Evaluating SCAS Tests in Closed Source 5G Core Network

Leonardo Sagratella[1,*], Andrea Bernardini[1,†], Francesco D'Alterio[1,†], Marina Settembre[1,†] and Roberta Terrugia[2]

[1] *Fondazione Ugo Bordoni, Viale del Policlinico 147, 00161, Rome, Italy*
[2] *RSE S.p.A., Via Raffaele Rubattino, 54, 20134, Milan, Italy*

Abstract

5G networks offer faster speeds, lower latency, and the potential to serve a wide range of vertical applications and use cases. However, the growing complexity of 5G infrastructures, driven by the proliferation of physical and virtual components, also expands the attack surface and raises cyber risks. As a result, there is a pressing need for innovative and efficient strategies to manage and optimize the deployment of 5G assurance frameworks. The main focus of the paper deals with the security assurance of a commercial closed source 5G core network deployed in energy test facilities for research activities, leveraging the 5G Security Assurance Specifications (SCAS) as outlined by 3GPP. For that purpose, we propose (1) a test selection pipeline that identifies the most relevant tests for the target network, and (2) a proxy-based approach for conducting SCAS testing on a commercial 5G core network with limited access to the network interfaces. Our approach utilizes a custom SCTP (Stream Control Transmission Protocol) proxy to intercept and modify network traffic. This method allows us to conduct SCAS tests without needing direct access to the core network components and enables a focused analysis of inputs, outputs, and the system's overall behavior. This evaluation revealed several potential issues that require confirmation from the vendor, underscoring the need for further investigation into the validity of identified vulnerabilities.

Keywords

5G, 3GPP SCAS, Security Assurance Specification, Vulnerability Assessment, GSMA, NESAS, SCTP proxy

1. Introduction

As 5G becomes the backbone for industries and critical national infrastructure, ensuring its security is paramount to guarantee to stakeholders that the network is resilient against cyber threats, supports critical applications, and protects data confidentiality, integrity, and availability. 5G security assurance refers to the systematic processes and frameworks designed to evaluate, validate, and maintain the security of 5G networks, their equipment, and their associated infrastructure[1]. This necessitates a global, standardized framework for evaluating and assuring the security of 5G network equipment. To address this challenge, the Network Equipment Security Assurance Scheme (NESAS), developed collaboratively by the 3rd Generation Partnership Project (3GPP) and the Global System for Mobile Communications Association (GSMA), offers a comprehensive, industry-wide framework. NESAS serves as a globally recognized benchmark to assess and certify the security of 5G network equipment and associated processes. By establishing both evaluation methodologies and consistent security requirements, the NESAS framework provides a baseline for 5G security assurance.

This study delves into the practical application of security testing on a private closed source 5G core network infrastructure. Specifically, we conducted a series of tests using the SCAS tests on a proprietary and commercially available platform, henceforth named the ACME 5G Core, that presents unique challenges due to its restricted access and limited transparency into its internal components and interfaces. This work aims to address these challenges by proposing a methodological framework with test selection criteria and a proxy-based test execution approach to systematically evaluate the security of this closed network infrastructure.

The remainder of this article is organized as follows:

Joint National Conference on Cybersecurity (ITASEC & SERICS 2025), February 03-8, 2025, Bologna, IT

[*] Corresponding author.

[†] These authors contributed equally.

Section 2 provides an overview of the relevant background and related works. Section 3 outlines our test plan, discussing the challenges associated with performing SCAS tests on a closed source network, and presenting criteria for selecting executable SCAS tests based on available interfaces and configurations. Section 4 describes the methodology used to implement SCAS tests using a proxy-based approach. This section details the development of an SCTP proxy to intercept, modify, and analyze network traffic without requiring direct access to the core network's components. Section 5 presents the results of our test execution on the ACME Core Network, including a comprehensive vulnerability assessment and the discovery of potential security issues that require further investigation by the vendor. Finally, Section 6 summarizes our findings and suggests future research directions to improve the automation and depth of SCAS testing for closed source 5G core networks.

2. Background and Related Work

The 3GPP Security Assurance Methodology (SECAM) [2] and its associated Security Assurance Specifications (SCAS) are tools designed to address the challenges of securing telecommunications networks. As outlined by 3GPP, these frameworks aim to ensure that network products meet stringent security requirements through a systematic approach that includes threat modeling, risk analysis, test case development, and independent verification.

The SECAM outlines a systematic process to provide complete security assurance throughout the life-cycle of a network product. It is composed of various phases, including vendor network product development assessment, network product life-cycle management process evaluation, security compliance testing, and basic vulnerability testing. This methodology ensures that all critical aspects of 5G network security are scrutinized in a structured and methodical way during the product life-cycle.

