ONAP Security Recommendation Development

Beijing Security Release Notes Template

1 Introduction

1.1 Purpose

This section captures recommendations for handling certain security questions that are studied by the security sub-committee. These recommendations, when implemented, can lead to new best practices. The recommendation states are:

- **Draft: The ONAP Security sub-committee is working on the recommendation**
- **Recommended: The ONAP security sub-committee agrees that this is a recommendation**
- **Approved: The recommendation is approved by the TSC.**

1.2 Threat Analysis

Some known threats in Micro Service architectures :

1. **Credential stealing and then used get the high level prilvileges:**
    a. **Attacker analyzes the container images to steal secrets such as SSH private keys, X.509v3 certificate private keys, passwords etc...**
    b. **Attacker analyzes the captured traffic among services to steal secrets such as passwords and other secrets.**
    c. **Attacker analyzes environment variables (to containers) via orchestrator log files to steal password and other secrets.**
    d. **Attacker getting hold of default credentials or weak passwords**
    e. **Attacker has credentials because at one point was authorized to have access to credentials, is no longer authorized after a job change, but the credentials have not been changed. New**
2. **Denial Of Service Attacks:**
    a. **Attacker bombards the container services with new connections, leading to large number forked processes and threads leading to resource issues on other workloads (containers) in the system.**
    b. **Attacker exploiting the container to get access to Kernel.**
    c. **Attacker exploits the container runtime and deletes executing containers**
3. **Tampering of images (ONAP container images)**

a. **Attacker keeping tampered images with similar looking name in the registry, leading to running containers from attacker images.**
b. **Attacker has self-commit privileges and introduces malware into the images in the registry.**

Typical vulnerabilities are:

- **Secrets/passwords/sensitive-data in images.**
- **Unchanged default passwords**
- **Weak passwords**
- **Unsecured communication**
- **Usage of environment variables to pass sensitive information**
- **Poor Security configuration**
- **Vulnerable system software and libraries**

Mitigation techniques are:

- **Host operating system (Not valid if ONAP is being installed in Hyperscale data centers) - Hardened operating system, Vulnerability scanning, Trusted computing infrastructure**
- **Containers images:**
  - **Only have required software packages.**
  - **No password, secrets, private key in the image.**
  - **Vulnerable scanning and ensuring only patched versions of the packages are used.**
  - **Trusted image repository / Image signing by VNF vendors.**
- **Container image download**
  - **Secure communication with repositories**
  - **Verifying the signature of images before they are launched.**
  - **Periodic check for patched container images from the repository.**
- **Container run time**
  - **Secret Management**
  - **Mutual TLS for network security**
  - **IPSEC for network security**
  - **Syscall white listing, MAC (Mandatory Access Control)**
  - **Usage of cgroups for resource isolation for all shared resources.**
  - **Monitoring of system call usage**
  - **Immutable - No run time patches to the packages. Always download full container image.**

Open Source Threat Modeling: https://www.coreinfrastructure.org/news/blogs/2017/11/open-source-threat-modeling

1.3 Main discussed topics

The main captured topics (Main focus areas):

1. **ONAP Credential Management & Secret Management**
2. **static code scanning**
3. **Known vulnerability analysis**
4. **Image signing/verification**
5. **3SP support from security perspective (recommendation done)**

2 ONAP Credential Management.

Status: Draft

2.1 ONAP level use cases

The following are the high level onap use cases that need to be supported.

2.1.1 Package signing

A package to be onboarded is signed

When onboarding the package it is validated for integrity.

Note: Need to be clear on whether it is the vendor credential used for signing or the ONAP operator credential.

2.1.2 ONAP operator signing into the manage the onap system

The operational staff that is using ONAP authenticate with the ONAP system and have authorized privilege's based on the authenticated persona.

2.1.3 Secure communication between ONAP components

The ONAP components can securely communicate between themselves

The components are authenticated prior to establishing a connection

The connection is encrypted

2.1.4 External APIs being used to access ONAPs capabilities

ONAP offers API for external systems to use the ONAP capabilities. For this, the external system is authenticated and authorized.

2.1.5 ONAP accessing the services from another system.

ONAP can be a consumer of services offered from other external systems. These can include e.g. the virtulization resources, the VNFs, or other external systems.

