HOCHSCHULE MITTWEIDA
University of Applied Sciences

BACHELORARBEIT

Herr
Johannes Micketeit

Sicherheitstechnische Analyse der richtlinienbasierten Steuerung bei Open RAN

Mittweida, November 2023

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Sicherheitstechnische Analyse der richtlinienbasierten Steuerung bei Open RAN

Autor:
Johannes Micketeit

Studiengang:
Allgemeine und Digitale Forensik

Seminargruppe:
FO20-w5b

Erstprüfer:
Prof. Ronny Bodach

Zweitprüfer:
Markus Walter, M.A.

Einreichung:
Mittweida, 20.11.2023

Verteidigung/Bewertung:
Mittweida, 2023

Faculty of **Applied Computer Sciences and Biosciences**

BACHELOR THESIS

Security analysis of policy-based control at Open RAN

Author:
Johannes Micketeit

Course of Study:
General and digital forensics

Seminar Group:
FO20-w5b

First Examiner:
Prof. Ronny Bodach

Second Examiner:
Markus Walter, M.A.

Submission:
Mittweida, 20.11.2023

Defense/Evaluation:
Mittweida, 2023

Bibliografische Beschreibung

Micketeit, Johannes:
Sicherheitstechnische Analyse der richtlinienbasierten Steuerung bei Open RAN. – 2023. – 47 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2023.

Referat

In dieser Bachelorarbeit wird eine Sicherheitsanalyse für das Anwendungsszenario Traffic Steering im Open RAN durchgeführt. Dabei werden die Architektur und Sicherheitsaspekte des Open RAN sowie des Traffic Steering genauer analysiert. Nach der Ausarbeitung theoretischer Angriffsszenarien soll anschließend eine virtualisierte Open-RAN-Implementierung als Testumgebung für einen geeigneten Angriff dienen, der anschließend evaluiert wird.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Einführung in das Thema

Mit dem Aufstieg des Open Radio Access Network (Open RAN) als vielversprechender Ansatz für den Aufbau von Mobilfunknetzen stehen auch Fragen zur Sicherheit im Zusammenhang mit dieser Technologie im Vordergrund. Open RAN ermöglicht die Interoperabilität und den Einsatz von Hardware und Software verschiedener Hersteller im Radio Access Network (RAN), was einerseits zu einer erhöhten Flexibilität und Kosteneinsparungen führen kann, andererseits aber auch neue Sicherheitsrisiken mit sich bringt. Neben der Interoperabilität des Netzwerks kommt mit den RAN Intelligence Controllern auch eine intelligente Steuerung des RAN hinzu. Hier kommt es zum Einsatz von verschiedenen Applikationen, welche von Drittherstellern angeboten werden und unter anderem eine richtlinienbasierte Steuerung ermöglichen. Die O-RAN Alliance hat verschiedene Use Cases formuliert, bei denen richtlinienbasierte Steuerung verwendet werden soll. Einer dieser Use Cases nennt sich Traffic Steering (TS). Der Ansatz besteht darin, dass Benutzer nicht mehr wie bisher gleich behandelt werden, sondern auch ihr Nutzerszenario beachtet wird. Dies bedeutet, dass Entscheidungen zur Zellenübergabe oder Zellenneuauswahl auf der Grundlage durchschnittlicher KPI-Werte getroffen werden können. Die Hauptidee besteht darin, Richtlinien für verschiedene mobile Verkehrsszenarien, welche das RAN nicht selbst erkennen kann, zu erstellen und das RAN entsprechend zu steuern. Dies geschieht über eine Richtlinie, die festlegt, welche Benutzer sich mit welchen Zellen verbinden sollen oder nicht.

Die richtlinienbasierte Steuerung ist eine wichtige Technologie in Open RAN, die es Netzwerkbetreibern ermöglicht, das Verhalten des RAN durch den Einsatz von Richtlinien zu steuern. Diese Richtlinien definieren Regeln und Parameter, die das Verhalten der Netzwerkkomponenten und Dienste beeinflussen, um bestimmte Ziele wie Netzwerkeffizienz, Ressourcenoptimierung und Sicherheit zu erreichen. Angesichts der zunehmenden Bedeutung von Open RAN in der Telekommunikationsindustrie ist es von entscheidender Bedeutung, die Sicherheitsaspekte dieser Technologie gründlich zu untersuchen und zu analysieren.

Die richtlinienbasierte Steuerung bei Open RAN kann potenzielle Sicherheitsrisiken und Bedrohungen mit sich bringen, die es zu identifizieren und zu bewerten gilt. Da Richtlinien das Verhalten des Netzwerks steuern, könnten bösartige oder fehlerhafte Richtlinien zu Netzwerkausfällen, Datenschutzverletzungen, unerlaubtem Zugriff oder anderen Sicherheitsproblemen führen. Diese Bachelorarbeit zielt darauf ab, eine sicherheitstechnische Analyse der richtlinienbasierten Steuerung bei Open RAN durchzuführen, um potenzielle Sicherheitslücken zu identifizieren und Empfehlungen für eine sichere Implementierung zu geben. Im ersten Schritt sollen in Bezug auf den Use Case und das Open RAN besonders sicherheitsrelevante Komponenten und Schnittstellen der Open RAN Architektur erkannt werden, welche anschließend in der Theorie auf mögliche Schwachstellen und daraus resultierenden Angriffsvektoren evaluiert werden. Dabei gilt es vor allem aktuelle Literatur zu berücksichtigen, aber auch eigene Ansätze einfließen zu lassen. Im Anschluss zur theoretischen Ausarbeitung soll eine praktische Umsetzung auf einer ausgewählten Open Source Implementation für Open RAN durchgeführt werden.

1.2 Zielsetzung der Arbeit

Open RAN ist ein neuer und innovativer Ansatz für Mobilfunkzugangsnetze. Es gibt bereits einige theoretische Zusammenfassungen der bisherigen O-RAN Alliance-Spezifikationen sowie theoretische Ausarbeitungen zu Schwachstellen und Bedrohungen. Darüber hinaus existieren Erkenntnisse, Lösungen und Implementierungen von xApps und rApps. Im Bereich der Sicherheit wurde bisher viel in der Theorie über Schwachstellenanalysen, Bedrohungsmodellierungen und Abwehrmechanismen diskutiert. In der praktischen Umsetzung wurden jedoch bisher nur wenige Schritte unternommen, und Evaluierungen sind begrenzt. Dies liegt unter anderem daran, dass bisher nur wenige Implementierungen von Open RAN vorhanden sind, und ihre Einrichtung in vielen Fällen einen erheblichen Aufwand erfordert.

Für Open RAN wurden verschiedene Anwendungsfälle spezifiziert, für deren Umsetzung insbesondere die Ran Intelligent Controller eine große Rolle spielen. Primär kann der Anwendungsfall der richtlinienbasierten Steuerung sehr relevante Änderungen und sensible Daten im Netzwerk durchführen und abrufen. Ebenfalls existiert eine Open-Source Lösung für Open RAN mit einer entsprechenden xApp für den Anwendungsfall Traffic Steering. Aus genannten Gründen wird sich diese Arbeit mit der Analyse des aktuellen Stands der Sicherheit in Open RAN Systemen befassen. Anschließend sollen theoretische Angriffsszenarien ausgearbeitet werden, sowie mindestens ein Angriff auf eine Open RAN Implementation mit einer xApp für die richtlinienbasierte Steuerung praktisch durchgeführt werden. Daraus sollen Fragen zu offenen Schwachstellen im Bereich der üblichen Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit geklärt werden, sowie die Bedeutung der Umsetzung von Sicherheitsprotokollen, die in Security-Spezifikationen aufgeführt sind.

1.3 Aufbau der Arbeit

Die Arbeit ist in mehrere Kapitel aufgeteilt und beginnt mit einer Einführung in den Mobilfunk sowie dessen allgemeinen Aufbau. Anschließend soll in Kapitel 2 die Architektur von Open RAN vorgestellt werden, wobei genauer auf bestimmte Komponenten und Schnittstellen eingegangen wird. Auch die Sicherheitsaspekte von Open RAN werden hier erläutert. In Kapitel 3 wird der Anwendungsfall der richtlinienbasierten Steuerung erklärt und der dazugehörige Workflow im Open RAN System. Anschließend wird das Open RAN System anhand der aktuellen Versionen der Security Spezifikationen der O-RAN Alliance auf seine Schutzmechanismen und Bedrohungen kontrolliert. Basierend auf diesen Erkenntnissen und dem Verständnis der O-RAN Architektur sollen Angriffsziele ausgearbeitet werden, die die drei zentralen Sicherheitsziele Integrität, Vertraulichkeit und Verfügbarkeit angreifen. Abschließend wird ein Angriff zur Manipulation von Richtlinien im praktischen Szenario vorgestellt.

1.4 Einführung in den Mobilfunk

Das Mobilfunknetz hat sich seit 1980 über viele Jahre und Generationen weiterentwickelt. Während anfangs nur eine analoge Kommunikation über Telefonie stattfand, bietet die 4. Generation (4G) hohe Datenraten, von bis zu 1000 Mbit/s im Downlink und 500 Mbit/s im Uplink, um modernste Dienste wie Streaming, Gaming, Voice-over-IP (Skype) und vieles mehr zu realisieren [BKM22].

Darüber hinaus sollen in der aktuellen Generation (5G) nicht nur möglichst hohe Datenraten erreicht werden, sondern auch bestimmte Anwendungsfälle erfüllt werden. Damit werden nicht nur Nutzer angesprochen, sondern auch IoT-Geräte und Szenarien, wie autonomes Fahren oder Virtual Reality. Die Nutzungsszenarien wurden zusammengefasst und in drei Schwerpunkte aufgeteilt: Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low Latency Communications (URLLC) und Massive Machine Type Communication (mMTC). Enhanced Mobile Broadband verbessert die Kommunikation von Personen an überfüllten Hotspots mit hohen Bitraten und geringer Mobilität und ermöglicht auch unterbrechungsfreie Konnektivität in einem größeren Gebiet mit moderaten Bitratenanforderungen, selbst bei hoher Mobilität. URLLC ist besonders für zeitkritische Dienste zuständig und soll eine gute Verfügbarkeit und Durchsatz bieten, für bspw. autonomes Fahren. Als Letztes bietet die mMTC eine hohe Anzahl an energieeffizienten Verbindungen in einer Zelle, wie bspw. Sensoren [Tri20].

Die grundlegende Architektur eines Mobilfunknetzes besteht aus dem Nutzer und dessen Endgerät, einem Zugangsnetz und einem Kernnetz. Das Funkzugangsnetz, auch Radio Access Network (RAN) genannt, besteht aus mehreren Basisstationen, zu denen sich die Endgeräte verbinden. Über ein Transportnetz ist das RAN mit dem Kernnetz, auch Core Network (CN) genannt, verbunden. In den folgenden Abschnitten wird näher auf diese beiden Komponenten eingegangen.

1.4.1 Das Kernnetz

Das Kernnetz in 5G entfernt sich von der herkömmlichen Architektur mit monolithischen Netzelementen. Wie in Abbildung 1.1 zu sehen, basiert das 5G Kernnetz nun auf einer Service Based Architecture (SBA), die aus mehreren feingranularen Netzwerkfunktionen besteht und einheitliche Schnittstellen besitzt. Außerdem werden die Netzwerkfunktionen in eine Control Plane (CP) und User Plane (UP) unterteilt. Dabei ist die User Plane Function (UPF) für die User Plane zuständig und die weiteren Netzwerkfunktionen, welche in Abbildung 1.1 im Kasten Control Plane zu sehen sind, für die Control Plane.

Die Schnittstelle der Netzwerkfunktionen wird jeweils über eigene APIs erfüllt, welche die Service-based Interfaces sind und die Dienste ihrer Netzwerkfunktionen bereitstellen. Eine API (Application Programming Interface) ermöglicht den Informationsaustausch zwischen verschiedenen Applikationen, ohne dass diese Applikationen ihren Code offenlegen müssen. Die Hauptrolle spielt dabei die Network Repository Function (NRF) bei der Dienste registriert, deregisriert und abgefragt werden können. Des Weiteren sind N1 bis N6 die sogenannten Point-to-Point Schnittstellen (P2P Interfaces) von den dargestellten Entitäten, wie einem User Equipment (UE), dem RAN, einer UPF und dem Data Network (DN), welches die Applikationen und eigentlichen Dienste für das UE bereitstellt. Das Kernnetz bietet somit eine gewisse Offenheit und Modularisierung, welche zur Virtualisierung gut geeignet ist und die in Abschnitt 1.4 beschriebenen Anprüche und Schwerpunkte von 5G umsetzen kann [Tri20].

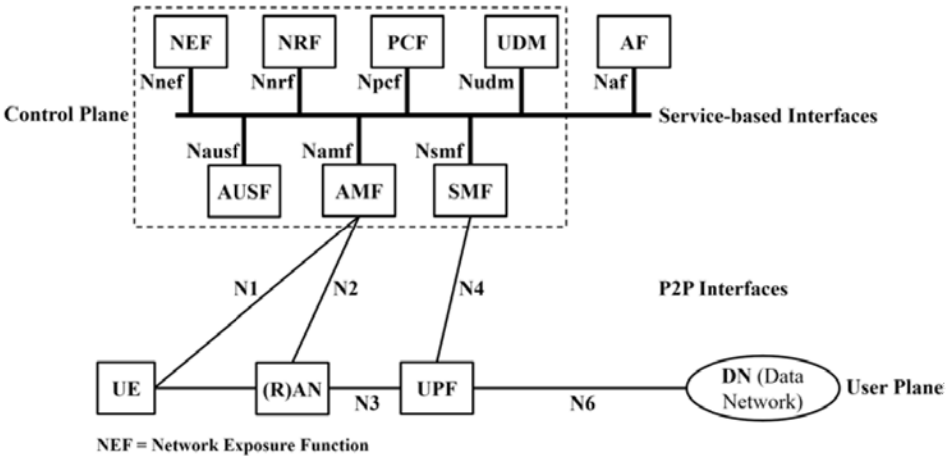

Abbildung 1.1: SBA Architektur des Kernnetzes [Tri20]

1.4.2 Das Funkzugangsnetz

Das Funkzugangsnetz besteht hauptsächlich aus einer Basisstation, welche im LTE e-NodeB (eNB) und bei 5G g-NodeB (gNB) genannt wird. Sie bildet eine Schnittstelle zwischen dem Netzwerk und den Endgeräten. Das Funkzugangsnetz von 5G kann Non-Standalone (NSA) oder Standalone (SA) betrieben werden. In der NSA-Variante existiert neben dem Next Generation RAN von 5G (NG-RAN) auch das RAN von 4G (Evolved Universal Terrestrial Radio Access) und beide sind zum Kernnetz von 4G, dem Evolved Packet Core (EPC) verbunden. Diese Variante erlaubt eine sogenannte Multi-Radio Dual Connectivity (MR-DC), wo Nutzer sich über 4G und 5G verbinden können. In der SA Variante gibt es nur das NG-RAN, welches zum 5G Kernnetz, dem Next Generation Core (NGC), verbunden ist.

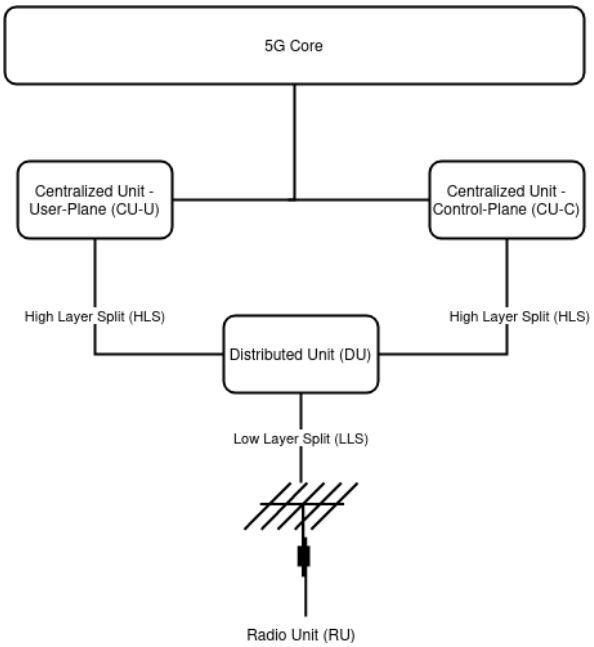

Abbildung 1.2: Aufbau des Funkzugangsnetzes in 5G angelehnt an [Tri20]

Des Weiteren kann man in Abbildung 1.2 sehen, dass eine Modularisierung vorgenommen wurde und das RAN in eine Radio Unit (RU), eine verteilte Distributed Unit (DU) und eine zentrale Central Unit (CU) aufgesplittet ist. Der Split zwischen der RU und der DU/CU nennt sich Low Layer Split (LLS), während der Split zwischen RU/DU und CU bzw. DU und CU sich High Layer Split (HLS) nennt. Diese Komponenten können verschieden kombiniert und an verschiedenen Orten verwaltet werden. DU und CU können auch softwarisiert und virtualisiert werden. Außerdem wird die CU in zwei Einheiten aufgeteilt, die entsprechend die User Plane (UP) bzw. die Control Plane (CP) verarbeiten. Somit werden verschiedene Anforderungen an 5G wie Dual-Connectivity, Virtualisierung und Softwarisierung, Modularisierung, heterogene Technik und Abwärts- sowie Aufwärtskompatibilität erfüllt [Tri20].

2 Open RAN

2.1 Allgemeine Architektur von Open RAN

Das Open RAN (O-RAN) Netzwerk besteht aus verschiedenen Komponenten und Schnittstellen, die diese miteinander verbinden. Der Kerngedanke des Open RAN ist, offene Schnittstellen zwischen den Komponenten einzuführen, sodass diese herstellerunabhängig miteinander kommunizieren können. Außerdem soll eine intelligente Steuerung und Optimierung des Netzwerks möglich sein.

In Abbildung 2.1 sind alle Komponenten und Schnittstellen des Open RAN aufgeführt. Zu den Hauptkomponenten gehört das Service Management and Orchestration (SMO) Framework, die Open RAN Central Unit auf der Control Plane (O-CU-CP) und User Plane (O-CU-UP), sowie die Open RAN Distributed Unit (O-DU) und die Open RAN Radio Unit (O-RU). Als neue Technologie im Open RAN wurden zwei logische Controller zur Optimierung des RAN eingeführt. Der Non-Real-Time RAN Intelligent Controller (Non-RT RIC) und der Near-Real-Time RAN Intelligent Controller (Near-RT RIC). Der Non-RT RIC erfüllt Aufgaben in einem Zeitrahmen von 1s und länger, während der Near-RT RIC Aufgaben in einem Zeitrahmen von 10ms bis 1s ausführt.

Als Hauptschnittstellen im Open RAN zählen die O1, O2, E2 und A1 Schnittstelle. Sie verbinden unterschiedliche Funktionen miteinander und werden von der O-RAN Alliance spezifiziert. Weitere Schnittstellen, welche auch in Abbildung 2.1 aufgeführt sind, sind die F1-c, F1-u, X2-c, X2-u, NG-u, NG-c, Xn-c, XN-u und E1 Schnittstelle. Diese sind von der 3GPP spezifiziert und stellen Verbindungen zum Kernnetz und anderen RANs her [All23e].

