O-RAN.WG11.Security-Log-Management-TR.0-R004-v05.00.03

Technical Report

O-RAN Work Group 11 (Security Work Group)

Study on Security Log 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

Register of Associations, Bonn VR 11238, VAT ID DE321720189

# Contents

Contents 2

List of figures 5

List of tables 5

Foreword 6

Modal verbs terminology 6

Introduction 6

1 Scope 6

2 References 6

2.1 Informative references 6

3 Definition of terms, symbols and abbreviations 9

3.1 Terms 9

3.2 Abbreviations 10

4 Security Log Management background, concepts and assumptions 12

4.1 Introduction 12

4.2 Definition of Event Sources 14

4.3 Security Log Data Lifecycle Management 15

5 Key Issues 15

5.1 Introduction 15

5.2 Security logs contents 15

5.2.1 Key Issue SLC#1: Logging all security related activity and events. 15

5.2.2 Key Issue SLC#2: Log authentication attempts 19

5.2.3 Key Issue SLC#3: Log encryption configuration information 19

5.2.4 Key Issue SLC#4: Logging accesses to personal data 20

5.2.5 Key Issue SLC#5: Logging events related to the Master Clock 20

5.2.6 Key Issue SLC#6: Logging details of xApps 21

5.2.7 Key Issue SLC#7: Logging details of rApps 21

5.2.8 Key Issue SLC#8: Logging details of VNF/CNF 21

5.2.9 Key Issue SLC#9: Logging information of dynamically loaded libraries 21

5.2.10 Key Issue SLC#10: Logging accesses to ML training data 21

5.2.11 Key Issue SLC#11: No differentiation between M2M and human interfaces 21

5.2.12 Key Issue SLC#12: Logging the identity of the person or process that originated the request, response, or the entity that originated the log message in the metadata 21

5.2.13 Key Issue SLC#13: Maintaining unique system references in the logs 22

5.2.14 Key Issue SLC#14: Synchronization of logs using common time sources 22

5.2.15 Key issue SLC#15: Emergency calls related logs 23

5.2.16 Key issue SLC#16: Logging details of O-Cloud 23

5.3 Security logs transfer mechanisms 24

5.3.1 Introduction 24

5.3.2 Key Issue SLTM#1: Secure storage at rest 24

5.3.3 Key Issue SLTM#2: Study security logging best practices of ONAP Project 25

5.3.4 Key Issue SLTM#3: Secure transfer protocols 25

5.3.5 Key Issue SLTM#4: Availability of logging 25

5.4 Security logs formats 27

5.4.1 Key Issue SLF#1: Standardized logging metadata to support security analytics 27

5.4.2 Key Issue SLF#2: Logging the field “date/time” in the metadata of the security audit logs. 28

5.4.3 Key Issue SLF#3: Standardized logging format 28

5.5 Log management security framework 28

6 Mitigations and Solutions 31

6.1 Introduction 31

6.2 Solution#1: Trusted environment for cluster node 31

6.2.1 Introduction 31

6.2.2 Solution details 31

6.2.3 Evaluation 32

6.3 Solution #2: Trusted environment for data repository 32

6.3.1 Introduction 32

6.3.2 Solution details 32

6.3.3 Evaluation 34

6.4 Solution #3: Solution on log-data life cycle management 34

6.4.1 Introduction 34

6.4.2 Solution details 34

6.4.3 Evaluation 35

6.5 Solution #4: Solution on (D)DoS to log-data management 35

6.5.1 Introduction 35

6.5.2 Solution details 35

6.5.3 Evaluation 35

6.6 Solution #5: Solution on authenticated time stamping 35

6.6.1 Introduction 35

6.6.2 Solution details 35

6.6.3 Evaluation 36

6.7 Solution #6: Solution on due diligence and log-data auditing 36

6.7.1 Introduction 36

6.7.2 Solution details 36

6.7.3 Evaluation 39

6.8 Solution #7: Log format 40

6.8.1 Introduction 40

6.8.2 Solution details 40

6.8.3 Evaluation 40

6.9 Solution #8: Log content: key fields for all security log events 41

6.9.1 Introduction 41

6.9.2 Solutions 41

6.9.3 Evaluation 41

6.10 Solution #9: Secure storage at rest 42

6.10.1 Introduction 42

6.10.2 Solution details 42

6.10.3 Evaluation 42

6.11 Solution #10: Secure transfer of logs 42

6.11.1 Introduction 42

6.11.2 Solution details 42

6.11.3 Evaluation 43

6.12 Solution #11: Logging in O-RAN AI/ML systems 43

6.12.1 Introduction 43

6.12.2 Solution details 43

6.12.3 Evaluation 45

6.13 Solution #12: Solution on preventing the tampering of log-Data 45

6.13.1 Introduction 45

6.13.2 Solution details 46

6.13.3 Evaluation 46

6.14 Solution #13: Security-related activities and events to be logged 46

6.14.1 Introduction 46

6.14.2 Solution details 47

6.14.3 Evaluation 49

6.15 Solution #14: Log content: solution for harmonizing timestamps 49

6.15.1 Introduction 49

6.15.2 Solutions 49

6.15.3 Evaluation 50

6.16 Solution #15: Solution on Logs Using Common Time Sources 50

6.16.1 Introduction 50

6.16.2 Solution Details 50

6.16.3 Evaluation 51

6.17 Solution #16: Availability of Logging: Solution for ensuring logging reliability 51

6.17.1 Introduction 51

6.17.2 Solutions 51

6.17.3 Evaluation 51

7 Conclusions 51

7.1 Key Issue Solution Mapping 51

7.1.1 Security logs contents 51

7.1.2 Security logs transfer mechanisms 53

7.1.3 Security logs formats 53

7.1.4 Log Management Security Framework 53

7.2 Alignment to other studies 54

7.2.1 Application Lifecycle Management Security 54

8 Recommendations 55

Annex: Change history/Change request (history) 56

# List of figures

Figure 4.1‑1 Log-management architecture 13

Figure 4.1‑2 Definition of Asset 14

Figure 5.3‑1 possible configuration for log availability 26

Figure 5.4‑1 Security Event Entry 27

Figure 6.8‑1 Log Normalization 40

Figure 6.8‑2 Log message structure and example log message 40

Figure 6.12‑1 - O-RAN AI/ML Workflow [i.43] 44

Figure 6.15‑1 Example of timestamp-adaptation in the SIEM 50

Figure 8‑1 An example Security Log Management Schema 56

# List of tables

[Table 4.2‑1: Cloud Services vs. Event Sources (Overview) 14](#_Toc172880666)

[Table 6.9‑1 Log Content 41](#_Toc172880667)

[Table 6.9‑2 Key fields 41](#_Toc172880668)

[Table 6.12‑1 - Pipeline components in reference [i.42] compared to reference [i.43]. 43](#_Toc172880669)

[Table 6.14‑1 Event categories to be logged 47](#_Toc172880670)

[Table 7.1‑1 Key Issue Solution mapping for Security logs contents 51](#_Toc172880671)

[Table 7.1‑2 Key Issue Solution mapping for Security logs transfer mechanisms 53](#_Toc172880672)

[Table 7.1‑3 Key Issue Solution mapping for Security logs formats 53](#_Toc172880673)

[Table 7.1‑4 Key Issue Solution mapping for Log Management Security Framework 53](#_Toc172880674)

[Table 7.2‑1 - Application LCM Logging Events 54](#_Toc172880675)

# 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 addresses the Security Log Management across the entire O-RAN architecture.

Security Log Management has not been part of O-RAN specifications yet, although the term FCAPS (Fault, Configuration, Accounting, Performance, Security) has been used in several documents. Specifically, the following definitions are still to be specified in O-RAN:

1. Security logs contents (payload)
2. Security logs transfer mechanisms
3. Security logs formats/schemas

Security mechanisms applied to protection of the security logs at rest and in transit, i.e. confidentiality, integrity and access control mechanisms are considered.

A major goal of the work item related to this Technical Report is to define requirements and solutions that comply with industry standards for Security Log Management.

Another goal of the work item related to this Technical Report is to provide guidance for the implementation of security monitoring use cases to O-RAN infrastructure operators.

# Scope

The present document considers the security log management as far as it applies to entities defined in the O-RAN architecture.

Taking into consideration the threats and risks identified in O-RAN Security Threat Modeling and Risk Assessment Technical Report [i.1] , the study analyses a list of key issues on Security logs contents (payload), Security logs transfer mechanisms and Security logs formats/schemas and proposes solutions in order to derive security requirements, intended to be part of O-RAN Security related Specifications.

# 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-Security-Protocols-Specifications”
4. Kubernetes Logging: A Comprehensive Guide For Beginners: <https://devopscube.com/kubernetes-logging-tutorial/>
5. Red Hat - Understanding the logging subsystem for Red Hat OpenShift: <https://docs.openshift.com/container-platform/4.10/logging/cluster-logging.html>
6. IBM - Audit logging adoption guide: <https://www.ibm.com/docs/en/cloud-private/3.2.0?topic=guide-audit-logging-adoption>
7. ETSI GR NFV 003 V1.5.1 (2020-01), Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV
8. NIST: Role-based Access Control (RBAC), <https://csrc.nist.gov/projects/role-based-access-control>
9. Kubernetes: Logging Architecture: <https://kubernetes.io/docs/concepts/cluster-administration/logging/>
10. Open Web Application Security Project (OWASP), <https://owasp.org/>
11. OWASP Logging Guide, <https://owasp.org/www-pdf-archive/OWASP_Logging_Guide.pdf>
12. NIST Special Publication 800-145, The NIST Definition of Cloud Computing
13. ISO/IEC 17789:2014, Cloud Computing - Reference Architecture
14. NIST Special Publication 800-92, Guide to Computer Security Log Management; <http://csrc.nist.gov/publications/PubsSPs.html>
15. OSI Reference Model, ISO Standard 7498-1
16. IETF TCP/IP Protocol Suite; <https://www.ietf.org/>
17. Guide to Storage Encryption Technologies for End User Devices, NIST SP800-111
18. ENISA – NFV Security in 5G - Challenges and Best Practices; <https://www.enisa.europa.eu/publications/nfv-security-in-5g-challenges-and-best-practices>
19. Red Hat, Viewing system event information in an OpenShift Container Platform cluster; <https://docs.openshift.com/container-platform/4.9/nodes/clusters/nodes-containers-events.html#nodes-containers-events>
20. Kubernetes Monitoring, Logging, and Debugging; <https://kubernetes.io/docs/tasks/debug/>
21. Docker - Audit logs; <https://docs.docker.com/docker-hub/audit-log/>
22. VMware - Audit Logging; <https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.security.doc/GUID-9BBC8AA0-4533-4943-AEF5-6A8BC64D7A5D.html>
23. OpenStack - Monitoring and logging; https://docs.openstack.org/security-guide/monitoring-logging.html
24. The MITRE Corporation, “MITRE ATT&CK® Containers Matrix”, Accessed on 03 AUG 2022, <https://attack.mitre.org/matrices/enterprise/containers/>
25. Linux Foundation, “ONAP Documentation”, Accessed on 03 AUG 2022, <https://docs.onap.org/projects/onap-vnfrqts-requirements/en/latest/Chapter4/Security.html>
26. 3GPP, TS 33.117 v17.0.0, Catalogue of general security assurance requirements
27. OWASP Foundation, OWASP Cheat Sheet Series, “Logging Cheat Sheet”
28. “Security Logs Standard: NOCH IMMER EINE UNGEKLÄRTE FRAGE”: <https://www.scip.ch/?labs.20180315>
29. O-RAN ALLIANCE TS, “O-RAN Architecture Description”
30. O-RAN ALLIANCE TR, “WG11 Study on Security for Application Lifecycle Management”
31. (EU) 2016/679: General Data Protection Regulation: <https://gdpr-info.eu/>
32. O-RAN ALLIANCE TS, “WG4 Management Plane Specification”
33. 3GPP, TS 28.537 v17.3.0, Management and orchestration; Management capabilities
34. O-RAN ALLIANCE TS, “O-RAN Operations and Maintenance Interface Specification”
35. ONAP, Security Logging Best Practice, <https://wiki.onap.org/display/DW/Security+Logging+Best+Practice>
36. IETF RFC 5424, The Syslog Protocol
37. IETF RFC 5425, Transport Layer Security (TLS) Transport Mapping for Syslog
38. NIST Special Publication 800-162, “Guide to ABAC Definition and Considerations”, January 2014, <https://doi.org/10.6028/NIST.SP.800-162>
39. Log normalization: Log Normalization - Glossary | CSRC (nist.gov)
40. O-RAN ALLIANCE TR, “O-RAN WG1 Use Cases Analysis Report”
41. O-RAN ALLIANCE, “Discussions on the AI/ML in O-RAN feature in the Near-RT RIC architecture”, December 6, 2022. <https://oranalliance.atlassian.net/wiki/download/attachments/2709356787/CMCC-2022.11.30-WG3-AIML%20in%20O-RAN-v02.pptx>
42. O-RAN ALLIANCE TR, “O-RAN WG11 WI AI/ML Security Adversarial Machine Learning (AML) Threat Analysis and Remediation”, January 2023. <https://oranalliance.atlassian.net/wiki/download/attachments/2537881609/WI_MLSec_SecAnalysisDoc_v1.docx>
43. O-RAN ALLIANCE TR, “O-RAN WG2 AI/ML workflow description and requirements” <https://oranalliance.atlassian.net/wiki/download/attachments/2123366557/O-RAN.WG2.AIML-v01.03.02.docx>
44. ISO/IEC 27000:2018, Information technology - Security techniques - Information security management systems - Overview and vocabulary
45. O-RAN ALLIANCE TS, “O-RAN Security Test Specifications”
46. IETF RFC 3339, Date and Time on the Internet: Timestamps
47. IEEE 1588-2019, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
48. ISO 8601, Date and time format
49. Cypress: How To Prevent Data Tampering In Your Business; <https://www.cypressdatadefense.com/blog/data-tampering-prevention/>
50. Security log management and logging best practices | TechTarget: <https://www.techtarget.com/searchsecurity/tip/Security-log-management-and-logging-best-practices>
51. RELP output module, <https://www.rsyslog.com/doc/v8-stable/configuration/modules/omrelp.html>
52. RELP input module, <https://www.rsyslog.com/doc/v8-stable/configuration/modules/imrelp.html>
53. IETF RFC 3080, The Blocks Extensible Exchange Protocol Core
54. IETF RFC 3195, Reliable Delivery for syslog
55. NIST Special Publication 800-34, “Contingency Planning Guide for Federal Information Systems”, <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-34r1.pdf>
56. NIST Special Publication 800-61, “Computer Security Incident Handling Guide”, August 2012, <https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf>

# Definition of terms, symbols and abbreviations

## Terms

For the purposes of the present document, the following terms and definitions apply.

