitable, potentially compromising critical infrastructure or causing a DoS attack.

Numerous vulnerabilities stem from factors such as IoT/M2M implementations [36], multistack environments [37], open-source programs [38], improper API usage [39], third-party apps [40], and insufficient testing [41]. Cloud deployment introduces shared security responsibility and necessitates additional security controls beyond 3GPP [27].

Different works on vulnerability analysis of 5G systems have identified threats to various NFs. For example, Mahyoub et al. [10] examine critical 5G interfaces, providing a set of possible vulnerabilities and threats. Comprehensive vulnerabilities are also detailed in [27], categorized as shown in Table 1. It is worth noting that this section specifically addresses the subset of threats pertinent to the scope of the presented study.

Table 1. Summary of vulnerabilities in 5G system.

User Equipment	Air Interface	RAN	MEC	5G Packet Core	N6 and Roaming
Malware	MitM	Rouge Nodes	DDoS Attacks	Virtualization Orchestration Vulnerabilities	IoT Core Integration
Firmware Hacks	Jamming	Rouge Applications	API Vulnerabilities	Improper Access Control	App Server Vulnerabilities
Tampering		Insecure S1, X2	Side Channel Attacks	Network Slice Vulnerabilities	Application Vulnerabilities
Sensor Susceptibility		Insecure Xx, Xn	NFVi Vulnerabilities	API Vulnerabilities	API Vulnerabilities
				NEF Vulnerabilities	
				Roaming Partner	
				DDoS and DoS Attacks	

4.2.1. Insecure API

The vulnerabilities of each NF deployment mode include infrastructure (hardware and container) and communication (API) vulnerabilities:

- Container vulnerabilities: Image vulnerabilities, insufficient isolation, improper configuration, container run-time vulnerabilities;
- Hardware and host vulnerabilities: hardware and software vulnerabilities or improper access control;
- Internal interfaces: API vulnerabilities (within SBI), insecure networking (eavesdropping);
- External interfaces: API vulnerabilities.

API security is a crucial security aspect for 5GC cloud-native functions. APIs often have simple code that assumes that clients handle filtering, lacking adequate security scrutiny for request and response messages. Developers may overlook the sensitivity of transferred information.

Modern automated tools can exploit poorly implemented API interfaces for API injection attacks, causing information compromise, malfunction, or DoS attacks. An example of an automated tool is Burp Suite [42], used in the presented experimental evaluation to test the security level of the NF's API interfaces.

Note that the wide use of distributed computing architectures like Multiaccess Edge Computing (MEC) to improve the end-user experience or application performance introduces new concerns in terms of security. Indeed, MEC connections to core 5G could allow attacks exploiting weak implementations within MEC to establish indirect connections to core 5G, leading to API injection. Proper Transport Layer Security (TLS) usage, as mandated by 3GPP, can mitigate this problem, although challenges persist in TLS management and implementation [43].

Furthermore, encrypted container communication hampers network security controls, necessitating anomaly-based Intrusion Detection Systems (IDSs) (possibly based on AI and ML [44]) to analyze encrypted packets without decryption and detect malicious packets (with all the limitations related to false positives and false negatives).

For MEC applications, API gateways or API firewalls are advised, along with adherence to established API security protocols to counter API attacks.

4.2.2. DDoS Attacks

Fifth-generation core network functions must handle malformed requests and offer mechanisms to filter or limit requests during service overload. These controls should be intrinsic to each NF implementation to avoid relying solely on external measures. Inadequate handling of malicious requests and the absence of proper input controls can lead to attacks like DDoS. To ensure the correct implementation of crucial security checks, optional devices like the Security Gateway (SecGW) are required.

To defend against inbound DDoS attacks, a DDoS protection system featuring anti-DDoS and Web Application Firewall (WAF) capabilities can be positioned between the Internet/web layer and the MEC layer (or 5G core, if MEC is unused). This safeguards against HTTP flood attacks, where excessive HTTP requests from bots attempt to overwhelm the system. The solution must distinguish legitimate from illegitimate requests, permitting legitimate traffic and diverting DDoS traffic for analysis.

Security strategies against DDoS attacks [45] involve vulnerability assessments, software updates, enhanced visibility, anti-DDoS tools, and AI/ML-based threat intelligence for proactive risk identification.

4.2.3. Traffic Interception

The traffic between the centralized 5GC data center and the Enterprise MEC can utilize an insecure public network: indeed, data transmitted from the MEC traverse the N2 (gNB to AMF), N4 (UPF to SMF), and N9 (Inter-UPF Interface) interfaces. The exposure of these interfaces to a backhaul sniffing threat requires protection for integrity, confidentiality, and replay in insecure networks, as these security layers are not inherently present.

The use of IPsec via SecGWs between the enterprise MEC and centralized 5GC can safeguard data transmission over an unsecured public network, including N2, N4, and N9 interfaces, ensuring packet confidentiality.

Similar vulnerabilities exist in the communication channel between the UE and the gNB. The absence of confidentiality, integrity, and protection against replay attacks exposes the N2 interface or the N1 interface to DDoS and replay attacks.

5. Experimental Methodology

Our work focuses on the security aspects related to the NFs that are exposed to the outside of 5GC, either directly or indirectly, such as the AMF (considering both the N1 interface with the UE, through the NAS, and the N2 interface) and the NRF/NEF, which acts as an API gateway between the 5GC and third-party applications, to offer services outside the 5G network. At first, our investigation is centered on the evaluation of the reliability and security posture of the AMF component within the 5GC. Specifically, our goals encompass the identification of vulnerabilities related to replay attacks and DoS attacks. Then we test the effectiveness of security measures implemented in the NRF/NEF, focusing on DoS attacks and API injection techniques, which are common in web application security.

While the analysis of the NEF component is restricted due to the absence of open-source implementations at the time of this study, we leverage simulations and tests on the NRF to gain insights into the security landscape of the SBI. In more detail we aim to scrutinize the vulnerability landscape by examining the OWASP Top 10 vulnerabilities and assessing their applicability to 5GC, considering its unique communication methodologies, particularly via REST API requests. Furthermore, we explore scenarios involving communication between different operators' NRFs, leading to considerations of potential malicious actions that might compromise competitors' 5GC infrastructures.

