 O-RAN.WG11.AppLCM-Security-TR.0-R003-v03.00

Technical Report

O-RAN Work Group 11 (Security Work Group)

Study on Security for Application Lifecycle Management

Copyright © 2024 by the O-RAN ALLIANCE e.V.

The copying or incorporation into any other work of part or all of the material available in this document in any form without the prior written permission of O-RAN ALLIANCE e.V. is prohibited, save that you may print or download extracts of the material of this document for your personal use, or copy the material of this document for the purpose of sending to individual third parties for their information provided that you acknowledge O-RAN ALLIANCE as the source of the material and that you inform the third party that these conditions apply to them and that they must comply with them.

O-RAN ALLIANCE e.V., Buschkauler Weg 27, 53347 Alfter, Germany

Contents

List of figures 4

List of tables 4

Foreword 5

Modal verbs terminology 5

Introduction 5

1 Scope 6

2 References 6

2.1 Informative references 6

3 Definition of terms, symbols and abbreviations 8

3.1 Terms 8

3.2 Symbols 8

3.3 Abbreviations 8

4 Application Lifecycle Management background and assets 10

4.1 Introduction 10

4.2 Application Lifecycle 10

4.3 Assets 11

5 Key issues 12

5.1 Introduction 12

5.2 Development of Application Package 12

5.2.1 Key Issue Dev-1: Package Authenticity and Integrity 12

5.2.2 Key Issue Dev-2: Software Bill of Materials (SBOM) for Applications 12

5.2.3 Key Issue Dev-3: Security Descriptor 13

5.3 Onboarding and Deployment of Application 13

5.3.1 Key Issue Onboard-1: Initial Authentication of Application Package 13

5.3.2 Key Issue Onboard-2: Image Tampering during Application Onboarding 14

5.3.3 Key Issue Onboard-3: Risk of Deploying a Malicious Application 14

5.3.4 Key Issue Onboard-4: rAppID Abuse 15

5.3.5 Key Issue Onboard-5: Registration of Applications 17

5.4 Operation and Maintenance of Application 17

5.4.1 Key Issue Op-1: Security for Application Updates 17

5.4.2 Key Issue Op-2: Application Integrity during Operation 18

5.4.3 Key Issue Op-3: Security Logging and Monitoring of Applications 19

5.4.4 Key Issue Op-4: Secure Decommissioning of Applications 19

5.4.5 Key Issue Op-5: Exploitation of Application Vulnerabilities 20

5.4.6 Key Issue Op-6: PNF Software Lifecycle Security 20

5.4.7 Key Issue Op-7: Application Misconfiguration 21

5.4.8 Key Issue Op-8: Secure Deletion of Sensitive Data 21

6 Potential Solutions 22

6.1 Solution #1: Application Package Authenticity and Integrity 22

6.1.1 Introduction 22

6.1.2 Solution details 22

6.2 Solution #2: Secure Update of Applications 23

6.2.1 Introduction 23

6.2.2 Solution details 23

6.3 Solution #3 Application Security Descriptor 24

6.3.1 Introduction 24

6.3.2 Solution details 25

6.4 Solution #4: Secure Logging and Monitoring for Applications 29

6.4.1 Introduction 29

6.4.2 Solution details 29

6.5 Solution #5: Software Bill of Materials Requirement for Applications 30

6.5.1 Introduction 30

6.5.2 Solution details 30

6.6 Solution #6: Recommendations for Vulnerability Scanning 30

6.6.1 Introduction 30

6.6.2 Solution details 30

6.7 Solution #7: Recommendations for Addressing Application Vulnerabilities 31

6.7.1 Introduction 31

6.7.2 Solution details 31

6.8 Solution #8: PNF Software Lifecycle Security Recommendations 32

6.8.1 Introduction 32

6.8.2 Solution details 32

6.9 Solution #9: Application Integrity during Operation 34

6.9.1 Introduction 34

6.9.2 Solution details 34

6.10 Solution #10: Secure Decommissioning of Applications 35

6.10.1 Introduction 35

6.10.2 Solution details 35

6.11 Solution #11: Secure Deletion of Sensitive Data 35

6.11.1 Introduction 35

6.11.2 Solution details 35

6.12 Solution #12: rAppID Uniqueness 36

6.12.1 Introduction 36

6.12.2 Solution Details 37

6.13 Solution #13: Secure Application Registration Procedure 37

6.13.1 Introduction 37

6.13.2 Solution Details 37

7 Conclusions 39

7.1 Key Issue Solution Mapping 39

7.2 Recommendations 40

History 41

# List of figures

Figure 4.1‑1: SDLC Model [4] 10

Figure 4.2‑1: Application Lifecycle Phases [4] 11

Figure 6.1‑1 Application package authenticity and integrity protection 23

Figure 6.2‑1 Application update authenticity and integrity protection 24

Figure 6.3‑1: Application Package High Level Model from WG10 OAM [4] 25

Figure 6.3‑2: Application interconnection 25

Figure 6.13‑1: Registration process for rApps 37

Figure 6.13‑2: Registration process for xApps 38

# List of tables

[Table 6.3‑1: Security group rules descriptor 26](#_Toc149743729)

[Table 6.3‑2: SAL requirements descriptor 28](#_Toc149743730)

[Table 6.4‑1: Application LCM Logging Event Coverage 29](#_Toc149743731)

[Table 6.8‑1: Recommended security mechanisms 32](#_Toc149743732)

[Table 7.1‑1 - Mapping of Key Issues to Solutions 39](#_Toc149743733)

# Foreword

This Technical Report (TR) has been produced by the O-RAN Alliance.

# 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 O-RAN Drafting Rules (Verbal forms for the expression of provisions).

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

# Introduction

This technical report focuses on the security of O-RAN applications during their lifecycle phases as defined in WG10 OAM Architecture [4]: development, onboarding, and operation. O-RAN applications in the scope of this technical report include xApps, rApps, VNFs/CNFs such as O-CU, O-DU, O-RU, and Near-RT RIC, as well as software for physical network functions such as O-RU and O-DU.

The technical report takes into consideration the threats identified in the O-RAN Threat Modeling and Risk Assessment TR [1] and develops a list of key issues from this. The key issues relating to application life cycle management are divided into the appropriate lifecycle phase and include a description, security threats, and potential requirements to mitigate these threats. Solutions are proposed to address the key issues. The conclusion clause contains a mapping between key issues and solutions, as well as additional recommendations.

The order of the clauses of this document is as follows:

1. Scope
2. References
3. Definition of terms, symbols, and abbreviations
4. Application LCM background and assets
5. Key Issues
6. Potential Solutions
7. Conclusions

# Scope

The contents of the present document are subject to continuing work within O-RAN and may change following formal O-RAN approval. Should the O-RAN Alliance modify the contents of the present document, it will be re-released by O-RAN with an identifying change of version date and an increase in version number as follows:

version xx.yy.zz

where:

xx: the first digit-group is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have xx=01). Always 2 digits with leading zero if needed.

yy: the second digit-group is incremented when editorial only changes have been incorporated in the document. Always 2 digits with leading zero if needed.

zz: the third digit-group included only in working versions of the document indicating incremental changes during the editing process. External versions never include the third digit-group. Always 2 digits with leading zero if needed.

This technical report is the study of security of applications throughout the lifecycle management phases (development, onboarding, and operation)

Taking into consideration the threats and risks identified in O-RAN Threat Modeling and Risk Assessment TR [1], this study analyzes a list of key issues for applications during their lifecycle in order to derive security requirements, intended to be part of O-RAN Security Requirements and Controls Specifications [2].

# References

## 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, O-RAN 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.

[1] O-RAN ALLIANCE TR, “O-RAN Security Threat Modeling and Risk Assessment”

[2] O-RAN ALLIANCE TS, “O-RAN Security Requirements and Controls Specifications”

[3] O-RAN ALLIANCE TS, “O-RAN Architecture Description”

[4] O-RAN ALLIANCE TS, “O-RAN OAM Architecture”

[5] O-RAN ALIANCE TS, “O-RAN Information Model and Data Models”

[6] “The Minimum Elements for a Software Bill of Materials (SBOM), Pursuant to Executive Order 14028 on Improving the Nation’s Cybersecurity”, U.S. DoC and NTIA, July 2021.

[7] The MITRE Corporation, “MITRE ATT&CK® Containers Matrix”, Accessed on 03 AUG 2022, https://attack.mitre.org/matrices/enterprise/containers/

[8] The MITRE Corporation, “MITRE FiGHT™ Matrix”, Accessed on 16 AUG 2022, <https://fight.mitre.org/>

[9] O-RAN ALLIANCE TS, “O-RAN R1 Interface: Use Cases and Requirements”

[10] O-RAN ALLIANCE TS, “O-RAN Near-RT RIC Architecture”

[11] ETSI GS NFV-IFA 011 - VNF Descriptor and Packaging Specification,

<https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/011/04.02.01_60/gs_NFV-IFA011v040201p.pdf>

[12] IANA Protocol Registries, <https://www.iana.org/protocols>

[13] The "[Fossies](https://fossies.org/)" Software Archive – VNFD TOSCA template

<https://fossies.org/linux/tacker/samples/vnf_packages/Definitions/etsi_nfv_sol001_vnfd_types.yaml>

[14] O-RAN ALLIANCE TR, “O-RAN Study on Security Log Management”

[15] NISTIR 8397, Guidelines on Minimum Standards for Developer Verification of Software, 2021. https://doi.org/10.6028/NIST.IR.8397

[16] NIST SP 800-53r5, Security and Privacy Controls for Information Systems and Organizations, 2020. https://doi.org/10.6028/NIST.SP.800-53r5

[17] Enduring Security Framework (ESF) Securing the Software Supply Chain, Recommended Practices Guide for Developers, 2022

[18] OpenSSF FLOSS Best Practices Criteria, <https://bestpractices.coreinfrastructure.org/en/criteria/0>

[19] OpenSSF Concise Guide for Developing More Secure Software, <https://github.com/ossf/wg-best-practices-os-developers/blob/main/docs/Concise-Guide-for-Developing-More-Secure-Software.md#readme>

[20] Enduring Security Framework (ESF) Securing the Software Supply Chain, Recommended Practices Guide for Suppliers, 2022

[21] Center for Internet Security (CIS) Critical Security Controls, <https://www.cisecurity.org/controls/cis-controls-list>

[22] O-RAN ALLIANCE TS, “O-RAN Management Plane Specification”

