

Securing 5G: Network Slicing Phase 2 Test Report

5G Security Test Bed's Network Slicing Tests Strengthen 5G Security Spring 2024

Table of Contents

Introduction
Scope of Report
Background
Network Slicing4
Network Slice Identifiers5
Definition of Test Cases5
Test Results7
Summary of Process and Findings7
Test Case 1: Malicious Access to a Different Slice by Modifying Slice Differentiator7
Test Case 2: Static Subscriber S-NSSAI Override of UE-Requested NSSAI by Dynamic Core Signaling Assignment15
Conclusions and Next Steps
Summary of Test Results20
5G Security Test Bed Strengthened 3GPP Standards20
Next Steps
About the 5G Security Test Bed 22
Appendix: Acronyms
References

Introduction

The 5G Security Test Bed Is Designed to Rigorously Test and Advance 5G Security

The wireless industry prioritizes stronger security and reliability with every generation of its mobile networks. With 5G in particular, secure connectivity is the foundation that supports and enhances the many benefits these networks provide. The wireless industry devotes significant resources to 5G security and has expanded its efforts through the 5G Security Test Bed.

Formally launched in 2022, the 5G Security Test Bed is a unique collaborative endeavor between wireless providers, equipment manufacturers, cybersecurity experts, academia, and government agencies, created with a sole focus on testing and validating 5G security recommendations and use cases from government groups, wireless operators, and others. It is the only initiative that uses commercial-grade network equipment and facilities to demonstrate and validate how 5G security standards recommendations will work in practical, real-world conditions.

One of the 5G Security Test Bed's core values lies in its ability to validate 5G security use cases in a real-world environment, using an actual 5G network architecture. Leveraging a significant investment and in-kind contributions, the Test Bed's founding members built this state-of-the-art, private 5G network from scratch for the singular purpose of evaluating 5G network security.

The 5G Security Test Bed's previous testing activities have worked to validate the recommendations of the FCC's Communications Security, Reliability, and Interoperability Council (CSRIC) advisory group, for both non-standalone (NSA) and standalone (SA) network configurations. In addition, the Test Bed draws on recommendations from its own Technical Advisory Committee to address emerging industry priorities.

This report addresses network slicing security on 5G networks, adding to three previous test cases conducted in the Phase 1 network slicing report released in 2023.ⁱ The 5G Security Test Bed will continue evaluating additional recommendations and use cases from CSRIC and other entities in future tests. It is not set up to be a platform for identifying vulnerabilities or conducting penetration testing of networks or equipment.

Scope of Report

This 5G Security Test Bed report's scope is to demonstrate potential exploits that result from network slicing vulnerabilities outlined in the AdaptiveMobile Security (AMS) white paper *A Slice in Time: Slicing Security in 5G Core Networks*, submitted to GSMA as a Common Vulnerability Disclosure (CVD).^{II} While the vulnerability was theoretical in the AMS paper, the Test Bed replicated the hypothetical exploit outlined in the report on its own live network infrastructure. The Test Bed also developed and tested an additional potential exploit based on the vulnerability.

These two test cases—the first to validate and assess this theoretical vulnerability using realworld testing—along with the AMS report, led to the 3rd Generation Partnership Project (3GPP) updating the 5G network technical specifications found in TS 33.501, Version 18,ⁱⁱⁱ resolving the vulnerability.

Background

Network Slicing

Network slicing is a technology that enables mobile network operators to provide fine-grained, customizable network offerings to meet the diverse needs of a variety of customers and applications, such as in public safety, transportation, security, and many other contexts.

Often, network slices are discussed in the context of leading commercial applications, such as the four wireless network service types defined by 3GPP: eMBB (Enhanced Mobile Broadband), URLLC (Ultra-Reliable Low-Latency Communication), mMTC (Massive Machine-Type Communication), and Vehicle to X (V2X). In addition, network slices for specific uses, such as vehicle-to-infrastructure, or a specific company's industrial control system, are also considered for application of the network slicing concept.

Network slices can be viewed as logical networks sharing a common physical infrastructure. The security for network slicing will be critical to certain segments of commercial customers. Regarding network slice security, because network slices leverage network function (NF) virtualization and a service-oriented architecture, the main focus for slice security has been to ensure isolation among different slices. Specifically, there are two aspects of isolation: resource provision/isolation and security isolation. Security isolation not only requires slice-specific access control and security measures, but also ensures that potential problems in one slice will not spill over to other slices.

Network Slice Identifiers

In the 5G network core, each slice is given a unique identity called the Single-Network Slice Selection Assistance Information (S-NSSAI). The S-NSSAI is used to identify, differentiate, authorize, and route each slice and the data traveling on it. The S-NSSAI identifies each slice in two parts: a mandatory Slice Service Type (SST), which is a predefined value for one of the wireless network service types defined by 3GPP listed above, and an optional Slice Differentiator (SD), which is used to differentiate between slices of the same type.

In some cases, user equipment (UE) may be authorized to access more than one slice, and the network further differentiates these with a group or list of up to eight slices identified by Network Slice Selection Assistance Information (NSSAI). NSSAIs can fall into several categories, including Allowed, Rejected, Configured, or Requested NSSAIs.

AdaptiveMobile's 2021 CVD Report, *A Slice in Time: Slicing Security in 5G Core Networks*, noted that there was no requirement for mobile network operators (MNOs) that each slice has a unique identifier, or that Slice Differentiators be random, adding that, "With all the new network functions and services in 5G, roaming and legacy interaction will become quite complex."

3GPP's technical standards on network slicing note that "the subscription information shall include at least one default S-NSSAI," requiring network operators to include a default described S-NSSAI in the subscriber slice profile for the UE. However, with no guidance for interoperability between vendors in 3GPP's technical standards at the time of AdaptiveMobile's report release, *A Slice in Time* highlighted security challenges through configuration mistakes and missing layer matching. The report listed some hypothetical examples, showing how slices could potentially be manipulated to do more than they should be able to do, including by brute forcing or guessing slice identifiers to access unauthorized information or resources within the network.

Definition of Test Cases

Based on guidance from its Technical Advisory Committee and the AdaptiveMobile report, the 5G Security Test Bed wrote and executed two tests to demonstrate whether the theoretical vulnerability could be exploited in a real-world environment.

In the first potential exploit outlined in the report, "Malicious Access to a Different Slice by Modifying Slice Differentiator," a rogue network function or a rogue slice (Slice 2 in this example) from a compromised partner could theoretically establish a TLS connection that appears valid because it is using the correct slice identity. This rogue slice would then be allowed to access the network function shared with the valid Slice 1 and granted a valid identification token. With this access token, the network's transport/IP layer security would determine the rogue slice is a valid connection, granting it access to personal data or resources within the network.

The report discusses the cause of the vulnerability:

The underlying problem is that no layer matching is mandated by the specifications. As there is no matching between layers, the NRF would only see, on the lower transport/IP layer, an "authenticated partner" and, on the upper signalling layer, a valid slice identity and service request. There is no cross-check that the slice identity in the request matches the slice identity used for the TLS tunnel. As a result, the NRF would issue an authorization ticket to the rogue Slice 2 to use services on Slice 1.

The 5G Security Test Bed performed the first real-world tests to confirm whether this hypothetical vulnerability could be exploited, listed in this report as Test Case 1. In addition, the Test Bed developed a second test case based on the vulnerability, "Static Subscriber S-NSSAI Override of UE-Requested NSSAI by Dynamic Core Signaling Assignment," or referred to more simply as "5G Network Override to Reroute a Mis-Provisioned Network User Slice."

In Test Case 2, the Test Bed assumed that a user device had been incorrectly provisioned to a network slice that was inconsistent with how it was recorded by the network operator. In this example, the UE should be provisioned for Slice 2, but it incorrectly presents Slice 1 as the Requested NSSAI to the network. Test Case 2 was conducted to determine whether the network would override the incorrect request and direct the UE to the correct slice.

In summary, the 5G Security Test Bed conducted the following tests:

1. Malicious Access to Different Slice by Modifying Slice Differentiator

- a. **5G STB Test Case 1:** Confirm the vulnerability by executing a successful modified request first from the compromised network function (NF) to the network repository function (NRF), and then to the shared NF.
- 2. Static Subscriber S-NSSAI Override of UE-Requested NSSAI by Dynamic Core Signaling Assignment
 - a. **5G STB Test Case 2:** Test and validate the cellular network's ability to override a mis-provisioned, mis-configured, or manipulated user device (UE) that has the network slice provisioned for Slice 1 where it should be Slice 2.

The vulnerability outlined by AdaptiveMobile is a niche edge case that requires a very specific set of circumstances to occur and would be difficult to execute in the real world. However, by confirming with real-world tests and disclosing the findings to 3GPP, the 5G Security Test Bed helped drive the development of updated standards to address the potential issue and override the vulnerability.

Test Results

Summary of Process and Findings

The configuration used for these tests is comprised of radio access network (RAN) equipment hosted at the University of Maryland (UMD) and a dual-mode core (DMC), which provides both 4G LTE and 5G functionality hosted at the MITRE Corporation. The connection between the RAN at UMD and the DMC at MITRE goes over the internet. The DMC is an Ericsson 5G Core, PCC version 1.19.

The core is configured to support two network slices. Mutual TLS (mTLS) is implemented among network functions.

Tests were run with band N41 for the new radio (NR), using a Sierra Wireless EM9190 card connected to a laptop by USB as a cellular modem, as well as a Qualcomm Mobile Test Platform (MTP) device. For the purposes here, this report refers to the combination of that laptop and the cellular modem as the UE. The UE used for Slice 1 was the Qualcomm MTP. It was set inside a Faraday Cage and connected remotely through a laptop.

Test Case 1: Malicious Access to a Different Slice by Modifying Slice Differentiator

Test Case ID: TC-NetSlic-01

Description:

Test Case 1 is designed to confirm the existence of the vulnerability reported by AdaptiveMobile in Section 3.1.3.1 of its report, *A Slice in Time: Slicing Security in 5G Core Networks* (submitted to GSMA as CVD-2021-0047). To confirm the vulnerability, the test's objectives are to use a compromised network function belonging to Slice 2 to send a modified request to the NRF, gain access to a valid authentication token, and then use that token to access a network function shared between both Slice 1 and 2, gaining access to personal user data or resources.

This test uses the SMF network function in Slice 2 as the compromised NF, and it uses the AMF as the network function that is shared between both slices. It then uses the Namf_Communication_N1N2MessageTransfer service request to attempt to release an existing connection that the UE in Slice 1 is connected to in order to see if the AMF will respond to this maliciously crafted request.

The test uses Ericsson's internally developed HTTP/2 and REST API call interception and modification tool called HALO. The tool operates as a man-in-the-middle that can inspect the HTTP/2 requests and responses. Using HALO, requests and responses can be modified in the HTTP/2 Body, within the JSON or any payload type, the HTTP/2 Headers, and the URI query parameters. For the test, HALO sits outside the 5G network core, and traffic is directed to HALO.

by configuring the IP addresses of the AMF and the NRF to be interfaces on HALO. HALO masquerades as the SMF on Slice 2, generating and receiving messages as appropriate. HALO is capable of having a PKI certificate for the core installed, and it can decode TLS traffic. It also has the Wireshark capability to capture traffic in PCAP format. With mTLS implemented, everything is encrypted between NFs. As a result, Wireshark captures from outside HALO cannot read the contents of messages. In order to display message content, we need to read them (and modify and/or generate them) inside HALO, where they have been decrypted.

