ETSI GS NFV-SEC 002 V1.1.1 (2015-08)

Network Functions Virtualisation (NFV);
NFV Security;
Cataloguing security features in management software

Reference

DGS/NFV-SEC002

Keywords

NFV, open source, 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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, 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.

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.

1 Scope

The present document gives a survey of the security features in the open source management software relevant to NFV, in particular OpenStack™ [i.1] as the first case study. It addresses the OpenStack modules that provide security services (such as authentication, authorization, confidentiality protection, integrity protection, and logging) together with the full graphs of their respective dependencies down to the ones that implement cryptographic protocols and algorithms. It also identifies a set of recommendations on the use of and enhancements to OpenStack as pertinent to NFV.

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 reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://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.

Not applicable.

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 reference 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.

NOTE: http://www.openstack.org/.

[i.2] United States Computer Emergency Readiness Team.

NOTE: http://www.us-cert.gov/.

[i.3] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.4] ETSI GS NFV-SEC 001: "Network Functions Virtualisation (NFV); NFV Security; Problem Statement".

[i.5] ETSI GS NFV 004: "Network Functions Virtualisation (NFV); Virtualisation Requirements".

[i.6] ETSI GS NFV-MAN 001: "Network Functions Virtualisation (NFV); Management and Orchestration", (work in progress).

[i.7] Memcached.

NOTE: http://memcached.org/.

[i.8] OpenID Connect.

NOTE: http://openid.net/connect/.

[i.9] IETF Application Bridging for Federated Access Beyond web (ABFAB) Working Group.

NOTE: http://tools.ietf.org/wg/abfab/charters.

[i.10] IETF RFC 5905 (June 2010): "Network Time Protocol Version 4: Protocol and Algorithms Specification".

NOTE: https://tools.ietf.org/html/rfc5905.

[i.11] IEEE 1588-2008 (July 2008): "IEEE Standard for a Precision Clock Synchronization for Networked Measurement and Control Systems".

[i.12] The OpenStack Security Guide.

NOTE: http://docs.openstack.org/sec/.

[i.13] Trusted Computing Group: Storage Work Group Storage Security Subsystem Class: Opal.

NOTE: http://www.trustedcomputinggroup.org/resources/storage_work_group_storage_security_subsystem_class_opal.

[i.14] IETF RFC 3164: "The BSD syslog Protocol".

[i.15] IETF RFC 5424: "The Syslog Protocol".

[i.16] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[i.17] FIPS PUB 186-4: "Digital signature Standard".

[i.18] DMTF: "Cloud Auditing Data Federation (CADF)".

NOTE: Available at: http://www.dmtf.org/standards/cadf.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS NFV 003 [i.3] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in [i.3] and the following apply:

AMQP	Advanced Message Queuing Protocol
AH	Authentication Header
API	Application Program Interface
ARP	Address Resolution Protocol
CADF	Cloud Auditing Data Federation
CMS	Cryptographic Message Syntax
DHCP	Dynamic Host Configuration Protocol
DMTF	Distributed Management Task Force
ESP	Encapsulating Security Payload
GRE	Generic Route Encapsulation
HMAC	Hashed Message Authentication Code
HTTP	HyperText Transfer Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
JSON	JavaScript Object Notation
KVS	Key Value Stores

LDAP	Lightweight Directory Access Protocol
LUKS	Linux Unified Key Setup
MAC	Media Access Control
MAC/IP	Media Access Control / Internet Protocol
NSS	Network Security Services
NTP	Network Time Protocol
PEM	Privacy Enhanced Mail
PKI	Public Key Infrastructure
PTP	Precision Time Protocol
RPC	Remote Procedure Call
SAML	Security Assertion Mark-up Language
SASL	Simple Authentication and Security Layer
SED	Self Encrypting Drive
SQL	Structured Query Language
SR-IOV	Single Root Input Output Virtualization
SSH	Secure SHell
SSL	Secure Socket Layer
TCP	Transfer Control Protocol
URI	Uniform Resource Identifier
UUID	Universally Unique IDentifier
VLAN	Virtual LAN
VM	Virtual Machine
VNC	Virtual Network Computing
VPN	Virtual Private Network
VXLAN	Virtual eXtensible Local Area Network
WSGI	Web Server Gateway Interface

4 Introduction

Building on open source software can help advance certain goals of NFV, such as accelerated time-to-market and improved interoperability. To do so effectively calls for having a knowledge base of the security features and cryptographic algorithms supported in each relevant code base. In particular, NFV applications are subject to privacy and security regulations. The knowledge base helps shed light on how to best apply the pertinent software and on enhancements necessary to meet the NFV security needs. It is also useful for other reasons. Chief among them are:

- export control of cryptographic software;
- compliance with procurement processes;
- follow-up on alerts from US-CERT [i.2] and other similar organizations; and
- determination of the relevant elements for security analytics.

Such a knowledge base is of particular importance in the area of management and orchestration, which plays a critical role in NFV security.

The present document addresses OpenStack, a widely adopted cloud operating system, as the first case study. It aims to cover all applicable aspects of information and network security, including:

- Identity and access management
- Communication security
- Stored data security
- Firewalling, zoning, and topology hiding
- Availability

- Logging and monitoring
- Compute isolation

NOTE: OpenStack™ is a set of open source tools for building and managing cloud-computing software platforms for public and privvate clouds.

It consists of a group of interrelated projects that control pools of processing, storage, and networking resources throughout a data center e.g. Neutron, Nova, Keystone, Barbican, Swift, Glance, Trove, Cinder, etc.