5.1. Testing Methodology

The testing methodology is based on the simulation of various attack scenarios and on the evaluation of the system's response to these threats. As already mentioned, the testing covers the vulnerability assessment of AMF and NRF, considering two well-known open-source 5G projects, namely Open5GS and OAI. A summary of the main features of the tested software is shown in Table 2.

Table 2. Main features of the tested OAI and Open5GS frameworks.

	OAI	**Open5GS**
Version used	1.4.0	v2.4.11-17
Deployment mode supported	VM, Container, Kubernets	VM, Container, Kubernets
Protocols supported in SBI	HTTP/1.1 and HTTP/2	HTTP/2
TLS in SBI	No	No
Declared compliance 3GPP standard	Release 16	Release 16
Different deployment modes supported	Yes, with script	Yes, configuration to be performed manually
Configuration needed before run	No	Yes, Configuration on AMF and UPF
API Adherence to the 3GPP Rel 16 Standard	Partial	Partial

For both NFs, the DoS attack has been tested using a free and open-source tool, **MHDDoS** [46], which allows us to determine the behavior of these NFs under heavy loads.

As concerns the AMF, we emulate a replay attack utilizing the **5Greplay** tool [19]. The goal of this test is to assess AMF resilience to the reply of registration messages from the UE. Finally, leveraging the **Burp Suite** tool [47] (and the ad hoc **5GHTTP modifier** tool for the Open5GS testbed), we perform the API injection test on the NRF by examining the OWASP Top 10 vulnerabilities. The **5GHTTP modifier** tool, configuration files and some acquired data of the presented analysis are available in [6].

The rest of this section presents in detail all the above-mentioned specific attack scenarios. Then, the experimental results will be reported in the following sections.

5.2. DoS Attacks (TCP ACK Flooding) on AMF and NFR

In the realm of cybersecurity, TCP ACK flooding (known also as PSH–ACK flooding) emerges as a form of DDoS attack, targeting the transport layer implementation on the victim. The core of this attack lies in flooding the victim with an excessive number of TCP packets, each having the ACK and PSH flags set. This is performed to overwhelm the target's available network bandwidth and resources. The PSH flag, in particular, poses an additional burden on the victim as it initiates instant data delivery to the receiving application, bypassing the standard TCP receive buffer handling process. As a result, the victim's capacity to manage legitimate requests and respond to concurrent network traffic becomes severely impaired. Note that unlike SYN flooding, the TCP three-way handshake

has been completed with a multitude of different source port numbers and the connections are in the established state.

The effectiveness of the attack can be magnified by harnessing a network of compromised devices (botnets) distributed across various locations.

It is essential to highlight that the AMF encompasses two distinct types of interfaces. The first type comprises interfaces linked to the 5G access network, exemplified by N1 and N2. These interfaces employ a protocol stack that concludes with the NAS protocol and may utilize suitable transport protocols for control plane traffic, such as SCTP implemented on the N2 interface. The second type is the SBI, establishing connections between the AMF and other NFs of the SBA, including NRF, UDM, and PCF. This interface employs TCP to facilitate the RESTful communication model integral to the SBI.

To mount this TCP ACK flooding attack, we employ **MHDDoS** with the specific configuration parameters reported in Table 3.

Table 3. MHDDoS configuration parameters.

Parameter	Value *(comments)*
method	TCP *(Flooding TCP)*
url	URL to attack *(in our case, the IP address of AMF and NRF)*
threads	200 *(number of threads employed for the attack)*
duration	Attack duration *(set to 500 s)*
debug	True *(to collect information about the incoming requests in the AMF logs)*

It is important to highlight two notable limitations inherent to this test. Indeed, the absence of a multitude of machines for simultaneous DDoS attack execution and the lack of active users within the network who might experience disruptions constrain the scope of our experimentation. Thus, our evaluation extends primarily to augmenting requests to the NF and assessing its responsiveness in relation to other interfaces within the SBI. Our assessment delves into whether the NF can continue to receive messages from other NFs and fulfill API requests under the duress of the attack.

5.3. Replay Attack on AMF

The TS 33.512 [48] furnishes comprehensive security controls and specifications for the AMF, encompassing both safety and security aspects. Within this framework, the test `TC_NAS_REPLAY_AMF` mandates that the AMF be fortified against replay attacks: *Verify that the NAS signalling messages are replay protected by AMF over N1 interface between UE and AMF*.

As reinforced in TS 33.501 [28], clause 5.5.2 mandates that *AMF shall support replay protection of NAS signalling messages between UE and AMF on N1 interface*.

Therefore, our initial test scenario involves scrutinizing the susceptibility of the AMF implementation in OAI and Open5GS to replay attacks. To accomplish this, we harness the capabilities of 5Greplay [19]. This tool enables us to simulate replay attacks using intercepted traffic on interfaces N1 and N2. A similar attack scenario is also explored in [18] (Section 4.3 "Scenario 3") considering a different implementation of the 5GC, namely free5G.

During the UE interaction with the 5G system, the registration procedure is vital for network entry. This entails authentication and the establishment of encryption and integrity protection parameters at the Access Stratum and NAS. The NAS SMC procedure plays a pivotal role in this process, ensuring secure communication between the UE and the AMF (see Figure 3). After a series of exchanges between the two elements, this procedure involves the downlink transmission of the **NAS Security Mode Command** and the subsequent **NAS Security Mode Complete** message from the UE to the AMF. The AMF may also send an NAS Security Mode Command message to modify the security algorithm within an existing security context [49].

