ETSI GS NFV-SEC 022 V4.5.1 (2024-01)

Network Functions Virtualisation (NFV) Release 4; Security; Access Token Specification for API Access

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGS/NFV-SEC022ed451

Keywords
API, authentication, authorization, NFV, security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The common aspects for RESTful NFV MANO APIs have been defined in ETSI GS NFV-SOL 013 [22].

The ETSI NFV-MANO APIs are only allowed to be accessed by authorized consumers.

The Authorization of API Request and Authorization of notifications sending has been defined in SOL group. One solution for authorizing access is the use of OAuth with access token.

The aim of the present document is to define the Access Token for this access Authorization and associated procedure for the verification of the Access Token, ensuring security and interoperability. The present document results in a NFV profile of the OAuth2.0 for the NFV-MANO API Request and notification sending Authorization.

1 Scope

The present document defines the access tokens and related metadata for RESTful protocols and data model for ETSI NFV Management and Orchestration (MANO) interfaces. It defines also the process for the token verification by the API Producer.

For this aim, the present document:

- Analyses the security threat arising from the misuse of the access token and defines the security requirements associated to access token.
- Analyses existing specifications related to access token for API access and their compliancy with the requirements defined.
- Defines the token request and generation profile, the token format and associated metadata considering the result of existing access token specifications analysis.
- Defines the token verification procedures for the API Producer.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[2] ETSI GS NFV-SEC 002: "Network Functions Virtualisation (NFV); NFV Security; Cataloguing security features in management software".

[3] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".

[4] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification".

[5] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

[6] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[7] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

[8] IETF RFC 7519: "JSON Web Token (JWT)".

[9] IETF RFC 3339: "Date and Time on the Internet: Timestamps".

[10] IETF RFC 7515: "JSON Web Signature (JWS)".

[11] IETF RFC 7516: "JSON Web Encryption (JWE)".

[12] NIST Special Publication 800-90B: "Recommendation for the Entropy Sources Used for Random Bit Generation", January 2018.

[13] IETF RFC 8414: "OAuth 2.0 Authorization Server Metadata".

[14] IETF RFC 7033: "WebFinger".

[15] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification".

[16] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[17] IETF RFC 8615: "Well-Known Uniform Resource Identifiers (URIs)".

[18] IETF RFC 7591: "OAuth 2.0 Dynamic Client Registration Protocol".

[19] IETF RFC 7517: "JSON Web Key (JWK)".

[20] IETF RFC 7518: "JSON Web Algorithms (JWA)".

[21] IETF RFC 7662: "OAuth 2.0 Token Introspection".

[22] ETSI GS NFV-SOL 013: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".

[23] IETF RFC 8705: "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens".

[24] ETSI GS NFV-IFA 005: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".

[25] ETSI GS NFV-IFA 006: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Openstack®: "All about keystone tokens".

[i.2] OpenID® Connect Core 1.0 incorporating errata set 2.

[i.3] IETF RFC 8471: "The Token Binding Protocol Version 1.0".

[i.4] IETF RFC 6819: "OAuth 2.0 Threat Model and Security Considerations".

[i.5] ETSI GS NFV-SEC 006: "Network Functions Virtualisation (NFV); Security Guide; Report on Security Aspects and Regulatory Concerns".

[i.6] ETSI TS 133 501: "5G; Security architecture and procedures for 5G System (3GPP TS 33.501)".

[i.7] draft-ietf-oauth-token-binding-08: "OAuth 2.0 Token Binding", Work in progress.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS NFV 003 [1] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [1] and the following apply:

3GPP	3rd Generation Partnership Project
HSM	Hardware Security Module
JRD	JSON Resource Descriptor
JSON	JavaScript Object Notation
JWE	JSON Web Encryption
JWS	JSON Web Signature
JWT	JSON Web Token
MAC	Message Authentication Code
MTLS	Mutual TLS
NRF	Network Resource Function
OTP	One-Time Password
PKI	Public Key Infrastructure
REST	REpresentational State Transfer
SAML	Security Assertion Markup Language
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

4 Security requirements for API access tokens

4.1 Authorization for API access using OAuth2.0 defined in ETSI GS NFV-SOL 013

4.1.0 Authorization for API access using OAuth2.0

The requirements on interfaces supported by the reference point of MANO's entities have been defined in ETSI GS NFV-IFA 005 [24], ETSI GS NFV-IFA 006 [25], ETSI GS NFV-IFA 007 [3], ETSI GS NFV-IFA 013 [4] and ETSI GS NFV-IFA 008 [5].

