- O-RAN.WG11.TR.AIML-Security.0-R004-v03.00

Technical Report

Copyright © 2025 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 DE32172018

O-RAN Work Group 11 (WG11)

Study on Security for

Artificial Intelligence and Machine Learning (AI/ML)

in O-RAN

Contents

Foreword 5

Modal verbs terminology 6

1 Scope 7

2 References 8

2.1 Informative references 8

3 Definition of terms, symbols and abbreviations 11

3.1 Terms 11

3.2 Symbols 12

3.3 Abbreviations 12

4 AI/ML Assets 13

4.1 Introduction 13

4.2 Functions 16

4.3 Interfaces 16

4.4 Data 17

4.5 Models 17

4.6 Pipeline 18

5 Threats 19

5.1 Threat Model 19

5.2 Threat Template 19

5.3 Potential Threats and Exploits 20

5.3.1 Common Threats associated with ML 20

5.3.2 Input Manipulation Attack (ML01:2023) 22

5.3.3 Data poisoning attacks (ML02:2023) 23

5.3.4 Model Inversion Attack (ML03:2023) 24

5.3.5 Membership Inference Attack (ML04:2023) 25

5.3.6 Model Stealing (ML05:2023) 26

5.3.7 AI Supply Chain Attacks (ML06:2023) 26

5.3.8 Transfer Learning Attack (ML07:2023) 27

5.3.9 Model Skewing (ML08:2023) 27

5.3.10 Output Integrity Attack (ML09:2023) 28

5.3.11 Model Poisoning (ML10:2023) 29

5.3.12 Evasion attacks 30

5.3.13 AI Energy-Latency Attack 30

6 Threat Analysis 32

6.1 Input Manipulation Attack (ML01:2023) 32

6.1.1 Training Data Control 32

6.1.2 Model Control 32

6.1.3 Testing Data Control 32

6.1.4 Label Manipulation 32

6.1.5 AI/ML Query Exploitation 33

6.2 Data poisoning attacks (ML02:2023) 33

6.2.1 Black-box Data Poisoning 33

6.2.2 Grey-box Data Poisoning 33

6.2.3 White-box Data Poisoning 34

6.3 Model Inversion Attack (ML03:2023) 34

6.3.1 Black-box Model Inversion 34

6.3.2 White-box Model Inversion 35

6.4 Membership Inference Attack (ML04:2023) 35

6.4.1 Information Access – Membership Inference 35

6.4.2 Inference Technique – Membership Inference 35

6.4.3 Target based Membership Inference 36

6.5 Model Stealing (ML05:2023) 36

6.5.1 Intellectual Property Theft 36

6.5.2 Data Privacy Violations 36

6.5.3 Adversarial Attacks with Stolen Models 37

6.6 AI Supply Chain Attacks (ML06:2023) 37

6.6.1 Data Poisoning via Supply Chain 37

6.6.2 Model Tampering 37

6.6.3 Backdoor via Supply Chain 38

6.6.4 Hardware Tampering 38

6.6.5 Insecure Maintenance APIs 38

6.6.6 Third-party Libraries and Dependencies 39

6.7 Transfer Learning Attack (ML07:2023) 39

6.7.1 Data Poisoning for Transfer Learning 39

6.7.2 Model Stealing for Transfer Learning 39

6.7.3 Adversarial Examples 40

6.7.4 Backdoor for Transfer Learning 40

6.8 Model Skewing (ML08:2023) 40

6.8.1 Data manipulation for Model Skewing 40

6.8.2 Feedback manipulation for Model Skewing 41

6.9 Output Integrity Attack (ML09:2023) 41

6.9.1 Denial of Service Impacting Output Integrity 41

6.9.2 Subscriber QoE Change 41

6.10 Model Poisoning (ML10:2023) 42

6.10.1 Model Parameter Poisoning 42

6.10.2 Inference Algorithm Poisoning 42

6.10.3 Learning Algorithm Poisoning 43

6.10.4 Model Control 43

6.11 Evasion attacks 43

6.11.1 Grey-box based Data Evasion 43

6.11.2 White box evasion attack 43

6.11.3 Black box evasion attack 44

6.12 AI Energy-Latency Attack 44

6.12.1 Model Extraction 44

6.12.2 Data Inference 45

6.12.3 Denial of Service via Side Channels 45

7 Security Controls 46

8 Risk Assessment 50

8.1 Introduction 50

8.2 Input Manipulation Attack (ML01:2023) 50

8.3 Data poisoning attacks (ML02:2023) 51

8.4 Model Inversion Attack (ML03:2023) 51

8.5 Membership Inference Attack (ML04:2023) 52

8.6 Model Stealing (ML05:2023) 52

8.7 AI Supply Chain Attacks (ML06:2023) 53

8.8 Transfer Learning Attack (ML07:2023) 54

8.9 Model Skewing (ML08:2023) 54

8.10 Output Integrity Attack (ML09:2023) 55

8.11 Model Poisoning (ML10:2023) 55

8.12 Evasion attacks 55

8.13 AI Energy-Latency Attack 56

9 Primary Issues 58

10 Recommendations 59

10.1 General 59

10.2 Managing Risk 59

Annex: Change history/Change request (history) 62

**List of Tables**

[Table 4.1‑1 O-RAN Elements that Contain ML Models by Deployment Scenario 15](#_Toc184043472)

[Table 5.3‑1 ETSI Identified ML Threats by O-RAN AI/ML General Procedure Component 21](#_Toc184043473)

[Table 5.3‑2 ML Attacks Mapped to O-RAN AI/ML Hosts by Deployment Scenario 22](#_Toc184043474)

[Table 8.2‑1 AI/ML Risk Analysis – Input Manipulation Attack 50](#_Toc184043475)

[Table 8.3‑1 AI/ML Risk Analysis - Data poisoning attacks 51](#_Toc184043476)

[Table 8.4‑1 AI/ML Risk Analysis – Model Inversion Attack 51](#_Toc184043477)

[Table 8.5‑1 AI/ML Risk Analysis - Membership Inference Attacks 52](#_Toc184043478)

[Table 8.6‑1 AI/ML Risk Analysis – Model Stealing 52](#_Toc184043479)

[Table 8.7‑1 AI/ML Risk Analysis - AI Supply Chain Attacks 53](#_Toc184043480)

[Table 8.8‑1 AI/ML Risk Analysis - Transfer Learning Attack 54](#_Toc184043481)

[Table 8.9‑1 AI/ML Risk Analysis - Model Skewing Attacks 54](#_Toc184043482)

[Table 8.10‑1 AI/ML Risk Analysis – Output Integrity attacks 55](#_Toc184043483)

[Table 8.11‑1 AI/ML Risk Analysis – Model Poisoning Attacks 55](#_Toc184043484)

[Table 8.12‑1 AI/ML Risk Analysis - Evasion attacks 55](#_Toc184043485)

[Table 8.13‑1 AI/ML Risk Analysis - AI Energy-Latency Attack 56](#_Toc184043486)

[Table 10.2‑1. Mapping of AI/ML TR attacks with NIST AML Taxonomy of attacks 60](#_Toc184043487)

**List of Figures**

[Figure 4.1‑1: AI/ML General Procedure [Figure 4.1 in [i.14]] 13](#_Toc184043488)

[Figure 4.1‑2: Deployment scenario 1.1 -– AI/ML training and inference host locations [Figure 5.1 in [i.14]] 14](#_Toc184043489)

[Figure 4.1‑3: Deployment scenario 1.2 -– AI/ML training and inference host locations [Figure 5.2 in [i.14]] 14](#_Toc184043490)

[Figure 4.1‑4: Deployment scenario 1.3 - AI/ML training and inference host locations [Figure 5.3 in [i.14]] 15](#_Toc184043491)

[Figure 4.1‑5: Deployment scenario 1.4 – AI/ML training and inference host locations [Figure 5.4 in [i.14]] 15](#_Toc184043492)

# Foreword

This Technical Report (TR) has been produced by 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.

# Scope

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

version xx.yy.zz

where:

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

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

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

This technical report provides the threat model and risk assessment for AI/ML in O-RAN. The report identifies threats and risks and recommends potential security controls to protect against those threats through safeguards or mitigations.

The steps of the threat modelling process are as follows:

1. Identify assets: Identify the assets of AI/ML that must be protected.
2. Identify threats: Identify the threats that could adversely impact AI/ML and threats that can use AI/ML to adversely impact other components of O-RAN.
3. Identify the attack surface and attack vectors: Identify the points of attack in AI/ML where an attacker could:
   1. gain access to AI/ML assets.
   2. gain entry to another O-RAN system through AI/ML
   3. exploit a vulnerability or misconfiguration in AI/ML
   4. compromise AI/ML data or other AI/ML assets.
4. Measure risk: The extent to which confidentiality, integrity, or availability is threatened, based upon a risk-based analysis considering the impact level resulting from an attack and the likelihood of occurrence.
5. Recommend controls: The management, operational, and technical controls for an information system to protect the confidentiality, integrity and availability of AI/ML.

This Technical Report makes the following considerations:

1. The attack surface of the AI/ML includes its interfaces, functions, and data. Data-at-rest, Data-in-motion, and Data-in-use must be considered.
2. The O-RAN Alliance is pursuing a zero-trust architecture (ZTA) for its specifications based upon NIST SP 800-207 [i.8]. This will affect the risk scoring and selection of security controls for internal and external threats.
3. Security controls are recommended for specifications of AI/ML. The recommended controls provided in this report will be shared with the responsible O-RAN Alliance working group, such as WG1, WG2, WG3, WG10, and nGRG so that the appropriate specifications relevant to the recommendation can be updated.

# References

## Informative references

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

1. O-RAN ALLIANCE TS: “O-RAN Architecture Description”
2. O-RAN ALLIANCE TS: “O-RAN Security Requirements and Controls Specifications”
3. O-RAN ALLIANCE TS: “O-RAN Security Protocols Specifications”
4. O-RAN ALLIANCE TR: “O-RAN Security Threat Modeling and Risk Assessment”
5. O-RAN ALLIANCE TS: “O-RAN Security Test Specification”
6. O-RAN ALLIANCE TS: “Non-RT RIC Architecture”
7. Edan Habler et al: “Adversarial Machine Learning (AML) Threat Analysis and Remediation in Open Radio Access Network (O-RAN)”, <https://arxiv.org/abs/2201.06093>
8. NIST Special Publication 800-207 “Zero Trust Architecture”, August 2020, https://doi.org/10.6028/NIST.SP.800-207.
9. NIST AI 100-2e2023 Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations, draft white paper, US DoC NIST January 2024
10. NIST AI 100-1: Artificial Intelligence Risk Management Framework (AI RMF)”, January 2023. https://doi.org/10.6028/NIST.AI.100-1
11. US National Security Agency (NSA) / Cybersecurity and Infrastructure Security Agency (CISA), Security Guidance for 5G Cloud Infrastructures, Part I, Oct 28, 2021, Part II, Nov 18, 2021, Part III, Dec 2, 2021, Part IV, Dec 16, 2021. <https://www.nsa.gov/Press-Room/Cybersecurity-Advisories-Guidance/smdpage11747/2/>, last visited April 24, 2023.
12. ISO/IEC 27001: 2013 Information Security Management System (ISMS).
13. MITRE ATLAS, MITRE, [MITRE | ATLAS™](https://atlas.mitre.org/), last visited April 24, 2023.
14. O-RAN ALLIANCE TR: “AI/ML workflow description and requirements”
15. Threat Intelligence Report 2023: <https://www.nokia.com/networks/security-portfolio/threat-intelligence-report/>
16. Florian Tramèr et al: “Stealing Machine Learning Models via Prediction APIs.”, <https://arxiv.org/abs/1609.02943>
17. Matt Fredrikson et al: “Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures“, CCS '15: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications SecurityOctober 2015Pages 1322–1333, <https://doi.org/10.1145/2810103.2813677>
18. Shokri, Reza, Marco Stronati, and Congzheng Song. "Membership inference attacks against machine learning models.", proceedings of the IEEE Symposium on Security and Privacy, 2017, <https://doi.org/10.48550/arXiv.1610.05820>
19. ENISA: “Securing Machine Learning Algorithms”; <https://www.enisa.europa.eu/publications/securing-machine-learning-algorithms>
20. OWASP Top 10 Machine Learning Security risks, 2023 <https://owasp.org/www-project-machine-learning-security-top-10/>
21. 3GPP TR 21.905: “Vocabulary for 3GPP Specifications”
22. O-RAN ALLIANCE TR: “Study on Security for O-Cloud”
23. O-RAN ALLIANCE TS: “E2 General Aspects and Principles (E2GAP)”
24. OWASP Top 10 Proactive Controls, 2021, <https://owasp.org/projects/spotlight/historical/2021.02.10/>
25. Center for Internet Security (CIS) Critical Security Controls, <https://www.cisecurity.org/controls/cis-controls-list>
26. Cloud Security Alliance (CSA) Cloud Control Matrix (CCM), <https://cloudsecurityalliance.org/research/cloud-controls-matrix/>
27. ISO/IEC 27001: 2013 Information Security Management System (ISMS)
28. NIST SP 800-53r5, Security and Privacy Controls for Information Systems and Organizations, 2020.
29. US National Security Agency (NSA) / Cybersecurity and Infrastructure Security Agency (CISA), Security Guidance for 5G Cloud Infrastructures, Part I, Oct 28, 2021, Part II, Nov 18, 2021, Part III, Dec 2, 2021, Part IV, Dec 16, 2021. <https://www.nsa.gov/Press-Room/Cybersecurity-Advisories-Guidance/smdpage11747/2/> (as of Feb 28, 2022).
30. Principal Component Analysis, Wikipedia, <https://en.wikipedia.org/wiki/Principal_component_analysis>
31. Linear discriminant analysis for dimensionality reduction: <https://rasbt.github.io/mlxtend/user_guide/feature_extraction/LinearDiscriminantAnalysis/>
32. Machine Learning Standardization (Z-Score Normalization) with Mathematics, Towards AI; <https://towardsai.net/p/machine-learning/machine-learning-standardization-z-score-normalization-with-mathematics>
33. Voting Classifiers in Machine Learning: <https://medium.com/@imamitsingh/voting-classifiers-in-machine-learning-a532935fe592>
34. Stacking in Machine Learning: <https://www.javatpoint.com/stacking-in-machine-learning>
35. LIME: <https://github.com/marcotcr/lime>
36. SHAP: <https://shap.readthedocs.io/en/latest/index.html>
37. NIST blog: “How to deploy machine learning with differential privacy”; <https://www.nist.gov/blogs/cybersecurity-insights/how-deploy-machine-learning-differential-privacy>
38. Peter Darveau P. Eng., „Using Enclaves and Model Splitting to Improve AI Application Security in High Performance Computing – Thinking Beyond the Network“; <https://www.linkedin.com/pulse/using-enclaves-model-splitting-improve-ai-application-peter/>
39. Andrew Marshall, Jugal Parikh, Emre Kiciman and Ram Shankar Siva Kumar: “Threat Modeling AI/ML Systems and Dependencies”; <https://learn.microsoft.com/en-us/security/engineering/threat-modeling-aiml>
40. BSI: “AI SECURITY CONCERNS IN A NUTSHELL”; <https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/Practical_Al-Security_Guide_2023.pdf?__blob=publicationFile&v=5>
41. ENISA: “Artificial Intelligence Cybersecurity Challenges”; <https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges>
42. TOWARDS AI: “Machine Learning Standardization (Z-Score Normalization) with Mathematics”; <https://towardsai.net/p/machine-learning/machine-learning-standardization-z-score-normalization-with-mathematics>
43. Xinyu Tang et al, “Mitigating Membership Inference Attacks by Self-Distillation Through a Novel Ensemble Architecture”; <https://www.usenix.org/system/files/sec22fall_tang.pdf>
44. Ismat Jarin, Birhanu Eshete, “PRICURE: Privacy-Preserving Collaborative Inference in a Multi-Party Setting“; <https://arxiv.org/pdf/2102.09751.pdf>
45. BSI: Security of AI-Systems: Fundamentals: Adversarial Deep Learning; <https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/Security-of-AI-systems_fundamentals.pdf?__blob=publicationFile&v=4>
46. Pramoditha, R.: “Parameters Vs Hyperparameters: What is the difference?”, Medium web article, April 7, 2022. Retrieved from <https://medium.com/data-science-365/parameters-vs-hyperparameters-what-is-the-difference-5f40e16e2e82>
47. O-RAN ALLIANCE TR: “Use Cases Analysis Report”.
48. Casey, Cindy: “AI Poisoning Attacks”, lecture slides, 2022, Retrieved from <https://www.bucks.edu/media/bcccmedialibrary/con-ed/itacademy/fos2022/Casey-AI-Poisoning-pdf.pdf>
49. ETSI GR SAI 004 v1.1.1: “Securing Artificial Intelligence (SAI); Problem Statement”, December 2020.
50. Output Integrity Attack, <https://owasp.org/www-project-machine-learning-security-top-10/docs/ML09_2023-Output_Integrity_Attack>
51. HONGSHENG HU et al: “Membership Inference Attacks on Machine Learning: A Survey”, ACM Computing Surveys, Vol. 54, No. 11s, Article 235. Publication date: September 2022, <https://dl.acm.org/doi/pdf/10.1145/3523273>
52. Jiayuan Ye et al: “Enhanced Membership Inference Attacks against Machine Learning Models”, <https://arxiv.org/pdf/2111.09679.pdf>
53. Trent, Rod. “Must Learn AI Security Part 6: Model Inversion Attacks Against AI.”, <https://rodtrent.substack.com/p/must-learn-ai-security-part-6-model>
54. Tramèr et al, “The Space of Transferable Adversarial Examples”, arXiv:1704.03453, <https://arxiv.org/abs/1704.03453>
55. Mohamad Fazelnia et al: “Supporting Artificial Intelligence/Machine Learning Security Workers Through an Adversarial Techniques, Tools, and Common Knowledge Framework”; IEEE Security & Privacy (Volume: 21, Issue: 1, Jan.-Feb. 2023); DOI: 10.1109/MSEC.2022.3221058
56. NIST Privacy Framework; <https://www.nist.gov/privacy-framework>
57. General Data Protection Regulation, Regulation (EU) 2016/679, version OJ L 119, 04.05.2016; cor. OJ L 127, 23.5.2018, Art. 25: “Data protection by design and by default “; <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679>
58. ISO/IEC WD 27091, Cybersecurity and Privacy, Artificial Intelligence, Privacy protection: <https://www.iso.org/standard/56582.html>
59. ITU-T TR XSTR-SEC-AI: “Guidelines for security management of using artificial intelligence technology”; <https://www.itu.int/dms_pub/itu-t/opb/tut/T-TUT-ICTS-2022-PDF-E.pdf>
60. Yangsong Gao, et al, “STRIP: A Defence Against Trojan Attacks on Deep Neural Networks”, January 17, 2020, <https://arxiv.org/pdf/1902.06531.pdf>.
61. Sponge Examples: Energy-Latency Attacks on Neural Networks; <https://arxiv.org/abs/2006.03463>
62. I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators; <https://arxiv.org/pdf/1803.05847.pdf>
63. Stealing Neural Networks via Timing Side Channels; <https://arxiv.org/pdf/1812.11720.pdf>
64. Xiaobei Yan et al, Defense against ML-based Power Side-channel Attacks on DNN Accelerators with Adversarial Attacks; <https://arxiv.org/html/2312.04035v1>