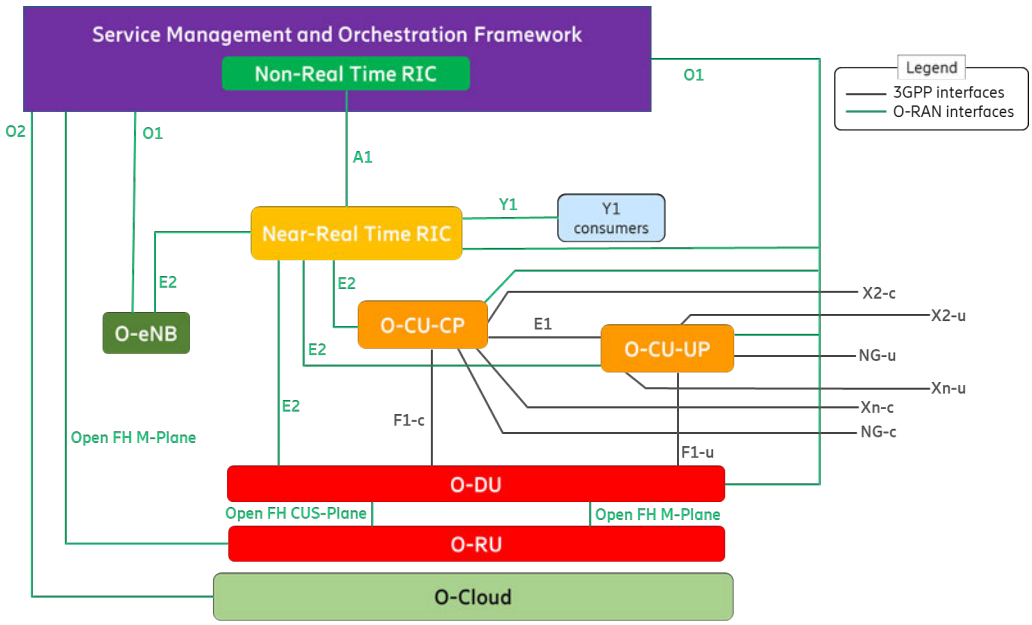

Abbildung 2.1: O-RAN Architektur [All23e]

2.2 Wichtige Komponenten im Open RAN

2.2.1 Das SMO Framework

Ein Service Management and Orchestration (SMO) Framework ist allgemein zuständig für Domain-Management und Operation, Administration und Maintenance (OAM) Aufgaben. Im Open RAN kümmert es sich um das RAN Domain-Management. Die Hauptfunktionen des SMO sind FCAPS Aufgaben, die Orchestrierung des Open RAN und eine RAN Optimierung mithilfe des Non-RT-RICs. FCAPS Aufgaben stellen dabei folgende Managementfunktionen dar: Fault Management, Configuration Management, Accounting Management, Performance Management und Security Management. Das SMO Framework besitzt vier Schnittstellen zu verschiedenen RAN-Elementen. Dazu zählen die A1 Schnittstelle zwischen dem Non-RT RIC und dem Near-RT RIC, die O1 Schnittstelle zwischen dem SMO und mehreren O-RAN Netzwerkelementen für den FCAPS Support (CU, DU), die Open Fronthaul M-Plane Schnittstelle zur O-RU und die O2 Schnittstelle zur O-Cloud.

Abgrenzungen zum integrierten Non-RT RIC sind nicht definiert und können individuell definiert werden. In Abbildung 2.2 ist auch die R1 Schnittstelle zu sehen, welche die Verwendung von rApps ermöglicht, und entweder über das Non-RT RIC oder das SMO zur Verfügung gestellt werden kann [All23e].

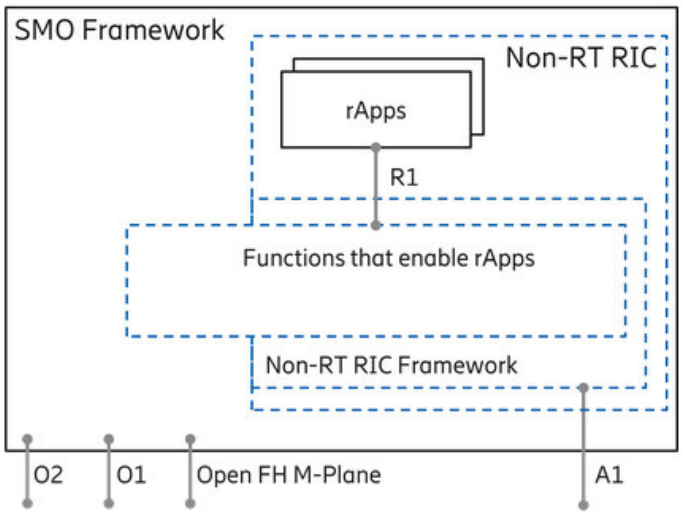

Abbildung 2.2: SMO und Non-RT RIC [All23e]

2.2.2 Der Non-RT RIC und die A1 Schnittstelle

Der Non-RT RIC ist ein intelligenter Controller, der in das SMO Framework eingebettet ist. Er sorgt für eine intelligente Optimierung des RAN und übernimmt Teilaufgaben des SMO Frameworks. Er terminiert die A1 Schnittstelle zum Near-RT RIC und bietet dadurch verschiedene Funktionen wie z.B. richtlinienbasierte Steuerung des RAN, das Bereitstellen von angereicherten Informationen und verfügt über Möglichkeiten künstliche Intelligenz (AI) und maschinelles Lernen (ML) zu verwenden. AI/ML kann dabei bspw. für eine automatisierte Anpassung von Richtlinien für die richtlinienbasierte

Steuerung verwendet werden. Der Non-RT RIC kann auf Funktionen des SMO Frameworks zugreifen und diese auch beeinflussen, wie bspw. Daten, die über die O1, Open FH M-plane und O2 Schnittstelle transportiert werden.

Der Non-RT RIC besteht aus zwei Komponenten, welche auch in Abbildung 2.2 zu sehen sind. Zum einen besteht er aus dem Non-RT RIC Framework, welches die A1 Schnittstelle terminiert und R1 Dienste für die rApps bereitstellt und zum anderen aus den rApps. Diese kümmern sich um die Verwaltung von Funkressourcen, Datenanalysen und die Bereitstellung von angereicherten Informationen. Um diese Dienste zur Verfügung zu stellen, nutzen Sie die Funktionalitäten des SMO/Non-RT RIC Frameworks [All23e].

2.2.2.1 Dienste und Funktionen des Non-RT RIC

Dienste sind Sammlungen von Funktionen, welche einem gemeinsamen Zweck dienen und voneinander abhängig sein können. Es gibt Dienstproduzenten, die Dienste bereitstellen und Dienstkonsumenten, die Dienste benötigen und anwenden. Dabei werden sogenannte Endpunkte zur Kommunikation verwendet, welche meist APIs sind. Dienstproduzenten und Dienstkonsumenten können dabei sowohl die rApps als auch die logischen Funktionen des SMO/Non-RT RIC Frameworks sein [All23b].

Folgende Dienste werden über die R1 Schnittstelle bereitgestellt [All23b]:

- **Service Management and Exposure Dienste:** Diese Dienste bieten verschiedene Möglichkeiten wie die Registrierung von Diensten, die Erkennung von Diensten, Authentisierung und Authentifizierung von Diensten, sowie Unterstützung zur Kommunikation.
- **Data Management and Exposure Dienste:** Dienste zur Registrierung von Daten, Suche und Erkennung von Daten, Datenanfrage, Datentransport und Anbieten von Daten. Auch eine Data-Subscription ist möglich, wodurch der Datenkonsument immer aktuelle Daten vom Datenproduzenten bekommt.
- **A1 bezogene Dienste:** Zum einen gibt es hier Dienste, welche sich auf die richtlinienbasierte Steuerung beziehen und zum weiteren gibt es Dienste, die sich auf die angereicherten Informationen beziehen.
  Für die richtlinienbasierte Steuerung gibt es Dienste, mit denen rApps die unterstützten A1-Richtlinientypen finden können. rApps können A1-Richtlinien erstellen, aktualisieren, abfragen und löschen. Es ist auch möglich, ein A1-Richtlinien-Abonnement zu erstellen, um immer aktuelle Informationen zu A1-Richtlinien zu erhalten.
  Für die angereicherten Informationen stehen Dienste zur Verfügung, damit rApps Enrichment Information types (EI types) registrieren und wieder abmelden können. Enrichment Information steht Synonym für die angereicherten Informationen und stellt auch weiterverarbeitete Informationen dar, die vom Non-RT RIC bereitgestellt werden.
- **RAN OAM bezogene Dienste:** Sorgen dafür, dass rApps bestimmte OAM Informationen über das RAN bekommen und teilweise auch Konfigurationen von Netzwerkelementen ändern können.
- **O2 bezogene Dienste:** Diese Dienste ermöglichen rApps einen Zugriff auf Informationen bezüglich der O-Cloud, sowie Änderungen an der Konfiguration der O-Cloud vorzunehmen.
- **AI/ML Workflow Dienste:** Diese Dienste stellen Möglichkeiten für das Management von AI/ML Training, wie bspw. Registrierung, Erkennung, Subscription und Autorisierung von AI/ML Training für rApps bereit.

- **rApp Management Dienste:** Erlaubt rApps das Reporten von ihrer Performance, Fehlern und Logs. Außerdem können rApps durch die Dienste Konfigurationen lesen und schreiben.

Um die vorhergehend aufgeführten Dienste umzusetzen, gibt es verschiedene Funktionen im SMO/Non-RT RIC Framework. Diese werden in 3 Kategorien eingeteilt. Es gibt Funktionen, die außerhalb des Non-RT RIC Framework verankert sind, Funktionen die im Non-RT RIC verankert sind und nicht verankerte Funktionen, die ein Teil des Non-RT RIC sein können, aber nicht müssen. In Abbildung 2.3 sind diese Kategorien und zugehörige Funktionen zu sehen und farblich dargestellt.

Folgende Funktionen werden durch den Non-RT RIC bereitgestellt [All23b]:

- **R1 service Management and exposure functions**
- **Data management and exposure functions**
- **A1-related functions**
- **RAN OAM-related functions**
- **O2-related functions** [All23b].

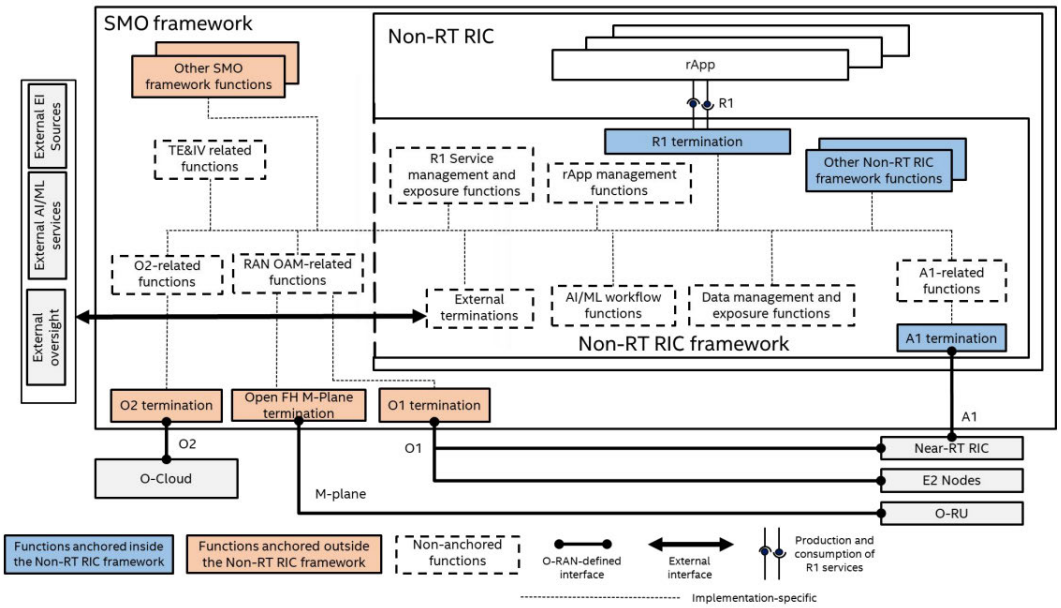

Abbildung 2.3: Architektur mit Funktionen des Non-RT RIC [All23b]

2.2.3 Der Near-RT RIC und die E2 Schnittstelle

Der Near-RT RIC ist eine logische Komponente im RAN. Near Real Time bedeutet, der RIC arbeitet in einem Control-Loop von 10ms bis 1s. In diesem Zeitfenster kann er sogenannte E2 Netzwerkelemente (Komponenten, die mit der E2 Schnittstelle verbunden sind) steuern und Informationen von ihnen abfragen.

Prinzipiell besteht der Near-RT RIC aus zwei Komponenten. Zum einen aus der Near-RT RIC Plattform, welche viele verschiedene Funktionen und Dienste bereitstellt und des Weiteren aus sogenannten xApps, welche von verschiedenen Herstellern stammen können. Diese xApps können

Informationen von der Near-RT RIC Plattform bekommen und mit diesen eine RAN-Optimierung herbeiführen, welche durch das Ansteuern der E2 Netzwerkelemente über den Near-RT RIC ausgeführt wird [All23e].

Um mit anderen Komponenten im RAN zu kommunizieren, besitzt der Near-RT RIC mehrere Schnittstellen wie in Abbildung 2.4 zu sehen ist. Zum SMO geht die O1 Schnittstelle für OAM Dienste und Informationen. Die A1 Schnittstelle verbindet den Near-RT- und den Non-RT RIC miteinander für den Austausch von Richtlinien, Enrichment Information und Richtlinien Feedback. Die E2 Schnittstelle verbindet den Near-RT RIC mit allen E2 Netzwerkelemente, welche im Open RAN die O-CU-CP und O-CU-UP, sowie die O-DU und der O-eNB/gNB sind.

Der Near-RT RIC besitzt verschiedene Funktionen, welche die xApp in ihren Aufgaben unterstützen sollen. Diese Funktionen sind ebenfalls in Abbildung 2.4 zu sehen. Folgend wird kurz auf die Funktionen eingegangen und welche Aufgaben diese erfüllen.

- **Database und Shared Data Layer (SDL):** Diese Funktion besteht aus der zugrunde liegenden Datenbank, welche aus UE-NIB und R-NIB zusammengesetzt ist. Die UE-NIB beinhaltet Informationen über die UEs, welche sich im RAN angemeldet sind. Die R-NIB hingegen stellt Informationen über Netzwerkelemente wie z.B. verschiedene Zellen bereit. Um auf diese Informationen zuzugreifen, gibt es den Shared Data Layer. Die UE-NIB stellt UE-bezogene Informationen bereit, während die R-NIB RAN bezogene Informationen beinhaltet. Der SDL sorgt dafür, die Informationen aus der UE-NIB und R-NIB via Dienste bereitzustellen.
- **xApp Subscription Management:** Führt die Subscription von xApps zu E2 Netzwerkelemente aus und bietet ein Subscription Merging, wobei mehrere xApps Subscriptions an ein E2 Netzwerkelement stellen und diese zu einer Subscription zusammengefasst werden. Diese Methode soll vermeiden, dass das Netz zu sehr ausgelastet wird.
- **Conflict Mitigation:** Soll Konflikte zwischen verschiedenen xApps erkennen und wenn möglich lösen. Dabei wird unterschieden zwischen direkten, indirekten und impliziten Konflikten. Direkte Konflikte sind offensichtliche Konflikte, bei denen zwei oder mehr xApps direkt miteinander in Konflikt stehen. Indirekte Konflikte sind Konflikte, bei denen die xApps nicht direkt miteinander in Konflikt stehen, aber ihre Handlungen oder Entscheidungen indirekte Auswirkungen aufeinander haben. Implizite Konflikte sind nicht explizit definiert, sondern treten aufgrund von unerwarteten Interaktionen oder systemweiten Auswirkungen auf. Die direkten und indirekten Konflikte sind relativ simpel abzufangen und zu lösen, während die impliziten Konflikte ein Problem darstellen, weil sie kaum zu erkennen oder vorherzusagen sind.
- **Messaging Infrastructure:** Stellt einen Low-Latency Messaging Dienst für die Endpunkte im Near-RT RIC zur Verfügung.
- **Interface Termination:** Dies sind die jeweiligen Terminatoren der Schnittstellen O1, A1, Y1 und E2. Diese dienen dazu, verschiedene Informationen mit anderen Netzwerkelementen auszutauschen und bereitzustellen.
- **API Enablement:** Bietet eine Unterstützung für die Registrierung, Erkennung und Nutzung von Near-RT RIC APIs innerhalb des Near-RT RIC.
- **AI/ML Support:** Ermöglicht eine Datenverarbeitung via AI/ML-Mechanismen. Dafür werden Daten von E2 Netzwerkelemente, A1 Enrichment Information, Informationen von xApps und der Messaging Infrastructure gesammelt und für das Trainieren von AI/ML verwendet.

- **xApp Repository function:** Wählt die xApps aus, welche für die aktive Richtlinie und/oder Betreiberkonfiguration benötigt wird. Stellt der A1 Schnittstelle die vom Near-RT RIC unterstützten Policy Types zur Verfügung [All23k].

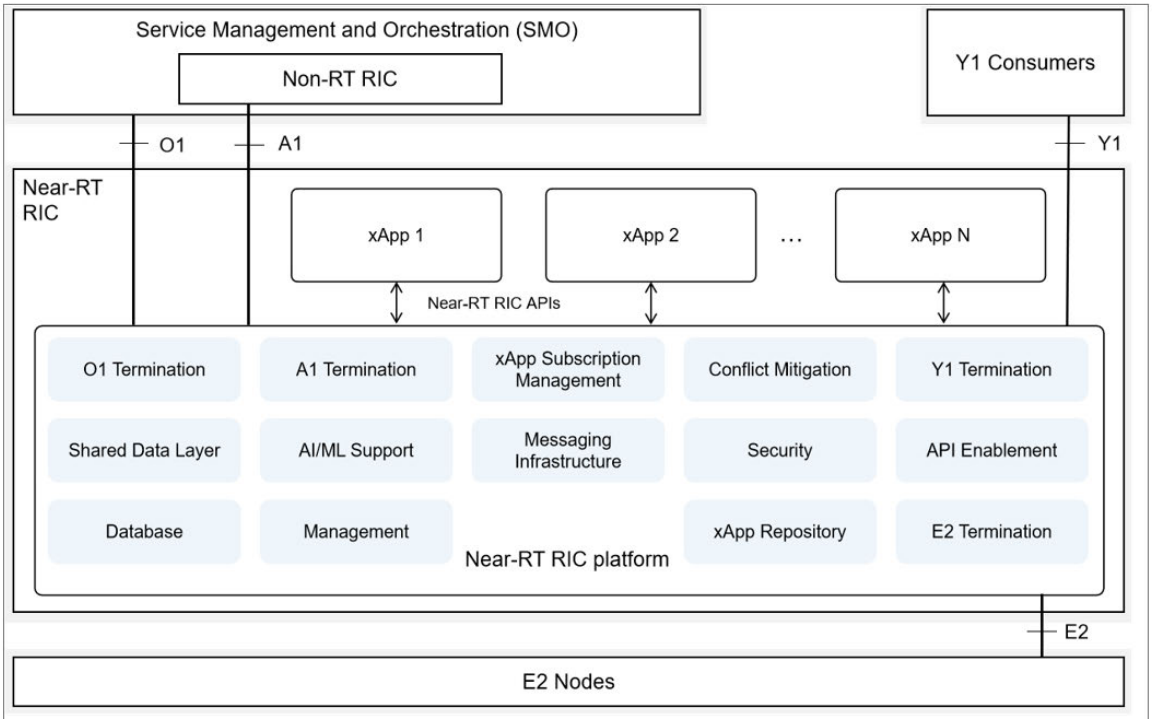

Abbildung 2.4: Near-RT RIC Architektur [All23k]

2.2.4 Die E2 Netzwerkfunktionen

Die zentralisierte Einheit O-CU führt die Protokolle Radio Resource Control (RRC), Packet Data Convergence Protocol (PDCP) und für gNBs im 5G-Netz zusätzlich das Service Data Adaption Protocol (SDAP) Protokoll aus und terminiert die F1 Schnittstelle zur O-DU um diese zu kontrollieren und zu steuern. Die O-CU wird in die Control- und User Plane gesplittet. Die O-CU-CP übernimmt die Protokolle RRC und den Control-Plane Part von PDCP. Die O-CU-UP übernimmt den User-Plane Part des PDCP für en-gNBs und zusätzlich das SDAP Protokoll für gNBs im 5G Netz [3GP23]. Weitere Schnittstellen sind die E1 Schnittstelle zwischen O-CU-CP und O-CU-UP, die E2 Schnittstelle zum Near-RT RIC, die O1 Schnittstelle zum SMO, die NG Schnittstelle zum Kernnetz, die X2 Schnittstelle zu eNBs oder en-gNBs, die Xn Schnittstelle zu gNBs oder ng-eNBs. Wobei F1, NG, X2 und Xn Schnittstellen entsprechend mit dem Kürzel -c oder -u für die Control- bzw. User Plane vorhanden sind (Bsp. NG-c, NG-u) [All23e].