One of these requirements concerns authentication and authorization of the API consumer for all operations on interfaces supported by the reference point.

To fulfil this requirement for the NFV-MANO reference points, authorization of API requests and notifications has been defined in ETSI GS NFV-SOL 013 [22].

One solution defined to handle these authorizations for API request and notification is the use of OAuth 2.0 protocol as defined by IETF RFC 6749 [6].

4.1.1 Mapping roles for Authorization for API access using OAuth2.0

For API calls, the producer functional block of an API in NFV terms corresponds to the *"resource server"*, and the consumer functional block of an API corresponds to the *"client"* as defined by IETF RFC 6749 [6]. For sending a notification, these roles are reversed: the producer (notification sender) corresponds to the *"client"*, and the consumer (notification receiver) corresponds to the *"resource server"*.

Before invoking an HTTP method on a REST resource provided by a *resource server*, a consumer functional block (referred to as "*client*" from now on) first obtains authorization from another functional block fulfilling the role of the "*authorization server*".

4.1.2 Authorization grant for Authorization for API access using OAuth2.0

Authorization grant, which is a credential representing the resource owner's authorization to access the API resources is used by the client to obtain an access token from the authorization server as defined by IETF RFC 6749 [6]. OAuth 2.0 defined 4 types of authorization grant (authorization code, implicit, resource owner password credentials, and client credentials).

For the reference points listed in clause 4.1, access to API is performed by a machine which is a non-interactive Client, acting on its own behalf and being the Resource owner. Example of such client is the EM requesting the creation of an instance of its related VNF to the corresponding VNFM; this EM is the resource owner for the management resource of the VNF. The authorization grant suitable to this case is the client credentials authorization grant. This is the authorization grant type that has been selected for the NFV-MANO interfaces and defined in ETSI GS NFV-SOL 013 [22].

4.1.3 High level procedures for API access and notifications using OAuth2.0

The roles and exchanges are shown in figure 4.1.3-1 in case of API calls and in figure 4.1.3-2 for sending a notification.

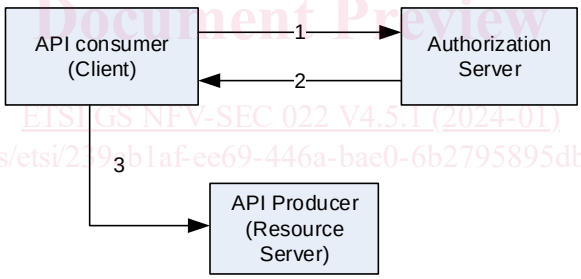

Figure 4.1.3-1: OAuth 2.0 roles in case of API calls

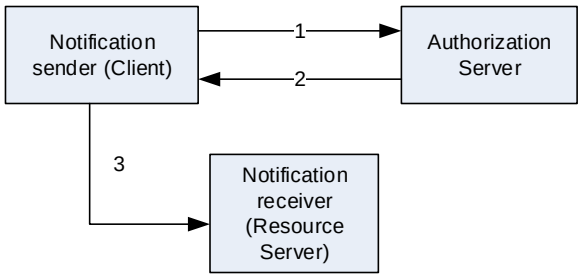

Figure 4.1.3-2: OAuth 2.0 roles in case of sending notifications

NOTE: The numbered steps below correspond to the steps of figure 4.1.3-1.

The procedure for API access is as follows:

Step 1. Before invoking the RESTful HTTP based API on the API producer, the API consumer authenticates with an Authorization Server by presenting its credentials consisting of its Client Id and Client Secret. It is assumed that authorization-related configuration parameters such as the client credentials are pre-populated in the API consumer together with other information such as the address of the *token endpoint* exposed by the *authorization server*.

Step 2. The Authorization server after authentication and validation of the API consumer returns an *access token*.

Step 3. The API consumer uses the access token in the API Request.

Same procedure is used for the notifications case shown in figure 4.1.3-2.

4.1.4 Access token for API access and notifications using OAuth2.0

An *access token* represents a particular access right (defining the particular set of protected resources to access in a particular manner) with a defined duration. The access token is usually used as a Bearer credential and transmitted in an HTTP Authorization header to the API. The token may denote an identifier used to retrieve the authorization information or may self-contain the authorization information in a verifiable manner (i.e. a token string consisting of some data and a signature). Access tokens can have different formats, structures, and methods of utilization (e.g. cryptographic properties) based on the resource server security requirements.