**Audit Records**   
“Audit records contain security event information such as successful and failed authentication attempts, file accesses, security policy changes, account changes (e.g., account creation and deletion, account privilege assignment), and use of privileges. OSs typically permit system administrators to specify which types of events should be audited and whether successful and/or failed attempts to perform certain actions should be logged.” [i.14]

**Information Security Event**

“Identified occurrence of a system, service or network state indicating a possible breach of information security policy or failure of controls, or a previously unknown situation that can be security relevant” [i.44]

**Information Security Incident**

“Single or a series of unwanted or unexpected information security events that have a significant probability of compromising business operations and threatening information security.” [i.44]

**Log**  
“A log is a record of the events occurring within an organization’s systems and networks. Logs are composed of log entries; each entry contains information related to a specific event that has occurred within a system or network. Logs have evolved to contain information related to many different types of events occurring within networks and systems. Within an organization, many logs contain records related to computer security; common examples of these computer security logs are audit logs that track user authentication attempts and security device logs that record possible attacks.” [i.14]

**Log Data**   
The term log data is referring to a log entry as defined by the guide to computer security log management: “A log is a record of the events occurring within an organization’s systems and networks. Logs are composed of log entries: each entry contains information related to a specific event that has occurred within a system or network.” [i.14] .

**Log Preservation**

“Logs often need to be preserved to meet legal or regulatory requirements. Log preservation is keeping logs that normally would be discarded, because they contain records of activity of particular interest. Log preservation is typically performed in support of incident handling or investigations.” [i.14]

**Log Retention**

The term ‘log retention’ refers mainly to the archiving of log data. The “Log archival is retaining logs for an extended period of time, typically on removable media, a storage area network (SAN), or a specialized log archival appliance or server.” [i.14] The “old logs are archived for the appropriate length of time and then destroyed when no longer needed, in compliance with the organization’s logging, data retention, and media sanitization policies.” [i.14]

**Log Rotation**   
“Log rotation is closing a log file and opening a new log file when the first file is considered to be complete. Log rotation is typically performed according to a schedule (e.g., hourly, daily, weekly) or when a log file reaches a certain size. The primary benefits of log rotation are preserving log entries and keeping the size of log files manageable. When a log file is rotated, the preserved log file can be compressed to save space.” [i.14]

**Perimeter Security**   
The Perimeter refers to the boundary line between something that is trusted and some other that is seen as untrusted. This boundary line can be singular, but it can also span a surface/area/environment. In any case, this boundary line and the surface/area/environment is secured by security controls in place which is referring further to as perimeter security. The fact that this boundary line can be adapted according to the product makes it possible to define the perimeter security on the edge of a product, on the edge of a device, or it can be defined on service and application level.

**Storage Media**   
The term storage media refers to the physical storage of electronic data. It can be any technology to place the electronic data (e.g., hard drives, SSDs, DVDs, and others). The storage media can be internal or external to a system.

**System Events**   
“System events are operational actions performed by OS components, such as shutting down the system or starting a service. Typically, failed events and the most significant successful events are logged, but many OSs permit administrators to specify which types of events will be logged. The details logged for each event also vary widely; each event is usually timestamped, and other supporting information could include event, status, and error codes; service name; and user or system account associated with an event. “[i.14]

## Abbreviations

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

ABAC Attribute Based Access Control

AC Access Control

AES-CMAC Advanced Encryption Standard - Cipher-based Message Authentication Code

AI/ML Artificial Intelligence/Machine Learning

API Application Programming Interface

CP Control Plane

CPU Central Processing Unit

(D)DoS (Distributed) Denial of Service

DLP Data Loss Prevention

DTLS Datagram Transport Layer Security

DVD Digital Versatile Disc

E2SM E2 Service Model

eNB e NodeB (applies to LTE)

FCAPS Fault, Configuration, Accounting, Performance, Security

FM Fault Management

GDPR General Data Protection Regulation

gNB g NodeB (applies to NR)

GNSS Global Navigation Satellite System

GUI Graphical User Interface

HDD Hard Drive Disk

HW Hardware

IaaS Infrastructure as a service

IDS Intrusion Detection System

IP Internet Protocol

IPS Intrusion Prevention System

IPsec Internet Protocol Security

JWS JSON Web Signature

JWT JSON Web Token

LCM Life-Cycle Management

MEC Multi-access Edge Computing

MIMO Multiple Input Multiple Output

Near-RT RIC Near-real-time RAN Intelligent Controller

NIB Network Information Base

Non-RT RIC Non-real-time RAN Intelligent Controller

NPE non-person entity

NTP Network Time Protocol

NTPv4 Network Time Protocol version 4

NTS Network Time Security

O-CU O-RAN Central Unit

O-DU O-RAN Distributed Unit

O-RU O-RAN Radio Unit

OS Operating System

OSI Open Systems Interconnection

OWASP Open Web Application Security Project

PaaS Platform as a service

PII Personally Identifiable Information

PM Performance Management

PNF Physical Network Function

PTP Precision Time Protocol

QoE Quality of Experience

QoS Quality of Service

RAN Radio Access Network

RBAC Role Based Access Control

RB-RBAC Rule Based RBAC

RELP Reliable Event Logging Protocol

RIC RAN Intelligent Controller

R-NIB Radio-Network Information Base

RRM Radio Resource Management

SAN Storage Area Network

SaaS Software as a service

SBA Service Based Architecture

SDL Shared Data Layer

SIEM Security Information and Event Management

SMO Service Management and Orchestration

SOC Security Operations Center

(S)RAM (Static) Random Access Memory

SSD Solid State Disk

SW Software

TCP/IP Transport Control Protocol / Internet Protocol

TLS Transport Layer Security

UE User Equipment

UE-NIB UE-Network Information Base

UP User Plane

UTC Universal Time Coordinated

VM Virtual Machine

VNFC Virtual Network Function Component

# Security Log Management background, concepts and assumptions

## Introduction

Figure 4.1‑1 depicts one possible log-management architecture for VM-based or containerized implementation and introduction of SIEM. SIEM is out of scope for the O-RAN Alliance and is not specified by the O-RAN Alliance. The figure attempts to describe the log-management architecture on an abstract level, independent to whether the implementation is physical, VM-based or containerized. The figure has been created based on different sources (see [i.4], [i.5], [i.6], and [i.9]).

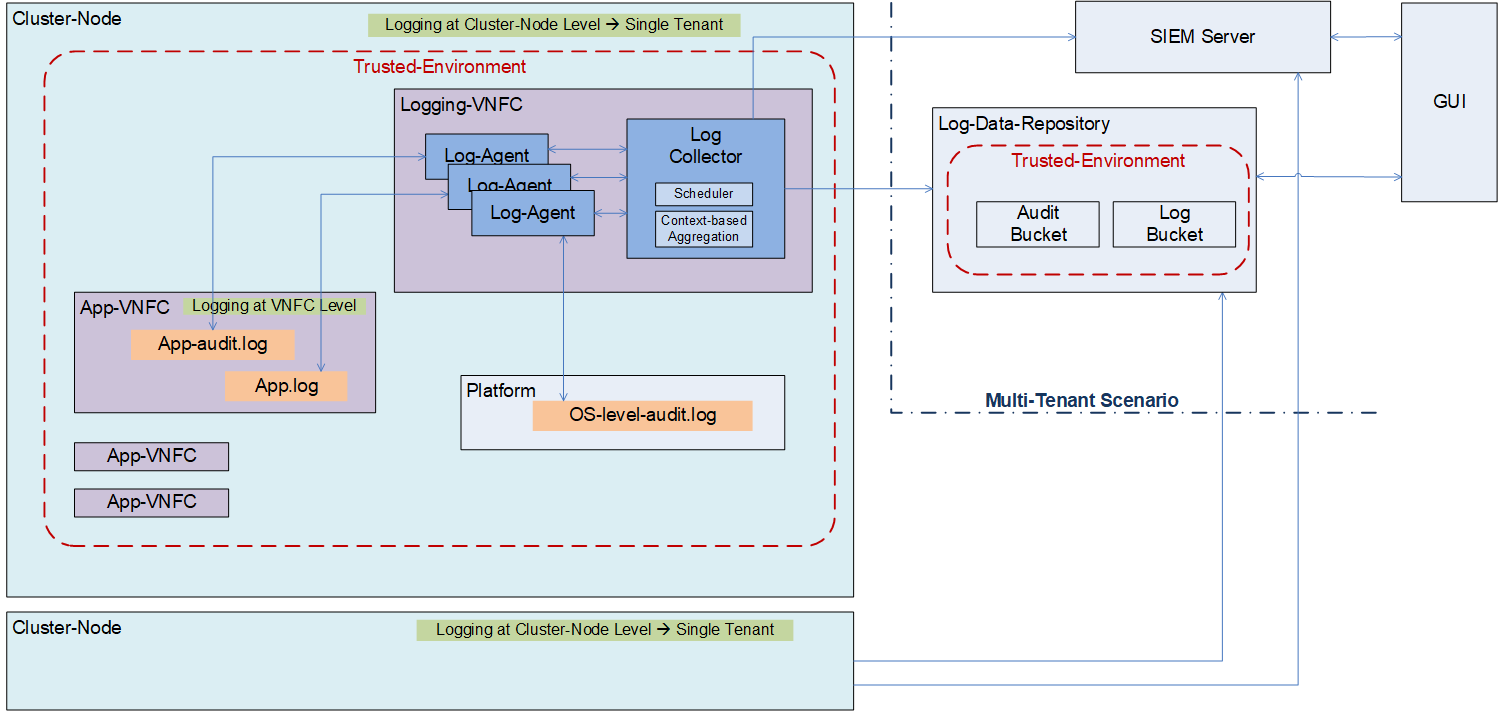


Figure 4.1‑1 Log-management architecture

The Cluster-Node is maintaining the VM-based and/or containerized product applications. These applications run as virtual network function components (VNFCs). (Refer to ETSI GR NFV 003 V1.5.1 (2020-01), [i.7]).

The perimeter of a single tenant = trusted environment is represented by the Cluster-Node. The term trusted environment implies that any piece of SW that is up and running has been verified, either during boot process and/or during systems run-time.

From log-management viewpoint, three different types of log-files are considered, these are the ‘normal’ application-level (normal-logs), the authentication-level (audit-logs), and the OS-level logs (system-logs). Each log-data is (logically) interfaced to a log-agent. The log-data is pushed to the log-agent, further, the log-agent is filtering for the relevant information and is providing this information to the log-collector. The log-collector is (logically) interfaced to every log-agent and is receiving the log-data. The log-collector scheduler is for push/pull of logs from/to the log-agents. The log-collector is providing the log-data to a log-data repository, which can then be accessed by a graphical user interface (GUI) for data visualization. The log-data will also be provided to the security information and event management (SIEM). The SIEM is a data aggregation tool. The aggregated data could be used for intrusion detection systems (IDS) and subsequent for intrusion prevention systems (IPS).

**Definition of Asset:**

The term asset in this context refers to what is of worth to log-management and needs to be protected.

The entire log-management function and the log-data represent the asset. Therefore the log-management function and the generated log-data must maintain confidentiality and integrity at rest and in use. For the definition of the security for the log-data in motion, the log-data server can be either standalone or can be part of the O-DU/SMO server. The lower layers (refer to TCP/IP [i.16] or OSI [i.15] networking model) are providing support for the confidentiality, integrity, and authentication protection of log-data in motion.

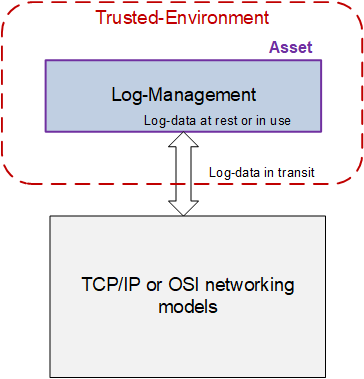


Figure 4.1‑2 Definition of Asset

The log-management as a function is creating log-data. For log-data that stays within the Trusted Environment, the log-data security applies to in use and at rest. For log-data that leaves the Trusted Environment, the log-data security applies to in motion.

## Definition of Event Sources

The NIST cloud model is defining, among other things, cloud services, specifically Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). For a more detailed description please refer to NIST SP800-145 [i.12].

The definition of cloud computing activities is as specified by ISO/IEC 17789:2014 [i.13].

Each of the cloud service is creating own log-data.

Table 4.2‑1: Cloud Services vs. Event Sources (Overview)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Physical Devices | IaaS | PaaS | SaaS |
| application-logs  (incl. normal-logs) | App | N/A | N/A | App |
| Audit-logs | App | App | App | App |
| System-logs  (incl. Cluster components-logs, Journal-logs) | App | App | App | N/A |
| Developer-logs | App | App | App | App |

Legend: App = Applicable, N/A = Not Applicable

**Application-logs** (normal-logs), refer to log-data that has been created by the application SW which is running on top of the cloud infrastructure.

**Audit-logs**, refers to Audit records that contain security event information such as successful and failed authentication attempts, file accesses, security policy changes, account changes (e.g., account creation and deletion, account privilege assignment), and use of privileges ([i.14]).

**System-logs**, refer to cluster-node level logs. Typical log-sources are VNF components and/or journal-logs.

**Developer-logs**, refer to log-data that has been created during product development. The log-data typically include development specific information which can be used for troubleshooting and debugging purposes.