Along with SECAM, 3GPP released the Security Assurance Specifications (SCAS), a set of security documents to delineate security requirements and associated test cases for specific 5G network product classes [3]. Each SCAS document is created to address the unique security needs of a particular network product class, ensuring that all essential security functionalities are accounted for. The primary objective of SCAS is to determine whether network products are compliant with established security standards. Upon successful completion of these tests, it can be inferred that the network element satisfies the security requirements for the given product class.

However, this compliance does not guarantee an absence of limitations within the SCAS framework itself [4]. One such limitation pertains to the descriptive nature of each test, which is presented as a series of objectives, preconditions, steps, and evidence in textual form.

These detailed yet interpretive test descriptions, facilitate understanding of the tasks that need to be performed ("*what to do*") to execute the test. However, they complicate the comprehension of the procedural steps ("*how to do*") required for effective execution. Moreover, the test description often lacks specificity regarding the resource costs, both in time and money, required for implementation and execution [1].

Although this flexibility enables evaluators to choose suitable tools aligned with vendor-specific implementations, it simultaneously compromises the replicability of tests across various environments, making it challenging to ensure consistent and reliable security assessments.

The fragmented nature of 5G deployments, often with different vendors developing various network functions, creates significant difficulties when testing systems involving multiple vendors. This diversity makes both the implementation and the security of these systems more challenging, as they require coordination and standardization across different proprietary technologies. To address these challenges, ScasDK [5] represents a relevant approach. By providing a unified interface, ScasDK simplifies the configuration and control of responses from different network functions, thus making security assurance processes more manageable in complex 5G environments.

The novelties in ScasDK are its protocol-specific proxies, each designed to intercept, modify, and extract data according to the protocol it handles and test logic. By using proxies, ScasDK can implement custom behaviors for individual components without necessitating modifications to their source code.

The proxies operate transparently by forwarding packets while incorporating hooks to execute operations on received messages before forwarding them. This dual functionality makes them indispensable tools for conducting detailed security assessments that require dynamic packet manipulation.

Mancini et al. [6] introduce a black-box fuzzing framework specifically tailored for assessing the security of the AMF. Their primary objective is to develop and validate a fuzzing tool that can effectively identify vulnerabilities in the AMF. The authors emphasize the necessity of protocol fuzzing, particularly in environments where access to source code is limited, a common scenario in commercial deployments. The study presents a proof-of-concept implementation that has been validated against three open source 5G core network implementations. Their methodology involves crafting malformed messages and injecting them into the AMF using the ScasDK framework to observe its responses, thus uncovering various implementation bugs. The results demonstrate the efficacy of their black-box fuzzing approach in identifying security vulnerabilities without requiring access to the internal architecture or source code.

3. Test Plan

To address the unique challenges posed by executing SCAS tests on a closed source network like the ACME 5G Core, a comprehensive test plan is essential. This involves developing methodologies to refine and select executable SCAS tests from the extensive set proposed by 3GPP, given the constraints of available interfaces and functionalities. The subsequent sections detail the specific challenges encountered during this process, and the criteria used for selecting relevant SCAS tests.

3.1. The challenge of SCAS testing on a closed source network

The ACME 5G Core comes as a Linux virtual appliance, installed in a VM. It is composed of two main components: Control Plane (CP), organized into modular components; and User Plane (UP), both implemented as Open Container Initiative (OCI) containers. A web interface supports the GUI for operators to configure and monitor the system and an API for programming access. The ACME 5G Core exposes the following 3GPP interfaces:

- **N1:** NAS over N2 between the gNBs and the AMF
- **N2:** NGAP over SCTP between the gNBs and the AMF
- **N3:** GTP-U over UDP between the gNBs and the UPFs
- **N6:** IP connectivity to external Data Networks (DN)

Figure 1 depicts the ACME's infrastructure, the red boxes highlight the exposed interfaces. Due to the closed source nature of the core network and the limited number of exposed interfaces, it is impossible to execute all the SCAS tests proposed by 3GPP. Therefore, a methodology is required to refine the tests set by selecting a subset of executable tests based on some given criteria. The selection process should exclude tests that rely on interfaces or network functions not available in the 5G Core under exam.

3.2. Test selection criteria

The methodology for selecting executable SCAS tests comprises a process that ensures the identification of pertinent security tests based on established criteria. The selection process starts with all the SCAS available documents and is divided into two phases.

Initially, during the **Document Selection** phase, general criteria are applied to the stack of SCAS documents. These criteria are used to identify pertinent documents that are relevant to the scope of the testing objectives and to filter out irrelevant documents. Figure 2 shows the test selection methodology. For this evaluation, only SCAS documents pertinent to Network Functions that utilize the available interfaces were retained.