[23] O-RAN ALLIANCE TS, “O-RAN Operations and Maintenance Interface Specification”

[24] Cybersecurity & Infrastructure Security Agency (CISA) Known Exploited Vulnerabilities Catalog, <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>

[25] O-RAN ALLIANCE TS, “Cloud Architecture and Deployment Scenarios for O-RAN Virtualized RAN”

[26] ETSI GS NFV-SEC 021, VNF Package Security Specification, <https://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/021/02.06.01_60/gs_nfv-sec021v020601p.pdf>

[27] O-RAN ALLIANCE TS, “O-RAN Security Test Specifications”

[28] “Six Steps Toward Better Application Decommissioning”, <https://www.zag-hq.com/2018/10/09/six-steps-toward-better-application-decommissioning/>

[29] Enduring Security Framework (ESF) Securing the Software Supply Chain, Recommended Practices Guide for Customers, 2022

[30] NIST SP 800-88 R1, Guidelines for Media Sanitization, 2014. https://doi.org/10.6028/NIST.SP.800-88r1

[31] ETSI GS NFV-SEC 013, NFV Security; Security Management and Monitoring specification

[32] RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace. <https://https://datatracker.ietf.org/doc/html/rfc4122>

[33] O-RAN ALLIANCE TR: “Study on Security for Near Real Time RIC and xApps”

[34] O-RAN ALLIANCE TS, “O-RAN Non-RT RIC Architecture”

# Definition of terms, symbols and abbreviations

## Terms

For the purposes of the present document, the terms and definitions provided in O-RAN Architecture Description [3] and O-RAN Operations and Maintenance Architecture [4] apply:

**Application Package**: The basic unit exchanged between the Solution Provider and Service Provider.

**Near-RT RIC:** O-RAN Near-Real-Time RAN Intelligent Controller: A logical function that enables near-real-time control and optimization of RAN elements and resources via fine-grained data collection and actions over E2 interface. It may include AI/ML (Artificial Intelligence / Machine Learning) workflow including model training, inference and updates.

**Non-RT RIC:** O-RAN Non-Real-Time RAN Intelligent Controller: A logical function within SMO that drives the content carried across the A1 interface. It is comprised of the Non-RT RIC Framework and the Non-RT RIC Applications (rApps) whose functions are defined below.

**Non-RT RIC Applications (rApps):** Modular applications that leverage the functionality exposed via the Non-RT RIC Framework’s R1 interface to provide added value services relative to RAN operation, such as driving the A1 interface, recommending values and actions that may be subsequently applied over the O1/O2 interface and generating “enrichment information” for the use of other rApps. The rApp functionality within the Non-RT RIC enables non-real-time control and optimization of RAN elements and resources and policy-based guidance to the applications/features in Near-RT RIC.

**O-Cloud:** O-Cloud is a cloud computing platform comprising a collection of physical infrastructure nodes that meet O-RAN requirements to host the relevant O-RAN functions (such as Near-RT RIC, O-CU-CP, O-CU-UP, and O-DU), the supporting software components (such as Operating System, Virtual Machine Monitor, Container Runtime, etc.) and the appropriate management and orchestration functions.

**rApp instance:** An individual occurrence of an application running in the Non-RT RIC runtime environment. [34]

**rApp instance identifier:** A unique identifier for each rApp instance, assigned by the SMO/Non-RT RIC framework during rApp registration. [34]

**Service Planning:** Activities internal to the Service Provider

**Service Provider**: A network provider who is planning to deploy applications into their network.

**Solution Provider**: An application developer who delivers applications to Service Providers.

**SP Exchange:** Not a formal interface in terms of between systems. However, the structure and content of the exchange is defined such that a Solution Provider can deliver applications to a Service Provider for deployment.

**xApps:**  Applications designed to run on the Near-RT RIC. Such an application is likely to consist of one or more microservices and at the point of on-boarding will identify which data it consumes and which data it provides. The application is independent of the Near-RT RIC and may be provided by any third party. The E2 enables a direct association between the xApp and the RAN functionality.

## Symbols

For the purposes of the present document, the following symbols apply:

## Abbreviations

For the purposes of the present document, the following abbreviations apply:

AI/ML Artificial Intelligence/Machine Learning

API Application Programming Interface

AppPackage Application Package

CNF Cloud-native Network Function

eNB eNodeB (applies to LTE)

gNB gNodeB (applies to NR)

KPI Key Performance Indicator

LCM Lifecycle Management

ML Machine Learning

MNO Mobile Network Operator

NF Network Function

NFV Network Function Virtualisation

O-CU O-RAN Central Unit as defined by O-RAN ALLIANCE

O-CU-CP O-CU Control Plane

O-CU-UP O-CU User Plane

O-DU O-RAN Distributed Unit (uses Lower-level Split)

O-RU O-RAN Radio Unit

RAN Radio Access Network

rAppId rApp Instance Identifier

RIC O-RAN RAN Intelligent Controller

SBOM Software Bill of Materials

SDLC Software Development Lifecycle

SMOService Management and Orchestration

VM Virtual machine

VNF Virtualised Network Function

# Application Lifecycle Management background and assets

## Introduction

Lifecycle Management of applications deployed in O-RAN follows standard Software Development Lifecycle (SDLC) basic models. Figure 4.1-1 depicts the model that applications described in this document generally follow. The states in this SDLC model can be generally classified into three categories: Development, Onboarding, and Operation. Two entities involved in Lifecycle Management are the Solution Provider and the Service Provider. A Solution Provider provides the application that fulfils the requirements and needs of a Service Provider, who deploys the application on the platform. The basic unit exchanged between the Solution Provider and Service Provider is the Application package (AppPackage). The AppPackage contains data including the application itself, as well as its descriptors. For more details on the AppPackage, please refer to [4].

Chart, pie chart

Description automatically generated

Figure 4.1‑1: SDLC Model [4]

## Application Lifecycle

The three application lifecycle phases defined in [4] are Development, Onboarding, and Operation. Development involves the creation of the application, preparation, and delivery from Solution Provider to Service Provider. Onboarding consists of the onboarding of the application onto its ultimate runtime environment. Some activities included in this phase are establishment of configurations, content verification, and app/AppPackage cataloguing. Operation encompasses the runtime of the application. In this phase, applications are deployed, updated, monitored, managed and ultimately after some analysis, either terminated or healed/scaled.

Diagram

Description automatically generated

Figure 4.2‑1: Application Lifecycle Phases [4]

## Assets

The assets within scope of this study are Application Packages for virtualized network functions (i.e., O-CU, O-DU, O-RU\*, and Near-RT RIC), xApps, rApps, and software packages for physical network functions (i.e., O-RU and O-DU). This aligns with WG10 Application Lifecycle Management [4].

\* O-RU is considered a physical device in current O-RAN architecture [3] but may be virtualized in future deployment scenarios [25].

The application packages include the following assets already defined in [1]:

* Data assets:
  + ASSET-D-15: Application software package
  + ASSET-D-23: Patches for vulnerable SW components
  + Secure stores:
    - ASSET-D-16: X.509 certificates
    - ASSET-D-17: Private keys
    - ASSET-D-32: Cryptographic keys used during secure boot
  + AI/ML Data:
    - ASSET-D-25: Training or test data
    - ASSET-D-26: The trained ML model
    - ASSET-D-27: The ML prediction results built into the model
    - ASSET-D-28: The behaviour of the ML system
* Components:
  + ASSET-C-02: Near-RT RIC software
  + ASSET-C-03: O-CU-CP software
  + ASSET-C-04: O-CU-UP software
  + ASSET-C-05: O-DU software
  + ASSET-C-06: O-RU software
  + ASSET-C-09: xApps
  + ASSET-C-10: rApps
  + ASSET-C-11: Non-RT RIC software

Software for PNFs is included in this study to the extent of defining high-level security requirements for the software package. Management for PNFs, to include PNF registration and management, is out of scope for this study. PNF registration and management is an important topic and may require additional security analysis when the 3GPP specifications are adopted by O-RAN [5].

NOTE: ‘Application’ in this document refers to xApps, rApps, O-CU, etc. and could be implemented as a VNF or CNF. Software for PNFs is not included in this definition and will still be referred to as “PNF” or “PNF software package”.

# Key issues

## Introduction

This clause details the key issues identified for security aspects related to application lifecycle management, subdivided into the LCM areas of 1) Development of Application Package, 2) Onboarding and Deployment of Application, and 3) Operation and Maintenance of Application. Each key issue details the general description of the key issue, defines the threats related to the issue, and proposes requirements that resolve or mitigate the key issue.

## Development of Application Package

### Key Issue Dev-1: Package Authenticity and Integrity

#### General description

The Solution Provider is responsible for creating an application that fulfils the needs of a Service Provider. An Application package delivered to the Service Provider that goes unverified in terms of authenticity and integrity is subject to multiple threats. For example, the Service Provider may unintentionally deploy an application that is either not actually provided by the Solution Provider, or that is from the Solution Provider but its integrity was compromised during transit. Such an occurrence may disrupt Service Provider operation and/or present security risks.

#### Security threats

The security threats associated with Package Authenticity and Integrity are:

* Malicious actor replacing the Service Provider’s package with their own package
* Service disruption caused by deploying an application with degraded functionality caused by compromise of Application package integrity
* Insider threat

The following threat(s) from [1] are applicable to Dev-1:

* T-O-RAN-09: An attacker compromises O-RAN components integrity and availability
* T-IMG-01: VM/Container images tampering
* T-NEAR-RT-02: Risk of deployment of a malicious xApp on Near-RT RIC
* T-rApp-05: An attacker deploys and exploits malicious rApp

#### Potential security requirements

The following security requirements from [2] involving Application software package protection are proposed to mitigate the threats stemming from the key issue of package authenticity and integrity.

* REQ-SEC-ALM-PKG-2
* REQ-SEC-ALM-PKG-3
* REQ-SEC-ALM-PKG-4
* REQ-SEC-ALM-PKG-5

### Key Issue Dev-2: Software Bill of Materials (SBOM) for Applications

#### General description

Software bill of materials for O-RAN software products is an important part of the software development process and applies also to the application life cycle process. SBOMs document the software used in software products in order to understand the software supply chain for proper notification of vulnerabilities and updates.