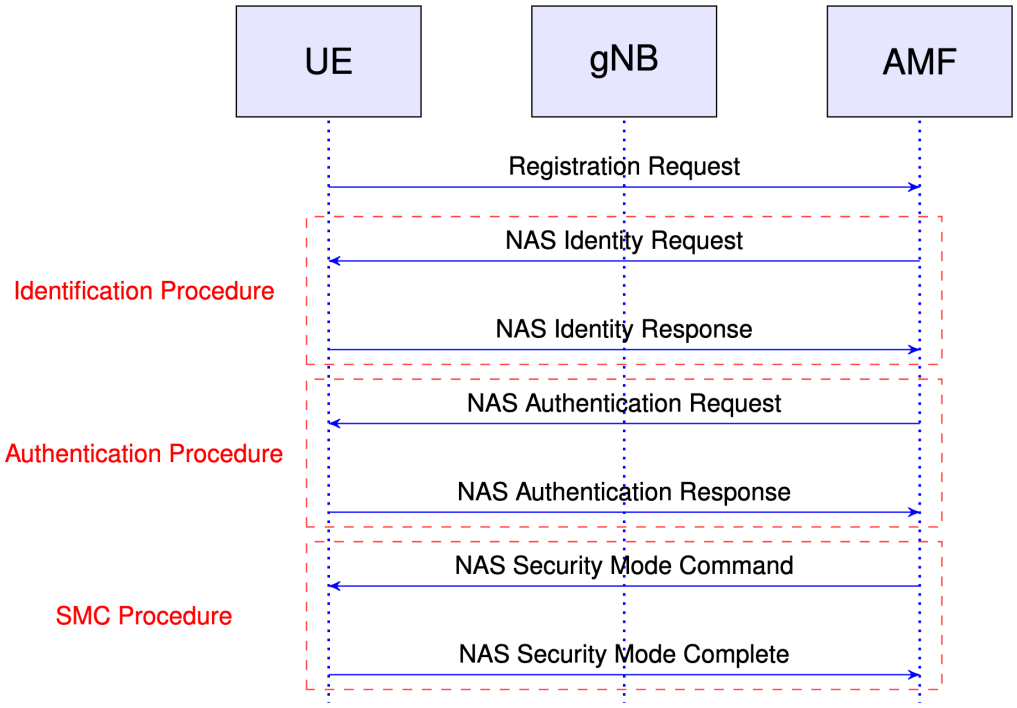

Figure 3. UE authentication procedure: The messages exchanged between UE and AMF during the identification, authentication, and SMC procedure.

Preceding the establishment of the security context, NAS messages are transmitted in an unencrypted state. This vulnerability emerges from the potential interception of a clear NAS Security Mode Command sent from the AMF to the UE. An attacker with access to NAS traffic on the N1 interface could replicate the NAS SeQuence Number (NAS SQN) and craft a replayed NAS Security Mode Complete message. Alternatively, the attacker could intercept a NAS Security Mode Complete message and retransmit it to the AMF.

If the AMF lacks sufficient safeguards against such attacks, the network may not detect the replayed packet. Consequently, the attacker could impersonate a UE, prompting the system to adopt a weaker security algorithm or disable security altogether, in both cases undermining the system integrity.

The results presented in [18] point out that the free5Gc AMF is protected against the NAS-5G SMC replay attack. Our focus is to ascertain whether Open5GS and OAI have incorporated adequate security measures to thwart this kind of replay attack.

The rules necessary for executing a replay attack on the N1 interface need to be configured within a configuration file (named *nas-smc-replay-attack.xml*) in the directory *5greplay/rules*.The rule comprises two events with a minimal delay of 0 ms and a maximum delay of 10 ms. The first event identifies a `NAS Security Mode Command` message using the `NAS message type`, set to 93. The second event detects the expected receipt of a `NAS Security Mode Complete` message after the first event, identified by the *NAS security type*, which is 4. This rule serves to filter captured network packets, transmitting only those aligned with the specified conditions.

5.4. API Injection on NRF

API injection stands as a prevalent attack vector, wherein malicious input is injected into an API to execute unintended actions or access confidential information. This breed of attack manifests through various channels, encompassing SQL injection, command injection, and payload injection among others.

SQL injection revolves around infusing malevolent SQL commands into an API endpoint responsible for interacting with databases, aiming to gain unauthorized access to sensitive data or manipulate existing information.

Command injection transpires when malicious commands are injected into an API endpoint capable of executing system commands, facilitating the execution of arbitrary code on the server.

Payload injection ensues when harmful payloads are injected into an API endpoint that processes untrusted data, ultimately leading to the execution of arbitrary code on the server.

The implications of API injection are grave, encompassing data breaches, illicit access to confidential information, and service disruptions. Mitigating this threat necessitates the implementation of robust input validation, fortified authentication, and access control mechanisms.

Given the 5GC's reliance on REST API requests (HTTP/2), the NFs of the SBA become susceptible to the entire gamut of web application attacks, thereby expanding its attack surface. Consequently, the adoption of automated tools like network vulnerability assessment tools becomes indispensable. In our tests, we employed the **Burp Suite** tool, but for Open5GS experiments an additional ad hoc "translator", named **5GHTTPModifier**, was also necessary, due to the lack of support (by Burp Suite and similar tools) for active scans on HTTP/2 servers that do not support TLS, such as Open5GS. In more detail, 5GHTTPModifier takes an XML file containing requests for a specific API as input, and then transforms them into legitimate requests for Open5GS. The tool ensures proper request forwarding and logs response data. In the case of AOI, the direct utilization of Burp Suite was possible using HTTP1.1. For this HTTP version, Burp Suite does not require TLS usage.

Burp Suite is equipped to assimilate an API schema (defined in OpenAPI) as input and initiate an active scan on the specified target, subsequently generating a comprehensive report detailing the identified vulnerabilities.

Burp Suite encompasses an extensive array of vulnerability tests, with a comprehensive list available at *portswigger.net/kb/issues*. Notable tests include OS command injection, SQL injection, File path traversal, PHP code injection, Server-side JavaScript code injection, and HTTP request smuggling. Moreover, the tool undertakes injection attempts across various elements of a request (such as parameters, body, headers, cookies, etc.), or at predefined points.

The testing methodology we adopted for API injection entails the following steps:

1. Parse the API schema definition (OpenAPI 3.0.0) using Burp Suite.
2. Initiate an automated scan for each API service in the OAI testbed, injecting malicious parameters/payloads at various junctures of the HTTP requests, including headers, body, and URLs.
3. Analyze the scan results for the OAI testbed and export them as an XML file, which serves as the input for the **5GHTTPModifier** tool.
4. Execute the **5GHTTPModifier** tool for each output obtained from the previous scan in the Open5GS testbed.

6. Vulnerability Assessment of AMF and NRF to DoS Attacks

As already mentioned, the MHDDoS tool was used to launch a TCP ACK flooding attack against the AMF and NRF in both testbeds to evaluate their ability to handle such an attack. It is noteworthy that the ACK flooding attack under consideration is designed to compel the operating system to instantly deliver the data to the application, in this case, the NF undergoing examination, thus consuming system resources. The objective is to assess the response and resilience of the specified NFs when subjected to this particular form of attack.