## Security Log Data Lifecycle Management

The security log-data lifecycle management includes the following:

* The log-data generation refers to the creation and the collection of the log-data. The collection includes the filtering of log-data, i.e., which log-data to be stored and which not. This includes the classification of the log-data, specifically labelling of the log-data.
* The log-data at rest refers to the storage of log-data inside non-volatile memory (e.g., HDD, SSD, …) ([i.17]). This includes the security controls for confidentiality and integrity protection of log-data, which is stored local and/or remote inside some repository/backup memory. This includes the tamper protection of log-data, and proper access controls to avoid unauthenticated access to sensitive log-data.
* The log-data in motion refers to the log-data transfer, i.e., moving data between compute nodes and over a data network. This includes the security controls for confidentiality, integrity, and peer-mutual authentication protection. This includes the consideration of regulatory requirements related to privacy, i.e., general data protection regulation (GDPR) valid regulatory for handling of privacy related data within European Union.
* The log-data in use refers to log-data stored in volatile memory (e.g., (S)RAM, memory caches, CPU registers). The main purpose of log-data in use is for the analysis and auditing.
* The log-data disposal refers to the retirement (end-of-life) of log-data. This includes the time when log-data is not needed any longer and therefore can be permanently deleted. This includes the sanitisation of the storage media.

# Key Issues

## Introduction

This clause details the key issues identified for security aspects related to Security logs contents (payload), Security logs transfer mechanisms and Security logs formats/schemas. Each key issue defines the background to the issue, defines the threats related to the issue and proposes requirements that resolve or mitigate the key issue.

## Security logs contents

### Key Issue SLC#1: Logging all security related activity and events.

#### Key issue detail

Logging is an integral part of any defense-in-depth strategy. The logging of specific events makes it possible to detect adversarial behavior. O-RAN comprises many different technologies that present an adversary with a large attack surface. Therefore, multiple events originating from the network and application layers should be logged. In addition, events generated dealing with data access, accounts and identity, and security mechanisms should also be logged. Aggregating these events is necessary to develop effective analytics that will detect adversarial behavior.

#### Security threats

The following is a selection of threats applicable to O-RAN that have potentially loggable events. These threats have been selected from the references O-RAN Security Threats [i.1] and ATT&CK™ Containers Matrix [i.24].

##### Adversaries may abuse a container administration service to execute commands within a container [i.24, T1609].

##### Adversaries may build a container image directly on a host to bypass defenses that monitor for the retrieval of malicious images from a public registry [i.24, T1612].

##### Adversaries may deploy a container into an environment to facilitate execution or evade defenses [i.24, T1610].

##### An adversary may rely upon specific actions by a user in order to gain execution [i.24, T1204].

##### Adversaries may break out of a container to gain access to the underlying host [i.24, T1611].

##### Adversaries may exploit software vulnerabilities in an attempt to elevate privileges [i.24, T1068].

##### Adversaries may search compromised systems to find and obtain insecurely stored credentials [i.24, T1552].

##### Adversaries may attempt to discover containers and other resources that are available within a container’s environment [i.24, T1613].

##### Adversaries may implant cloud or container images with malicious code to establish persistence after gaining access to an environment [i.24, T1525].

##### Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code [i.24, T1053].

##### Void

##### Altering a machine learning model (System manipulation and compromise of ML data confidentiality and privacy) [i.1, T-ML-02].

##### An attacker exploits insufficient/improper mechanisms for authentication and authorization to compromise O-RAN components [i.1, T-O-RAN-06].

##### Attacks from the internet exploit weak authentication and access control to penetrate O-RAN network boundary [i.1, T-O-RAN-03].

##### An attacker bypasses authentication and authorization [i.1, T-rAPP-04].

##### An attacker can exploit the improper/missing authentication weakness on SMO functions [i.1, T-SMO-01].

##### Insecure channels with images repository [i.1, T-IMG-02].

##### Weak orchestrator configurations, access controls and isolation [i.1, §5.4.2].

##### Misuse of a VM/CN to attack other VM/CN, hypervisor/container engine, other hosts (memory, network, storage), etc. [i.1, §5.4.2].

##### Spoofing and eavesdropping on network traffic to access all O-RAN network data processed in the workload [i.1, §5.4.2].

##### Poisoning the ML training data (Data poisoning attacks) [i.1, T-ML-01].

##### Transfer learning attack [i.1, T-ML-03].

#### Potential security requirements

The following list of events are recommended to be logged. These events pertain to the network management and orchestration functions of O-RAN and not the radio parameters. The references 3GPP 33.117 [i.26] and VNF Security Events [25] list security events to log and have been reiterated here for the convenience of the reader. The references O-RAN Security Threats [i.1] and ATT&CK™ Containers Matrix [i.24] list threats so logging events have been extrapolated from those sources.

1. **Log Network Events (including physical and virtual) such as:**
   1. Network connections, port down/up, packets over limit from firewall (see 5.2.1.2.20) [i.25, R-94525] [i.26, §4.2.3.6.1].
   2. Changes to network configurations (see 5.2.1.2.20) [i.26, §4.2.3.6.1].
   3. Changes to system configuration (see 5.2.1.2.20) [i.26, §4.2.3.6.1].
   4. Resource usage, shortages, system crashes, reboots, and shutdowns [i.26, §4.2.3.6.1].
2. **Log Application Layer Events to include Virtual Network Function and Container Operations such as:**
   1. Lifecycle events, e.g., service creation/deletion/stop/start (see 5.2.1.2.18 and 5.2.1.2.19).
   2. Container and OS level administration services activities and executed commands. (e.g., Build requests, Runtime commands) (see 5.2.1.2.1 and 5.2.1.2.2).
   3. Image registry events (e.g., additions or modifications) (see 5.2.1.2.2 and 5.2.1.2.17).
   4. API calls (such as: system calls, those that deploy containers, discovery API, etc.) (see 5.2.1.2.3, 5.2.1.2.4, 5.2.1.2.5, 5.2.1.2.6, 5.2.1.2.7, 5.2.1.2.8, 5.2.1.2.9)
   5. Creation of scheduled jobs (see 5.2.1.2.10).
   6. Maintenance activity undertaken, recording whether maintenance was preventive or corrective [i.26, §4.2.3.6.1].
   7. Change notifications [i.26, §4.2.3.6.1].
3. **Log Account and Identity Events such as:**
   1. All successful and unsuccessful authentication attempts, e.g., logins, to include maximum retries alert (see 5.2.1.2.13, 5.2.1.2.14, 5.2.1.2.15 or 5.2.1.2.16) [i.25, R-55478] [i.26, §4.2.3.6.1].
   2. Successful and unsuccessful creation, modification, or deletion of user accounts or their privilege levels [i.25, R-07617] [i.26, §4.2.3.6.1].
   3. All successful and unsuccessful authorization attempts to create a session, initiate a transaction, or assume an elevated privilege (see 5.2.1.2.13 and 5.2.1.2.15)[i.25, R-54520] [i.26, §4.2.3.6.1].
   4. Termination of sessions, transactions, and downgrade in privileges.
   5. Change of group memberships [i.26, §4.2.3.6.1].
   6. Creation, modification, or deletion of identities; and transactions executed by users in applications.
   7. Anonymous requests.
4. **Log Data Access Events such as:**
   1. Successful and rejected system or data access attempts such as database queries, read or writes to files and read or writes to learning models (see 5.2.1.2.12., 5.2.1.2.21, and 5.2.1.2.22).
   2. The addition, deletion, or modification of files (see 5.2.1.2.9).
   3. Types of files accessed.
5. **Log Security Events such as:**
   1. Starting and stopping of security logging [i.25, R-13344].
   2. Activation and deactivation of security software (e.g., malware protection, DLP or IDS).
   3. Use of privileged rights or powerful utilities.
   4. Cryptographic operations

### Key Issue SLC#2: Log authentication attempts

See clause 5.2.1

### Key Issue SLC#3: Log encryption configuration information

#### Key issue detail

Misconfiguration or malicious configuration in cryptographic protocols may indicate a threat actor or identify a potential threat vector within the system.

O-RAN security specification requires O-RAN interfaces and -APIs to use security measures to protect vulnerable data and assets. Improper or missing security measures create vulnerabilities that can potentially be exploited by an attacker.

E.g., O-RAN interfaces allow use of TLS or SSH. Industry best practices mandate the use of TLS (v1.2 or higher) or SSH PKI-based authentication. An implementation that implements TLS version lower than 1.2 or a SSH password authentication, may become the key source of vulnerability that a malicious code will exploit to compromise the O-RAN system ([i.1] , T-O-RAN-05).

An attacker could, in such case, either inject malwares and/or manipulate existing software, harm the O-RAN components, create a performance issue by manipulation of parameters, or reconfigure the O-RAN components and disable the security features with the purpose of eavesdropping or wiretapping on various CUS & M planes, reaching northbound systems, attack broader network to cause denial-of-service, steal unprotected private keys, certificates, hash values, or other type of breaches ([i.1] T-O-RAN-06).

O-RAN components may not be secured to industry best practices, i.e. improper or missing ciphering and integrity protection of sensitive data in storage or in transfer ([i.1] T-O-RAN-08).

If the authentication of O-RAN external (e.g., AI/ML, Human-Machine) or internal (e.g., over O1 or O2 interfaces, with Non-RT RIC) interfaces on SMO is not supported or not properly implemented, those interfaces without valid certificates, or pre-shared keys could be able to establish a TLS connection with the SMO. The data stored in the SMO may be exposed to an attacker.

#### Security threats

An adversary attack using vulnerabilities due to improper setup of security measures may not be detected.

An adversary attack changing security configuration settings to open vulnerabilities may not be detected.

#### Potential security requirements

Any change to a security-related configuration item should be logged as a security event. The logged event must include the metadata specified in SLF#1 and the event message must indicate the specifics of the configuration change [i.26, §4.2.3.6.1].

### Key Issue SLC#4: Logging accesses to personal data

#### Key issue detail

O-RAN components may not be secured to industry best practices, i.e., ciphering and integrity protection of sensitive data in storage or in transit may be improper or missing [i.1, T-O-RAN-08]. As a result, these O-RAN components might be vulnerable to attackers inappropriately accessing and/or disclosing of personal information. Further, this data may be subject to national or international law, such as the GDPR [i.31]. A security analyst may not notice the disclosure of personal data without traceability (logging) of access to personal data.

#### Security threats

An adversary attack using vulnerabilities due to improper setup of security measures accessing personal data may not be detected. An attacker could, in such case, cause information disclosure, spoofing identity, elevation of privilege, etc.

#### Potential security requirements

Any access to personal data should be logged, and the log should contain who accessed what data without revealing personal data in clear text as a security event. The logged event must include the metadata specified in SLF#1 and the event message must indicate the specifics of the access [i.26, §4.2.3.2.5].

### Key Issue SLC#5: Logging events related to the Master Clock

#### Key issue detail

The S-plane may be subject to attacks involving a Master clock of the timing network trying to impact the availability and accuracy of the delivered time and synchronisation information. To detect and analyse these attacks, events related to the master clock should be logged.

#### Security threats

An attacker may

* attack a master clock by sending an excessive number of time protocol packets or impersonate a legitimate clock, a slave, or an intermediate clock, by sending malicious messages to the master, thus degrading the victim’s performance [i.1, T-SPLANE-01].
* impersonate the master clock’s grandmaster Identity value and proposing itself as a grandmaster candidate by sending fake ANNOUNCE messages declaring itself to be the best clock in the network ([i.1] , T-SPLANE-02).
* propose itself as a grandmaster candidate by sending manipulated/malicious ANNOUNCE messages declaring itself to be the best clock in the network. The attacker causes other nodes in the network to believe it is a legitimate master [i.1, T-SPLANE-03].
* position itself in such way that allows it to intercept and remove or delay valid synchronization packets. This leads to clock synchronization errors and inaccuracies of all clocks downstream or makes them go into free-running mode [i.1, T-SPLANE-04, T-SPLANE-05].

#### Potential security requirements

Any significant change to the configuration and performance of the PTP endpoints should be logged as a security event. The logged event must include the metadata specified in SLF#1 and the event message must indicate the specifics of the change.

### Key Issue SLC#6: Logging details of xApps

See clause 5.2.1

### Key Issue SLC#7: Logging details of rApps

See clause 5.2.1

### Key Issue SLC#8: Logging details of VNF/CNF

See clause 5.2.1

### Key Issue SLC#9: Logging information of dynamically loaded libraries

#### Key issue detail

Any SW system potentially introduces security risks. These vulnerabilities are e.g., publicly available on the US- National Vulnerability Database (NVD). While this is intended for developers to disclose and resolve vulnerabilities, it is also used by hackers to exploit these.

Vulnerabilities frequently propagate as developers re-use open- or closed- source code enabling backdoors to attacks. There have been notable vulnerabilities from downloading open-source libraries and dependencies, as well as supply chain risks when downloading Open-Source code from untrusted repositories. Developers may select SW components with known vulnerabilities and untrusted libraries that can be exploited by an attacker through a backdoor attack. Also, developers may intentionally insert backdoors in open- and closed- source code.

#### Security threats

An attacker may

* Exploit an existing vulnerability in a vulnerable version of dynamically loaded library [i.1, T-OPENSRC-01]
* Replace an invulnerable version of a dynamically loaded library with a vulnerable version, e.g. by downgrading to a previous, vulnerable version or by introducing a specifically crafted version
* Insert a vulnerability in the source code of a particular version of a dynamically loaded library [i.1, T-OPENSRC-02],

which may not be detected.

#### Potential security requirements

Load events of dynamically loaded libraries should be logged as a security event. The logged event must include the metadata specified in SLF#1 and the name and version information of the library being loaded.