# Definition of terms, symbols and abbreviations

## Terms

This document uses the verbal forms for the expression of provisions as defined in Study on Security for O-Cloud [i.22].

This document uses the term Zero Trust Architecture (ZTA) as defined by US NIST in [i.8], and applied to 5G cloud deployments by US CISA in [i.9].

A ZTA provides protection from external and internal threats, assuming the following:

1. there is no implicit trust granted to an asset based upon ownership, physical location, or network location [i.8]
2. the adversary is already inside the network. Perimeter defences are no longer sufficient to secure a network, and there should always be an assumption that a threat actor has established a foothold in the network [i.9]

This document uses the term “attack surface” defined by US NIST as

*The set of points on the boundary of a system, a system element, or an environment where an attacker can try to enter, cause an effect on, or extract data from, that system, system element, or environment* [https://csrc.nist.gov/glossary/term/attack\_surface]

This document refers to “sensitive information” defined by US NIST as

*information whose loss, misuse, or unauthorized access or modification could adversely affect security* [https://csrc.nist.gov/glossary/term/sensitive]

For the purposes of the present document, the terms and definitions provided in [i.1]apply:

**Data Wrangling:** The process of converting raw data into a usable form. Data wrangling is called data preparation in [i.14].

**Denial of Features Attack**: refers to a type of attack where an attacker, with full or partial knowledge of feature weightings or saliency measures in unsupervised learning systems, manipulates feature values to render salient features unavailable. This is achieved by reversing the weights, thereby disrupting the system's functionality [i.49].

**Hyperparameter:** In machine learning, hyperparameters are variables that specify how a model is trained and are external to the model. These values are set manually and remained fixed during the model training process. They are independent from the training data [i.46].

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

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

**Non-RT RIC Framework:** That functionality internal to the SMO that logically terminates the A1 interface to the Near-RT RIC and exposes to rApps, via its R1 interface, the set of internal SMO services needed for their runtime processing. The Non-RT RIC Framework functionality within the Non-RT RIC provides AI/ML workflow including model training, inference and updates needed for rApps.

**Parameters:** In machine learning, parameters are variables that the model uses to make predictions and are internal to the model. The values are set through an interaction of the training algorithm and training data. The parameters are dependent on the data they are trained with [i.46].

**SMO:** Service Management and Orchestration system. For more details please refer to Clause 5.3.1 of [i.1].

**xApp**: An application designed to run on the Near-RT RIC. Such an application is likely to consist of one or more microservices and at the point of on-boarding will identify which data it consumes and which data it provides. The application is independent of the Near-RT RIC and may be provided by any third party. The E2 enables a direct association between the xApp and the RAN functionality [i.23].

## Symbols

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

## Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [i.21], and the following apply:

AI Artificial Intelligence

AML Adversarial Machine Learning

DAR Data at Rest

DIM Data in Motion

DIU Data in Use

LDA Linear discriminant analysis, [i.31]

LIME Line Modelling Engine

ML Machine Learning

MNO Mobile Network Operator

NF Network Function

PCA Principal Component Analysis, [i.30]

RIC RAN Intelligent Controller

RT Real-Time

SHAP SHapley Additive exPlanations

SMOService Management and Orchestration

SMPC Secure Multi-Party Computation

TEE Trusted Execution Environment

ZTA Zero Trust Architecture

# AI/ML Assets

## Introduction

Artificial Intelligence and Machine Learning (AI/ML) are core to optimizing the operation of O-RAN. The primary goal of the Non-RT RIC is to support intelligent RAN optimization by providing policy-based guidance, ML model management and enrichment information to the Near-RT RIC function. It can also perform intelligent radio resource management function within a non-real-time interval (i.e., greater than 1 second). The Non-RT RIC can use data analytics and AI/ML training/inference to determine the RAN optimization actions for which it can leverage SMO services such as data collection and provisioning services of the O-RAN nodes as well as the O1 and O2 interfaces. The AI/ML capabilities in the Non-RT RIC are provided by rApps. Similarly, the near-Real-Time RAN Intelligent Controller may use AI/ML to provide near-real-time control and optimization of RAN elements and resources via fine-grained data collection and actions over E2 interface. The Near-RT RIC capabilities are provided by xApps [i.1].

The affected assets are derived from the AI/ML General Lifecycle Procedure and Interface Framework, and deployment scenarios in [i.14].

图示

描述已自动生成

Figure 4.1‑1: AI/ML General Procedure [Figure 4.1 in [i.14]]

The assets are also derived from the figures describing the deployment models in Clause 5 of [i.14]. These figures are included below.

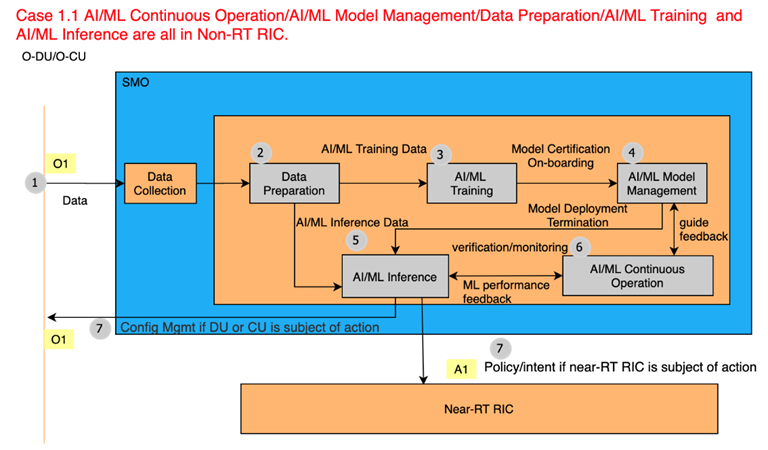


Figure 4.1‑2: Deployment scenario 1.1 -– AI/ML training and inference host locations [Figure 5.1 in [i.14]]

图示

描述已自动生成

Figure 4.1‑3: Deployment scenario 1.2 -– AI/ML training and inference host locations [Figure 5.2 in [i.14]]

图示, 日程表

描述已自动生成

Figure 4.1‑4: Deployment scenario 1.3 - AI/ML training and inference host locations [Figure 5.3 in [i.14]]

A picture containing text, screenshot, diagram, font

Description automatically generated

Figure 4.1‑5: Deployment scenario 1.4 – AI/ML training and inference host locations [Figure 5.4 in [i.14]]

Successful exploitation of threats against the AI/ML or intentionally malicious x/rApps used by the non- and Near-RT RICs can impact the O-RAN assets listed in the clauses below. Where an asset name is listed is an indication that this asset has been identified in the O-RAN Security Threat Modelling and Risk Assessment Technical Report [i.4].

The following table further identifies O-RAN elements that host models per deployment scenarios.

Table 4.1‑1 O-RAN Elements that Contain ML Models by Deployment Scenario

|  |  |  |  |
| --- | --- | --- | --- |
| **Deployment**  **Scenario** | **Initial Models** | **Trained Models** | |
| **AI/ML Training Host** | **AI/ML Inference Host** | **AI/ML Model Management Host** |
| 1.1 | Non-RT RIC | Non-RT RIC | Non-RT RIC |
| 1.2 | Non-RT RIC | Near-RT RIC | Out of Non-RT RIC |
| 1.3 | Out of Non-RT RIC | Non-RT RIC | Out of Non-RT RIC |
| 1.4 | Offline training: SMO/Non-RT RIC Online learning: Near-RT RIC | Near-RT RIC | Non-RT RIC and  Near-RT RIC |
| 1.5 | Non-RT RIC | O-CU / O-DU | Non-RT RIC |

NOTE: from [i.14]the phrase “Out of Non-RT RIC” appears to indicate it could either reside in the SMO but not in the Non-RT RIC or resides outside the SMO.

## Functions

* [ASSET-C-01]: Logical module: Service Management and Orchestration (SMO).
* [ASSET-C-02]: Near-RT RIC software.
* [ASSET-C-03]: O-CU-CP software.
* [ASSET-C-04]: O-CU-UP software.
* [ASSET-C-05]: O-DU software.
* [ASSET-C-06]: O-RU software.
* [ASSET-C-07]: O-eNB.
* [ASSET-C-09]: xApp.
* [ASSET-C-10]: rApp.
* [ASSET-C-11]: Non-RT RIC software.
* [ASSET-C-12]: ML components deploying machine learning such as: ML training and interference hosts, ML applications (xApps, rApps).
* [ASSET-C-13]: A1 termination.
* [ASSET-C-15]: R1 termination.
* [ASSET-C-17]: SMO Framework/Platform.
* [ASSET-C-18]: SMO Functions.
* [ASSET-C-19]: R1 Service Exposure Functions.
* [ASSET-C-20]: A1 functions.
* [ASSET-C-29]: AAL software including software, libraries, drivers, etc.
* [ASSET-C-30]: The hardware accelerator device firmware
* [ASSET-C-41]: E2 functions.
* [ASSET-C-40]: E2 termination.
* [ASSET-C-42]: Y1 termination.
* [ASSET-C-43]: Y1 functions.
* [ASSET-C-24]: M-plane.

## Interfaces

* [ASSET-C-14]: A1 interface, including protocol stack.
* [ASSET-C-16]: R1 interface, including protocol stack.
* [ASSET-C-22]: O1, including protocol stack.
* [ASSET-C-24]: OFH M-Plane, including protocol stack.
* [ASSET-C-26]: External interfaces.
* [ASSET-C-27]: External interfaces termination at SMO Framework/Platform.
* [ASSET-C-28]: External interfaces termination at Non-RT RIC Framework.
* [ASSET-C-40]: E2 interface, including protocol stack.
* [ASSET-C-43]: Y1 interface, including protocol stack.

## Data

* [ASSET-D-07]: A1 policies that are provided to the Near-RT RIC over the A1 interface to guide the RAN performance towards the overall goal expressed in RAN Intent. The A1 policies are declarative policies that contain statements on policy objectives and policy resources applicable to UEs and cells. A1 policies are created, modified and deleted by the Non-RT RIC.
* [ASSET-D-08]: A1 Enrichment Information that is collected or derived at SMO/Non-RT RIC either from non-network data sources or from network functions themselves and provided over the A1 interface to be utilized by Near-RT RIC, e.g. an ML model, to improve its performance. Discovery and request of A1 Enrichment Information from Near-RT RIC to Non-RT RIC. External Enrichment Information that is provided by an O-RAN external information source to Near-RT RIC over A1.
* [ASSET-D-09]: Data transported over the O1 interface such as the persistent configuration used by the Near-RT RIC to control the RAN, xApp-related messages, policies used by the Near-RT RIC to monitor, suspend/stop, override or control the behaviour of E2 node, Near-RT RIC services messages (REPORT, INSERT, CONTROL and POLICY), interface management messages (E2 Setup, E2 Reset, E2 Node Configuration Update, Reporting of General Error Situations), Near-RT RIC Service Update messages.
* [ASSET-D-10]: Database holding data from xApp applications and E2 Node.
* [ASSET-D-11]: E2 Node data (e.g. configuration information (cell configuration, supported slices, PLMNs, etc.), network measurements, context information, etc.)
* [ASSET-D-25]: Training or test data and associated labels: data sets collected externally or internally from the Near-RT RIC, O-CU and O-DU and passed to the ML training hosts in a ML system.
* [ASSET-D-29]: Security event log files generated by Non-RT RIC, rApps, Near-RT RIC, and xApps.

## Models

* [ASSET-D-26]: The trained ML model which includes the configured hyperparameters, inference algorithm, and learned parameters.
* [ASSET-D-27]: The ML prediction results built into the model (e.g. expected outcomes).
* [ASSET-D-40]: ML models that have not been trained yet, i.e., Initial Models and their associated learning algorithm.

## Pipeline

* [ASSET-D-28]: The behaviour of the ML system including tasks for data collection, data wrangling, pipeline management, model retraining, and model deployment.

# Threats

## Threat Model

For identifying threats, we are using STRIDE:

1. S - Spoofing identity. An application or program can masquerade as another to gain advantages not typically allowed for that program.
2. T - Tampering with data. This involves the malicious modification of data, including making unauthorized changes to a database and alteration of data as it flows between computers.
3. R - Repudiation. A user or program refuses the authenticity of a good or reasonable command or action.
4. I - Information disclosure. This involves the exposure of information to individuals with unauthorized access to it. For example, users gain the ability to read a file that they normally would not have been granted access to, or an intruder can read data in transit between computers.
5. D - Denial of service. These attacks deny service to valid users, such as making a website unavailable or unusable by flooding it with illegitimate requests to keep legitimate users without access.
6. E - Elevation of privileges. An unauthorized user gains privileged rights to access previously no granted to compromise or destroy the system, such as a change in membership.

|  |  |
| --- | --- |
| **Threat types** | **Impact types** |
| Spoofing | Authenticity |
| Tampering | Integrity |
| Repudiation | Non-repudiation |
| Information disclosure | Confidentiality |
| Denial of Service | Availability |
| Elevation of Privilege | Authorization |

## Threat Template

Template to present the threat characteristics:

|  |  |
| --- | --- |
| **Threat ID** |  |
| **Threat title** |  |
| **Threat description** |  |
| **Threat type** | Spoofing  Tampering  Repudiation  Information disclosure  Denial of Service  Elevation of Privilege |
| **Vulnerabilities** |  |
| **Impact type** | Authenticity  Integrity  Non-repudiation  Confidentiality  Availability  Authorization |
| **Affected Assets** |  |

## Potential Threats and Exploits

### Common Threats associated with ML

The most common threats associated with machine learning are identified and summarized in the Top ten list by the Open Worldwide Application Security Project (OWASP) [i.20].