#### Security threats

The security threats that arise from not properly documenting software products via an SBOM includes:

* Lack of process to detect known software vulnerabilities and lack of accountability for notification

The following threat(s) from [1] are applicable to Dev-2:

* T-GEN-01: Software flaw attack
* T-OPENSRC-01: Developers use SW components with known vulnerabilities and untrusted libraries that can be exploited by an attack through a backdoor attack

#### Potential security requirements

Clause 6 of the WG11 Security Requirements and Controls Specifications [2] provides SBOM guidelines and requirements for O-RAN, based on guidance from NTIA [6]. This covers requirements on the minimum data fields and depth for SBOMs for O-RAN software.

An additional potential requirement could explicitly state that application packages also need SBOMs from the application provider with the same fields and to the same depth as other O-RAN software products are specified.

### Key Issue Dev-3: Security Descriptor

#### General description

WG10 has proposed a SecurityDescriptor in the Application Package, to be reviewed by WG11 to ensure all security aspects are being considered and it is in line with WG11 requirements.

#### Security threats

The following threat(s) are applicable to Dev-3:

* Missing or improperly defined elements of the SecurityDescriptor, such as information regarding the application certificate and security rules, could lead to unauthorized access to data and services.

#### Potential security requirements

Proper and comprehensive definition of the SecurityDescriptor helps ensure elements of security needed for the Application package are present.

The SecurityDescriptor is part of the Application package and therefore it should be protected in terms of authenticity and integrity.

## Onboarding and Deployment of Application

### Key Issue Onboard-1: Initial Authentication of Application Package

#### General description

After the Application package has been delivered from the Solution Provider to the Service Provider, preparation for application deployment commences where it goes through the Onboarding phase prior to deployment. If the application is unauthenticated during onboarding, Service Providers can unintentionally deploy an application that can cause harm to their platforms. Depending on the platform, the unauthenticated application could be also used by a threat actor to perform malicious activities on the Near-RT RIC, Non-RT RIC, and O-Cloud platforms.

#### Security threats

The security threats associated with Initial Authentication of Application package are:

* Unauthenticated application onboarded in the O-Cloud Image Repository
* Unauthenticated application running on the Service Provider/SMO platform

The following threat(s) from [1] are applicable to Onboard-1:

* T-O-RAN-06: An attacker exploits insufficient/improper mechanisms for authentication and authorization to compromise O-RAN components
* T-O-RAN-08: An attacker compromises O-RAN data integrity, confidentiality and traceability
* T-O-RAN-09: An attacker compromises O-RAN components integrity and availability
* T-IMG-01: VM/Container images tampering
* T-NEAR-RT-02: Risk of deployment of a malicious xApp on Near-RT RIC
* T-xApp-01: An attacker exploits xApp vulnerabilities and misconfigurations
* T-rApp-05: An attacker deploys and exploits malicious rApp

#### Potential security requirements

The following security requirements from [2] involving Application software package protection are proposed to mitigate the threats stemming from the key issue of initial authentication of application/VNF.

* REQ-SEC-ALM-PKG-2
* REQ-SEC-ALM-PKG-3
* REQ-SEC-ALM-PKG-4
* REQ-SEC-ALM-PKG-5

### Key Issue Onboard-2: Image Tampering during Application Onboarding

#### General description

An attacker can inject malicious code or tamper with the information in the Application package during onboarding. After instantiation of the application, the tampered code can cause DoS, information stealing, etc.

#### Security threats

The following threat(s) from [1] are applicable to Onboard-2:

* T-O-RAN-09: An attacker compromises O-RAN components integrity and availability
* T-IMG-01: VM/Container images tampering

#### Potential security requirements

The following security requirements from [2] involving Application software package protection are proposed to mitigate the threats stemming from the key issue of image tampering during onboarding.

* REQ-SEC-ALM-PKG-7a, 7b, 7c
* REQ-SEC-ALM-PKG-12
* REQ-SEC-ALM-PKG-13

### Key Issue Onboard-3: Risk of Deploying a Malicious Application

#### General description

Deploying a malicious application can negatively affect the O-RAN platform and its functions. Malicious xApps can impact radio service and provide unauthorized access to subscriber information. Malicious rApps may disrupt offered network service and affect Non-RT RIC functions. Malicious code in VM/Container images can cause DoS, information stealing, etc., when the VNF/CNF is instantiated.

#### Security threats

The following threat(s) from [1] are applicable to Onboard-3:

* T-rApp-05: An attacker deploys and exploits malicious rApp
* T-NEAR-RT-01: Malicious xApps can exploit UE identification, track UE location and change UE priority
* T-IMG-01: VM/Container images tampering

#### Potential security requirements

The following security requirements from [2] involving Application software package protection are proposed to help mitigate the threats stemming from the key issue of the risk of deploying a malicious application. Properly verifying authenticity and integrity of the Application package and testing the package for known vulnerabilities help address the threats from deploying a malicious application.

* REQ-SEC-ALM-PKG-2
* REQ-SEC-ALM-PKG-3
* REQ-SEC-ALM-PKG-4
* REQ-SEC-ALM-PKG-5
* REQ-SEC-ALM-PKG-7a, 7b, 7c

### Key Issue Onboard-4: rAppID Abuse

#### General Description

The rAppID is assigned to an rApp instance during rApp registration, which occurs after the onboarding and instantiation of the rApp as described in [9] and shown in Figure 5.3-1.

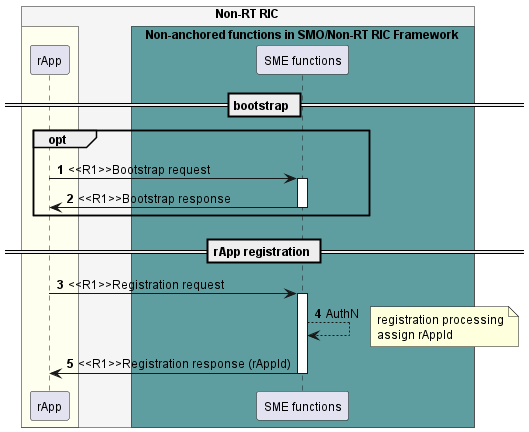


Figure 5.3-1: Assignment of rAppID during rApp registration process [9]

The rAppID is used to identify the rApp to the Service Management and Exposure (SME) or Data Management and Exposure (DME) services during use cases such as service registration, service discovery, data type discovery, and subscribing for service availability. Figures 5.3-2 and 5.3-3 depict examples of R1 use cases from [9] where the rAppID is sent to the SME or DME respectively to identify the rApp making the request. The rAppID may be one of the mechanisms used for authorization of an rApp for certain use cases.

Treemap chart

Description automatically generated

Figure 5.3-2: Register service use case flow diagram (rAppID to SME) [9]

Diagram

Description automatically generated

Figure 5.3-3: Discover data type use case flow diagram (rAppID to DME) [9]

An adversary may obtain a legitimate rAppID and use it to gain unauthorized access to services and data. Accidental or intentional use of non-unique identifiers for the rAppID can cause operational problems. Duplicate rAppIDs may cause services to be unavailable to the rApps requesting the service and be provided to rApps not needing the service (in the case of multiple rApps having the same rAppID).

NOTE: Security for IDs for xApps has also been studied in the Near-RT RIC and xApp security technical report [33], with similar threats, requirements, and solution noted. The rAppID Abuse key issue and corresponding solution (clause 6.12) considers existing xApp ID security work.

#### Security Threats

The security threats associated with rAppID abuse are:

* rAppID collisions
* Unauthorized access to R1 services and data

#### Potential Security Requirements

The Non-RT RIC should create each rAppID using procedures that avoid duplicates. One example of this is UUID as described in IETF RFC 4122.

### Key Issue Onboard-5: Registration of Applications

#### General Description

During deployment of xApps and rApps, the registration procedure occurs where the applications are registered to the Near-RT RIC platform or Non-RT RIC/SMO Framework respectively. During both registration procedures, the application is authenticated before it is assigned an ID. For more information about the application registration processes, refer to WG2 R1 Use Cases [9] for rApps and WG3 RICARCH [10] for xApps. Application package signatures are checked before registration during onboarding and deployment, but there can be a delay between onboarding and registration of the application which necessitates a validation during the registration phase. The location where the application is instantiated may also be different from the location of onboarding, so application validation at registration may be able detect potential modifications.

#### Security Threats

The following threat(s) from [1] are applicable to Onboard-5:

* T-O-RAN-06: An attacker exploits insufficient/improper mechanisms for authentication and authorization to compromise O-RAN components
* T-O-RAN-09: An attacker compromises O-RAN components integrity and availability
* T-NEAR-RT-02: Risk of deployment of a malicious xApp on Near-RT RIC
* T-rApp-05: An attacker deploys and exploits malicious rApp

#### Potential security requirements

The following security requirement is proposed to mitigate the threats stemming from the key issue of application registration:

* Application package should be authenticity and integrity checked during registration.

## Operation and Maintenance of Application

### Key Issue Op-1: Security for Application Updates

#### General description

According to OAM architecture [4], the update process is an orchestrated process of deploy and terminate, and it is possible for two versions of an application to be active at the same time. Examples of situations where an update may be made available are when new features are introduced for the application or when the update addresses a vulnerability. In any situation where the application is updated, it must be ensured that the update is delivered, onboarded, and deployed in a secure manner involving authentication and integrity checks.

The timeline for an update will remain a Service Provider implementation decision, although Service Providers should prioritize security updates and apply critical ones in a timely manner.

#### Security threats

The security threats associated with Security for Application Updates are:

* Malicious actor replacing the Service Provider’s package with their own package
* Service disruption caused by deploying an application with degraded functionality caused by compromise of Application package integrity
* Insider threat

The following threats from [1] are applicable to Op-1:

* T-O-RAN-01: An attacker exploits insecure designs or lack of adoption in O-RAN components
* T-O-RAN-06: An attacker exploits insufficient/improper mechanisms for authentication and authorization to compromise O-RAN components
* T-O-RAN-08: An attacker compromises O-RAN data integrity, confidentiality and traceability
* T-O-RAN-09: An attacker compromises O-RAN components integrity and availability
* T-IMG-01: VM/Container images tampering
* T-NEAR-RT-02: Risk of deployment of a malicious xApp on Near-RT RIC
* T-xApp-01: An attacker exploits xApp vulnerabilities and misconfigurations
* T-rApp-05: An attacker deploys and exploits malicious rApp

#### Potential security requirements

NOTE: The security requirements from [2] involving authentication of application by Solutions Providers and Service Providers is proposed to mitigate the key issue of onboarding a malicious application. These same requirements should be applied to address security threats of onboarding an update to an existing application. This is due to updates to applications being treated in the same manner as an application’s initial installation onto its respective platform in terms of delivery, onboarding, and deployment.