The MHDDoS tool was initialized with the configuration parameters as specified in Section 5.2, and the attack was repeated twice at different times.

In both testbeds, throughout the attack, the AMF continued to send its reports to the NRF and managed incoming connections. The only difference is that in the Open5GS case, the AMF logs displayed a *nghttp2_session_mem_recv() failed* message, indicating that the incoming requests did not adhere to the HTTP/2 protocol. Such an error message was expected since the tool only supports HTTP/1.1, while Open5GS (unlike OAI) only allows core network deployment through HTTP/2.

Although the attacks were unsuccessful (see the limitations of our implementation discussed in Section 5.2), they pointed out the need to implement a limiting mechanism to counteract TCP ACK flooding attacks.

In both the OAI and Open5GS testbeds, the attacks on the NRF did not lead to any noticeable slowdowns or unfulfilled requests. Throughout the attacks, we were able to ping the machine consistently, and the heartbeats among various NFs remained unaffected.

7. Vulnerability Assessment of AMF to Replay Attacks

In this section, we delve into the experimental results of the replay attack against AMF in the two considered testbeds. This attack focuses on N1 and N2 traffic, primarily dealing with UE registration and gNB association procedures. In more detail, in both scenarios, we captured the traffic exchanged between the gNB and the AMF during the simulation of UEs'registration. It should be noted that the captured traffic interactions and protocols heavily rely on the Stream Control Transmission Protocol (SCTP) as well as the new SBA of 5G networks. The replay attack's consequences and the ability of the respective 5GC components to withstand such attacks will be dissected for the two testbeds. All the relevant files (`pcap` files and log excerpts) were made available on the GitHub page [6]. These data provide visual evidence of the replay attack's effects, including registering new devices, UE disconnection, and the subsequent corrective actions taken by the 5GC components.

7.1. Testbed 1—Open5GS

The first step to perform the replay attack is the registration in a `pcap` file of the traffic exchanged between the RAN and the 5GC. This traffic encompasses UE registration and gNB association messages.

The replaying process resulted in the forwarding of all acquired packets. Analyzing the log of the program, we observed that 5Greplay successfully forwarded all 22 packets in the file. However, the outcome of forwarding all packets was the registration of a new gNB and UE with the same SUPI, essentially replicating the devices. This unexpected behavior led to the misbehavior of the entire 5G infrastructure.

In more detail, on the RAN-side, a replay attack led to the gNB receiving a UE Context Release Command message, followed by the release of the UE's RCC (Radio Connection Control) connection. This disconnection prompted the UE to switch to the "Connection Management Idle" (CM-IDLE) state. This state indicates that the UE is powered on and has registered with the network but is not actively engaged in any data transmission or communication.

When a user tries to access the Internet, the gNB receives a request to establish an RCC connection (RCC setup for UE) and an initial NAS message from UE, but these attempts fail. Only after the UE is rebooted and received a new ID from the gNB, a successful connection of UE to the core network could be established and the normal operation restored.

On the core side, three different blocks of events can be observed: the attack start time, the data network (DN) reachability request from legitimate UE, and the UE restart. During the start time of the attack, the core receives the message requesting registration of a new gNB (NGSetupRequest). The core does not perform checks on this message and it is accepted as coming from a legitimate radio station. So, no replay attack is detected, and the core registers a new gNB with the parameters given in the request (IP, TAX, PLMN_ID, etc.). It then notifies that there are two gNBs. Then, a registration procedure request (InitialUEMessage) is sent to negotiate the radio connection parameters required for network access and data communication. As the UE's data sre already registered in the

core (and the core identifies this through the SUCI - known UE by SUCI), it sends a context release request to the legitimate gNB to which the UE was previously connected, resulting in the UE losing connectivity to the core.

Replay of the registration request begins, but this time, core 5G realizes that the message has been compromised through MAC verification (NAS MAC verification) and sends an "Authentication Rejected" message. Subsequent messages from the UE to the AMF (UplinkNASTransport) are all verified through the MAC, which can detect a replay attack and trigger a context release request from the UE. This action results in the removal of the UE from the core. The UPF and SMF then proceed to eliminate the context information associated with the gNB that registered through the replay attack (which was fake). The fake gNB is instructed to release the UE's context, but since no actual gNB exists, the connection request is denied. Following this, the AMF removes the UE information and reports the count of UEs connected to the gNBs, which now amounts to 0. This indicates that there are no longer any UE devices connected to the gNB (5G base station). The MFA (Mobility Management Function for Access) removes the fake gNB from its list and reports the total number of gNBs, which is now 1.

Analyzing the DN reachability request from a legitimate UE, a request for an InitialUEMessage is received from the genuine gNB. The MFA proceeds to add this device to the list of known devices (now totaling 1) and associates it with a TMSI. However, the request ultimately fails because there is no active session established for this UE at that moment.

Finally, let us consider the UE restart. The registration process begins with the transmission of the InitialUEMessage to initiate registration at the core. Consequently, the count of UEs registered to the gNBs rises to 2, even though it is actually the same UE (the core is not aware of this yet). When the AMF recognizes an already known SUCI, it requests the gNB to release the context. During this registration request, the AMF detects an already-allocated 5G-GUTI, leading to the release of the context, resulting in the UE count associated with the gNB reverting to 1. At this juncture, an authentication request is dispatched to the AMF, initiating the SMC procedure. Regrettably, the SMC procedure fails to terminate correctly, resulting in the rejection of the registration. The UE context is released, and the SUCI-related data are erased. Subsequently, after a few seconds, the UE retries the registration procedure, which concludes successfully this time. Normal operation of the UE is restored.

7.2. Testbed 2—OAI

The observed consequences of the replay attack in the OAI testbed are more severe. The AMF crashed due to the inability to handle unexpected requests and messages: this crash caused a DoS, rendering the AMF nonfunctional.

Additionally, testing 5Greplay with different captured traffic (`ue_authentication.pcap` file [6] further substantiated the AMF's inability to handle unexpected messages. This caused the AMF to crash once again, exposing its vulnerability to replay attacks.