The OWASP Top 10 Security Risks of Machine Learning systems updated in 2023 includes:

ML01:2023 Input Manipulation Attack

ML02:2023 Data Poisoning Attack

ML03:2023 Model Inversion Attack

ML04:2023 Membership Inference Attack

ML05:2023 Model Stealing

ML06:2023 AI Supply Chain Attacks

ML07:2023 Transfer Learning Attack

ML08:2023 Model Skewing

ML09:2023 Output Integrity Attack

ML10:2023 Model Poisoning

The AI/ML Threat analysis considers the threats listed above that are relevant to O-RAN AI/ML.

When considering threats to machine learning in O-RAN it is necessary to consider where the threats might occur in the ML components as described in the AI/ML General Procedure (see Figure 1 and Figure 4-1 of [i.14]). In support of that we referenced the ETSI Group Report on “Securing Artificial Intelligence (SAI); Problem Statement” [i.49]. This ETSI document contains a table, labelled Table 1, that maps the AI/ML lifecycle phases to relevant attacks and what security issue they might cause. The attacks mentioned in the ETSI document are not directly referenced in the table, instead, they are referred to specific clauses. The issues identified by ETSI appear to map directly to the impact types of Confidentiality, Integrity, and Availability as outlined in the STRIDE model. [i.49]does not address the impact types of Authenticity, Non-Repudiation and Authorization.

Drawing on the ETSI document, we have developed Table 5.3‑1. This table maps the defined ETSI ML Lifecycle phases to the O-RAN AI/ML General Procedure Components as defined in [i.14]. Through this mapping, we can determine the relevance of the attacks and potential impacts identified by ETSI for each AI/ML component defined by O-RAN. In addition, the OWASP ML Security Top Ten Risks have been mapped to the equivalent ETSI threats in the OWASP ML SEC Top 10 column. It is worth noting that the Train phase and Upgrades phase of the ETSI Lifecycle are both mapped to the same O-RAN AI/ML General Procedure component, ML Training Host. This was done since the O-RAN training host encompasses both ETSI ML Lifecycle phases per [i.14].

The ETSI Lifecycle phases of Model Design and Software Build have been listed as “out of scope for O-RAN” as those activities will occur outside of the O-RAN architecture. ETSI defines an attack referred in the table as “untrusted devices”. ETSI calls this out as a model confidentiality issue if the model resides on a device that is not trusted. There is no equivalent OWASP attack and has been indicated in the table with the text “No Mapping”. Two OWASP attacks, ML05:2023 Model Theft and ML08:2023 Model Skewing have no equivalent mapping to ETSI relevant attacks.

Table 5.3‑1 ETSI Identified ML Threats by O-RAN AI/ML General Procedure Component

| **ETSI ML Lifecycle Phase [i.49]** | **O-RAN AI/ML General Procedure Component [i.14]** | **Issues [i.49]**  **(Potential Impacts)** | **ETSI Relevant Attacks [i.49]** | **OWASP ML SEC Top 10 [i.20]** |
| --- | --- | --- | --- | --- |
| Data Acquisition | **Data Preparation** | Integrity | Poisoning of input data | ML01:2023 Input Manipulation Attack |
| Data Curation | Integrity | Poisoning on data labelling for supervised learning | ML02:2023 Data Poisoning Attack |
| Model Design | **Out of Scope for O-RAN** | | | |
| Software Build | **Out of Scope for O-RAN** | | | |
| Train | **ML Training Host**  **(AI/ML Training)** | Confidentiality | Training data leaked  Full Knowledge Attacks  Partial Knowledge Attacks  Zero Knowledge Attacks | ML02:2023 Data Poisoning Attack |
| Integrity | Transfer Learning Attack  Backdoor Attack  Modify Leaning Algorithm | ML07:2023 Transfer Learning Attack  ML06:2023 AI Supply Chain Attacks ML10:2023 Model Poisoning |
| Availability | Poisoning Attack at Input Layer  Denial of Features Attack | ML01:2023 Input Manipulation Attack  ML10:2023 Model Poisoning |
| Test | Availability | Lack of testing for adversarial examples | ML01:2023 Input Manipulation Attack |
| Deployment | **Model Management**  **and**  **ML Inference Host** | Confidentiality | Backdoor  Membership Inference  Inversion Attacks  Untrusted Devices | ML06:2023 AI Supply Chain Attacks  ML04:2023 Membership Inference Attack  ML03:2023 Model Inversion Attack  No Mapping |
| Integrity | Input attack  Backdoor attack | ML01:2023 Input Manipulation Attack  ML06:2023: AI Supply Chain Attacks |
| Availability | Evasion Attacks | ML09:2023 Output Integrity Attack |
| Upgrades | **ML Training Host (Optimization / Refinement)** | Integrity | Backdoor attack during training phase then triggered by an update | ML06:2023 AI Supply Chain Attacks |
| Availability | Poisoning attack on model parameters | ML10:2023 Model Poisoning |

**Note for Table 5.3‑1:**

* Although ETSI does not explicitly list the impact types of Authenticity, Non-Repudiation and Authorization, Authenticity and Authorization are controls used to enforce confidentiality, while non-repudiation is a control used to enforce integrity.
* The ETSI attack “untrusted devices” does not map to OWASP ML Security Top Ten. This does not cause an issue as this is out of scope for this technical report.
* The ETSI AI/ML phase of Deployment encompasses both the Model Management and ML Inference components. This causes a 1-to-many mapping which causes a challenge when mapping OWASP attacks to O-RAN AI/ML hosts by deployment scenario (see Table 5.3‑2).
* ETSI AI/ML phases “Train” and “Upgrades” both map to the same O-RAN AI/ML General Procedure component, ML Training Host. This does not appear to cause an issue for the mapping in Table 5.3‑2.
* The OWASP attacks, ML05:2023 Model Theft and ML08:2023 Model Skewing have no equivalent mapping to ETSI relevant attacks nor impact types. These attacks where mapped to appropriate AI/ML hosts in Table 5.3‑2.

Using the mapping completed in Table 5.3‑1 the following table, Table 5.3‑2, was developed by referencing Table 2 in [i.14].

Table 5.3‑2 ML Attacks Mapped to O-RAN AI/ML Hosts by Deployment Scenario

|  |  | **Attacks by O-RAN AI/ML Host** | | | |
| --- | --- | --- | --- | --- | --- |
|  |  | Data Preparation | ML Training Host | ML Inference | Model Management |
| **O-RAN Elements by Deployment Scenario** | | Input Manipulation  Data Poisoning  Model Skewing | Data Poisoning Attack  Transfer Learning Attack  AI Supply Chain Attacks  Model Poisoning  Input Manipulation Attack  Model Theft | AI Supply Chain Attacks  Membership Inference Attack  Model Inversion Attack  Input Manipulation Attack  Output Integrity Attack  Model Theft  Model Poisoning | AI Supply Chain Attacks  Model Theft  Model Poisoning |
| 1 | Non-RT RIC | X | X | X | X |
| Near-RT RIC |  |  |  |  |
| O-CU/O-DU |  |  |  |  |
| Out of Non-RT RIC |  |  |  |  |
|  | | | | | |
| 2 | Non-RT RIC | X | X |  |  |
| Near-RT RIC | X |  | X |  |
| O-CU/O-DU |  |  |  |  |
| Out of Non-RT RIC |  |  |  | X |
|  | | | | | |
| 3 | Non-RT RIC |  |  | X |  |
| Near-RT RIC |  |  | X |  |
| O-CU/O-DU |  |  |  |  |
| Out of Non-RT RIC | X | X |  | X |
|  | | | | | |
| 4 | Non-RT RIC | X | X |  | X |
| Near-RT RIC | X | X | X | X |
| O-CU/O-DU |  |  |  |  |
| Out of Non-RT RIC |  |  |  |  |
|  | | | | | |
| 5 | Non-RT RIC | X | X |  | X |
| Near-RT RIC |  |  |  |  |
| O-CU/O-DU |  |  | X |  |
| Out of Non-RT RIC |  |  |  |  |

**Notes for Table 5.3‑2:**

* Model Poisoning was added to model management and ML inference host because the scope of what is included in a model poisoning attack was expanded in relation to [i.14].
* The ETSI defined attack “untrusted devices” is out of scope for this technical report.
* The OWASP Model Skewing and Model Theft attacks were mapped where appropriate.
* “Out of Non-RT RIC” appears to indicate it could either reside in the SMO but not in the Non-RT RIC or resides outside the SMO [i.14].

### Input Manipulation Attack (ML01:2023)

#### Introduction

OWASP defines an input manipulation attack as a type of attack in which an attacker deliberately alters input data to mislead the model. Input manipulation attacks include adversarial attacks [i.20]. In these types of attacks, the adversary gains control of the inputs to the development of the model, its training and testing data, labels, the model, the code, and can use queries of the model to obtain predictions in the form of labels and model confidences [i.9].

Input manipulation attacks can occur during the training stage, while the model is learning, or during the inference/deployment stage. Inference/deployment stage attacks are further broken down into two cases. In the first case, the adversary modifies testing samples to create adversarial examples that are similar to the original sample but alter the model predictions to the adversary’s choices. The second type of attack, the adversary uses query access to the ML model to obtain confidential data within the model [i.9].

#### Potential Exploits

AI/ML input manipulation attacks pose several security threats as documented in [i.9]:

**Manipulation of training data:** An adversary can take control of a subset of the training data by inserting or modifying training samples.

**Manipulation of the testing data:** An adversary may add perturbations to testing samples at model deployment time to generate adversarial examples or in backdoor poisoning attacks.

**Manipulation of the labels:** An adversary may intentionally mis-assign labels to training data.

**Manipulation of the queries:** An adversary may submit queries to the model and receive predictions (either labels or model confidences) and confidential data. This capability is used by black-box evasion attacks, ENERGY-LATENCY ATTACKS, and all privacy attacks.

### Data poisoning attacks (ML02:2023)

#### Introduction

As per Threat Intelligence Report 2023 [i.15], 60% of attacks in telecom mobile networks are linked to Internet of Things (IoT) bots scanning for vulnerable hosts to expand their botnets for use in distributed denial-of-service (DDoS) attacks.

With knowledge of inputs and outputs for different AI/ML based network optimization use-cases, hackers can use such bots to inject poisoned data which may be used for AI/ML training.

The poisoned data injected by the hacker can vary based on the hacker’s knowledge of the AI/ML models and algorithms used in the AI/ML training. Based on the hacker’s level of knowledge of the AI/ML models and algorithms used for AI/ML training, the poisoning attacks are as defined in the sub-clauses below.

#### Black box data poisoning attacks

For Black box data poisoning attacks, the attacker does not need to know any details of the AI/ML models and algorithms used for the AI/ML training, or the parameters being used. Knowledge of inputs and outputs of AI/ML model alone is sufficient for the attacker to create poisoned data and to launch the (Black box) data poisoning attacks.

Details of input data used for different use-cases are already provided in O-RAN specifications. An adversary can use these details to identify which input data is used for training AI/ML models for which use-cases. An attacker in possession of a compromised data-source (bot) can inject poisoned data that can impact the behaviour of the AI/ML model used in the O-RAN deployment.

#### Grey box data poisoning attacks

Training AI/ML models requires historic data as input, in some use-cases. Because of this, usually, training data is stored in a database. For training online ML models, streaming input data is used, which is not historic in nature. Also, for some use-cases, trained models can be stored in databases.

In Grey box data poisoning attacks, the attacker leverages the knowledge (primarily the vulnerabilities) of the AI/ML data store and attempts to determine the AI/ML model/algorithm used for the AI/ML training, or the parameters being used. For example, if the database security has vulnerabilities allowing SQL-injection kinds of query-based attacks, an adversary can use such vulnerabilities to interactively obtain information about AI/ML models/algorithms and their training data sets.

For online ML models, the attacker leverages the knowledge of the streaming data source, or the streaming protocol vulnerabilities, or the transport protocol vulnerabilities or the transport network vulnerabilities to manipulate the training data used for online training.

Knowledge of inputs and outputs of the AI/ML model along with the level of information about the AI/ML models/algorithms is used by attacker to create poisoned data and to launch the (Grey box) data poisoning attacks. Grey box data poisoning attacks poison the training data in a manner that would affect that particular algorithm or its parameters the most.

#### White box data poisoning attacks

In White box data poisoning attacks, the attacker has obtained enough information i.e. full knowledge about the AI/ML models and algorithms being used, the configurable parameters like thresholds, hyper-parameters, etc.

Knowledge of inputs and outputs of AI/ML model along with the full knowledge of the AI/ML models/algorithms is used by attacker to create poisoned data and to launch the (White box) data poisoning attacks. White box data poisoning attacks can be done in a manner which can modify the behaviour of AI/ML algorithm with slow poisoning and can make it very challenging to detect such attacks.

### Model Inversion Attack (ML03:2023)

#### Introduction

OWASP defines a Model Inversion attack as a type of attack in which an attacker reverse-engineers the model to extract information from it [i.20]. Model inversion attacks are also called data reconstruction attacks [i.9], because they allow an attacker to recover personal or other types of information from aggregate and/or training information. This type of attack poses a significant risk to the privacy of the data used for training the model. Model Inversion attacks are more likely to be successful in cases where the model is overfitted, as it may have memorized specific training examples [i.53].

Model inversion attacks can lead to privacy violations, identity theft, loss of trust, misuse of sensitive information, and damage to reputation.

Model Inversion attacks typically involve the following steps.

1. The attacker accesses the model through a public API, a stolen copy of the model, or other means of interacting with the model's predictions.
2. The attacker selects a target individual or data point whose information they want to reconstruct from the model.
3. The attacker starts with an initial input that could be random or based on some prior knowledge of the target domain.
4. The attacker inputs the generated data into the model and collects the model's predictions and confidence scores.
5. Using the information from the model's outputs, the attacker iteratively refines the input data to maximize the model's confidence in the target label or class. This process involves optimization techniques like gradient descent or genetic algorithms.
6. The attacker repeats steps 4 and 5 until the input converges to a close approximation of the target data point, or the confidence scores reach a certain threshold.
7. The attacker now has a data point that closely resembles the original training data point, effectively compromising the privacy of the target individual or data point.

#### Potential Exploits

Potential exploits of Model Inversion attacks are identified in [i.9][i.53].

* **Black-box Model Inversion attack:** In this type of attack, the attacker only has access to the model's input-output pairs, meaning they can input data and receive the corresponding predictions. They have no knowledge of the model's architecture, parameters, or the training data. The attacker generates inputs, analyses the model's outputs, and iteratively refines the inputs based on the obtained information. This process continues until the attacker is able to approximate the original training data.
* **White-box Model Inversion attack:** In a white-box attack, the attacker has access to the AI model, including its architecture, parameters (such as weights and biases), and possibly partial knowledge of the training data. This additional information allows the attacker to exploit the inner workings of the model more effectively and reconstruct the original training data with higher accuracy. In this scenario, the attacker can use gradient-based optimization techniques to maximize the model's confidence in the target label or class, ultimately converging to a close approximation of the target data point.

### Membership Inference Attack (ML04:2023)

#### Introduction

A membership inference attack (MIA) is a privacy attack where an attacker tries to determine whether a specific individual's data was used to train the model. The attacker will observe a model’s output for various inputs, including the specific individual data [i.18][i.51][i.52].

