Confused Producer Attack On 5G core Network

1 Introduction

The 5G system offers significant improvements in data speed, latency, and reliability compared to previous cellular networks. However, opening up the 5G core network to third parties presents new security and access control challenges. To address these challenges, the 5G system utilizes the OAuth 2.0 authorization framework as its access control mechanism for the first time. However, there has been no formal analysis of this access control mechanism to date.

The Service-Based Architecture (SBA) of 5G system enables third-party partners to set up additional/complementary core components, e.g., slices on top of network provider's. The introduction of third parties presents new security and access control challenges. To address these challenges, the 5G system utilizes the OAuth 2.0 authorization framework as its access control mechanism for the first time. However, there has been no formal security analysis of the design of this access control mechanism to date.

In our recent research endeavor, we attempted to formally verify the access control mechanism of 5G core. However, this poses several critical challenges, including the lack of commercially deployed 5G Core networks, incomplete open-source implementations, and scalability issues due to numerous configurations.

To address these challenges, we developed NGCoreVerifier, a model-based framework that leverages parameterized model checking to test various 5G core network configurations. We reduce the problem of verifying the access control mechanism of 5G core into a model checking problem. Our framework incorporates a modular design and several 5G system-specific abstraction mechanisms to ensure flexibility, customizability, and scalability of

our analysis. Our testing upon 73 safety properties on 27 different core network configurations uncovered six new weaknesses in the 5G access control mechanism which can be exploited by an compromised NF to gain unauthorized access to various sensitive resources. A research paper on our work is currently under review.

We responsibly disclose one of our findings in the following.

2 Confused Producer Attack

2.1 Main Idea

The vulnerability is triggered when an NRF registers multiple NF instances of the same NFType and authorizes only a subset of them for a certain consumer NF. If the consumer NF requests an access token for a specific NFType, the NRF shall grant it for that subset of producers. However, our analysis reveals the consumer NF can use the same token to access any producer NF of the same NFType, even those that the consumer is not authorized for. Currently, there is no way for an NF service producer to determine whether the access token is coming from an authorized NF service consumer or not. This scenario only happens when the token is requested for a specific NFType.

2.2 Example Demonstrating the Vulnerability

Consider a partial core network setup (Figure 1) where a compromised NF Service consumer C_1 (e.g. AMF) wants some services from a certain NF Service producer (e.g. UDM) registered in NRF. Assume that there are two UDM instances such as P_2 and P_3. Consumer NF C_1 and candidate producer NF P_3 are in the same slice, i.e., sNssai 1 whereas another candidate producer NF P_2 is in different a network slice, i.e., sNssai 3. Assume that C_1 wants some services from UDM. Before service access, C_1 needs an access token from NRF. So, C_1 will invoke an access token request (which is for specific NF Type) to the NRF (step ①). NRF verifies that C_1 has permission to access P_3, and grants it an *accessToken* T containing proper scopes for P_3 (step ②). However, instead of making the service request to P_3, C_1 uses *accessToken* T to request a service from P_2 (step ③). Finally, P_2 shall verify the attributes in T and, grants service access to P_2(step ③ & ④). Note

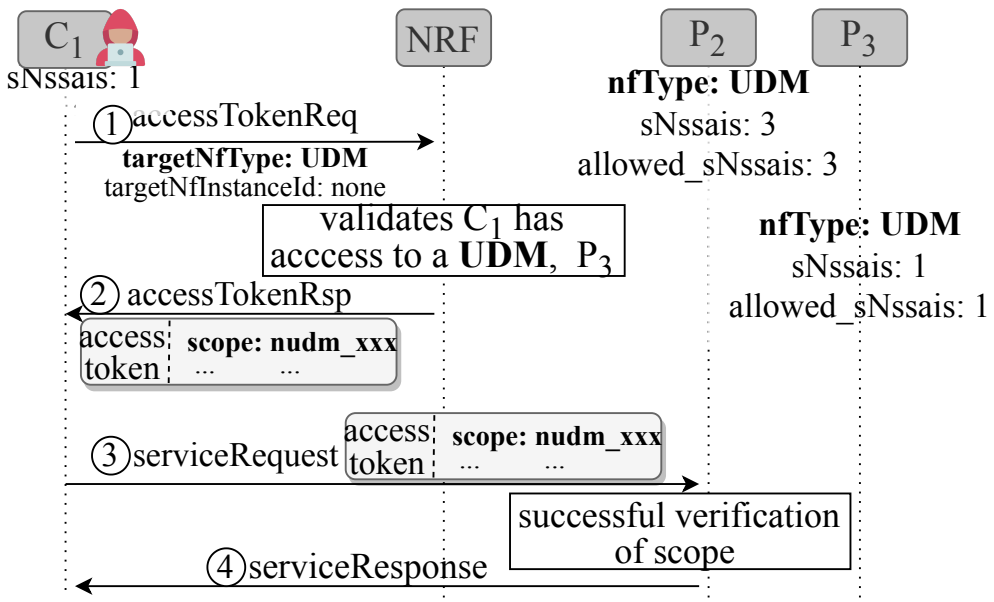

Figure 1: Demonstration of Confused Producer Attack

that this vulnerability is generic to producers of any NFType, not limited to UDM.

2.3 Adversary Assumption

To successfully carry out the attack, C_1 needs to be malicious. Also, the adversary requires the knowledge of victim's (P_2) FDN/IP address to make the service request.

2.4 Root Cause

The above attack is possible because the identifying information of the producer NF instance (e.g. NFInstanceID of NF Set ID, etc.) is absent in the access token for which the access token is granted. This information gap causes ambiguity/confusion when the producer NF receives a service request based on the access token, as it cannot determine with certainty if the access token is granted for the that instance or other instances of the same NF type. Although there is an attribute called *producerSnssaiList*, this attribute is *optional*.

2.5 Possible Fixes

From our speculation, two fixes are possible.

1. **Addition of NFInstanceID and NFSetID.** When the access token request is for a specific NF Type (i.e. not for a specific NF instance), access token shall include the NFInstanceID and NFSetID (if applicable) of the producer NF.

2. **Making producerSnssaiList attribute conditional.** *producerSnssaiList* (and possibly other authorization attributes as well) can be made *conditional* when the access token request is for a specific NF Type. However, this workaround is **not recommended** because of two reasons. First, the verification of authorization attributes has already been done once by NRF. Repeating the verification process in the producer NF can result in additional overhead and inefficiencies. It also, makes the access token bulky, i.e., the size of the token larger. Secondly, the responsibility of verifying authorization attributes lies on NRF. If this verification process is shifted to the producer NF, the role and existence of NRF may become questionable.