IETF has defined two aspects of access token use:

1) Bearer token as defined by IETF RFC 6750 [7] focuses on the transmission of the access token as an opaque string and makes no assumption about the structure of the token.

2) JSON Web Token (JWT) as defined by IETF RFC 7519 [8] focuses on the structure of the token, and allows it to be encrypted (JWE) or signed (JWS).

ETSI GS NFV-SOL 013 [22] specifies the transmission aspects of the token as a bearer token, according to the definitions by IETF RFC 6750 [7].

The present document analyses the different access token used by different standardization and open source organizations and the security threats around this access token.

4.2 Threat Analysis

4.2.0 Access token defined in ETSI GS NFV-SOL 013

The access token defined by ETSI GS NFV-SOL 013 [22] to authorize access to the API of NFV MANO interfaces is the bearer token as defined in IETF RFC 6750 [7].

The bearer token is defined in IETF RFC 6750 [7] as follows:

"Bearer Token: A security token with the property that any party in possession of the token (a "bearer") can use the token in any way that any other party in possession of it can. Using a bearer token does not require a bearer to prove possession of cryptographic key material (proof-of-possession)."

The Authorization grant type defined by ETSI GS NFV-SOL 013 [22] is the client credentials type as defined in IETF RFC 6749 [6].

4.2.1 Risk analysis and assessment

This threat analysis takes as basis the OAuth 2.0 Threat Model as presented in IETF RFC 6819 [i.4]. This risk analysis in table 4.2.1-1 uses the format found in Annex A of ETSI GS NFV-SEC 006 [i.5].

Table 4.2.1-1: Risk analysis and assessment

A Security Environment		
a.1 Assumptions		
a.1.1	It is assumed that the attacker has access to the communication between the client (API consumer) and the authorization server, and between the client (API consumer) and the resource server (API producer).	
a.1.2	An attacker has unlimited resources to mount an attack.	
a.1.3	Two of the three parties involved in the OAuth protocol may collude to mount an attack against the 3rd party.	
a.2 Assets		
a.2.1	Access token.	
a.2.2	Refresh Token.	
a.2.3	Protected Resources.	
a.2.4	Client id, client credentials.	
a.3 Threat agents		
a.3.1	Malicious authorization server: this malicious authorization server delivers bogus token and get access to client credentials or refresh token (and then obtains access token with the refresh token by counterfeiting the client).	Threats: a.4.2.2. a.4.2.3 a.4.2.4 a.4.3.3 a.4.3.6
a.3.2	Malicious client: this malicious client may modify the content of the token.	Threats: a.4.1.2
a.3.3	Attacker of client: This attack could be through malicious software within the client itself.	Threats: a.4.1.7 a.4.1.8 a.4.1.2 a.4.1.3 a.4.1.4 a.4.1.5 a.4.1.9 a.4.2.1 a.4.2.3 a.4.2.4 a.4.3.1 a.4.3.2 a.4.4.3
a.3.4	Malicious resource server: this malicious resource server gain access to the access token sent by the client by counterfeiting the resource server.	Threats: a.4.1.11 a.4.1.2 a.4.1.3 a.4.1.4 a.4.1.5 a.4.1.9 a.4.4.3 a.4.4.4

a.3.5	Malicious entity acting as a Man in the Middle on the communication between Authorization server and client.	Threats: a.4.1.1 a.4.1.2 a.4.1.3 a.4.1.4 a.4.1.5 a.4.1.9 a.4.3.4 a.4.3.6 a.4.4.1 a.4.4.3
a.3.6	Malicious entity acting as a Man in the Middle on the communication between the Client and the resource server.	Threats: a.1.4.10 a.4.1.2 a.4.1.3 a.4.1.4 a.4.1.5 a.4.1.9 a.4.4.1 a.4.4.2 a.4.4.3
a.3.7	Attacker of the Authorization server: This attack could be through malicious software within the Authorization server itself.	Threats: a.4.1.6 a.4.1.2 a.4.1.3 a.4.1.4 a.4.1.5 a.4.1.9 a.4.3.5 a.4.4.3
a.4 Threats		
a.4.1 Threats on access token		
a.4.1.1	Token Interception or token eavesdropping in transit from authorization server and client.	Mitigation by: b.1.3 b.1.4
a.4.1.2	Token Manipulation.	Mitigation by: b.1.1 b.1.18 b.1.22
a.4.1.3	Token disclosure - misuse.	Mitigation by: b.1.2 b.1.3 b.1.4 b.1.10 b.1.12 b.1.13 b.1.14 b.1.21 b.1.24
a.4.1.4	Token redirect.	b.1.3 b.1.4 b.1.10 b.1.12 b.1.13 b.1.14 b.1.21 b.1.24