In the telecommunications domain typically telecom providers typically handle vast amounts of user data which could include their location information, call behaviour, data access behaviour etc., These data are quite sensitive and can reveal individual patterns.

#### Potential Exploits

There are many different types of membership inference attack (MIA), each exploiting different aspects of the target model which can reveal the membership information in unique ways. Some of the common types of MIA are:

a) MIA based on information access, i.e., how much information the attacker is aware of the model, its internals. The attack types are Black-box MIA, Gray-box MIA and White-box MIA.

b) MIA based on inference technique: i.e., by applying different analysis methods the attacker tries to infer input to output relation. Some of the attack types are Decision threshold analysis, Gradient-based analysis and Distance-based analysis. In Decision threshold analysis, the subtle changes in input’s relation to the model’s output confidence score is observed. In Gradient-based analysis techniques such as back propagation is used to analyse input to output variation. In Distance-based analysis the distance between model’s output for different inputs in multi-dimensional space is used.

c) MIA based on target: i.e., how the attacker would like to understand whether specific individual data or data of a group of individuals are involved in the training set. The attack types are known as Individual membership inference attack and Group membership inference attack, respectively.

Potential MIAs in case of Telecom domain are:

1) Identifying Specific Individuals: The attacker tries to determine if a specific user information is included in the training data. This attack can exploit model vulnerabilities to target specific individuals.

2) Group membership inference: The attacker tries to determine behaviour of individuals and group them based on available data. The attack can lead to determination of sensitive information that can then be used for potential discrimination or targeted attacks.

### Model Stealing (ML05:2023)

#### Introduction

AI/ML model stealing or extraction attacks refer to unauthorized attempts to steal or extract machine learning models. These attacks are driven by the value and intellectual property associated with trained models, which can be leveraged by malicious actors for various purposes such as replicating the model’s functionality, using it for financial gain, or performing adversarial attacks [i.16].

AI/ML model stealing/extraction attacks typically involve the following steps:

1. **Obtaining Access**: Attackers attempt to gain unauthorized access to the target environment where the model is deployed or stored. This can be done through various means such as exploiting vulnerabilities in the infrastructure, social engineering, insider threats, or targeting weak authentication mechanisms.
2. **Model Extraction**: Once access is obtained, attackers aim to extract the trained model from its storage or runtime environment. This can be achieved through techniques like reverse engineering, model inversion attacks [i.17], membership inference attacks [i.18], or by directly exfiltrating the model from the target system.
3. **Model Utilization**: After stealing the model, attackers can use it for several malicious purposes. For example, they may retrain or fine-tune the model on their own data, use it to understand the vulnerabilities and weaknesses of the target system, prepare adversarial examples, conduct adversarial attacks, or create replicas of the model for unauthorized distribution.

#### Potential exploits

AI/ML model stealing/extraction attacks pose several security threats:

* **Intellectual Property Theft**: Stolen models can lead to the theft of intellectual property, allowing attackers to replicate the model's functionality without incurring the associated development costs.
* **Competitive Advantage Erosion**: If a competitor gains unauthorized access to a trained model, it can undermine the original developer's competitive advantage by using the model for similar or competing purposes.
* **Data Privacy Violations**: The stolen model may contain sensitive information embedded within its learned parameters. This can result in privacy breaches, especially if the model was trained on sensitive or personal data.
* **Adversarial Attacks**: Attackers may exploit the extracted model to craft adversarial examples (to mount an evasion attack) or launch targeted attacks against the system that initially deployed the model, using obtained knowledge about possible vulnerabilities.

### AI Supply Chain Attacks (ML06:2023)

#### Introduction

AI supply chain attacks refer to malicious activities targeting the various stages of Artificial Intelligence (AI) system development and deployment. These attacks aim to manipulate data, models, software and hardware used in AI systems with the goal of introducing vulnerabilities that can be exploited later for unauthorized access, data exfiltration, or other malicious purposes.

#### Potential exploits

AI Supply Chain Attacks pose several security threats:

* **Data poisoning:** Adversaries may corrupt or manipulate training datasets to cause the AI model to learn incorrect associations or make biased predictions. This can lead to degraded performance, wrong decisions, or even backdoors in the final system.
* **Model tampering:** Attackers might inject malicious code or logic into the AI model itself during training or deployment. This could allow them to gain control over the target system, steal sensitive data, or force the AI system to produce incorrect results on demand.
* **Backdoor attacks (a.k.a. Trojan attacks):** In these attacks, a seemingly benign AI model is infected with malicious code that remains dormant until it receives a specific trigger. At this point, the attacker can remotely activate the backdoor and use it to compromise the target system or exfiltrate data.
* **Hardware tampering:** Malicious actors may also focus on manipulating the hardware used for AI processing, such as GPUs, ASICs, or FPGAs. By introducing backdoors or vulnerabilities in these components, attackers can compromise the confidentiality, integrity, and availability of AI systems even if the software and data are secure.
* **Insecure maintenance APIs:** Many AI services are exposed through APIs, for example for deployment and maintenance, that can be abused by attackers to launch supply chain attacks. If these APIs are not properly secured, they can be used to inject malicious code or exfiltrate sensitive information from the target system.
* **Third-party libraries and dependencies:** AI systems often rely on third-party libraries and open-source components which might contain vulnerabilities or backdoors that attackers can exploit. Insecure software supply chains increase the risk of such attacks.

### Transfer Learning Attack (ML07:2023)

#### Introduction

Transfer learning is a machine learning (ML) technique where a pre-trained model (a ‘source’ model) is used as a starting point for a new related task (a ‘target’ model), instead of training a model from scratch. This approach saves time, computational resources, and often leads to better performance. However, this ability to transfer knowledge can also be exploited by attackers through AI/ML Transfer Learning Attacks, leading to security threats ([i.54]).

#### Potential exploits

Transfer Learning Attacks pose several security threats as they allow attackers to mount attacks on target models with limited or no access to the original training data. They are mainly based on the assumption that attacks that work on the source model most likely also work on the target model. The main types of Transfer Learning Attacks are:

* **Data poisoning:** Attackers can poison the source model's training data by injecting malicious samples, which get propagated into the target model during transfer learning, causing misclassification or unintended behaviour in the latter.
* **Model stealing:** By using a series of carefully crafted queries to the source model, attackers can extract sensitive information about its architecture and parameters, which they can then use to build a similar model for malicious purposes.
* **Adversarial examples:** Attackers can create input samples that fool both the source and target models, even if these inputs are visually or semantically similar to benign ones. These adversarial examples can be used to bypass security systems or cause misclassifications in critical applications like self-driving cars or healthcare.
* **Backdoor attacks:** Attackers can embed hidden malicious functionality into the source model, which remains dormant until triggered by specific input patterns. When the trojans are transferred to the target model, they can cause unexpected behaviour or leak sensitive information.

### Model Skewing (ML08:2023)

#### Introduction

A model skewing AI/ML attack is a data poisoning attack, where an attacker manipulates the training data, specifically the distribution of the training data, used to build a machine learning model [i.20]. This manipulation aims to cause the model to behave in a way that benefits the attacker, potentially harming the model's accuracy, fairness, and reliability.

In model skewing attacks,

* Attackers typically inject malicious data. This data can be fake or manipulated real data, designed to influence the model's learning process to meet the attackers needs.
* Attackers aim to skew the model's predictions towards their desired outcome. For example, they make a bias to favour some action i.e., approvals, or to overcome some action i.e., bypass security rules.

In the Telecommunications networks machine learning models are used for tasks like:

* Network traffic prediction, which is to optimize network resources based on predicted traffic patterns.
* Quality of service (QoS) optimization, which is to ensure consistent and reliable network performance.
* Fraud detection, which is to identify and block fraudulent activities like unauthorized access or SIM swap scams.
* Customer churn, whereby predicting customer churn behaviour develop retention strategies.

Model skewing attacks in telecommunication networks can significantly impact the operation of networks causing service disruptions, reputational damage, and financial losses.

* The attacker could create a network flood with dummy traffic. This behaviour can skew the model to predict higher traffic than usual, leading to over-provisioning of resources and unnecessary costs.
* The attacker could manipulate network performance data, artificially degrading certain areas' performance metrics. This could prompt the model to optimize resources towards those areas unnecessarily.
* The attacker could manipulate labels of transactions, i.e., they could change "fraudulent" labels to "legitimate," allowing malicious activity to remain undetected.
* The attacker could alter predictions resulting in higher churn rates and this could cause the operators to invest unnecessarily in retention strategies for non-churning customers.

#### Potential Exploits

Model skewing attacks are performed primarily by exploiting weaknesses in the model's training process, such as lack of data validation or feedback mechanisms. There are many different types of model skewing attacks, based on attackers’ intention. However, they can be broadly categorized into the following two exploits.

1. Data manipulation-based Model skewing attacks: The attacker manipulates the data involved in the training of the model. The attacker modifies/injects the training data with modified data, or modifies the labels associated with the training data, resulting in learnt model behaviour to be changed. The attack can significantly impact the operation of networks causing service disruptions, reputational damage, and financial losses.
2. Feedback manipulation-based Model Skewing attacks: The attacker manipulates the feedback mechanisms used by the AI/ML for continuous improvement during deployment. The attack can significantly impact the operation of networks causing service disruptions, reputational damage, and financial losses.

### Output Integrity Attack (ML09:2023)

#### Introduction

Output integrity attack (see [i.50]) in the context of AI/ML model is to alter or manipulate the output of an ML model to attack the system that the model is being used in. The model output can be used by a system to implement critical logic/decisions, and a malicious actor can influence the output of the model to force incorrect decisions by the system. These incorrect decisions can lead to manipulation of other systems with which the current system interacts with.

From O-RAN point of view, the Non-RT RIC and Near-RT RIC use the AI/ML model to perform inference. The result of an inference using a model is an output which in the case of the Non-RT RIC is used to generate A1 policies and/or changing the configuration of O-CU and O-DU/O-RU using the O1 interface or the open fronthaul m-plane. Similarly, the Near-RT RIC uses the model inference output to generate E2 policies and configuration changes over the E2 interface.

AI/ML model output attacks typically involve the following steps:

1. **Obtaining Access**: Attackers attempt to gain unauthorized access to the target environment where the model output can be modified or altered. This can be done through various means such as exploiting vulnerabilities in the infrastructure, social engineering, insider threats, or targeting weak/misconfigured authentication/authorization mechanisms.
2. **Output manipulation**: After gaining access to the target environment, the attackers can manipulate the AI/ML model output in several ways. Some of the attack vectors are mentioned below.

|  |  |
| --- | --- |
| **Attack vector** | **Description** |
| Output in transit | This method involves modifying the output while it is in transit over a network. |
| Output storage | This method involves modifying the output stored in the model output storage. |
| Output layer of the model | This method involves changing the model output layer of the AI/ML model to achieve output manipulation. |
| Model inputs | This method does not involve a direct output manipulation but achieves the output change by modifying the training, inference, prediction inputs to the model. |
| Model hyperparameters | This method does not involve a direct output manipulation but achieves the output change by modifying the hyperparameters used for training the model. |

1. **Monitoring and recalibration**: Attackers monitor the effect of model output change and perform recalibration to achieve the desired result.

#### Potential exploits

AI/ML model output attacks pose several security threats:

* **Denial of Service**: The output of the AI/ML models is used to configure the O-RAN network functions and generate A1 policies for Near-RT RIC. A manipulated output of a model can cause service disruption of the O-RAN elements that are using configuration / policies derived from the output.
* **Subscriber QoE change:** The QoE that the subscribers experience depends on several configuration parameters of O-RAN elements and the radio conditions of the subscribers. AI/ML models are used to perform QoE optimization and QoE based traffic steering in O-RAN [i.14]. A carefully crafted manipulation of the output of AI/ML model can be used to improve or degrade the QoE of a subscriber or a group of subscribers.

### Model Poisoning (ML10:2023)

#### Introduction

Model poisoning attacks involve an adversary modifying an ML model’s internals to alter its predictions. Specifically, these changes could be made to the model parameters or to the code of the model algorithms. Unauthorized code changes to the model inference algorithm or alteration of the model parameters file pertain to trained models, while code changes to the model learning algorithm pertain to initial (untrained) models [i.20][i.19][i.48][i.49].

In literature this attack may also be referred to as logic corruption [i.48] and can sometimes be inclusive to the manipulation of training data [i.20]. This aspect overlaps with Input Manipulation and Data Poisoning attacks, covered in clauses 5.3.2 and 5.3.3 respectively. For this report, model poisoning will focus on the direct manipulation of model internals.

As an example, consider the O-RAN use case numbered 21 on Energy Saving [i.47]. This use case details AI/ML assist to automatically control the power to one or more carriers or a cell of a given technology to reduce power consumption for the O-CU/DU/RU. An adversary with direct access to the Non-RT RIC model could alter its parameters to embed a backdoor, which could later be triggered to power-off of specific carriers or cells.

In the O-RAN AI/ML lifecycle, trained models reside in the “AI/ML Inference” and “AI/ML Model” hosts while initial models reside in the “AI/ML Training” host referred to in Figure 4-1 of [i.14]. Depending on the deployment scenarios outlined in table 2 of [i.14], the AI/ML lifecycle hosts (and their models) mentioned above might reside in the O-CU, O-DU, Non-RT RIC, SMO, or Near-RT RIC (see Table 4-1).

#### Potential Exploits

AI/ML model poisoning attacks pose several security threats as documented in [i.9]:

**Manipulation of the model:** An adversary can take control of the model parameters by either generating a backdoor or Trojan trigger and inserting it in the model or by sending malicious local model updates in federated learning.

### Evasion attacks

#### Introduction

As per ENISA report 2021 [i.19], evasion attacks are widely prevalent in cases where AI/ML solutions are deployed.

With knowledge of inputs and outputs pairs during inference phase for different AI/ML use cases, attackers can work on the ML algorithm's inputs to find small perturbations leading to large modification of its outputs (e.g. decision errors).

#### White box evasion attacks

White box evasion attacks are based on gradient computation and the attacker has full knowledge of the AI/ML models and algorithms being used [i.9], the configurable parameters like thresholds, hyper-parameters, etc. The attacker manipulates an input sample in such a way that the classification loss is maximized. The specific manipulation is determined by calculating the gradients of the classification loss with respect to the input sample. For example, the attacker adds noise to the input sample, in the direction of the gradient of the classification loss with respect to the input sample.

#### Black box evasion attacks

Black-box evasion attacks are designed under a realistic adversarial model, in which the attacker has no prior knowledge of the model architecture or parameters. Instead, the adversary can interact with a trained ML model by querying it on various data samples and obtaining the model’s predictions. There are two main classes of black-box evasion attacks [i.9].

a) Decision based attacks: In these attacks, the adversary can only access the model's final decision or predicted label. Techniques include the Boundary Attack, which employs random walks along the decision boundary and rejection sampling, and the HopSkipJumpAttack, which enhances gradient estimation to minimize the number of queries. Other methods involve finding the direction of the nearest decision boundary (OPT attack), using sign SGD instead of binary searches (Sign-OPT attack), and applying Bayesian optimization.

b) Score-based attacks: In this scenario, attackers access the model’s confidence scores or logits and use various optimization techniques to generate adversarial examples. A common approach is zeroth-order optimization, which estimates the model’s gradients without directly computing derivatives. Other techniques include discrete optimization, natural evolution strategies, and random walks.

The attacker can also employ transfer-based evasion techniques, exploiting the potential that an input identified as an adversarial example for a surrogate white-box model could similarly be effective as an adversarial example for another black-box model.

### AI Energy-Latency Attack

#### Introduction

The AI/ML Energy-Latency Attack is a type of side-channel attack that targets machine learning systems by exploiting the correlation between energy consumption and computation latency in hardware. These attacks can be used to infer sensitive information about the model, its parameters, or the data it's processing by observing power usage and timing information. This type of attack is particularly relevant for systems where machine learning models are deployed on edge devices or in cloud environments where multiple users share hardware resources.