2.2 Credentials to be managed

Credentials may be certificates, passwords and the like. These need to be managed through the entire lifecycle. The types of credentials that need to be managed are:

- **Credentials for ONAP users to access ONAP. These are referred to as ONAP_User credentials.**
- **Credentials for using the APIs exposed by ONAP. These are referred to as ONAP_ExtAPI credentials.**
- **Credentials for ONAP to communicate to other ONAP components. These are referred to as ONAP_Component credentials.**
  - **Note: This includes credentials for VNF SDK to package the artifacts onboarded into SDC.**
  - **Note: Other ONAP components include VNFs that need to communicate with ONAP services such as DCAE securely.**
  - **Note: ONAP components can spread across geographical locations. For example, DCAE systems at Edge communicating with Central ONAP services.**
- **Credentials for ONAP to communicate with other systems. These are referred to as ON AP_Foreign credentials.**
  - **As an example, if ONAP is to communicate to an external SDN controller or a cloud infrastructure, these credentials need to be managed.**
  - **A another example is the credentials to access a VNF**

2.3 Credential Management Requirements

The credential management solution considers the following:

General Requirements

- **The credential management solution MUST be able to interact with existing credential creation and validation schemes**

- **The following types of certificates SHOULD be supported by ONAP:**
    - **a, b, c, ...**
- **Securing the private keys - CA private keys shall be secured using PKCS11 based HSMs (e.g secure generation and storage of private key)**
- **Usage of certificate identity wherever possible(binding an identity to a credential using the X. 509v3 certificate)**

Requirements for ONAP_USER credentials:

- **ONAP MUST support ONAP_User credentials of type user-ID and Password**
- **ONAP Should support ONAP_User credentials as certificates.**

Requirements for ONAP_ExtAPI credentials:

- **ONAP MUST support ONAP_ExtAPI credentials of type user-ID and Password**
- **ONAP MUST support ONAP_ExtAPI credentials as certificates.**

Requirements for ONAP_Component credentials:

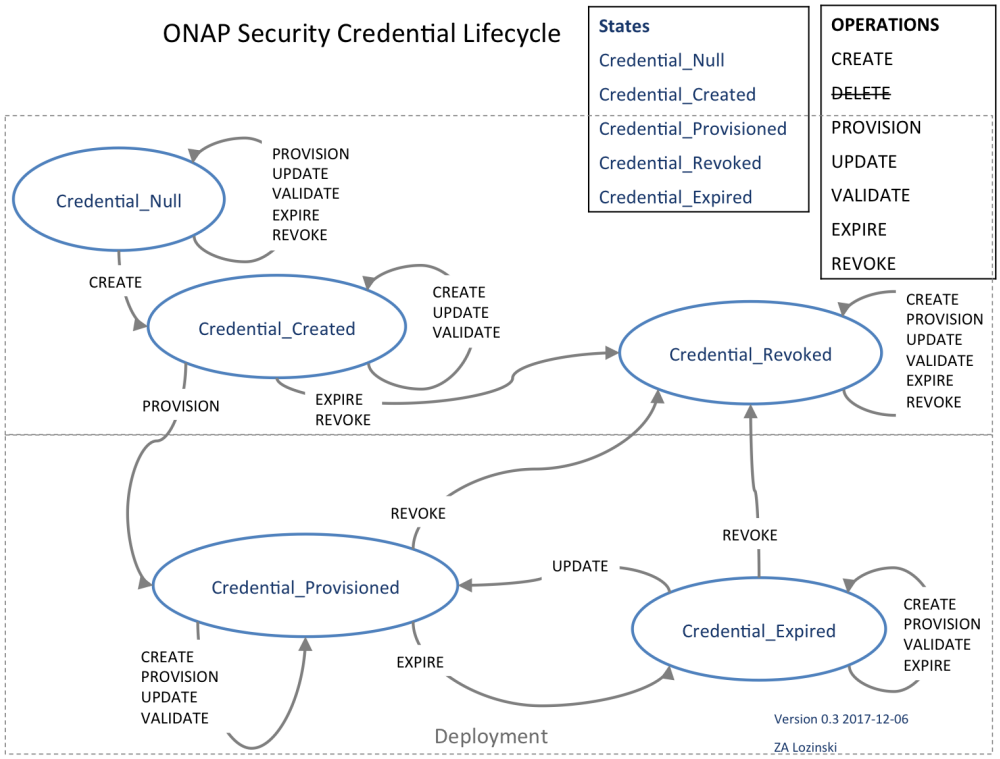
- **~~ONAP MUST support ONAP_Component credentials of type user-ID and Password~~**
- **ONAP MUST support ONAP_Component credentials as certificates.**
- **ONAP components SHOULD use credentials based on certificates for communication with other ONAP components. The use of user-ID and Password is a fallback in the case of components that do not support certificates.**