### Key Issue SLC#10: Logging accesses to ML training data

See clause 5.2.1

### Key Issue SLC#11: No differentiation between M2M and human interfaces

See clause 5.2.1

### Key Issue SLC#12: Logging the identity of the person or process that originated the request, response, or the entity that originated the log message in the metadata

See clause 5.2.1

### Key Issue SLC#13: Maintaining unique system references in the logs

#### Key issue detail

The lack of security events logged together with a unique system reference (e.g. host name, IP or MAC address) and the exact time the incident occurred do not allow a correct and rapid audit in case of security incident occurrence. Security restoration is delayed. Compromised availability and integrity of security event log files could conduct to delays, wrong audit results, delays in security restoration, threats persistence.

#### Security threats

An attack may not be identified and located correctly and timely [i.1, T-O-RAN-07].

#### Potential security requirements

Security events are logged together with a unique system reference (e.g. host name, IP or MAC address) and the exact time the incident occurred [i.26, §4.2.3.6.1]. Proper unique system references are listed in clause 5.4.1.3 as ‘Location Event Attribute’ and ‘Source Event Attribute’.

### Key Issue SLC#14: Synchronization of logs using common time sources

#### Key issue detail

A common time source is crucial for security-related log timestamping because it ensures accurate and consistent timekeeping across various systems, devices, and applications within a network or organization. When security events occur and logs are generated, having synchronized timestamps allows for effective correlation, analysis, and response to incidents. Main uses are:

**Accurate Event Sequencing**

In security monitoring and incident investigation, the sequence of events is crucial. Accurate and synchronized timestamps enable analysts to reconstruct the chronological order of activities accurately, helping to identify the root cause and timeline of an incident.

**Forensic Analysis**

When investigating security breaches or incidents, forensic analysis requires the ability to track events across multiple systems. Accurate and synchronized timestamps facilitate the correlation of logs from different sources, aiding in the identification of patterns and anomalies.

**Auditing and Compliance**

Many industries and regulatory frameworks require organizations to maintain accurate logs for auditing and compliance purposes. Inconsistent or inaccurate timestamps could result in non-compliance issues and potential legal consequences.

**Incident Response**

During an incident, rapid response is essential. Accurate timestamps help security teams quickly identify the scope of an attack, assess the impact, and mitigate the threat effectively.

**Distributed Environments**

In modern IT landscapes, systems and applications are often distributed across various locations and time zones. Accurate and synchronized timestamps ensure that events are timestamped consistently, regardless of their physical location.

#### Security threats

REF: ([i.1] T-O-RAN-07).

**Data Integrity and Non-Repudiation:** Malicious actors may manipulate local system time to cover their tracks or cast doubt on the authenticity of evidence. This undermines data integrity and non-repudiation, making it difficult to prove the sequence of events accurately.

**Attack Detection and Attribution:** If timestamps are inconsistent, identifying the source of an attack and its impact becomes challenging, potentially leading to delayed response and recovery.

**Legal and Regulatory Compliance:** Organizations may be subject to legal actions or regulatory penalties if they cannot provide accurate and synchronized logs as required by regulations.

**Delayed Incident Response:** Inaccurate timestamps may lead to confusion and delay in incident response efforts. Security teams may waste valuable time trying to reconcile conflicting timestamps rather than focusing on containing and mitigating the threat.

#### Potential security requirements

The clock used to timestamp log events should be synchronized with a common time source. O-RAN elements should support NTP and/or GNSS. If NTP and GNSS are not available, then IEEE 1588 [i.47] might be an option.

### Key issue SLC#15: Emergency calls related logs

1. void

### Key issue SLC#16: Logging details of O-Cloud

#### Key issue detail

1. The continuous security of O-Cloud software (running VNFs/CNFs, Hypervisor/Container Engine, Host OS, control management, NFO/FOCOM) depends on the ability to identify attacks and reconstruct them. The earlier operators identify suspicious behavior and possible attacks, the faster they can initiate countermeasures. In the best case, they are able to stop severe attacks before harm is done. In the worst case, they at least are able to reduce the attacks severity and collect important details on the attack itself. This knowledge is important to find and fix flaws and vulnerabilities to prevent similar attacks in the future. Therefore, a key activity is to perform logging, monitoring and alerting.

#### Security threats

REF: [i.1] §5.4.2, [i.18] CH-V10 - Security logs troubleshooting failure

#### Potential security requirements

1. Any change to the O-Cloud components should be logged as a security event, and the logged event must include the identity of the entity making the change, the change, the date and the time of the change.
2. Security logs should be time synchronized and regularly scanned for events of interest.

Logs should be transferred to a log management platform outside O-Cloud to maintain their integrity and remove the risk of tampering.

O-Cloud should secure and protect logs both in-transit in terms of integrity/authenticity and at rest in terms of integrity/availability.

1. Such O-Cloud security events to be logged include:

* The presence or loading of known vulnerable software that adversaries may drop and exploit to execute code in kernel mode.
* The deployment of suspicious or unknown App/VNF/CNF images in the O-Cloud platform.
* Interactions with images and VMs/Containers by users to identify ones that are added or modified anomalously. Changes is detectable by monitoring the VL (Virtualization layer) and the management platform audit logs. Logs provides insight into API calls including those that deploy VM/Containers.
* The change of VMs/Containers and O-Cloud configurations.
* Actions that could be taken to gather information about O-Cloud, including the use of discovery API calls by new or unexpected users.
* Resource creation, modification, or deletion for detecting misuse of the O-Cloud.
* Resource usage to determine anomalous activity associated with malicious hijacking of resources such as CPU, memory, network, and graphics processing resources.
* Attempts to escape from a privileged VM/Container to the underlying host and hardware.
* Any missing security patches.
* Activities related to the management and use of certificates and keys installed in O-Cloud (e.g., TLS certificates and keys). All operations should be logged such as view/list, renew, export, import, modify or delete commands.
* All cryptographic operations carried out within O-Cloud should be logged such as signature, encryption/decryption, hashing, key generation, key destruction.

More details on events that should be logged in the VL and the management platform are provided in [i.19] for Red Hat OpenShift, [i.20] for Kubernetes, [i.21] for Docker, [i.22] for VMware and [i.23] for OpenStack.

## Security logs transfer mechanisms

### Introduction

Logs are transferred across all the assets of the ORAN system with various possible capabilities at different layers. Whichever the mechanism, it is recommended that the logs are transferred securely across the system.

### Key Issue SLTM#1: Secure storage at rest

#### Key issue detail

Improper or missing controls for protection of security event logs at rest, generated by O-RAN components, may not allow a correct and rapid audit in case of a security incident occurrence. Security restoration is delayed. Missing availability and integrity of security event logs could conduct to delays, wrong audit results, delays in security restoration and threat persistence ([i.1] §5.4.1.1).

The security storage medium may have limited capacity, so lack of storage might cause security log event to be discarded.

#### Security threats

REF: ([i.1] T-O-RAN-07).

#### Potential security requirements

It is recommended that O-RAN security event logs stored in external systems and related to O-RAN are protected with access controls so only privileged users have access to those logs ([i.1] , §4.2).

The security storage medium should be monitored for approaching capacity limits, and proper methods should be in place to avoid discarding log events, e.g. by

* Issuing an alarm ([i.35], CON-LOG-REQ-15)
* Formalizing a log rotation mechanism ([i.35], CON-LOG-REQ-F9)
* Defining for the storage a minimum time period during which it needs to support the storage of all security log event ([i.35], CON-LOG-REQ-18)

### Key Issue SLTM#2: Study security logging best practices of ONAP Project

See clause 5.3.2.

### Key Issue SLTM#3: Secure transfer protocols

#### Key issue detail

Improper or missing controls for protection of security event logs in motion, generated by O-RAN components, may not allow a correct and rapid audit in case of a security incident occurrence. Security restoration is delayed. Missing availability and integrity of security event logs could conduct to delays, wrong audit results, delays in security restoration and threat persistence ([i.1] §5.4.1.1).

In the O-RAN Open Fronthaul specification SFTP, FTPES have been already specified as secure file transfer protocols ([i.32], §5.1.4). HTTPS has been specified in 3GPP TS 28.537 ([i.33], §7.1.3). O-RAN O1 specification has been aligned ([i.34], §6.5.3).

Log transfer protocols supporting log streaming would be a necessary addition to support near real time processing of the log events by a log analyzer.

#### Security threats

REF: ([i.1] T-O-RAN-07). Potential security requirements

#### Potential security requirements

It is recommended that security event logs being transferred are protected for confidentiality and integrity using reliable, secure protocols. It is recommended that secure file transfer protocols are used as defined in [i.32] and [i.34] when security event logs are transferred in the form of files.

In addition, security event log transfer in the form of data streams should be supported, using reliable and secure stream transfer protocols such as syslog [i.36] over TLS [i.37].

### Key Issue SLTM#4: Availability of logging

#### Key issue detail

The process of logging in a complex network is subject to error conditions. In order to provide for logging availability, the logging system needs to be robust against these conditions.

#### Security threats

The security threats associated with logging reliability include:

* Outage of core elements of the logging system such as the SIEM
* Network connectivity outages
* Late arrival of logged events due to too large file size thresholds

#### Potential security requirements

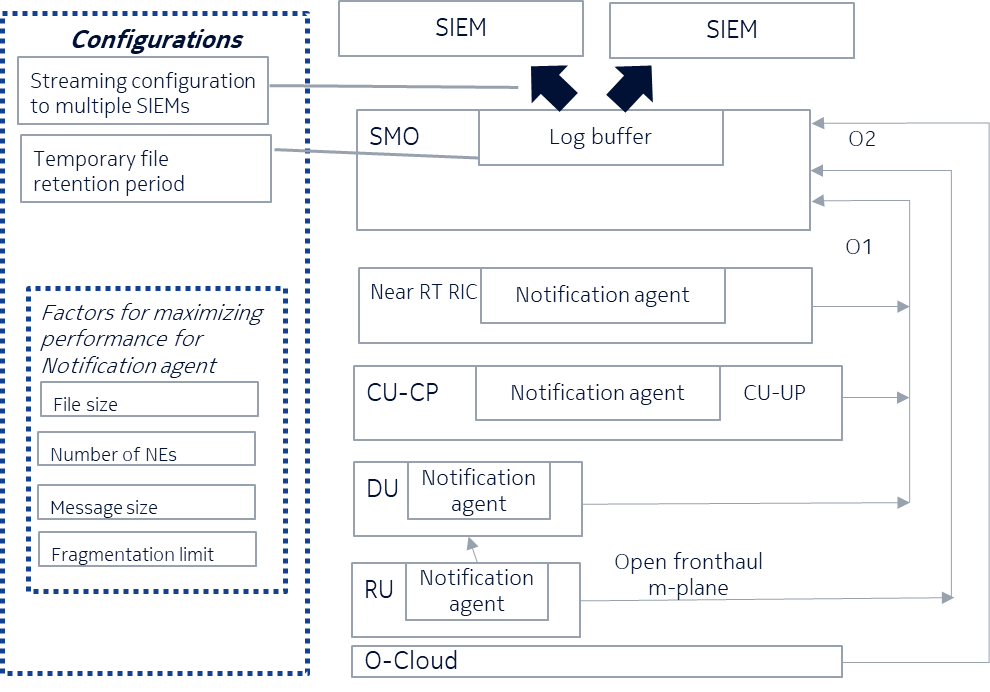


Figure 5.3‑1 possible configuration for log availability

SMO – The SMO plays the role of enabling availability of the logs. In case the SIEMs server support the configuration of redundancy then the SMO server will be capable of sending the logs to any of the instances depending on the active SIEM server at that point in time.

In addition, the SMO server also maintains an optional buffer to store logs in case the connectivity to the SIEM server goes down.

The other entities below the SMO which are forwarding the logs also strive to ensure the entire system is available by supporting parameters which keep the performance of the system optimal. The optional parameters that could be supported by the various entities are

* File size – the limit of the file size after which a notification could possibly be sent to the SMO that the file can be retrieved
* Number of NEs – the maximum number of Network Elements that can dump security logs in one file
* Message size – the length of the message size in case streaming is supported
* Fragmentation Limit – the size of the granular fragments that can be supported when data can be collected for a minimal size and then forwarded

## Security logs formats

### Key Issue SLF#1: Standardized logging metadata to support security analytics

#### Key issue detail

The intent of this key issue is not to define a logging format. There are many existing logging formats [i.28], and despite the selection of a logging format, many of the analytics tools presently are very flexible and support multiple formats. Instead, this key issue specifies that logged security event messages require additional detail referred to as event attributes to enable effective security analytics. Event attributes should indicate the time, location, source, and event type [i.27] (see Figure 5.4‑1). An analytics engine can use these event attributes downstream to identify adversarial operations within the O-RAN environment.

Event attributes typically include a field to indicate the event type. However, the event type field has been omitted since the scope of this technical report is limited to only security events. Thus, the field would have just been a static value.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Logged Security Event   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Event Attributes**   |  |  |  | | --- | --- | --- | | **Time** | **Location** | **Source** | | Security Event Message | |

Figure 5.4‑1 Security Event Entry

#### Security threats

As adversaries operate within an environment, they will generate many log events. Therefore, a correlation of these log events will be necessary to detect adversary behavior. For successful correlation, it will be essential for log events to have event attributes as specified in Figure 5.4‑1.

#### Potential security requirements

The following event attributes should be included with every logged security event.