The fundamental concept behind energy-latency attacks is that different computations consume varying amounts of energy and take different lengths of time to complete. In the context of machine learning, the execution of different neural network layers or operations can lead to measurable differences in power consumption and processing time. An attacker with access to these measurements can potentially infer the model's architecture, the data being processed, or even reconstruct the input data, which can be a significant privacy concern ([i.62], [i.63]). In addition, an attacker may craft inputs that maximize energy consumption or computation time (so called ‘Sponge Examples’, [i.61]), leading to a denial of service by overloading the system or draining battery life.

#### Potential Exploits

**Model Extraction:** By monitoring the energy and latency patterns, an attacker might deduce the structure of the neural network, including the types of layers, their sequence, and possibly hyperparameters.

**Data Inference:** Observing the energy and latency signatures of different computations could allow an attacker to make inferences about the input data, potentially leading to privacy breaches, especially if the data is sensitive.

**Denial of Service:** An attacker could exploit these side channels to craft inputs that maximize energy consumption or computation time, leading to a denial of service by overloading the system or draining battery life.

# Threat Analysis

## Input Manipulation Attack (ML01:2023)

### Training Data Control

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-IM-1 |
| **Threat title** | Training Data Control |
| **Threat description** | An adversary takes control of a subset of the training data by inserting or modifying training samples. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Integrity |
| **Affected Assets** | Training or test data: data sets collected externally or internally from the Near-RT RIC, O-CU and O-DU and passed to the ML training hosts in a ML system. [ASSET-D-25]  The trained ML model [ASSET-D-26],  The ML prediction results built into the model [ASSET-D-27],  The behaviour of the ML system [ASSET-D-28]. |

### Model Control

void

### Testing Data Control

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-IM-2 |
| **Threat title** | Testing Data Control |
| **Threat description** | An adversary adds perturbations to testing samples at model deployment time to generate adversarial examples or in backdoor poisoning attacks |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Integrity |
| **Affected Assets** | Training or test data: data sets collected externally or internally from the Near-RT RIC, O-CU and O-DU and passed to the ML training hosts in a ML system. [ASSET-D-25]  The trained ML model [ASSET-D-26],  The ML prediction results built into the model [ASSET-D-27],  The behaviour of the ML system [ASSET-D-28]. |

### Label Manipulation

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-IM-3 |
| **Threat title** | Label Manipulation |
| **Threat description** | An adversary may intentionally mis-assign labels to training data. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Integrity |
| **Affected Assets** | Training or test data: data sets collected externally or internally from the Near-RT RIC, O-CU and O-DU and passed to the ML training hosts in a ML system. [ASSET-D-25]  The trained ML model [ASSET-D-26],  The ML prediction results built into the model [ASSET-D-27],  The behaviour of the ML system [ASSET-D-28]. |

### AI/ML Query Exploitation

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-IM-4 |
| **Threat title** | AI/ML Query Exploitation |
| **Threat description** | An adversary may submit queries to the model and receive predictions (either labels or model confidences) and confidential data. |
| **Threat type** | Information Disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in AI/ML APIs |
| **Impact type** | Confidentiality |
| **Affected Assets** | A1 Enrichment Information that is collected or derived at SMO/Non-RT RIC either from non-network data sources or from network functions themselves and provided over the A1 interface to be utilized by Near-RT RIC, e.g., an ML model, to improve its performance. [ASSET-D-08]  Training or test data: data sets collected externally or internally from the Near-RT RIC, O-CU and O-DU and passed to the ML training hosts in a ML system. [ASSET-D-25]  The trained ML model [ASSET-D-26],  The ML prediction results built into the model [ASSET-D-27],  The behaviour of the ML system [ASSET-D-28]. |

## Data poisoning attacks (ML02:2023)

### Black-box Data Poisoning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-DP-01 |
| **Threat title** | Black-box Data Poisoning |
| **Threat description** | An attacker in possession of a compromised data-source can inject poisoned data which can be consumed by AI/ML for training. For such an attack, the attacker does not need to know any details of the AI/ML algorithm, or the parameters being used. Knowledge of inputs and outputs of AI/ML is sufficient to launch such attacks. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Grey-box Data Poisoning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-DP-02 |
| **Threat title** | Grey-box Data Poisoning |
| **Threat description** | An attacker being able to query the database which stores the training data, and/or streaming data used for online ML models, and/or AI/ML models may be able to get more details about the AI/ML algorithms being used and parameters configured if any. This can help the attacker to launch an attack which can poison the training data in a manner that would affect that particular algorithm or its parameters the most. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in data transport, vulnerabilities in training database |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11],  O1 interface for streaming data [ASSET-C-22], A1 interface [ASSET-C-14], E2 interface for streaming data [ASSET-C-40] |

### White-box Data Poisoning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-DP-03 |
| **Threat title** | White-box Data Poisoning |
| **Threat description** | An attacker having full knowledge of the AI/ML algorithm being used, the configurable parameters like thresholds, hyper-parameters, etc. can launch a data poisoning attack in a manner which can modify the behaviour of AI/ML algorithm with slow poisoning and can make it very challenging to detect such attacks. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11], O1 interface for streaming data [ASSET-C-22], A1 interface [ASSET-C-14], E2 interface for streaming data [ASSET-C-40] |

## Model Inversion Attack (ML03:2023)

### Black-box Model Inversion

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MOI-01 |
| **Threat title** | Black-box Model Inversion |
| **Threat description** | An attacker has access to the model’s input-output pairs allowing them to input data and receive predictions. The attacker generates inputs, analyzes the model's outputs, and iteratively refines the inputs based on the obtained information. This process continues until the attacker is able to approximate the original training data. The attacker has no knowledge of the model's architecture, parameters, or the training data. |
| **Threat type** | Information Disclosure |
| **Vulnerabilities** | Model is overfitted, model may have memorized specific training examples, model API not properly protected |
| **Impact type** | Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10], Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26]  The ML prediction results built into the model [ASSET-D-27]  The behavior of the ML system including tasks for data collection, data wrangling, pipeline management, model retraining, and model deployment. [ASSET-D-28] |

### White-box Model Inversion

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MOI-02 |
| **Threat title** | White-box Model Inversion |
| **Threat description** | An attacker has access to the AI model, including its architecture, parameters (such as weights and biases), and possibly partial knowledge of the training data. This information allows the attacker to exploit the inner workings of the model more effectively and reconstruct the original training data with higher accuracy. In this scenario, the attacker can use gradient-based optimization techniques to maximize the model's confidence in the target label or class, ultimately converging to a close approximation of the target data point. |
| **Threat type** | Information Disclosure |
| **Vulnerabilities** | Model is not protected, training data is not protected, model is overfitted, model may have memorized specific training examples, model API not protected |
| **Impact type** | Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26]  The ML prediction results built into the model [ASSET-D-27]  The behaviour of the ML system including tasks for data collection, data wrangling, pipeline management, model retraining, and model deployment. [ASSET-D-28] |

## Membership Inference Attack (ML04:2023)

### Information Access – Membership Inference

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MEI-01 |
| **Threat title** | Information Access – Membership Inference |
| **Threat description** | An attacker possesses different levels of information on the model architecture, training parameters, training data, and relation between the output to the input. Based on the level of information access, the MIA can be Black-box MIA, or Gray-box MIA or White-box MIA. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Confidentiality |
| **Affected Assets** | Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Inference Technique – Membership Inference

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MEI-02 |
| **Threat title** | Inference Technique – Membership Inference |
| **Threat description** | An attacker by applying different analysis methods tries to infer input to output relation. Some of the attack types are Decision threshold analysis, Gradient-based analysis, Distance-based analysis. In Decision threshold analysis, the subtle changes in input’s relation to the model’s output confidence score is observed. In Gradient-based analysis techniques such as back propagation is used to analyse input to output variation. In Distance-based analysis the distance between model’s output for different inputs in multi-dimensional space is used. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Confidentiality |
| **Affected Assets** | Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Target based Membership Inference

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MEI-03 |
| **Threat title** | Target based- Membership Inference |
| **Threat description** | In a target-based membership attack the attacker would like to understand whether specific individual data or data of a group of individuals are involved in the training set. The attack types are known as Individual membership inference attack, Group membership inference attack, respectively. In Individual membership inference attack, the attacker tries to determine if a specific user information is included in the training data. In Group membership inference attack, the attacker tries to determine behaviour of individuals and group them based on available data. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Confidentiality |
| **Affected Assets** | Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

## Model Stealing (ML05:2023)

### Intellectual Property Theft

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-ME-01 |
| **Threat title** | Intellectual Property Theft |
| **Threat description** | Stolen models can lead to the theft of intellectual property, allowing attackers to replicate the model's functionality without incurring the associated development costs. If a competitor gains unauthorized access to a trained model, it can undermine the original developer's competitive advantage by using the model for similar or competing purposes. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Data Privacy Violations

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-ME-02 |
| **Threat title** | Data Privacy Violations |
| **Threat description** | The stolen model may contain sensitive information embedded within its learned parameters. This can result in privacy breaches, especially if the model was trained on sensitive or personal data. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Adversarial Attacks with Stolen Models

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-ME-03 |
| **Threat title** | Adversarial Attacks with Stolen Models |
| **Threat description** | Attackers may exploit the extracted model to craft adversarial examples (to mount an evasion attack) or launch targeted attacks against the system that initially deployed the model, using obtained knowledge about possible vulnerabilities. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

## AI Supply Chain Attacks (ML06:2023)

### Data Poisoning via Supply Chain

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-SC-01 |
| **Threat title** | Data Poisoning via Supply Chain |
| **Threat description** | Adversaries may corrupt or manipulate training datasets to cause the AI model to learn incorrect associations or make biased predictions. This can lead to degraded performance, wrong decisions, or even backdoors in the final system. |
| **Threat type** | Tampering, Denial of Service |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model transfer |
| **Impact type** | Integrity, Availability |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Model Tampering

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-SC-02 |
| **Threat title** | Model Tampering |
| **Threat description** | Attackers might inject malicious code or logic into the AI model itself during training or deployment. This could allow them to gain control over the target system, steal sensitive data, or force the AI system to produce incorrect results on demand. |
| **Threat type** | Tampering, Information disclosure |
| **Vulnerabilities** | Vulnerabilities in ML model transfer |
| **Impact type** | Integrity, Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Backdoor via Supply Chain

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-SC-03 |
| **Threat title** | Backdoor via Supply Chain |
| **Threat description** | In these attacks, a seemingly benign AI model is infected with malicious code that remains dormant until it receives a specific trigger. At this point, the attacker can remotely activate the backdoor and use it to compromise the target system or exfiltrate data. |
| **Threat type** | Tampering, Information disclosure |
| **Vulnerabilities** | Vulnerabilities in ML model transfer |
| **Impact type** | Integrity, Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25], Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02,  Non-RT-RIC SW [ASSET-C-11] |

### Hardware Tampering

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-SC-04 |
| **Threat title** | Hardware Tampering |
| **Threat description** | Malicious actors may also focus on manipulating the hardware used for AI processing, such as GPUs, ASICs, or FPGAs. By introducing backdoors or vulnerabilities in these components, attackers can compromise the confidentiality, integrity, and availability of AI systems even if the software and data are secure. |
| **Threat type** | Tampering, Information disclosure |
| **Vulnerabilities** | Vulnerabilities in the AI processing HW |
| **Impact type** | Integrity, Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (ML training and interference hosts, xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Insecure Maintenance APIs

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-SC-05 |
| **Threat title** | Insecure Maintenance APIs |
| **Threat description** | Many AI services are exposed through APIs, for example for deployment and maintenance, that can be abused by attackers to launch supply chain attacks. If these APIs are not properly secured, they can be used to inject malicious code or exfiltrate sensitive information from the target system. |
| **Threat type** | Tampering, Information disclosure |
| **Vulnerabilities** | Vulnerabilities in the ML maintenance APIs |
| **Impact type** | Integrity, Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Third-party Libraries and Dependencies

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-SC-06 |
| **Threat title** | Third-party Libraries and Dependencies |
| **Threat description** | AI systems often rely on third-party libraries and open-source components which might contain vulnerabilities or backdoors that attackers can exploit. Insecure software supply chains increase the risk of such attacks. |
| **Threat type** | Tampering, Information disclosure |
| **Vulnerabilities** | Vulnerabilities in the ML-related SW |
| **Impact type** | Integrity, Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

## Transfer Learning Attack (ML07:2023)

### Data Poisoning for Transfer Learning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-TL-01 |
| **Threat title** | Data Poisoning for Transfer Learning |
| **Threat description** | Attackers can poison the source model's training data by injecting malicious samples, which get propagated into the target model during transfer learning, causing misclassification or unintended behaviour in the latter. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Model Stealing for Transfer Learning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-TL-02 |
| **Threat title** | Model Stealing for Transfer Learning |
| **Threat description** | By using a series of carefully crafted queries to the source model, attackers can extract sensitive information about its architecture and parameters, which they can then use to build a similar model for malicious purposes. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Adversarial Examples

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-TL-03 |
| **Threat title** | Adversarial Examples |
| **Threat description** | Attackers can create input samples that fool both the source and target models, even if these inputs are visually or semantically similar to benign ones. These adversarial examples can be used to bypass security systems or cause misclassifications in critical applications like self-driving cars or healthcare. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Backdoor for Transfer Learning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-TL-04 |
| **Threat title** | Backdoor for Transfer Learning |
| **Threat description** | Attackers can embed hidden malicious functionality into the source model, which remains dormant until triggered by specific input patterns. When the trojans are transferred to the target model, they can cause unexpected behaviour or leak sensitive information. |
| **Threat type** | Tampering, Information disclosure |
| **Vulnerabilities** | Vulnerabilities in ML model transfer |
| **Impact type** | Authenticity, Integrity, Availability, Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25], Trained ML model [ASSET-D-26],  ML components deploying machine learning (xApps, rApps) [ASSET-C-12], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

## Model Skewing (ML08:2023)

### Data manipulation for Model Skewing

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MS-01 |
| **Threat title** | Data manipulation for Model Skewing |
| **Threat description** | The attacker manipulates the data involved in the training of the model. The attacker modifies/injects the training data with modified data, or modifies the labels associated with the training data, resulting in learnt model behaviour to be changed. The impacted models have reduced accuracy / biased trained models, thereby the ability to made correct predications is compromised, or the output is biased. The model skewing attacks can significantly impact the operation of networks causing service disruptions, reputational damage, and financial losses. |
| **Threat type** | Tampering, (Applicable Threats T-SMO-11, T-SMO-12, T-SMO-31, t\_rApps\_05) |
| **Vulnerabilities** | Vulnerabilities in (i) training data validation, (ii) less robust model training against cyber-attacks, (iii) less model performance monitoring for monitoring of suspicious activity |
| **Impact type** | Integrity, and use-case specific impacts on O-RAN network. |
| **Affected Assets** | Trained ML model [ASSET-D-26],  ML Prediction Results [ASSET-D-27],  ML system [ASSET-D-28] |

### Feedback manipulation for Model Skewing

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MS-02 |
| **Threat title** | Feedback manipulation for Model Skewing |
| **Threat description** | The attacker manipulates the feedback mechanisms used by the AI/ML for continuous improvement during deployment. The attack can significantly impact the operation of networks causing service disruptions, reputational damage, and financial losses. |
| **Threat type** | Tampering, (Applicable Threats T-SMO-11, T-SMO-12, T-SMO-31, t\_rApps\_05) |
| **Vulnerabilities** | Vulnerabilities in (i) less secure feedback mechanisms to prevent manipulation, (ii) less robust model training against cyber-attacks, (iii) less model performance monitoring for monitoring of suspicious activity |
| **Impact type** | Integrity, and use-case specific impacts on O-RAN network. |
| **Affected Assets** | Trained ML model [ASSET-D-26],  ML Prediction Results [ASSET-D-27],  ML system [ASSET-D-28] |

## Output Integrity Attack (ML09:2023)

### Denial of Service Impacting Output Integrity

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-OI-01 |
| **Threat title** | Denial of Service Impacting Output Integrity |
| **Threat description** | The AI/ML output can be manipulated to contain values that are incorrect or out of range. This can lead to service degradation, partial failure or total failure of the O-RAN element that is consuming this output and lead to denial of service to other O-RAN elements or subscribers. |
| **Threat type** | Denial of service |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Availability |
| **Affected Assets** | Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26], ML prediction results [ASSET-D-27], A1 policies [ASSET-D-07], E2 node data [ASSET-D-11], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Subscriber QoE Change