Requirements for ONAP_Foreign credentials:

- **ONAP MUST support ONAP_Foreign credentials of type user-ID and Password**
- **ONAP MUST support ONAP_Foreign credentials as certificates**

2.4 Credential Lifecycle

2.4.1 Credential State Diagram



In the implementation, some types of credentials have to be provisioned into ONAP components, e. g. certificate-based credentials or (user-ID,password) have to be added to VM images or containers before deployment. It is probably better to do this during the deployment rather than storing images with imbedded credentials. The Secrets Vault is used to store these credentials securely. The transition to the Credential_Provisioned state means the credential is stored in the Secrets Vault.

2.4.2 Credential States

State	Definition
Credential_ Null	No credential currently exists. The only valid operation is to create a credential. (The mechanism for creating a credential is out of scope of ONAP.)
Credential_ Created	A credential has been created. The credential is not yet available within ONAP, and cannot be validated.
Credential_ Provisioned	The credential is provisioned into ONAP. The credential can be validated within ONAP.
Credential_ Expired	The credential has expired. Credential validation within ONAP will fail. The credential can be updated.
Credential_ Revoked	The credential has been revoked. Credential validation within ONAP will fail. The credential cannot be updated.
Credential_ Destroyed	Note: Credentials can be copied, and the copy can be presented for validation. Credentials can never be destroyed.

2.4.3 Credential Operations

Operation	Definition

CREATE	**Creates a new credential. Credential creation is external to ONAP.**
~~DELETE~~	**Credentials may not be deleted. (Design Note 1).**
PROVISION	**Provisions an existing credential into ONAP. A credential must go through state Credential_Provisioned before it can be used within ONAP.**
UPDATE	**Updates an existing credential within ONAP. UPDATE is used to update a credential in state Credential_Expired and return it to state Credential_Provisioned. UPDATE may also be used to update internal parts of a credential.**
VALIDATE	**Validates an existing credential. VALIDATE is used to test that a presented credential gives permission for access to a resource within ONAP (e.g. to access an ONAP component, perform an ONAP operation, or access data).**
EXPIRE	**Expires an existing credential. EXPIRE may be an implicit operation, as some credentials have a defined lifetime, and will expire automatically. EXPIRE may be an explicit operation, where a specific credential is expired. Credentials in state Credential_Expired may be updated.**
REVOKE	**Revokes an existing credential. Once a credential is in state Credential_Revoked there are no valid operations. A new credential is required.**

Design Notes:

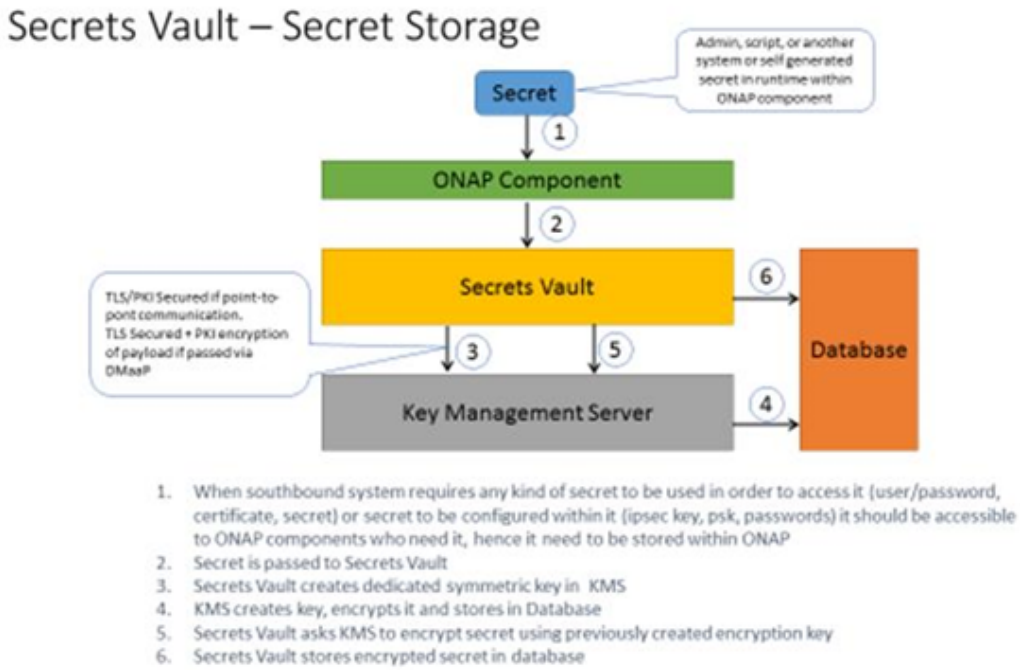
- **Design Note 1 - this is intended to make explicit that digital credentials may always be re-used, even if they are expired or revoked.**