The present document describes the OpenStack modules that provide security services (e.g. authentication, authorization, confidentiality protection and integrity protection) together with their respective dependencies on cryptographic protocols and algorithms. It also makes a set of recommendations on the use of and enhancements to OpenStack as pertinent to NFV. The case study takes into account the issues identified in ETSI GS NFV-SEC 001 [i.4] and the related requirements specified in ETSI GS NFV 004 [i.5] and ETSI GS NFV-MAN 001 [i.6].

5 Identity and access management

5.1 General

Keystone is the component in OpenStack that provides centralized authentication and authorization. It is used by all OpenStack components for API access control. Hence, at a high level, a user is authenticated by Keystone first before gaining access to any other service (Keystone may employ an external authentication system). Upon successful authentication, the user is given a temporary token. From this point on, to get a service, the user includes the token in the service request. The user can receive the service if and only if the token is validated and if the user has the proper roles.

Keystone is organized as a set of internal services, including the identity service, token service, and catalog service. The identity service handles user authentication and user-data validation. The following constructs are basic to the service:

- User, which may be a person or a process using an OpenStack service.
- Project (or tenant), which owns a set of OpenStack resources. A project shall be assigned a domain.
- Group, which is a set of users. A group shall be assigned a domain. A user may be assigned one or multiple groups.
- Domain, which is a set of users, groups, and projects.
- Role, which specifies a set of rights and privileges. Roles can be granted at either the domain or project level. A group may be assigned one or multiple roles on a domain. A user may be assigned one or multiple roles on a project or domain. An example role is *admin*. A user shall have a role assigned to have access to a resource.

The identity service supports basic management of user data (e.g. create, read, update and delete). It also has the flexibility to use a pluggable authentication or authorization module through a backend. Common backends include Lightweight Directory Access Protocol (LDAP) servers, SQL databases and Key Value Stores (KVS). Keystone uses an SQL backend by default.

The identity service is accessible through a REST API. The corresponding API endpoint is, in fact, the entry point to all services. An endpoint is a network-accessible address in the form of a Uniform Resource Identifier (URI). The identity service may support a separate endpoint for administrative purposes. It goes without saying that the transport of all API access transactions needs to be protected. In general, access control is based on configurable policy stored in a JSON file. Other components in OpenStack can further customize the policy according to their respective service contexts. Keystone supports an SQL policy backend.

The token service deals with token management and validation. It relies on a database to store tokens and the associated data, including the token revocation list (or token revocation events) and per-token information (e.g. lifespan and scope). The scope of a token is determined by a combination of projects (or domains) and roles associated with the user. An unscoped token does not include a specified role. Such a token may be issued during the initial authentication of the user, who can then use the token to discover accessible projects and then exchange it for a scoped token.

As the basis for service access, tokens shall be protected from forgery, and from unauthorized access and alteration in transit and at rest. The token service also provides protection in this regard. Several types of tokens are supported, including Public Key Infrastructure (PKI) and Universally Unique IDentifier (UUID). The token type in use as well as other specifics (e.g. token lifespan) is configurable. The default token type is UUID. Depending on the token type, it may be useful to cache tokens to enhance performance. OpenStack services can be configured to this end. When used, token caches need to be protected and expiration times need to be set appropriately. Custom token types are also possible through external modules.

The catalogue service manages a registry of all OpenStack services. It allows a user to discover the entitled services and the corresponding endpoints. Services can be organized in terms of regions, while endpoints classified as public, internal or administrative. It is also possible to have tenant-specific endpoints. Keystone supports an SQL catalogue backend.

5.2 PKI tokens

5.2.0 General

A PKI token is a Cryptographic Message Syntax (CMS) string, essentially data that are digitally signed and base64 encoded. The specifics of the data signed are context-dependent. They may include information on, for example, the user, tenant, role, trust, timestamp and entitled services. One characteristic of such a token is its long length. It is possible that a PKI token is too long to fit into either a header or URL. To reduce the token size, Keystone supports compression through `zlib`. Still the size of a compressed PKI token is much larger than that of a UUID token.

PKI tokens are verifiable by any API endpoints as long as they have access to Keystone's signing certificate, the information for verifying the signing certificate (i.e. the certificate chain and certificate revocation list), and the token revocation list (or revocation event records). Keystone provides an API for retrieval of relevant signing certificates. Decentralized token validation reduces the chance of Keystone becoming a bottleneck. For this reason, PKI had been the default token type since the Grizzly release. Nevertheless, it has been changed back to UUID in the Juno release based on deployment experience. The concerns are largely due to the large size of PKI tokens.

5.2.1 PKI set-up

Keystone provides the utility for generating the signing key, the corresponding certificate and the certificate chain that are required for token generation and management. The required material may be externally generated and imported. Either way, it is stored in separate files in the Privacy Enhanced Mail (PEM) format in the directories as specified in the Keystone configuration file (i.e. `keystone.conf`). Keystone does not support encryption of private key files but relies on the access control mechanisms of the underlying operating system to protect such files.

The Keystone utility for generating signing keys and certificates is the command `keystone-manage pki_setup`, which is based on OpenSSL. The key size and certificate lifespan are configurable through `keystone.conf`. The signature algorithm in use is RSA-SHA256. RSA is hardcoded in `keystone/common/openssl.py` and SHA256 in `keystoneclient/common/cms.py`.

5.2.2 Token generation

Table 1

Cryptographic module used	`openssl cms -sign -signer /etc/keystone/ssl/certs/signing_cert.pem -inkey /etc/keystone/ssl/private/signing_key.pem -outform PEM -nosmimecap -nodetach -nocerts –noattr –md –sha256`
Signature algorithm	default RSA-SHA1 (with `key-size = 2048`)
Configurability	configurable through the signing certificate and key as part of PKI setup
Invoking module	`keystone/common/cms.py/cms_sign_text()` or `keystoneclient/common/cms.py/cms_sign_text()`

The token lifespan is configurable through `keystone.conf`. The default is one hour.