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-OI-02 |
| **Threat title** | Subscriber QoE Change |
| **Threat description** | The AI/ML output can be manipulated to contain values which can influence the configuration or policies being used by the O-RAN elements. This can lead to a behaviour of O-RAN network elements where a subscriber or a group of subscribers is treated unfairly or with a preference. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in data sources, vulnerabilities in training database, vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Integrity |
| **Affected Assets** | Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26], ML prediction results [ASSET-D-27], Data transported over the O1 interface [ASSET-D-09], A1 policies [ASSET-D-07], E2 node data [ASSET-D-11], Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

## Model Poisoning (ML10:2023)

### Model Parameter Poisoning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MP-01 |
| **Threat title** | Model Parameter Poisoning |
| **Threat description** | The adversary has access to the parameters file of a trained model. The parameters file is altered to change the model’s predictions to achieve the goals of the adversary. |
| **Threat type** | Tampering |
| **Vulnerabilities** | * Insufficient access controls to the model’s parameters [i.20] * Inadequate monitoring and logging of model’s activity [i.19] * Lack of control for poisoning in model [i.19] * Weak access protection mechanisms for ML model components [i.19] |
| **Impact type** | Integrity |
| **Affected Assets** | [ASSET-D-26] Trained ML model.  [ASSET-C-12] ML components deploying machine learning (xApps, rApps).  [ASSET-D-27] The ML prediction results built into the model (e.g. expected outcomes). |

### Inference Algorithm Poisoning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MP-02 |
| **Threat title** | Inference Algorithm Poisoning |
| **Threat description** | The adversary has access to the code of a trained model. The inference algorithm is altered to change the model’s predictions to achieve the goals of the adversary. |
| **Threat type** | Tampering |
| **Vulnerabilities** | * Insufficient access controls to the model’s code [i..20] * Inadequate monitoring and logging of model’s activity [i.19] * Lack of code integrity * Weak access protection mechanisms for ML model components [i.19] |
| **Impact type** | Integrity |
| **Affected Assets** | [ASSET-D-26] Trained ML model.  [ASSET-D-27] The ML prediction results built into the model (e.g. expected outcomes). |

### Learning Algorithm Poisoning

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MP-03 |
| **Threat title** | Learning Algorithm Poisoning |
| **Threat description** | The adversary has access to the code of an untrained model. The learning algorithm is altered to cause the model to make incorrect predictions. |
| **Threat type** | Tampering |
| **Vulnerabilities** | * Insufficient access controls to the model’s code [i.20] * Inadequate monitoring and logging of model’s activity [i.19] * Weak access protection mechanisms for ML model components [i.19] |
| **Impact type** | Integrity |
| **Affected Assets** | [ASSET-C-12]ML components deploying machine learning (xApps, rApps). |

### Model Control

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-MP-04 |
| **Threat title** | Model Control |
| **Threat description** | An adversary takes control of the model parameters by either generating a Trojan trigger and inserting it in the model or by sending malicious local model updates in federated learning |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in ML model storage, vulnerabilities in ML model transfer |
| **Impact type** | Integrity |
| **Affected Assets** | ML components deploying machine learning [ASSET-C-12]  The trained ML model [ASSET-D-26],  The ML prediction results built into the model [ASSET-D-27],  The behavior of the ML system [ASSET-D-28]. |

## Evasion attacks

### Grey-box based Data Evasion

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-EV-01 |
| **Threat title** | Grey-box based Data Evasion |
| **Threat description** | Possibility to tamper with the input to AI ML application during the inference phase in the Near-RT RIC or Non-RT RIC (model, model parameters, etc.) for example the model's gradient |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in the input APIs of the AI ML model application  Vulnerabilities in the source of input data used by AI ML for inferences |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### White box evasion attack

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-EV-02 |
| **Threat title** | White-box based data evasion attacks |
| **Threat description** | An attacker has complete knowledge of the AI/ML models and algorithms in use, including configurable parameters such as thresholds and hyper-parameters. By calculating the gradients of the classification loss with respect to the input sample, the attacker manipulates the input to maximize the classification loss |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in the input APIs of the AI ML model application  Vulnerabilities in the source of input data used by AI ML for inferences |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

### Black box evasion attack

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-EV-03 |
| **Threat title** | Black-box based data evasion attacks |
| **Threat description** | An attacker can interact with a trained ML model by querying it on various data samples and obtaining the model’s predictions and estimating the gradients of the classification loss with respect to the input sample with no prior knowledge of the AI/ML algorithm, or the parameters being used. |
| **Threat type** | Tampering |
| **Vulnerabilities** | Vulnerabilities in the input APIs of the AI ML model application  Vulnerabilities in the source of input data used by AI ML for inferences |
| **Impact type** | Integrity |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Training or test data sets collected externally or internally [ASSET-D-25],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11] |

## AI Energy-Latency Attack

### Model Extraction

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-EL-01 |
| **Threat title** | Model Extraction |
| **Threat description** | By monitoring the energy and latency patterns, an attacker might deduce the structure of the neural network, including the types of layers, their sequence, and possibly hyperparameters. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources |
| **Impact type** | Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11],  AAL software [ASSET-C-29],  Hardware accelerator device firmware [ASSET-C-30] |

### Data Inference

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-EL-02 |
| **Threat title** | Data Inference |
| **Threat description** | Observing the energy and latency signatures of different computations could allow an attacker to make inferences about the input data, potentially leading to privacy breaches, especially if the data is sensitive. |
| **Threat type** | Information disclosure |
| **Vulnerabilities** | Vulnerabilities in data sources |
| **Impact type** | Confidentiality |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11],  AAL software [ASSET-C-29],  Hardware accelerator device firmware [ASSET-C-30] |

### Denial of Service via Side Channels

|  |  |
| --- | --- |
| **Threat ID** | T-AIML-EL-03 |
| **Threat title** | Denial of Service via Side Channels |
| **Threat description** | An attacker could exploit these side channels to craft inputs that maximize energy consumption or computation time, leading to a denial of service by overloading the system or draining battery life. |
| **Threat type** | Denial of service |
| **Vulnerabilities** | Vulnerabilities in data sources |
| **Impact type** | Availability |
| **Affected Assets** | Database holding data from xApp applications and E2 Node [ASSET-D-10],  Trained ML model [ASSET-D-26],  Near-RT-RIC SW [ASSET-C-02],  Non-RT-RIC SW [ASSET-C-11],  AAL software [ASSET-C-29],  Hardware accelerator device firmware [ASSET-C-30] |

# Security Controls

Generic industry recommendations for strong security controls are provided from sources such as the OWASP Top 10 Proactive Controls [i.24], Center for Internet Security (CIS) Critical Security Controls [i.25], Cloud Security Alliance (CSA) Cloud Control Matrix (CCM) [i.26], ISO/IEC 27001:2013 Information Security Management System (ISMS) [i.27], NIST SP 800-53r5 Security and Privacy Controls for Information Systems and Organizations [i.28], and Cybersecurity and Infrastructure Security Agency (CISA) Security Guidance for 5G Cloud Infrastructures [i.29].

AI/ML- specific industry recommendations for strong security controls are provided from e.g. ENISA [i.19], and OWASP [i.20].

The ENISA security controls recommendations are as follows:

**Organisational:**

* Apply a RBAC model, respecting the least privileged principle
* Apply documentation requirements to AI projects
* Assess the regulations and laws the ML application must comply with
* Ensure ML applications comply with data security requirements
* Ensure ML applications comply with identity management, authentication, and access control policies
* Ensure ML applications comply with protection policies and are integrated to security operations processes
* Ensure ML applications comply with security policies
* Include ML applications into detection and response to security incident processes
* Include ML applications in asset management processes
* Integrate ML applications into the overall cyber-resilience strategy
* Integrate ML specificities to existing security policies

**Technical:**

* Assess the exposure level of the model used
* Check the vulnerabilities of the components used so that they have an appropriate security level
* Conduct a risk analysis of the ML application
* Control all data used by the ML model
* Define and monitor indicators for proper functioning of the model
* Ensure appropriate protection is deployed for test environments
* Ensure ML applications comply with third parties’ security requirements
* Ensure ML projects follow the global process for integrating security into projects

**Specific ML:**

* Add some adversarial examples to the training dataset
* Apply modifications on inputs
* Build explainable models
* Choose and define a more resilient model design
* Enlarge the training dataset
* Ensure that models are unbiased
* Ensure that models respect differential privacy to a sufficient degree
* Ensure that the model is sufficiently resilient to the environment in which it will operate.
* Implement processes to maintain security levels of ML components over time
* Implement tools to detect if a data point is an adversarial example or not
* Integrate ML specificities to awareness strategy and ensure all ML stakeholders are receiving it
* Integrate poisoning control after the “model evaluation” phase
* Reduce the available information about the model
* Reduce the information given by the model
* Use federated learning to minimize risk of data breaches
* Use less easily transferable models

OWASP recommends security controls related to their top 10 Machine Learning Security Risks. In sum they are:

* Adversarial training
* Robust models
* Input/Data validation and verification
* Secure data storage
* Data separation
* Access control
* Testing, Monitoring and auditing
* Model validation
* Model ensembles
* Anomaly detection
* Model transparency
* Model retraining
* Model training on randomized or shuffled data
* Model Obfuscation
* Regularisation
* Reducing the training data
* Encryption
* Regular backups
* Watermarking
* Legal protection
* Verify Package Signatures
* Use Secure Package Repositories
* Keep Packages Up-to-date
* Use Virtual Environments
* Perform Code Reviews
* Use Package Verification Tools
* Educate Developers
* Regularly monitor and update the training datasets
* Use secure and trusted training datasets
* Implement model isolation
* Use differential privacy
* Perform regular security audits
* Verify the authenticity of feedback data
* Continuously train the model
* Tamper-evident logs
* Regular software updates

With consideration of these external sources, the following security controls are being evaluated for the AI/ML risk analysis:

* Control 1: Obfuscation ([i.20], in " ML05:2023 Model Stealing"):
  + Hiding the inner workings of an ML model by adding noise or complexity, making it harder for attackers to reverse engineer the model. Techniques include gradient masking and neuron shuffling.
* Control 2: Watermarking ([i.20], in " ML05:2023 Model Stealing"):
  + Embedding a unique digital fingerprint into the model’s weights or architecture during training. If stolen and used without permission, the watermark can be detected, revealing the original owner.
* Control 3: Encryption ([i.20], in " ML05:2023 Model Stealing"):
  + Encrypting the ML model, training data and other sensitive information.
* Control 4: Differential Privacy [i.37]:
  + Adding random noise to the training data or model’s outputs, which helps maintain privacy while preserving overall accuracy. However, this method may introduce some error into the stolen model’s predictions, making it less valuable for malicious use.
* Control 5: Model Splitting [i.38]:
  + Divide a single ML model into multiple sub-models, each trained on a different part of the training data. Then, distribute these sub-models across different servers or organizations so that no single party has access to the complete model. This approach makes it harder for attackers to steal and reconstruct the original model.
* Control 6: Intellectual Property (IP) Protection ([i.20], in " ML05:2023 Model Stealing"):
  + Using copyright, patents, or trade secrets laws to protect your ML models as intellectual property. However, this method may not always deter sophisticated attackers, and enforcement can be challenging across international borders.
* Control 7: Access Control ([i.20], in " ML05:2023 Model Stealing"):
  + Implement access controls, such as role-based access control (RBAC) or attribute-based access control (ABAC), to restrict who can view, train, or deploy ML models within your organization. This approach helps prevent insider threats and unauthorized access.
* Control 8: Monitoring ([i.20], in " ML05:2023 Model Stealing"):
  + Continuously monitor model usage, data flows, and system logs for any suspicious activity that might indicate an attack in progress. Machine Learning Security (MLSec) tools can help detect such anomalies.
* Control 9: Training and Education ([i.20], in "ML06:2023 AI Supply Chain Attacks"):
  + Regularly train employees on best practices for ML model security, including how to recognize phishing attacks or social engineering attempts that could lead to model theft. Raising awareness of these threats can help reduce the likelihood of human error causing a breach.
* Control 10: Data Sanitisation ([i.39], [i.40]):
  + This technique involves cleaning and preprocessing input data to remove or encode any malicious characters or sequences that could be used to launch an attack, or access sensitive information. This includes techniques like input validation, encoding/decoding, and escaping special characters, but also de-identification or pseudonymization on the training dataset.
* Control 11: Feature Selection ([i.41], section 2.3.5):
  + By selecting a reduced set of relevant features for model training, this method can help reduce the likelihood of attacks by eliminating less important or redundant features that might be manipulated. Feature selection techniques include feature ranking (e.g., PCA) and dimensionality reduction (e.g., LDA).
* Control 12: Input Transformation ([i.42]):
  + This method involves applying mathematical transformations to input data, such as normalization or standardization, to make the inputs less susceptible to small perturbations that could lead to misclassification. Examples include Min-max scaling and Z-score normalization [i.32].
* Control 13: Adversarial Training ([i.20], in "ML01:2023 Input Manipulation Attack"):
  + In this approach, a model is trained on both clean data and adversarial examples (inputs slightly perturbed to cause misclassifications). This can help the model become more robust against input manipulation attacks by learning to generalize better from noisy or corrupted inputs.
* Control 14: Ensemble Methods ([i.20], in "ML02:2023 Data Poisoning Attack"):
  + Combining multiple models with different architectures, feature representations, and training sets can improve overall model performance and reduce the likelihood of all models being fooled by the same attack. Techniques include voting classifiers ([i.33]) and stacked ensembles ([i.34]).
* Control 15: Robust Models ([i.20], in "ML01:2023 Input Manipulation Attack"):
  + This approach involves proving that a model's output will remain within a specified range for a given input region, which can help to certify the robustness of the model against small perturbations or adversarial examples. Techniques include formal verification and interval analysis.
* Control 16: Secure data acquisition ([i.20], in "ML01:2023 Input Manipulation Attack"):
  + Ensure that the data is acquired from trusted sources and maintain a proper chain of custody to prevent tampering during collection or preprocessing phases.
* Control 17: Model agility ([i.20], in "ML03:2023 Model Inversion Attack"):
  + Regularly update and retrain ML models with fresh and clean data to reduce the effectiveness of stale poisoned samples. This approach, coupled with techniques like model stacking or ensembles, can further enhance resilience against attacks.
* Control 18: Model Distillation ([i.43]):
  + This method involves training a smaller, less complex model (called a distilled or student model) on the softmax function outputs of a larger, more complex teacher model. The student model is designed to mimic the behaviour of the teacher model but with fewer parameters and reduced representational power. Attackers will have a harder time extracting sensitive information from the student model's intermediate representations compared to those of the original teacher model.
* Control 19: Secure Multi-Party Computation (SMPC) ([i.44]):
  + SMPC protocols allow for securely training models on distributed data without revealing the individual records to any party involved in the computation. By splitting both the input data and the model itself across multiple parties, membership inference attacks become much more difficult as no single party has access to the complete information required to perform the attack.
* Control 20: Code signing and verification ([i.20], in "ML06:2023 AI Supply Chain Attacks"):
  + Signing machine learning code with strong cryptographic signatures can help ensure its authenticity and prevent unauthorized tampering during distribution. Verifying these signatures upon receipt ensures the integrity of the received codebase.
* Control 21: Hardware-based security measures:
  + Using trusted execution environments (TEEs) or secure enclaves provided by hardware vendors such as Intel SGX, AMD SEV, or ARM TrustZone can isolate sensitive ML computations from potentially compromised parts of the software stack. This way, even if an attacker manages to compromise other portions of the system, they cannot tamper with the protected code and data within these secure enclaves.
* Control 22: Explainable Models ([i.19]):
  + Ensuring that the inner workings and decision-making process of an ML model are interpretable can help detect anomalies in its behaviour more easily. Explainable AI (XAI) techniques, such as LIME ([i.35]) or SHAP ([i.36]), can be used to understand how each input feature contributes to a particular prediction, which can aid in identifying potential evasion attempts.
* Control 23: Data Augmentation ([i.45], clause 1.2.2):
  + Increasing the diversity of the training data set can make it more robust against evasion attacks. By augmenting the data with various transformations (such as rotation, scaling, flipping, etc.), models are exposed to a wider range of input patterns and become less susceptible to small perturbations in input features.
* Control 24: Secure software supply chain management:
  + Implementing secure software development practices for AI projects, including regular audit and update of outdated packages to patch known vulnerabilities.
* Control 25: Model Poisoning Control [i.19]:
  + Integrate poisoning control after the "model evaluation" phase in the ML Training Host. Before moving the model to production and then on a regular basis, the model is evaluated to ensure it has not been poisoned. For example: deep learning classification algorithms can be checked for poisoning using the STRIP technique [i.60]. The principle is to disturb the inputs and observe the randomness of the predictions.
* Control 26: Energy and Latency Obfuscation [i.64]:
  + Techniques such as adding noise to the power supply or equalizing computation times across different operations can help obscure the relationship between energy/latency and the actual computations being performed.
* Control 27: Secure Multi-Tenancy:
  + For cloud environments, ensuring that different users' computations are properly isolated and do not interfere with each other is crucial to prevent cross-user energy-latency attacks.