2.4 ONAP Credential Management Overview

ONAP requires two components to improve the security of credentials used in orchestration.

a. **a secrets vault to store credentials used by ONAP**
b. **a process to instantiate credentials**

Component 1: Secrets Vault - A service that can be integrated with ONAP that provides secure storage of the credentials used by ONAP to authenticate to VNFs.



2.5 Credential Management Use cases (credential perspective)

Use Cases:

For ONAP_User Credentials

For ONAP_User Credentials, two uses cases are shown.

1. **Provisioning the credentials**

   The ONAP_Admin credentials are directly provisioned. The root administrator can create the onap admininstrator user-identifier and credentials. Intially a temporary credential is created and the ONAP operational staff can update their credentials.

   The credentials are securely stored (in a hashed format???)

2. **Authenticating the user**

   When a ONAP operational staff attempts to log in for the first time. ONAP challenges the user (with xxxxx). This is done by comparing the hash of the entered credentials with the stored hash of the credentials.

For ONAP_ExtAPI credentials:

There are two cases here. The first case is when the user credentials have to be specifically provisioned. The second case is when an identity management scheme is used. What do we want to describe.

For ONAP_ExtAPI credentials, 3 use cases are described

1. **Provisioning the credentials**

<< insert here >>

2. **Distributing the credentials**

<< Insert here >>

3. Retrieving the credentials

<< Insert here >>

For ONAP_Component credentials:

For ONAP_Component credentials, few use cases are described here

1.Certificate Authority Instance creation : This is normally required to be only one per ONAP deployment.

Steps are given below:

- **Administrator user creates CA instance by providing details such as following to CA Service**
  - **Subject name to use on self-signed CA certificate**
  - **PKCS11 slot ID and Key ID to use (in case PKCS11 based HW protection of CA private key)**
  - **Public key algorithm**
  - **In case of RSA, key size**
  - **In case of ECDSA, curve**
  - **Hash algorithm and key size**
  - **Validity time of CA certificate**
  - **Whether to create token backend. If token backend is needed, life time and usage count of tokens to be supplied.**
  - **Returns:**
    - **Token request URL**
    - **Certificate request URL**
    - **CA Certificate**
- **Administrator user also creates policy rules to apply on user certificate request with information such as**
  - **Subject name prefix, CA instance should accept.**
  - **Signing algorithm, key sizes or curves that are acceptable.**
  - **Hashing algorithm and key sizes that are acceptable.**
  - **MAC addresses it should accept in the subject name**
  - **Whether to verify MAC address in the subject name of PKCS10 request with the MAC address of the VM/Container.**
  - **Check for valid token (Yes/No)**
  - **Validity time of certificate.**

2. Certificat request - Creation of credentials required for secure communication : This normally occurs when service (e.g java application service) is started or when the certificate renewal is due

Steps are given below:

- **Java application gets the CA URL, Token, Subject name prefix to be used via environment variables in case of containers or via cloud-init user data in case of VM.**
- **Certificate Credential Client agent is called by application during its startup to create and get the certificate signed by CA by giving CA URL, token information.**
- **Certificate Credential agent does following:**
  - **If there is an existing certificate and private key and if it is still valid, it returns back to the application immediately. If not, it does following**
  - **Generate ECDSA key pair.**
  - **Create PKCS10 request with subject name prefix + MAC address as Common Name of the subject name.**

- Sends PKCS10 request, token to CA.
- Gets the x.50v3 certificate from CA.
- Stores the certificate in file system.
- Returns back private key handle, slot ID and path to the certificate.