1. **Time Event Attribute:**The date and time should be included with every log entry. This will allow for correlations and time series analysis of security events.
   1. The format of the date and time should be standardized. ISO 8601 is an example of such an appropriate standard.
   2. Precision for the time component should be at least to the second. Precision to the millisecond may be considered for near-real-time and real-time events.
2. **Location Event Attribute:**The location refers to what VM, VNF, PNF, code module, or communication exchange the security event was generated. With location information, log correlation will allow for tracing adversary movement across the O-RAN architecture or alerting a SOC to monitor a specific O-RAN component more closely. Depending on the context for which a security event is logged, one of the below items may be a suitable reference for the location event attribute.
   1. O-RAN function name (e.g., O-DU, O-RU, O-CU, Near RT RIC, etc.).
   2. The O-RAN interface name (e.g., A1, O1, O2, etc.) if it is an inter-function communication.
   3. The source IP address and destination IP address if it is an inter-function communication.
   4. Applications Identifiers (app IDs) for xApps and rApps.
   5. VM or Container names.
   6. PNF names.
   7. Code module or function name.
3. **Source Event Attribute:**The source refers to the person, process, or service that is the cause of the security event. Depending on the context for which a security event is logged, one of the below items may be a suitable reference for the source event attribute.
   1. Source IP address.
   2. User identity if known.
   3. UE associated identifiers as specified in ([i.29], §4.5)

### Key Issue SLF#2: Logging the field “date/time” in the metadata of the security audit logs.

See clause 5.4.1

### Key Issue SLF#3: Standardized logging format

See clause 5.4.1

## Log management security framework

5.5.1 Key Issue LMSF#1: Trusted environment for cluster node

5.5.1.1 Key issue detail

The VNFC is up and running (instantiated) and is creating entries into the log-files. Security professionals will use log-entries for forensic analysis in case of a security event, therefore any piece of SW (VNFC) that is going to be executed must be verified. The verification is ensuring that at time of verification the VNFC has not been altered. The verification could be at boot-time and/or during run-time. On lower layers the physical devices, as such, create log-entries, which also need to be taken into consideration.

5.5.1.2 Security threats

The attacker is running unauthenticated VNFC.

5.5.1.3 Potential security requirements

Secure boot and/or Trusted boot.

5.5.2 Key Issue LMSF#2: Trusted environment for log-data repository

5.5.2.1 Key issue detail

The log-data repository has stored audit-logs and normal-logs. Some of the logs might include sensitive information, and some other log-data might be from different owners/tenants. Access controls need to be implemented, and the permission for changes is assigned to privileged users.

5.5.2.2 Security threats

Users gain unauthenticated access to log-data stored in repository.

Unauthorized users granted read/write access to log-data stored in repository.

Users remove log-entries and/or log-data stored in repository.

5.5.2.3 Potential security requirements

Enforcement of user access controls (identity management)

Enforcement of confidentiality and integrity protection of log-data

5.5.3 Key Issue LMSF#3: Log-data lifecycle management

5.5.3.1 Key issue detail

The log-data lifecycle management is defined inconsistently across the different lifecycle phases.

5.5.3.2 Security threats

NOTE: Adding the whole Log-Data Lifecycle as a key-issue, is just to get reviewed every phase, and from that to identify potential (new) security threats.

5.5.3.3 Potential security requirements

Definition of a Retention Policy per Tenant.

5.5.4 Key Issue LMSF#4: (D)DoS to log-data management

5.5.4.1 Key issue detail

Security events will be documented in (tracked by) log-data. Every security event is creating at least one log-entry. Every log-entry is filling-up the log-data. Finite memory space for storing the log-data, consequently means size of log-data is finite as well.

5.5.4.2 Security threats

The attacker is driving (most probably wrong) actions, with the aim to trigger security events, the security events will be logged, the log-files will be overfilled. The overfilling/-loading of logging is primary aimed for hiding actions, this attack vector might also be used to drive (D)DoS for the whole systems, i.e., DoS due to VNFC processing performance leakage.

5.5.4.3 Potential security requirements

Log-data injection detection. Log-data shaping. Log-data backing-up procedures. Proper configuration of log-data management system.

5.5.5 Key Issue LMSF#5: Tampering of log-data

5.5.5.1 Key issue detail

Attackers/hackers attempt to access and tamper log-data with the aim to hide their actions. Access to log-data in use, at rest and in motion.

5.5.5.2 Security threats

5.5.5.3 Potential security requirements

Remote logging (log real time streaming). Replication of Log-data.

5.5.6 Key Issue LMSF#6: Authenticated timestamping and log-data

5.5.6.1 Key issue detail

The timestamping is the essential part of log-file-reviews, i.e., after a security event the security professionals try to reproduce who was doing what and when. The time stamping is helping to reconstruct all logged security events versus and logged actions over time. The reconstruction can be over different log-files and some cases additionally over different VNFC, maybe also over different locations. Therefore, the time must be standardized across all involved system components.

5.5.6.2 Security threats

The attacker causes the clock to become out of synchronization with the other clocks in the O-RAN deployment in order to hide the sequence of security events.

5.5.6.3 Potential security requirements

Network Time Security for NTP (RFC8915), NTPv4 support (RFC5905)

5.5.7 Key Issue LMSF#7: Due diligence - log-data auditing

5.5.7.1 Key issue detail

The log-data creation and provision to log-data repository is primary used as input to detective controls, like detecting an unauthorized use of, or attack upon a VNFC. The activation of log-management and the forensic analysis of the log-data is helpful only if security professionals perform log-data auditing regular or at least at security event notifications.

5.5.7.2 Security threats

Every undetected attack attempt increases the risk of a successful attack on the O-RAN deployment.

5.5.7.3 Potential security requirements

Definition of checklist for log-data auditing. Definition of log-data auditing process.

# Mitigations and Solutions

## Introduction

This clause contains any potential Mitigations or Solutions that may be used to address or reduce the risk associated with Key Issues identified in clause 5.

The idea behind collecting security logs is to monitor who is doing what, when and why in a network. This is typically carried out with the help of a SIEM, which is out of scope for the O-RAN Alliance and is not specified by the O-RAN Alliance.

SIEM is a common approach used to aggregate log data from multiple sources. SIEM systems can parse and analyze data in real time to identify deviations from established baselines. If an anomaly is detected, SIEM systems can generate alerts, possibly activating additional security mechanisms. They can be rules-based, often employing a statistical correlation engine to establish relationships between event log entries. Advanced systems also rely on user and entity behavior analytics, as well as security orchestration, automation and response tools. To detect and be alerted on deviations from the baselines it is recommended to implement quality logs with the following guidelines:

• The sources capturing the logs should focus on the appropriate content captured at the appropriate time

• All layers in ORAN record log events appropriately in accordance with the larger usecase in context

## Solution#1: Trusted environment for cluster node

### Introduction

The log-data life starts with the log-data generation and their storage in either local (e.g., non-volatile memory) or remote storage, i.e., log-data will be streamed to a remote log collection entity and/or log-data repository. Different cloud services create different log-data with different type of content.

For the log-data the lifecycle management applies as described by clause 4.3.

However, for all above phases of a lifecycle the trusted environment plays a major role, as this defines the boundary from trustworthy to untrustworthy environment. In a trusted area, security controls may be neglected, which does not mean that efforts are negligible, but rather that efforts are focused on achieving the trusted environment. In more simple words, the trusted environment is creating a trust boundary and allows reduction of security controls within.

### Solution details

Log-Data Storage:

* The log-data should be protected with the help of the trusted environment.
* The log-data should be created within a trusted environment.
* The log-data should be stored in non-volatile memory of the local trusted environment. This applies to back-up log-data as well.
* Any log-data modification should be logged.

Log-Data Access Management:

* The access to log-data should only be possible within the local trusted environment.
* The authorized access to log-data should be supported. The term ‘access’ in this context refers to write permission and read permission.
* The change of access rights of log-data should only be possible for privileged users.

Log-Data Handling during Boot Process, at Run-Time and System Failure:

NOTE: This clause is as well considering the handling of log-data for the different services.

* All the log-data relevant for platform services should be created.
* All the log-data relevant for infrastructure service should be created.
* All the log-data per tenant (i.e., App VNFC) should be created and maintained in dedicated files. The log-data per tenant should be kept independent of the lifetime of the App VNFC.
* The log-data should provide information to deduce the root cause of a system failure.
* Any type of log-data in use should be stored in non-volatile memory before system shutdown.

Log-Data Files in Motion:

The log-data files that are going to leave the own trusted environment should be confidentiality, integrity and replay protected. A mutual authentication should be performed for the setup of the secure communication channel.

Log-Data (Files) Policy:

* For the log-data file lifetime a retention policy should be defined.
* For details, please refer to clause 6.4 on log-data lifecycle management.

### Evaluation

The log-files and the log-data respectively will be created and maintained with the trusted environment.

Authorized read and write permission to log-data and only throughout the own trusted environment. In other words, it will not happen that any subject and/or object get access to log-data from outside of the trusted environment.

The log-data from different tenants remain valid independent to whether the tenant has deactivated its App VNFC. Rationale: This ensures a post-analysis of an App VNFC activity.

## Solution #2: Trusted environment for data repository

### Introduction

The log-data life starts with the log-data generation and their storage in either local (e.g., non-volatile memory) or remote storage, i.e., log-data will be streamed to a remote log collection entity and/or log-data repository. Different cloud services create different log-data with different type of content.

For the log-data the lifecycle management applies as described by clause 4.3.

However, for all above phases of a lifecycle the trusted environment plays a major role, as this defines the boundary from trustworthy to untrustworthy environment. In a trusted environment, some security controls protecting security logs may be optional as the trusted environment provides the necessary protection for all its contents.’ In more simple words, the trusted environment is creating a trust boundary and allows reduction of security controls within.

### Solution details

Log-Data Storage:

* The log-data should be protected with the help of the trusted environment.
* The log-data that is stored inside the trusted environment can be from one source (cluster node) or from different sources (cluster nodes). The trusted environment of a cluster node and that of the repository should be mutually authenticated. For details, please refer to the clause 6.11 Solution #10: Secure transfer of logs.
* The log-data should be stored in non-volatile memory of the local trusted environment. This also applies to back-up log-data.
* Log-data integrity needs to be always protected.

Log-Data Access Management 🡪 ***Authentication***:

It should be verified whether the subject has been authorized to use the presented identifier by a trusted identity provider organization.

Log-Data Access Management 🡪 ***Authorization***:

For the log-data management the support of attribute-based access control according to NIST SP800-162 should be supported.

NIST definition of Attribute Based Access Control (ABAC) [i.38]:  
*“An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions.”*

Subject in this context is meant to be a human user or non-person entity (NPE), while the Object in this context refers to be a system and/or resource for which access is managed by the ABAC. An operation in this context refers to the execution of a function at the request of a subject upon an object. This includes the read, write, edit, delete, copy, execute and modify. The policy in this context refers to the rules and relationships that makes it possible to determine if requested access should be allowed, given the values of the attributes of the subject, object, and possibly environmental conditions.

The core ABAC mechanisms include:

* Subject Attributes
* Resource Objects Attributes
* Access Control Rules (Policy)
* Environmental Conditions

The perimeter for the access control is given/defined by the trusted environment.

The change of access rights of log-data should only be possible for privileged users and always from inside the log-data repositories trusted environment.

Log-Data Files in Motion:

The log-data that are going to leave the own trusted environment should be confidentiality- and integrity protected. A mutual authentication should be performed for the setup of the secure communication channel between trusted environments.

For the log-data repository there should be support for log-data streaming (one-way) from the cluster node. The log-data repository should separate/isolate the log-data from different trusted environments by properly assigning the attributes.

For the log-data repository there should be support for authenticated and authorized access from a graphical user interface (GUI) for instance.

Log-Data Policy:

For the log-data file lifetime a retention policy should be defined. For details, please refer to clause 6.xon “log-data lifecycle management”.

### Evaluation

The log-files and the log-data respectively will be collected and maintained within the trusted environment of the log-data repository.

The log-data of different trusted environments need to be kept isolated from each other.

The access controls for different trusted environments need to be kept isolated from each other.

## Solution #3: Solution on log-data life cycle management

### Introduction

This log-data life cycle management clause has been defined as an own clause, this just to ensure all phases of life of a log-data has been considered properly.

The log-data lifecycle management can be based on the NIST Guide to Computer Security Log Management [i.14].

### Solution details

Log-Data Creation and Storage (at rest):

The log-data, specifically log-data of security events, will be created and thereafter need to be stored inside a non-volatile memory. The log-data should be stored inside a log-data repository, immediate after its creation. (please refer to clause 6.13 on “preventing tampering of log-data”).

Support of log-data rotation which is configurable at regular time (e.g., hourly, daily, weekly) and when the maximum log size is reached.

The creation and storage of security log-data must always be possible. In other words, stop the logging of security events is an unacceptable option.

Log-Data in Motion:

The log-data should be privacy and integrity protected.

Log-Data in Use:

The log-data should be protected with the help of the trusted environment.

Log-data Retention:

The retention policy is specific to business rules and strictly depends on applicable laws, like GDPR. Therefore the applicable laws should be considered.

For the long-term storage of log-data the following applies:

* The storage format of log-data should be in a standardized format. This ensures that log-data remain readable and interpretable after years.
* If the log-data source is proprietary, then the storage format of log-data should be in proprietary format as well. The assumption here is, that proprietary logging tools collect a lot more log-data, which might disappear in case of their transfer into a standardized format. However, this retaining the log-data in proprietary format ensures that all logged information is available.
* The archived log-data and their media should be checked periodically to determine whether there is a risk for becoming inaccessible.
* The archived log-data need to receive adequate physical protection.
* The archived log-data should not include personally identifiable information (PII).

### Evaluation

The log-data lifecycle requirements are now specified for the different phases of the life of a log-data. Some of the log-data requirements depend on applicable laws and company business, therefore applicable laws and company policies need to be taken into consideration.

## Solution #4: Solution on (D)DoS to log-data management

### Introduction

In practice the denial-of-service attack is a performance consuming attack that has the primary goal of preventing the legitimate activity on a victimized system. There exist two basic forms of denial-of-service, the first attack is exploiting a vulnerability, and the second is that the attack is overflooding the victim’s system. In practice either the communication channel is going to be overflooded which leads to that legitimate traffic cannot reach the victim’s system, or which leads to the victim’s processing power is used up completely.

From the log-data collection perspective the denial-of-service is leading to blocking of collecting and storing of security events.

### Solution details

The log-data management system should support a rate limiting and a shaping. Both functions should be configurable with the help of the trusted environment.

The log-data management system should create a log-data entry for any type of denial-of-service. In the unlikely event of a denial-of-service the victims’ system should still maintain sufficient processing power for security logging.

### Evaluation

The response of a denial-of-service attempt can be rate-limiting and/or shaping. This does not prevent from a denial-of-service, rather this is saving some performance capacity for logging while the ongoing denial-of-service.