# Risk Assessment

## Introduction

This clause provides risk assessment tables for each of the identified assets. These tables list the assets, threats, impacts, and possible security controls.

A malicious actor may be a nation-state adversary, cybercriminal, or employee. In a ZTA, perimeter defences alone are insufficient. AI/ML must be protected from untrusted external sources attempting to have access, while also assuming internal threat actors are inside the network with access to its functions and data. Security controls for a ZTA, protecting against external and internal threats, should be implemented through a risk-based approach. A risk analysis calculates risk levels by assessing the threat’s Likelihood of attack and the Impact from the attack. External and internal threats are from the perspective of the O-RAN architecture. External Threats are external to the O-RAN and Internal Threats are internal to O-RAN.

Impact scores can be lowered with consideration of existing security controls. Impact scoring is based upon current security controls. Impact scoring does not consider security controls that may be potentially specified in the future.

Likelihood scores may be higher when the goal is a ZTA, because external and internal threats must be considered. When likelihood scoring during a risk analysis, it is necessary to consider internal threats performing reconnaissance attacks impacting confidentiality and privacy and attacks causing damage or degrading performance impacting availability. Internal threat actors are less likely to perform damaging attacks that are quickly and easily detected and blocked, but more likely to attempt reconnaissance attacks to collect information. As a result, reconnaissance type attacks can be scored Likelihood = High while damaging/availability attacks can be scored Likelihood = Medium or Low.

This risk assessment covers the identified assets and not the organization implementing the AI system. There may be additional risks that are not apparent at the system level. A voluntary framework to define and manage risk at the organizational level is described in [i.10]. Some of the threats documented in this report may be used as content when performing the MAP and MANAGE functions described in [i.10].

## Input Manipulation Attack (ML01:2023)

Table 8.2‑1 AI/ML Risk Analysis – Input Manipulation Attack

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-D-25,  ASSET-D-26,  ASSET-D-27,  ASSET-D-28 | Training or test data,  Trained ML model,  ML prediction results,  ML system behaviour | T-AIML-IM-1 | Training Data Control | Impact = High  Likelihood = High | Access Control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Secure data acquisition, | Control 7  Control 8  Control 10  Control 11  Control 12  Control 16 |
| ASSET-D-25,  ASSET-D-26,  ASSET-D-27,  ASSET-D-28 | Training or test data,  Trained ML model,  ML prediction results,  ML system behaviour | T-AIML-IM-2 | Testing Data Control | Impact = High  Likelihood = Medium | Access Control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Adversarial Training,  Robust Models,  Secure data acquisition | Control 7  Control 8  Control 10  Control 11  Control 12  Control 13  Control 15  Control 16 |
| ASSET-D-25,  ASSET-D-26,  ASSET-D-27,  ASSET-D-28 | Training or test data,  Trained ML model,  ML prediction results,  ML system behaviour | T-AIML-IM-3 | Label Manipulation | Impact = High  Likelihood = Medium | Access Control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input transformation,  Secure data acquisition | Control 7  Control 8  Control 10  Control 11  Control 12  Control 16 |
| ASSET-D-08,  ASSET-D-25,  ASSET-D-26,  ASSET-D-27,  ASSET-D-28 | A1 Enrichment Information,  Training or test data,  Trained ML model,  ML prediction results,  ML system behaviour | T-AIML-IM-4 | AI/ML Query Exploitation | Impact = High  Likelihood = High | Adversarial Training | Control 13 |

## Data poisoning attacks (ML02:2023)

Table 8.3‑1 AI/ML Risk Analysis - Data poisoning attacks

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Training data,  Trained ML model | T-AIML-DP-01 | Black-box Data Poisoning | Impact= High Likelihood= Medium | Access control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Adversarial Training,  Ensemble Methods,  Secure data acquisition,  Model agility,  Data Augmentation | Control 7,  Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 14,  Control 16,  Control 17,  Control 23 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-C-14,  ASSET-C-22,  ASSET-C-40,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  O1-, A1-, E2- Interface, Database data,  Training data,  Trained ML model | T-AIML-DP-02 | Grey-box Data Poisoning | Impact= High Likelihood= Medium | Access control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Adversarial Training,  Ensemble Methods,  Secure data acquisition,  Model agility,  Data Augmentation | Control 7,  Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 14,  Control 16,  Control 17,  Control 23 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-C-14,  ASSET-C-22,  ASSET-C-40,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  O1-, A1-, E2- Interface,  Database data,  Training data,  Trained ML model | T-AIML-DP-03 | White-box Data Poisoning | Impact= High Likelihood= Medium | Access control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Adversarial Training,  Ensemble Methods,  Secure data acquisition,  Model agility,  Data Augmentation | Control 7,  Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 14,  Control 16,  Control 17,  Control 23 |

## Model Inversion Attack (ML03:2023)

Table 8.4‑1 AI/ML Risk Analysis – Model Inversion Attack

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| ASSET-D-10,  ASSET-D-25,  ASSET-D-26,  ASSET-D-27,  ASSET-D-28 | Database data,  Training data,  Trained ML model,  ML prediction results,  ML system behaviour | T-AIML-MOI-01 | Black-box Model Inversion | Impact= High  Likelihood= High | Obfuscation,  Differential Privacy,  Model Splitting,  Access Control,  Monitoring,  Data Sanitisation,  Model agility,  Secure Multi-Party Computation (SMPC) | Control 1,  Control 4,  Control 5,  Control 7,  Control 8,  Control 10,  Control 17,  Control 19 |
| ASSET-D-10,  ASSET-D-25,  ASSET-D-26,  ASSET-D-27,  ASSET-D-28 | Database data,  Training data,  Trained ML model,  ML prediction results,  ML system behaviour | T-AIML-MOI-02 | White-box Model Inversion | Impact= High  Likelihood= High | Obfuscation  Differential Privacy  Model Splitting  Access Control  Monitoring  Data Sanitisation,  Model agility,  Secure Multi-Party Computation (SMPC) | Control 1  Control 4  Control 5  Control 7  Control 8  Control 10,  Control 17,  Control 19 |

## Membership Inference Attack (ML04:2023)

Table 8.5‑1 AI/ML Risk Analysis - Membership Inference Attacks

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Trained ML model | T-AIML-MEI-01 | Information access – Membership Inference | Impact= High Likelihood= Medium | Differential Privacy,  Access Control,  Monitoring,  Feature Selection,  Input Transformation,  Adversarial Training,  Model agility,  SMPC | Control 4,  Control 7,  Control 8,  Control 11,  Control 12,  Control 13,  Control 17,  Control 19 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Trained ML model | T-AIML-MEI-02 | Inference Technique – Membership Inference | Impact= High Likelihood= Medium | Differential Privacy,  Monitoring,  Feature Selection,  Input Transformation,  Adversarial Training,  Model agility,  Model Distillation,  SMPC | Control 4,  Control 8,  Control 11,  Control 12,  Control 13,  Control 17,  Control 18,  Control 19 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Trained ML model | T-AIML-MEI-03 | Target based – Membership Inference | Impact= High Likelihood= Medium | Encryption,  Differential Privacy,  Access Control,  Monitoring,  Feature Selection,  Input Transformation,  Adversarial Training,  Model agility,  SMPC | Control 3,  Control 4,  Control 7,  Control 8,  Control 11,  Control 12,  Control 13,  Control 17,  Control 19 |

## Model Stealing (ML05:2023)

Table 8.6‑1 AI/ML Risk Analysis – Model Stealing

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Database data,  Training data,  Trained ML model | T-AIML-ME-01 | Intellectual Property Theft | Impact= High Likelihood= Low | Obfuscation,  Watermarking,  Encryption,  Differential Privacy,  Model Splitting,  Intellectual Property (IP) Protection,  Access control,  Monitoring,  Training and Education | Control 1,  Control 2,  Control 3,  Control 4,  Control 5,  Control 6,  Control 7,  Control 8,  Control 9 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Database data,  Training data,  Trained ML model | T-AIML-ME-02 | Data Privacy Violations | Impact= High Likelihood= Low | Obfuscation,  Encryption,  Differential Privacy,  Model Splitting,  Access control,  Monitoring,  Training and Education | Control 1,  Control 3,  Control 4,  Control 5,  Control 8,  Control 9 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Database data,  Training data,  Trained ML model | T-AIML-ME-03 | Adversarial Attacks with Stolen Models | Impact= High Likelihood= Low | Obfuscation,  Encryption,  Differential Privacy,  Model Splitting,  Access control,  Monitoring,  Training and Education | Control 1,  Control 3,  Control 4,  Control 5,  Control 8,  Control 9 |

## AI Supply Chain Attacks (ML06:2023)

Table 8.7‑1 AI/ML Risk Analysis - AI Supply Chain Attacks

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | ML components,  Database data,  Training data,  Trained ML model | T-AIML-SC-01 | Data Poisoning via Supply Chain | Impact= High Likelihood= Medium | Access Control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Adversarial Training,  Ensemble Methods,  Secure data acquisition,  Model agility,  Code signing and verification | Control 7,  Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 14,  Control 16,  Control 17,  Control 20 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Training data,  Trained ML model | T-AIML-SC-02 | Model Tampering | Impact= High Likelihood= Medium | Access Control,  Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Adversarial Training,  Ensemble Methods,  Secure data acquisition,  Model agility,  Code signing and verification | Control 7,  Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 14,  Control 16,  Control 17,  Control 20 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Trained ML model | T-AIML-SC-03 | Backdoor via Supply Chain | Impact= High Likelihood= Low | Access Control,  Monitoring,  Feature Selection,  Input Transformation,  Ensemble Methods,  Secure data acquisition,  Model agility,  Code signing and verification,  Explainable Models | Control 7,  Control 8,  Control 11,  Control 12,  Control 14,  Control 16,  Control 17,  Control 20,  Control 22 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Trained ML model | T-AIML-SC-04 | Hardware Tampering | Impact= High Likelihood= Low | Access Control,  Monitoring,  Code signing and verification,  Hardware-based security measures | Control 7,  Control 8,  Control 20,  Control 21 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Training data,  Trained ML model | T-AIML-SC-05 | Insecure Maintenance APIs | Impact= Medium Likelihood= Medium | Encryption,  Access Control,  Monitoring,  Data Sanitisation,  Code signing and verification measures | Control 3,  Control 7,  Control 8,  Control 10,  Control 20 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Training data,  Trained ML model | T-AIML-SC-06 | Third-party Libraries and Dependencies | Impact= Medium Likelihood= Medium | Code signing and verification measures,  Secure software supply chain management | Control 20,  Control 24 |

## Transfer Learning Attack (ML07:2023)

Table 8.8‑1 AI/ML Risk Analysis - Transfer Learning Attack

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Training data,  Trained ML model | T-AIML-TL-01 | Data Poisoning for Transfer Learning | Impact= High Likelihood= Medium | Monitoring,  Data Sanitisation,  Feature Selection,  Input Transformation,  Adversarial Training,  Ensemble Methods,  Secure data acquisition,  Model agility | Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 14,  Control 16,  Control 17 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Training data,  Trained ML model | T-AIML-TL-02 | Model Stealing for Transfer Learning | Impact= High Likelihood= Low | Obfuscation,  Watermarking,  Encryption,  Differential Privacy,  Model Splitting,  Intellectual Property (IP) Protection,  Access control,  Monitoring,  Training and Education | Control 1,  Control 2,  Control 3,  Control 4,  Control 5,  Control 6,  Control 7,  Control 8,  Control 9 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Database data,  Training data,  Trained ML model | T-AIML-TL-03 | Adversarial Examples | Impact= High Likelihood= Low | Obfuscation,  Encryption,  Differential Privacy,  Model Splitting,  Access control,  Monitoring,  Training and Education | Control 1,  Control 3,  Control 4,  Control 5,  Control 8,  Control 9 |
| ASSET-C-02,  ASSET-C-11,  ASSET-C-12,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  ML components,  Database data,  Training data,  Trained ML model | T-AIML-TL-04 | Backdoor for Transfer Learning | Impact= High Likelihood= Low | Access Control,  Monitoring,  Feature Selection,  Input Transformation,  Ensemble Methods,  Secure data acquisition,  Model agility,  Code signing and verification,  Explainable Models | Control 7,  Control 8,  Control 11,  Control 12,  Control 14,  Control 16,  Control 17,  Control 20,  Control 22 |

## Model Skewing (ML08:2023)

Table 8.9‑1 AI/ML Risk Analysis - Model Skewing Attacks

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| ASSET-D-26,  ASSET-C-27,  ASSET-C-28 | ML model,  ML Prediction Results,  ML system | T-AIML-MS-01 | Data Manipulation for Model Skewing | Impact= High Likelihood= Medium | Access Control,  Monitoring,  Data Sanitation,  Feature Selection,  Input Transformation,  Adversarial Training,  Robust Models,  Model agility,  Explainable models | Control 7,  Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 15,  Control 17,  Control 22 |
| ASSET-D-26,  ASSET-C-27,  ASSET-C-28 | ML model,  ML Prediction Results,  ML system | T-AIML-MS-02 | Feedback Manipulation for Model Skewing | Impact= High Likelihood= Medium | Monitoring,  Data sanitation,  Feature Selection,  Input Transformation,  Adversarial Training,  Model agility,  Explainable models,  Data Augmentation | Control 8,  Control 10,  Control 11,  Control 12,  Control 13,  Control 17,  Control 22,  Control 23 |

## Output Integrity Attack (ML09:2023)

Table 8.10‑1 AI/ML Risk Analysis – Output Integrity attacks

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-07,  ASSET-D-25,  ASSET-D-26,  ASSET-D-27,  ASSET-D-11 | Near-RT-RIC SW,  Non-RT-RIC SW,  A1 policies,  Training data,  Trained ML model,  ML prediction results,  E2 node data | T-AIML-OI-01 | Denial of Service impacting Output Integrity | Impact= High Likelihood= Medium | Encryption,  Access Control,  Monitoring,  Secure data acquisition,  Explainable Models,  Secure software supply chain management | Control 3,  Control 7,  Control 8,  Control 16,  Control 22,  Control 24 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-07,  ASSET-D-09,  ASSET-D-11,  ASSET-D-25,  ASSET-D-26,  ASSET-D-27 | Near-RT-RIC SW,  Non-RT-RIC SW,  A1 policies,  Data transported over the O1 interface,  E2 node data,  Training data,  Trained ML model,  ML prediction results | T-AIML-OI-02 | Subscriber QoE Change | Impact= High Likelihood= Low | Encryption,  Access Control,  Monitoring,  Secure data acquisition,  Explainable Models,  Secure software supply chain management | Control 3,  Control 7,  Control 8,  Control 16,  Control 22,  Control 24 |