Prior to executing the test below, HALO snoops the connection PDU setup for the UE in Slice 1 in order to gather the needed parameters to craft the malicious REST API call to release the connection from Slice 2.

A number of features are configured for the test case in the Service-Based Architecture (SBA) of the 5G core in addition to mutual TLS using PKI certificates. The NRF acts as the OAuth 2.0 Authorization server and generates the Access Token Response, which is a JWS-signed JSON object containing the access token, a token type, expiration, and scope, which is per REST API service. Tokens are authorized per REST API service for the NF where one side acts as the NF service consumer and the other NF acts as the NF service producer. The tokens generated by the NRF authorize the Namf_Communication_N1N2MessageTransfer service to be used between the SMF as the NF service consumer, and the AMF as the NF service producer, as an example in this test case.

This test case recognizes that the above scenario is difficult in the real world to execute, but the purpose is to test the layer matching problem highlighted by Adaptive Mobile. The actual testing is constrained by the actual supported REST APIs for the SMF to the AMF, which are limited in the current software to the Namf_Communication_N1N2MessageTransfer service, the N1N2Transfer Failure Notification service, and the EBIAssignment service. These allow for only PDU session establishment, modification, and release operations. Additional software releases add more REST API services.



Figure 1: Message Flow of Hypothesized Vulnerability

Because packet captures occur inside the HALO tool, the IP addresses associated with the messages are in some cases the internal HALO IP addresses facing the NFs. The relevant IP addresses are listed in Table.

Table: Network Function IP Addresses

AMF	NRF	SMF
192.168.56.143	192.168.56.143	192.168.56.129
172.17.20.108 (HALO)	172.17.95.197 (HALO)	192.168.56.131
	172.17.96.50 (HALO)	

Prior to conducting the test, the UE is connected to Slice 1 and establishes a data connection to the Slice 1 server. Figure 2 shows the UE registration in which it provides its Subscription Concealed Identifier (SUCI) (310014791791003), and Figure 3 is the UE establishing its PDU session (PDU Session ID 1) on Slice 1 (SST=1, SD=1). HALO captures the PDU session establishment in order to use the needed parameters to craft the malicious REST API call to release the PDU session later in the test.

•	•					 23-11-06_17. ⁻	18.54_F	RID-716_UE.pcap			
		1	= 🗎 🗙	ه ۹ 🗧	🔸 🖀 著 速 🖡			↓ Ⅲ			
📕 ng	р									+ ~ <	
No.	8 10 11	Time 2023-11-06 2023-11-06 2023-11-06	17:19:01.6270. 17:19:01.6304. 17:19:01.6304.	Source 192.168.48.130 10.205.67.204 10.205.67.204	Destination 10.205.67.204 192.168.48.130 192.168.48.130	Protocol I NGAP NGAP NGAP	Length 146 226 82	Info NGSetupRequest NGSetupResponse NGReset			
	13 17 19	2023-11-06 2023-11-06 2023-11-06	17:19:01.6305. 17:19:04.1258. 17:19:04.2119.	. 192.168.48.130 . 192.168.48.130 . 10.205.67.204	10.205.67.204 10.205.67.204 192.168.48.130	NGAP NGAP/NAS-5GS NGAP/NAS-5GS	70 138 138	NGResetAcknowledge InitialUEMessage, Registr DownlinkNASTransport, Aut	ration request :hentication request		l
	20 21 22 24	2023-11-06 2023-11-06 2023-11-06 2023-11-06	17:19:04.2120. 17:19:04.2204. 17:19:04.2204. 17:19:04.4948.	. 192.168.48.130 . 10.205.67.204 . 192.168.48.130 . 10.205.67.204	10.205.67.204 192.168.48.130 10.205.67.204 192.168.48.130	NGAP/NAS-5GS NGAP/NAS-5GS NGAP/NAS-5GS NGAP	150 130 198 182	SACK (Ack=2, Arwnd=32768) SACK (Ack=3, Arwnd=32768) SACK (Ack=3, Arwnd=32768) InitialContextSetupReques	 , UplinkNASTransport, Authentication response , DownlinkNASTransport, Security mode command , UplinkNASTransport 		
			Ex 00 5G ~ NA ~ 5G	tended protocold 000 = Spare H ssage type: Regit ssage type: Regit 000 = Securi ssage type: Regit 000 = Formation 001 = Socold 001 = Socold 001 = Spar 001 = Spa	<pre>iiscriminator: 5G mo laif Octat: 0 y header type: Plai tration request (0% ow-On Request bit (f registration type: j ier of security context key set identifier: / e: 0 format: IMSI (0) e: 0 of identity: SUCI (ode (MCC): United St ode (MCC): TEST IMSJ r: 0 ection scheme Id: NL lic key identifier:</pre>	h NAS message, n NAS message, n 11) OR): Follow-on nitial registra' flag (TSC): Nar 7 1) ates (310) HNI (014) LL scheme (0) 0	nt messa not secu request tion (1) tive sec	ges (126) rity protected (0) pending) curity context (for KSIAMF))		
			∨ UE	security capabi	ity						Fra
0	NG	Application Pro	tocol: Protocol						Packets: 462 · Displayed: 24 (5.2%)	Profile: Defau	alt

Figure 2: UE Registration Showing SUCI

(
			23-11-06_17	7.18.54	ND-716_UE.pcap	
	🔬 🐵 🔚 🗋 🕱 🙆 🔍 👄 🖷) 😫 🕌 🛓 🗌	. 🗌 🔍		2 亜	
						× +
No.	Time Source	Destination	Protocol	Length	Info	
	25 2023-11-06 1/:19:04.4949 192.168.48.130	10.205.07.204	NGAP	100	SALK (ACK=4, Arwng=32/68) , InitialContextSetupKesponse	
	26 2023-11-06 17:19:04.4949 192.168.48.130	10.205.67.204	NGAP	134	UERadioCapabilityInfoIndication	
	28 2023-11-06 17:19:04.5024 10.205.67.204	192.168.48.130	NGAP/NAS-5GS	176	DownlinkNASTransport	
	29 2023-11-06 17:19:04.5024 192.168.48.130	10.205.67.204	NGAP/NAS-5GS	138	SACK (Ack=5, Arwnd=32768) , UplinkNASTransport	
	30 2023-11-06 17:19:04.5104 10.205.67.204	192.168.48.130	NGAP/NAS-5GS	138	SACK (Ack=7, Arwnd=32768) , DownlinkNASTransport	
	33 2023-11-06 17:19:07.1254 192.168.48.130	10.205.67.204	NGAP/NAS-5GS	178	UplinkNASTransport	
LL	35 2023-11-06 17:19:07.3351 10.205.67.204	192.168.48.130	NGAP/NAS-5GS	254	PDUSessionResourceSetupRequest	
	36 2023-11-06 17:19:07.3352 192.168.48.130	10.205.67.204	NGAP	126	SACK (Ack=7, Arwnd=32768) , PDUSessionResourceSetupResponse	
	72 2023-11-06 17:20:05.5322 10.205.67.204	192.168.48.130	NGAP/NAS-5GS	114	Down LinkNASTransport	
	73 2023-11-06 17:20:05.5323 192.168.48.130	10.205.67.204	NGAP/NAS-5GS	150	SACK (ACK=8, Arwnd=32768), UplinkNASTransport	
	74 2023-11-06 17:20:05.5400 10.205.67.204	192.168.48.130	NGAP/NAS-565	130	SACK (ACK=10, Arwnd=32768), DownlinkNASiransport	
-	Ttem 2: id-PDUSessionResourceSetupl	istSUReg	RICH P	0		1
1	<pre>v ProtocolIE-Field</pre>	xo to onlog				
4	id: id-PDUSessionResourceSetup	ListSUReg (74)				
	criticality: reject (0)					
	✓ value					
	v PDUSessionResourceSetupList	UReq: 1 item				
	∨ Item 0					
	v PDUSessionResourceSetu	DItemSUReq				
1	pDUSessionID: 1					
1	v pDUSessionNAS-PDU: 7	e02a1ce4c98037e0068	0100472e0101c2	21100090	000631200101ff09060107d0059896	
	Von-Access-Stratu	n 5GS (NAS)PDU				
	Security protect	ted NAS 5GS message				
	Extended pro	tocol discriminator	: 5G mobility	manageme	nt messages (126)	
	0000 =	Spare Half Octet: 0	.			
	0010 =	security header typ	e: integrity p	rotected	and cipnered (2)	
	message auth	entication code: 0x	a1ce4c98			
	Sequence num	ber: 3				
	STIJAL					
	s51. 01					
	v pDUSessionResourceSe	tupRequestTransfer	0000040082000	0a081e84	04002540be400008b000a01f00acd43d156ac0e720086000100	
	PDUSessionResource	SetupRequestTransf	er			
	v protocolIEs: 4	items				
	∨ Item 0: id-P	DUSessionAggregateM	aximumBitRate			
	V ProtocolIE	-Field				
	id: id-	PDUSessionAccrecate	MaximumBitRate	(130)		
07	NG Application Protocol: Protocol				Packets: 462 · Displayed: 24 (5.2%)	Profile: Default

Figure 3: UE PDU Session Setup Request Showing pDUSessionID and s-NSSAI

HALO performs the function of the rogue NF, the SMF, on Slice 2. Figure 4 shows the NF instance details of this NF, including NF type (SMF) and NSSAI (SST=1, SD=2).

```
CCCQCCONTOL_PLARE_MITGMC_ECONT_MASS4_SY04:~/MILS_CCLIS_SECURED_SED/8043> CURL -X GEL INTERS://MILGARE_MICGU4.mcC310.3gppnetwork.org/nnrt-ntm/
v1/nf-instan
nccs/2a735ed1-927c-446a-b4b6-6860391da167' --<u>cacert SecureG-cacert-root.nem</u> --cert <u>SecureG-nrf-cert.nem</u> --key <u>SecureG-nrf-key.nem</u> | jg | grep -i smf
% Total % Received % <u>Xferd</u> Average Speed Time Time Current
<u>Dlaad</u> Upload Total Spent Left Speed
                                                                                0 --:--:- --:-- 0
0 --:--:- --:-- 86888
       0 0 0
782 100 782
100
                                                           86888
    "ipv4Addresses": [
"192.168.56.131"
   "nfInstanceId": "2a735ed1-927c-446a-b4b6-6860391da167",
           pervices
           "allowedNfTypes": [
          ], "fado": "mtrdmcsmf02.smf.dmc.mnc014.mcc310.3appnetwork.org",
"ipEndPoints": [
              {
"<u>ipv4Address</u>": "192.168.56.131",
"port": 7070
          ),

"nfServiceStatus": "REGISTERED",

"scheme": "https",

"serviceInstanceId": "<u>nsmf-pdusession.2a735ed1-927c-446a-b4b6-5860391da167</u>",

"serviceName": "<u>nsmf-pdusession</u>",

"serviceName": "
            "versions": [
              {
    "apiFullVersion": "1.1.2",
    "apiVersionInUri": "v1"

          1
    "nfStatus": "REGISTERED",
"nfIype": "SMF",
"Dumnist": 1
           "mcc": "310",
"mnc": "014"
       }
     'sNssais": [
          "sd": "000002",
"sst": 1
       }
     ,,
'smfInfo": {
''pgwEado": "topoff.pgw-s5gn.mcn-sgwu.node.epc.mnc014.mcc310.3gppnetwork.org",
"sNssaiSmfInfoList": [
               "dnnSmfInfoList": [
                      "dnn": "dnn-embb-stb2.mitre.net"
                  }
               "<u>sNssai</u>": {
"sd": "000002",
                  "sst": 1
              ı
```

Figure 4: Compromised NF (SMF) Parameters (nfType, sNssai, etc.)