- Certificate credential agent informs application on acquiring credentials
- Application moves forward to inform TLS service with CA certificate and subject prefix to validate incoming requests.
- If Application is making TLS connection to another service, then it uses certificate enrolled and private key handle while creating TLS endpoint.

For ONAP_Foreign credentials:

For ONAP_Foreign credentials, two use cases are described.

1. **Provisioning the credentials**

   <<insert here>>
2. **Retrieving the credentials**
3. **Accessing VNFs during runtime and installation**

<< Describe the flow for the credentials to access VNFs . To be more specif, who owns the credentials for the case when ONAP has to configure the VNFs>> (Zyg)

4. onboarding VNFs.

<< Describe the case where the VNF image and VNF package is signed from the vendor (with or without VNF package) >>

Assumptions: Vendor signs the image, not encrypts.

Use case:

NOTE to seccom: Probably should describe how this works for all lifecycle steps.

Recommendation: ONAP should provide a reference implementation of a secrets vault service as an ONAP project.

Next Steps:

- **Find a project lead for a reference implementation.**

Component 2: A process to provision ONAP instances with credentials. These credentials may be used for interprocess communication (e.g., APPC calling A&AI) or for ONAP configuring VNFs.

Automatic provisioning of certificates and credentials to ONAP components: AAF can provision certificates. ECOMP DCAE is currently using AAF to provision certificates.

Next steps:

- **Work with the AAF team to include this functionality in Release 2. It is important to understand that the AAF solution depends on the CA supporting the SCEP protocol.**
- **Enhance AAF to provision userIDs & passwords to ONAP instances and VNFs. Most VNFs only support userID/password authentication today. ETSI NFV SEC may issue a spec in the future on a more comprehensive approach to using PKI for NFV which can be visited by ONAP SEC when released. Steve is working on this right now but doesn't know when he'll be done.**

2.6 Recommended approach

2.7 Implications to the ONAP

Describe what this means to ONAP

QUESTIONS:

3 ONAP Static Code Scans

Status: Recommended, and recommendation approved by TSC on 11/2/2018

3.1 ONAP Static Code Scanning

The purpose of the ONAP static code scanning is perform static code scans of the code as it is introduced into the ONAP repositories looking for vulnerabilities.

3.2 Approaches

Tools that have been assessed: Coverity Scan (LF evaluation), HP Fortify (AT&T evaluation), Checkmarx (AT&T evaluation), Bandit (AT&T evaluation)

Prelimary Decision: Coverity Scan https://scan.coverity.com/

Motivation: Coverity Scan is a service by which Synopsys provides the results of analysis on open source coding projects to open source code developers that have registered their products with Coverity Scan. Coverity Scan is powered by Coverity® Quality Advisor. Coverity Quality Advisor surfaces defects identified by the Coverity Static Analysis Verification Engine (Coverity SAVE®). Synopsys offers the results of the analysis completed by Coverity Quality Advisor on registered projects at no charge to registered open source developers. Coverity is integrated into OPNFV and other Open Source projects and operating successfully. The Linux Foundation recommends the use of the tool.

Current Activity: In conversations with Coverity to understand the definition of "project" – does it refer to ONAP or the projects under an ONAP release to ensure that the limitation on free scans does not lead to bottlenecks in submissions and commits.

Open Source use: 4000+ open source projects use Coverity Scan

Frequency of builds:

Up to 28 builds per week, with a maximum of 4 builds per day, for projects with fewer than 100K lines of code

Up to 21 builds per week, with a maximum of 3 builds per day, for projects with 100K to 500K lines of code

Up to 14 builds per week, with a maximum of 2 build per day, for projects with 500K to 1 million lines of code

Up to 7 builds per week, with a maximum of 1 build per day, for projects with more than 1 million lines of code

Once a project reaches the maximum builds per week, additional build requests will be rejected. You will be able to re-submit the build request the following week.