## Solution #5: Solution on authenticated time stamping

### Introduction

For any type of log-data review the time (UTC) needs to be synchronized, especially if the reconstruction of security events can be over different tenants.

### Solution details

The Network Time Protocol (NTP) version 4 should be supported as specified by RFC5909 in case no other reliable time source is available.

If NTPv4 authentication is supported, then AES-CMAC (RFC4493) should be supported. In this case the NTP client can verify the integrity of the received NTP-packet.

If NTP security is supported, i.e., integrity and replay protection of NTP-packets, then NTS (RFC8915) should be supported. In this case the NTP client can verify the authenticity of the NTP packets by use of X.509 PKI infrastructure. For this purpose, the NTP client needs to be authenticated with the help of the trusted environment.

### Evaluation

The support of NTPv4 (RFC5905) together with AES-CMAC is providing evidence in case the NTP-packets have been altered/modified. This solution does not mitigate the replay attack.

The integrity and replay protection can be supported by implementing the NTS.

## Solution #6: Solution on due diligence and log-data auditing

### Introduction

This solution suggests the formalization of security log management policies and procedures. In addition, it suggests the development of formalized security logging structure which is based on standards, baselines, and guidelines. All these aspects refer to Due Care. On organization level, the Due Care is protecting the interest of an organization, while the Due Diligence is practicing the policies and procedures.

The term Due Diligence is the context is referring to practicing of all formalized security log management.

The Due Diligence on log-data is based in the NIST Computer Security Log Management ([i.14]).

### Solution details

This clause is not intended to serve as requirements, as such matters may fall outside the scope of the present document. For further guidance it is suggested to reference NIST SP 800-34 [i.55] and NIST SP 800-61 [i.56].

Log Analysis Scope and Criteria

* Log-data scope definition
  + Define the types of log-data to be analyzed, including specific criteria for application software, infrastructure and platform levels.
* Analysis frequency guidelines
  + Establish guidelines for how often log-data should be analyzed for each level: Critical, High, Low, Informational and Debug
* Critical information identification
  + Identify critical log-data components that require regular in-depth analysis, which may include
    - Security events
    - Performance metrics
    - Error and exception log-data
    - User activity
    - Infrastructure changes
* Automated analysis considerations
  + Consider the implementation of automated log-data analysis tools to efficiently analyze large volumes of log-data.
  + Ensure the automated log-data analysis tools and processes effectively identify patterns and recognize critical information.
  + Automated log-data analysis tools scope is limited to the escalation to personnel able to triage, interpret and initiate response effectively, unless pre-determined automated actions have been defined and tested.
* Documentation of analysis criteria
  + Ensure criteria for log-data analysis is clearly documented.
  + Log-data documentation is available for reference by analysts and other relevant stakeholders.

Access to log-data and the logging of log-data access:

* Access Authorization Definition
  + A comprehensive and clear definition is established to identify individuals and specify roles for access to log-data.
* Access Control Mechanism
  + Each component receiving O-RAN security log-data implements a robust access control mechanism to ensure only authorized individuals are able to access log-data.
* Logging Requirement
  + The availability of log-data for all authorized individuals is ensured by redundant measures.
  + Each attempt to access log-data is logged, recording details such as identity, purpose of log-data access and timestamp.
* Audit Trail
  + The system maintains a detailed audit trail for all access or access attempts to log-data.
  + The audit trail is only accessible to individuals responsible for overseeing logging and system security.
* Review and Update
  + The access authorization and logging mechanisms are periodically reviewed and updated as necessary for changing security and personnel requirements.

Log Management Infrastructure Governance:

* Definition of Authority and Responsibility
  + Clearly define roles and personnel authorized to establish and manage the log management infrastructure within the organization.
* Authorized Activities
  + Implement security controls to allow designated individuals to perform activities, including infrastructure setup, configuration management, access control and log-data rotation.
* Documentation of Responsibilities
  + Document and clearly communicate responsibilities to personnel authorized to establish, maintain and manage log infrastructure.
* Security and Compliance
  + Ensure prioritization of security best practices, conduct regular audits, and ensure compliance with regulatory standard requirements.
* Training and Coordination
  + Provide training and skill development programs that align with the organization’s IT and security policies.
  + Establish effective coordination across log-data management and security teams.

Intrusion Response:

* Definition of intrusion response
  + A comprehensive and clear definition of the intrusion response is established to include the necessary response activities to be performed in the event of a breach or intrusion.
* Response Activities
  + Identification – Promptly detect and identify signs of unauthorized access.
  + Containment – Isolate affected systems to prevent further spread and potential damage.
  + Eradication – Identify and eliminate the root cause, ensuring complete removal of any unauthorized access or changes.
  + Recovery – Restore affected components, verifying the integrity of restored components.
  + Improve security posture – Conduct a thorough analysis of the intrusion and proceeding response to help improve security posture, detection capabilities and response methods.
* Response Team and Roles
  + There is a clearly defined intrusion response team with corresponding duties for the response plan to ensure effective execution.
* Communication Protocols
  + Effective and reliable communications are established and tested to ensure the accurate and efficient spread of information within the response team and to relevant stakeholders.
* Documentation and Reporting
  + Comprehensive documentation and reporting mechanisms are established to record each step of the intrusion response, including actions taken and timeline.
* Legal and Compliance Adherence
  + The intrusion, the intrusion response plan and corresponding activities are reported to legal and compliance as per organization policy and jurisdiction.

Log Analysis Security Measures

* Confidentiality, Integrity and Availability
  + Clearly define measures for ensuring the confidentiality, integrity and availability of log-data analysis capabilities, ensuring the protection of data at rest and in motion.
* Confidentiality safeguards
  + Implement encryption protocols as per O-RAN Security Protocols Specification [i.3] to ensure confidentiality, mitigating unauthorized access.
* Integrity verification
  + Implement mechanisms to verify and maintain integrity, employing techniques such as digital signatures outlined in O-RAN Security Requirements and Controls Specifications [i.2].
* Redundancy for availability
  + Establish redundancy measures such as high availability nodes to ensure availability and minimize downtime.
  + Define and implement backup measures to ensure recovery availability in the event of catastrophe.
* Secure storage
  + Define secure storage practices for data at rest, including access controls and regular audits.

Inadvertent Disclosure Handling

* Inadvertent disclosure definition
  + Clearly define what constitutes an inadvertent disclosure of sensitive log-data, establishing a clear understanding within the organization.
* Incident Identification
  + Develop procedures to promptly identify disclosures, utilizing monitoring tools and anomaly detection.
* Notification
  + Establish clear notification protocols to inform relevant stakeholders when a disclosure has been identified. Ensure transparency and timely response.
* Containment
  + Implement containment strategies to limit the impact of disclosures, such as isolating affected systems or restricting access to log-data.
* Root Cause Analysis
  + Conduct a thorough root cause analysis to gain understanding of contributing factors.
    - Review of system logs – Identify points of failure or vulnerabilities.
    - Review access controls – Assess the effectiveness of controls and permissions to help identify misconfigurations.
    - Analyze user action – Identify whether the disclosure resulted from misconfiguration, malicious activity, or human error.
    - Document findings – Comprehensively document details of the sequence of events and identified vulnerabilities.
  + Utilize root cause analysis results to facilitate preventive measure improvements.
* Continuous Improvement
  + Integrate learned lessons from disclosures into an improvement cycle, adjusting policies to improve the organization’s security posture.
* Legal and Regulatory Compliance
  + Ensure that all response aligns with legal, regulatory and jurisdiction requirements.

### Evaluation

This Due Diligence aspect is more related to practicing all the defined policies and procedures throughout the log-management infrastructure. The policies and procedures are dependent on applicable regulatory and laws, and especially on the organizations interest.

## Solution #7: Log format

### Introduction

Logs are captured in various formats across the network sources and collected at the central trusted location where the logs need to be unified. Hence we have the need of normalizing the logs.

The common approach is that log formats are chosen for various components preferably based on known standards with a higher-level central function which focuses on the normalization.

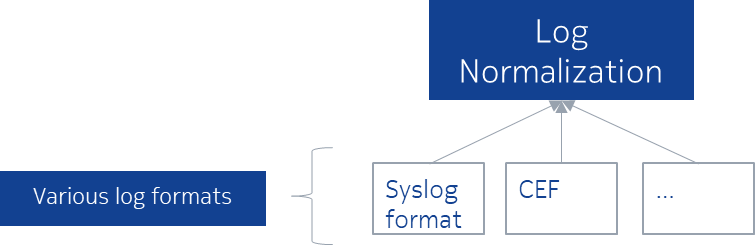


Figure 6.8‑1 Log Normalization

### Solution details

#### Normalisation of logs

o Converting each log data field to a particular data representation and categorizing it consistently for higher level SIEM systems to parse one format of data efficiently. One possible option is defined in [i.39].

#### Syslog – possible standard format

• Syslog is a standard format providing key value pairs. This is defined in [i.36]. Meaningful keys can be created such that they are able to be used by the SIEM system for analytics

• Values are free text string, numeric, enumerators, etc.

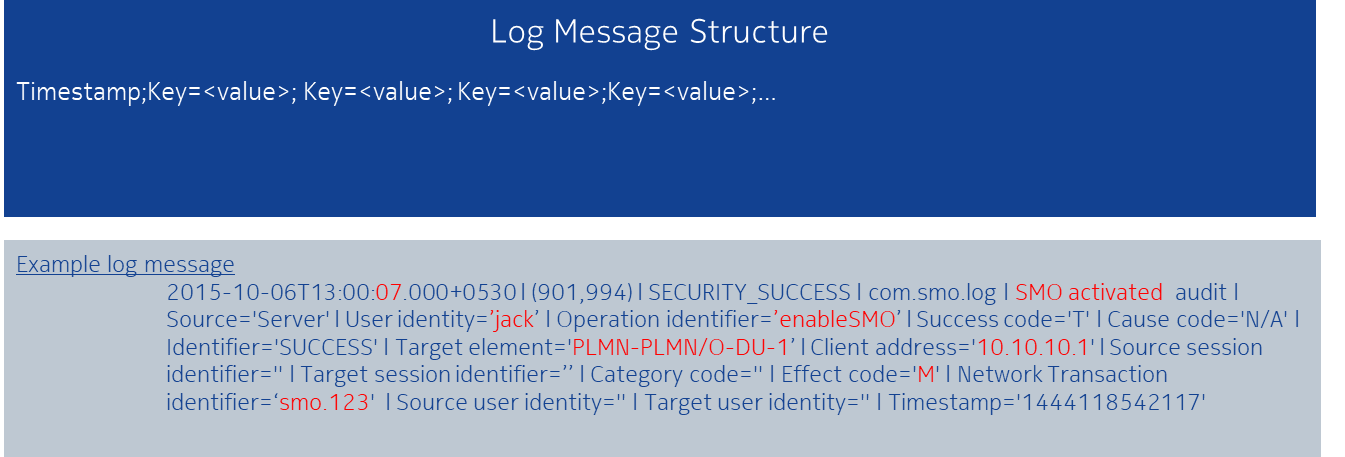


Figure 6.8‑2 Log message structure and example log message

### Evaluation

The proposed solutions aim at resolving key issues SLF#1-3.

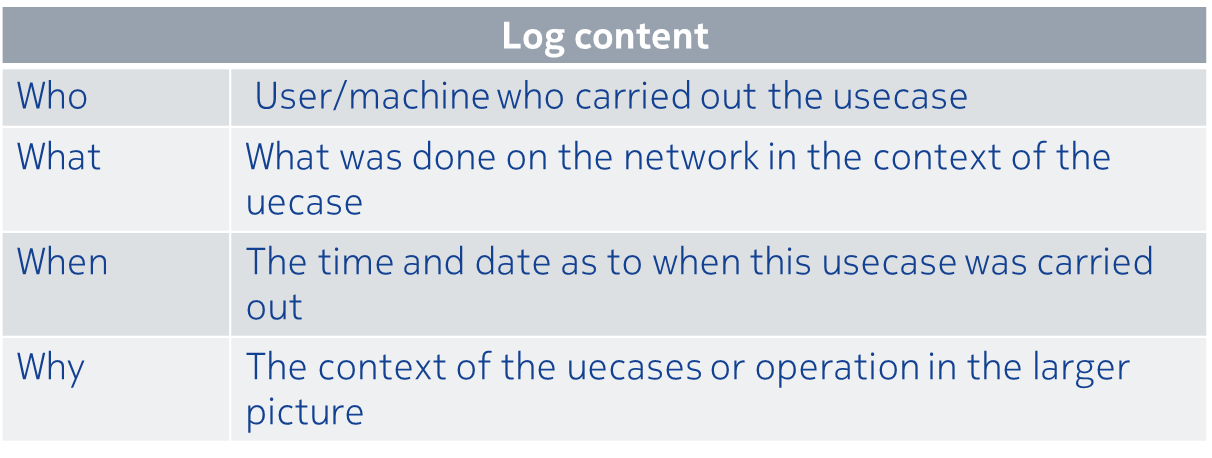
## Solution #8: Log content: key fields for all security log events

### Introduction

Several events that should always be captured and logged to ensure user accountability and to help operators detect, understand and recover from an attack as explained in [i.50]. Such events include all the key issues addressed in clause 3.

Basically the goal would be to categorize the content under the below themes as explained in the table.