Overall, the experimental results from both Open5GS and OAI testbeds reveal that despite the 5G network's advanced security measures, there are vulnerabilities in the core network components that can be exploited through replay attacks. These vulnerabilities highlight the importance of deploying robust countermeasures to protect against such attacks and ensure the security of the 5G infrastructure.

8. Vulnerability Assessment of NRF to API Injection Tests

This section provides an in-depth analysis of the security vulnerabilities and potential risks associated with the NRF components of both frameworks, highlighted by API injection tests. In more detail, the 3GPP standard defines several services offered by the NRF, including NF Service Management, NF Service Discovery, and Access Tokens. The primary focus of our testing is on the NF Service Management, specifically the NRF NF Management (nnrf-nfm) interface. This interface offers various methods for managing subscriptions

and instances of NFs. The imported schema for the nnrf-nfm interface is depicted in Figure 4: the various available services and methods include creating, updating, deleting, and retrieving NF subscriptions and instances.

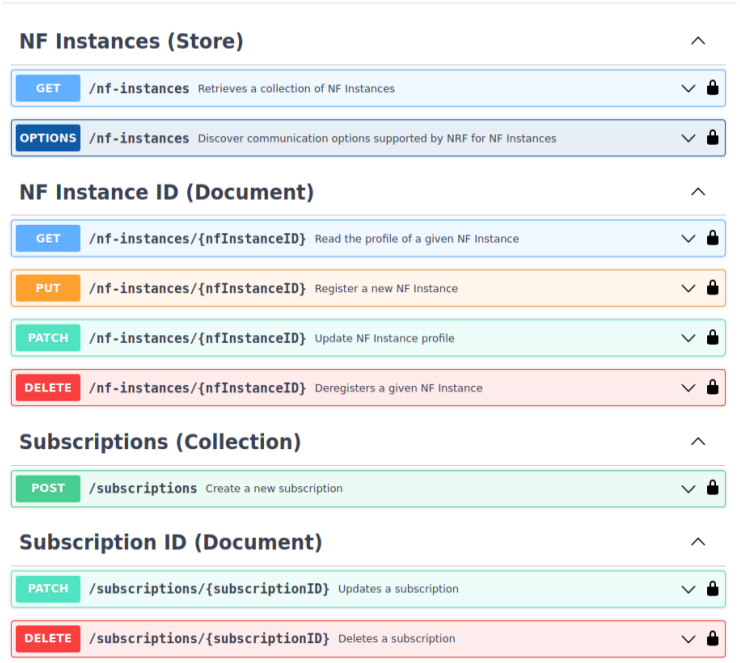

Figure 4. Imported schema for NRF NF management.

Recall that Open5GS and OAI employ different communication protocols for their Core Network deployment; however, it is important to note that both frameworks do not support TLS. While OAI supports HTTP/1.1 and HTTP/2 SBA communications, Open5GS only supports HTTP/2. As already pointed out, this disparity in protocol support leads to distinct testing approaches for the two frameworks:

- OAI Testing Approach: we utilized Burp Suite, which can receive input (via the 5GC API Parse plugin) in the form of a definition file containing the OpenAPI 3.0 schema of the API interfaces, and it delivers requests in the format prescribed by the 3GPP standard. The parameters of Burp Suite were configured to maximize the number of tests performed with the imported API schema, including the activation of the "HTTP smuggler" and "ActiveScan++" extensions to enhance the diversity of tests.
- Open5GS Testing Approach: Starting with the requests that were exported and executed on OAI, we employed the custom tool 5GHTTPModifier. This tool takes as input the path containing all the tests exported from Burp Suite and modifies the parameters, configuring them for the new 5G core. It then sends these requests after transforming them into the HTTP/2 format. Using this tool, we were able to execute identical tests as those performed on OAI, facilitating an objective comparison of the security levels between the two frameworks.

8.1. OAI API Injection Tests

Starting from the API schema imported through the appropriate plugin **(OpenAPI Parser)**, an automatic scan was initiated for each service.The requests sent through Burp Suite are reported in Table 4.

The testing of each API did not reveal any significant security vulnerabilities, and there were no instances of data breaches or disruptions. However, the tests did uncover some noteworthy points:

- `PUT-NF Instance ID (Document)`: Not all responses adhere to the standard JSON output format. Errors such as "request method does not exist," Java Errors, and header

errors were observed. In some cases, these errors provided overly informative details, including information about the library and programming language used, which could be valuable to potential attackers.

- `OPTIONS-NF Instances (Store)`: This particular API was found to be unimplemented, as every request simply yielded the response "Do some magic".
- `GET-NF Instances (Store)`: Requests containing SQL commands within the parameters were not rejected but instead accepted and ignored. Ideally, any requests that deviate from the specifications outlined in the schema should be rejected. For instance, if a parameter is declared as an integer but passed as a string, the entire API request should be rejected. Additionally, it was possible to input any value in optional parameters (such as the "limit" parameter in "nf-instances?nf-type&=""limit="), and the API would still return a valid response.

Table 4. Requests sent through Burp Suite.

Method	Description	Number of Requests
DELETE	Subscription ID (Document)	628
PATCH	Subscription ID (Document)	630
POST	Subscription (collection)	2398
DELETE	NF Instance ID (Document)	628
PUT	NF Instance ID (Document)	2987
PATCH	NF Instance ID (Document)	630
GET	NF Instance ID (Document)	761
GET	NF Instances (Store)	820
OPTIONS	NF Instances (Store)	550

In summary, the tests highlighted the following relevant issues:

- **Response Inconsistencies:** Some responses from the API did not conform to the standard JSON output format, revealing errors in request methods and providing excessive information about the implementation, which could be exploited by attackers.
- **Lack of Input Sanitization:** The APIs lacked proper input sanitization, enabling the acceptance of invalid parameters and SQL commands. This could lead to security vulnerabilities like SQL injection.
- **Invalid Method Handling:** Inconsistent handling of HTTP methods was observed. For instance, changing the method from GET to POST in certain APIs caused a crash of the NRF, leading to a potential DoS vulnerability.
- **Unimplemented APIs:** Some API services were not properly implemented and returned generic error messages or "magic" responses.

8.2. Open5GS API Injection Tests

To make a fair comparison between the two testbeds in spite of the limitations related to HTTP protocol, we developed the custom 5GHTTPModifier tool to transform the requests exported from Burp Suite for OAI into HTTP/2 requests compatible with Open5GS. The testing uncovered several security vulnerabilities within the Open5GS framework.