a.4.1.5	Token replay.	b.1.3 b.1.4 b.1.10 b.1.12 b.1.13 b.1.14 b.1.21 b.1.24
a.4.1.6	Obtaining Access tokens from authorization server database.	b.1.4 b.1.10 b.1.12 b.1.13 b.1.14 b.1.21 b.1.24
a.4.1.7	Attacker of client obtains access tokens from the storage device.	b.1.25 b.1.9 b.1.10 b.1.12 b.1.14
a.4.1.8	Redirection on client to malicious server: Attacker of client takes the control of the client and get access to token and authorization code.	b.1.25 b.1.7 b.1.9 b.1.10 b.1.12 b.1.14
a.4.1.9	Guessing the access token.	b.1.4 b.1.10 b.1.12 b.1.13 b.1.14 b.1.17 b.1.21 b.1.24
a.4.1.10	Token Interception or token eavesdropping in the request sent from client to resource server.	b.1.4 b.1.25 b.1.10 b.1.12 b.1.13 b.1.14 b.1.16 b.1.21 b.1.24
a.4.1.11	A malicious resource server gain access to a valid access token sent by a legitimate client.	b.1.4 b.1.25 b.1.10 b.1.11 b.1.12 b.1.13 b.1.16 b.1.19 b.1.20 b.1.21 b.1.23 b.1.24

a.4.2 Threats on refresh token		
a.4.2.1	Attacker of client obtains refresh token stored in client.	b.1.6 b.1.7 b.1.9
a.4.2.2	Refresh token phishing by counterfeiting the authorization server.	b.1.4 b.1.5 b.1.6 b.1.8
a.4.2.3	Refresh token replay.	b.1.4 b.1.5 b.1.6 b.1.8
a.4.2.4	Guessing the refresh token.	b.1.26
a.4.3 Threats on client credentials		
a.4.3.1	Attacker obtains client secrets from source code.	b.1.9
a.4.3.2	Attacker obtains client secrets from a client installation.	b.1.9
a.4.3.3	malicious authorization server get access to client credentials.	b.1.15
a.4.3.4	Disclosure of client credentials during client authentication process or token requests.	b.1.27 b.1.28
a.4.3.5	Obtaining client secrets from authorization server database.	b.1.29
a.4.3.6	Guessing the client credentials.	b.1.30
a.4.4 Threats on protected resources		
a.4.4.1	An attacker eavesdrops Access tokens on transport and gain access to the protected resources.	b.1.13 b.1.21
a.4.4.2	Replay of authorized resource server requests.	b.1.16
a.4.4.3	Gain access to the protected resources by guessing the access tokens.	b.1.17
a.4.4.4	Malicious resource server gain access to a valid access token and uses it to gain access to protected resources.	b.1.19 b.1.13 b.1.21 b.1.23
B Security Objectives		
b.1 Security objectives for the asset		
b.1.1	Integrity protection of the token using digital signature or Message Authentication Code (MAC).	
b.1.2	Confidentiality protection.	
b.1.3	Access tokens should not be sent in clear over insecure channel. Use Secure transmission as TLS.	
b.1.4	Binding of the token to ID of authorized party.	
b.1.5	The authorization server should validate the client id associated with the refresh token.	
b.1.6	Revocation of refresh tokens.	
b.1.7	Revocation of client secrets.	
b.1.8	Refresh token rotation.	
b.1.9	Store secrets in secure storage.	
b.1.10	Limit token scope.	
b.1.11	Limit the token to a resource server.	
b.1.12	Limit lifetime of the access token, short access token duration.	
b.1.13	Binding of access token to client id, and client prove legitimate ownership of the token to the resource server.	
b.1.14	Allow one-time access token usage.	
b.1.15	Verification of authorization server's authenticity.	
b.1.16	Resource server uses transport security measures to avoid replay attacks (TLS) or uses signed requests with nonces and timestamps.	
b.1.17	Access token should have a reasonable level of entropy making the guessing of the token infeasible.	