HALO, masquerading as the compromised SMF on Slice 2, establishes a TLS connection to the NRF to request access to the AMF shared by both Slice 1 and Slice 2. HALO then sends a request to the NRF to provide the rogue SMF a token to access Slice 1 on the shared NF server. In this test case execution, the slice for which the request corresponds is not included in the request, as targetSnssaiList is optional in the AccessTokenReq. Line 56 of Figure 5 shows the rogue SMF (the IP address appears as 172.17.96.50, which corresponds to the HALO interface facing the NRF) requesting the AMF access token from the NRF (192.168.56.143), where the nfInstanceId matched that in Figure 4. Note the details in the request do not include the targetSNSSAIList. Line 60 of Figure 6 shows the NRF (192.168.56.143) returning the AMF access token without any specific S-NSSAI listed. The scope allows access to the Namf_Communication_N1N2MessageTransfer service.

			🚄 work	ing masquerade	slice2 nrf	Halo-dsb.pca	a	
🔼 📕 🙋 🥹		<u> </u>	* 🎽 🎌 👱 🛛					
Apply a display filter	<%/>>							+
No. Time		Source	Destination	Protocol	Length	Info		
49 2023-11-	06 17:20:05.4355.	. 192.168.56.143	172.17.96.50	TCP	72	443 → 52994	4 [ACK] Seq=1572 Ack=2797 Win=60160 Len=0 TSval=4278169265 TS	-
50 2023-11-	06 17:20:05.4356.	. 192.168.56.143	172.17.96.50	HTTP2	4215	SETTINGS [0]], SETTINGS[0], WINDOW_UPDATE[0]	
51 2023-11-	06 17:20:05.4356.	. 172.17.96.50	192.168.56.143	TCP	72	52994 → 443	3 [ACK] Seq=2797 Ack=5715 Win=57088 Len=0 TSval=1267033481 TS	
52 2023-11-	06 17:20:05.4382.	. 172.17.96.50	192.168.56.143	HTTP2	194	HEADERS[1]:	: POST /oauth2/token, WINDOW_UPDATE[1]	
53 2023-11-	06 17:20:05.4382.	. 192.168.56.143	172.17.96.50	тср	72	443 → 52994	4 [ACK] Seq=5715 Ack=2919 Win=60160 Len=0 TSval=4278169268 TS	-
54 2023-11-	06 17:20:05.4386.	. 172.17.96.50	192.168.56.143	HTTP2	236	DATA[1]		
55 2023-11-	06 17:20:05.4386.	. 192.168.56.143	1/2.1/.96.50	TCP	/2	443 → 52994	4 [ACK] Seq=5/15 ACK=3083 Win=60160 Len=0 ISval=42/8169268 IS	_
56 2023-11-	06 17:20:05.4388.	. 172.17.96.50	192.168.56.143	HTTP2	103	DATA[1] (ap	pplication/x-www-form-urlencoded)	
57 2023-11-	06 17:20:05.4388.	. 192.168.56.143	1/2.1/.96.50	TCP	12	443 → 52994	4 [ACK] Seq=5/15 ACK=3114 Win=60160 Len=0 ISval=42/8169268 IS	
58 2023-11-	06 17:20:05.4400.	. 1/2.1/.96.50	192.168.56.143	HTTP2	103	SETTINGS [0]] 4. [ACK]: Come F715, April 20145, Mine C0160, Long 0, Tours], 4270160270, TO	
59 2023-11-	06 17:20:05.4400.	. 192.108.50.143	172.17.90.50		12	443 → 52994	4 [ACK] Seq=5/15 ACK=3145 Win=60160 Len=0 ISVal=42/81692/0 IS	
60 2023-11-	06 17:20:05.4424.	. 192.168.56.143	1/2.1/.96.50	HTTP2/JSUN	652	HEADERS[1]	: 200 UK, DATA[1], JavaScript Ubject Notation (application/js	
1 cuy51 0/01	End Scream							1 6
0000 .00.	= Unused: 0x00							6
0	= Padded: False							6
	= End Stream: I	rue						e
0		= Kese	rved: 0x0					é
.000 0000 0	000 0000 0000 000	0 0000 0001 = Stre	am identifier: i					e
[Pad Length	(122 butos)	+F4(122)]						6
V [1 Bouy Ira	A povlood A-12	(122 but oc)						e
[Pody fra	amont county 11	2 (133 Dytes/]						
[Douy 11d	led body length	1221						
[Reassent	led body data: 6	133] 772616e745f7470706	346366606566745fe	37265646560746	0616c7326	606640607374	4]	
Data: 67726	60745f747070652d	626660656674546272	65646566746061667	22660664060727	N 2010C/220	00004900/3/4	4	
HTML Form IIPI	Encoded: applicat	tion/x_www_form_url	encoded	32002004902737	4			
V Form item:	arant type" = "c	lient credentials"	tencoueu					
Key: arar	t type	cienc_creacherates						0
Value: cl	ient credentials							
value. ct	nfInstanceId" =	"2a735ed1_927c_446	a-h4h6-6860391da1	67"				
Key: nfTr	stanceId	20/55001 52/0 440						
Value: 2a	735ed1-927c-446a-	-b4b6-6860391da167						
✓ Form item:	nfType" = "SMF"	0400 0000000000000000						
Key: nfTy	pe							-
Value: SN	F							
<pre>~ Form item: '</pre>	'targetNfType" =	"AMF"						
Key: tare	etNfType							-
Value: AM	F							
V Form item:	scope" = "namf-c	omm namf-evts"						
Kev: scor	e							
Value: na	mf-comm namf-evts	s						Fr
O Z Text item (text)	11 bytes						Packete: 104 . Displayed: 104 (100 0%)	ault
	11 0 9 10 3							, and