* REQ-SEC-ALM-PKG-2
* REQ-SEC-ALM-PKG-3
* REQ-SEC-ALM-PKG-4
* REQ-SEC-ALM-PKG-5
* REQ-SEC-ALM-PKG-7a, 7b, 7c

### Key Issue Op-2: Application Integrity during Operation

#### General description

The operation and maintenance phase of the application represents an opportunity for malicious actors to negatively affect the O-RAN platform. While an application is running on its respective platform, it may be susceptible to malicious activities such as the overwriting of the application. Vulnerabilities in the application could also be exploited to execute malicious behavior. Verifying the integrity of running applications could ensure no modifications were made to introduce malicious behavior into the applications.

#### Security threats

Examples of security threats associated with xApp integrity during operation are:

* Vulnerabilities in trusted xApps (T-xApp-01)
* Security misconfiguration (T-xApp-01)
* Compromise of xApp isolation (T-xApp-03)

Examples of the security threats associated with rApp integrity during operation are:

* Exploitation of services across R1 interface (T-rAPP-07)
* Security misconfiguration (T-rAPP-03)

Examples of the security threats associated with VNF/CNF integrity during operation are:

* VM/Container escape (T-VM-C-02)
* VM/Container data theft (T-VM-C-03)
* Changing virtualization resources without authorization (T-VM-C-05)

Other threats during runtime operation include:

* Buffer overflow (T-O-RAN-01)

#### Potential security requirements

* Integrity of running applications should be periodically verified by comparing the cryptographic hash of the application with a known good hash.

### Key Issue Op-3: Security Logging and Monitoring of Applications

#### General description

Security logging and monitoring for applications needs to be properly defined so that threats during the application lifecycle management process can be detected.

#### Security threats

The following security threats are an example of threats that could be detected by logging a comprehensive set of security events:

* Adversaries may exploit software vulnerabilities of the application in an attempt to elevate privileges [7, T1068]
* Adversaries may implant application images with malicious code to establish persistence after gaining access to an environment [7, T1525].
* Adversaries may deploy fake VNFs to collect information from the network [8, FGT5013]

The following threats from [1] are applicable to Op-3:

* T-O-RAN-07: An attacker compromises O-RAN monitoring mechanisms and log files integrity and availability

#### Potential security requirements

To detect these threats, the following application life cycle events are recommended for security logging. This list of events may be used to develop security analytics that have the potential to detect anomalous behavior. The specific development of analytics and the definition of anomalous behavior depend on the operational environment:

* Cryptographic hash of application package
* Onboarding of application packages
* Instantiation of application packages
* Creation of app IDs (for x/rApps)
* Changes to application configuration
* Updates to applications (REQ-SEC-ALM-PKG-14)
* Creation, modification, or deletion of identities; and transactions executed by users in applications.
* Application resource usage (disk space, CPU load)
* Decommissioning of applications
* Application API calls

### Key Issue Op-4: Secure Decommissioning of Applications

#### General description

The decommissioning process for applications is an important part of the lifecycle management. During application operation, an application may perform operations on data, use resources and access secrets. The proper termination of an application can prevent excessive resource usage, accidental deletion of pertinent data (application data, cryptographic keys, etc.), and misallocation to a malicious application by a malicious actor. The removal of trust relationships associated with the application is critical in order to prevent accidental exposure to attacker.

#### Security threats

The following threat(s) are applicable to Op-4:

* T-VM-C-06: Failed or incomplete VNF/CNF termination or releasing of resources

In addition, failure to revoke or remove credentials or other trust artifacts associated with the application could leave an exposure for an attacker to gain access. [29]

#### Potential security requirements

The following prospective requirements may help mitigate key issue Op-4:

* Identification of the replacement application prior to decommissioning of an application.
* Proper documentation of decommissioning.
* Preserving worthwhile legacy data related to the decommissioned application.
* Adherence to organization’s legal guidelines.
* Deactivation of credentials
* Deactivation of permissions, accesses and rights
* Removal of all rules/roles/groups for application in existing security infrastructure
* Removal of application from allowlists
* Adding the application to the denylist in security and network monitoring tools
* Removal of relevant configuration data [31]

### Key Issue Op-5: Exploitation of Application Vulnerabilities

#### General description

Vulnerabilities in an application can be exploited by a malicious actor to disrupt network service, degrade network performance, extract sensitive information, alter data, etc.

#### Security threats

The following threats from [1] are applicable to Op-5:

* T-O-RAN-01: An attacker exploits insecure designs or lack of adoption in O-RAN components
* T-xApp-01: An attacker exploits xApp vulnerabilities and misconfigurations
* T-rApp-02: An attacker exploits rApp vulnerabilities

#### Potential security requirements

The following security requirements from [2] involving testing, discovery, and identification of vulnerabilities are proposed to assist in the mitigation of this key issue involving the exploitation of application vulnerabilities.

* REQ-SEC-ALM-PKG-1
* REQ-SEC-SYS-1

### Key Issue Op-6: PNF Software Lifecycle Security

#### General description

PNFs may include software packages that will also require lifecycle management events, including updates. Updates to PNF software could be used to inject malicious code to compromise the PNF in order to launch reverse attacks and other attacks against VNFs/CNFs. In addition, failure to implement updates that affect security of the system may result in exploitation of the vulnerabilities addressed in the updates to gain access into the system.

This work item may propose general abstract security requirements for PNF lifecycle management; however, the PNF vendor would ultimately be responsible for PNF software management and its security.

#### Security threats

The following threats from [1] are applicable to Op-6:

* T-PNF-01: An attacker compromises a PNF to launch reverse attacks and other attacks against VNFs/CNFs

#### Potential security requirements

Although general security recommendations may be proposed for PNF Software in this report, vendors are ultimately responsible for the security of their products.

* PNF Software packages should be protected in terms of integrity and authenticity.

### Key Issue Op-7: Application Misconfiguration

#### General description

Security misconfigurations can exist in an application which can provide an attacker an exploit that they can use to disrupt network service, degrade network performance, extract sensitive information, alter data, etc. These misconfigurations could include improperly configured application permissions and accesses, unnecessary open ports, and unused protocols enabled.

#### Security threats

The following threats from [1] are applicable to Op-7:

* T-O-RAN-02: An attacker exploits misconfigured or poorly configured O-RAN components
* T-rApp-03: An attacker exploits rApps misconfiguration
* T-xApp-01: An attacker exploits xApps vulnerabilities and misconfiguration

#### Potential security requirements

REQ-SEC-ALM-PKG-1 from [2] may help in mitigating application misconfiguration, and additional security controls on configuration validation and maintaining up-to-date applications may be recommended.

### Key Issue Op-8: Secure Deletion of Sensitive Data

#### General description

If data from a terminated application or any application that has released resources are not erased securely, a malicious application can access that data.

#### Security threats

The following threat(s) are applicable to Op-8:

* T-VM-C-06: Failed or incomplete VNF/CNF termination or releasing of resources

#### Potential security requirements

The following prospective requirements may help mitigate key issue Op-8:

* Availability of methods for secure deletion of data.
* Availability of methods for overwriting data with various sensitivity levels.

# Potential Solutions

## Solution #1: Application Package Authenticity and Integrity

### Introduction

The WG11 Security Requirements and Controls Specifications [2] contains a set of requirements and security control(s) pertaining to authentication and integrity-checking of the Application package that help ensure the package has not been tampered with or damaged in transit and that the package does indeed come from the Solution Provider. The set of requirements and security controls from [2] provide a baseline level of security for Application packages.

ETSI GS NFV-SEC 021 [26] can provide further information about the application package signing/verification procedure during onboarding and instantiation described in the Solution details below.

### Solution details

Figure 6.1‑1 provides an overview of requirements from [2] that help provide integrity and authentication to the Application package throughout its lifecycle.

During the development phase:

* The Application Provider signs the Application package which minimally includes the image, signing certificate, and Application Provider’s signature(s).
* NOTE: Before the onboarding of application, the SMO should contain a pre-installed Root CA certificate for verifying the application integrity and authenticity.

During the onboarding phase:

* After the Application package is sent to the Service Provider, the Service Provider verifies the Application package artifacts and re-signs the package before it is onboarded to the catalogue.
  + Signing of the Application package by the Service Provider prior to its onboarding is required for the following reasons:
    - For checking the Application Provider signature during the application instantiation, it is not needed that the O-Cloud Platform contact the Application Provider outside of the Service Provider's security domain.
    - It gives the Service Provider the opportunity and means to authorize Application packages for deployment on its network (e.g., avoid an Application package intended for one Service Provider with a valid Application Provider certificate being loaded by an attacker into another Service Provider’s network).
    - Furthermore, some Service Providers might wish to undertake additional security validation of the Application package during the onboarding process and Service Provider signing could be used to certify the application as authorized to onboard into the Service Provider’s network.

During the operation phase:

* During application instantiation, the Application package is authenticated and verified by the O-Cloud Platform using signatures from Service Provider first and then the Application Provider.

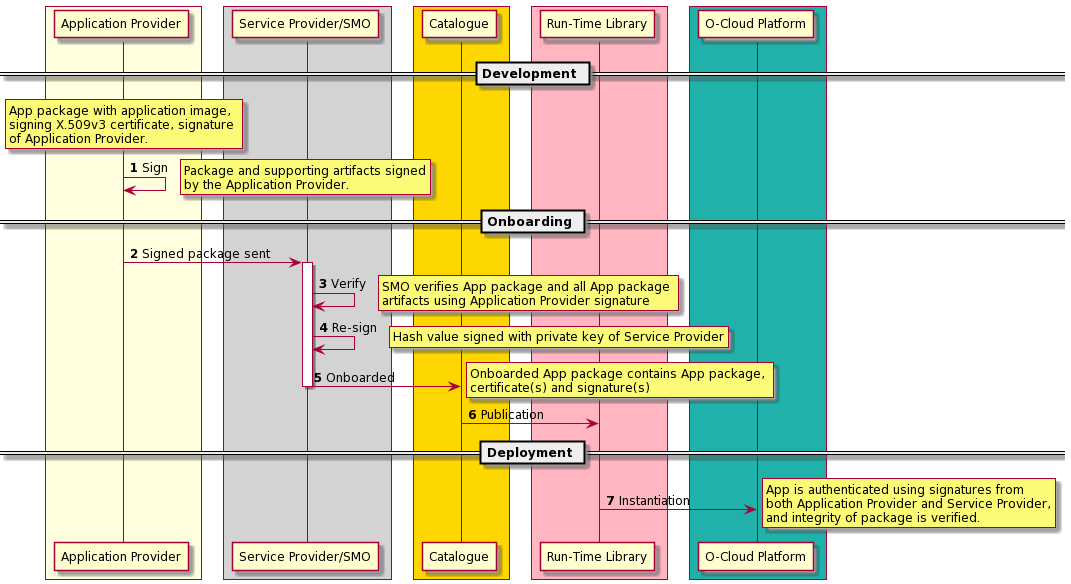


Figure 6.1‑1 Application package authenticity and integrity protection

## Solution #2: Secure Update of Applications

### Introduction

During operation, an application may need to update to a newer version to modify its features or address security. As stated in Solution #1, Application package authenticity and integrity, existing requirements in [2] relating to integrity and authentication of an Application package help provide a baseline level of application security that can be also applied to their updates.