As a result, three documents were selected:

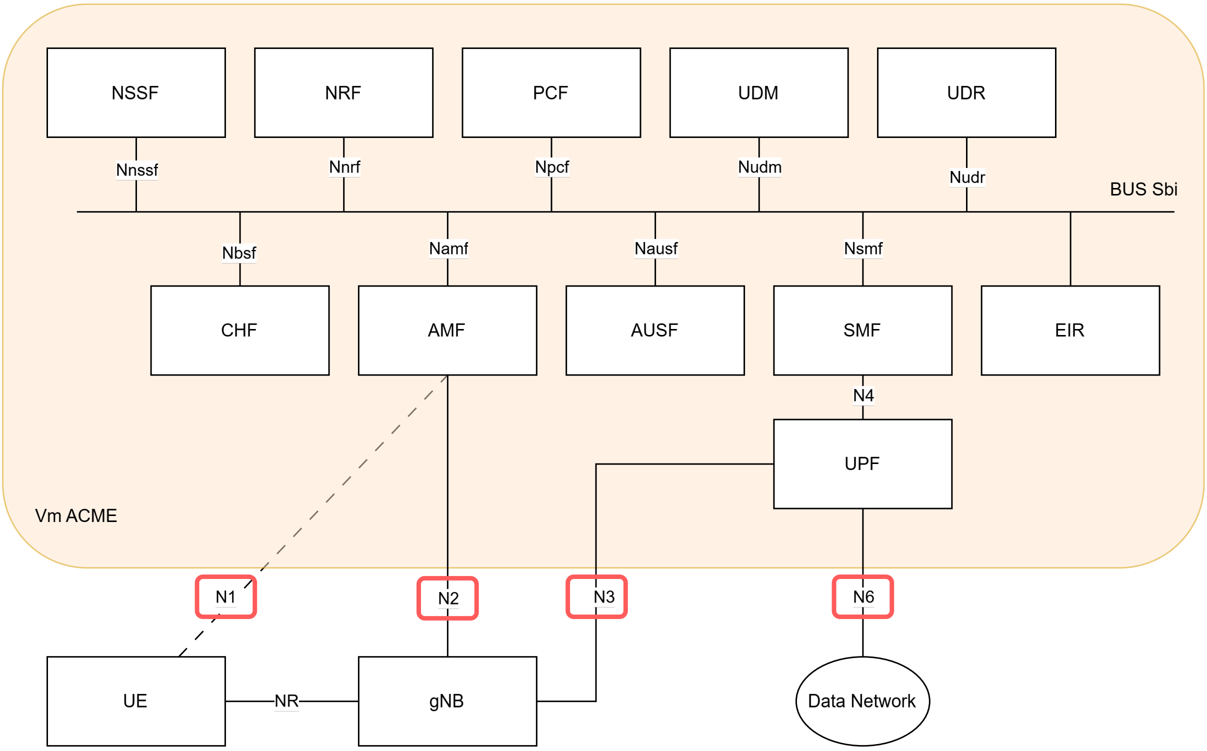

Figure 1: ACME Core architecture and exposed interfaces.

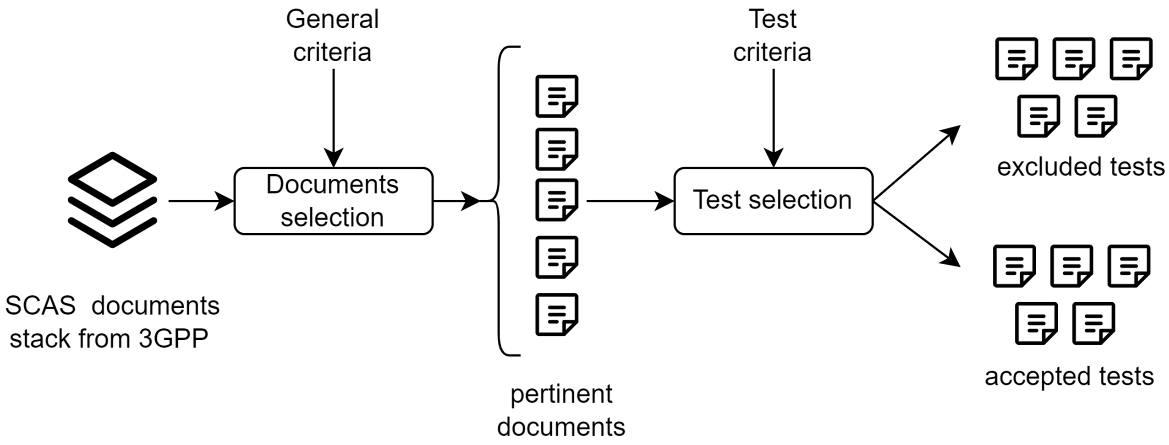

Figure 2: Test selection criteria.