- SQL Injection: Requests containing SQL commands within the parameters are not rejected; instead, they are accepted and ignored. It is unclear whether there is a sanitization process in place. Regardless, all requests that do not conform to the schema specifications should be rejected. For example, if a parameter is declared as an integer but passed as a string, the entire API request should be rejected.
- `GET-NF Instance (Store)`: This API requires two parameters, "NFtype" and "Limit," but inconsistencies were discovered during testing:

  - Any non-existing NF entered in the "NFtype" parameter results in a response with all instances in the core network.
  - The "limit" parameter is declared as an integer, but no checks are performed on this parameter, allowing it to contain various data types, including OS/SQL commands or strings.
  - Changing the HTTP method from GET to POST in any request to this API causes the NRF to crash, leading to a DoS for the entire Core Network.
- `DELETE NF Subscription ID (Document)`: This API contains an error that causes the network function to crash. Failing to enter any value for the requested parameter leads to the NRF becoming unresponsive.
- `DELETE NF Instance ID (Document)`: Similar to the previous API, this one also has an error that can crash the network function if no value is entered for the requested parameter.
- `PUT-NF Instance ID (Document)`: During some tests, network functions were recorded with dummy parameters, suggesting a potential security flaw.
- `OPTIONS-NF Instances (Store)`: Every request made to this API crashes the NRF.

In summary, the tests lead to the following conclusions.

- **Response Inconsistencies:** Similar to OAI, some API responses did not adhere to the standard JSON output format, leading to information leakage about the implementation, which could be exploited by attackers.
- **Lack of Input Sanitization**: Similar to the OAI testbed, Open5GS suffered from insufficient input validation. Requests with SQL commands as parameters were accepted and ignored, potentially leading to security risks.
- **Improper Data Handling:** During testing, NFs were registered with dummy parameters, highlighting issues in the handling of data inputs.
- **Crash on Invalid Inputs:** Certain API calls, such as deleting NF subscriptions and instances, could cause the entire core network to crash when provided with invalid inputs.

8.3. General Security Considerations Drawn from API Injection Tests

Several key observations about common weaknesses emerged from the tests carried out in the two testbeds.

Both OAI and Open5GS have issues related to input sanitization, allowing for the acceptance of invalid or malicious parameters. Moreover, the inconsistencies in API responses and handling of HTTP methods point to a lack of proper API specification adherence and input validation: this absence of standardization in API responses could lead to interoperability issues when integrating NFs from different vendors. Despite the 3GPP specification's requirement for messages that are not valid according to the protocol to be rejected or discarded by the NF, both testbeds exhibited weaknesses in this area. Finally, the absence of proper TLS encryption within the SBI of the testbeds exposes vulnerabilities that could be exploited for various attacks.

In conclusion, while the open-source 5G projects Open5GS and OAI offer promising features for core network deployment, the API injection tests revealed security vulnerabilities that require attention. The results underscore the need for robust input validation, proper handling of invalid inputs, and adherence to standardized API responses to ensure the security and stability of 5G networks.

9. Final Remarks on Security Concerns

A detailed summary of the main problems identified in the two implementations of the API interfaces is provided in Figure 5.

Attack Type		Open5GS	OAI
Replay attack with sniffed packets (gNB UE registration) on AMF		1. NAS MAC Failure (Authentication Reject) with AMF Release Context 2. UE (legitimate user) can't associate communicate with 5G Core	1. AMF Crash
Replay attack with traffic from another PLMN network on AMF		1. Reject gNB association	1. AMF Crash
DDoS on AMF		1. HTTP/2 rejected connection - No rate limiting Mechanisms	1. No rate limiting Mechanisms
DDoS on NRF		1. HTTP/2 rejected connection - No rate limiting Mechanisms	1. No rate limiting Mechanisms
API Injection to NRF: TS 29.510_Nnrf_NFManagement_Rel16	***DELETE - Subscription ID (Document)***	1. NRF Crash if no ID is provided	1. no vulnerability found
	PATCH - Subcription ID (Document)	1. no vulnerability found	1. no vulnerability found
	POST - Subscription (collection)	1. no vulnerability found	1. no vulnerability found
	DELETE - NF Instance ID (Document)	1. NRF Crash if no ID is provided	1. no vulnerability found
	PUT - NF Instance ID (Document)	Some network function registerd with fiticious parameters (no parameter check on body values)	Errors with too much informations: (1) [json.exception.parse_error.101] parse error at line 1, column 1: syntax error while parsing value - unexpected end of input; expected '[', '{', or a literal. (2) [json.exception.parse_error.101] parse error at line 1, column 1: syntax error while parsing value - invalid literal; last read: '<'
	PATCH - NF Instance ID (Document)	1. no vulnerability found	1. no vulnerability found
	GET - NF Instance ID (Document)	1. no vulnerability found	1. no vulnerability found
	GET - NF Instances (Store)	1. Returns all NFs if an existing NF is not entered. 2. Parameter limit not checked. 3. NRF crash with method change from GET to POST.	1. Parameter limit not checked.
	OPTIONS - NF Instances (Store)	1. every tested request will crash the NRF	1. not implemented

Figure 5. OAI vs. Open5GS vulnerability assessment comparison.

To conclude the paper we would like to provide a high-level view of the security issues, highlighting also architectural aspects of the implementations:

- Requests to the NRF can be sent by any machine that *can reach* it, i.e., there is no **authentication token** check to verify identity, and thus authorization, to access the network and services. Relying on perimeter security (firewalls and traffic closures) to secure an NF is a weak approach; instead, we must assume a zero-trust design, assuming that anyone can contact the NFs. A `DELETE NF Instance` by any NF within the core would result in a DoS attack.
- **Traffic not encrypted in SBI:** in the core 5G, another high-impact issue is the interception of SBA traffic. In both Open5GS and OAI, traffic in the SBI is not encrypted, exposing the core to replay attacks, API injections, and other attacks.
- **No rate-limiting** on requests to the AMF or NRF: in fact, all requests are answered. This means that no security mechanism is in place to mitigate a DDoS attack, like a SecGW might be.
- No control mechanism (**filtering**) on user-set values: reliance is placed on client-side filtering when control should also be carried out server-side.
- **Replay attacks on N1/N2 interface** successfully executed: no security mechanism (sequence number) inserted to counter this attack.