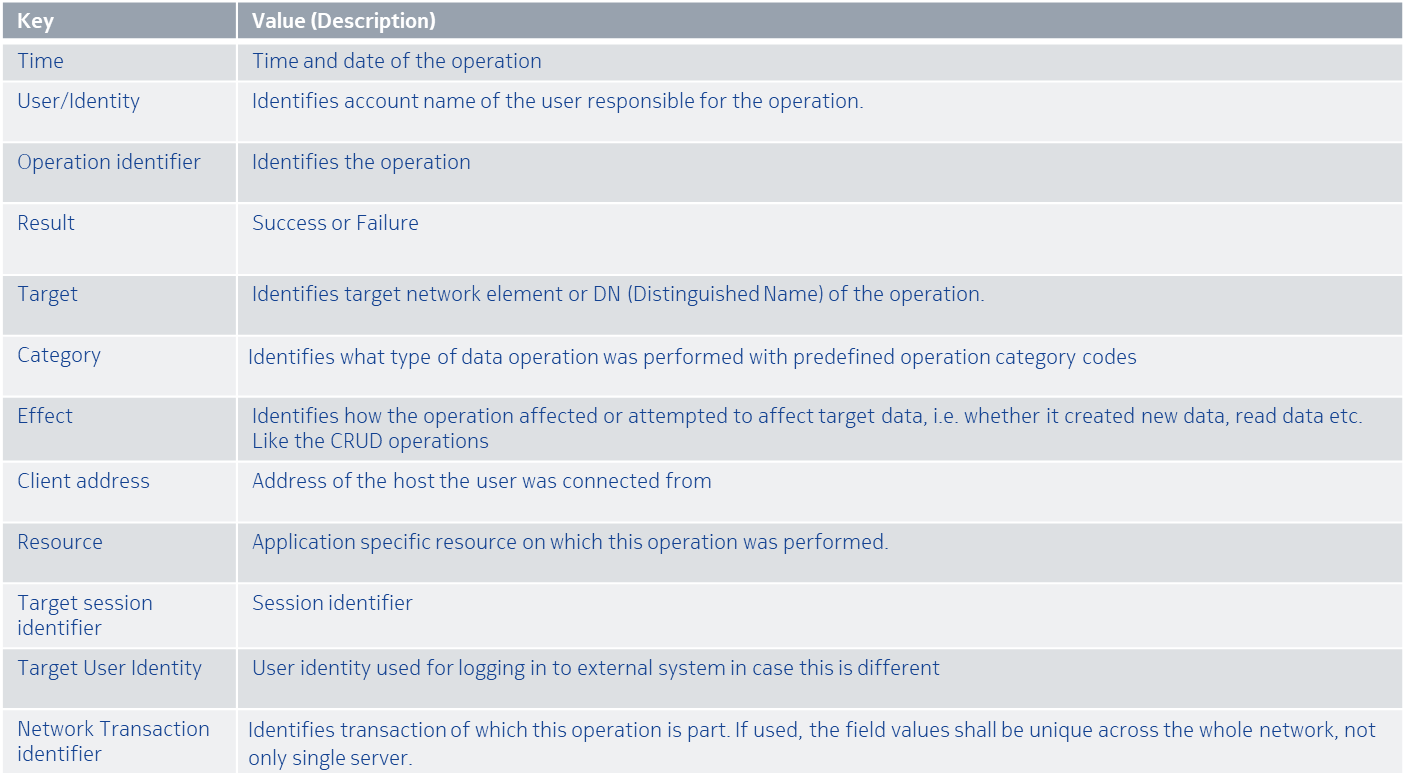
Table 6.9‑1 Log Content



### Solutions

The who did what, when and why are translated to the below table which provides some important key fields and its description but not limited to:

Table 6.9‑2 Key fields



### Evaluation

The proposed solutions aim at resolving key issues KI SLC 1, SLC12,SLC13, SLC14 .

## Solution #9: Secure storage at rest

### Introduction

Security log storage involves the safekeeping and retention of security logs for a certain period of time. It should ensure that all data security log event is retained reliably for a certain, defined, time so that no data is lost or altered, and that access to the data is restricted to authorized personnel only.

### Solution details

The following are some key requirements for effective security log storage ([i.14] ):

* Centralization: Security logs should be stored in a centralized location for easy management and analysis. This can be achieved through a centralized logging server or a cloud-based service.
* Tamper-proof: Security logs should be stored in a tamper-proof manner to ensure their integrity and authenticity. This can be achieved through digital signature, encryption, and hashing techniques.
* Retention: Retention policies should be established to determine how long logs should be kept. The retention period should be based on legal, regulatory, and compliance requirements, as well as the organization's own policies.
* Access Control: Access to the log storage should be restricted to authorized personnel only, and all access should be logged.
* Backup: Regular backups of the log storage should be performed and stored in a secure location, in case of data loss or corruption.
* Resource availability: it should be ensured that sufficient storage resources are always available in terms of both storage capacity and -bandwidth, in order to avoid data loss due to resource congestion.

### Evaluation

This solution addresses key issue SLTM#1 of the present document.

## Solution #10: Secure transfer of logs

### Introduction

Security log transfer involves the movement of security logs from one location to another, such as from a local device to a centralized logging server.

### Solution details

#### General

The following are some key requirements for effective security log transfer ([i.14] ):

* Confidentiality: Security logs should be confidentiality-protected during transfer to protect them from unauthorized access or tampering.
* Authentication: The parties involved in the log transfer should be authenticated to ensure that the logs are coming from a trusted source and going to a trusted destination.
* Integrity: Mechanisms should be in place to ensure the integrity of the logs during transfer, such as digital signatures or hash-based message authentication codes (HMACs).
* Non-repudiation: Logs transferred in bulk files should be digitally signed or otherwise marked in a way that prevents the sender from later denying having sent the logs.
* Auditing: The log transfer process should be auditable to enable the tracking and identification of any unauthorized or suspicious log transfers.

#### Recommendation of Syslog as a protocol to provide for security log streaming

Syslog is a widely used de-facto standard for logging system events. It can be used to transfer security log events from devices such as routers, servers, and security appliances to a centralized logging server. RFC 5424 [i.36] defines the protocol which is used to convey the event notification messages and RFC 5425 [i.37] defines the transport mapping of syslog to TLS.

The general recommendation is to use reliable syslog log event transfer using TCP, while utilizing TLS as the transport security layer. TLS should be used to provide:

* Confidentiality protection ([i.37] , clause 3)
* Integrity protection ([i.37] , clause 3)
* Mutual authentication ([i.37] , clause 4.2.1)
* End-entity Certificate Based Authorization ([i.37] , clause 5.1)

### Evaluation

This solution addresses key issue SLTM#3of the present document.

## Solution #11: Logging in O-RAN AI/ML systems

### Introduction

There are several use cases that have been defined within O-RAN that will utilize some form of artificial intelligence and machine learning (AI/ML) [i.40][i.41]. Broadly the use cases address automation dealing with network management, beamforming, quality of service, resource optimization, and energy saving. As illustrated by [i.42] there are several threats and attacks pertaining to AI/ML. As part of a defence in depth security posture security logging within AI/ML systems is recommended.

In addition to the ML Model itself, AI/ML systems are comprised of many different components. These components collectively are referred to as a ML pipeline. The pipeline deals with managing ML data, training the models and the lifecycle management of the trained models. The components of the ML pipeline reside in the Non-RT RIC, Near-RT RIC, O-CU, O-DU. The specific location of the ML components in the O-RAN architecture depend on the deployment scenario [42, Fig. 3: Deployment Scenarios].

There are several capabilities that an attacker can utilize against an AI/ML system as presented in [i.42]. Only a subset of attack capabilities lends themselves to an event that can be logged. Having this subset logged will provide valuable data for analytics that will indicate an AI/ML attack within O-RAN. For full coverage, other mitigations would need to be applied and that is out of scope for this technical report.

The adversary events that can be logged are adversary accesses to score-based queries, decision-based queries, hyperparameters, training data, features data, raw data, sensor data, and labelled data.

### Solution details

#### AI/ML Workflow Comparison from general to specific

A generalized ML pipeline has been proposed in [i.42] while [i.43] presents a ML pipeline that is specific to the O-RAN environment. Following is a mapping of the general ML pipeline in [i.42] to the O-RAN specific pipeline in [i.43]. This mapping will allow us to apply the threats enumerated in [i.42] to be applied to the O-RAN environment.

Table 6.12‑1 - Pipeline components in reference [i.42] compared to reference [i.43].

|  |  |
| --- | --- |
| General ML Pipeline Component [i.42] | O-RAN Specific Pipeline Component [i.43] |
| Data Lake | Data Preparation |
| Data Host |
| Feature Store | Feature Engineering in the ML Training Host and ML Inference Host |
| Training Host | ML Training Host |
| Model Catalog | Model Management |
| Serving Host | ML Inference Host |
| ML-Assisted App Host | Actor & Action |

The below figure depicts the AI/ML pipeline for O-RAN. It has been adapted from ([i.43], Figure 1) to include some coloring of blocks to relate a general pipeline presented in the O-RAN AML Threat Analysis [i.42]. In addition, the arrows that connect the boxes together have been labelled A through J. This is to assist in discussing what should be logged in the next clause.

Diagram

Description automatically generated

Figure 6.12‑1 - O-RAN AI/ML Workflow [i.43]

#### O-RAN AI/ML Workflow Description

In the O-RAN AI/ML workflow specified in Figure 6.12‑1 [i.43], a series of phases make up the workflow, starting with data collection and preparation and moving through model building, training, deployment, execution, validation, continuous self-monitoring, and self-learning/retraining. The ML Training host trains the model, while the Model Management host manages the ML models for deployment in the ML Inference host. Model compiling, which can be performed offline, optimizes the trained models for efficient execution and can be handled by the optional Model Compiling host. The compiled models are then published to the Model Management host for deployment to the ML Inference host, where the model is executed, and the Actor makes decisions. These hosts are network functions and depending on the deployment scenario for a particular O-RAN use case, can reside in either the Non-RT RIC, Near-RT RIC, or O-CU/O-DU.

#### O-RAN AI/ML Workflow Threats

In the O-RAN AML Threat Analysis [i.42] there are several adversary attacks and threats that will produce events that can be logged.

1. During an evasion or model extraction attack an adversary will repeatedly query a ML Model to recover probability vectors and classification labels ([i.42]: ACM1, ACM2). This adversary behaviour can be observed and correlated on *connections A, F and H*.
2. To facilitate evasion, poisoning, membership inference, data reconstruction, model corruption, or resource exhaustion attacks, an adversary will require knowledge about the internals of a model ([i.42], AKM1), such as model parameters, architecture, training time, and hyperparameters (e.g., the number of epochs and learning rate). All these assets should be considered sensitive information, and access to them should be logged. Model parameters reside in the ML Training Host, the ML Compiling Host, Model Inference Host, and the Model Management Host. In addition, model parameters also transit connections D, E and G.
3. With access to training and features data an adversary can corrupt ML models via poisoning attacks ([i.42], ACD1, ACD2). Training data should be considered sensitive information and access should be logged. Training data resides in the Data Preparation host and ML Training host and data transits *connection B*.
4. An adversary can carefully craft the data that is ingested into an ML system to facilitate several attacks pertaining to evasion, corruption, membership inference, data reconstruction, model extraction, and resource exhaustion ([i.42], ACD3, ACD4). However, other knowledge is required by the attacker to make use of this raw data. Logging raw data over *connection A* may not provide much value to a defender. Logging metadata over *connection A* may provide valuable data to enable correlation when developing analytics. It is worth noting that there are many sources of raw data with O-RAN. See Figure 6.12‑1 in the Data box for the full list.
5. In general, there are several points within the ML pipeline that an adversary can gain access to data that is sensitive. The exposure of that information or changing of that data can lead to several undesirable impacts to the secure operation of the ML assisted application. Therefore, it is recommended that all reads and writes to any component in the ML pipeline (data preparation host, ML training host, model management host, ML compiling host, ML inference host) are logged.

#### AI/ML Logging Requirements

This clause specifies the logging requirements to address the threats defined in the clause 6.12.2.3.

1. Metadata on connections A, F and H should be logged. It is important that the metadata contain an identifier that makes it possible to correlate that data collected from the connections A, F and H.
2. All reads, writers or modifications to model parameters should be logged. Model parameters are stored on the ML Training Host, the ML Compiling Host, Model Inference Host, and the Model Management Host.
3. All reads, writes or modifications to training data or features data should be logged. This sensitive data resides on the Data Preparation host and the ML Training host.
4. Logging requirement for the Threat number 4 from the above clause 6.12.2.3 is covered by logging requirement number 1.
5. All accesses to the Data Preparation host, ML Training host, Model Management host, ML Compiling host, and ML Inference host should be logged.

### Evaluation

This solution addresses key issue SLC#10 of the present document.

## Solution #12: Solution on preventing the tampering of log-Data

### Introduction

The log files are the first artifacts that attackers will use to hide their actions. Therefore, the tampering of log-data should be prevented at first place.

### Solution details

**Remote Logging / Replication:**

A copy of the log-data (e.g., a security event) should be stored immediate inside the log-data-repository, i.e., at the point in time when log-data is to be stored local a copy should be created for the storage in the log-file-repository.

Rationale: If the log-data is stored in a ‘physically separate’ memory space, the attacker, even if privileged, needs to obtain access to the log-data-repository. This can be seen as a second level of defense, i.e., the attacker obtains access to the cluster-node (1st level of defense) and thereafter needs to obtain access to the log-data-repository (2nd level of defense).

**One-way communication:**

The communication between the cluster-node and the log-data repository should be one-way, i.e., the log data transfer from the cluster node to the log-data-repository should be via an API (e.g., stdout). This API allows the cluster node to write log data to the log data repository and prevents the reading of log data.

Rationale: If there is a One-way transfer of log-data, then there is no way for the attacker to obtain access to the cluster node and from there to obtain relevant information which is stored inside the log-file-repository.

**Cryptographic Hash Chaining (Block Chaining):**

The digest of log-data will be created and appended to existing and preceding digests with the aim to keep the cryptographic chain and to attest the completeness and the integrity of the security events.

Rationale: If the attacker is removing at least one digest, then this is breaking the cryptographic chain. This Block-Chaining should be done over the log-file digest, just to safe computational power.

**Data Integrity using digital signatures:**

This scheme utilizes asymmetric cryptography as well as hash function. Initially the digest of a log is obtained and is then signed by a private key. When a log is needed, the use of the public key and hash function ensure the integrity and authenticity of the log.

**Limit access to log data repository:**

Users should not have any access to log data repository unless some level of access is necessary for creating log entries. Additionally, users should not be able to rename, delete, or perform other operations on log data repository.

### Evaluation

Remote logging ensures that log-data will be stored at different locations, and the hash chaining is providing some sort of book-keeping of security events.

