**5G NAS Protocol Fuzzing Report**

**Testbed:** Open5GS + srsRAN\_Project   
**Goal:** Fuzz the 5G NAS protocol stack by directly injecting malformed NAS messages into Open5GS using SCTP sockets, and observe core network behavior under invalid input conditions.

**System Components Used**

| **Component** | **Purpose** |
| --- | --- |
| **Open5GS** | 5G Core Network (AMF, SMF, UPF) |
| **srsRAN\_Project** | Provided the gNB (not the UE) |
| **Python** | Used to simulate UE by sending NAS |
| **SCTP socket** | Used to deliver NAS payloads to AMF |
| **AMF logs** | Monitored for decode failures & behavior |

**Environment Setup**

**1. Open5GS Core Network**

* Components Used: amfd, smfd, upfd, nrfd
* Version: Open5GS v2.7.5
* Configuration:
  + AMF bound to 127.0.0.5:38412
  + MongoDB used for subscriber provisioning
  + Test IMSI: 001010123456789

**2. Access Network via srsRAN\_Project**

* Used only the gNB from srsRAN\_Project
* No nr-ue included in this version
* UE behavior simulated manually via Python

**Fuzzing Strategy: Mutated NAS Registration Requests**

To simulate realistic traffic and trigger NAS-layer parsing logic, a base 5G **NAS Registration Request** message was constructed manually:

python

CopyEdit

BASE\_NAS = bytes.fromhex(

"7e00" # Extended protocol discriminator + security header

"41" # Registration Request

"00" # Registration type

"f0" # SUCI format

+ "01" \* 10 # Dummy IMSI

)

Each iteration of the fuzzer randomly mutated 1–2 bytes in this payload:

python

CopyEdit

mutated[idx] = random.randint(0, 255)

The result was sent over **SCTP directly to the AMF**, mimicking gNB-originated traffic.

**Outcome**

* AMF accepted the SCTP connection and attempted to decode messages
* Logs showed:

vbnet

CopyEdit

WARNING: Failed to decode ASN-PDU

ERROR: Cannot decode NGAP message

connection refused!!!

* AMF forcibly removed the gNB context after failing to decode the NGAP layer
* Demonstrated Open5GS behavior under malformed NAS embedded in NGAP

**Observations & Findings**

| **Observation** | **Result** |
| --- | --- |
| SCTP connection established | Yes |
| AMF logs NGAP/NAS decode errors | Yes |
| Controlled gNB disconnects | Yes |
| Core crash | None observed |
|  |  |

The experiment involved sending **mutated 5G NAS Registration Requests** over **SCTP** directly to the **AMF** in Open5GS, without using a real UE. The goal was to analyze how Open5GS handles malformed or semi-valid NAS messages.

**What Worked**

1. **SCTP connections were successfully established** from the fuzzer to the AMF (127.0.0.5:38412).
2. **Mutated NAS messages** were accepted by the SCTP stack and delivered to the AMF.
3. Open5GS **attempted to decode the NAS payloads**, which were embedded directly in the expected NGAP context.
4. The AMF **logged detailed decode errors**, confirming it tried to process the fuzzed data:

[core] WARNING: Failed to decode ASN-PDU

[ngap] WARNING: Failed to decode NGAP-PDU

[amf] ERROR: Cannot decode NGAP message

1. Open5GS responded to malformed input by:
   * Rejecting the gNB connection
   * Cleaning up the failed context
   * Resetting its internal gNB count (Number of gNBs is now X)

**Behavior Was Repeatable**

* Each new fuzzed message triggered the same lifecycle:
  + Accept → Decode → Fail → Reject → Remove
* No AMF crash occurred (controlled error handling)

**gNB Startup**A screenshot of a computer

AI-generated content may be incorrect.

5G Core Startup

A screenshot of a computer program

AI-generated content may be incorrect.

Fuzzer Execution:

A screenshot of a computer

AI-generated content may be incorrect.

Output:

A screenshot of a computer

AI-generated content may be incorrect.