Die verteilte Einheit O-DU ist eine Komponente im Open RAN, welche sich um die Radio Link Control (RLC), Medium Access Control (MAC) und den High-Physical Layer (High-PHY) kümmert. Sie wird teilweise über die F1 Schnittstelle von einer O-CU gesteuert und unterstützt das Management einer oder mehrerer O-RUs über die Open FH Schnittstelle [3GP23].

Die Funkeinheit O-RU erfüllt die Low-Physical Layer (Low-PHY) Funktionen zum UE und terminiert die Open FH Schnittstelle zur O-DU. Während sie zur O-DU die M-Plane und CUS-Plane der Open-FH Schnittstelle bereitstellt, terminiert sie zum SMO nur die M-Plane der Open-FH Schnittstelle [All23e].

2.3 Sicherheitsaspekte im Open RAN

2.3.1 Schutzziele und Zero-Trust

Ähnlich wie andere Netzwerke und Architekturen in der Informationstechnologie, hat auch Open RAN die typischen Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit [Köp+22].

- **Vertraulichkeit:** Nur autorisierte und berechtigte Nutzer dürfen Informationen und Daten erhalten.
- **Integrität:** Daten müssen vollständig und korrekt sein, und eine Manipulation derselben muss erkannt werden können.
- **Verfügbarkeit:** Alle Daten und Informationen müssen jederzeit abrufbar sein, sobald ein Nutzer darauf zugreifen möchte. Das System sollte so ausgelegt sein, dass es gegenüber bspw. DDoS-Angriffen ausfallsicher ist.

Durch verschiedene Protokolle und Vorschriften sollen diese Schutzziele eingehalten werden. In 5G, und speziell für Open RAN, übernimmt diese Aufgabe die O-RAN Alliance mit ihrer Arbeitsgruppe WG11, welche in Abschnitt 2.3.2 vorgestellt wird.

Neben den Schutzzielen sollte ein Open RAN-System auch die Grundzüge einer Zero-Trust-Architektur besitzen. Das NIST (National Institute of Standards and Technology) definiert in ihrem Standard [Ros+20] das Zero-Trust-Prinzip, welches eine Autorisierung und Authentifizierung bei jedem Kommunikationsaustausch zwischen verschiedenen Netzwerkelementen vorsieht. Dabei gelten die Netzwerkressourcen als höchstes Sicherheitsgut, wodurch keiner Komponente zur Weiterleitung von Daten vertraut werden darf.

2.3.2 Spezifikationen der O-RAN Alliance

Die O-RAN Alliance hat eine eigene Arbeitsgruppe für die Sicherheit im Open RAN eingeführt, die WG11 - Security Work Group. Diese umfasst derzeit 12 Spezifikationen zum Thema Security im Open RAN. Die wichtigsten Spezifikationen sind die O-RAN Security Requirement Specification, O-RAN Security Protocols Specification und die O-RAN Security Threat Modeling and Remediation Analysis Specification.

Aus diesen drei Spezifikationen ergeben sich Security Tests mit passenden Threats, Requirements und einer detaillierten Test-Prozedur. Die Security Requirements und Security Controls sind bereits für alle Komponenten relativ umfangreich ausgeführt und aufeinander abgestimmt. Ein großes Problem besteht jedoch noch in der Security-Test Spezifikation, die für einige Komponenten entweder nur einen Verweis auf Protokolltests enthält oder sogar gar keine vorhandenen Security-Tests aufweist.

Insbesondere Komponenten wie die beiden RAN Intelligence Controller und die jeweiligen xApps oder rApps besitzen keine Security Tests, obwohl hier eine erhebliche Gefahr für die Security besteht. Dies liegt daran, dass in diesen Bereichen unter anderem Apps von Dritt-Anbietern bereitgestellt werden können, und diese Applikationen sehr nah an der Netzstruktur arbeiten und diese verwalten.

- **Security Anforderungen:** Diese Spezifikation beleuchtet die Sicherheitsanforderungen gemäß der Version 6.0. Hier werden die spezifischen Sicherheitsanforderungen für O-RAN detailliert beschrieben, um sicherzustellen, dass das Netzwerk gegen verschiedene Bedrohungen geschützt ist. Dies umfasst Anforderungen an Vertraulichkeit, Integrität, Verfügbarkeit sowie Maßnahmen gegenüber spezifischen Bedrohungen [All23i].
- **Security Protokolle:** Hier werden die Sicherheitsprotokolle gemäß der Version 6.0 Spezifikation beleuchtet. Der Fokus liegt auf Protokollen wie TLS, SSH, IPsec und anderen, die für die Sicherheit des O-RAN-Netzwerks von entscheidender Bedeutung sind. Jedes Protokoll wird detailliert erläutert, einschließlich seiner Anwendungsbereiche und Sicherheitsziele [All23h].
- **Analyse von Gefahren und Gegenmaßnahmen:** Diese Spezifikation widmet sich der Gefahrenmodellierung und der Analyse von Gegenmaßnahmen gemäß der Version 6.0. Es werden potenzielle Bedrohungen und Schwachstellen im Open RAN identifiziert. Darüber hinaus werden effektive Strategien zur Abwehr und Beseitigung dieser Bedrohungen vorgestellt [All23g].
- **Security Tests:** Hier werden die Testspezifikationen für die Sicherheit im O-RAN-Netzwerk gemäß der Version 4.0 beleuchtet. Dieser Abschnitt beschäftigt sich mit den Methoden und Techniken, um sicherzustellen, dass die Sicherheitsmaßnahmen gemäß den Spezifikationen wirksam implementiert sind. Es umfasst auch Testverfahren zur Überprüfung der Reaktion des Systems auf bekannte Bedrohungen [All23f].

2.3.3 Aktueller Stand der Forschung

Open RAN ist sehr aktuell, aber auch ein äußerst modernes Konzept. Aufgrund dessen gibt es viele theoretische Forschungen, jedoch wenig praktische Erhebungen. Einige wissenschaftliche Arbeiten konzentrieren sich größtenteils auf eine Einführung und Vorstellung des Konzepts sowie der Architektur von Open RAN in Bezug auf die Spezifikationen der O-RAN Alliance [SSK20; Thi+23]. Eine sehr detaillierte und umfangreiche Einführung zum Open RAN bietet [Pol+23], welche auch Themen wie Workflows, maschinelles Lernen, vorhandene Testbeds und Sicherheit im Open RAN behandelt. Dabei wird auf verschiedene Stakeholder, Bedrohungen und potenzielle Gegenmaßnahmen im Bereich der Sicherheit von Open RAN eingegangen.

Neben den eher allgemeinen Arbeiten gibt es auch einige, die sich mit der Theorie rund um Security und Best-Practice-Prinzipien beschäftigen. Dabei ist zum einen die sehr umfangreiche Risikoanalyse zur Sicherheit im Open RAN vom Bundesamt für Sicherheit in der Informationstechnik (BSI) [Köp+22] zu erwähnen. Diese behandelt verschiedene Stakeholder und Angreifertypen und analysiert Szenarien sowohl im Best-Case (alle optionalen Sicherheitsmaßnahmen sind aktiviert) als auch im Worst-Case (alle optionalen Sicherheitsmaßnahmen sind deaktiviert). Das BSI kam zu dem Entschluss, dass das Sicherheitsrisiko im Open RAN mittel bis hoch ist, und gibt am Ende ihrer Studie Verbesserungsvorschläge. Sehr aktuell ist das O-RAN Security Whitepaper von Rimedo-Labs [Dry23]. Dabei liegt das Augenmerk auf den aktuellen Trends in Bezug auf Security im Open RAN und den Meinungen verschiedener Stakeholder, wie zum Beispiel Betreiber, internationale Organisationen und sogar Regierungen. Des Weiteren gibt es viele weitere Papers, die sich mit

dem aktuellen Sicherheitsstand beschäftigen und Bedrohungen sowie Gegenmaßnahmen erörtern [Kle+22; Rah+22; Mim+22; Gro+23]. Einen besonderen Standpunkt nimmt das Paper Open RAN security: Challenges and opportunities [Liy+23] ein, weil es viel umfangreicher als die vorher genannten Arbeiten ist und einen detaillierten Einblick in die Sicherheit von Open RAN-Systemen gibt.

Erste speziellere Forschungen haben sich mit der Open-FH-Schnittstelle zwischen der RU und DU beschäftigt, weil dort sehr sensible Informationen ausgetauscht und gleichzeitig hohe Leistungsanforderungen erfüllt werden müssen. Die folgende Arbeit hat die Schwachstellen und Bedrohungen der Open-FH-Schnittstelle untersucht, MACsec als Sicherheitslösung vorgeschlagen, verschiedene Implementierungen vorgestellt und einem Machbarkeitstest unterzogen [DB23]. Eine ähnliche Studie untersucht ebenfalls die Sicherheit, wobei der Fokus auf der Open-FH-Schnittstelle liegt. Als Gegenmaßnahme zu Bedrohungen werden zertifikatsbasierte Authentifizierung, IP-basierte Authentifizierung (IPsec) oder portbasierte Authentifizierung (EAP) vorgeschlagen [AM23].

Zur Detektion von Denial-of-Service Attacken (DoS) wurde eine xApp entwickelt, die durch maschinelles Lernen Angriffe an der DU erkennen und abfangen kann. Dies wurde sogar auf einem realitätsnahen Testbed durchgeführt und erkennt Angriffe mit einer Wahrscheinlichkeit von 95 Prozent [Xav+23]. Auch Signaling-Storm Attacken, eine Form des DoS-Angriffs, können durch maschinelles Lernen erkannt werden [HK23].

Wie zu erkennen ist, gibt es wenig Forschung zur praktischen Umsetzung der Theorie im Hinblick auf Schwachstellen und Gegenmaßnahmen. An diese Forschungslücke soll diese Arbeit anknüpfen und Angriffsszenarien in der Theorie, sowie praktisch erörtern und umsetzen.

3 Die richtlinienbasierte Steuerung

3.1 Erklärung des Konzepts

Die richtlinienbasierte Steuerung ist ein spezieller Use Case, der von der O-RAN Alliance definiert wurde. Dabei können viele mögliche Szenarien umgesetzt werden. Beispielsweise sorgt Traffic Steering für einen Lastausgleich bei Zellen. Wenn der Non-RT RIC erkennt, dass eine Zelle viel mehr Last trägt als seine Nachbarzellen, kann er einen Prozess einleiten, um UEs auf die weniger belastete Zelle zu verlagern, wodurch eine optimale Ressourcennutzung gewährleistet wird. Ein weiteres Szenario wäre, die UEs aufgrund ihrer Quality of Service (QoS) Anforderungen entsprechenden Zellen zuzuweisen. QoS-Anforderungen können sich auf spezifische UEs beziehen, sei es für Telefondienste oder für die ausschließliche Nutzung von Breitbanddiensten. Ein drittes Szenario könnte sein, UEs zu den jeweiligen Zugriffstechnologien zuzuweisen, wenn Dual-Connectivity verwendet wird [All23l]. Dual-Connectivity bedeutet das sowohl 4G- wie auch 5G-Zugangsnetze verwendet werden und zu einem gemeinsamen Kernnetz weitergeleitet werden. Entsprechend können in diesem dritten Szenario die UEs auf das 4G- oder 5G-Zugangsnetz geleitet werden.

3.2 Workflow im Open RAN System

Der Workflow vom Anwendungsfall Traffic Steering ist in Abbildung 3.1 abgebildet. Das SMO Framework sammelt via O1 Schnittstelle Informationen über das Netzwerk und gibt diese an den Non-RT RIC weiter. Dieser gleicht mithilfe der Netzwerkinformationen die Performance des Systems mit dem RAN-intent ab und entscheidet, ob eine Richtlinie erstellt werden muss. Das RAN-intent ist sozusagen ein vorgegebenes Ziel des Betreibers eines Mobilfunknetzes, wie das Netz im Optimal laufen sollte. Wenn dies der Fall ist, wird eine Richtlinie erstellt und über die A1 Schnittstelle an den Near-RT RIC weitergegeben. Der Near-RT RIC setzt die Richtlinie mithilfe der xApp um und steuert entsprechend die E2 Funktionen an. Das SMO Framework und der Non-RT RIC sammeln weiterhin Informationen über das Netzwerk und übergeben diese an den Non-RT RIC. Der Non-RT RIC kann nun entscheiden, ob die Richtlinie weiter ausgeführt werden soll oder gelöscht werden kann, weil die Performance des Netzwerks dem RAN-intent entspricht. Soll die Richtlinie gelöscht werden, muss der Non-RT RIC dem Near-RT RIC dies mitteilen und die Richtlinie wird im Near-RT RIC gelöscht. Der Near-RT RIC kann dem Non-RT RIC über die A1 Schnittstelle Feedback über den Status der Ausführung von Richtlinien geben [All23l].

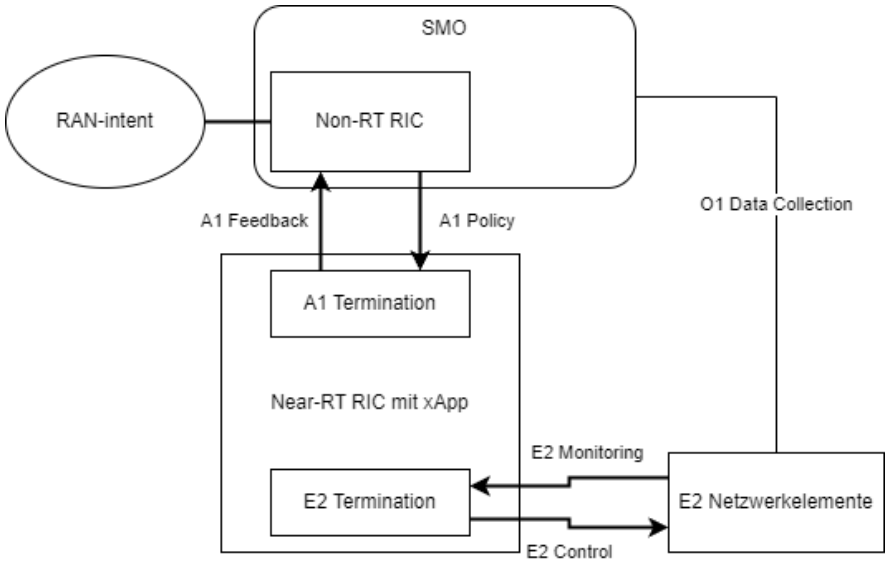

Abbildung 3.1: Workflow Traffic Steering in Anlehnung an [All23l]

3.3 Implementierung durch Rimedo Labs

Rimedo-Labs ist eine ausgelagerte Organisationseinheit des polnischen Instituts für Funkkommunikation an der Technischen Universität Posen in Polen. Sie sind Mitglied der Open Network Foundation (ONF) und der O-RAN Alliance und widmen sich hauptsächlich der Entwicklung von xApps für Open RAN. Rimedo-Labs hat eine eigene xApp entwickelt, um eine praktische Umsetzung und Evaluation von Traffic Steering zu ermöglichen [Lab].

Ihre Traffic Steering xApp ermöglicht einen intelligenten Lastausgleich von Funkzellen, eine dienstbasierte Zuordnung von Endgeräten zu Funkzellen und eine energieeffiziente Nutzung des RAN. Dabei orientiert sich Rimedo-Labs bei der Implementierung sehr nah an den Spezifikationen der O-RAN Alliance. So empfängt die xApp ihre Richtlinien vom Non-RT RIC über die A1 Schnittstelle. Auch die Richtlinie wird mit dem entsprechenden Policy Type „Traffic Steering Preferences “ übermittelt. Diese Richtlinien werden mit einer bestimmten Syntax im JSON Format übermittelt, welche in Abbildung 3.2 zu sehen ist.

Die Richtlinie ist in 2 Teile aufgebaut, den Scope und die tspResources. Im Scope werden der UE-identifier (Zeile 3) und Zusätzlich den QoS-Dienst angegeben. Wenn es z.B. um den Dienst Telefonie für das jeweilige UE gehen soll, muss der entsprechende qos-identifier angegeben werden, was in diesem Fall eine 1 ist (Zeile 4). Danach kommen die Traffic Steering Preferences-Resources (tspResources) bei denen die Liste der Zellen (CellIdList), die Präferenz, und optional ein Parameter primary angegeben werden kann. Die Liste der Zellen beinhaltet pro Zelle immer die Cell Global Identity (CGI), welche sich aus Public Land Mobile Network Identity (PLMN ID) und New Radio Cell Identity (NCI) zusammensetzt. Die Präferenz gibt an, wie das UE sich in Bezug auf die angegeben Zellen verhalten soll. Dabei kann aus 4 Zuständen gewählt werden:

- SHALL - Die angegebenen Zellen auswählen
- PREFER - die angegebenen Zellen priorisieren
- AVOID - die angegebenen Zellen vermeiden

- FORBID - die angegeben Zellen auf keine Fall auswählen

Bei SHALL und FORBID werden die Zellen als gleich wichtig angesehen, wohingegen bei PREFER und AVOID die angegebene Zellen der Reihenfolge nach unten mehr bzw. weniger priorisiert/vermieden werden. Das primary Attribut soll angeben ob die angegeben Zellen Primärzellen oder Sekundärzellen sind. Die Primärzelle ist sozusagen die Zelle worüber die Hauptverbindung läuft und die Sekundärzelle unterstützt diese zur Verbesserung der Gesamtleistung [Rim23].

```
{
    "scope":{
       "ueId":"0000000009295552"
       "qosId":"1"
    },
    "tspResources":[
       {
          "cellIdList":[
             {
                "plmnId":{"mcc":"138", "mnc":"426"},
                "cId":{"ncI":12035}
             }
             {
                "plmnId":{"mcc":"138", "mnc":"426"},
                "cId":{"ncI":12034}
             }
          ],
          "preference":"FORBID"
          "primary":"false"
       }
    ]
 }
```

Abbildung 3.2: Richtlinie für Traffic Steering im JSON-Format

4 Sicherheitsanalyse für Open RAN

Um in Kapitel 5 Angriffsszenarien zu entwickeln benötigt es im Vorfeld eine Analyse von diversen Aspekten in Bezug auf Security in Open RAN Systemen. Diese Analyse soll das Identifizieren von potenziellen Angreifern, deren Angriffszielen und eine Einschätzung der Sicherheitsanforderungen für Open RAN umfassen.

4.1 Potenzielle Angreifer

Sowie jedes andere Netzwerk tauscht auch Open RAN wichtige und sensible Informationen zwischen Komponenten aus und stellt einen sehr wichtigen Dienst zur Verfügung, nämlich den Mobilfunk, den heutzutage fast jeder Mensch nutzt. Viele verschiedene Angreifertypen haben Interesse daran diese Netzwerkarchitektur in bestimmten Sicherheitszielen, wie sie in Abschnitt 2.3.1 aufgeführt sind, zu beeinträchtigen. Diese Angreifertypen sollen in diesem Kapitel kurz aufgeführt und erläutert werden [All23g; Zwa23].

1. **Externe Hacker:** Externe Hacker können einzeln oder in Gruppen agieren. Sie haben verschiedene Interessen, von finanziellen über politische, bis hin zu wirtschaftlichen Interessen. Sie versuchen, in das Netzwerk einzudringen und vertrauliche Informationen zu erlangen, die Verfügbarkeit des Netzwerks zu beeinträchtigen oder andere schädliche Aktivitäten durchzuführen.
2. **Interne Mitarbeiter:** Insider-Bedrohungen von aktuellen oder ehemaligen Mitarbeitern können eine ernsthafte Gefahr darstellen. Dies kann unbeabsichtigte Fahrlässigkeit oder absichtliche Handlungen umfassen, die auf persönlichen Gründen, finanziellen Anreizen oder Rache basieren.
3. **Insider durch Drittanbieter:** Drittanbieter oder externe Dienstleister könnten als potenzielle Angreifer fungieren, insbesondere wenn sie privilegierten Zugriff auf das Open RAN-Netzwerk haben.
4. **Konkurrenzunternehmen:** Andere Unternehmen im Sektor des Mobilfunks können Interesse daran haben, dem Ruf oder der Verfügbarkeit des Netzes eines anderen Anbieters zu schaden.
5. **Staatlich gesponserte Angreifer:** Nationale Regierungen oder staatlich unterstützte Gruppen könnten versuchen, Open RAN anzugreifen, um strategische oder geopolitische Ziele zu erreichen. Dies könnte Spionage, Sabotage oder Manipulation von Kommunikation umfassen.