Finally, it is worth mentioning that the implementation of the considered frameworks is still evolving and some features might be included only temporarily. However, the analysis methodology presented in the paper can be applied with minor changes to any new release and will permit tracking the evolution of open-source software for the 5G core network.

Author Contributions: Conceptualization, F.D., R.G.G. and M.P.; investigation, F.D.; resources, F.D.; writing—original draft preparation, R.G.G. and M.P.; writing—review and editing, R.G.G. and M.P.; supervision, R.G.G. and M.P.; funding acquisition, R.G.G. and M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence) and by the University of Pisa in the framework of the PRA_2022_64 "hOlistic Sustainable Management of distributed softWARE systems (OSMWARE)" project.

Data Availability Statement: The GitHub repository [6] contains traffic captures, scripts, and results of the tests described in this paper.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Acronyms

The following acronyms (listed in alphabetic order) are used in this manuscript:

5GC	Fifth-Generation Core Network
5G	Fifth-Generation
AMF	Access and Mobility Management Function
API	Application programming interface
CAPIF	Common API Framework
CP	Control Plane
DDoS	Distributed Denial of Service
DN	Data Network
EPC	Evolved Packet Core
gNB	gNodeB (Next-Generation Node B)
GUTI	Globally Unique Temporary ID
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things

IPsec	IP Security
MCC	Mobile Country Code
MEC	Multiaccess Edge Computing
MNC	Mobile Network Code
MNO	Mobile Network Operator
MSIN	Mobile Subscriber Identification Number
N3IWF	Non-3GPP Interworking Function
NAS	Non-Access Stratum
NEF	Network Exposure Function
NF	Network Function
NRF	Network Repository Function
NSI	Network Specific Identifier
OAI	Open-Air Interface
RAN	Radio Access Network
SBA	Service-Based Architecture
SBI	Service-Based Interface
SecGW	Security Gateway
SEPP	Security Edge Proxy Protection
SIM	Subscriber Identity Module
SMC	Security Mode Command
SMF	Session Management Function
SUCI	Subscription Concealed Identifier
SUPI	Subscriber Permanent Identifier
TLS	Transport Layer Security
UE	User Equipment
UPF	User Plane Function
UP	User Plane
WAF	Web Application Firewall

References

1. Witkowski, D. *Bridging the Gap: 21st Century Wireless Telecommunications Handbook, 2nd ed.*; Independent Publishing Platform: Joint Venture Silicon Valley, CA, USA, 2019.
2. CISCO. CISCO Annual Internet Report. 2020. Available online: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed on 16 April 2023).
3. Open5GS. Available online: https://open5gs.org/ (accessed on 30 September 2023).
4. Openairinterface: The Fastest Growing Community and Software Assets in 5g Wireless. Available online: https://openairinterface.org/ (accessed on 30 September 2023).
5. Wadatkar, P.V.; Garroppo, R.G.; Nencioni, G. 5G-MEC Testbeds for V2X Applications. *Future Internet* **2023**, *15*, 175. [CrossRef]
6. Dolente, F. Security Analysis of 5G Core Network. Available online: https://github.com/Spartan-F117/VAPT_CoreNetwork5G (accessed on 30 September 2023).
7. Nencioni, G.; Garroppo, R.G.; Olimid, R.F. 5G Multi-Access Edge Computing: A Survey on Security, Dependability, and Performance. *IEEE Access* **2023**, *11*, 63496–63533. [CrossRef]
8. Bozorgchenani, A.; Zarakovitis, C.C.; Chien, S.F.; Lim, H.S.; Ni, Q.; Gouglidis, A.; Mallouli, W. Joint Security-vs-QoS Framework: Optimizing the Selection of Intrusion Detection Mechanisms in 5G Networks. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022. [CrossRef]
9. Salazar, Z.; Zaïdi, F.; Nguyen, H.N.; Mallouli, W.; Cavalli, A.; de Oca, E.M. A Network Traffic Mutation Based Ontology, and Its Application to 5G Networks. *IEEE Access* **2023**, *11*, 43925–43944. [CrossRef]
10. Mahyoub, M.; AbdulGhaffar, A.; Alalade, E.; Ndubisi, E.; Matrawy, A. Security Analysis of Critical 5G Interfaces. *TechRxiv* **2023**. [CrossRef]
11. Park, S.; Kim, D.; Park, Y.; Cho, H.; Kim, D.; Kwon, S. 5G Security Threat Assessment in Real Networks. *Sensors* **2021**, *21*, 5524. [CrossRef] [PubMed]
12. Park, S.; Kwon, S.; Park, Y.; Kim, D.; You, I. Session Management for Security Systems in 5G Standalone Network. *IEEE Access* **2022**, *10*, 73421–73436. [CrossRef]
13. Basin, D.; Dreier, J.; Hirschi, L.; Radomirovic, S.; Sasse, R.; Stettler, V. A Formal Analysis of 5G Authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; CCS '18; pp. 1383–1396.