Figure 5: Wireshark Capture of Rogue SMF Request to NRF for AMF Token

• •			/ work	ing masquerade	slice2 nrfl	Halo-dsb.pcap	
						TT	
		S 🖸 K 🗢 🖣	'≝ ● ⊻			 ■ 	
App	ly a display filter <\#/>			1			
lo.	Time	Source	Destination	Protocol	Length		
	53 2023-11-06 17:20:05.43	382 192.168.56.143	172.17.96.50	TCP	72	443 → 52994 [ACK] Seq=5715 Ack=2919 Win=60160 Len=0 TSval=427816	9268 TS
	54 2023-11-06 17:20:05.43	386 172.17.96.50	192.168.56.143	HTTP2	236	DATA[1]	
	55 2023-11-06 17:20:05.43	386 192.168.56.143	172.17.96.50	TCP	72	443 → 52994 [ACK] Seq=5715 Ack=3083 Win=60160 Len=0 TSval=427816	9268 TS
	56 2023-11-06 17:20:05.4	388 172.17.96.50	192.168.56.143	HTTP2	103	DATA[1] (application/x-www-form-urlencoded)	
	57 2023-11-06 17:20:05.4	388 192.168.56.143	172.17.96.50	TCP	72	443 → 52994 [ACK] Seq=5715 Ack=3114 Win=60160 Len=0 TSval=427816	9268 TS
	58 2023-11-06 17:20:05.44	100 172.17.96.50	192.168.56.143	HTTP2	103	SETTINGS[0]	
	59 2023-11-06 17:20:05.44	400 192.168.56.143	172.17.96.50	TCP	72	443 → 52994 [ACK] Seq=5715 Ack=3145 Win=60160 Len=0 TSval=427816	9270 TS
	60 2023-11-06 17:20:05.4	124 192.168.56.143	172.17.96.50	HTTP2/JSON	652	<pre>HEADERS[1]: 200 OK, DATA[1], JavaScript Object Notation (applica)</pre>	tion/js
_	61 2023-11-06 17:20:05.4	124 172.17.96.50	192.168.56.143	TCP	72	52994 → 443 [ACK] Seq=3145 Ack=6295 Win=56832 Len=0 TSval=126703	3488 TS
	62 2023-11-06 17:20:05.4	179 172.17.96.50	169.254.20.10	DNS	120	Standard query 0x9cff A amf-halo.halo.svc.cluster.local.halo.svc	.cluste
	63 2023-11-06 17:20:05.4	179 172.17.96.50	169.254.20.10	DNS	120	Standard query 0x91fb AAAA amf-halo.halo.svc.cluster.local.halo.	svc.clu
	64 2023-11-06 17:20:05.4	184 169.254.20.10	172.17.96.50	DNS	213	Standard query response 0x91fb No such name AAAA amf-halo.halo.s	vc.clus
	[Pad Length: 0]						
	Data: 7b226163636573735f7	46f6b656e223a2265794a	68624763694f694a46	5557a49314e694	9		
~	JavaScript Object Notation:	application/ison					
	< Object	-++					
	Member: access_token						
	Member: access_token [Path with value [tr	runcated]: /access_tok	ten:eyJhbGci0iJFUz	I1NiIsImtpZCI6	Im5yZmtleT	EiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiIwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3Ni1h	ZTJmNWZhł
	Member: access_token [Path with value [tr [Member with value]	<pre>uncated]: /access_tok truncated]: access to</pre>	en:eyJhbGci0iJFUz	I1NiIsImtpZCI6 zI1NiIsImtpZCI	Im5yZmtleT 6Im5yZmtle	EilcJ0eXAiOiJKVlQifQ.eyJpc3MiOiIwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3Ni1h TFilcI0eXAiOiIKVlOifQ.evInc3WiOiIwYzc2NTA4NC05Y2M1LTQ5YzYtOTq3Ni1	ZTJmNWZh
	<pre>> Member: access_token [Path with value [tr [Member with value String value [trunca</pre>	<pre>runcated]: /access_tok truncated]: access to ted]: eyJhbGci0iJFUZI</pre>	en:eyJhbGciOiJFUz ken:eyJhbGciOiJFUz 1NiIsImtpZCI6Im5y	<mark>I1NiIsImtpZCI6</mark> z <mark>I1NiIsImtpZCI</mark> ZmtleTEiLCJ0eX	<mark>Im5yZmtleT</mark> 6Im5yZmtle Ai0iJKV1Qi	EilCJ@eXA101JKV10if0.eyJpc3M101IvYzc2NTA4NC@SY2M1LTOSYzYt0Tg3Niih TFilCJ@eXA101JKV10if0.evjnc3M101IvYzc2NTA4NC@SY2M1LTOSYzYt0Tg3Ni 0.eyJpc3M101JKV2C2NTA4NC@SY2M1LTOSYZYt0Tg3NihZTJmWZMMHE2NY2iLC	<mark>ZTJmNWZh</mark> hZT1mNWZ JzdWIi0i
	Member: access_token [Path with value [tr [Member with value] String value [trunca Key: access_token	<pre>uncated]: /access_tok truncated]: access_to ted]: eyJhbGci0iJFUzI</pre>	en:eyJhbGciOiJFUz ken:evJhbGciOiJFUz NiIsImtpZCI6Im5y	<mark>I1NiIsImtpZCI6</mark> z <mark>I1NiIsImtpZCI</mark> ZmtleTEiLCJ0eX	Im5yZmtleT 6Im5yZmtle Ai0iJKV1Qi	EilCJ0eXA101JKV10if0.eyJpc3M101IwYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11h TEilCJ0eXA101JKV101f0.evJpc3W101IwYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11 f0.eyJpc3M101IwYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11hZTJmWwZhMmE2M2Y1LC	<mark>ZTJmNWZh</mark> hZT]mNWZ JzdWIiOi
	<pre> Member: access_token [Path with value [tr [Member with value String value [trunca Key: access_token [Path: /access_token </pre>	<pre>runcated]: /access_tok truncated]: access_to ted]: eyJhbGci0iJFUz1 </pre>	en:eyJhbGciOiJFUz kken:evlhbGciOiJEU (NiIsImtpZCI6Im5y)	<mark>I1NiIsImtpZCI6</mark> z <mark>T1NiIsImtpZCI</mark> ZmtleTEiLCJ0eX	<mark>Im5yZmtleT</mark> 6Im5yZmtle AiOiJKV1Qi	EILCI0eXAIOIJKVIQIfQ.eyJpc3MiOIIwYzc2NTAANC05Y2MILTO5Y2Yt0Tg3NiI TEilCI0eXAIOIJKVIQIfQ.eyJpc3MiOIIwYzc2NTAANC05Y2MILTO5Y2Yt0Tg3Ni fQ.eyJpc3MiOIIwYzc2NTA4NC05Y2MILTQ5Y2Yt0Tg3NiIhZTJmWWZhMmE2M2YILC	ZTJmNWZh hZTImNWZ JzdWIiOi
	 Member: access_token IPath with value [tr IMember with value String value [truncz Key: access_token IPath: /access_token IPath: /access_token 	runcated]: /access_tok itruncated]: access to ted]: eyJhbGciOiJFUzI	<mark>ken:eyJhbGciOiJFUz</mark> ken:eylbbGciOiJEU 1NiIsImtpZCI6Im5y	<mark>IlNiIsImtpZCI6</mark> z <mark>IlNiIsImtpZCI</mark> ZmtleTEiLCJ0eX	<mark>Im5yZmtleT</mark> 6Im5vZmtle Ai0iJKV1Qi	EilCJ@eXA101JKV1QifQ.eyJpc3M101IwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3Ni1h TEilCJ@eXA101JKV1QifQ.evJpc3M101IwYzc2NTA4NC05Y2M1LTQ5YzYtOTc3Ni1 fQ.eyJpc3M101IwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3N11hZTJmNWZhMmE2M2YiLC	ZTJmNWZh hZTJmNWZ JZdWIiOi
	 Member: access_token IPath with value [tr IMember with value String value [truncz Key: access_token IPath: /access_token IPath: /access_token [Path: token_type [Path with value: /1 	runcated]: /access_tok fruncated]: access fo (tred]: eyJhbGci0iJFUzI]] :oken_type:Bearer]	<mark>ken:eyJhbGciOiJFUz</mark> ken:evlhbGciOiJFUz 1NiIsImtpZCI6Im5y	<mark>IlNiIsImtpZCI6</mark> z <mark>IlNiIsImtpZCI</mark> ZmtleTEiLCJ0eX	<mark>Im5yZmtleT</mark> 6Im5vZmtle Ai0iJKV1Qi	EilCJ0eXA101JKV1QifQ.eyJpc3M101IwYzc2NTA4NC05Y2M1LTQ5Y2YtOTg3Ni1h TEilCJ0eXA101IKV1D1fQ.evJpc3W101IwYzc2NTA4NC05Y2M1LTQ5Y2YtOTg3Ni1 fQ.eyJpc3M101IwYzc2NTA4NC05Y2M1LTQ5Y2YtOTg3N11hZTJmNwZhMmE2M2YiLC	<mark>ZTJmNWZh</mark> hZTJmNWZ JzdWIiOi
	 Member: access_token Path with value [trunce String value [trunce Key: access_token Path: /access_token Member: token_type [Path with value: /r [Member with value: /n 	runcated]: /access_tok truncated]: access tr tted]: eyJhbGci0iJFUzJ J oken_type:Bearer] token_type:Bearer]	<mark>ken:eyJhbGciOiJFUz</mark> ken:eyJhbGciOiJEU (NiIsImtpZCI6Im5)	11NiISImtpZCI6 211NiISImtpZCI ZmtleTEiLCJ0eX	<mark>Im5yZmtleT 6Im5yZmtle</mark> AiOiJKV1Qi	EilCi0eXAl0ijKV10if0.eyjpc3Mi0iIwYzc2NTA4NC05Y2MilT05Y2Yt0Tg3Ni1 TEilCi0eXAl0ijKV10if0.evjpc3Mi0iIwYzc2NTA4NC05Y2MilT05Y2Yt0Tg3Ni f0.eyjpc3Mi0iIwYzc2NTA4NC05Y2MilT05Y2Yt0Tg3Ni1hZTJmNWZhMmE2M2YilC	<mark>ZTJmNWZh</mark> hZTImNWZ JZdWIiOi
	 Member: access_token (Path with value [tr (Member with value) String value [trunce Key: access_token (Path: /access_toker Member: token_type (Path with value: /1 (Member with value: String value: Bearer 	uncated]: /access_to itruncated]: access to ted]: eyJhbGciOiJFUZI] .oken_type:Bearer] token_type:Bearer]	<mark>ken:eyJhbGciOiJFUz</mark> ken:eyJhbGciOiJEU (NiIsImtpZCI6Im5y)	<mark>I1NiIsImtpZCI6</mark> Z11NiIsImtpZCI ZmtleTEiLCJ0eX	<mark>Im5yZmtleT 6Im5yZmtle</mark> AiOiJKV1Qi	EiLCJ@eXA101JKV1QifQ.eyJpc3Mi01IwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3Ni1h TEiLCJ@eXA101JKV1QifQ.evJpc3Wi0ITwYzc2NTA4NC05Y2MILTQ5YzYtOTg3Ni1 fQ.eyJpc3Mi0iIwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3Ni1hZTJmNWZhMmE2M2YiLC	<mark>ZTJmNWZh</mark> hZTJmNWZ JZdWIIOi
	 Member: access_token Path with value [trunc: String value [trunc: Key: access_token IPath: /access_token Member: token_type [Path with value: /access_trunc: Key: access_token Kember: token_type 	<pre>uncated]: /access_tok truncated]: access to ted]: eyJhbGc101JFU21]</pre>	<mark>en:eyJhbGci0iJFUz</mark> ken:evJhbGci0iJFU lNiIsImtpZCI6Im5y;	<mark>I1NiIsImtpZCI6</mark> z <mark>T1NiIsImtpZCI</mark> ZmtleTEiLCJ0eX	<mark>Im5yZmtleT 6Im5yZmtle</mark> AiOiJKV1Qi	EILCI0eXAIOIJKVIQIfQ.eyJpc3MiOIIwYzc2NTA4NC05Y2MILTO5Y2Yt0Tg3NiI TEILCI0eXAIOIJKVIQIfQ.eyJpc3MiOIIwYzc2NTA4NC05Y2MILTO5Y2Yt0Tg3Ni fQ.eyJpc3MiOIIwYzc2NTA4NC05Y2MILTQ5Y2Yt0Tg3Ni1hZTJmWWZhMmE2M2YILC	<mark>ZTJmNWZh hZTJmNWZ</mark> JZdWIiOi
	 Member: access_token IPath with value [tr IMember with value] String value [trunca Key: access_token Member: token_type [Path with value: /t IMember with value: String value: Bearer Key: token_type [Path: /token_type] 	<pre>uncated]: /access_to/ truncated]: access_to ted]: eyJhbGciOiJFUZI] oken_type:Bearer] token_type:Bearer]</pre>	xen:eyJhbGciOiJFUz; kken:eyJhbGciOiJFU; INiIsImtpZCIOIm5y;	IINIISImtpZCI6 7INIISImtpZCI ZmtleTEiLCJ0eX	<mark>Im5yZmtleT 6Im5yZmtle</mark> AiOiJKV1Qi	EilCl0eXAl01JKV101f0,eyJpc2M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11 TEilCl0eXAl01JKV10f0,evJpc3N101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N1 f0,eyJpc3M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11hZTJmNWZhMmE2M2Y1LC	<mark>ZTJmNWZh hZTJmNWZ</mark> JZdWIiOi
	 Member: access_token (Path with value [tr String value [truncs String value [truncs Key: access_token (Path: /access_token (Path: /access_token (Path: with value: / (Member with value: / (Member with value: String value: Bearer Key: token_type (Path: /token_type (Path: /token_type (Peth: /token_type Member: expires_in 	<pre>uncated]: /access_tok fruncated]: access to tted]: eyJhbGc10iJFUZI]</pre>	<mark>ken:eyJhbGciOiJFUz</mark> ken:evJhbGciOiJFUz 1NiIsImtpZCIOIm5y	11NiIsImtpZC16 ZINiIsImtpZCI ZmtleTEiLCJOeX	<mark>Im5yZmtleT 6Im5yZmtle</mark> AiOiJKV1Qi	EiLCJ@eXA10iJKV1QifQ.eyJpc3Mi0iIwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3Ni1h TEiLCJ@eXA10iJKV1QifQ.evJpc3Mi0iIwYzc2NTA4NC05Y2MILTQ5YzYtOTg3Ni fQ.eyJpc3Mi0iIwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3Ni1hZTJmNWZhMmE2M2YiLC	ZTJmNWZh hZTJmNWZ JzdWIiOi