- **TS 33.117:** Generic tests applicable on any NF [7].
- **TS 33.512:** SCAS tests specific to AMF [8].
- **TS 33.513:** SCAS tests specific to UPF [9].

The TS 33.117 document includes test cases derived from both 3GPP security requirements and generic security requirements. For this evaluation, only tests derived from 3GPP security requirements were selected, as they are directly pertinent to the 5G environment.

The second phase, **Test Selection**, refines the focus further by evaluating the pertinent documents identified in the previous phase.

Each test from the selected documents was thoroughly analyzed, with particular attention given to the prerequisites and procedural steps. The following exclusion criteria were applied:

- Tests requiring access to interfaces that are not available.

- Tests requiring the interaction with Network Functions that are not available or behind unavailable interfaces.
- Tests requiring configurations that are not supported in ACME core.

Table 1
Final set of selected tests.

Test Name	SCAS Document
TC_BVT_PORT_SCANNING	33.117
TC_BVT_VULNERABILITY_SCANNING	33.117
TC_RES_STAR_VERIFICATION_FAILURE case A, B	AMF
TC_AMF_NAS_INTEGRITY_FAILURE case 1, 2	AMF
TC_NAS_REPLAY_AMF	AMF
TC_NAS_INT_SELECTION_USE_AMF	AMF
TC_UE_SEC_CAP_HANDLING_AMF	AMF
TC_UE_SEC_CAPS_AS_CONTEXT_SETUP	AMF

The results of the test selection process are summarized in Table 1. The list of excluded tests, along with the reasons for their exclusion, is provided in Appendix A in Tables 2, 3, and 4.

The tests were implemented using a gradual approach. Beginning with tests that were simpler to implement, priority was given to those requiring minimal setup and execution effort. Subsequently, tests with higher levels of complexity were addressed, ensuring a systematic progression that facilitated thorough validation at each stage.

We successfully implemented all the selected SCAS tests. Specifically, this includes 6 out of 16 available tests for the AMF, none out of 8 available tests for the UPF, and 2 out of 7 available 3GPP-specific tests from 33.117.

4. SCAS Implementation approach

Analyzing in detail the steps required to execute the tests, it is evident that the tester needs to alter the default behavior of a network component when necessary. One possible way to achieve this is to modify the component's source code to align with the desired functionality. However, this approach requires an open source code and a detailed knowledge of the code implementation and language used. Moreover, it could be necessary to generate multiple versions of the same component with different behavior.

Mancini et al. [5] proposed a novel approach for the execution of SCAS tests using a proxy. Their approach is effective in altering the component's behavior without modifying its source code. Moreover, the proxy approach can also be effective for executing tests with closed source components, as long as they can connect to a different component. For example, to effectively use a proxy for the gNB and AMF, both components must accept a new connection from the proxy.

The proxy acts as an intermediary component between two network elements, with the possibility to log, forward, and edit messages that transit through. It is the core of the testing framework, implementing different test logic required by SCAS tests. The proxy will look like a gNB from the AMF perspective and will resemble an AMF from the gNB perspective.

The authors developed three different proxies for the HTTP, SCTP, and PFCP protocols. However, given the limitation of ACME Core exposed interfaces, for this work, it will be used only a proxy for the SCTP protocol, the one used to carry NGAP and NAS messages between gNB and AMF. The details for the proxy implementation are shown in the next section.

4.1. SCTP Proxy

The SCTP proxy was developed from scratch using Go as a coding language.This choice was motivated by the fact that Go is also utilized in free5gc, an open-source 5G Core network platform. Consequently,