### Solution details

According to WG10 OAM Architecture [4], the update process consists of terminating the older version of the application and deploying the newer version of the application, as applications are atomic. This solution proposes that the deployment of the newest version of the application will follow the standard process of the onboarding of any application, which is subject to authenticity and integrity controls. Figure 6.2‑1 below is tailored to application updates and depicts the set of requirements from [2] that are proposed to help secure the updating of applications in O-RAN.

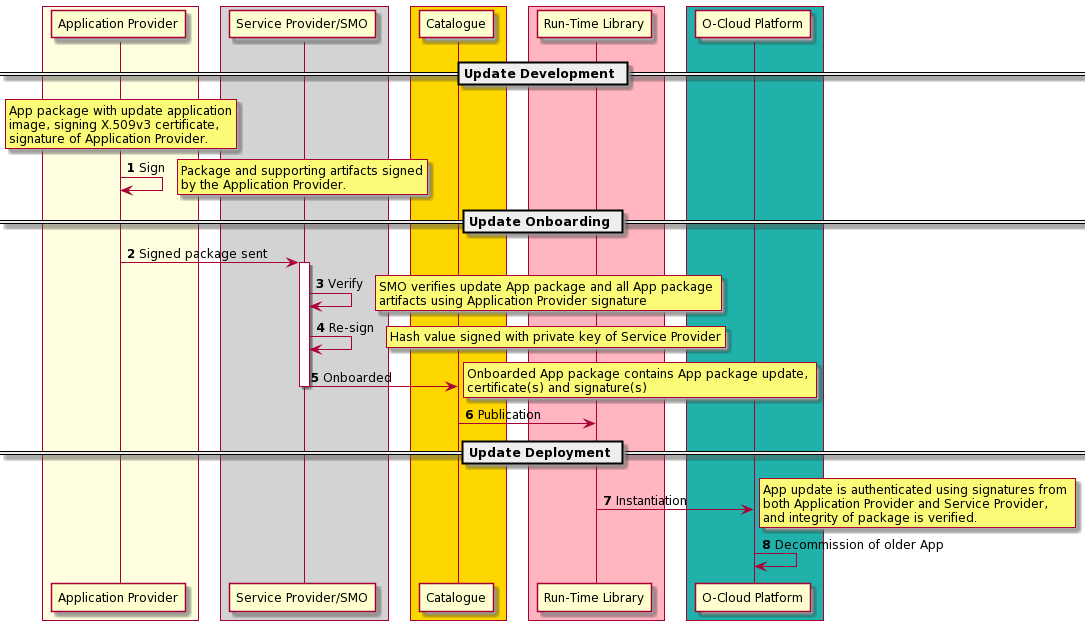


Figure 6.2‑1 Application update authenticity and integrity protection

## Solution #3 Application Security Descriptor

### Introduction

O-RAN WG10 describes a high-level composition of the Application Package which includes the security descriptor focusing on privileges, roles, users, and certificates as illustrated in the following figure. In order to completely define the characteristics of a security descriptor of an O-RAN application, this clause proposes an extension to the preliminary O-RAN WG10 security descriptor by capturing and analysing additional security attributes applicable to O-RAN application. These security attributes should be packaged with the application image, forming an Application package.

It is expected that information in this clause may face changes and adjustments to accommodate unforeseen needs that may arise from O-RAN WG10.

Diagram

Description automatically generated

Figure 6.3‑1: Application Package High Level Model from WG10 OAM [4]

The following figure shows a basic representation of the structure of an application. An application is composed of VMs/CNs that are interconnected. In the definition of this interconnection, the expected security policies for the traffic and communication need to be provided. Further, an application exposes services to external applications or users, in the form of APIs or dashboards. An application should support the definition of Service Availability Level (SAL) with requirements on scaling and migration over the set of metrics exposed by an application for ensuring availability and continuity of the provided services.

Graphical user interface, application

Description automatically generated

Figure 6.3‑2: Application interconnection

### Solution details

ETSI GS NFV-IFA 011 [11] specifies the following requirements applicable to application descriptor attributes:

* **VNF\_PACK.META.025**: The VNFD shall support a description of the security rules to filter the ingress/egress packets related to the VNF.
  + The filtering rules include, but are not limited to, the packet direction, TCP/UDP port range, IP protocol, etc.
* **VNF\_PACK.META.026**: The VNFD shall support associating the security rules to the relevant VNF connection points.
* **VNF\_PACK.META.017**: The VNFD shall support a description of Service Availability Level (SAL) requirements for virtual resources on the underlying NFVI.

For fulfilling the ETSI security requirement on application descriptors, the following two categories of attributes are proposed:

1. Security group rules descriptor: used by the O-Cloud platform to enforce network security between Apps/VNFs by disallowing sniffing of network traffic.
2. Service Availability Level (SAL) requirements descriptor: used by the SMO to monitor and guarantee the compliance of Apps/VNFs and O-Cloud with the defined SALs in the descriptor. Such monitoring activities include the monitoring of the status of an App/VNF, resource provisioning, and detecting resource overloading.

#### Security group rules descriptor

**Related ETSI requirement**: VNF\_PACK.META.025, VNF\_PACK.META.026

This security descriptor contains the security group rules. It is used by the O-Cloud platform for specifying the matching criteria for the ingress and/or egress traffic to/from visited connection points as defined in ETSI GS NFV-IFA 011 [11]. These rules aim to enforce security for applications at the Ethernet and IP layers by disallowing sniffing of network traffic. They cover the following elements:

* The direction of traffic in which the security group rule is applied. The direction of ‘ingress’ or ‘egress’ is specified against the associated CP/ECP, i.e., ‘ingress’ means the packets entering a CP/ECP (response), while ‘egress’ means the packets sent out of a CP/ECP (request).
* Protocol to filter the Ethernet layer. Permitted values include IPV4 and IPV6.
* Protocol to filter the IP layer. Permitted values include any protocol defined in the IANA protocol registry [12], e.g., TCP, UDP, ICMP, etc.
* Start of port range (inclusive). It indicates minimum port number in the range that is matched by the security group rule. If a value is provided at design-time, this value may be overridden at run-time based on other deployment requirements or constraints. It is required if protocol is ‘tcp’ or ‘udp’: Defaults to 0.
* End of port range (inclusive). It indicates maximum port number in the range that is matched by the security group rule. If a value is provided at design-time, this value may be overridden at run-time based on other deployment requirements or constraints. It is required if protocol is ‘tcp’ or ‘udp’: Defaults to 65535.

Each of the elements outlined here above are translated to security group rules attributes and summarized in this table.

Table 6.3‑1: Security group rules descriptor

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| Security\_group\_rule.id | Integer | id of the Security Group Rule |
| Security\_group\_rule.description | String | Description of Security Group Rule |
| Security\_group\_rule.direction | String | Direction of the Security Group Rule. Allowed values:   * Ingress * Egress |
| Security\_group\_rule.ether\_type | String | Type of the Security Group Rule. Allowed values:   * IPV4 * IPV6 |
| Security\_group\_rule.protocol | String | Protocol of the Security Group Rule. Allowed values:   * TCP * UDP * Any |
| Security\_group\_rule.port\_range\_min | Integer | Minimum port number of the range applicable to the Security Group Rule. |
| Security\_group\_rule.port\_range\_max | Integer | Maximum port number of the range applicable to the Security Group Rule. |

Here below is an example of an application descriptor template structure and its security group rules attributes based on TOSCA standards and written in YAML [13]. It should be on-boarded in the Application package.

**tosca.policies.nfv.SecurityGroupRule:**

**derived\_from:** tosca.policies.Root

**description:** The SecurityGroupRule type is a policy type specified the matching criteria for the ingress and/or egress traffic to/from visited connection points as defined in ETSI GS NFV-IFA 011 [11].

**properties:**

**description:**

**type:** string

**description:** Human readable description of the security group rule.

**required:** false

**direction:**

**type:** string

**description:** The direction in which the security group rule is applied. The direction of 'ingress' or 'egress' is specified against the associated CP. I.e., 'ingress' means the packets entering a CP, while 'egress' means the packets sent out of a CP.

**required:** false

**constraints:**

- **valid\_values:** [ ingress, egress ]

**default:** ingress

**ether\_type:**

**type:** string

**description:** Indicates the protocol carried over the Ethernet layer.

**required:** false

**constraints:**

- **valid\_values:** [ ipv4, ipv6 ]

**default:** ipv4

**protocol:**

**type:** string

**description:** Indicates the protocol carried over the IP layer. Permitted values include any protocol defined in the IANA protocol registry, e.g. TCP, UDP, ICMP, etc.