	 Member: access_token IPath with value [1 IPath with value String value [trunca Key: access_token IPath: /access_token Member: token_type [Path with value: /a [Member with value: Bearer Key: token_type [Path: /token_type] Member: expires_in [Path with value: /a 	<pre>uncated]: /access_tok truncated]: access_to ted]: eyJhbGci0iJFUZ]] ioken_type:Bearer] token_type:Bearer] </pre>	<mark>ten:eyJhbGciOiJFUz</mark> k <u>ken:evJhbGciOiJFUz</u> iNiIsImtpZCI6Im5y	INNISIMtpZCI6	<mark>Im5yZmtleT 6Im5vZmtle</mark> AiOiJKVlQi	EilCJ0eXAIOIJKVIQIfO.eyJpc3MiOIIwYzc2NTA4NC05Y2MILTO5Y2YtOTg3NiI TEilCJ0eXAIOIJKVIQIfO.evJpc3MiOIIwYzc2NTA4NC05Y2MILTO5Y2YtOTg3Ni fQ.eyJpc3MiOiIwYzc2NTA4NC05Y2MILTQ5Y2YtOTg3Ni1hZTJmWWZhMmE2M2YiLC	LZTJmNWZh hZT]mNWZ JZdWIi0i
	 Member: access_token (Path with value [tr String value [truncs String value [truncs Key: access_token [Path: /access_token [Path with value: /t [Member with value: /t [Member with value: /t [Member with value: /t [Path with value: /t [Path with value: /t [Member i expires_in [Member with value: /t 	<pre>uncated]: /access_tok fruncated]: access fo tted]: eyJhbGci0iJFUZI] oken_type:Bearer] token_type:Bearer] expires_in:21600] expires_in:21600]</pre>	ten:eyJhbGciOiJFUz kken:eyJhbGciOiJFUz INiIsImtpZCIOIm5y;	<mark>11NiIsImtpZCIG</mark>	<mark>Im5yZmtleT 6Im5yZmtle</mark> AiOiJKV1Qi	EilCJ0eXA101JKV10if0.eyJpc3M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3Ni1 TEilCJ0eXA101JKV10if0.evJpc3M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tr3Ni1 f0.eyJpc3M10iIvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3Ni1hZTJmNWZhMmE2M2YiLC	<mark>h ZTJmNWZh</mark> h ZT <u>mVWZ</u> JzdWIi0i
	 Member: access_token Path with value (frucc Key: access_token IPath: /access_token IPath: /access_token IPath: /access_token IPath: /access_token IPath: /access_token Kember: token_type IPath: /token_type Path: /token_type Member: expires_in [Path with value: /fe [Member value: 21600 	<pre>uncated]: /access_tok fruncated]: access_to tted]: eyJhbGci0iJFU21 [] [] [] [] [] [] [] [] [] [] [] [] []</pre>	<mark>ken:eyJhbGciOiJFUz</mark> <u>ken:evJhbGciOiJFUz</u> INiIsImtpZCI6Im5y;	<mark>IlNiIsIntpZCI6</mark> ZIINiIsIntpZCI ZmtleTEiLCJ0eX	Im5yZmtleT 6Im5yZmtle AiOiJKV1Qi	EILC10eXAi01JKV101f0,eyjpc3Mi01IwYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1 TEILC10eXAi01JKV101f0,eyjpc3Mi01IwYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1 f0,eyjpc3Mi01IwYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1hZTJmWWZhMmE2M2YiLC	<mark>ZTJMNWZh</mark> hZTJMNWZ
	 Member: access_token IPath with value [trunct Key: access_token Path: /access_token Member: token_type [Path with value: /access_token Member: token_type [Path with value: Bearer Key: token_type [Path: /token_type [Path: /token_type [Path: /token_type [Path: with value: ?dember with value: ?dember [Member with value: ?dember with value: ?dember [Member with value: ?dember with	<pre>uncated]: /access_to/ truncated]: access_to ted]: eyJhbGci0iJFUZ]] ioken_type:Bearer] token_type:Bearer]</pre>	ten:eyJhbGciOiJFUz ken:evIhbGciOiJFUz :NiIsImtpZCIGIm5y;	[1N115IntpZC16	Im5yZmtleT fIm5yZmtleA	EilCl0eXAl01JKV101f0.eyJpc3Mi01IvYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1 TEilCl0eXAl01JKV101f0.evJpc3Ni01IvYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni f0.eyJpc3Mi01IvYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1hZTJmNWZhMmE2M2YiLC	<mark>IZTJMNWZh</mark> hZTJMNWZ JZdWIIOI
	 Member: access_token (Path with value [tr (Path with value] String value [truncs String value [truncs Key: access_token (Path with value: // (Member with value: // (Member with value: // (Member with value: // (Member with value: / (Member with value: / Number value: 21600 Key: expires_in (Path: /expires_in (Path: /expires_in 	<pre>uncated]: /access_tok fruncated]: access_to tted]: eyJhbGc10iJFU21 oken_type:Bearer] token_type:Bearer] expires_in:21600] expires_in:21600]</pre>	<mark>en:eyJhbGciOiJFUz</mark> ken:evJhbGciOiJFUz 1NiIsImtpZCIOIm5y	INIISINTPZCI6	Im5yZmtleT fim5yZmtle Al0iJKVlQi	EilcJ0eXAiOiJKV1QifQ.eyJpc3MiOIIwYzc2NTA4NC05Y2MILTO5Y2Yt0Tg3NiI TEILCJ0eXAiOiJKV1QifQ.evJpc3WiOiTwYzc2NTA4NC05Y2MILTO5Y2Yt0Tg3Ni fQ.eyJpc3MiOiIwYzc2NTA4NC05Y2MILTQ5Y2Yt0Tg3Ni1hZTJmNwZhMmE2M2YiLC	<mark>LTJMNWZh</mark> hZTJMNWZ JZdWII01
	 Member: access_token Path with value [trunc: Key: access_token Tring value [trunc: Key: access_token IPath: /access_token Member: token_type [Path with value: Bearer Key: token_type [Path with value: Bearer Key: token_type [Path: /token_type] Member: expires_in [Path with value: 21600 Key: expires_in [Path: /cepires_in] [Path: /cepires_in] Member: scope 	<pre>uncated]: /access_to/ truncated]: access_to ted]: eyJhbGc101JFU21] uoken_type:Bearer] token_type:Bearer]</pre>	<mark>ten:eyJhbGciOiJFUz</mark> <u>ken:evJhbGciOiJFUz</u> INiIsImtpZCI6Im5y	INIISINT2CI6	<u>Im5yZmtleT</u> 6 <u>Tm5yZmtle</u> A10iJKV10i	EILC30eXAi013KV101f0.ey3pc3Mi01IwYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1 TEilC10eXAi013KV10if0.ev3pc3Mi01IwYzc2NTA4NC05Y2MILT0SY2Yt0Tg3Ni f0.ey3pc3Mi01IwYzc2NTA4NC05Y2M1LT0SY2Yt0Tg3Ni1hZTJmWWZhMmE2M2YiLC	<mark>h Z JJMWZh</mark> <u>h Z JJMW Z J</u>
	 Member: access_token (Path with value [truck String value [truck String value [truck Key: access_token [Path: /access_token [Path with value: // [Member with value: // [Member: acpires_in [Path with value: 21600 Key: expires_in [Path: /cexpires_in 	<pre>uncated]: /access_tok fruncated]: access fo tted]: eyJhbGci0iJFUZI] oken_type:Bearer] token_type:Bearer] expires_in:21600] expires_in:21600] cope:namf-comm_namf-e</pre>	en:eyJhbGci0iJFUz ken:evJhbGci0iJFUz 1NiIsImtpZCI6Im5y	IlliisImtpZCIG	1m5y2mtle1 61m5y2mtle A101JKv101	EilCl0eXAl01JKV101f0,eyJpc2M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11 TEilCl0eXAl01JKV101f0,evJpc3N101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N1 f0,eyJpc3M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11hZTJmNWZhMmE2M2Y1LC	MSW/MCTS LST LST LST LST LST LST LST LST LST L
	 Member: access_token Path with value (truck Keyiaccess_token IPath: /access_token Yender with value: /a IPath: /acken_type Path with value: /a IPath with value: /a 	<pre>uncated]: /access_tok fruncated]: access_to tted]: eyJhbGci0iJFU21 ul oken_type:Bearer] token_type:Bearer]</pre>	en:eyJhbGciOiJFUz ken:evJhbGciOiJFUz INiIsImtpZCIGIm5y NiIsImtpZCIGIm5y evts]	ilNilsIntpZCI6	<u>Im5y2mt le T</u> 6 <u>Im5y2mt le</u> A101JKv101	Eilcl0eXAi0iJKV10if0,eyjpc3Mi0iIwYzc2NTA4NC65Y2MILT05Y2Yt0Tg3Ni1 TEILCl0eXAi0iJKV10if0.evjpc3Mi0iIwYzc2NTA4NC65Y2MILT05Y2Yt0Tg3Ni f0.eyjpc3Mi0iIwYzc2NTA4NC05Y2MiLT05Y2Yt0Tg3Ni1hZTJmWWZhMmE2M2YiLC	<mark>h∑TJmNWZh h∑TimNwZW</mark> J JZdWIi0i
	<pre>> Member: access_token</pre>	<pre>uncated]: /access_to/ truncated]: access_to ted]: eyJhbGci0iJFUZI] ooken_type:Bearer] token_type:Bearer] expires_in:21600] expires_in:21600] expires_inanf-comm_namf- scope:namf-comm_namf-c</pre>	en:eyJhbGci0iJFUz ken:evIhbGci0iJFUz SNiIsImtpZCI6Im5y NiIsImtpZCI6Im5y evts] evts]	IlNIISImtpZCIG	1m5y2mtleT 6fm5y2mtle A101JKv101	EILC30eXA1013KV101f0.ey3pc3M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11 TELLC30eXA1013KV101f0.ev3pc3M101IvYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N1 f0.ey3pc3M1011vYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11hZTJmNwZhMmE2M2Y1LC	TJmNWZh IzTimWz JzdWI101
	 Member: access_token Path with value [tr String value [truncs String value [truncs String value [truncs Key: access_token IPath: /access_token IPath: /access_token IPath: /access_token IPath: value: /a [Path with value: /a [Path	<pre>uncated]: /access_tok fruncated]: access_to tted]: eyJhbGciOiJFUZI uoken_type:Bearer] token_type:Bearer] expires_in:21600] expires_in:21600] expires_in:21600] expires_inamf-comm_namf- scope:namf-comm_namf- comm_namf-evts</pre>	en:eyJhbGci0iJFUz ken:evJhbGri0iJFUz 'NNISImtpZCI6Im5y' 'NNIS wits] evts]	INIISIMtpZCI6	1m5yZmtleT 6Tm5yZmtleA	EILCJ0eXAI0JJKV10jf0.eyjpc3Mi0JIwYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1 TEILCJ0eXAI0JJKV10jf0.eyjpc3Mi0JIwYzc2NTA4NC05Y2MILT05Y2Yt0Tg3Ni1 f0.eyjpc3Mi0iIwYzc2NTA4NC05Y2MiLT05Y2Yt0Tg3Ni1hZTJmNWZhMmE2M2YiLC	ZTJmNWZh hZTmtWz JzdWIi0i
	<pre>V Member: access_token (Path with value (trucc Key: access_token IPath: /access_token IPath: /access_token Vember: token_type (Path with value: /acress_token Key: token_type (Path: /token_type) Member: expires_in (Path with value: /ac (Member with value: /ac (Member value: 21600 Key: expires_in (Path with value: / Member: scope (Path with value: / Member with value: / Member with value: / Member with value: / Member: scope (Path with value: / Member with value:</pre>	<pre>uncated]: /access_to/ truncated]: access_to ted]: eyJhbGc101JFU21 uoken_type:Bearer] token_type:Bearer] expires_in:21600] expires_in:21600] expires_in:21600] expires_in:21600] expires_in:21600] expires_in:21600] expires_in:21600]</pre>	en:eyJhbGciOiJFU2 ken:evJhbGciOiJFU2 INiIsImtpZCIGIm5y NiisImtpZCIGIm5y evts] evts]	IINIISImtpZCI6	Im5yZmtleT	EILC30eXA1013KV101f0.ey3pc3M101IwYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11 TEILC30eXA1013KV101f0.ev3pc3N101IwYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N1 f0.ey3pc3M1011wYzc2NTA4NC05Y2M1LT05Y2Yt0Tg3N11hZTJmWWZhMmE2M2Y1LC	ZTJmNUZh JzUmlJZ JzUmlJJ