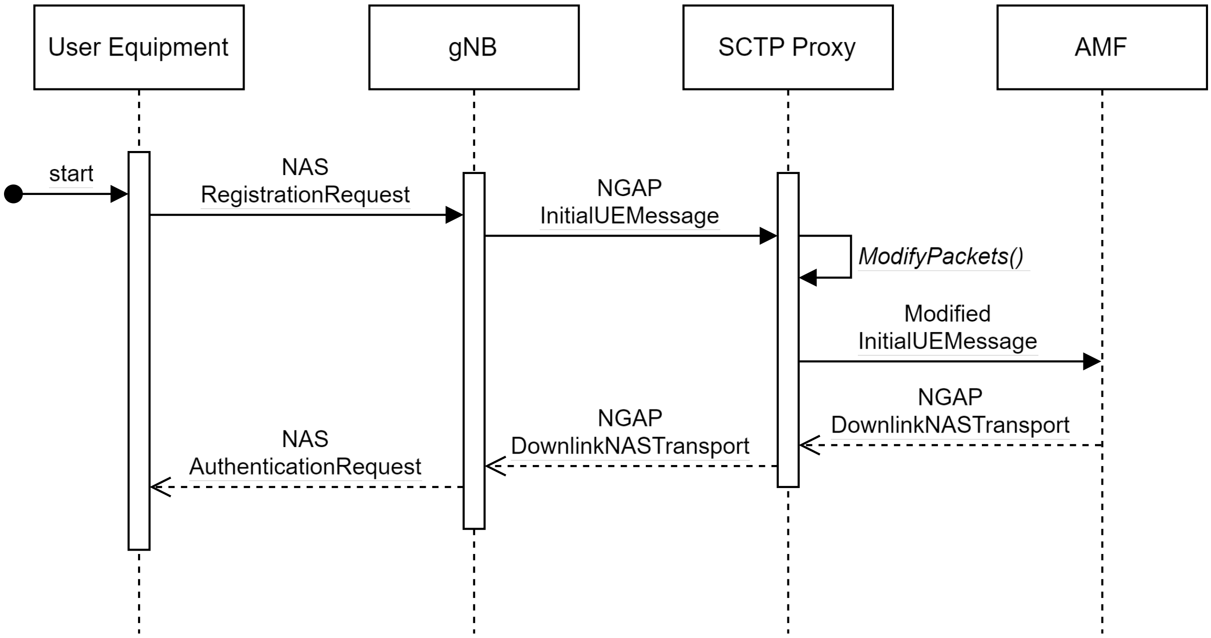
User Equipment
gNB
SCTP Proxy
AMF
start
NAS RegistrationRequest
NGAP InitialUEMessage
ModifyPackets()
Modified InitialUEMessage
NGAP DownlinkNASTransport
NGAP DownlinkNASTransport
NAS AuthenticationRequest

Figure 3: Proxy sequence diagram.

numerous essential 5G functionalities, including NGAP and NAS decoding, have already been implemented, facilitating integration and development efficiency. Additionally, Go is a powerful and versatile language that enables easy scaling without significant performance loss.

The proxy operates as a non-transparent proxy and thus is visible in network traffic and packet captures. A transparent version is planned for future development.

The minimum requirements for the proxy to work are:

- The AMF must permit the addition of a new gNB to the core network.
- The gNB must allow for the configuration of the AMF's IP address.

The proxy is built using an implementation of the SCTP server provided by the repository [10] and operates between the AMF and the gNB.

A *testStruct* interface is created to represent a generic interface for a SCAS test; it defines the method *ModifiyPacket*. Each SCAS test must be implemented as a distinct *struct* that implements the *testStruct* interface.

The proxy takes as input parameter a *struct* ***TestS*** that implements the *testStruct* interface; during test execution, packets received from both sides are processed by the ***TestS*** *ModifyPacket* function which encapsulates the test logic.

At startup, an SCTP socket is created, listening to a specified IP and port from the command line arguments. When the gNB connects, the proxy establishes a new SCTP connection with the AMF. Once connected, every SCTP packet exchanged between the AMF and the gNB is intercepted, forwarded, and potentially modified by the proxy based on the test logic. All packets go through the *ModifyPacket* function, where they are analyzed and altered as needed before being sent.

Figure 3 depicts a portion of a sequence diagram during the UE authentication phase. The picture shows how the packets are routed and modified by the *ModifyPacket* function, only when required by test logic. In the example shown, the proxy modifies only the *InitialUeMessage* message from gNB to AMF and not its response. Additionally, the proxy can save the network trace in a packet capture file, useful for later analysis and as evidence for test results.

4.2. Local setup

The laboratory used for test execution is composed of a physical host connected directly to the same AMF subnet, running Ubuntu 22. The proxy is installed directly on the test host. Figure 4 depicts the