**required:** false

**constraints:**

- **valid\_values:** [ hopopt, icmp, igmp, ggp, ipv4, st, tcp, cbt, egp, igp, bbn\_rcc\_mon, nvp\_ii, pup, argus, emcon, xnet, chaos, udp, mux, dcn\_meas, hmp, prm, xns\_idp, trunk\_1, trunk\_2, leaf\_1, leaf\_2, rdp, irtp, iso\_tp4, netblt, mfe\_nsp, merit\_inp, dccp, 3pc, idpr, xtp, ddp, idpr\_cmtp, tp++, il, ipv6, sdrp, ipv6\_route, ipv6\_frag, idrp, rsvp, gre, dsr, bna, esp, ah, i\_nlsp, swipe, narp, mobile, tlsp, skip, ipv6\_icmp, ipv6\_no\_nxt, ipv6\_opts, cftp, sat\_expak, kryptolan, rvd, ippc, sat\_mon, visa, ipcv, cpnx, cphb, wsn, pvp, br\_sat\_mon, sun\_nd, wb\_mon, wb\_expak, iso\_ip, vmtp, secure\_vmtp, vines, ttp, iptm, nsfnet\_igp, dgp, tcf, eigrp, ospfigp, sprite\_rpc, larp, mtp, ax.25, ipip, micp, scc\_sp, etherip, encap, gmtp, ifmp, pnni, pim, aris, scps, qnx, a/n, ip\_comp, snp, compaq\_peer, ipx\_in\_ip, vrrp, pgm, l2tp, ddx, iatp, stp, srp, uti, smp, sm, ptp, isis, fire, crtp, crudp, sscopmce, iplt, sps, pipe, sctp, fc, rsvp\_e2e\_ignore, mobility, udp\_lite, mpls\_in\_ip, manet, hip, shim6, wesp, rohc ]

**default:** tcp

**port\_range\_min:**

**type:** integer

**description:** Indicates minimum port number in the range that is matched by the security group rule. If a value is provided at design-time, this value may be overridden at run-time based on other deployment requirements or constraints.

**required:** false

**constraints:**

- **greater\_or\_equal:** 0

- **less\_or\_equal:** 65535

**default:** 0

**port\_range\_max:**

**type:** integer

**description:** Indicates maximum port number in the range that is matched by the security group rule. If a value is provided at design-time, this value may be overridden at run-time based on other deployment requirements or constraints.

**required:** false

**constraints:**

- **greater\_or\_equal:** 0

- **less\_or\_equal:** 65535

**default:** 65535

**targets:** [ tosca.nodes.nfv.VduCp, tosca.nodes.nfv.VnfExtCp ]

#### Service Availability Level (SAL) requirements descriptor

Editor’s Note: It may be necessary to liaise very closely with WG6 and WG10 on the SAL descriptor to ensure that the proposed descriptor is in line with the descriptor under progress by WG6/WG10.

**Related ETSI requirement**: VNF\_PACK.META.017

An application should support the definition of SALs with requirements over the set of metrics exposed by the applications, which can be used by the SMO to trigger resource scaling or migration actions for avoiding resource overloading and application downtime. By performing these actions, the compliance with the defined SALs in the security descriptor can be guaranteed.

This security descriptor is used by the SMO for monitoring the status of an App/VNF to provide correct service availability and continuity. The monitoring should not only check if an application is “up and running”, but also it should provide resource provisioning and detect resource overloading. This requires two efforts from application providers. Firstly, they must expose a way to extract the real-time value of their SLI (Service Level Indicator). Secondly, they must instruct the SMO and O-Cloud on how to collect, analyse, and react whenever an SAL is being breached. While the first effort is about creating an HTTP endpoint, the second is to provide an application descriptor that describes: HTTP endpoints, expression to evaluate every SAL, and the instructions to follow when a SAL is being breached (e.g., migrate service, change the number of resources allocated, etc.).

The SAL requirements descriptor should cover the following elements:

* Support of SAL requirements over the set of metrics exposed by the VNFs, which can be used by SMO (NFO, FOCOM) and O-Cloud to trigger scaling or migration actions.
  + Required resources (CPU, Memory, Storage, ...): Namespace quotas, required (initial deployment) or Limit (runtime/scaling) Quota
* Monitoring the status of an application is crucial to provide correct service availability and continuity.
* Descriptor for SMO (NFO, FOCOM) on how to collect, analyse, and react whenever a SAL requirement is being breached. It should describe, among others:
  + Expression to evaluate every requirement, and
  + Instructions to follow when a requirement is being breached (e.g., migrate service, change the number of resources allocated, etc.).

The proposed SAL requirements descriptor is described in the following table:

Table 6.3‑2: SAL requirements descriptor

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| SAL | Object | Service Availability Level |
| SAL.name | String | Name of the SAL |
| SAL.metric | String | Reference to the metric when presented as already aggregated. Either the field metric or expression exist in an SAL object. |
| SAL.threshold | Integer | If the value of the SAL is GREATER than or LOWER than this value, it constitutes a violation of the SAL |
| SAL.threshold\_type | String | Type of the threshold. Allowed values are GT (Greater Than) or LT (Lower Than) |
| SAL.action | Object | Action type object describing the action to be taken if SAL is violated |
| SAL.check | Integer | Every number of seconds that this SAL will be checked, the SAL.threshold must be crossed in order to consider a violation of SAL |
| Action | Object | Action container object |
| Action.target\_ref | String | Reference to the target application on which to perform an action. |
| Action.step | String | Action to be executed every time the SAL is violated. Allowed values are:  • TRIGGER\_SCALE\_UP  • TRIGGER\_SCALE\_IN  • TRIGGER\_MIGRATION |

## Solution #4: Secure Logging and Monitoring for Applications

### Introduction

Key Issue Op-3 identified security threats that could be detected by logging application life cycle events and recommended a set of application life cycle events for security logging. These events were mapped to Key Issue SLC#1 Logging Security Related Activity and Events in the WG11 technical report “Study on Security Log Management” [14] in order to ensure coverage of these events. Within that technical report, Key Issue SLC#1 was updated to account for missing application life cycle logging events.

### Solution details

Table 6.4‑1 shows the mapping of specific application LCM events to the categories described in the updated Secure Log Management Key Issue SLC#1 to show coverage of the application events. The application events recommended to be logged are fully covered by the events listed in the Secure Log Management technical report [14] and are planned to be included in subsequent solutions and requirements from the Secure Log Management Work Item.

Table 6.4‑1: Application LCM Logging Event Coverage

|  |  |
| --- | --- |
| **AppLCMSec TR KI Op-3**  **Application LCM Event** | **Security Log Management TR KI SLC#1**  **Security Logging event** |
| Cryptographic hash of application package | 5d: Cryptographic operations |
| Onboarding of application packages | 2c: Image registry events, addition |
| Instantiation of application packages | 2a: Lifecycle events, service creation |
| Creation of app IDs (for x/rApps) | 3f: Creation of identities |
| Changes to application configuration | 2g: Change notifications |
| Updates to applications | 2f: Maintenance activity undertaken  2g: Change notifications |
| Creation, modification, or deletion of identities; and transactions executed by users in applications | 3f: Creation, modification, or deletion of identities; and transactions executed by users in applications. |
| Application resource usage (disk space, CPU load) | 1c: Resource usage |
| Decommissioning of applications | 2a: Lifecycle events, service deletion |
| Application API calls | 2d: API calls |

## Solution #5: Software Bill of Materials Requirement for Applications

### Introduction

The requirements in Clause 6 of the WG11 Security Requirements and Controls Specifications [2] for Software Bill of Materials for O-RAN were reviewed to ensure coverage for application packages.

### Solution details

It was determined that the initial requirement REQ-SBOM-001 covers O-RAN applications, and the subsequent requirements REQ-SBOM-002 through REQ-SBOM-011 cover the necessary fields, depth, format, and signature and encryption requirements for SBOMs for applications.

**REQ-SBOM-001**: The O-RAN vendor should provide the SBOM with every O-RAN software delivery, including patches, to the network operator.

This solution recommends changing REQ-SBOM-001 to be a mandatory requirement in the WG11 Security Requirements and Controls Specifications [2], instead of optional, to comply with current supply chain security best practices.

## Solution #6: Recommendations for Vulnerability Scanning

### Introduction

The goal of this solution is to provide application vendors with recommendations on scanning their software throughout the software development lifecycle for vulnerabilities. This scanning can take the form of static application security testing (SAST), software composition analysis (SCA), or dynamic application security testing (DAST).

Beyond the requirements listed below which are already in the O-RAN specifications, these recommendations will stay in the technical report as best practices. The details of the vulnerability scanning and specific tools are left up to vendor implementation and out of scope of normative O-RAN security specifications.

This solution is part of mitigating Key Issue Op-5: Exploitation of Application Vulnerabilities. Discovery and identification of application vulnerabilities will allow for vendors to mitigate those vulnerabilities before integrating the application into the O-RAN environment.

### Solution details

Performing vulnerability scanning is an important part of secure software development. NIST [15], [16] and ESF [17] recognize that early and frequent vulnerability scanning is a key aspect of software security assurance, and automation can allow this scanning to be performed more efficiently and repeatedly.

NISTIR 8397 [15] provides a guideline for minimum standard techniques for software developers to use to verify their software. These minimum recommendations for best practices should also be used for O-RAN application developers. Additional references, alternatives, and example tools are provided in [15] for more information.

NIST SP 800-53 [16] provides more details on application security testing, including Interactive Application Security Testing (IAST), Dynamic Application Security Testing (DAST), and Runtime Application Self-Protection (RASP) tools.

These general recommendations are comprised in REC-SEC-ALM-PKG-1 in the O-RAN Security Requirements and Controls Specifications [2]:

**REQ-SEC-ALM-PKG-1:** The Application package shall be certified by the Application Provider.

EXAMPLE: Software testing suites for certification include vulnerability scanning, static and dynamic testing, and penetration testing. Refer to Annex C clause C.2.1 for additional information.

Additionally, Annex C of [2] provides non-normative guidance on implementing this requirement during development, on-boarding, and runtime. It details the type of testing, responsible party for the testing, and rationale for the testing.

REQ-SEC-ALM-PKG-1 and Annex C of [2] show the requirement for O-RAN application vulnerability scanning in the specifications, and the additional industry resources and standards provided above give more details for application vendors to follow the industry best practices for vulnerability scanning during application development.

## Solution #7: Recommendations for Addressing Application Vulnerabilities

### Introduction

The goal of this solution is to provide application vendors with recommendations on addressing security vulnerabilities in their software.

Beyond any requirements listed below which are already in the O-RAN specifications, these recommendations will stay in this technical report as best practices. The details of the vulnerability fixes, including exact timeframe, are left up to vendor implementation and out of scope of O-RAN specifications.

This solution is part of mitigating Key Issue Op-5: Exploitation of Application Vulnerabilities. Timely vulnerability fixes for O-RAN applications will help prevent the exploitation of application vulnerabilities and therefore reduce overall risk for the O-RAN system.

### Solution details

O-RAN Security Requirements and Controls Specifications [2] includes REC-SEC-ALM-PKG-1 which is documented in Solution #6. It requires the Application provider to eliminate any flaws and defects found during testing before delivery to the Service Provider and then provide the testing results.