Next, HALO, again masquerading as the rogue SMF on Slice 2, presents to the shared AMF network function the AMF access token that does not specify a particular slice. Line 24 of Figure 7 shows an HTTP/2 POST specific to the UE on Slice 1 (SUCI/IMSI 310014791791003), providing to the AMF (172.17.20.108) the AMF access token received from the NRF as shown in Figure 6. Subsequently, HALO uses the Namf_Communication_N1N2MessageTransfer service for the rogue SMF to initiate a Request to Release the existing PDU connection using the needed parameters acquired by snooping the PDU session setup prior. Line 28 of Figure 8 shows the message to the AMF (172.17.20.108) requesting the release of PDU Session ID 1, the PDU session associated with the UE as originally established in Figure 3.

Finally, Figure 9 shows the UE packet capture in which the UE receives the PDU session release command. The PDU session is then released and torn down.

•••			4	working_masque	erade_slice	2_amfHalo-dsb.pcap
	1 🖉 💿 🚞 🗋 🕺	🗿 ९ 👄 🔿	2 🖌 🛓			LΞ
Apply	a display filter <೫/>					
No.	Time	Source	Destination	Protocol	Length	Info
	19 2023-11-06 17:20:05.4587	172.17.20.108	172.17.96.50	TCP	72	8443 → 50140 [ACK] Seq=4847 Ack=2797 Win=60160 Len=0 TSval=2029027275 TSecr=41174
	20 2023-11-06 17:20:05.4588	172.17.20.108	172.17.96.50	HTTP2	145	SETTINGS [0]
	21 2023-11-06 17:20:05.4589	172.17.96.50	172.17.20.108	TCP	72	50140 → 8443 [ACK] Seq=2797 Ack=4920 Win=58112 Len=0 TSval=4117468350 TSecr=20290
	22 2023-11-06 17:20:05.4593	172.17.20.108	172.17.96.50	HTTP2	103	SETTINGS [0]
	23 2023-11-06 17:20:05.4594	172.17.96.50	172.17.20.108	TCP	72	50140 → 8443 [ACK] Seq=2797 Ack=4951 Win=58112 Len=0 TSval=4117468350 TSecr=20290
	24 2023-11-06 17:20:05.4630	172.17.96.50	172.17.20.108	HTTP2	523	<pre>HEADERS[1]: POST /namf-comm/v1/ue-contexts/imsi-310014791791003/n1-n2-messages, W</pre>
	25 2023-11-06 17:20:05.4630	172.17.20.108	172.17.96.50	TCP	72	8443 → 50140 [ACK] Seq=4951 Ack=3248 Win=59904 Len=0 TSval=2029027279 TSecr=41174
	26 2023-11-06 17:20:05.4635	172.17.96.50	172.17.20.108	HTTP2	636	DATA[1]
	27 2023-11-06 17:20:05.4635	172.17.20.108	172.17.96.50	TCP	72	8443 → 50140 [ACK] Seq=4951 Ack=3812 Win=59392 Len=0 TSval=2029027280 TSecr=41174
	28 2023-11-06 17:20:05.4637	172.17.96.50	172.17.20.108	HTTP2/JSON/	103	DATA[1], JavaScript Object Notation (application/json), PDU session release comma
	29 2023-11-06 17:20:05.4637	172.17.20.108	172.17.96.50	TCP	72	8443 → 50140 [ACK] Seq=4951 Ack=3843 Win=59392 Len=0 TSval=2029027280 TSecr=41174
	30 2023-11-06 17:20:05.4650	172.17.96.50	172.17.20.108	HTTP2	103	SETTINGS[0]
L	31 2023-11-06 17:20:05.4650	172.17.20.108	172.17.96.50	TCP	72	8443 → 50140 [ACK] Seq=4951 Ack=3874 Win=59392 Len=0 TSval=2029027281 TSecr=41174
	32 2023-11-06 17:20:05.4719	172.17.20.108	169.254.20.10	DNS	115	Standard query 0xbca8 A mtrdmcamf01.amf.5gc.mnc014.mcc310.3gppnetwork.org
	Index: 7 Header::path:/namf-comm/v1/ Name Length: 5 Name::path Value Length: 61 Value:/namf-comm/v1/ue-co [Unescaped:/namf-comm/v1/ Representation: Literal He Index: 4 Header: content-type: multipar/re Name Length: 12 Name: content-type: multipart/related; Value: multipart/related; Content-type: multipart/related; Unescaped: multipart/related; Index: 31 Header: authorization: Bearer Name: content-type: Saver Header: Authorization Name: authorization Name: authorization Name: authorization Value [runcated]: Bearer	<pre>/ue-contexts imsi- intexts/imsi-310014 ntexts/imsi-310014 ue-contexts/imsi-3 ader Field with In art/related; boundary=Mult tade; boundary=Mult tade; Field with In - eyJhbGci0iJFUZIN eyJhbGci0iJFUZIN</pre>	10014791791003, 791791003,n1-n2 791791003,n1-n2 10014791791003,n cremental Indexis inyr=HultipartData ListBoundary UtipartDataListBou cremental Indexis HiISImtpZCI6Im5yZ IsImtpZCI6Im5yZm	1-n2-messages messages L-n2-messages] ng - Indexed Nar ListBoundary ndary] ng - Indexed Nar mtleTEiLCJ0eXAi	ne 0iJKV1QifQ	Q.eyJpc3M101IwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3N11hZTJmNWZhMmE2M2Y1LCJzdWI101IyYTczNWVKM50 eyJpc3M101IwYzc2NTA4NC05Y2M1LTQ5YzYtOTg3N11hZTJmNWZhMmE2M2Y1LCJzdWI01IyYTczNWVKM50
	authorization [truncated]: [Unescaped [truncated]: Be	Bearer eyJhbGci0i arer eyJhbGci0iJFU	JFUzI1NiIsImtpZC zI1NiIsImtpZCI61	I6Im5yZmtleTEiL0 n5yZmtleTEiLCJ00	CJ0eXAiOi eXA101JKV	JKV1QifQ.eyJpc3Mi0iIwYzc2NTA4NC05Y2M1LTQ5YzYt0Tg3Ni1hZTJmNWZhMmE2M2YiLCJzdWIi0iIyYTc3 LQifQ.eyJpc3Mi0iIwYzc2NTA4NC05Y2M1LTQ5YzYt0Tg3Ni1hZtJmNwZhMmE2M2Y1LCJzdWIi0iIyYTc2NWV
	Representation: Literal He	ader Field never I	ndexed - Indexed	Name		
	Index: 23					
~	Header: content-length: 533					
	Name Length: 14					
	Name: content-length					
0 7	working masquerade slice? amfHalo-del	hocan				Packete: 55 . Displayed: 55 (100.0%)
	Norking_masquerade_sidez_ammai0=dsi	o.poop				Profile: Delaut

Figure 7: Rogue SMF Presentation of AMF Access Token to AMF

						🚄 working_m	asquerade_slice	2_amfHalo-dsb.pcap	. 1
				2 0 4				- · · ·	
		۲		<u> </u>	' 🏓 🞽 🕦			 ↓ 	
Apply	a displa	ay filter «	<೫/>						+
No.	Ti	me		Source	Destination	Protocol	Length	Info	
	19 2	023-11-0	6 17:20:05.4587.	172.17.20.10	172.17.96	.50 TCP	72	8443 → 50140 [ACK] Seq=4847 Ack=2797 Win=60160 Len=0 TSval=2029027275 TSecr=41174	
	20 2	023-11-0	6 17:20:05.4588.	172.17.20.10	172.17.96	.50 HTTP2	145	SETTINGS[0]	
	21 2	023-11-0	6 17:20:05.4589.	172.17.96.50	172.17.20	.108 TCP	72	50140 → 8443 [ACK] Seq=2797 Ack=4920 Win=58112 Len=0 TSval=4117468350 TSecr=20290	
	22 2	023-11-0	6 17:20:05.4593.	172.17.20.10	172.17.96	.50 HTTP2	103	SETTINGS[0]	
	23 2	023-11-0	6 17:20:05.4594	172.17.96.50	172.17.20	.108 TCP	72	50140 → 8443 [ACK] Seq=2797 Ack=4951 Win=58112 Len=0 TSval=4117468350 TSecr=20290	
	24 2	023-11-0	5 17:20:05.4630	172.17.96.50	172.17.20	.108 HTTP2	523	HEADERS[1]: POST /namf-comm/v1/ue-contexts/imsi-310014791791003/n1-n2-messages, W	- 1
	25 2	923-11-0	6 17:20:05.4630.	172.17.20.10		100 100	636	0443 → 50140 [ACK] SEQ=4951 ACK=3240 WIN=39904 LEN=0 ISV8(=202902/2/9 ISECT=411/4	-
1	27 2	023-11-0 023-11-0	6 17:20:05.4635	172.17.20.10	172.17.96	50 TCP	72	8443 → 50140 [ACK] Seg=4951 Ack=3812 Win=59392 Len=0 TSval=2029027280 TSecr=41174	
	28 2	023-11-0	6 17:20:05.4637.	172.17.96.50	172.17.20	.108 HTTP2/JS	50N/ 103	DATA[1], JavaScript Object Notation (application/ison), PDU session release comma	
	29 2	023-11-0	6 17:20:05.4637.	172.17.20.10	172.17.96	.50 TCP	72	8443 → 50140 [ACK] Seg=4951 Ack=3843 Win=59392 Len=0 TSval=2029027280 TSecr=41174	
	30 2	023-11-0	6 17:20:05.4650.	172.17.96.50	172.17.20	.108 HTTP2	103	SETTINGS [0]	
L	31 2	023-11-0	6 17:20:05.4650.	172.17.20.10	172.17.96	.50 TCP	72	8443 → 50140 [ACK] Seq=4951 Ack=3874 Win=59392 Len=0 TSval=2029027281 TSecr=41174	
	32 2	023-11-0	6 17:20:05.4719.	172.17.20.10	169.254.2	0.10 DNS	115	Standard query 0xbca8 A mtrdmcamf01.amf.5gc.mnc014.mcc310.3gppnetwork.org	
		00	Jecc						0
		\sim	Member: n1Messa	ageClass		e3 eu1			0
			[Path with v	alue: /niMessa	igeContainer/niMe	ssagec lass: SMJ			0
			String value	 value: nimess sm 	agectassishij				0
			Kev: n1Messa	neClass					0
			[Path: /n1Me	ssageContainer	/n1MessageClass]				0
		~	Member: n1Messa	ageContent	,				
			v Object						
			Member: co	ontentId					
			[Path w	ith value: /n1	MessageContaine	<pre>/n1MessageConter</pre>	t/contentId:P	duSessionReleaseCommand]	
			Monhon	with values of	ontontTd. DduCocc	ionReleaseCommar	d]		
			String	value: PduSess	ionReleaseCommar	nd			
			Key: co	ntentId					
			[Path:	/n1MessageCont	ainer/n1Message0	content/contentio	11		
			[Path: (n1Me	gecontent		+1			
		Key	n1MessageCont	tainer	/minessageconcer				0
		[Pi	ath: /n1Message	Containerl					
		Member	r: n1n2FailureT	xfNotifURI					
		[Pi	ath with value:	/n1n2FailureT	xfNotifURI:http:	//10.194.27.2:40	40/notificati	ns/amf/n1-n2-messages/v1/referenceid/1050265072]	
		[Me	ember with value	e: n1n2Failure	TxfNotifURI:http	://10.194.27.2:4	040/notificat	.ons/amf/n1-n2-messages/v1/referenceid/1050265072]	
		St	ring value: http	p://10.194.27.	2:4040/notificat	ions/amf/n1-n2-m	essages/v1/re	erenceid/1050265072	
		Ke	y: n1n2FailureT	xfNotifURI					
		[Pi	ath: /n1n2Failu	reTxfNotifURI]					
		 Member 	r: pduSessionId						
		TM.	ambor with value.		d.11				
		LP10	mber with Value	e: puusessioni	0.11				
		Key	v: pduSessionId						
		IPa	ath: /pduSession	nIdj					
	Bound	lary: \r\	nMultipartDat	taListBoundary	\r\n				
\sim	Encap	sulated	multipart part:	: (applicatio	n/vnd.3gpp.5gnas)			Fr
0 7	workin	ig_masquer	ade_slice2_amfHalo-d	isb.pcap				 Packets: 55 · Displayed: 55 (100.0%) Profile: Defa 	ult
									-