## Model Poisoning (ML10:2023)

Table 8.11‑1 AI/ML Risk Analysis – Model Poisoning Attacks

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-D-26, ASSET-C-12, ASSET-D-27 | Trained ML model, ML components, ML prediction results | T-AIML-MP-01 | Model Parameter Poisoning | Impact=High  Likelihood=High | Access Control, Monitoring, Secure software supply chain management, Model Poisoning Control, Explainable Models | Control 7, Control 8, Control 24, Control 25, Control 22 |
| ASSET-D-26, ASSET-D-27 | Trained ML model, ML prediction results | T-AIML-MP-02 | Inference Algorithm Poisoning | Impact=High  Likelihood=High | Access Control, Monitoring, Code Signing and verification, Secure software supply chain management, Explainable Models | Control 7, Control 8, Control 20, Control 24, Control 22 |
| ASSET-C-12 | ML components | T-AIML-MP-03 | Learning Algorithm Poisoning | Impact=High  Likelihood=High | Access Control, Monitoring, Code Signing and verification, Secure software supply chain management, Explainable Models | Control 7, Control 8, Control 20, Control 24, Control 22 |
| ASSET-C-12,  ASSET-D-26,  ASSET-D-27,  ASSET-D-28 | ML components,  Trained ML model,  ML prediction results,  ML system behaviour | T-AIML-MP-04 | Model Control | Impact = High  Likelihood = Medium | Access Control,  Monitoring,  Data Sanitization,  Input Transformation,  Adversarial Training,  Robust Models,  Secure data acquisition | Control 7  Control 8  Control 10  Control 12  Control 13  Control 15  Control 16 |

## Evasion attacks

Table 8.12‑1 AI/ML Risk Analysis - Evasion attacks

| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| --- | --- | --- | --- | --- | --- | --- |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Database data,  Training data,  Trained ML model | T-AIML-EV-01 | Grey-box based Data Evasion | Impact= High Likelihood= Medium | Obfuscation,  Encryption,  Access Control,  Monitoring,  Data Sanitisation,  Adversarial Training,  Robust Models,  Data Augmentation | Control 1,  Control 3,  Control 7,  Control 8,  Control 10,  Control 13,  Control 15,  Control 23 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Database data,  Training data,  Trained ML model | T-AIML-EV-02 | White-box based data evasion attacks | Impact= High Likelihood= Medium | Obfuscation,  Encryption,  Access Control,  Monitoring,  Data Sanitisation,  Adversarial Training,  Robust Models,  Data Augmentation | Control 1,  Control 3,  Control 7,  Control 8,  Control 10,  Control 13,  Control 15,  Control 23 |
| ASSET-C-02,  ASSET-C-11,  ASSET-D-10,  ASSET-D-25,  ASSET-D-26 | Near-RT RIC,  Non-RT RIC,  Database data,  Training data,  Trained ML model | T-AIML-EV-03 | Black-box based data evasion attacks | Impact= High Likelihood= Medium | Obfuscation,  Encryption,  Access Control,  Monitoring,  Data Sanitisation,  Adversarial Training,  Robust Models,  Data Augmentation | Control 1,  Control 3,  Control 7,  Control 8,  Control 10,  Control 13,  Control 15,  Control 23 |

## AI Energy-Latency Attack

Table 8.13‑1 AI/ML Risk Analysis - AI Energy-Latency Attack

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Asset-Id | Asset Name | Threat-Id | Threat Description (Brief) | Impact/ Likelihood Raw Score | Possible Security Controls | Security Control-id |
| ASSET-D-10, ASSET-D-26, ASSET-C-02, ASSET-C-11, ASSET-C-29, ASSET-C-30 | Database data,  ML model,  Near-RT-RIC,  Non-RT-RIC,  AAL software,  accelerator firmware | T-AIML-EL-01 | Model Extraction | Impact= High Likelihood= Medium | Differential Privacy, Access Control, Monitoring, Data Sanitization, Secure data acquisition, Model agility, Hardware-based security measures, Explainable Models, Energy and Latency Obfuscation, secure multi-tenancy | Control 4, Control 7, Control 8, Control 10, Control 16, Control 17, Control 21, Control 22, Control 26, Control 27 |
| ASSET-D-10, ASSET-C-02, ASSET-C-11, ASSET-C-29, ASSET-C-30 | Database data,  Near-RT-RIC,  Non-RT-RIC,  AAL software,  accelerator firmware | T-AIML-EL-02 | Data Inference | Impact= High Likelihood= Medium | Differential Privacy, Access Control, Monitoring, Data Sanitization, Secure data acquisition, Model agility, Hardware-based security measures, Explainable Models, Energy and Latency Obfuscation, secure multi-tenancy | Control 4, Control 7, Control 8, Control 10, Control 16, Control 17, Control 21, Control 22, Control 26, Control 27 |
| ASSET-D-10, ASSET-D-26, ASSET-C-02, ASSET-C-11, ASSET-C-29, ASSET-C-30 | Database data,  ML model,  Near-RT-RIC,  Non-RT-RIC,  AAL software,  accelerator firmware | T-AIML-EL-03 | Denial of Service via Side Channels | Impact= High Likelihood= Medium | Differential Privacy, Access Control, Monitoring, Data Sanitization, Secure data acquisition, Model agility, Hardware-based security measures, Explainable Models, Energy and Latency Obfuscation, secure multi-tenancy | Control 4, Control 7, Control 8, Control 10, Control 16, Control 17, Control 21, Control 22, Control 26, Control 27 |

# Primary Issues

The field of AI/ML security is a rapidly evolving area that has garnered significant attention in recent years due to the growing adoption of these technologies across various domains. While there have been many advances in both research and standardization efforts, it could still be considered as being in an early stage compared to the overall maturity of AI/ML itself.

In terms of research, there is a burgeoning community focused on AI/ML security. This includes work on adversarial examples, privacy-preserving machine learning, and formal verification techniques for deep neural networks. However, much of this work remains theoretical or confined to specific settings. Also, many AI/ML models are currently evaluated based on their accuracy or performance on specific tasks, but there is often little consideration given to their security properties. This can lead to vulnerabilities that can be exploited by attackers. There's still a need for more practical, scalable solutions that can be easily adopted by practitioners without requiring specialized expertise in security or AI/ML.

In terms of standardization, there have been some initial efforts such as the NIST Privacy Framework ([i.56]) and GDPR's "data protection by design" principle ([i.57]) which touch upon AI/ML security but do not specifically address it comprehensively. Organizations like IEEE ([i.55]), ISO ([i.58]), ETSI ([i.49]) and ITU ([i.59]) are also working on developing more targeted standards for AI/ML security; however, these efforts are at various stages of maturity and may take some time before they yield tangible results.

To advance the state of AI/ML security, several things are missing:

1. **More foundational research:** There's a need for deeper understanding of the inherent tradeoffs between privacy, model performance, and security in machine learning systems. This will help inform better-informed design decisions and guide standardization efforts. More specifically there is a need for more research on explainability and interpretability in AI/ML systems. Currently, many AI/ML models are "black boxes" that make decisions based on complex mathematical calculations that are difficult to understand or interpret. This lack of transparency can make it difficult to identify when an AI/ML system has been compromised or manipulated. Developing techniques for explaining and interpreting the decision-making processes of AI/ML systems will be important for ensuring their security.

2. **Practical tools and frameworks**: Researchers should focus on developing easy-to-use, scalable, and open-source tools for AI/ML security that can be seamlessly integrated into existing development pipelines without significant overheads.

3. **Collaboration between researchers and practitioners:** Closer collaboration between academia, industry, and government stakeholders is crucial to ensure the relevance of research findings and inform policy decisions based on real-world needs.

4. **Awareness and education:** There's a need for greater awareness and education about AI/ML security among practitioners, developers, and end-users alike so that they are aware of potential risks and best practices to mitigate them.

5. **Comprehensive standardization efforts:** Standardization bodies should prioritize the development of comprehensive, industry-wide standards for AI/ML security that take into account privacy, safety, and robustness considerations. These standards should be flexible enough to accommodate future advances in technology while providing a baseline for assessing compliance and promoting interoperability.

In summary, while significant progress has been made in the field of AI/ML security, there is still much work to be done. Developing robust testing and evaluation frameworks, researching adversarial attacks and defences, improving explainability and interpretability, and standardizing approaches to security will all be important areas of focus in the coming years.

# Recommendations

## General

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.5]. 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.4]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 WG2 and WG3 to ensure the WG11 security specifications are appropriately referenced by WG2 and WG3 in their specifications.

## Managing Risk

There may be additional risks to an organization implementing or operating an AI/ML application or component as part of their service infrastructure. These risks may be non-technical in nature and involve regulatory or business risk that also must be managed, or technical risks that include a non-technical portion to their mitigation strategy. Organizations should include these risks in their comprehensive risk management strategy by using a framework described in [i.10], or similar frameworks that may be required in the area of operation.

Annex A (informative):

Attack Mapping to NIST AML Taxonomy

To ensure comprehensive coverage of AI/ML attacks in this technical report, a gap analysis was conducted using a NIST Trustworthy and Responsible AI Report, NIST AI 100-2e2023 titled, “Adversarial Machine Learning: A Taxonomy and terminology of Attacks and Mitigations [i.9] as a reference. What follows is a table (Table 10.2‑1) that documents the mapping between attacks listed in this technical report and the attacks enumerated in Figure 1 of [i.9]. There was an attempt to list the section number from [i.9] where the attack is described. However, some of the attacks in [i.9] have dedicated sections while other attacks are referred inside a major section. Any inconsistencies in the below table (Table 10.2‑1) regarding section numbers in [i.9] will be a result of that document’s formatting inconsistency.

There are three attacks (i.e., transfer learning attack, model skewing, and output integrity attack) listed in this technical report where there are no equivalent attacks listed in Figure 1 of [i.9].

The Transfer Learning Attack does have treatment in [i.9]. However, it is not listed within the taxonomy diagram in Figure 1. It appears that [i.9] has left this out as it is considered an inherent model property that attackers can take advantage of and not an attack itself. For this technical report it is important to call out this attack specifically as there are specific mitigations referenced.

The Model Skewing attack is covered by [i.9] in section 2.3 titled Data Poisoning.

The Output Integrity Attack is not encompassed in [i.9]. This type of attack specifically targets the output data from an AI/ML model. As [i.9] is exclusively concerned with attacks that target the inputs and processing in a model, the Output Integrity Attack falls outside its scope. This is shown in the table (Table 10.2‑1) below with the text “out of scope for [i.9]”.

Table 10.2‑1. Mapping of AI/ML TR attacks with NIST AML Taxonomy of attacks

|  |  |
| --- | --- |
| **AI/ML TR Attacks (Clause and Attack Name)** | **[i.9] Attacks (Section and Attack Name)** |
| 5.3.2 Input Manipulation Attacks 5.3.12 Evasion Attacks | 2.2.2 Black-Box Evasion |
| 2.2 Evasion |
| 5.3.3 Data Poisoning Attacks | 2.1.3 Clean-Label Poisoning |
| 2.3.2 Targeted Poisoning |
| 2.3 Data Poisoning |
| 5.3.4 Model Inversion Attacks | 2.4.1 Reconstruction |
| 5.3.5 Membership Inference Attacks | 2.4.2 Membership Inference |
| 2.4.4 Property Inference |
| 5.3.6 Model Stealing Attacks | 2.4.3 Model Extraction |
| 5.3.7 AI Supply Chain Attacks | 2.3.3 Backdoor Poisoning |
| 2.3.3 Clean-Label Backdoor |
| 5.3.8 Transfer Learning Attacks | 2.2.3 Transferability of Attacks |
| 5.3.9 Model Skewing Attacks | 2.3 Data Poisoning |
| 5.3.10 Output Integrity Attacks | Outside scope for [i.9] |
| 5.3.11 Model Poisoning Attacks | 2.3.4 Model Poisoning |
| 5.3.X AI Energy-Latency Attacks | 3.4.1 Energy Latency |

# Annex: Change history/Change request (history)

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision** | **Description** |
| 2023.03.28 | V0.00.01 | Document creation, template |
| 2023.04.26 | V0.00.02 | Approved draft with comments incorporated |
| 2023.06.06 | V0.00.03 | Incorporated approved CR : [ATT-2023.04.26-WG11-CR-0020-TR-AIMLSecurityAssets](https://oranalliance.atlassian.net/wiki/download/attachments/2840133726/ATT-2023.04.26-WG11-CR-0020-TR-AIMLSecurityAssets-v03.docx?api=v2) |
| 2023.10.19 | V00.00.04 | Incorporated approved CRs :  [NOK-2023.06.06-WG11-CR-0078-Datapoisoning-attack-exploits-threats](https://oranalliance.atlassian.net/wiki/download/attachments/2840133726/NOK-2023.06.06-WG11-CR-0078-Datapoisoning-attack-exploits-threats-v06.docx?api=v2)  [NOK-2023.07.17-WG11-CR-0094-Potential exploits and threat analysis for Model extraction attacks](https://oranalliance.atlassian.net/wiki/download/attachments/2840133726/NOK-2023.07.17-WG11-CR-0094-Potential%20exploits%20and%20threat%20analysis%20for%20Model%20extraction%20attacks-01.docx?api=v2)  [NOK-2023.07.07-WG11-CR-0092-Evasion attack-exploits-threats-v02](https://oranalliance.atlassian.net/wiki/download/attachments/2840133726/NOK-2023.07.07-WG11-CR-0092-Evasion%20attack-exploits-threats-v02.docx?api=v2)  [NIST-2023.09.22-WG11-CR-0001-AIRMF-v01](https://oranalliance.atlassian.net/wiki/download/attachments/2840133726/NIST-2023.09.22-WG11-CR0001-AIRMF-v01.docx?api=v2)  [NEC-2023.09.25-WG11-CR0004-AI\_ML\_TR\_AssetIDUpdates\_ODR\_updates-v02](https://oranalliance.atlassian.net/wiki/download/attachments/2840133726/NEC-2023.09.25-WG11-CR0004-AI_ML_TR_AssetIDUpdates_ODR_updates-v01.docx?api=v2) |
| 2023.10.25 | V00.00.05 | ready for plenary review |
| 2023.11.05 | V00.00.06 | Merged:  NOK-2023.11.08-WG11-CR-0123-AIMLsecTR ODR conformance-v05 |
| 2024.01.17 | V00.00.07 | Merged:  NOK-0132 |
| 2024.02.07 | V00.00.08 | Merged:   * NOK-0133 * NOK-0134 * NOK-0135 * NOK-0136 * MTR-0077 * MTR-0078 * ATT-0021   SYM-0038 |
| 2024.02.27 | V00.00.09 | Merged:   * MTR-0082 * NEC-0011 * ATT-0040 * SYM-0043 * NOK-0144 * NOK-0145 * NOK-0146   NOK-0147 |
| 2024.02.28 | V00.00.10 | Added a part missing from ATT-0040 |
| 2024.03.07 | V00.00.11 | First set of WIG review comments resolved |
| 2024.03.08 | V00.00.12 | Merged:   * MTR0087 * ATT-0041 * ATT-0042 * NOK.AO-0152   NEC-0013 |
| 2024.03.12 | V00.00.13 | First batch of review comments processed |
| 2024.03.20 | V00.00.14 | Second batch of review comments processed |
| 2024.03.20 | V01.00 | Published as Final version 01.00 |
| 2024.03.25 | V01.01 | Editorial fixes |
| 2024.03.26 | V01.02 | Removing line numbers from part of the document |
| 2024.05.21 | V01.03 | Adding MTR-0090, ATT-0043, MTR-0091and NOK-0159 |
| 2024.06.06 | V01.04 | Adding NOK-0165, RMI-0021 |
| 2024.06.11 | V01.05 | Adding NOK-0166, RMI-0023 |
| 2024.07.24 | V02.00 | Published as Final version 02.00 |
| 2024.09.11 | V02.01 | Merged RMI-0024, RMI-0028, NEC-0025, NEC-0026 |
| 2024.11.05 | V02.02 | Updated copyright notices |
| 2024.11.27 | V02.03 | Editorial fixes |
| 2024.11.27 | V02.04 | Further editorial fixes |
| 2024.11.27 | V03.00 | Published as Final version 03.00 |