In addition to this general requirement, several resources external to O-RAN provide guidelines and best practices for addressing vulnerabilities in software or applications. These resources should be used by O-RAN application vendors to address vulnerabilities in their software to increase the security of the O-RAN system.

Linux Foundation’s OpenSSF [18], [19] provides guidelines and badging criteria that help ensure open-source software is developed following their best practices. This includes secure software development practices, bug and vulnerability reporting processes, automated testing, use of basic good cryptographic practices, fixing publicly known vulnerabilities, and code analysis.

ESF’s Software Supply Chain for Suppliers [20] highlights several of these practices as well, including vulnerability reporting and the subsequent response process, verification of third-party software, and code reviews for security and continuous software security improvement. Many of these same practices are included in NIST 800-53 [16] and NISTIR 8397 [15].

Center for Internet Security (CIS) Critical Security Controls list [21] includes Control 7 on Continuous Vulnerability Management and Control 16 on Application Software Security, with the focus on discovering vulnerabilities and resolving or remediating vulnerabilities on a monthly or more frequent basis in order to reduce risk.

To summarize these recommendations:

* Recommendations for vendor verification of third-party packages in their code:
  + Application providers should perform continuous verification of third-party binaries via binary scanning and software composition analysis.
  + Application providers should monitor known vulnerabilities in their application’s direct and indirect dependencies and keep dependencies reasonably up to date.
  + Application providers should verify third party applications are compliant with relevant security requirements.
* Recommendations for vendor fixes of security issues in their code:
  + Application providers should fix all publicly known vulnerabilities in a reasonable timeframe and fix all critical vulnerabilities in a timely manner after they are confirmed.
  + Application providers should have a vulnerability report process documented to notify the community of vulnerabilities in their application.

## Solution #8: PNF Software Lifecycle Security Recommendations

### Introduction

PNFs such as O-RUs and O-DUs may include software packages that will be downloaded, installed, activated, and updated. These PNF software packages have similar lifecycles events as outlined for applications in this technical report. It is important that security is addressed in these lifecycle events for the reasons mentioned in Key Issue Op-6: PNF Software Lifecycle Security.

This solution outlines general recommendations for security aspects during these software lifecycle events and maps to the security mechanisms and processes already in place in O-RAN specifications. It is beyond the scope of this work item and technical report to consider PNF registration and certificate management for PNFs. In addition, it is noted that PNF software format is vendor implementation specific and not standardized in O-RAN [23].

Additional information for security of O-RU software packages is listed in the WG4 M-Plane specification [22], while information for O-DU software package security is listed in WG10 O1 Interface specification [23].

### Solution details

Recommended security requirements and mechanisms are listed by PNF software lifecycle event in Table 6.8-1. These security mechanisms are derived from industry recommendations for strong security controls. Specifically, the CIS Critical Security Controls [21] provides information on access control management, protecting data and data access, and controlling configuration of software assets. NIST SP 800-53 [16] provides details on access control and authorization, malicious code protection, software integrity, failover capability, and component authenticity.

Table 6.8‑1: Recommended security mechanisms

|  |  |
| --- | --- |
| **Software Lifecycle Event** | **Recommended Security Mechanisms for PNF software package** |
| PNF Software Download & Install | * Integrity check * Authentication * Secure file transfer |
| PNF Software Activation | * Authorization |
| PNF Software Update | * Integrity check * Authentication * Secure transport * Authorization * Ability to restore to previous secure state if errors in update |

These security mechanisms are evaluated to help mitigate the threats listed in Key Issue Op-6. Additional vulnerability and SBOM analysis by the PNF software vendor is recommended to detect malware or potential vulnerabilities and protect against introduction of malicious code to the PNF.

Many of these security aspects are covered in technical specifications from WG4 and WG10, specifically Clause 8 Software Management from WG4 M-plane Specification [22] and Clause 2.8 PNF Software Management Services from WG10 O1 Interface Specification [23]. Currently, WG10 O1 specification covers O1 compliant nodes, which includes O-DU but not O-RU. WG4 M-plane specification covers O-RU. Below, software management security controls are summarized from these specifications per the lifecycle event. Specifics of the security mechanisms can be found at the referenced clause of the specifications.

Security mechanisms currently in O-RAN specifications:

* PNF Software Download & Install:
  + NOTE: Software Download for O-DU includes installation of the software [23]. For O-RU, software download and install are separate processes [22].

|  |  |  |
| --- | --- | --- |
|  | **O-RU** | **O-DU** |
| Integrity check | May be supported at download and during software install [22, Clause 8.5 & 8.6] | Must be supported on downloaded software [23, REQ-SWD-FUN-5]. |
| Authentication | Must be supported for software download [22, Clause 8.5] | Must be supported for software download [23, Clause 2.8.3] |
| Secure file transfer | Must be supported for software file download to the O-RU [22, Clause 8.5] | Must be supported for software package download [23, Clause 2.8.3] |

* PNF Software Activate:

|  |  |  |
| --- | --- | --- |
|  | **O-RU** | **O-DU** |
| Authorization | Authorization check is not supported [22, Clause 8.7] | Must be supported upon activation request [23, Clause 2.8.5, REQ-SWA-FUN-1] |

* PNF Software Update:
  + The update process outlined in the WG4 and WG10 specifications follows the same process of the lifecycle events above: inventory, download, install, and activate.
  + NOTE: A “software slot” is an independent storage location for a single software build. Only one software slot can be active at any time. [22]

|  |  |  |
| --- | --- | --- |
|  | **O-RU** | **O-DU** |
| Integrity check | May be supported at software download and during install [22, Clause 8.5 & 8.6] | Must be supported on downloaded software [23, REQ-SWD-FUN-5]. |
| Authentication | Must be supported for software download [22, Clause 8.5] | Must be supported for software download [23, Clause 2.8.3] |
| Secure file transfer | Must be supported for software file download to the O-RU [22, Clause 8.5] | Must be supported for software package download [23, Clause 2.8.3] |
| Authorization | Authorization check is not supported [22, Clause 8.7] | Must be supported upon activation request [23, Clause 2.8.5, REQ-SWA-FUN-1] |
| Ability to restore to previous secure state if errors in update | Supported [22, Clause 8.8] | Supported [23, Clause 2.8.5, REQ-SWA-FUN-8 and 9] |

Comparison of specifications to recommendations and additional recommendations:

As can be seen by the above mapping of recommended security mechanisms to the specifications, the current WG4 and WG10 specifications for the O-RU and O-DU offer support for many of the recommended security requirements and controls for PNF software. PNF software management requirements are expected to be specified in a 3GPP specification from SA5 Workgroup when complete [23], so this should be reviewed again at that time.

To further enhance the security posture, it is recommended that the PNF software package be authenticity and integrity protected. Currently, this is supported by WG10 specifications for the O-DU and optionally supported by WG4 specifications for the O-RU.

Additional recommendations to consider include authorization prior to software activation, including during update. Authorization would ensure that software is permitted to run on the PNFs.

## Solution #9: Application Integrity during Operation

### Introduction

The operation lifecycle phase of O-RAN applications presents an opportunity for actors to perform malicious activities. Key Issue Op-2 provides information about the applicable threats and potential security requirement recommendation(s) for mitigation. Detecting and preventing threats during application runtime is still an on-going research problem. Proposing security controls in the O-RAN Alliance specifications to prevent these threats presents some challenges, especially within the context of standards and normative specifications, and may not be feasible at this time.

Defining security for ensuring application integrity during operation may be implementation and architecture specific and may not necessarily apply to all O-RAN application deployments. Many exploits to application integrity rely on the underlying architecture and mitigations are system dependent**.** Application integrity monitoring during operation presents some challenges which are outlined in the subsequent paragraphs.

Monitoring application integrity during runtime requires that there is an entity that monitors the application and verifies its integrity. The monitoring entity itself needs to be protected from an adversary to ensure the monitor is not compromised by a malicious actor and that integrity verification results are unmodified.

Another area to explore is how exactly an untampered application should look like during operation in terms of memory data. Verifying the integrity of a running application requires knowledge of the known good states of the application and what is not. Changing application data within memory may not necessarily indicate a tampered application, especially considering AI/ML applications.

Another factor to consider is the performance impact of integrity monitoring for control loop execution times for xApps and rApps. The Near-RT RIC requires control loops from 10 milliseconds to 1s, and the Non-RT RIC control loops are specified for more than 1 second. These control loop execution times must be considered, especially for xApps, when factoring in potential negative effects on performance by monitoring for integrity on running O-RAN applications.

### Solution details

Although there may not be any feasible normative requirements proposed to help mitigate the threats during application runtime, recommendations and already existing requirements and security controls can help with this issue:

* Before application delivery to the service provider, security mechanisms such as scanning for vulnerabilities and dynamic and static testing of the application package may be performed for certification by the solution provider before software delivery (REC-SEC-ALM-PKG-1).
* During onboarding, the O-RAN application also undergoes a dual signature process, featuring signing of the application package by the solution provider and service provider, that verifies the authenticity and integrity of the application package (SEC-CTL-ALM-PKG-2).
* Every O-RAN application requires the SBOM with every O-RAN software delivery (REQ-SBOM-001). Scanning of the SBOM artifacts for comparison with the Common Vulnerabilities and Exposures (CVE) database (for example, in [24]) for known vulnerabilities may enable discovery of potential known exploits for code/libraries used in the application.

Although directly monitoring for application integrity during operation may not be feasible to specify, security measures in place already in the specifications help ensure that the application is tested for vulnerabilities, authenticity, and integrity before operation, and can lower the likelihood of exploits that affect application integrity during operation.

## Solution #10: Secure Decommissioning of Applications

### Introduction

As mentioned in the description clause, the decommissioning process for applications is an important part of the lifecycle management. As such, the related literature was reviewed and various approaches to the decommissioning were closely examined [28], [29], [31].

### Solution details

When dealing with secure decommissioning of applications, various organizations recommend some best practices. The essence of those is captured in the following requirements:

* A replacement application (or lack of it) should be known and available to avoid any service and performance gaps prior to decommissioning of an application.
* Any decommissioning of applications should be properly documented.
* Sometimes parts of legacy data are valuable because of historical reasons or because they are difficult to reproduce. As such, worthwhile legacy data related to the decommissioned application should be preserved.
* All trust relationships associated with the application should be removed or revoked.