Figure 8: PDU Session Release Request from Rogue SMF

a 23-	-11-06_17.18.54RID	-/16_UE.pcap					- U
ile 8	Edit View Go C	apture Analyze Statisti	cs Telephony Wireless	Tools Help			
6.10	/ @ 😑 🗁 🕅	8 🖸 🍳 👄 🔿 🖙 7		2 11			
1/	74	1.6-)		•			
(]:(eu	1.STC == 74:80:96:DE:1	1:0e)					
lo.	Time	Source	Destination	Protocol	Lengtł Info		
	55 38.709670	10.205.67.204	192.168.48.130	SCTP	126 HEARTBEAT		
	56 38.709678	192.168.48.130	10.205.67.204	SCIP	126 HEARTBEAT_ACK		
	72 66 996189	10.205.07.205	192.100.40.130	NGAP/N	114 DownlinkWASTransport DL WAS transport PDU session release command (P	agular d	leactivation)
	73 66 906268	192 168 48 130	10 205 67 204	NGAP/N	150 SACK (Ack=8 Acword=32768) UnlinkNASTransport III NAS transport PDU	session	release complete
	74 66.913920	10.205.67.204	192.168.48.130	NGAP/N	130 SACK (Ack=10, Arwind=32768), DownlinkNASTransport, DL NAS transport, 5	GSM stat	us (Invalid PIT value
	75 67,005930	192,168,48,130	10,205,67,204	SCTP	62 SACK (Ack=9, Arwnd=32768)		(
	86 86.788068	10.205.67.205	192.168.48.130	SCTP	1314 HEARTBEAT PAD		
	87 86.905971	192.168.48.130	10.205.67.204	NGAP	102 UEContextReleaseRequest		
	88 86.912337	10.205.67.204	192.168.48.130	NGAP	106 SACK (Ack=11, Arwnd=32768) , UEContextReleaseCommand		
		✓ value				0000	00 50 56 84 20 6c 66
		✓ NAS-PDU: 7e02f	6bc6ad5047e0068010005	2e0500d324:	1201	0010	00 64 de f3 00 00 3c
		Non-Access-	Stratum 5GS (NAS)PDU			0020	30 82 96 0C 96 0C e1
		✓ Security	protected NAS 5GS me	ssage		0040	40 2f 00 00 03 00 0a
		Exten	ded protocol discrimin	nator: 5G m	obility management messages (126)	0050	55 00 05 c0 6c 00 00
		0000	= Spare Half Octo	et: 0		0060	bc 6a d5 04 7e 00 68
		(0010 = Security header	type: Int	egrity protected and ciphered (2)	0070	01 00
		Messa	ge authentication code	e: Øxt6bc6a	d5		
		Sequer	S EGS Mossage				
		- Fidin NA	ded protocol discrimin	ator: 56 m	nchility management messages (126)		
		0000	= Spare Half Oct	t: 0	obility management messages (120)		
			0000 = Security header	type: Pla	in NAS message, not security protected (0)		
		Messa	ge type: DL NAS trans	ort (0x68)			
		0000	= Spare Half Octo	et: 0			
		✓ Paylo	ad container type				
			0001 = Payload cont	ainer type	: N1 SM information (1)		
		✓ Paylo	ad container				
		Ler	ngth: 5				
		P1a	ain NAS 5GS Message				
			Extended protocol dis	criminator	: 5G session management messages (46)		
			PDU session identity:	PDU sessi	on identity value 5 (5)		
	v se command (Avd3)						
		~	565M cause	sion refea			
			565M cause: Regula	r deactiva	tion (36)		
		✓ PDU set	ession identity 2 - PI	U session	ID		
		Ele	ement ID: 0x12				
		PDU	J session identity: PE	U session	identity value 1 (1)		
			,				

Figure 9: UE Packet Capture Showing PDU Session Release Command

Expected Results:

- The compromised NF is allowed to access the AMF shared network function, and a valid token for the shared AMF NF is generated by the NRF. The token contains the AuthenticationTokenClaims, which define the scope of the token for the Namf_Communication_N1N2MessageTransfer service. The token is sent to the Compromised SMF Test NF by the NRF.
- 2. The Shared AMF NF responds to the maliciously crafted REST API call and releases the PDU session, thus indicating that the vulnerability exists.

Results:

Success Criteria	Status
A valid token for the shared AMF NF is generated by	The NRF, in response to a request from
the NRF and provided to the rogue NF.	the rogue SMF on Slice 2, without any
	slice specified in request, provides the
	AMF an access token without slice
	restriction.
The token contains the	The NRF-generated token includes
AuthenticationTokenClaims.	namf-comm and namf-evts in scope.
Shared AMF NF responds to the maliciously crafted	After the rogue SMF sends a PDU
REST API call and releases the PDU session, thus	session release request to the AMF for
indicating that the vulnerability exists.	the UE on Slice 1, the UE receives a PDU
	session release command and tears
	down the PDU session.
Overall Test	Confirmed

Test Case 2: Static Subscriber S-NSSAI Override of UE-Requested NSSAI by Dynamic Core Signaling Assignment

Test Case ID: TC-NetSlic-02

Description:

The purpose of this test case, which can also be called "5G Network Override to Reroute a Mis-Provisioned User Slice," is to validate the ability of the cellular network to override a misprovisioned, mis-configured, or manipulated user device (UE) that has the network slice provisioned inconsistently with how it is recorded by the network operator.

In this test, the UE presents Slice 1 as the Requested NSSAI to the network. However, the UE subscription data in the UDM/UDR has the proper provisioning for Slice 2. Slice 2 is configured as the default subscribed S-NSSAI in the UDM Slice Profile, and this setting is expected to override the Requested NSSAI given by the UE to the network.

The AMF is the network function that performs this override function as it retrieves the subscribed S-NSSAI from the UDM and sends to the UE the Allowed and Configured NSSAI, which includes the subscribed S-NSSAI in order to ensure that the network wins in a mismatch. The Rejected NSSAI is not sent as this is used for the Serving PLMN in a roaming scenario. These exchanges are defined in 3GPP TS 23.501.^{iv} The exchange of information is shown in Figure 10.

Note: Depending on the UE implementation, the Allowed NSSAI information that was once received during registration might be permanently stored on the device and be used to compose the Requested NSSAI in any following registration procedures including the initial registration after a UE power off-on cycle.



Figure 10: Slice Selection and Use

In the executed test, the UE was provisioned with a SIM card dedicated in the core network to slice configuration SST=1 and SD=2; however, the SIM card itself was configured for slice configuration SST=1 and SD=1. It had not previously attached to the 5G core with this SIM card. As shown in Figure 11, the IMSI was 310014791791022 and the IMEI was 351735110121742. From Figure 12, we confirm the default configured NSSAI as SST=01, SD=000001. Also, as expected, it does not contain any PLMN values.

Manufacturer: Sierra Wireless, Incorporated Model: Sierra Wireless EM9190 Firmware: 03.09.06.00_GENERIC Network type: GSM Data class: UMTS, HSDPA, HSUPA, LTE, 5G-NSA, 5GSA/TDS IMEI: 351735110121742 Mobile number: 7917910022 IMSI: 310014791791022 SIM ICCID: 8901002000011163739

Figure 11: UE SIM Card Details



Figure 12: SIM NSSAI Details from AT Command

The test started with the UE in airplane mode. Upon exiting airplane mode, the UE attached to the 5G network. During the attach process, we collected attach logs data for the UE, and slice information using AT (attention) commands.

Figure 13 shows the UE-requested NSSAI details from the Registration Request message in packet 1620, indicating the requested S-NSSAI=1, SST=1, and SD=1. However, the NSSAI details from Core downlink Registration Accept message shows both the Allowed NSSAI and Configured NSSAI as S-NSSAI=1 and SST=1, but the SD has changed to SD=2 (see Figure 14, packet 1633).

	TC5	testing	10-05	-23_092	20_ pro	file-B-UB	E_attach.p	pcapng					- [1
File	E	dit Vi	ew	Go C	apture	Analy	rze Stat	tistics Telepl	hony	Wireless	Tools	Help		
		1 🖲		D	۵	۹ 🤃	• 🔿 🖻	1 1 🕹 📃		ଭ୍ଭ୍€	1			
	ngap													1
No.			Time					Source		Destination	1	Protocol	Length Info	
Г		1604	2023	10-05	09:1	7:09.2	60762	10.220.67.	18	10.205.6	57.204	NGAP/NAS-5GS	GS 178 InitialUEMessage, Registration request	
		1609	2023	10-05	09:1	7:09.4	21569	10.205.67.	204	10.220.6	57.18	NGAP/NAS-5GS	GS 138 DownlinkNASTransport, Authentication request	
		1612	2023	10-05	09:1	7:09.4	90273	10.220.67.	18	10.205.6	57.204	NGAP/NAS-5GS	GS 138 UplinkNASTransport, Authentication response	
		1616	2023	10-05	09:1	7:09.5	36347	10.205.67.	204	10.220.6	57.18	NGAP/NAS-5GS	GS 114 DownlinkNASTransport, Security mode command	
		1620	2023	10-05	09:1	7:09.5	50330	10.220.67.	18	10.205.6	57.204	NGAP/NAS-5G	G 254 SACK (Ack=1, Arwnd=16384), UplinkNASTransport, Security mode complete, Registration request	_
		1621	2023	10-05	09:1	7:09.5	77951	10.205.67.	204	10.220.6	57.18	NGAP/NAS-5GS	GS 138 SACK (Ack=2, Arwnd=32768), DownlinkNASTransport, Security mode command	
		1622	2023	10-05	09:1	7:09.5	90233	10.220.67.	18	10.205.6	57.204	NGAP/NAS-5GS	GS 154 SACK (Ack=2, Arwnd=16384), UplinkNASIransport, Security mode complete	
		1628	2023	10-05	09:1	7:09.9	32144	10.205.67.	204	10.220.6	57.18	NGAP	1/8 InitialContextSetupRequest	
		1629	2023	10-05	09:1	7:09.9	50304	10.220.67.	18	10.205.0	7.204	NGAP	926 SALK (ACK=3, ArWnd=16384), UERadioCapabilityInfoIndiCation	
		1630	2023	10-05	09:1	7:09.9	50314	10.220.67.	18	10.205.0	7.204	NGAP	90 InitialContextSetupResponse	
		1655	2023	-10-05	09:1	7:09.9	90808	10.205.67.	204	10.220.0	.18	NGAP/NAS-505	us 1/0 Downlinkwastransport, kejistration accept	
		1054	2023	10-05	09:1	7:10.0	05259	10.220.67.	10	10.205.0	7 10	NGAP/NAS-505	as 142 SACK (ACK=4, Arwind=16364), OpiinkivAsiransport, kegistration complete	
		1055	2025	-10-05	09:1	/:10.0	25290	10.205.07.	204	10.220.0	07.10	NGAP/NAS-505	ds 136 SACK (ACK=6, Arwhu=52768), DownlinkwaStransport, Configuration update Command	
									0.	= EIA4	4: Not	supported		
									0	= EIA	5: Not	supported		
										0. = EIA	5: Not	supported		
									· · · · <u>· ·</u>	.0 = EIA	7: Not	supported		
								NSSA1	I - Re	quested	NSSAI			
								El	lement	ID: 0x2	F			
								Le	ength:	5				
								× S-	NSSAI	1				
									Leng	th: 4				
									Slic	e/service	e type	(SST): eMBB (1	(1)	
									Slice	e differe	entiato	or (SD): 1		
								✓ UE ne	etwork	capabil	179			
<i>_</i> ··		1 -		~ ~ ^			c		ement	10: 0X1	, 			
FIC	lui	'e 1:	5: IV.	SSAI	De	tails	trom	i UE Rec	qıstr	ation	кеді	Jest Messo	sage	



Figure 14: NSSAI Details from Core Downlink Registration Accept Message

Figure 15 shows how, after the UE attached, the SIM subsequently displayed PLMN=310014 and was configured with NSSAI as SST=01 and SD=000002, as well as Allowed NSSAI SST=01, SD=000002. Note the SD values differ from the original SIM configuration and match the configuration in the core.