14. Hussain, S.R.; Echeverria, M.; Karim, I.; Chowdhury, O.; Bertino, E. 5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; CCS '19; pp. 669–684.
15. Yang, T.; Wang, S.; Zhan, B.; Zhan, N.; Li, J.; Xiang, S.; Xiang, Z.; Mao, B. Formal Analysis of 5G Authentication and Key Management for Applications (AKMA). *J. Syst. Archit.* **2022**, *126*, 102478. [CrossRef]
16. Akon, M.; Yang, T.; Dong, Y.; Hussain, S.R. Formal Analysis of Access Control Mechanism of 5G Core Network. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, Melbourne Australia, 10–14 July 2023; CCS '23; pp. 666–680. [CrossRef]
17. Group, N.C.C. The Challenges of Fuzzing 5G Protocols. *5G Secur. Smart Environ.* **2021**. Available online: https://research.nccgroup.com/2021/10/11/the-challenges-of-fuzzing-5g-protocols/ (accessed on 5 October 2023).
18. Salazar, Z.; Nguyen, H.N.; Mallouli, W.; Cavalli, A.R.; Montes de Oca, E. 5Greplay: A 5G Network Traffic Fuzzer—Application to Attack Injection. In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; ARES '21. [CrossRef]
19. 5GReplay. Available online: https://5greplay.org/docs.html (accessed on 30 September 2023).
20. Dumitru-Guzu, O.M.; Vlădeanu, C. Analysis of Potential Threats in NextGen 5G Core. In Proceedings of the 2022 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 10–11 November 2022; pp. 1–4. [CrossRef]
21. 3GPP. *TS 23.501 V16.4.0; System Architecture for the 5G System (5GS)*; Stage 2 (Release 16); ETSI: Sophia Antipolis, France, 2020.
22. ETSI. *TS 129 501 V16.4.0 Principles and Guidelines for Services Definition*; Technical Report; ETSI TS 129 501 Version 16.4.0; European Telecommunications Standards Institute (ETSI): Sophia Antipolis, France, 2020.
23. Swagger. OpenAPI Specification Version 3.0.3. Available online: https://swagger.io/specification/v3/ (accessed on 30 September 2023).
24. 3GPP. OpenAPIs for the Service-Based Architecture. Available online: https://www.3gpp.org/technologies/openapis-for-the-service-based-architecture (accessed on 30 September 2023).
25. OAuth 2.0. Available online: https://oauth.net/2/ (accessed on 30 September 2023).
26. Polese, M.; Bonati, L.; D'Oro, S.; Basagni, S.; Melodia, T. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1376–1411. [CrossRef]
27. Nair, P. *Securing 5G and Evolving Architectures*; ADDISON WESLEY Publishing Company Incorporated: Boston, MA, USA, 2021.
28. ETSI. *TS 133 501—V17.8.0—5G; Security Architecture and Procedures for 5G System*; Technical Report; 3GPP TS 33.501 Version 17.8.0 Release 17; ETSI: Sophia Antipolis, France, 2022.
29. Akash Tripathi, A.J.W. 5G Network Security Threats and 3GPP Security Mechanisms. Available online: https://techblog.comsoc.org/2022/01/01/5g-network-security-threats-and-3gpp-security-mechanisms/ (accessed on 23 January 2023).
30. Palamà, I.; Gringoli, F.; Bianchi, G.; Blefari-Melazzi, N. IMSI Catchers in the wild: A real world 4G/5G assessment. *Comput. Netw.* **2021**, *194*, 108137. [CrossRef]
31. Chlosta, M.; Rupprecht, D.; Pöpper, C.; Holz, T. 5G SUCI-Catchers: Still Catching Them All? In Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Virtual Event, 28 June–2 July 2021; WiSec '21; pp. 359–364. [CrossRef]
32. Nie, S.; Zhang, Y.; Wan, T.; Duan, H.; Li, S. Measuring the Deployment of 5G Security Enhancement. In Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks, San Antonio, TX, USA, 16–19 May 2022; WiSec '22; pp. 169–174. [CrossRef]
33. Lasierra, O.; Garcia-Aviles, G.; Municio, E.; Skarmeta, A.F.; Costa-Pérez, X. European 5G Security in the Wild: Reality versus Expectations. In Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2023, Guildford, UK, 29 May–1 June 2023; Boureanu, I., Schneider, S., Reaves, B., Tippenhauer, N.O., Eds.; ACM: New York, NY, USA, 2023; pp. 13–18. [CrossRef]
34. 3GPP. *3GPP TS 23.003 V15.11.0, Numbering, Addressing and Identification (Release 15)*; ETSI: Sophia Antipolis, France, 2021.
35. Lemes, M.T.; Alberti, A.M.; Both, C.B.; De Oliveira Júnior, A.C.; Cardoso, K.V. A Tutorial on Trusted and Untrusted Non-3GPP Accesses in 5G Systems—First Steps Toward a Unified Communications Infrastructure. *IEEE Access* **2022**, *10*, 116662–116685. [CrossRef]
36. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
37. Abbasi, A.; Wetzels, J.; Holz, T.; Etalle, S. Challenges in designing exploit mitigations for deeply embedded systems. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 17–19 June 2019; pp. 31–46.
38. Molin, A.; Riviš, A.M.; Marinescu, R. Assessing the real impact of open-source components in software systems. *IEEE Access* **2023**, *11*, 111226–111237. [CrossRef]
39. Yun, I.; Min, C.; Si, X.; Jang, Y.; Kim, T.; Naik, M. {APISan}: Sanitizing {API} Usages through Semantic {Cross-Checking}. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 363–378.
40. Keman, H.; Madnick, S.; Pearlson, K. Is Third-Party Software Leaving You Vulnerable to Cyberattacks? *Harv. Bus. Rev.* **2021**. Available online: https://hbr.org/2021/05/is-third-party-software-leaving-you-vulnerable-to-cyberattacks (accessed on 14 December 2023).

41. Stradowski, S.; Madeyski, L. Exploring the challenges in software testing of the 5G system at Nokia: A survey. *Inf. Softw. Technol.* **2023**, *153*, 107067. [CrossRef]
42. PortSwigger. Available online: https://portswigger.net/burp/documentation (accessed on 14 September 2023).
43. Køien, G.M. On Threats to the 5G Service Based Architecture. *Wirel. Pers. Commun.* **2021**, *119*, 97–116. [CrossRef]
44. Shen, M.; Ye, K.; Liu, X.; Zhu, L.; Kang, J.; Yu, S.; Li, Q.; Xu, K. Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 791–824. [CrossRef]
45. Bousalem, B.; Silva, V.F.; Langar, R.; Cherrier, S. DDoS Attacks Detection and Mitigation in 5G and Beyond Networks: A Deep Learning-based Approach. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 1259–1264. [CrossRef]
46. MatrixTM. MHDDoS—DDoS Attack Script with 56 Methods. Available online: https://github.com/MatrixTM/MHDDoS (accessed on 30 September 2023).
47. Burp Suite. Available online: https://portswigger.net/burp (accessed on 30 September 2023).
48. 3GPP. *3GPP TS 33.512 V16.6.0; 5G Security Assurance Specification (SCAS); Access and Mobility Management Function (AMF) (Release 16)*; ETSI: Sophia Antipolis, France, 2021.
49. Hu, X.; Liu, C.; Liu, S.; You, W.; Li, Y.; Zhao, Y. A systematic analysis method for 5g non-access stratum signalling security. *IEEE Access* **2019**, *7*, 125424–125441. [CrossRef]