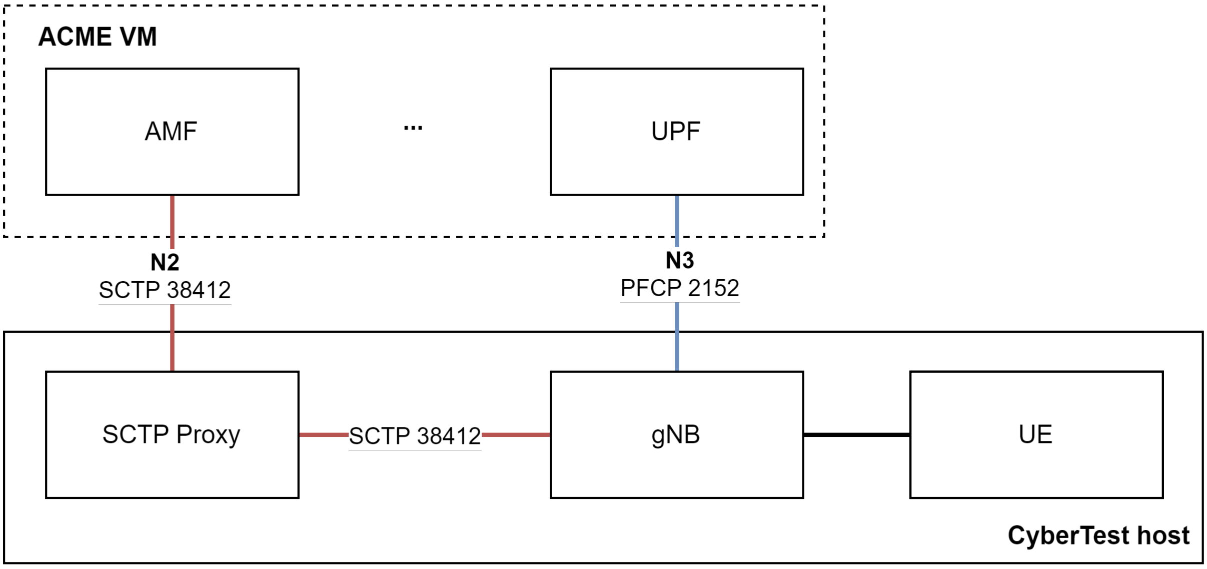

Figure 4: Laboratory network architecture.

laboratory topology. UERANSIM v2.3.6 is used to simulate gNB and UE functionalities.

5. Results

All the selected SCAS tests conducted on the ACME 5G Core network were completed and the evidences were gathered. This outcome indicates that the AMF adheres to the security standards established by the 3GPP, at least for the executed tests. The successful execution of these tests underscores the robustness of the security protocols and mechanisms implemented by AMF, providing a strong foundation for the secure deployment and operation of the 5G core infrastructure.

5.1. Vulnerability assessment

One of the SCAS tests conducted involved a comprehensive vulnerability assessment aimed at identifying known vulnerabilities within the network. The assessment was performed using Nessus Community Edition through an unauthenticated scan. Unauthenticated scans are more susceptible to false positives compared to authenticated scans, as they rely solely on banner information to infer software versions, whereas authenticated scans can directly access the machine and provide a detailed inventory of installed software versions. To verify whether a reported vulnerability is a false positive, the tester must access the system directly and manually verify the installed software version. However, in the case of the ACME network, this verification is not feasible due to their constraints.

The vulnerability scan identified a total of 16 potential vulnerabilities. Only vulnerabilities with significant severity levels (above Medium) will be addressed in this evaluation:

- **CVE-2024-6387:** Nessus identified potential vulnerabilities in OpenSSH, specifically CVE-2024-6387. Ubuntu has released security advisories detailing the updated versions of OpenSSH that mitigate these vulnerabilities. For instance, version 9.6p1-3ubuntu13.3 and earlier revisions such as 8.9p1-3ubuntu0.10 address the CVE-2024-6387. Despite these updates, Nessus cannot verify the actual version of the software installed and its revision by examining the service response banners; it can only estimate the version of the service. Consequently, it flagged the system running OpenSSH version 8.9 as vulnerable and recommended an upgrade to version 9.6. Despite this flagging, the vulnerability may have already been mitigated through backported patches. Backport patches are updates applied to older versions of software, typically backported from newer versions, to address security vulnerabilities, bugs, or compatibility issues without requiring a full upgrade. Therefore, the 8.9 version could contain the backported fix from version 9.6, thus version 8.9 could be not vulnerable. To confirm the actual patch status, a local inspection of the installed OpenSSH version is necessary.

- **CVE-2016-2183:** The scan revealed that certain hosts support weak SSL/TLS cipher suites, including ECDHE-RSA-DES-CBC3-SHA and DES-CBC3-SHA, which provide medium-strength encryption. These ciphers are considered vulnerable to modern cryptographic attacks. Nessus recommends disabling the use of these ciphers to enhance security.
- **CVE-2024-1442:** A critical vulnerability was identified in the Grafana instance related to insecure permissions settings. This vulnerability could allow unauthorized users to gain access to sensitive data and perform malicious actions within the Grafana dashboard. No online information suggests that a backport patch has been implemented to mitigate this vulnerability; Ubuntu's security advisory website does not indicate any updates for any version of the software. The recommended action is to update Grafana to a non-vulnerable version using official updates, following the security advisory released by Grafana.
- **False Positives and Minor Vulnerabilities:** The remaining vulnerabilities detected in the scan are either minor issues or are identified as false positives.