Figure 15: SIM NSSAI Details from AT Command after Successful UE Attach

Expected Results:

- 1. When the UE registers to the 5GS, the Requested NSSAI for Slice 1 is seen in the NAS REGISTRATION REQUEST message.
- 2. The AMF provides the UE with the Allowed and Configured NSSAI information as well as the NSSAI inclusion mode in the NAS REGISTRATION ACCEPT message. The AMF should embed Slice 2 in this information, which it received from the UDM slice profile. This will override the Requested NSSAI Slice 1 from the UE. This will be seen using the Software Probe in the PCC for the N2 interface.
- 3. The UE is connected to Slice 2.

Results

Success Criteria	Status
The UE, though provisioned with a Requested	AT Command on UE shows Slice 1
NSSAI for Slice 1, is overridden by the network	(SD=1) as the configured slice, but after
using the Default subscribed S-NSSAI in the	attaching, shows Slice 2 (SD=2).
subscriber UDM slice profile (Slice 2).	The registration messages show the
	same circumstance, with the UE
	requesting registration to Slice 1 and the
	core accepting registration to Slice 2.
The UE connects successfully to Slice 2	The UE successfully attached.
Overall Test	Success

Conclusions and Next Steps

The 5G Security Test Bed's network slicing phase 2 tests were designed to produce, in a realworld environment, two potential exploits enabled by a theoretical network slicing vulnerability in 5G cores. Both scenarios are niche cases that are extremely difficult to execute in practice, and only one exploit was confirmed to be executable, while the other was mitigated by the network. After the Test Bed shared its findings with 3GPP, the standards body released updated technical standards resolving the vulnerability.

Summary of Test Results

Test Case 1 was designed to replicate the exploit described in the 2021 AdaptiveMobile Security Common Vulnerability Disclosure report to GSMA. The test confirmed that a modified request could be sent from a rogue network function and slice, then shared with other network functions. To do so, the NRF generated a valid token for the shared AMF network function and provided it to the rogue network function. The shared AMF network function responded to the maliciously crafted token and granted the rogue network function access to the network, thus indicating that the vulnerability existed and was exploitable. The Test Bed reported its findings to 3GPP, leading to updated standards resolving the potential issue.

Test Case 2 was designed to test a second exploit and confirm that the network would override an incorrectly provisioned UE by comparing its network slice request to the network slice provisions recorded by the network operator. To do so, the UE was incorrectly provisioned with a Requested NSSAI for Slice 1, but the network overrode it using the default subscribed S-NSSAI in the subscriber UDM slice profile (Slice 2). This forced the UE to connect to the correct Slice 2.

Both tests successfully confirmed the Test Bed's hypotheses: Test Case 1 verified that the network slice identifier vulnerability could be exploited in the hypothetical scenario defined in the AdaptiveMobile report. Test Case 2 confirmed that for the second potential exploit, defined by the 5G Security Test Bed's Technical Advisory Committee, the network's built-in controls prevented the vulnerability from being exploited, and the mis-provisioned UE was redirected to its correct slice.

5G Security Test Bed Strengthened 3GPP Standards

As a result of AdaptiveMobile's Common Vulnerability Disclosure, the 3rd Generation Partnership Project (3GPP) standards body began assessing and addressing the hypothetical vulnerability in new releases, but 3GPP could not create definitive standards until the scenario was validated.

The 5G Security Test Bed's efforts, detailed in this report, were the first time the vulnerability was confirmed on a real-world network. As a result of the Test Bed reporting its findings to the

standards body, 3GPP was able to conclusively validate the vulnerability and add a solution in updated technical standards. 3GPP's Service and System Aspects Working Group 3 (SA3) has since fixed this vulnerability within a Change Request (CR) that was accepted and implemented recently in TS 33.501, *Security architecture and procedures for 5G system*, Release 18, in section 13.4.1.1.2.

Specifically, 3GPP now requires the NF Service Producer to confirm that the NF Service Consumer has authorization to access at least one of the slices the UE is registered to, by crosschecking and verifying that the UE's slice identifiers match the NF Service Producer's slice identifiers listed in the access token.

TS 33.501 added the following language to ensure that with shared NFs, the NF slice identifier is now checked:

If applicable (e.g., when the request is for information related to a specific UE), the NF Service Producer may check that the NF Service Consumer is allowed to access (as indicated by the NF Service Producer's NSSAIs in the access token presented by the NF Service Consumer) at least one of the slice(s) that the UE is currently registered to, e.g., by verifying that the UE's allowed NSSAI(s) intersect with the NF Service Producer's NSSAIs in the access token.

The successful execution of these tests also reflects the 5G Security Test Bed's value as a platform for testing and validating theoretical scenarios in real-world conditions, providing validation of use cases that had previously been discussed only in theory.

Next Steps

As new participants and the diversity of test cases grow, the 5G Security Test Bed will continue contributing to the evolving future of 5G network security. The 5G Security Test Bed is exploring future tests of network function security, false base stations, roaming security, and aspects of 5G cloud security that arise with use of the Network Exposure Function (NEF), the Application Function (AF), and Multi-access Edge Computing (MEC). The Test Bed is also exploring opportunities to test configurations and enhance Open Radio Access Network (Open RAN) security.

For more information, or to participate in the 5G Security Test Bed, please contact Harish Punjabi (<u>hpunjabi@ctia.org</u>; (202) 845-5701), or visit <u>https://5gsecuritytestbed.com/</u>.

About the 5G Security Test Bed

The 5G Security Test Bed reflects the industry's collaborative approach to 5G security—it was created by the Cybersecurity Working Group (CSWG), an industry initiative that convenes the world's leading telecom and tech companies to assess and address the present and future of cybersecurity. The Test Bed's members are wireless providers AT&T, T-Mobile, and UScellular; industry partners Ericsson, the MITRE Group, SecureG, Intel, and Syniverse; and academic partners the University of Maryland and Virgina Tech Advanced Research Corporation (VT-ARC).

The 5G Security Test Bed has a Technical Advisory Committee (TAC) made up of its members and the Test Bed Administrator. The TAC advises the Test Bed Administrator on the day-to-day technical and operational activities and decisions related to the Test Bed, including but not limited to: development of use cases to be tested, test plan development and review, raw test data analysis, test result and report generation, and development of recommendations to standards bodies based on results.

The 5G Security Test Bed further works with a broad array of government agencies, policymakers, international standards bodies, thought leaders, and partners in the telecommunications and information technology sectors. These groups include the 3rd Generation Partnership Project (3GPP), the International Telecommunication Union (ITU), the Department of Homeland Security (DHS), the National Institute of Standards and Technology (NIST), and the Federal Communications Commission (FCC), among others.

The 5G Security Test Bed Uses Real-World Equipment, Validating Real-World Applications

Real-World Testing

The 5G Security Test Bed advances wireless security by:

- Conducting real-world tests in a rigorous, transparent, and replicable manner that can assess and validate theoretical and policy concerns and overcome hypothetical laboratory testing limitations.
- Drawing on the expertise of government, wireless providers, and equipment manufactures to evaluate specific use cases and support new equipment development.
- Testing security functionality in different scenarios, enabling industry and government to identify, mitigate, and respond to evolving threats while protecting consumers, businesses, and government agencies.

Real-World Applications

The 5G Security Test Bed's tests and outcomes support several applications that can drive new technology and transform cities, government, and industries. Use cases include government and enterprise applications, general network security protections, and smart city applications such as:

- Primary Use Cases: Network Security
 - o Protecting Information in Transit
 - o Roaming Security
 - o Subscriber Privacy
 - o Zero Trust Network Security
 - o False Base Station Detection and Protection
 - o 5G Cloud Network Security

• Secondary Use Cases: Devices and Applications

- o High-Resolution Video Surveillance (e.g. Smart Cities, Large Venues)
- o LTE/5G Drones with High-Resolution Video Feedback (e.g. Smart Cities)
- o Dynamic Supply Chain Verification (Real-Time Monitoring and Logistics)
- o Automated, Reconfigurable Factories
- o Autonomous Vehicles
- o Immersive AR/VR

The 5G standalone network architecture and network slicing capability tested for this report are key components of these applications because they enable service to be customized to diverse needs and requirements. The test cases outlined here show how these new and evolving uses can successfully adopt enhanced security capabilities while improving performance.

Appendix: Acronyms

3GPP	Third Generation Partnership Project
5G STB	5G Security Test Bed
5GS	5G System
AMF	Access & Mobility Management Function
AMS	AdaptiveMobile Security
AT Command	Attention Command
BBU	Baseband Unit
СИОМ	Core Network Operations Manager
СР	Control Plane
CSRIC	Communications Security, Reliability, and Interoperability Council
CVD	Common Vulnerability Disclosure
DMC	Dual-Mode Core
DN	Data Network
DNN	Data Network Name
eMBB	Enhanced Mobile Broadband
eNB/eNodeB	Evolved Node B
FCC	Federal Communications Commission
FDD	Frequency Division Duplex
gNB/gNodeB	Next Generation Node B
GSMA	Global System for Mobile Communications Association
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPsec	Internet Protocol Security
LTE	Long Term Evolution
mMTC	Massive Machine-Type Communication
MNO	Mobile Network Operator
mTLS	Mutual Transport Layer Security
MTP	Mobile Test Platform
NAS	Non-Access Stratum
NF	Network Function
NIST	National Institute of Standards and Technology
NR	New Radio
NRF	Network Repository Function
NSA	Non-Standalone
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
PCAP	Packet Capture
PCC	Packet Core Controller

PCF	Policy Control Function
PDU	Protocol Data Unit
PGW	Packet Data Network Gateway
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
ppm	Packets per million
RAN	Radio Access Network
RAT	Radio Access Technology
SA	Standalone
SBA	Service-Based Architecture
SD	Slice Differentiator
SDR	Software-Defined Radio
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
SST	Slice/Service Type
SUCI	Subscription Concealed Identifier
TAS	Telecom Application Server
ТС	Test Case
TLS	Transport Layer Security
ТР	Test Point
UDM	Unified Data Management
UDR	Unified Data Repository
UE	User Equipment
UMD	University of Maryland
UP	User Plane
UP URI	User Plane Uniform Resource Identifier
UP URI URLLC	User Plane Uniform Resource Identifier Ultra-Reliable Low-Latency Communication
UP URI URLLC VNF	User Plane Uniform Resource Identifier Ultra-Reliable Low-Latency Communication Virtualized Network Function
UP URI URLLC VNF VPN	User Plane Uniform Resource Identifier Ultra-Reliable Low-Latency Communication Virtualized Network Function Virtual Private Network

References

¹ 5G Security Test Bed, Securing 5G: Network Slicing Phase 1 Test Report, Q1 2023. Available at

^a AdaptiveMobile Security, *A Slice in Time: Slicing Security in 5G Core Networks*, March 24, 2021. Available at <u>https://www.enea.com/insights/white-paper-slicing-security-in-5g/</u>

^{III} 3GPP TS 33.501, Security architecture and procedures for 5G System, Release 18.

^{iv} 3GPP TS 23.501, System architecture for the 5G System (5GS), Release 16.6.0.