4.2 Angriffsziele und Bedrohungen

4.2.1 Kritische Angriffsziele im Open RAN

Es gibt viele verschiedene Angriffsziele, welche durch die Verletzung der Sicherheitsziele Integrität, Vertraulichkeit und Verfügbarkeit erreicht werden können. Die O-RAN Alliance hat einige kritische Angriffsziele für Open RAN Systeme ausgearbeitet und dargestellt [All23g]. Diese sollen in diesem Abschnitt in Tabelle 4.1 vorgestellt werden. Die kritischen Angriffsziele haben jeweils eine ID, eine kurze Erläuterung, um welche Daten es genau geht und über welche Komponenten und Schnittstellen diese Daten übertragen oder verarbeitet werden.

ID	Beschreibung	O-RAN Funktionen	Schutzziele
ASSET-D-03	kritische Daten, die über die O1 Schnittstelle transportiert werden. Dazu gehören unter anderem Informationen zum aktuellen Netzstatus und Beobachtungswerte wie das Policy Feedback vom Near-RT RIC zum Non-RT RIC.	Near-RT RIC, Non-RT RIC, SMO, O-CU, O-DU, O1	Vertraulichkeit, Integrität
ASSET-D-07	A1 Policies zur RAN-Optimierung.	Near-RT RIC, Non-RT RIC, A1	Vertraulichkeit, Integrität
ASSET-D-08	A1 Enrichment Information, welche vom Non-RT RIC erstellt und im Near-RT RIC zur Verfügung gestellt werden.	Near-RT RIC, Non-RT RIC, A1	Vertraulichkeit, Integrität
ASSET-D-09	Daten, die über die E2 Schnittstelle transportiert werden. Das sind z.B. • persistente Konfigurationen des RANs • IDs von E2-Funktionen • xApp-basierende Nachrichten • Informationen der Control Plane • Richtlinien, die vom Near-RT RIC verwendet werden • Near-RT RIC Service Nachrichten	O-DU, O-CU, Near-RT RIC, E2	Vertraulichkeit, Integrität
ASSET-D-10	Datenbank, die Informationen über E2-Funktionen und xApp speichert	Near-RT RIC	Vertraulichkeit, Integrität, Verfügbarkeit
ASSET-D-11	Daten von E2-Funktionen z.B. • Konfigurationen und Informationen von Zellen • Netzwerkkennzahlen • Kontextinformationen zu E2-Funktion	E2-Funktionen	Integrität, Verfügbarkeit
ASSET-D-16	X.509 Zertifikate im O-RAN System	alle Komponenten und Schnittstellen	Integrität, Verfügbarkeit
ASSET-D-17	Private Schlüssel zur Authentifizierung, Autorisierung und Verschlüsselung	alle Komponenten und Schnittstellen	Vertraulichkeit, Integrität, Verfügbarkeit

ASSET-D-18	Informationen über O-RAN Komponenten und deren Konfiguration wie z.B. • Versionsnummern • IDs • IPs • Portnummer • unterstütze Protokolle • Datenmanagement-System • Hashwerte	alle Komponenten und Schnittstellen	Vertraulichkeit, Integrität, Verfügbarkeit
ASSET-D-20	Anmeldeinformationen	alle Komponenten und Schnittstellen	Vertraulichkeit, Integrität, Verfügbarkeit
ASSET-D-23	Patches für anfällige Software	alle Komponenten und Schnittstellen	Integrität, Verfügbarkeit
ASSET-D-24	Datenspeicher von NETCONF	alle Komponenten und Schnittstellen	Vertraulichkeit, Integrität, Verfügbarkeit
ASSET-D-25	Training- und Testdaten, die von Near-RT RIC und E2 Funktionen gesammelt werden und von ML-Prozessen genutzt werden	Near-RT RIC, Non-RT RIC, xApps, rApps, A1, O1, E2	Vertraulichkeit, Integrität, Verfügbarkeit
ASSET-D-26	das trainierte ML-Model	Near-RT RIC, Non-RT RIC, xApps, rApps	Vertraulichkeit, Integrität, Verfügbarkeit
ASSET-D-27	die ML-Vorhersageergebnisse	Near-RT RIC, Non-RT RIC, xApps, rApps	Vertraulichkeit, Integrität, Verfügbarkeit
ASSET-D-29	Log-Dateien, die bei Security Vorfällen generiert werden.	Alle Komponenten und Schnittstellen	Integrität, Verfügbarkeit
ASSET-D-30	UE IDs	Near-RT RIC, Non-RT RIC, xApps, rApps	Vertraulichkeit, Integrität
ASSET-D-39	xApp ID	Near-RT RIC, xApps	Integrität

Tabelle 4.1: Angriffsziele für Open RAN Systeme [All23g]

4.2.2 Bedrohungen und Attacken gegen Open RAN

Durch die Offenheit von Open RAN stehen diese Netzwerke vielen Bedrohungen und Attacken gegenüber. In [Liy+23] wurden diese sehr gut kategorisiert. Dabei wurden Attacken gegen Open-Source Code, Attacken gegen die Intelligenz und Attacken gegen die Virtualisierung berücksichtigt. Auch Attacken gegen die Open-FH Schnittstelle wurden extra beleuchtet, was aber kein Teil dieser Arbeit sein soll.

- **Attacken gegen Open-Source-Code:** Hier geht es um Schwachstellen im Open-Source-Code, welche oft vom Nutzer des Codes nicht geprüft und entsprechend auch nicht erkannt werden. Das können z.B. schon bekannte Schwachstellen sein oder eine Hintertür im Code, welche absichtlich vom Entwickler eingebaut wurde. Auch wenn es keine Standards für den Code gibt bzw. sich nicht an diese gehalten wird, können viele Schwachstellen in Open-Source-Code entstehen.
- **Attacken gegen die Intelligenz:** Die Intelligenz des Open RAN umfasst die beiden RAN Intelligent Controller und die jeweiligen Apps, die darauf laufen. Dabei gibt es Bedrohungen gegenüber dem Non-RT RIC und dem Near-RT RIC.
  Bedrohungen gegen den **Near-RT RIC** können z.B. das Orten von UEs oder Ändern von UE-Prioritäten, genauso wie die Identifikation von UEs über bösartige xApps sein. Des Weiteren bestehen auch Gefahren durch Fehlkonfigurationen oder Konflikte zwischen xApps.
  Bedrohungen gegen den **Non-RT RIC** sind DDoS Attacken, Sniffing auf der A1 Schnittstelle zum Auslesen von sensiblen Informationen, Schwachstellen und Fehlkonfigurationen von rApps, schwache Authentifizierung und Autorisierung zwischen Komponenten, die mit dem Non-RT RIC kommunizieren oder Konflikte zwischen rApps.
  Neben den Angriffen auf die beiden RICs sind auch generelle Angriffe gegen Machine Learning (ML) eine große Bedrohung für ein Open RAN. Dabei können z.B. die Daten, welche zum Lernen des ML-Algorithmus dienen, manipuliert werden oder komplett neue, aber fehlerhafte Daten eingeschleust werden. Genauso wie die Daten verändert werden können, kann auch der komplette Algorithmus verändert werden, sodass er andere Ergebnisse erzielt oder nicht mehr funktioniert.
- **Attacken gegen die Virtualisierung:** Typische Angriffe gegen die Virtualisierung von Netzwerken sind Attacken auf Schwachstellen von veralteten oder fehlkonfigurierten Images oder Man in the Middle Attacken durch das Ausnutzen von fehlender Authentifizierung oder Autorisierung. Ebenfalls können Angreifer durch die Kompromittierung einer virtuellen Funktion auf weitere Funktionen ausbrechen und sogar auf den Hypervisor, also dem Host-Client der Virtualisierung, Zugriff bekommen.

Dies sind die wichtigsten und gefährlichsten Attacken gegen Open RAN, wobei weitere Angriffsmöglichkeiten bestehen und in Abbildung 4.1 aufgeführt sind.

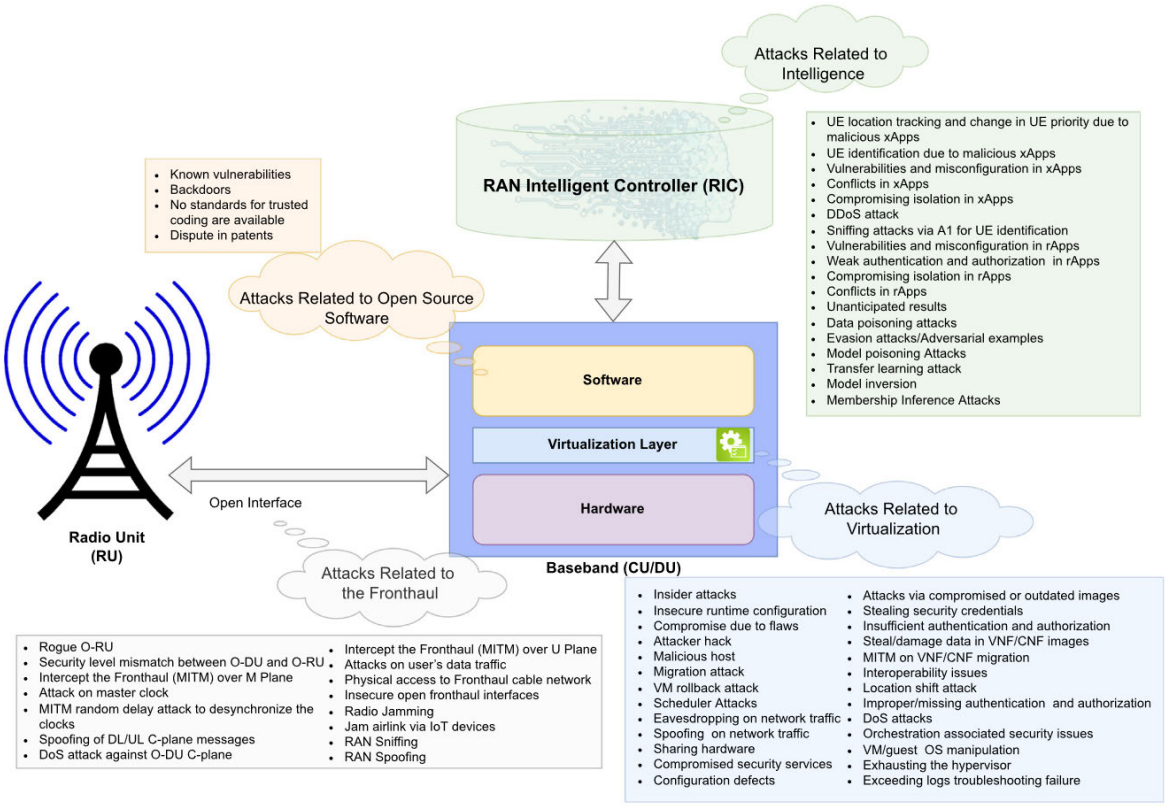

Abbildung 4.1: Bedrohungen und Attacken gegen Open RAN [Liy+23]

4.3 Anforderungen und Sicherheitsmaßnahmen von Funktionen im Open RAN

Im folgenden Kapitel geht es darum, die Sicherheitsanforderungen von Open RAN Funktionen zu identifizieren und ihre entsprechenden Umsetzungen durch Protokolle oder andere Sicherheitsmechanismen zu erläutern. Ein gründliches Verständnis dieser Eigenschaften von Open RAN Funktionen ist entscheidend für die Entwicklung von Angriffen in späteren Kapiteln. Bei dieser Analyse sollen bestimmte Komponenten und Schnittstellen einbezogen werden. Wie bereits in Abschnitt 2.3.3 erwähnt, wurden schon einige Arbeiten für die Open-FH-Schnittstelle durchgeführt. Deshalb konzentriert sich diese Arbeit vorrangig auf den Workflow der RAN Intelligent Controller und deren Schnittstellen, insbesondere weil diese die hauptsächliche Arbeit im Hinblick auf die richtlinienbasierte Steuerung übernehmen. Folgende Komponenten und Schnittstellen sollen analysiert werden:

- Non-RT RIC
- Near-RT RIC
- A1 Schnittstelle
- E2 Schnittstelle

Wie in Abschnitt 2.3.2 schon beschrieben, befasst sich die Security Requirements Spezifikation mit den Anforderungen und Umsetzungen für die Sicherheit der Open RAN Komponenten. Dort werden für jede Komponente und Schnittstelle bestimmte Anforderungen an die Sicherheit beschrieben und anschließend sogenannte Security Controls zur Umsetzung dieser Anforderungen identifiziert.

4.3.1 Anforderungen und Sicherheitsmaßnahmen des Non-RT RIC

Der Non-RT RIC verarbeitet viele sensible Informationen und leitet diese über verschiedene Schnittstellen weiter an andere Komponenten im Open RAN. Dafür werden bestimmt Sicherheitsanforderungen und Maßnahmen benötigt, welche folgend erläutert werden [All23i].

Sicherheitsanforderungen:

- Unterstützung von Autorisierung: Der Non-RT RIC muss die Autorisierung sowohl als Ressourceninhaber/Server wie auch als Client ermöglichen. Das Non-RT RIC Framework muss als Ressourceninhaber/Server eine Autorisierung für Anfragen von rApps als Client bereitstellen. Außerdem müssen rApps Autorisierungsanfragen an das Non-RT RIC Framework senden können.
- Schutz vor DDoS-Angriffen: Der Non-RT RIC soll sich von DDoS-Angriffen über die A1-Schnittstelle erholen können, genauso wie das Non-RT RIC Framework sich von DDoS-Angriffen über die R1-Schnittstelle erholen können soll. Auch rApps sollen in der Lage sein, sich von DDoS-Angriffen über die R1-Schnittstelle zu erholen.
- Authentifizierung und Autorisierung über die R1-Schnittstelle: Das SMO/Non-RT RIC Framework muss sowohl API-Produzenten als auch API-Verbraucher über die R1-Schnittstelle mit einem auf Kafka basierenden Protokoll authentifizieren. Kafka basierende Protkolle meint hier Protokolle, die auf apache Kafke basieren. Apache Kafka ist eine event-streaming Plattform, welche mit einem hohen Datendurchsatz in Echtzeit zum verarbeiten von Big-Data Datenströmen dient [Zel22]. Das SMO/Non-RT RIC Framework muss einen Autorisierungsmechanismus für das auf Kafka basierende Protokoll unterstützen, um den Zugang für Datenübertragung durch API-Produzenten und -Verbraucher über die R1-Schnittstelle zu gewährleisten.

Sicherheitsmaßnahmen:

- A1-EI (A1 Enrichment Information): Der Non-RT RIC soll OAuth 2.0 als Ressourceninhaber/Server für Serviceanfragen von einem oder mehreren Near-RT RICs unterstützen.
- A1-P (A1 Policy): Ebenfalls soll der Non-RT RIC eine Implementierung von OAuth 2.0 als Client unterstützen.
- R1-Schnittstelle: Für die R1 Schnittstelle soll der Non-RT RIC OAuth 2.0 zur Autorisierung unterstützen und kann TLS unterstützen.

4.3.2 Anforderungen und Sicherheitsmaßnahmen des Near-RT RIC

Der Near-RT RIC erlaubt eine echtzeitnahe RAN-Optimierung und kann xApps von Drittanbietern integrieren. Er kommuniziert mit den Non-RT RIC und E2 Netzwerkfunktionen. Entsprechend laufen über den Near-RT RIC viele wichtige und sensible Informationen sowie Aktionen, die vor Angreifern geschützt werden müssen. Dafür wurden folgende Sicherheitsanforderungen und Maßnahmen von der O-RAN Alliance spezifiziert [All23i].

Sicherheitsanforderungen:

- Near-RT RIC authentifiziert den xApp-Zugriff auf Near-RT RIC-Datenbanken während der SDL-Registrierung.
- Near-RT RIC ermöglicht autorisierten Zugriff auf Datenbanken.

- Die Kommunikation zwischen xApps und den Near-RT RIC-Plattform-APIs erfolgt durch gegenseitige Authentifizierung.
- Near-RT RIC-Architektur bietet ein Autorisierungsframework für die Nutzung von Diensten durch xApps unter Berücksichtigung von Betreiberrichtlinien.
- Near-RT RIC soll Autorisierung als Resourceninhaber/Server und Client unterstützen.
- Near-RT RIC soll sich von DDoS Angriffen über die A1 Schnittstelle, ohne größere Schäden, erholen können.
- Near-RT RIC soll sich vor Angriffen wie Injections oder Buffer-Overflow (content-bezogene Angriffe) schützen können.

Sicherheitsmaßnahmen:

- API-Sicherheit - Authentifizierung: Transaktionale APIs unterstützen gegenseitige Authentifizierung über TLS mit einem X.509v3 Zertifikate. Zeitkritische APIs ohne TLS-Unterstützung verwenden IPsec mit IKEv2-zertifikatsbasierter Authentifizierung.
- API-Sicherheit - Autorisierung: Der Near-RT RIC soll Autorisierung als Server und Client unterstützen. Dabei ist die Near-RT RIC Plattform selbst dafür zuständig die Rechte für den Zugriff auf ihre Dienste zu managen.
- Transaktionale APIs sollen das OAuth 2.0 framework unterstützen.
- Für A1-P soll der Near-RT RIC OAuth 2.0 als Server und für A1-EI OAuth 2.0 als Client unterstützen.
- API-Sicherheit - Vertraulichkeit und Integrität: Transaktionale APIs sollen TLS für Vertraulichkeit und Integrität unterstützen. Zeitkritische APIs sollen IPsec für Vertraulichkeit und Integrität unterstützen.
- Zusätzliche Sicherheitskontrollen: Near-RT RIC überprüft Richtlinien von der A1-Schnittstelle und protokolliert Sicherheitsereignisse bei Fehlern. Die Y1-Schnittstelle unterstützt mTLS-Authentifizierung und das OAuth 2.0 Autorisierungsframework.

4.3.3 Anforderungen und Sicherheitsmaßnahmen der A1 Schnittstelle

Die A1-Schnittstelle verbindet den Non-RT RIC mit dem Near-RT RIC. Über diese Schnittstelle läuft ein wichtiger Datenaustausch, insbesondere im Hinblick auf die richtlinienbasierte Steuerung, bei der Richtlinien, Feedback zu den Richtlinien und Netzinformationen ausgetauscht werden. Daher sollte auch diese Schnittstelle besonders gesichert sein und besitzt laut O-RAN Alliance folgende Anforderungen und Sicherheitsmaßnahmen. [All23i].

Sicherheitsanforderungen:

- Die A1-Schnittstelle muss sicherstellen, dass die übertragenen Daten vertraulich behandelt werden und vor Manipulation geschützt sind.
- Eine gegenseitige Authentifizierung und Autorisierung zwischen den beteiligten Systemen, insbesondere dem Non-RT RIC und den Near-RT RICs, ist erforderlich.

Sicherheitsmaßnahmen:

- Die Unterstützung von TLS auf der A1-Schnittstelle gewährleistet die Vertraulichkeit der übertragenen Daten und schützt vor unbefugten Zugriffen.
- Die Unterstützung von mTLS ermöglicht eine sichere Kommunikation durch die gegenseitige Authentifizierung zwischen dem Non-RT RIC und dem Near-RT RICs.

- Die Unterstützung von OAuth 2.0 bietet einen Mechanismus für die Autorisierung, um sicherzustellen, dass nur befugte Systeme auf die Schnittstelle zugreifen können.

4.3.4 Anforderungen und Sicherheitsmaßnahmen der E2 Schnittstelle

Die E2-Schnittstelle stellt den Datenaustausch zwischen dem Near-RT RIC und den E2 Netzwerkfunktionen dar. Es werden viele sensible Informationen zum Netzwerk ausgetauscht, und auch Aktionen zur Steuerung des RAN durch die E2 Netzwerkfunktionen werden über die E2-Schnittstelle ausgeführt. Um diese vor Manipulationen und unerlaubtem Mitlesen zu schützen, wurden folgende Anforderungen und Maßnahmen an die E2-Schnittstelle spezifiziert. [All23i].