5.2. Insecure Passwords

The TS 33.117 mandates the need to evaluate password policies and their robustness within network systems. In the context of the ACME 5G Network Platform, it was discovered that modifying or enforcing password policies is not feasible. Consequently, the assessment focused on identifying services that use weak usernames and passwords through dictionary-based attacks using Hydra.

Hydra is a powerful tool designed to perform dictionary and brute-force attacks across various protocols, including HTTP, FTP, SSH, and others.

The scan revealed a single SSH service employing an insecure password. However, this SSH service pertains to an antenna rather than the Core system under examination; consequently, apart from recommending its modification, no further action is necessary.

6. Conclusion

This paper investigates the security evaluation in a closed source 5G Core Network using the 3GPP Security Assurance Specifications (SCAS) tests. The study addresses the challenges of testing a commercial closed source 5G Core Network deployed in energy test facilities, which limits direct access to its components and interfaces. We proposed a methodological framework with test selection criteria and a proxy-based test execution approach to systematically evaluate the security of this closed network infrastructure. The selected SCAS tests were executed successfully, providing valuable insights into the current security posture of the ACME 5G platform under examination.

Given the constraints posed by the proprietary nature of the network, it is important to remark that not all SCAS tests could be conducted, thus leaving the security assurance coverage as partial.

The vulnerability assessment revealed some potential issues that require confirmation from the vendor. Since the network operates on a closed source framework, we cannot independently determine whether these vulnerabilities are genuine or false positives. Therefore, collaboration with the vendor is crucial to verify their validity.

A challenge encountered during test execution is the necessity for manual intervention, which specifically requires a restart of the proxy, gNB, and UE along with modifications to the test parameters for each test run. This manual process significantly increases the time required for testing. Future research in this domain should therefore concentrate on developing a fully automated system, thereby enhancing the time efficiency for SCAS test execution. Additionally, developing a transparent version of the proxy, which is currently nontransparent and thus visible in network traffic and packet captures, should be considered as future work. Furthermore, as access to non-exposed interfaces becomes feasible, new opportunities arise for improving network security. The implementation of novel proxy solutions, tailored to specific protocols utilized within the Service-Based Architecture (SBA), could facilitate more in-depth testing scenarios.

Acknowledgments

This work is original and has been partly supported by a collaboration between RSE S.p.A. and Fondazione Ugo Bordoni, financed by the Research Fund for the Italian Electrical System under the Three-Year Research Plan 2022-2024 (DM MITE n. 337, 15.09.2022), in compliance with the Decree of April 16th, 2018.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly and Qwen2.5-32B-instruct in order to: Grammar and spelling check, Text Translation.

References

[1] F. D'Alterio, M. Rotunno, M. Settembre, A. Bernardini, L. Sagratella, G. Bianchi, F. Mancini, A. Paci, N. Maunero, Navigating 5g security: Challenges and progresses on 5g security assurance and risk assessment, in: 2024 AEIT International Annual Conference (AEIT), IEEE, 2024, pp. 1–6.

[2] 3GPP, Security Assurance Methodology (SECAM) for 3GPP network products, Technical Specification (TS) 33.916, 3rd Generation Partnership Project (3GPP), 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2345, version 18.0.0.

[3] 3GPP, Security architecture and procedures for 5G system, Technical Specification (TS) 33.501, 3rd Generation Partnership Project (3GPP), 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169, version 18.7.0.

[4] Ericsson, Is your 5G network secured by NESAS and SCAS standards?, 2022. https://www.ericsson.com/en/blog/2022/12/is-your-5g-network-secured-by-nesas-and-scas-standards [Accessed: 13/10/2024].

[5] M. Francesco, B. Giuseppe, Scasdk-a development kit for security assurance test in multi-network-function 5g, in: Proceedings of the 18th International Conference on Availability, Reliability and Security, 2023, pp. 1–8.

[6] F. Mancini, S. Da Canal, G. Bianchi, Amfuzz: Black-box fuzzing of 5g core networks, in: 2024 19th Wireless On-Demand Network Systems and Services Conference (WONS), IEEE, 2024, pp. 17–24.

[7] 3GPP, Catalogue of general security assurance requirements, Technical Specification (TS) 33.117, 3rd Generation Partnership Project (3GPP), 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2928, version 18.3.0.