Languages supported: C/C++, C#, Java, Javascript, Python, Ruby

The scanning process can be triggered from Jenkins. OPNFV is currently using a basic gerrit plug in for some basic scans.

Question: What about Go? which versions of Python.

Comment: Add some motivation of why Coverity is a good idea.

Comment: We need to catch the commitment now.

Bring in a few prposals to the TSC.

3.3 ONAP process for static code scans

Two approaches are identified.

1. **Scan analysis in project**

   The PTL is informed of the scan analysis results on a regular basis (e.g. weekly).

   - **Project has the responsibility to analysis the scans and make required changes.**

   Note:
   •The work scales with projects
   •Security competence may not be in projects to understand the results
   •Have to work through the false positives.
   •Requires that the scan process is incorporated into Jenkins.

2 Create a support team to support the scan analysis with the projects.

•Under the guidance of the security sub-committee 1-2 team is created with project members (rally around timezones).
•Perform walkthrough of the static code scan results before MS-4.

In Either case, propose that MS-4 and Release criteria includes static code scan analysis.

3.4 Example of a CoverityScan report

The following report was generated by running Coverity against code from the Zephyr project.

https://wiki.onap.org/download/attachments/11928162/CoverityScanReportForZephyr.docx?api=v2

3.5 Recommendation

- **Use Coverity Scan https://scan.coverity.com/ to perform static code scans on all ONAP code.**
- **Automate scanning by enabling Jenkins to trigger weekly scans with Coverity Scan.**
- **Deliver scan reports to the PTLs for each project PTLs will be responsible for getting the vulnerabilities resolved (fixed or designated as false positive).**
- **All projects in a release must have the high vulnerabilities resolved by MS-3.**
- **All projects in a release must have the high and medium vulnerabilities resolved by MS-4.**
- **The Security Committee will host session to help projects walk through the scanning process and reports.**

4/11 Update

- The LF ONAP helpdesk is creating a Jenkins job to scan each repo with CoverityScan, following the job used by OpenDaylight (ticket #54456)
- The goal is to run one scan daily and have the results integrated into the Sonar dashboard
- A few projects use the Go and Closure languages which are not supported by CoverityScan
- Coverity scanning will not be implemented until after Beijing goes live

4. CII Badging process Learnings for ONAP.

Status: Draft

4.1 CII Badging process intro

This section captures the learning's of using the CII badging program in ONAP.

4.2 Learnings

The CLAMP project has been working as the CII badging certification. Their feedback is found here: CII Badging Program - Feedback. This is repeated below for simplicity:

4.2.1 CII Badging program introduction.

• Core Infrastructure Initiative Website:
-https://bestpractices.coreinfrastructure.org/

• Evaluate how projects follow best practices using voluntary self-certification

• Three levels: Passing, Silver and Gold

- **LF target level recommendation is Gold**

• ONAP Pilot Project: CLAMP
-https://bestpractices.coreinfrastructure.org/projects/1197

4.2.2 The Questionnaire

• Edition is limited to a subset of users

- **Main editor can nominate other users as editors**

• Divided into clear sections
- For each section, a set of questions is provided, addressing best practices relating to the parent section

• Each question asks if a criterion is

- **Met, unmet, not applicable, or unknown**

• Criteria are generally high-level as targeted to best practices, e.g.

- **"The project MUST have one or more mechanisms for discussion"**
- **"The project SHOULD provide documentation in English"**

4.2.3 The Goals

• Give confidence in the project being delivered

- **By quickly knowing what the project supports**

• See what should be improved

- **Self-questioning helps project stakeholders identifying strengths and weaknesses, do's and don'ts**