Sicherheitsanforderungen:

- Die E2 Schnittstelle soll Integrität, Vertraulichkeit, Schutz vor Replay-Angriffen, und Authentifizierung der Herkunft von Daten gewährleisten.

Sicherheitsmaßnahmen:

- Die E2 Schnittstelle ist eine Zeitkritische Shcnittstelle und sollte deshalb IPsec unterstützen.

5 Theoretische Angriffsszenarien auf Open RAN

In diesem Kapitel werden theoretische Angriffsszenarien entwickelt, um die Herausforderungen und potenziellen Risiken im Bereich der Informationssicherheit von Open RAN zu verstehen. Die Angriffsszenarien werden sich speziell mit dem Anwendungsfall Traffic-Steering, also der richtlinienbasierten Steuerung, beschäftigen. Die Angriffe werden nach einem methodischen System entwickelt, welches sich an verschiedenen bekannten Methoden für Hacking und IT-Sicherheit orientiert. Durch die systematische Planung eines solchen Angriffsszenarios kann man nicht nur potenzielle Bedrohungen erkennen, sondern auch die Effektivität von Sicherheitsvorkehrungen analysieren, was am Ende eines jeden Angriffs geschehen soll. Das methodische Vorgehen zur Entwicklung von Angriffsszenarien ist angelehnt an die fünf Phasen des Hacking, welche in [Eng21] erläutert werden. Die ersten vier Phasen bestehen aus der Informationsbeschaffung (Reconnaissance), Schwachstellenanalyse (Scanning), Einstiegspunkte finden (Gaining Access) und dem weiteren Vorgehen bzw. der Aufrechterhaltung vom Zugang (Maintaining Access). Die letzte Phase wäre das verwischen von Spuren (Clearin Tracks), welche in diesem Schema mit einer Evaluation des Angriffsszenarios und einer Diskussion über mögliche Verteidigungsmechanismen ersetzt werden soll.

In Abbildung 5.1 sind die einzelnen Phasen zur Entwicklung der Angriffsszenarien aufgeführt. Wie man sieht, ist den bisher genannten Schritten eine weitere Phase, zur Zieldefinition und zum Identifizieren von Zielobjekten, vorangestellt. Des Weiteren sollte ein Schritt die Überlegung zu möglichen Werkzeugen und Tools beinhalten, die im Rahmen des Angriffsszenarios eingesetzt werden können.

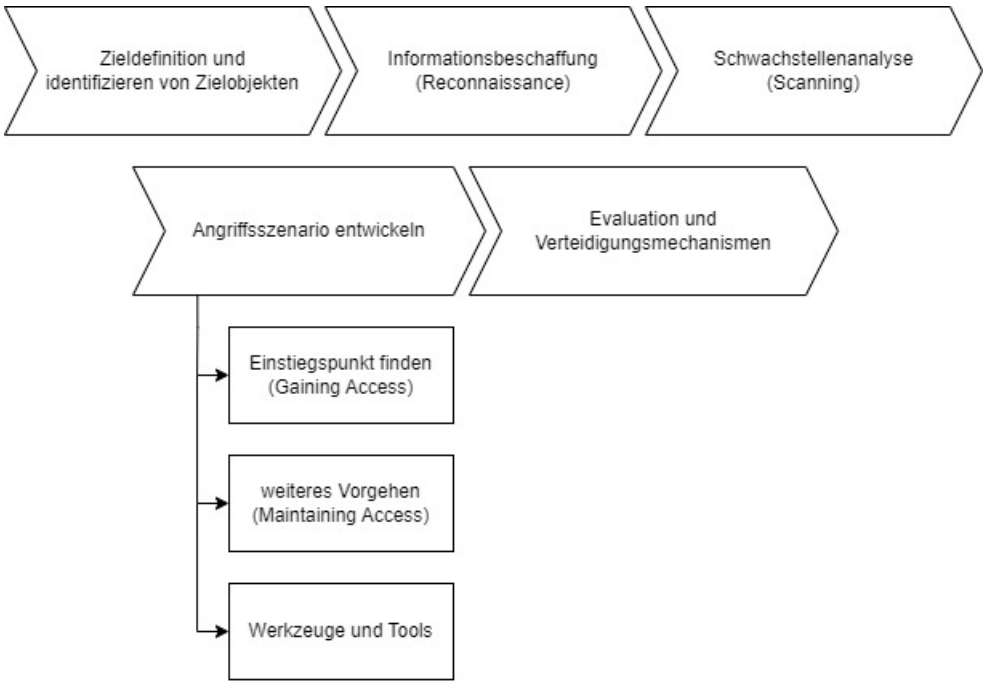

Abbildung 5.1: Methodisches Vorgehen für Angriffsszenarien angeleht an [Eng21]

5.1 Angriffsszenario 1: Manipulation einer A1-Richtlinie

5.1.1 Zieldefinition und Identifikation von Zielobjekten

Bei diesem Angriffsszenario sollen Richtlinien, welche zur Optimierung des RAN beitragen, abgefangen, manipuliert und weitergeleitet werden, um das RAN so zu steuern, wie der Angreifer möchte. Durch den Einsatz von Man-in-the-Middle-Angriffen oder dem Eindringen in die Kommunikationswege des RAN kann der Angreifer die übermittelten Richtlinien abfangen. Nach dem Abfangen kann der Angreifer die Richtlinien manipulieren, um beispielsweise die Priorisierung bestimmter Benutzer oder Dienste zu ändern oder die Zuweisung von Ressourcen zu beeinflussen. Die manipulierten Richtlinien werden dann wieder in das RAN eingespeist, möglicherweise durch Injektion in den Kommunikationsfluss, um sicherzustellen, dass die Netzwerkelemente die veränderten Anweisungen erhalten. Durch die gezielte Steuerung des RAN kann der Angreifer verschiedene Auswirkungen erzielen, wie z.B. Netzüberlastung, Beeinträchtigung der Dienstqualität für bestimmte Benutzergruppen oder sogar gezielte Störungen des Netzwerkbetriebs. Speziell in diesem Angriffsszenario soll die Richtlinie für ein spezielles UE so manipuliert werden, dass sich das UE mit keiner Zelle für die QoS-Anwendung Telefonie (5QI=1) verbinden darf und somit z.B. einen wichtigen Anruf, den der Angreifer unterbinden möchte nicht empfängt bzw. tätigen kann.

In diesem Szenario wird es ein bestimmtes UE geben, welches die QoS-Anwendungen Telefonie und MBB (Mobile Broadband) verwendet. Dafür wird ein lokales Setup mit 3 Zellen, wie in Abbildung 5.2 dargestellt, verwendet. Es gibt die Zelle B, welche für Telefonie und Zelle C und D, die für MBB verwendet werden sollen. Im RAN wird es eine Richtlinie geben, die diese Zuteilung der Dienste auf die Zellen vornimmt. Diese Richtlinie soll so manipuliert werden, dass das UE keinen Zugriff mehr auf Telefonie über die Zelle B hat.

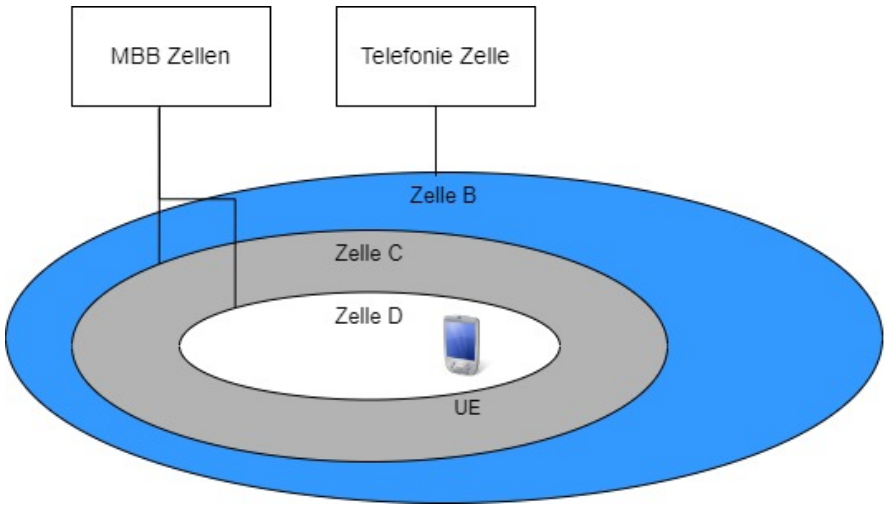

Abbildung 5.2: Verwendung der Zellen angelehnt an [All23d]

Durch diesen Angriff, werden die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit angegriffen. In Bezug auf Abschnitt 4.2.1 werden die Angriffsziele Asset-D-07, Asset-D-18 ,Asset-D-30 und Asset-D-39 identifiziert und angegriffen. Das Zielobjekt soll, wie auch in den kommenden Szenarien, ein Funkzugangsnetz eines Mobilfunkanbieters sein, welches auf Open RAN-Technik basiert.

5.1.2 Informationsbeschaffung

In dieser Phase werden Informationen zum Zielobjekt im Kontext der Angriffsziele gesucht. Durch die Analyse öffentlich zugänglicher Spezifikationen der O-RAN Alliance kann die vollständige Arbeitsweise und Struktur eines Open RAN Systems erkannt werden. Die Netzwerkfunktionen Non-RT RIC, Near-RT RIC sowie die E2 Netzwerkfunktionen CU und DU, sowie die Schnittstellen A1, E2 und R1 gelten als besonders gefährlich für den Angriff auf A1-Richtlinien, da sie am Austausch von Richtlinien für das Traffic Steering beteiligt sind.

Es ist auch sinnvoll, Informationen darüber zu sammeln, wie man Zugriff auf das System erhalten kann. Eine bekannte Schwachstelle im Bereich der Informationsbeschaffung für 5G Mobilfunknetze findet sich im Security Framework „MITRE FiGHT “ [FiG]. Die GSMA, ein weltweiter Verband der Mobilfunkindustrie, der Standards fördert und Mobilfunkbetreiber sowie Unternehmen in der Branche repräsentiert, verwaltet Datenbanken mit standardisierten Informationen für Mobilfunkbetreiber. Der Standard für diese Informationen heißt IR.21. Darunter sind auch Informationen zur Netzwerkstruktur und zu IPs von Netzwerkfunktionen, die einen ersten Zugang zum Netzwerk ermöglichen könnten.

5.1.3 Schwachstellenanalyse

Weiterführende Untersuchungen der Spezifikationen zeigen, dass in „O-RAN A1 interface: Transport Protocol 3.0 “ [All23a] die Unterstützung und Anwendung von TLS 1.2/1.3 für die Transportschicht gemäß den „O-RAN Security Requirements Specifications “ [All23i] vorgesehen ist. Die Sicherheitsmaßnahme „SEC-CTL-A1 “ für die A1-Schnittstelle gibt wiederum lediglich an, dass TLS zum Schutz der Transportschicht nur unterstützt werden muss. Dies deutet auf eine potenzielle Sicherheitslücke hin, da die über die A1-Schnittstelle transportierten Richtlinien möglicherweise im Klartext aus dem Datenverkehr aufgezeichnet werden können.

Ein ähnliches Problem gibt es auf der E2 Schnittstelle. Dort wird auch nur eine Unterstützung von IPsec, als Sicherheitsprotokoll zur Datenübertragung, vorgeschrieben. Entsprechend kann es auch auf der E2 Schnittstelle zu unverschlüsselter Übertragung von Richtlinien kommen, wenn der Betreiber sich gegen eine die Nutzung von IPsec entscheidet.

5.1.4 Angriffsszenario entwickeln

5.1.4.1 Einstiegspunkt finden

Durch Social-Engineering kann ein Angreifer Zugriff auf die Datenbank von IR.21 Informationen bekommen und entsprechend die Netzwerktopologie und IPs von Netzwerkfunktionen verwenden, um das System mit verschiedenen Tools passiv und aktiv zu scannen. Beim Scannen des Netzwerks stellt der Angreifer eine Sicherheitslücke beim Near-RT RIC fest, bei der zu schwache oder falsch konfigurierte Authentifizierungsmechanismen benutzt werden. Dadurch gibt es die Möglichkeit ein xApp-Image mit Schadcode zu erstellen und dieses beim Near-RT RIC zu registrieren [FiG]. Unter der Annahme, dass dieses Vorgehen erfolgreich ist, hätte man das Open RAN Netzwerk kompromittiert und könnte sich von diesem Ausgangspunkt weiter im System bewegen und Angriffe ausführen.

5.1.4.2 weiteres Vorgehen

Basierend auf der Annahme aus Abschnitt 5.1.4.1, kann die bösartige xApp des Angreifers den Zugriff auf den Near-RT RIC und dessen APIs ermöglichen, sowie anschließend den Datenverkehr zwischen dem Non-RT RIC und dem Near-RT RIC über die A1-Schnittstelle aufzeichnen, beispielsweise mit Wireshark. Dabei sollte ein Informationsaustausch über HTTP mit einem JSON-Format, wie in [All23a] definiert, sichtbar sein.

```
1  {
2      "policy_id": "1",
3      "scope":{
4          "ueId":"0000000009295552"
5          "qosId": "1",
6      },
7      "tspResources":[
8         {
9            "cellIdList":[
10              {"plmnId":{"mcc":"138", "mnc":"426"},
11               "cId":{"ncI":"B"}}],
12           "preference":"SHALL"
13        }
14     ]
15  }
16  {
17     "policy_id": "2",
18     "scope":{
19         "ueId":"0000000009295552"
20         "qosId": "9"},
21     "tspResources":[
22        {
23           "cellIdList":[
24               {"plmnId":{"mcc":"138", "mnc":"426"},
25               "cId":{"ncI":"C"}}
26               {"plmnId":{"mcc":"138", "mnc":"426"},
27                "cId":{"ncI":"D"}}],
28           "preference":"SHALL"
29        }
30     ]
31  }
```

Abbildung 5.3: Richtlinie für Zellenzuweisung zu entsprechenden Diensten

Die Richtlinie, welche im JSON-Format übertragen wird sollte wie in Abbildung 5.3 aussehen. In Zeile 1 bis 17 ist die erste Richtlinie zu erkennen, wo auf Zeile 4 und 5 die ueId, als Identifier des Endgerät, und qosId 9 für den Dienst Telefonie angegeben werden. In Zeile 11 ist der Zellen-Identifier (cId) mit B angegeben, also die Zelle die für Telefonie genutzt werden soll. Als Präferenz wird auf Zeile 13 „SHALL “ verwendet um die Zelle B zwingend für Telefonie zu nutzen. In der zweiten Richtlinie ist der Aufbau gleich, nur das die qosId 9, für den Dienst MBB, ist und die Zellen C und D verwendet werden sollen.

Diese Richtlinie soll nun entsprechend dem Angriffsziel, keine Telefonie mehr zuzulassen, manipuliert werden. Die Richtlinie sieht nach der Manipulation wie in Abbildung 5.4 aus. Die Manipulation beinhaltet, dass die Zellen C und D hinzugefügt werden, wie in Zeile 12 bis 15 zu sehen ist, und das Preference Attribut auf „FORBID “ gesetzt wurde, also keine der aufgeführten Zellen für Telefonie verwendet darf.

```
1  {
2      "policy_id": "1",
3      "scope":{
4          "ueId":"0000000009295552"
5          "qosId": "1",
6      },
7      "tspResources":[
8         {
9            "cellIdList":[
10              {"plmnId":{"mcc":"138", "mnc":"426"},
11               "cId":{"ncI":"B"}},
12               {"plmnId":{"mcc":"138", "mnc":"426"},
13               "cId":{"ncI":"C"}},
14               {"plmnId":{"mcc":"138", "mnc":"426"},
15               "cId":{"ncI":"D"}}
16           ],
17           "preference":"FORBID"
18        }
19     ]
20  }
```

Abbildung 5.4: Richtlinie für Zellenzuweisung zu entsprechenden Diensten nach der Manipulation

Da die Richtlinie nicht abgefangen, sondern nur mitgelesen und anschließend manipuliert haben, müssen wir davon ausgehen, dass die eigentliche Richtlinie durch den Workflow transportiert wurde und ausgeführt wird. Deshalb wird die UPDATE-Prozedur anstatt der CREATE-Prozedur durchgeführt. Die manipulierte Richtlinie muss manuell mit einem curl Befehl als HTTP PUT Nachricht an das Near-RT RIC gesendet werden um den Workflow zum Update der aktuellen Richtlinie einzuleiten. Dafür muss der HTTP Nachricht bestimmte Informationen wie policyTypeId, policyId und PolicyObject mitgegeben werden. Der Nachrichtenaustausch via HTTP sollte dann wie in Abbildung 5.5 erfolgen. Der Angreifer sendet also eine HTTP PUT Nachricht mit den entsprechenden Informationen an den Near-RT RIC und bekommt anschließend eine Rückmeldung mit dem HTTP Status Code 200: OK zurück [All23j].

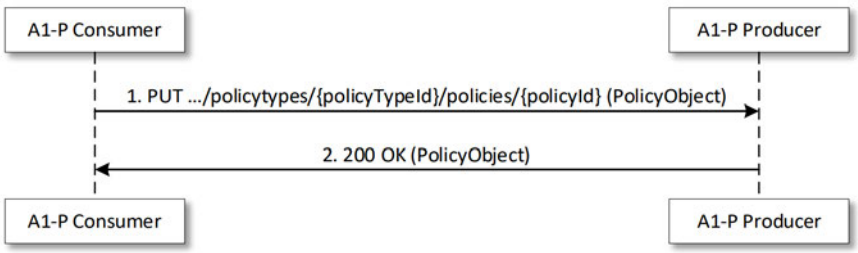

Abbildung 5.5: Policy-Update Prozedur [All23j]

5.1.4.3 Werkzeuge und Tools

1. **Nmap:** Für das Scannen eines Netzwerks ist Nmap eines der bekanntesten Tools. Nmap ist ein reiner Portscanner, also dient dem Scannen beliebig vieler Ports in Verbindung mit dessen Status. Nmap bietet aber auch weitere Funktionen wie z.B. eine Hosterkennung, wo Nmap alle aktiven Hosts in einem Netzwerk finden und anzeigen kann. Dafür gibt es den einfachen Befehl *nmap -sL „IP“*, um alle Hosts im Netzwerk aufzulisten. Effektiver ist allerdings der Befehl *nmap -sP „IP“*, der nur die aktiven Hosts auflistet. Wenn man nur *nmap „IP“* ausführt, bekommt man eine detaillierte Ausgabe zu genau dieser IP, mit Informationen wie z.B. Domainname, IP, Ports mit Status und deren Dienst [Eng21]
2. **Wireshark:** Wireshark ist ein Netzwerkanalyse-Tool, das dem Aufzeichnen und Auswerten von Datenverkehr zwischen verschiedenen Netzwerkgeräten dient. In diesem Szenario könnte Wireshark verwendet werden, um die Kommunikation über die A1 Schnittstelle aufzuzeichnen und die HTTP-Nachrichten aus dem Kommunikationsmitschnitt herauszufiltern. Um nur den Datenverkehr zwischen den beiden gewünschten Netzwerkgeräten (Non-RT RIC und Near-RT RIC) und nur die HTTP Nachrichten zu bekommen, kann man folgenden Filter - *http and ip.addr == 192.168.1.1 and ip.addr == 192.168.1.2* - verwenden, um nicht den kompletten Mitschnitt der Kommunikation durchgehen zu müssen [23j]
3. **Curl:** Curl stellt ein Befehlszeilen-Tool sowie eine Programmbibliothek dar, die zur Übertragung von Daten mittels URLs verwendet wird. Diese vielseitige Anwendung bietet Unterstützung für diverse Protokolle, darunter HTTP, HTTPS, FTP, FTPS, SCP, SFTP, LDAP und viele weitere. Aufgrund seiner breiten Anwendbarkeit wird Curl häufig eingesetzt, um Daten über verschiedene Netzwerkprotokolle zu senden und zu empfangen. Somit kann Curl auch in unserem Angriffsszenario verwendet werden um die manipulierte Richtlinie weiter an das NEar-RT RIC zu senden.