[8] 3GPP, 5G Security Assurance Specification (SCAS); Access and Mobility management Function (AMF), Technical Specification (TS) 33.512, 3rd Generation Partnership Project (3GPP), 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3445, version 18.2.0.

[9] 3GPP, 5G Security Assurance Specification (SCAS);User Plane Function (UPF), Technical Specification (TS) 33.513, 3rd Generation Partnership Project (3GPP), 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3446, version 18.1.0.

[10] ishidawataru, SCTP Go implementation, 2023. https://github.com/ishidawataru/sctp [Accessed: 06/07/2024].

Table 2
Excluded 3GPP specific SCAS tests from TS 33.117.

Test Name	Motivation
TC_PROTECT_TRANSPORT_LAYER	Test requires access to HTTP/2 SBA interfaces, not available in ACME 5G Core.
TC_AUTHORIZATION_TOKEN_VERIFICATION_FAILURE_ONE_PLMN	Test requires access to HTTP/2 SBA interfaces, not available in ACME 5G Core.
TC_AUTHORIZATION_TOKEN_VERIFICATION_FAILURE_DIFF_PLMN	Test requires access to HTTP/2 SBA interfaces, not available in ACME 5G Core.
TC_CLIENT_CREDENTIALS_ASSERTION_VALIDATION	Test requires access to HTTP/2 SBA interfaces, not available in ACME 5G Core.
TC_FUZZING_AND_ROBUSTNESS_TESTING	Test requires COTS tools for fuzz testing. Those tools are not available for the project.

Table 3
Excluded SCAS tests for UPF.

Test Name	Motivation
TC_UP_DATA_CONF_UPF	Test requires an IPSec tunnel enabled and the knowledge of security parameters for IPSec. None of them are available in ACME 5G Core.
TC_UP_DATA_INT_UPF	Test requires an IPSec tunnel enabled and the knowledge of security parameters for IPSec. None of them are available in ACME 5G Core.
TC_UP_DATA_REPLAY_UPF	Test requires an IPSec tunnel enabled and the knowledge of security parameters for IPSec. None of them are available in ACME 5G Core.
TC_UP_DATA_CONF_UPF_N9	Test requires access to N9 interfaces, not available in ACME 5G Core.
TC_CP_DATA_CONF _UPF_N4	Test requires access to N4 interfaces, not available in ACME 5G Core.
TC_TEID_ID_UNIQUENESS_UPF	Test requires access to N4 interfaces, not available in ACME 5G Core.
TC_IPUPS_PACKET_HANDLING	Test requires the roaming functionality, not available in ACME 5G Core.
TC_IPUPS_MALFORED_MESSAGES	Test requires specific fuzzing tools for GPRS Tunneling Protocol (GTP), not available.

A. Excluded Tests

This chapter presents the excluded SCAS tests along with their exclusion reason. Table 3 shows the tests excluded from the UPF document, Table 2 details the tests excluded from the 33.117 document, and Table 4 lists the tests excluded from the AMF document.

Table 4
Excluded SCAS tests for AMF.

Test Name	Motivation
TC_SYNC_FAIL_SEAF_AMF	Test requires to intercept traffic between AMF and AUSF. This interface is not available in ACME 5G Core.
TC_RES_STAR_VERIFICATION_FAILURE case C, D, E, F,	Test requires to intercept traffic between AMF and AUSF or to alter AUSF's behavior. This interface is not available in ACME 5g Core.
TC_AMF_REDIRECTION_5GS_EPS	Test requires an Evolved Packet Systems (EPS) architecture and the Cellular Internet Of Things (CIoT). Those functionalities are not available in ACME 5G Core.
TC_NAS_NULL_INT_AMF	Test requires a User Equipment with emergency mode support. UERANSIM does not support emergency mode.
TC_BIDDING_DOWN_XN_AMF	Test requires the handover between different networks. Additional networks are not available in ACME 5G Core
TC_NAS_ALG_AMF_CHANGE_AMF	Test requires the handover between different networks. Additional networks are not available in ACME 5G Core.
TC_5G_GUTI_ALLOCATION_AMF	Test requires the Cellular Internet Of Things (CIoT) support, which is not available in ACME 5G Core.
TC_AMF_REEST_CP_CIOT	Test requires the Cellular Internet Of Things (CIoT) support, which is not available in ACME 5G Core.
TC_VALIDATION_SNSSAI_IN_PDU_REQUEST	Test requires to intercept traffic between AMF and NSSAAF. This interface is not available in ACME 5G Core.
TC_NSSAA_REVOCATION	Test requires sending a packet to a specific interface to trigger an NSSAI revocation. This interface is not available in ACME 5G Core.