The one-way communication is hiding information.

## Solution #13: Security-related activities and events to be logged

### Introduction

As explained in clause 5.2.1 multiple events originating from the network and application layers should be logged. In addition, it is recommended that events generated dealing with data access, accounts and identity, and security mechanisms are also logged.

### Solution details

It is recommended to log the following events:

Table 6.14‑1 Event categories to be logged

| Event Category | Description |
| --- | --- |
| Network- and System Events | |
| Network connections | Network- related events such as:   * port down/up incl. cause, * packets over limit from firewall |
| Network configurations | Changes to network configurations |
| System configuration | Changes to system configuration |
| Resource usage | Resource-related events such as:   * Shortages * system crashes * reboots * shutdowns * (virtual) Resource creation and -deletion |
| System-wide events | System-wide events such as:   * Startup * Shutdown * Restart * Crash |
| Application-Layer events | |
| Lifecycle events | Lifecycle events such as:   * service creation/deletion/stop/start |
| Container and OS level administration services activities and executed commands | e.g. Build requests, Runtime commands |
| Image registry events | Image- registry related events such as:   * additions * modifications * removal |
| Lifecycle API calls | API calls related to the application lifecycle such as system calls   * to deploy containers * discovery API |
| Creation of scheduled jobs | Events of creation of scheduling of tasks (i.e. programs and scripts), to be executed e.g.   * at a particular time * at startup |
| Maintenance Activities | Maintenance activities undertaken |
| Account and Identity Events | |
| Authentication attempts | successful and unsuccessful authentication attempts, e.g.,   * logon (including max retries alert) * logoff |
| Change of account information | Successful and unsuccessful creation, modification, or deletion of user accounts or their privilege levels |
| Authorization Attempts | All successful and unsuccessful authorization attempts to create a session, initiate a transaction, or assume an elevated privilege |
| Termination of sessions, transactions, and downgrade in privileges |  |
| Change of group memberships | Any change of group membership for accounts |
| Identity- and transaction- related events | Creation, modification, or deletion of identities; and transactions executed by users in applications |
| Anonymous requests |  |
| Data Access Events | |
| Data access attempts | Successful and rejected system or data access attempts such as   * database queries * read or writes to files * read or writes to learning models |
| File operations | The addition, deletion, or modification of files |
| File types | Types of files accessed |
| General Security Events | |
| Security logging | Starting and stopping of security logging |
| Security SW | Activation and deactivation of security software (e.g., malware protection, DLP or IDS). |
| Privilege | Use of privileged rights or powerful utilities |
| Cryptographic operations |  |
| Security- related configuration | Any change to a security-related configuration item. The logged event must include the specifics of the configuration change |
| Accesses to personal data | Any access to personal data, including who accessed what data without revealing personal data in clear text. The logged event must include the specifics of the access |
| Master-Clock – related events | Any significant change to the configuration and performance of the PTP endpoint. The logged event must include the specifics of the change |
| Use of dynamically loaded libraries | Load events of dynamically loaded libraries. The logged event must include the name and version information of the library being loaded |
| 1. O-Cloud- component- related events | |
| Vulnerable software | The presence or loading of known vulnerable software |
| App/VNF/CNF images | The deployment of suspicious or unknown App/VNF/CNF images in the O-Cloud platform |
| Interactions with images and VMs/Containers | Interactions with images and VMs/Containers by users to identify ones that are added or modified anomalously. |
| Escape attempts | Attempts to escape from a privileged VM/Container to the underlying host and hardware |
| Security patches | Security patches submitted, but not applied |
| Certificates and keys | Activities related to the management and use of certificates and keys installed in O-Cloud (e.g., TLS certificates and keys). such as   * view/list * renew * export * import * modify * delete |
| Cryptographic operations | All cryptographic operations on resources such as   * signature * encryption/decryption * hashing * key generation * key destruction. |
|  |  |

### Evaluation

This solution addresses key issue SLC#1-12, 14 and 16 of the present document.

## Solution #14: Log content: solution for harmonizing timestamps

### Introduction

* Every security log event has a timestamp field to be populated
* Each product logs the security events with the timestamp information including the related timezone
* A possibility of harmonizing the timezones is required

### Solutions

* The top-level log-aggregating component, usually a SIEM, adapts the timestamp a unique, the desired timezone after conversion and harmonisation
* E.g.: Component in the CU logs with timestamp UTC +2:00 and the Near-RT RIC located elsewhere we have UTC -1:00. Logs are transferred with the timestamp retained and converted to required format in the SIEM

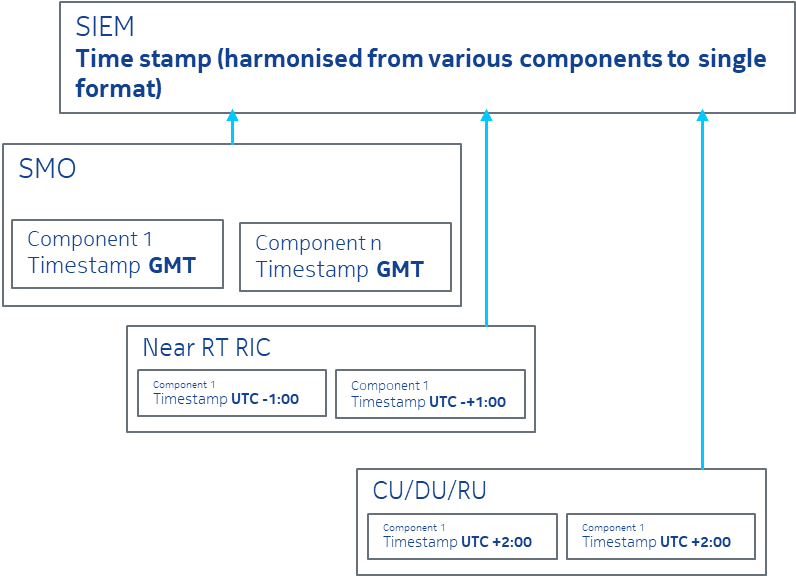


Figure 6.15‑1 Example of timestamp-adaptation in the SIEM

### Evaluation

The proposed solutions aim at resolving key issue SLF#2

## Solution #15: Solution on Logs Using Common Time Sources

### Introduction

This solution suggests the support of common time sources for logging.

### Solution Details

The logs should be time-stamped with common time source, typically NTP and/or GNSS should be supported. If NTP and GNSS are not available, then the IEEE 1588 [i.47] might be an option.

If the common time source is missing, then the usage of system time should be indicated in Log-data.

If the common time source is missing, then the log entry should be time-stamped with the current system time.

If the common time source is missing, then the order of the log entries should be kept as the (security) log events occur.

The timestamp representation should be in a standard format, and the used formatting profile should be logged. Reference to RFC 3339 [i.46] and ISO 8601 [i.48].

If the system time cannot be synchronized with any common time source, then the system should time stamp any log entry with the available system time.

NOTE: The system time is not the same as the local time. Although the usage of a ‘Unqualified Local Time’ is somewhat unwanted, the log entries should be time-stamped while the system attempts to synchronize with UTC.

The successful synchronization to UTC should be logged.

### Evaluation

This log support using common time sources is relevant for systems producing multiple log streams, though, the timestamps play a critical role in correlating entries across streams and in deriving a global order. The systems should use different mechanisms for synchronization to UTC.

This solution is solving the key issue SLC#14.

## Solution #16: Availability of Logging: Solution for ensuring logging reliability

### Introduction

As articulated in clause 5.3.4.2, there are significant challenges to achieving logging reliability. This clause is dedicated to providing an optional capability to address logging reliability problems via the addition of the Reliable Event Logging Protocol (RELP).

### Solutions

* Reliable Event Logging Protocol (RELP) extends the functionality of the syslog protocol to provide reliable delivery of event messages. RELP does not as yet have a dedicated RFC, but it is inspired by RFC 3080 [i.53] and RFC 3195 [i.54].
* RELP offers a backchannel and message receipt tracking to ensure that syslog is aware of which logs have been received correctly.
* RELP supports TLS which shall be enabled to ensure confidentiality, integrity and mutual authentication.

### Evaluation

The proposed solutions aim at resolving key issue SLTM#4.

# Conclusions

## Key Issue Solution Mapping

### Security logs contents

This document currently contains 16 key issues (SLC#1 to SLC#16) related to Security log contents (Key issue SLC #15 being void) and 3 solution proposals (Sol#8, Sol#13 and Sol#15) to address or reduce the risk associated with the key issues. A mapping of the solutions to the key issues is shown here.

Table 7.1‑1 Key Issue Solution mapping for Security logs contents

| *Key issue* | Sol # 1 | Soln#2 | Sol #3 | Sol #4 | Sol #5 | Sol #6 | Sol #7 | Sol #8 | Sol #9 | Sol #10 | Sol #11 | Sol #12 | Sol #13 | Sol #14 | Sol #15 | Sol #16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *SLC#1* |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |
| *SLC#2* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#3* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#4* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#5* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#6* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#7* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#8* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#9* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#10* |  |  |  |  |  |  |  |  |  |  | X |  | X |  |  |  |
| *SLC#11* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| *SLC#12* |  |  |  |  |  |  |  | X |  |  |  |  | X |  |  |  |
| *SLC#13* |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| *SLC#14* |  |  |  |  |  |  |  | X |  |  |  |  | X |  | X |  |
| *SLC#15 (void)* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *SLC#16* |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |

### Security logs transfer mechanisms

This document currently contains 4 key issues (SLTM#1 to SLTM#4) related to Security log transfer mechanism and 4 solution proposals (Sol#6, Sol9, Sol#10 and Sol#16) to address or reduce the risk associated with the key issues. A mapping of the solutions to the key issues is shown here.

Table 7.1‑2 Key Issue Solution mapping for Security logs transfer mechanisms

| *Key issue* | Sol # 1 | Soln#2 | Sol #3 | Sol #4 | Sol #5 | Sol #6 | Sol #7 | Sol #8 | Sol #9 | Sol #10 | Sol #11 | Sol #12 | Sol #13 | Sol #14 | Sol #15 | Sol #16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *SLTM#1* |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| *SLTM#*2 |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| *SLTM#3* |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| *SLTM#4* |  |  |  |  |  | X |  |  | X |  |  |  |  |  |  | X |

### Security logs formats

This document currently contains 3 key issues (SLF#1 to SLF#3) related to Security logs formats and 2 solution proposals (Sol#7 and Sol#14) to address or reduce the risk associated with the key issues. A mapping of the solutions to the key issues is shown here.

Table 7.1‑3 Key Issue Solution mapping for Security logs formats

| *Key issue* | Sol # 1 | Soln#2 | Sol #3 | Sol #4 | Sol #5 | Sol #6 | Sol #7 | Sol #8 | Sol #9 | Sol #10 | Sol #11 | Sol #12 | Sol #13 | Sol #14 | Sol #15 | Sol #16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *SLF#1* |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| *SLF#*2 |  |  |  |  |  |  | X |  |  |  |  |  |  | X |  |  |
| *SLF#3* |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |

### Log Management Security Framework

This document currently contains 7 key issues (LMSF#1 to LMSF#7) related to Log Management Security Framework and 7 solution proposals (Sol#1, Sol#2, Sol#3, Sol#4, Sol#5, Sol#6 and Sol#12) to address or reduce the risk associated with the key issues. A mapping of the solutions to the key issues is shown here.

Table 7.1‑4 Key Issue Solution mapping for Log Management Security Framework

| *Key issue* | Sol # 1 | Soln#2 | Sol #3 | Sol #4 | Sol #5 | Sol #6 | Sol #7 | Sol #8 | Sol #9 | Sol #10 | Sol #11 | Sol #12 | Sol #13 | Sol #14 | Sol #15 | Sol #16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *LMSF#1* | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *LMSF#*2 |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *LMSF#3* |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *LMSF#4* |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| *LMSF#5* |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| *LMSF#6* |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| *LMSF#7* |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |

## Alignment to other studies

### Application Lifecycle Management Security

The Application Lifecycle Management Security Work Item in WG11 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 in the technical report “Study on Security for Application Life Cycle Management” [i.30]. This list of events may be used to develop security analytics that have the potential to detect anomalous behaviour. These events were mapped to Key Issue SLC#1 Logging Security Related Activity and Events in order to ensure coverage of these events, and Key Issue SLC#1 was updated to account for missing application life cycle logging events.

Table 7.2‑1 shows this mapping of specific application LCM events to the categories described in Key Issue SLC#1 to show coverage of the application events.

Table 7.2‑1 - Application LCM Logging Events

|  |  |
| --- | --- |
| **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 |

# Recommendations

This is a list of recommendations on next steps to be done on the contents of the present document in future revisions and on other O-RAN specifications based on the current state of this document.

**Recommendation 1**: This work item should contribute to the WG11 Security Requirements specifications [i.2] with new requirements and security controls from the solutions outlined in this report. CRs will be generated as required.

**Recommendation 2**: This work item should contribute security test cases for relevant security requirements to the WG11 Security Test specification [i.45]. CRs will be generated as required.

**Recommendation 3**: This work item should contribute to the WG11 Security Protocols specifications [i.3] with new protocol- related requirements and controls from the solutions outlined in this report. CRs will be generated as required.

**Recommendation 4**: This work item should contribute to the WG11 O- Security Threat Modeling and Remediation Analysis [i.1] with new threats and risks from the key issues outlined in this report. CRs will be generated as required.

**Recommendation 5**: WG11 should collaborate with WG4 and WG10 to ensure the WG11 security specifications are appropriately referenced by WG4 and WG10 in their specifications.

**Recommendation 6:** the following diagram represents a possible structure to carry out log management. O-RAN does not define any such structure.



Figure 8‑1 An example Security Log Management Schema

# Annex: Change history/Change request (history)

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
| 11/8/2022 | v01.00 | Version 1 of the Technical Report |
| 23/02/2023 | v02.00 | Version 2 of the Technical Report |
| 20/07/2023 | V03.00 | Version 3 of the Technical Report |
| 22/11/2023 | V04.00 | Version 4 of the Technical Report |