5.1.5 Evaluation und Verteidgungsmechanismen

Dieses Angriffsszenario stellt eine hohe Gefahr für ein Mobilfunknetzwerk dar, weil es einzelne UEs sowie Gruppen von UEs komplett kontrollieren kann im Bezug auf ihre Verwendung von Diensten. Der schwere Teil des Angriffs sollte hierbei das Eindringen in das Netzwerk sein, weil im Bezug auf Authentifizierung und Autorisierung die Sicherheitsrichtlinien der Spezifikationen schon relativ ausgereift sind. Ist dieser Teil des Angriffs allerdings überwunden, durch Spoofen von Anmeldeinformationen oder durch Ausnutzen von Schwachstellen, wie in diesem Szenario, kann der Angreifer leicht agieren um die Richtlinien zu manipulieren. Durch die optionale Verwendung von Verschlüsselung bei HTTP Nachrichten, kann ein Angreifer diese einfach mitlesen, manipulieren und weiterleiten. Natürlich ist beim Weiterleiten ebenfalls zu beachten, dass dort Wahrscheinlich ebenfalls Zertifikate, Anmeldeinformationen oder andere Credentials gefordert sind. Solche können sich Angreifer allerdings auch einfach durch Social-Engineering Angriffe beschaffen.

Um solche Angriffe zu vermeiden, sollten Verschlüsselungsmechanismen wie TLS 1.2/1.3 verwendet werden und nicht nur unterstützt, wie es in den Spezifikation steht. Des weiteren sind Maßnahmen gegen Social-Engineering wie Schulungen zu Sensibilisierung und starke Authentifizierungsmechanismen zu empfehlen. Um Angriffe auf Open-Source-Code zu verhindern, sollten alle Funktionen des Netzwerks, die auf Open-Source-Code basieren, auf Schwachstellen und Sicherheitslücken überprüft werden.

5.2 Angriffsszenario 2: Auslesen von Mobilitätsmustern

5.2.1 Zieldefintion und Identifikation von Zielobjekten

Dieser Angriff umfasst das Mitlesen von Informationen zum Mobilitätsmuster eines Endgerät, welche im Anwendungsfall Traffic Steering im Non-RT RIC mit Hilfe einer rApp als Enrichment Information bereitgestellt werden. Die rApp nutzt ML-Algorithmen, um Mobilitätsmuster zu erlernen und den künftigen Standort von Endgeräten vorherzusagen. Das klar definierte Angriffsziel besteht darin, diese Informationen durch die Entwendung der Standortvorhersagen eines Endgerät zu bekommen. Dabei würden die beiden Sicherheitsziele Vertraulichkeit und Integrität geschädigt werden.

Das Zielobjekt in diesem Angriffsszenario soll wieder ein O-RAN Netzwerk sein. Spezieller soll der Angriff auf der Ebende des Non-RT RIC und einer rApp geschehen um dort die Mobilitätsdaten auszulesen. Ebenso die R1 Schnittstelle zwischen dem Non-RT RIC und einer rApp kann als Angriffsobjekt betrachtet werden, da hier die sensible Informationen übertragen werden.

5.2.2 Informationsbeschaffung

Im Rahmen eines potenziellen Angriffsszenarios auf ein Open RAN Netzwerk, insbesondere im Kontext von Traffic Steering, erfolgt die Informationsbeschaffung durch eine eingehende Analyse der Komponenten des Netzwerks. Hierbei stehen der Non-RT RIC, die rApp und die R1 Schnittstelle im Fokus des Angriffs. Durch die gezielte Untersuchung der Spezifikationen der O-RAN Alliance werden zunächst Einblicke in die allgemeine Architektur des Open RAN Netzes sowie die Abläufe innerhalb des O-RAN Rahmens gewonnen. Die rApp, die mittels ML-Algorithmen Mobilitätsmuster erkennt und zukünftige Standorte von Endgeräten vorhersagt, stellt dabei ein zentrales Angriffsziel dar. Der gezielte Diebstahl von Informationen, insbesondere der prognostizierten Standorte von Endgeräten, wird als primäres Ziel definiert. Hierbei wird der Fokus auf die Schwachstellenanalyse der rApp und deren Trainingsdaten gelegt, um mögliche Sicherheitslücken zu identifizieren. Der Non-RT RIC, als entscheidende Komponente für die Steuerung des RAN, und die R1 Schnittstelle, die für die Kommunikation zwischen dem RIC und der rApp verantwortlich ist, werden ebenfalls als potenzielle Angriffsziele betrachtet. Eine tiefgehende Untersuchung ihrer Spezifikationen ermöglicht die Identifikation von potenziellen Schwachstellen im Kontext von Traffic Steering.

Die Informationsbeschaffung für einen ersten Zugang zum Netzwerk und zur Erlangung unautorisierten Zugriffs auf das Open RAN Netzwerk kann mittels verschiedener Angriffstechniken wie Social-Engineering oder anderer taktischer Mittel erfolgen. Social-Engineering zielt darauf ab, menschliche Schwächen auszunutzen, um Zugriffsinformationen oder sensiblen Daten zu erlangen. Im Kontext eines potenziellen Angriffsszenarios auf das Open RAN Netzwerk könnten Angreifer versuchen, durch gezielte Phishing-Angriffe oder Spear-Phishing-Maßnahmen an sensible Zugangsdaten zu gelangen. Dies könnte beispielsweise durch das Versenden gefälschter E-Mails oder das Erstellen manipulierter Websites geschehen, die scheinbar mit dem Open RAN Netzwerk oder seinen Komponenten verbunden sind. Hierbei könnten gezielt Mitarbeiter oder Nutzer des Netzwerks angesprochen werden, um Zugangsdaten, Passwörter oder andere authentifizierungsrelevante Informationen zu

erschleichen. Entsprechend sammelt Der Angreifer in diesem Angriffsszenario persönliche Informationen über einen Mitarbeiter, insbesondere im Zusammenhang mit einem Hobby. Dies könnte durch Recherche in sozialen Medien, öffentlich verfügbaren Informationen oder möglicherweise durch Insider-Informationen erfolgen.

5.2.3 Schwachstellenanalyse

Eine mögliche Schwachstelle besteht in der Umsetzung der Anforderungen bezüglich Vertraulichkeit, Integrität und Replay-Schutz bei der R1 Schnittstelle. Falls diese Sicherheitsmerkmale unzureichend implementiert sind, könnten Datenschutzverletzungen oder Datenmanipulationen auftreten.Dahingehend sind laut Spezifikationen der O-RAN Allaince Sicherheitsprotokolle wie TLS, mTLS und OAuth 2.0 nur zu unterstützen, aber nicht umzusetzen. Sollten diese Protokolle nicht angewendet werden, bei der Datenübertragung zwischen dem Non-RT RIC und der rApp, kann der Datenverkehr unverschlüsselt mitgelesen werden [All23i].

Auch die Sicherheit des Non-RT RIC und der rApps im Open RAN ist von entscheidender Bedeutung, und es gibt klare Sicherheitsanforderungen gemäß den Spezifikationen. Diese sagen allerdings nur aus das der Non-Rt RIC und rApps die Sicherheitsprotokolle TLS und OAuth 2.0 nur unterstützen müssen. Darin besteht die potenzielle Schwachstelle , dass es laut den Spezifikationen auch zu unverschlüsselter Kommunikation sowohl vom Non-RT RIC als auch von den rApps kommen kann, da die Protokoll nicht zwingend verwendet werden müssen.

Eine kritische Schwachstelle im O-RAN-System betrifft die Integration von rApps durch Drittanbieter. Hier besteht das inhärente Risiko, dass solche Anbieter selbst Angreifer sind oder von diesen kompromittiert wurden. Zusätzlich könnte ein gezielter Angriff auf den rApp-Betreiber dazu führen, dass ein Angreifer Zugang zur Anwendung erhält und somit das gesamte O-RAN-System manipulieren oder abhören kann.

5.2.4 Angriffsszenario entwickeln

5.2.4.1 Einstiegspunkt finden

Zum Anfang des Angriffsszenarios wählt der Angreifer den Betreiber einer rApp als ersten Angriffspunkt. Diese rApp verwendet fortschrittliche ML-Algorithmen zur präzisen Vorhersage von Mobilitätsmustern von Endgeräten im Open RAN Netzwerk. Durch eine gezielte Identifizierung eines Mitarbeiters des Betreibers gelangt der Angreifer an persönliche Informationen, wobei ein besonderes Augenmerk auf einem spezifischen Hobby des Mitarbeiters liegt.

Mit diesen gewonnenen Erkenntnissen schmiedet der Angreifer eine zielgerichtete Phishing-E-Mail, die sich auf das identifizierte Hobby des Mitarbeiters bezieht. Die E-Mail ist so konzipiert, dass sie vertrauenswürdig erscheint und den Mitarbeiter dazu verleitet, auf Links zu klicken oder Anhänge zu öffnen. Durch diese geschickte Täuschung versucht der Angreifer, die Anmeldeinformationen des Mitarbeiters zu erschleichen.

Nach erfolgreicher Phishing-Attacke hat der Angreifer einen entscheidenden Zugangspunkt erlangt und kann nun unbefugten Zugriff auf die rApp im Open RAN Netzwerk erhalten. Dieser Zugang ermöglicht dem Angreifer potenziell weitreichende Manipulationen im Open RAN System.

5.2.4.2 Weiteres Vorgehen

Im vorliegenden Szenario verfolgt der Angreifer das Ziel, die zwischen dem Non-RT RIC und der rApp ausgetauschten Informationen unbemerkt mitzulesen. Zu diesem Zweck entscheidet sich der Angreifer für einen Man-in-the-Middle-Angriff, indem er das Image der rApp klonen und selbst in das O-RAN Netzwerk einspeisen möchte. Diese Vorgehensweise setzt voraus, dass der Angreifer durch den Zugriff auf die originale rApp auch die entsprechenden Zertifikate und Zugangsdaten erlangt hat, die erforderlich sind, um die Autorisierungs- und Authentifizierungsmechanismen zu umgehen.

Durch das Klonen des rApp-Images und die Integration in das Netzwerk kann der Angreifer eine Position zwischen dem Non-RT RIC und der tatsächlichen rApp einnehmen, um den gesamten Datenverkehr abzufangen. Der Einsatz von übernommenen Zertifikaten und Credentials ermöglicht dem Angreifer, sich als legitimer Kommunikationspartner auszugeben und so unbemerkt im Datenverkehr zwischen den beiden Endpunkten zu agieren.

Im Rahmen des Traffic Steering Workflows mit Enrichment Information, wie es in den Spezifikationen der O-RAN Alliance beschrieben ist, erfolgt durch die „UE Location “ rApp die Sammlung von bisherigen Standortinformationen zu einem bestimmten Endgerät (UE). Unter Einsatz eines trainierten ML-Algorithmus wird daraufhin die Standortvorhersage betrieben.

Abbildung 5.6 illustriert einen Abschnitt des Workflows des Traffic Steering mit Unterstützung durch Enrichment Information. Der Ablauf beginnt mit dem üblichen Austausch, der die Erkennung von Verstößen gegen durchzusetzende Richtlinien einschließt. Darauf folgt ein umfassender Datenaustausch zwischen sämtlichen Komponenten des Systems. Der entscheidende Abschnitt, auf den ein potenzieller Angriff abzielt und bei dem Daten aufgezeichnet werden sollen, manifestiert sich in Schritt 18 und Schritt 19.

In Schritt 18 nimmt die „UE Location“ rApp Standortvorhersagen für jedes individuelle Endgerät (UE) vor. Diese Vorhersagen werden anschließend in Schritt 19 an die "Traffic Steering"rApp weitergeleitet. Der kritische Punkt für den potenziellen Angriff liegt in diesem Übertragungsschritt (Schritt 19). Hier beabsichtigt der Angreifer, fortlaufend in die Weiterleitung einzugreifen, um die Standortinformationen eines spezifischen Endgeräts verdeckt nachzuverfolgen. Das kontinuierliche Abfangen dieser Weiterleitung eröffnet dem Angreifer die Möglichkeit, unbemerkt und beharrlich die Bewegungen eines bestimmten UEs zu verfolgen.

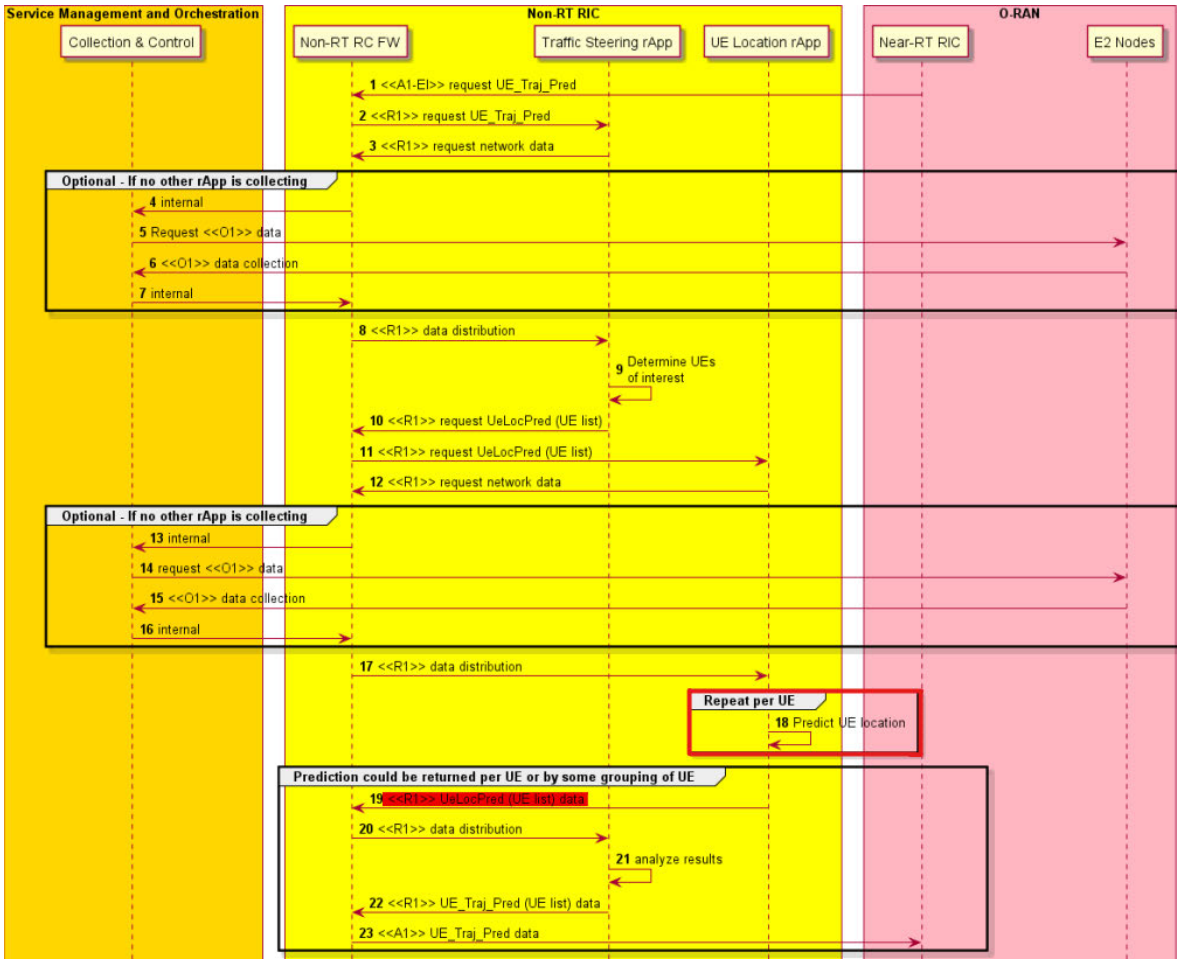

Abbildung 5.6: Workflow von Traffic Steering mit Enrichment Information [All23c]

5.2.4.3 Werkzeuge und Tools

1. **Soziale Medien und OSINT:**
   Maltego: Ein leistungsstolles OSINT-Tool, das Informationen aus verschiedenen Quellen sammelt und visuell darstellt. Es kann verwendet werden, um Beziehungen zwischen Personen, Organisationen und anderen Entitäten herzustellen [23d].
   Recon-ng: Ein Open-Source-Framework für die Informationsgewinnung. Es ermöglicht die Sammlung von Informationen aus verschiedenen öffentlichen Quellen und unterstützt die Analyse von Daten [23g].
2. **Phishing-Tools:**
   GoPhish: Ein Phishing-Framework, das die Erstellung, Ausführung und Analyse von Phishing-Kampagnen vereinfacht. Es ermöglicht die Erstellung von personalisierten Phishing-E-Mails und gefälschten Websites [23c].
   Social-Engineer Toolkit (SET): Ein Toolkit, das auf Social Engineering-Angriffe spezialisiert ist. Es bietet Tools für Phishing, Spear-Phishing und andere Social-Engineering-Techniken [23i].
3. **Analysewerkzeuge für Netzwerkkomponenten:**
   Auch hier können wie im Angriffsszenario zuvor Wireshark und Nmap genutzt werden.

4. **Sicherheitsbewertungstools:**
   Nessus: Ein weit verbreitetes Sicherheitsbewertungstool, das Schwachstellen in Netzwerken, Systemen und Anwendungen identifiziert. Es führt automatisierte Scans durch und bietet detaillierte Berichte über gefundene Schwachstellen [23e].
   OpenVAS (Open Vulnerability Assessment System): Ein Open-Source-Vulnerability-Scanner, der Schwachstellen in Netzwerken aufspürt und bewertet. Es bietet eine Datenbank mit bekannten Schwachstellen und ermöglicht die Durchführung von Sicherheitsaudits [23f].
5. **Man-in-the-Middle-Angriffstools:**
   Ettercap: Ein leistungsfähiges Tool für den Man-in-the-Middle-Angriff. Es kann den Datenverkehr zwischen Zielgeräten abfangen, analysieren und manipulieren. Ettercap unterstützt verschiedene Angriffstechniken [23b].
   Bettercap: Ein umfassendes Framework für Netzwerkanalyse und Penetration Testing. Es kann verwendet werden, um den Datenverkehr zu belauschen und zu manipulieren [23a].

5.2.5 Evaluation und Verteidgungsmechanismen

Die vorgestellten Angriffsszenarien zielen darauf ab, die Mobilitätsmuster von Endgeräten im Open RAN Netzwerk zu kompromittieren, was erhebliche Auswirkungen auf die Vertraulichkeit und Integrität der übertragenen Daten hat. Die Identifikation von Schwachstellen in der rApp, dem Non-RT RIC und der R1 Schnittstelle legt den Fokus auf potenzielle Angriffspunkte im System.

Die unzureichende Umsetzung von Sicherheitsmerkmalen wie Vertraulichkeit, Integrität und Replay-Schutz in der R1 Schnittstelle eröffnet Möglichkeiten für Datenschutzverletzungen und Datenmanipulation. Durch das Fehlen der vorgeschlagenen Sicherheitsprotokolle (TLS, mTLS, OAuth 2.0) könnte ein Angreifer den Datenverkehr unverschlüsselt mitlesen. Entsprechend zeigt Analyse der Spezifikationen, dass die rApp und der Non-RT RIC lediglich die Unterstützung, nicht jedoch die Verpflichtung zur Implementierung der Sicherheitsprotokolle TLS und OAuth 2.0 haben. Dies könnte zu unverschlüsselter Kommunikation führen, wenn die Protokolle nicht zwingend angewendet werden.

Die Integration von rApps durch Drittanbieter birgt das inhärente Risiko, dass diese Anbieter selbst Angreifer sind oder kompromittiert wurden. Ein gezielter Angriff auf den rApp-Betreiber könnte Zugang zur Anwendung ermöglichen und somit das gesamte O-RAN-System manipulieren oder abhören.