A recommendation to keep in mind is to adhere to the organization’s legal guidelines for storing legacy data.

## Solution #11: Secure Deletion of Sensitive Data

### Introduction

NIST instead of deletion utilizes clear and purge terminologies [30]. Clear and purge are used for sensitive and highly sensitive data, respectively. Purged data cannot be recovered with the state of art laboratory techniques, while cleared data can potentially be recovered using state of art laboratory techniques.

Storage device technologies can be divided into magnetic and solid state devices. Magnetic Media consists of ATA hard disk drives and SCSI hard disk drives. Solid state media, on the other hand, consist of ATA solid state, SCSI solid state and NVM express devices. Some purge options are applicable at the hardware disposal time.

### Solution details

The recommended clear/purge procedures are listed below:

**ATA Hard Disk Drives**

* **Clear**: at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used.
* **Purge**
* Option 1: Issue cryptographic erasing command to cause all Media Encryption Keys (MEKs) to be changed. Then optionally use the block erase command (or clear procedure, if block erase is not supported) to block erase the media.
* Option 2: ATA sanitize commands
* Option 3: ATA SECURE ERASE UNIT commands
* Option 4: Degauss using an approved degausser

**SCSI Hard Disk Drives**

* **Clear:** at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used.
* **Purge**
* Option 1: Issue cryptographic erasing command to cause all MEKs to be changed. Then optionally use the block erase command (or clear procedure, if block erase is not supported) to block erase the media.
* Option 2: SCSI sanitize commands
* Option 3: Degauss using an approved degausser

Flash memory-based storage devices are made of ATA solid state devices, SCSI solid state devices and NVM express SSDs. The recommended clear/purge procedures for each technology are provided below:

**ATA Solid State Devices**

* **Clear**:
* Option 1: at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used.
* Option 2: SECURITY ERASE UNIT command
* **Purge**:
* Option 1: Issue cryptographic erasing command to cause all MEKs to be changed. Then optionally use the block erase command (or clear procedure, if block erase is not supported) to block erase the media.
* Option 2: ATA sanitize command

**SCSI Solid state Devices**

* **Clear**: at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used.
* **Purge**
* Option 1: Issue cryptographic erasing command to cause all MEKs to be changed. Then optionally use the block erase command (or clear procedure, if block erase is not supported) to block erase the media.
* Option2: SCSI SANITIZE

**NVM Express Devices**

* **Clear**: at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used.
* **Purge**
* Option 1: Issue cryptographic erasing command to cause all MEKs to be changed. Then optionally use the block erase command (or clear procedure, if block erase is not supported) to block erase the media.
* Option 2: NVM Express Format command

## Solution #12: rAppID Uniqueness

### Introduction

A unique identifier for an rApp instance within a Non-RT RIC runtime environment would help prevent the threat of collision from two rApps with the same ID. A unique ID assignment is also an important part of security logging to be able to appropriately identify applications in log files, and a unique ID can help prevent operational disruptions due to misassigning resources or services to the incorrect application.

The Near-RT RIC Security Technical Report [33] proposes UUID v4 as described in RFC 4122 [32] as the format for xApp IDs to ensure its uniqueness. UUID v4 provides generation details to ensure uniqueness to prevent the threat of collision. UUID v4 can also be considered for the format for rAppIDs of rApp instances to help ensure each rApp instance may have their own unique identification. It is recommended that WG2 review and specify rAppID format to ensure uniqueness.

### Solution Details

The following recommendations should be considered for inclusion in the O-RAN Security Requirements and Controls TS to mitigate the threats from non-unique application identifiers.

* rAppID should be unique within the Non-RT RIC runtime environment.
* rAppIDs should be generated using strong randomization methods to resist brute force attacks.

## Solution #13: Secure Application Registration Procedure

### Introduction

The registration procedure for xApps and rApps occurs after onboarding and instantiation of the application. In the registration procedure, the applications are registered to the Near-RT RIC platform or Non-RT RIC/SMO Framework respectively. In both cases, the application is authenticated before it is assigned an identifier. For more information about the application registration process, refer to WG2 R1 Use Cases [9] for rApps and WG3 RICARCH [10] for xApps. This solution proposes a common security process during xApp/rApp registration for authenticating the application, which is proposed to be defined in WG11 specifications.

### Solution Details

Figures 6.13-1 and 6.13-2 depict the application registration procedures for rApps and xApps, respectively, with the addition of the authenticity check per WG11 specifications. The application sends a registration request message to the respective entity (Near-RT RIC or Non-RT RIC) which processes the request. The Near-RT RIC or Non-RT RIC should authenticate the application by:

* Verifying Service Provider Signature
* Verifying integrity of application package
* Verifying certificates are not revoked

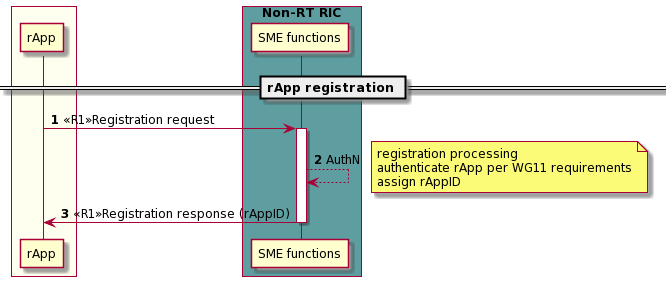


Figure 6.13‑1: Registration process for rApps

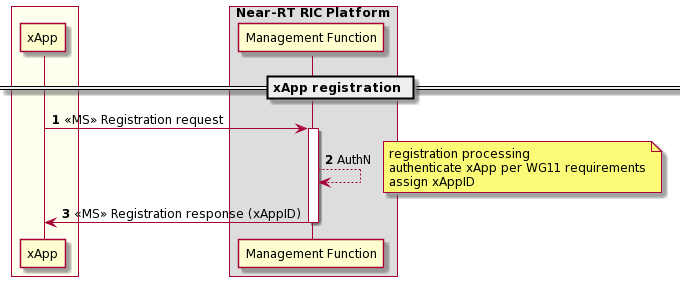


Figure 6.13‑2: Registration process for xApps

This solution defines a general procedure for authenticating applications during the registration process as specified by WG2 R1 Use Cases and Requirements [9] and WG3 RICARCH [10]. The authentication requirements should be defined in WG11 Security Requirements and Controls TS.

# Conclusions

## Key Issue Solution Mapping

This document currently contains 16 key issues across the application life cycle stages and 13 solutions proposals to address or reduce the risk associated with the key issues. The solutions are listed here for reference, followed by a mapping of the proposed solutions to the key issues in Table 7.1‑1.

Solution #1: Application Package Authenticity and Integrity

Solution #2: Secure Update of Applications

Solution #3: Application Security Descriptor

Solution #4: Secure Logging and Monitoring for Applications

Solution #5: Software Bill of Materials Requirement for Applications

Solution #6: Recommendations for Vulnerability Scanning

Solution #7: Recommendations for Addressing Application Vulnerabilities

Solution #8: PNF Software Lifecycle Security Recommendations

Solution #9: Application Integrity during Operation

Solution #10: Secure Decommissioning of Applications

Solution #11: Secure Deletion of Sensitive Data

Solution #12: rAppID Uniqueness

Solution #13: Secure Application Registration Procedure

Table 7.1‑1 - Mapping of Key Issues to Solutions

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Key issue*** | **O-RAN Security Req. TS** | **Sol. #1** | **Sol. #2** | **Sol. #3** | **Sol. #4** | **Sol. #5** | **Sol. #6** | **Sol. #7** | **Sol. #8** | **Sol. #9** | **Sol. #10** | **Sol. #11** | **Sol. #12** | **Sol. #13** |
| *Dev-1: Package Authenticity and Integrity* | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| *Dev-2: Software Bill of Materials (SBOM) for Applications* | X |  |  |  |  | X |  |  |  |  |  |  |  |  |
| *Dev-3: Security Descriptor* | X |  |  | X |  |  |  |  |  |  |  |  |  |  |
| *Onboard-1: Initial Authentication of Application Package* | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| *Onboard-2: Image Tampering during Application Onboarding* | X | X |  |  |  |  |  |  |  |  |  |  |  |  |
| *Onboard-3: Risk of Deploying a Malicious Application* | X | X |  |  |  |  | X |  |  |  |  |  |  |  |
| *Onboard-4: rApp ID Abuse* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| *Onboard-5: Registration of Applications* |  |  |  |  |  |  |  |  |  |  |  |  |  | X |
| *Op-1: Security for Application Updates* | X |  | X |  |  |  |  |  |  |  |  |  |  |  |
| *Op-2: Application Integrity during Operation* | X | X |  |  |  |  | X | X |  | X |  |  |  |  |
| *Op-3: Security Logging and Monitoring of Applications* | X |  |  |  | X |  |  |  |  |  |  |  |  |  |
| *Op-4: Secure Decommissioning of Applications* | X |  |  |  |  |  |  |  |  |  | X | X |  |  |
| *Op-5: Exploitation of Application Vulnerabilities* | X |  |  |  |  |  | X | X |  |  |  |  |  |  |
| *Op-6: PNF Software Lifecycle Security* |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
| *Op-7: Application Misconfiguration* | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *Op-8: Secure Deletion of Sensitive Data* | X |  |  |  |  |  |  |  |  |  |  | X |  |  |

## Recommendations

The following is a list of next steps and recommendations coming from this work item which will be addressed in future updates to this technical report and other O-RAN specifications.

**Recommendation 1:** This work item should define additional recommendations and requirements for secure decommissioning and deletion of sensitive data for applications. CRs will be generated.

**Recommendation 2:** This work item should continue contributing to the WG11 Security Requirements and Controls Specifications [2] with new requirements and security controls from the solutions outlined in this report. CRs will be generated.

**Recommendation 3:** This work item should contribute security test cases for relevant security requirements to the WG11 Security Test specifications [27]. CRs will be generated.

**Recommendation 4:** This work item should collaborate with other WG11 work items and other work groups on security verification during the application registration procedure for xApps and rApps.

**Recommendation 5:** WG11 should collaborate with WG6 and WG10 to ensure the WG11 security specifications are appropriately referenced by WG6 and WG10 in their specifications.

# History

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
| 2022.11.14 | v01.00 | First release, Version 1 of Technical Report |
| 2023.02.24 | v02.00 | Second release, Version 2 of Technical Report |
| 2023.11.03 | v03.00 | Third release, Version 3 of Technical Report |