• Align all projects using the same ratings

- **Makes projects connected together to follow the same practices**

• Call for continuous improvement

- **Increase self rating and reach better software quality**

4.2.4 Raised Questions

- **Introduce test coverage rules: how many tests should be added for each code changes**
- **Digital signature: use digital signature in delivered packages (already in the plan?)**
- **Vulnerability fixing SLA: vulnerabilities should be fixed within 60 days**
- **Security mechanisms**
  - **Which cryptographic algorithms to use to encrypt password**
  - **The security mechanisms within the software produced by the project SHOULD implement perfect forward secrecy for key agreement protocols so a session key**

derived from a set of long-term keys cannot be compromised if one of the long-term keys is compromised in the future.
    - If the software produced by the project causes the storing of passwords for authentication of external users, the passwords MUST be stored as iterated hashes with a per-user salt by using a key stretching (iterated) algorithm (e.g., PBKDF2, Bcrypt or Scrypt).
    - The security mechanisms within the software produced by the project MUST generate all cryptographic keys and nonces using a cryptographically secure random number generator, and MUST NOT do so using generators that are cryptographically insecure

5 ONAP Communication Security

Status: Draft

5.1 ONAP Communication Security

Assuming the credential management is in place, ONAP needs to have a common means to support secure communication between the onap components.

There are two high level use cases to cover.

1. **Real-time communication between ONAP components**
2. **Support for authentication and encryption of the modules and packages to be onboarded into SDK (from VNF SDK).**

5.2 ONAP communication security requirements

To guide the solution development for the ONAP communication security, the following requirements are identified:

For: Real-time communication between ONAP components:

- **The solution MUST support an approach that can be common to all onap modules.**
- **The solution MUST support the credential management solution and MUST NOT be tied to any particular credential management scheme.**
- **The solution MUST support secure communication between the ONAP components in the following sense:**
    - **A receiving ONAP component understands that the message is authentic**
    - **Any element in between the ONAP components cannot interpret or change the message.**
- **The solution MUST enable that a sending ONAP component does not rely on what the receiving ONAP component is, and the receiving ONAP component does not rely on what the sending ONAP component is. (This would put unnecessary restraints on the architecture).**
- **The solution SHOULD be easy for the ONAP components to Adopt.**
- **The solution MUST be independent of the underlying communication technology (i.e. communication buss technologies).**

For models and packages to be onboarded:

- **The solution MUST support the credential management solution and MUST NOT be tied to any particular credential management scheme.**
- **The soluction MUST allow Service Design and Creation to validate the package from a security perspective.**

6. ONAP known vulnerability management

Status: Draft

Background:

Sonatype Nexus can provide a number of reports. One report it can provide is identification of components with known vulnerabilities.

Policies can be provisioned for different types of vulnerabilities to identify them as critical, severe, moderate, etc.

A process is required to support this. A project with a component that has a known vulnerability can do one of two things. 1. It can upgrade the component to a component version that does not have the vulnerability. Alternatively, the project can investigate the vulnerability to and conclude that it doesn't effect the project due to the way it uses the component or the part of the component is uses.

A process is required to support this.

Next Steps

Decide approach with projects:

Recommended to have MS-4 criterial as not to include modules with known vulnerabilities > 60 days old. MS-4 and Release Criteria.

Topic	Status	Comment
7. Pluggable Security	In Review	Moved to own page for manageability

8 VNF Package Security

Status: Draft

8.1 Introduction

The scope of this item is verification of the integrity and authenticity of the

- VNF package
- The artifacts in the VNF package

In general, the intention is to align with ETSI NFV specifications on this area, see the references in 8.5. The published version of [ETSI NFV SOL004] is the main specification to follow. Going forward, [ETSI NFV SEC021] shall also be considered (as of Feb 2018 the work has started).

8.2 Use Cases

8.2.1 Priority 1: VNF Package Verification

Integrity of the VNF package needs to be verified prior to, or at the time of onboarding. The purpose is to ensure that the VNF package originates from the vendor, and that the content has not been tampered with. The verification is done against the signature provided by the vendor. Reference [ETSI NFV SOL004] contains the detailed specifications.

8.2.2 Priority 2: Integrity Verification at Instantiation

At instantiation, the integrity of VNF image and related files shall be verified. The options are:

A) Verify against the signature provided by the vendor. [ETSI NFV SOL004] specifies "*The VNF provider may optionally digitally sign some artifacts individually*".

B) Verify against the signature created by the service provider. [ETSI NFV SOL004] specifies "*If software images or other artifacts are not signed by the VNF provider, the service provider has the option, after having validated the VNF Package, to sign them before distributing the different package components to different function blocks or the NFVI*".