Die Implementierung von robusten Sicherheitsprotokollen wie TLS, mTLS und OAuth 2.0 in der R1 Schnittstelle ist entscheidend. Dies stellt sicher, dass der Datenverkehr zwischen dem Non-RT RIC und der rApp verschlüsselt und authentifiziert ist, wodurch das Risiko von unautorisiertem Zugriff minimiert wird. Neben der R1 Schnittstelle muss diese Regelung auch für den Non-RT RIC und die rApps gelten. Des weiteren sollte vor der Integration von rApps durch Drittanbieter sollte eine umfassende Risikobewertung durchgeführt werden. Dabei ist sicherzustellen, dass die Drittanbieter den erforderlichen Sicherheitsstandards entsprechen. Zusätzliche Sicherheitsmaßnahmen können die Überwachung von Drittanbieteraktivitäten und regelmäßige Sicherheitsaudits umfassen

Die vorgestellten Abwehrmechanismen sind entscheidend, um die Sicherheitsrisiken zu minimieren und das Open RAN Netzwerk vor potenziellen Angriffen zu schützen. Durch eine Kombination aus technologischen Sicherheitsverbesserungen, klaren Sicherheitsrichtlinien und sorgfältiger Auswahl

von Drittanbietern kann die Integrität und Vertraulichkeit des Netzwerks gewährleistet werden. Es ist von entscheidender Bedeutung, diese Maßnahmen kontinuierlich zu überwachen und anzupassen, um auf sich entwickelnde Bedrohungen angemessen reagieren zu können.

5.3 Angriffsszenario 3: Kollision von Richtlinien

5.3.1 Zieldefintion und Identifikation von Zielobjekten

In diesem Angriffsszenario soll durch das Einschleusen einer zweiten xApp auf dem Near-RT RIC eine Richtlinie erstellt werden, die zu einem Konflikt mit einer anderen Richtlinie führt und folglich zur Minimierung der Netzwerkperformance führen soll. Der Angriff findet unter der Annahme statt, dass der Angreifer durch das Aufzeichnen des Datenverkehrs zwischen dem Non-RT- und Near-RT RIC weiß, welche Richtlinie zur Zeit des Angriffs ausgeführt wird. Diese Richtlinie lässt alle Endgeräte mit der Quality-of-Service-Anwendung MBB auf eine spezielle Zelle einwählen. Der Angreifer steuert das RAN nun durch seine eingeschleuste xApp genau entgegen dieser Richtlinie. Er lässt also alle Endgeräte mit der Quality-of-Service-Anwendung MBB sich auf eine andere Zelle einwählen. Dadurch wechseln diese Endgeräte ständig zwischen beiden Zellen hin und her, und es entsteht eine Art Ping-Pong-Effekt.

Als Zielobjekt ist hier der Near-RT RIC anzusehen, da dieser kompromittiert werden soll und eine schädliche xApp in das System eingeschleust wird. Des Weiteren sind die Endgeräte als Zielobjekt zu betrachten, da diese letztendlich in ihrer Verfügbarkeit beeinträchtigt werden.

5.3.2 Informationsbeschaffung

Die strategische Erhebung von Informationen für einen potenziellen Angriff, der Konflikte zwischen xApps im Open RAN hervorruft, setzt eine umfassende Analyse der beteiligten Netzwerkkomponenten voraus. Hierbei liegt der Fokus auf den xApps selbst sowie den Schnittstellen, die ihre Interaktion ermöglichen. Durch eine sorgfältige Durchleuchtung der Spezifikationen der O-RAN Alliance erhalten wir Einblicke in die komplexe Architektur des Open RAN Netzwerks.

Die Schlüsselziele dieses Szenarios sind die xApps, die unterschiedliche Funktionen im Open RAN übernehmen. Unsere Analyse zielt darauf ab, potenzielle Schwachstellen innerhalb dieser Anwendungen zu identifizieren, die zu Konflikten zwischen ihnen führen könnten. Aspekte wie Datenverarbeitung, Kommunikation und Ressourcenzuweisung stehen dabei im Mittelpunkt unserer Untersuchung.

Um gezielt Konflikte zwischen den xApps herbeizuführen, konzentrieren wir uns auf mögliche Manipulationen der Kommunikationsprotokolle und Datenströme. Dies könnte durch präzise Eingriffe in die Schnittstellen zwischen den xApps geschehen. Die Erkennung von Schwachstellen in den xApps, insbesondere in Bezug auf unzureichende Validierung von Eingaben oder unsichere Datenübertragungen, ist hierbei von entscheidender Bedeutung.

Um einen erstmaligen Zugang zum Netzwerk zu erlangen, könnte weiterhin auf bewährte Angriffstechniken wie Social-Engineering zurückgegriffen werden. Durch geschickte Manipulation von Mitarbeitern oder Nutzern des Open RAN Netzes versuchen wir, Zugriffsinformationen zu erlangen. Hierbei könnten speziell angepasste Spear-Phishing-Angriffe zum Einsatz kommen, bei denen gefälschte Kommunikation oder manipulierte Informationen genutzt werden, um Vertrauen zu gewinnen.

Die bewusste Induktion von Konflikten zwischen den xApps erfordert jedoch ein tiefgreifendes technisches Verständnis, um die spezifischen Interaktionen und Abhängigkeiten zwischen den Anwendungen zu verstehen. Dies könnte die Ausnutzung von Schwachstellen in den Implementierungen der xApps oder sogar die zielgerichtete Einschleusung von schadhaftem Code umfassen, um Konflikte zu provozieren. Auch eine gefälschte zweite xApp könnte in das System eingeschleust werden, um eigene Richtlinien in das Netzwerk einfließen zu lassen und somit Konflikte zu verursachen.

Insgesamt erfordert ein derartiges Angriffsszenario ein präzises Wissen über die Architektur des Open RAN sowie die Funktionsweisen der involvierten xApps und Schnittstellen. Ebenso erfordert ein Social-Engineering-Angriff die Erhebung persönlicher Informationen von Mitarbeitern des betroffenen Unternehmens.

5.3.3 Schwachstellenanalyse

Für das Hervorrufen von Konflikten existieren mehrere Möglichkeiten, die in den Sicherheitsanforderungen und -maßnahmen nicht behandelt werden [All23i]. Es gibt zwei übergeordnete Szenarien, um derartige Konflikte zu initiieren. Man kann Konflikte zwischen mehreren xApps herbeiführen oder zwischen einer xApp und einem gNodeB. Im ersten Fall wird zwischen direkten, indirekten und impliziten Konflikten unterschieden, die in Abschnitt 2.2.3 erläutert wurden. Dabei gestaltet sich die Lösung von Konflikten besonders für indirekte und implizite Fälle als anspruchsvoll. Bei der zweiten Variante treten Konflikte zwischen xApps und einem gNodeB mit seinen Kontrollelementen CU und DU auf. Dabei kann ein gNodeB eigene, vom Betreiber gewünschte Ziele umsetzen, während die xApp entgegen diesen Zielen Richtlinien umsetzen möchte, was zu einem Konflikt und entsprechend einer Minderung der Leistung des RAN führt [Eri20].

5.3.4 Angriffsszenario entwickeln

5.3.4.1 Einstiegspunkt finden

Der Angreifer hat das Ziel, eine zweite xApp in das System einzuschleusen, um damit Richtlinien zu erstellen, die Konflikte mit den gNodeB oder anderen xApps hervorrufen. Der einfachste Weg besteht darin, eine bereits existierende xApp zu kompromittieren, ihr Image zu klonen und dieses Image als neue xApp zu instanziieren. Der Angreifer nutzt hierfür Social Engineering im Unternehmen, das die xApp entwickelt und für einen Mobilfunkbetreiber bereitstellt. Gefälschte E-Mails werden im gesamten Unternehmen versendet, wobei ein Mitarbeiter auf einen schädlichen Link in der E-Mail klickt. Dadurch erhält der Angreifer Zugriff auf den Account des Mitarbeiters im unternehmensinternen Intranet und hat Einsicht in Informationen sämtlicher xApps, die vom Unternehmen bereitgestellt und entwickelt werden.

Nachdem der Angreifer Zugriff auf den Account des Mitarbeiters im unternehmensinternen Intranet erlangt hat und somit Informationen sämtlicher xApps sichtbar sind, geht er zur nächsten Phase des Angriffs über. Der Angreifer nutzt die gewonnenen Erkenntnisse, um gezielt eine bereits existierende xApp zu kompromittieren. Durch den Klonprozess des Images dieser xApp erstellt der Angreifer eine exakte Kopie, die jedoch mit schädlichem Code infiziert ist.

Mit dieser manipulierten xApp startet der Angreifer den Integrationsprozess, bei dem er das gefälschte Image als neue xApp im Open RAN Netzwerk einführt. Durch die Tarnung als legitime Anwendung innerhalb des Netzwerks kann die schädliche xApp nun aktiv werden, Richtlinien erstellen und bewusst Konflikte mit den gNodeB und anderen xApps provozieren.

Da die Sicherheitsprotokolle und Maßnahmen des Open RAN Netzwerks nicht ausreichend vor solchen Angriffen schützen, gelingt es dem Angreifer, seine manipulierte xApp erfolgreich zu integrieren. Dies ermöglicht ihm, unbemerkt Schaden anzurichten, da die xApp nun ihre schädlichen Richtlinien implementiert und Konflikte im Netzwerk erzeugt.

5.3.4.2 weiteres Vorgehen

Nachdem die manipulierte xApp erfolgreich im Open RAN Netzwerk platziert wurde, setzt der Angreifer seinen Vorstoß fort, um gezielt Unruhe in der Netzwerkverfügbarkeit zu stiften. Sein primäres Ziel besteht darin, bewusst Konflikte zwischen den xApps oder zwischen einer xApp und den gNodeBs zu provozieren, um eine Beeinträchtigung der Gesamtfunktionalität zu bewirken.

Die schädliche xApp beginnt, Richtlinien und Anweisungen zu generieren, die unmittelbar im Widerspruch zu den üblichen Betriebsrichtlinien stehen. Dies könnte beispielsweise die unangemessene Zuweisung von Ressourcen, die gestörte Frequenznutzung oder die unsachgemäße Handhabung von Datenströmen umfassen. Die manipulierte xApp setzt aktiv darauf, die zugrunde liegenden Prozesse im Netzwerk zu destabilisieren, indem sie absichtlich falsche oder widersprüchliche Anweisungen an die gNodeBs sendet und somit bewusst Konflikte auslöst.

Darüber hinaus werden koordinierte Angriffe zwischen verschiedenen xApps orchestriert, um die Komplexität der Konflikte zu steigern und die Störungen im Netzwerk zu intensivieren. Diese gezielten Angriffe können auf unterschiedlichen Ebenen erfolgen, angefangen bei der Ressourcenzuweisung über die Signalverarbeitung bis hin zur Steuerung von Handovers.

Der Angreifer zielt darauf ab, eine sich ausbreitende Kettenreaktion von Konflikten im gesamten Open RAN Netzwerk auszulösen. Dies hat das Potenzial, nicht nur die Leistung einzelner xApps zu beeinträchtigen, sondern auch die Kommunikation zwischen den gNodeBs zu stören und somit die Gesamtverfügbarkeit des Netzwerks zu bedrohen.

5.3.4.3 Werkzeuge und Tools

Da dieser Angriff in den Techniken, dem Angriffsszenario 2 ähnelt, werden entsprechend auch ähnliche Tools und Werkzeuge verwendet. Entsprechend kann der Angreifer für die Phase des Social-Engineering Framewokrs wie Maltego oder Recon-ng verwenden um Informationen zu Mitarbeitern zu finden und übersichtlich Darzustellen. Für die Erstellung der gefälschte E-Mail können

Tools wie GoPhish verwendet werden. Wenn der Angreifer im System ist kann er zum mitlesen vom Datenverkehr Wireshark verwenden und somit auch bestehende Richtlinien abfangen. Da der Angreifer eine zweite xApp in das System einpflegen möchte, brauch er ein Tool um das Image der bestehenden xApp zu klonen. So etwas kann mit der Containerisierungsplattform Docker erreicht werden.

5.3.5 Evaluation und Verteidgungsmechanismen

Im Hinblick auf das beschriebene Angriffsszenario auf das Open RAN Netzwerk ist eine gründliche Bewertung der Sicherheitslage sowie die Implementierung effektiver Verteidigungsmechanismen von entscheidender Bedeutung. Um solche Angriffe zu verhindern und deren Auswirkungen zu minimieren, sollten verschiedene Aspekte der Netzwerksicherheit berücksichtigt werden.

Eine zentrale Maßnahme besteht in der Implementierung fortschrittlicher Netzwerküberwachungstools, die den Datenverkehr zwischen xApps und gNodeBs überwachen. Durch Anomalieerkennungssysteme können verdächtige Aktivitäten identifiziert werden, was auf mögliche Sicherheitsverletzungen hinweisen kann. Speziell für diesen Fall, gibt es schon Ausarbeitungen von ersten Konfliktminderungs Frameworks, welche direkte, indirekte und implizite Konflikte lösen können [AK23].

Zusätzlich dazu ist eine Stärkung der Zugriffskontrolle und Authentifizierung entscheidend, um unbefugten Zugriff auf interne Systeme zu verhindern. Die Anwendung von Multi-Faktor-Authentifizierung und regelmäßige Überprüfungen der Zugriffsrechte tragen dazu bei, potenzielle Schwachstellen zu minimieren.

Ein weiterer wichtiger Aspekt ist das Sicherheitsbewusstseinstraining für Mitarbeiter, insbesondere im Hinblick auf Social Engineering-Angriffe. Schulungen können dazu beitragen, das Personal für mögliche Risiken zu sensibilisieren und die Erfolgschancen von Phishing- und Spear-Phishing-Angriffen zu reduzieren.

Im Kontext von Containerisierungstechnologien wie Docker ist die Sicherheit von Docker-Images von Bedeutung. Die Überprüfung und Absicherung von Docker-Images sowie die Implementierung von Richtlinien zur Image-Integrität tragen dazu bei, die Manipulation von xApp-Images zu verhindern.

Ein regelmäßiges Schwachstellenmanagement, das eine kontinuierliche Überprüfung und Aktualisierung von Systemen und Software beinhaltet, ist ebenfalls entscheidend, um Schwachstellen zu identifizieren und zu beheben.

Des Weiteren sollte eine Verschlüsselung der Kommunikation zwischen xApps und gNodeBs implementiert werden, um die Sicherheit zu gewährleisten. Die Verwendung von Sicherheitsprotokollen wie TLS und die Sicherstellung der Integrität der Datenübertragung können das Risiko von Netzwerkmanipulationen minimieren.

Die kontinuierliche Bewertung von Sicherheitsmaßnahmen und die Anpassung an sich entwickelnde Bedrohungen sind unerlässlich, um die Integrität und Verfügbarkeit des Open RAN Netzwerks zu gewährleisten. Durch proaktive Verteidigungsstrategien und eine ganzheitliche Sicherheitsstruktur kann die Resilienz des Netzwerks gegenüber komplexen Angriffen gestärkt werden.

6 Durchführung eines Angriffsszenarios auf einer Open RAN Implementation

6.1 Testumgebung

Für die praktische Ausführung eines Angriffsszenarios wurde als Testumgebung eine Open RAN-Implementierung der Open Network Foundation genutzt, die sich SD-RAN nennt. Sie entspricht den Spezifikationen von 3GPP und der O-RAN Alliance. SD-RAN bietet die Implementierung eines Near-RT-RIC zur Kontrolle des virtualisierten RAN. Außerdem werden eine Reihe von xApps bereitgestellt, um die Elemente im RAN zu steuern. Abbildung 6.1 veranschaulicht die SD-RAN-Architektur. SD-RAN hat zentrale Komponenten wie die DU, RU und CU's. Das SMO beinhaltet das integrierte Non-RT-RIC und das Near-RT-RIC, auf dem entsprechende xApps laufen sollen. Auch die Schnittstellen O1, A1 und E2 sind hier vertreten.

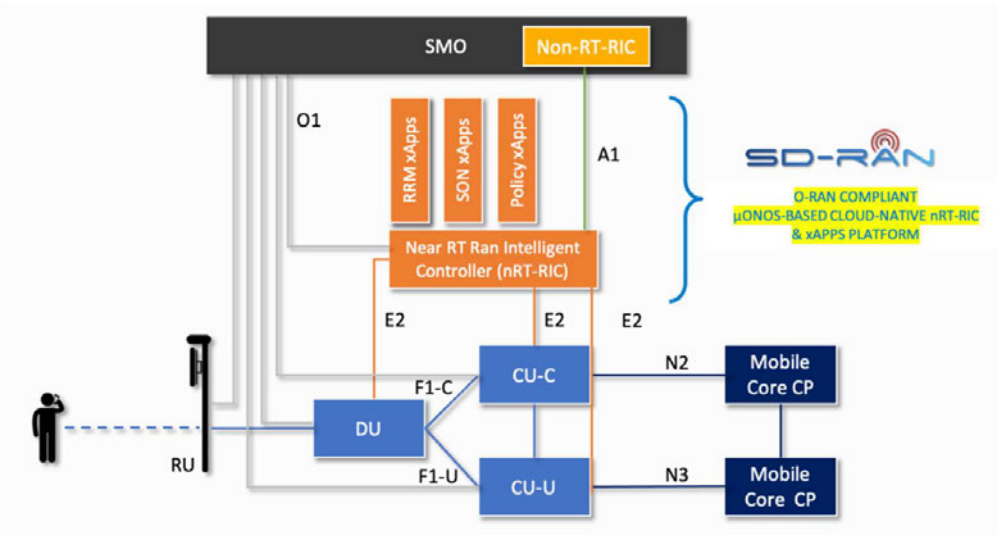

Abbildung 6.1: SD-RAN Architektur [23h]

SD-RAN bietet ein Cluster namens SDRAN-in-a-Box (RiaB), das eine Infrastruktur mit einem Evolved Packet Core (EPC), einem emulierten RAN (CU, DU, UE) und einem RAN Intelligent Controller mit verschiedenen Diensten bereitstellt. RiaB stellt verschiedene Varianten zur Verfügung, um ein RAN zu emulieren. In unserer Testumgebung wird die Variante eines RAN-Simulators verwendet, bei dem mehrere E2-Netzwerkfunktionen (CU-CP/DU/RU) und Endgeräte (UEs) simuliert werden, Steuerungsnachrichten für SD-RAN generiert werden, jedoch kein Datenverkehr zwischen UEs unterstützt wird.

6.2 Angriffsszenario

6.2.1 Annahme

Ein Angreifer schaffte es durch Social-Engineering-Methoden, einen Termin mit einem Mitarbeiter eines Mobilfunknetzbetreibers zu arrangieren. Im Zuge dessen platzierte er einen USB-Stick am Laptop des Mitarbeiters. Die darauf befindliche Schadsoftware wurde automatisch aktiviert und ermöglichte das Auslesen von Informationen, darunter Anmeldeinformationen, Zertifikate und Schlüssel für das virtualisierte Cluster des RAN. Der Angreifer entfernte sich mit dem USB-Stick und den extrahierten Daten, wodurch er Zugriffsinformationen für das RAN-Cluster erlangte. Gestützt auf diese Daten gelang es dem Angreifer, eine bösartige xApp in das Open RAN einzuführen, wodurch er den Host des virtuellen Clusters kompromittieren konnte und nun Zugriff auf das RAN hat.