8.2.3 Priority 3: Service Provider Ability to Sign the Artifacts

If the vendor did not sign artifacts (inside the VNF package) individually, service provider may want to sign those. Also, if the service provider needs to modify or add any artifacts, the service provider may want to sign those.

8.3 ONAP Impacts

Tentatively, following projects are impacted:

- VNF SDK: need to interpret the VNF Package according to [ETSI NFV SOL004]
- SDC, APP-C, VF-C: need to interpret the manifest file according to [ETSI NFV SOL004]
- VNF Requirements

AAF should not be impacted, because it already supports install of trusted certificates.

8.4 Certificate Assumptions

On the CA issuing the VNF package signing certificate, [ETSI NFV SOL004] specifies: "*This solution, either option 1 or option 2, relies on the existence in the NFVO of a root certificate of a trusted CA that shall have been delivered via a trusted channel that preserves its integrity (separate from the VNF package) to the NFVO and be pre-installed in the NFVO before the on-boarding of the VNF package.*
NOTE: The present document makes no assumption on who this trusted CA is. Furthermore, it does not exclude that the root certificate be issued by the VNF vendor or by the NFVI provider."

If the signing certificate has been issued by vendor's own CA, the related root CA has to be installed in ONAP AAF as trusted certificate.

8.5 References

[ETSI NFV SOL004]
ETSI GS NFV-SOL 004 V2.3.1 (2017-07): http://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/004/02.03.01_60/gs_nfv-sol004v020301p.pdf

[ETSI NFV SEC021]
ETSI NFV SEC021 work item description: https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=53601

10 (tmp) input to the S3P (carrier grade) discussions from a security perspective

Status: Draft

Note: This will be removed when the feedback is sent back.

The full list of the needs can be found at: https://wiki.onap.org/plugins/servlet/mobile?contentId=1015829#content/view/15998867

Security:

Per project:

- Level 0: None
- Level 1: CII Passing badge
- Level 2: CII Silver badge, plus:
  - All internal/external system communications shall be able to be encrypted.
  - All internal/external service calls shall have common role-based access control and authorization.
- Level 3: CII Gold badge

Note: When creating the CII project entry, it is recommended to use ONAP in the title to facilitate searching the onap projects.

Per Release:

- Level 1 70% of the projects included in the release at passing badge level
  - with non-passing projects reaching 80% towards passing level.
  - Non passing projects MUST pass these specific criteria:
    - The software produced by the project MUST use, by default, only cryptographic protocols and algorithms that are publicly published and reviewed by experts (if cryptographic protocols and algorithms are used).
    - If the software produced by the project is an application or library, and its primary purpose is not to implement cryptography, then it SHOULD only call on software specifically designed to implement cryptographic functions; it SHOULD NOT re-implement its own.
    - The security mechanisms within the software produced by the project MUST use default keylengths that at least meet the NIST minimum requirements through the year 2030 (as stated in 2012). It MUST be possible to configure the software so that smaller keylengths are completely disabled.
    - The default security mechanisms within the software produced by the project MUST NOT depend on broken cryptographic algorithms (e.g., MD4, MD5, single DES, RC4, Dual_EC_DRBG) or use cipher modes that are inappropriate to the context (e.g., ECB mode is almost never appropriate because it reveals identical blocks within the ciphertext as demonstrated by the ECB penguin, and CTR mode is often inappropriate because it does not perform authentication and causes duplicates if the input state is repeated).
    - The default security mechanisms within the software produced by the project SHOULD NOT depend on cryptographic algorithms or modes with known serious weaknesses (e.g., the SHA-1 cryptographic hash algorithm or the CBC mode in SSH).
    - If the software produced by the project causes the storing of passwords for authentication of external users, the passwords MUST be stored as iterated hashes with a per-user salt by using a key stretching (iterated) algorithm (e.g., PBKDF2, Bcrypt or Scrypt).

- Level 2 70% of the projects in the release passing silver
    - with non-silver projects completed passing level and 80% towards silver level
- Level 3 70% of the projects included in the release passing gold
    - with non-gold projects achieving silver level and achieving 80% towards gold level
- Level 4: 100% of the projects in the release passing gold level.

Examples of uses cases that people may want to see solved.

5. Examples of secure communication between ONAP components

6. Examples of security communiation between ONAP and other components.

7. User provisioning, and relation to access to other systems.