6.2.2 Durchführung des Angriffs

1. **Skizzieren der Netzwerktopologie:** In einem initialen Schritt wird ein Befehl zur Untersuchung der Netzwerktopologie des Kubernetes-Clusters ausgeführt. Die Netzwerktopologie eines Kubernetes-Clusters umfasst die Struktur und Verbindungen zwischen den verschiedenen Ressourcen innerhalb des Clusters. Der Befehl „kubectl get pods -A “ dient dazu, einen umfassenden Überblick über die architektonische Anordnung der Kubernetes-Ressourcen zu erhalten, einschließlich der Pods, Services, Nodes und deren Netzwerkverbindungen, wie man auch in Abbildung 6.2 sehen kann. Dort ist zu erkennen, dass es vier Namespaces gibt, in denen Pods mit Containern laufen, welche bestimmte feingranulare Aufgaben erfüllen.

```
sdran@sdran-VirtualBox:~/sdran-in-a-box$ sudo kubectl get pods -A
NAMESPACE           NAME                                                READY
calico-system       calico-kube-controllers-7f7959b5db-jmbqw            1/1
calico-system       calico-node-t67xz                                   1/1
calico-system       calico-typha-658685dd8-488jj                        1/1
kube-system         atomix-runtime-consensus-controller-6796b7556d-l694h  1/1
kube-system         atomix-runtime-controller-66cf5558c6-9z8km          0/1
kube-system         atomix-runtime-pod-memory-controller-858ff44659-2cfp7  0/1
kube-system         atomix-runtime-raft-controller-6699445c6d-wzw2f     0/1
kube-system         atomix-runtime-runtime-controller-8d6c956bc-x6qsg   0/1
kube-system         atomix-runtime-shared-memory-controller-5479b9564f-9ws8r  0/1
kube-system         atomix-runtime-sidecar-controller-7fb56cd648-wh6vt  0/1
kube-system         cloud-controller-manager-sdran-virtualbox           1/1
kube-system         etcd-sdran-virtualbox                               1/1
kube-system         helm-install-rke2-calico-crd-t5f6j                  0/1
kube-system         helm-install-rke2-calico-vvjnl                      0/1
kube-system         helm-install-rke2-coredns-n6j26                     0/1
kube-system         helm-install-rke2-ingress-nginx-qfgv2               0/1
kube-system         helm-install-rke2-metrics-server-tkwhn              0/1
kube-system         helm-install-rke2-multus-k6f6v                      0/1
kube-system         kube-apiserver-sdran-virtualbox                     1/1
kube-system         kube-controller-manager-sdran-virtualbox            1/1
kube-system         kube-proxy-sdran-virtualbox                         1/1
kube-system         kube-scheduler-sdran-virtualbox                     1/1
kube-system         onos-operator-app-6f5c4c8656-w9r6d                  0/1
kube-system         onos-operator-topo-55cfbdd947-6hlxx                 1/1
kube-system         rke2-coredns-rke2-coredns-775c5b4bb4-rntkw          1/1
kube-system         rke2-coredns-rke2-coredns-autoscaler-695fc554c9-pgbrw  1/1
kube-system         rke2-ingress-nginx-controller-f2c9x                 1/1
kube-system         rke2-metrics-server-644f588b5-285j5                 1/1
kube-system         rke2-multus-ds-fkp8k                                1/1
local-path-storage  local-path-provisioner-67f5f9cb7b-cfpfj             1/1
riab                onos-a1t-68c59fb46-hgwrw                            1/2
riab                onos-cli-8757f585f-rz2qg                            1/1
riab                onos-config-9574dcb95-729th                         2/3
riab                onos-e2t-6f7cb5cf99-sw7f4                           1/2
riab                onos-kpimon-b7df68b85-p28d9                         2/2
riab                onos-topo-7747cf6b4b-x4gmw                          1/2
riab                onos-uenib-dc584df69-x7mcv                          1/2
riab                ran-simulator-7c7c555978-mppbr                      1/1
riab                sd-ran-consensus-0                                  1/1
riab                sd-ran-consensus-1                                  1/1
riab                sd-ran-consensus-2                                  1/1
riab                sd-ran-rimedo-ts-b7b8c7fd4-8979l                    2/2
```

Abbildung 6.2: Kubernetes-Cluster Topologie

2. **Ziel identifizieren:** Wie im theoretische Angriffsszenario schon beschrieben, soll das Angriffsziel die A1 Schnittstelle sein, welche durch den Pod onos-a1t-68c59fb46-hgwrw im Cluster abgebildet wird. Über diesen Pod müssten die Richtlinien vom Non-RT RIC zum Near-RT RIC und anschließend zur xApp weitergeleitet werden.
   Um weitere Informationen zum Zielpod zu bekommen, bietet die SD-RAN Plattform ein Skript zum Ausgeben von Informationen zu verschiedenen Netzwerkknoten. Durch das Ausführen des Skripts bekommen wir Informationen zu ID, Quelle und IP/Port über die man mit dem Pod kommunizieren kann. Diese Ausgabe ist in Abbildung 6.3 zu sehen.

```
ID: a1:onos-a1t-68c59fb46-s4txm
Kind ID: a1t
Labels: <None>
Source of: uuid:48faa89a-8f2f-5d01-f857-a491cf4cb36b
Target of:
Aspects:
- onos.topo.A1TInfo={"interfaces":[{"type":"INTERFACE_A1AP","ip":"192.168.84.214","port":9639}]}
```

Abbildung 6.3: Informationen zum Pod der A1 Schnittstelle

3. **Datenverkehr aufzeichnen:** Richtlinien, welche vom Non-RT RIC zum Near-RT RIC geschickt werden, können mitgeschnitten werden, wie im Wireshark Mitschnitt in Abbildung 6.4 blau markiert ist. Das Datenpaket wird nur mit HTTP versendet und ist somit nicht verschlüsselt. Entsprechend kann man in Abbildung 6.5 sehen, dass die Richtlinie im Byte-Stream erkennbar ist und ausgelesen werden kann. Dort sind verschiedene Informationen enthalten wie z.B. Zell-ID, UE-ID und die Präferenzen der Richtlinie, welche in Abschnitt 3.3 auch schon erläutert wurden.

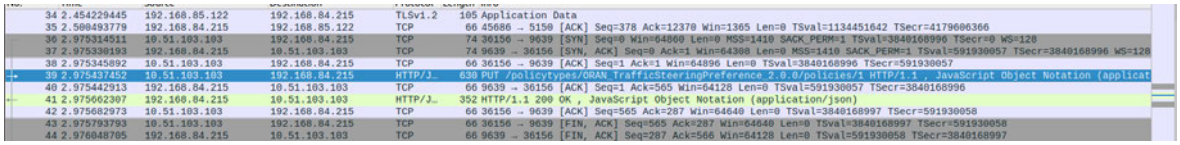

Abbildung 6.4: Wireshark Mitschnitt - HTTP PUT zum weiterleiten der Richtlinie

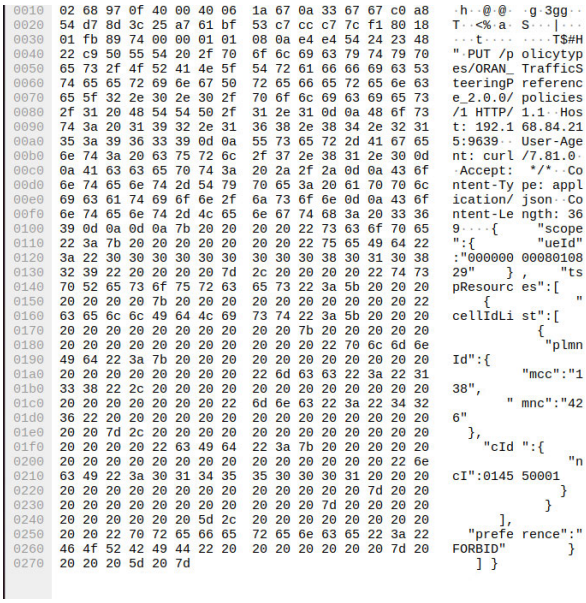

Abbildung 6.5: Unverschlüsselte Übertragung von Richtlinien

4. **Manipulieren und Weiterleiten einer eigenen Richtlinie:** Durch das Auslesen der Richtlinie in Abbildung 6.5 kann diese mit einem beliebigen Editor neu erstellt und mit den Präferenzen des Angreifers gefüllt werden. Anschließend kann die Richtlinie manuell über einen Curl-Befehl an die xApp weitergeleitet werden, was in Abbildung 6.6 abgebildet ist.

Abbildung 6.6: Weiterleiten der manipulierten Richtlinie

6.3 Interpretation und Gegenmaßnahmen

Das beschriebene Angriffsszenario verdeutlicht die Auswirkungen eines erfolgreichen Social-Engineering-Angriffs, bei dem ein Angreifer Zugriffsinformationen für ein virtualisiertes Cluster des RAN erlangt. Durch geschickte Platzierung von Schadsoftware auf einem Mitarbeiter-Laptop und dem gezielten Ausnutzen von Schwachstellen in der Sicherheitspraxis des Unternehmens konnte der Angreifer kritische Daten extrahieren und letztendlich eine bösartige xApp in das Open RAN einführen. Diese bösartige xApp ermöglicht es dem Angreifer, den Host des virtuellen Clusters zu kompromittieren und vollständigen Zugriff auf das RAN zu erlangen. Dabei kann der Angreifer die Richtlinien unverschlüsselt auf der A1 Schnittstelle mitlesen, Manipulieren und erneut in das Netzwerk einfließen lassen. Das führt zu einer Gefahr für das System, da der Angreifer somit die Verteilung von Endgeräten auf verfügbare Funkzellen verwalten kann.

Bei diesem praktischen Angriffsszenario konnte leider keine Darstellung oder Erkennung von der Durchsetzung der Richtlinie aufgezeichnet werden. Das liegt daran, das Rimedo Labs, die Betreiber der xApp, diese nicht aktualisiert hat und somit einen unterschiedlichen Versionsstand als das SD-RAN Framework hat. Dadurch wurde bei der Kommunikation über APIs kein einheitlicher Code mehr verwendet und die Richtlinien konnten in der xApp nicht erkannt werden.

Gegenmaßnahmen:

1. Security Awareness Training: Mitarbeiter sollten regelmäßig in Security Awareness geschult werden, um sich der Risiken von Social Engineering bewusst zu sein und verdächtige Aktivitäten zu erkennen.
2. Endpoint-Sicherheit: Verwendung von Endpunkt-Sicherheitslösungen, um Schutz vor Schadsoftware auf Geräten zu bieten, und regelmäßige Aktualisierungen der Sicherheitssoftware.
3. Verschlüsselung und Integritätsprüfung: Einsatz von Datenverschlüsselung, um sicherzustellen, dass selbst bei einem erfolgreichen Zugriff auf Daten diese nicht ohne weiteres gelesen werden können. Integritätsprüfungen können Manipulationen erkennen.
4. Starke Authentifizierung und Autorisierung: Implementierung von Mechanismen für starke Authentifizierung und Autorisierung, um sicherzustellen, dass nur autorisierte Benutzer auf sensible Systeme zugreifen können.

7 Fazit

7.1 Zusammenfassung

Diese Bachelorarbeit widmete sich einer gründlichen Erforschung des Open RAN, einem innovativen Ansatz im Bereich Mobilfunkzugangsnetze. Während bisherige Arbeiten theoretische Überblicke der O-RAN Alliance-Spezifikationen und Schwachstellenanalysen boten, hebt sich diese Arbeit durch ihre Schwerpunktsetzung auf die theoretische Ausarbeitung von spezifischen Angriffsszenarien und einem Versuch einer praktischen Umsetzung eines Angriffsszenarios ab.

In einem ersten Schritt wurde der Mobilfunk in seinen Grundzügen erklärt. Im Anschluss wurde der Fokus auf die Architektur von Open RAN gerichtet. Die Struktur dieses innovativen Ansatzes im Mobilfunk wurde aufgeschlüsselt, wodurch die spezifischen Komponenten und deren Zusammenspiel eingeführt wurden. Diese Architekturerklärung diente dazu, einen klaren Rahmen für die weiteren Untersuchungen zu schaffen.

Ein weiterer Schwerpunkt lag auf den Sicherheitsaspekten von Open RAN. Hier wurden nicht nur grundlegende Prinzipien der Netzwerksicherheit erörtert, sondern auch spezifische Sicherheitsüberlegungen im Kontext von Open RAN beleuchtet. Des Weiteren wurden verschiedene aktuelle Literatur verglichen und innovative Ansätze für Sicherheit im Open RAN aufgezeigt.

Anschließend wurde die richtlinienbasierte Steuerung als spezifischer Anwendungsfall von Open RAN aufgegriffen. Dabei wurde das allgemeine Prinzip erklärt, sowie der interne Workflow und eine Umsetzung durch eine xApp von Rimedo-Labs vorgestellt.

Darauf folgte eine Sicherheitsanalyse für Open RAN, die potenzielle Angreifer, Angriffsziele und Bedrohungen sowie die Sicherheitsanforderungen und Maßnahmen für den Near-RT RIC, Non-RT RIC, der E2 und A1 Schnittstelle aufgriff.

Des Weiteren wurden nicht nur die drei theoretischen Angriffsszenarien Manipulation einer A1-Richtlinie, Auslesen von Mobilitätsmustern und Kollision von Richtlinien entworfen, sondern es wurde auch ein Angriff auf eine Open RAN-Implementierung mit einer xApp für die richtlinienbasierte Steuerung in der Praxis durchgeführt. Die resultierenden Erkenntnisse dieser theoretischen und praxisorientierten Herangehensweise ermöglichen die Identifizierung von offenen Schwachstellen bezüglich der Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit.

Die besondere Betonung lag auf der Wichtigkeit der Umsetzung von Sicherheitsprotokollen gemäß den Security-Spezifikationen. Dabei wird nicht nur zur Unterstützung von Sicherheitsprotokollen, sondern zur dringenden Umsetzung dieser geraten. Insbesondere im Hinblick auf die Verschlüsselung des Datenverkehrs, um die Architektur des Zero-Trust-Prinzips zu erhalten. Diese Arbeit trägt nicht nur zur aktuellen Forschung im Bereich Open RAN bei, sondern gewährt auch praxisnahe Einblicke in die Sicherheitsaspekte und die Implementierung dieser vielversprechenden Technologie. Insgesamt stellt diese Bachelorarbeit einen wesentlichen Beitrag dar, indem sie die Lücke zwischen theoretischer Konzeption und praktischer Realisierung im Kontext von Open RAN überbrückt und Sicherheitsfragen hinterfragt.

7.2 Ausblick

Der Blick in die Zukunft des Open RAN bietet vielversprechende Perspektiven für weiterführende Forschung und Entwicklung. Eine entscheidende Richtung liegt in der fortlaufenden Verbesserung der Sicherheitsmechanismen im Open RAN. Die identifizierten potenziellen Schwachstellen und Angriffsszenarien unterstreichen die Notwendigkeit, robuste Sicherheitsprotokolle zu entwickeln und zu implementieren, um die Vertraulichkeit, Integrität und Verfügbarkeit des Netzwerks zu stärken.

Ein vielversprechender Ansatz für zukünftige Forschung liegt in der Integration von Machine Learning (ML) und Künstlicher Intelligenz (KI) in xApps. Dies könnte dazu beitragen, proaktive Sicherheitsmaßnahmen zu implementieren und Anomalien im Netzwerkverhalten frühzeitig zu erkennen, um potenzielle Bedrohungen effektiv abzuwehren.

Die Open-Source-Natur von Open RAN eröffnet Raum für aktive Community-Beteiligung. Zukünftige Entwicklungen könnten darauf abzielen, Open-Source-Lösungen weiter zu verbessern, um sie sicherer und widerstandsfähiger gegenüber potenziellen Bedrohungen zu machen.

Literaturverzeichnis

[3GP23] 3GPP. *NG-RAN Architecture description*. Techn. Ber. European Telecommunications Standards Institute (ETSI), 2023.

[AM23] Aly Sabri Abdalla und Vuk Marojevic. *End-to-End O-RAN Security Architecture, Threat Surface, Coverage, and the Case of the Open Fronthaul*. 2023. arXiv: 2304.05513 [cs.CR].

[AK23] Cezary Adamczyk und Adrian Kliks. „Conflict Mitigation Framework and Conflict Detection in O-RAN Near-RT RIC“. In: *IEEE Communications Magazine* (2023), S. 1–7. ISSN: 1558-1896. DOI: 10.1109/mcom.018.2200752.

[All23a] O-RAN Alliance. *O-RAN A1 interface: Transport Protocol*. Techn. Ber. O-RAN Alliance, 2023.

[All23b] O-RAN Alliance. *O-RAN Non-RT RIC Architecture*. Techn. Ber. O-RAN Alliance, 2023.

[All23c] O-RAN Alliance. *O-RAN Work Group 1 (Use Cases and Overall Architecture) Use Cases Detailed Specification*. Techn. Ber. O-RAN Alliance, 2023.

[All23d] O-RAN Alliance. *O-RAN Work Group 2 (Non-RT RIC and A1 Interface) Use Cases and Requirements*. Techn. Ber. O-RAN Alliance, 2023.

[All23e] O-RAN Alliance. *O-RAN Working Group 1 (Use Cases and Overall Architecture) O-RAN Architecture Description*. Techn. Ber. O-RAN Alliance, 2023.

[All23f] O-RAN Alliance. *O-RAN Working Group 11 (Security Work Group) O-RAN Security Test Specifications*. Techn. Ber. O-RAN Alliance, 2023.

[All23g] O-RAN Alliance. *O-RAN Working Group 11 (Security Working Group) O-RAN Security Threat Modeling and Remediation Analysis*. Techn. Ber. O-RAN Alliance, 2023.

[All23h] O-RAN Alliance. *O-RAN Working Group 11 (Security Working Group) Security Protocols Specifications*. Techn. Ber. O-RAN Alliance, 2023.

[All23i] O-RAN Alliance. *O-RAN Working Group 11 (Security Working Group) Security Requirements Specifications*. Techn. Ber. O-RAN Alliance, 2023.

[All23j] O-RAN Alliance. *O-RAN Working Group 2 (Non-RT RIC and A1 interface WG) A1 interface: Application Protocol*. Techn. Ber. O-RAN Alliance, 2023.

[All23k] O-RAN Alliance. *O-RAN Working Group 3 (Near-Real-time RAN Intelligent Controller and E2 Interface Workgroup) Near-RT RIC Architecture*. Techn. Ber. O-RAN Alliance, 2023.

[All23l] O-RAN Alliance. *Use Cases Analysis Report*. Techn. Ber. O-RAN Alliance, 2023.

[BKM22] Kurt Behnke, Jürgen Karla und Wilhelm Mülder. „Allgemeine Grundlagen des Mobilfunks“. In: *Grundkurs Mobilfunk und Mobile Business: Anwendungen, Technologien, Geschäftsfelder*. Wiesbaden: Springer Fachmedien Wiesbaden, 2022, S. 25–51. ISBN: 978-3-658-00141-4. DOI: 10.1007/978-3-658-00141-4_2.

[23a] *Bettercap*. https://www.bettercap.org/. Zugriff am 17. November 2023. 2023.

[DB23] Daniel Dik und Michael Stübert Berger. „Open-RAN Fronthaul Transport Security Architecture and Implementation“. In: *IEEE Access* 11 (2023), S. 46185–46203. DOI: 10.1109/ACCESS.2023.3274487.

[Dry23] Marcin Dryjanski. *The O-RAN Whitepaper 2023 – Security in O-RAN*. Techn. Ber. Rimedo-Labs, 2023.

[Eng21] Max Engelhardt. *Hacking und IT-Security für Einsteiger der leichte Weg zum IT-Security-Experten*. ger. Ausgabe 2021. 2021. ISBN: 9783966450829.

[Eri20] Ericsson. *Security Considerations for Open RAN*. https://www.ericsson.com/4a4b77/assets/local/security/security-considerations-open-ran.pdf. Zugriff am 17.Novemeber.2023. August 2020.

[23b] *Ettercap*. https://ettercap.github.io/ettercap/. Zugriff am 17. November 2023. 2023.

[FiG] MITRE FiGHT. *MITRE FiGHT - 5G Security Framework*. https://fight.mitre.org/. Accessed: 13.11.2023.

[23c] *GoPhish*. https://getgophish.com/. Zugriff am 17. November 2023. 2023.

[Gro+23] Joshua Groen u. a. *Implementing and Evaluating Security in O-RAN: Interfaces, Intelligence, and Platforms*. 2023. arXiv: 2304.11125 [cs.CR].

[HK23] Marcin Hoffmann und Pawel Kryszkiewicz. „Signaling Storm Detection in IIoT Network based on the Open RAN Architecture“. In: *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops INFOCOM WKSHPS*. IEEE, Mai 2023. DOI: 10.1109/infocomwkshps57453.2023.10226043.

[Kle+22] Felix Klement u. a. *Open or not open: Are conventional radio access networks more secure and trustworthy than Open-RAN?* 2022. arXiv: 2204.12227 [cs.CR].

[Köp+22] Stefan Köpsell u. a. *Open-RAN Risikoanalyse*. Techn. Ber. Bundesamt für Sicherheit in der Informationstechnologie, 2022.

[Lab] Rimedo Labs. *Rimedo Labs - Who We Are*. https://rimedolabs.com/about/. Accessed: 21.10.2023.

[Liy+23] Madhusanka Liyanage u. a. „Open RAN security: Challenges and opportunities“. In: *Journal of Network and Computer Applications* 214 (2023), S. 103621. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2023.103621.

[23d] *Maltego*. https://www.maltego.com/. Zugriff am 17. November 2023. 2023.

[Mim+22] Dudu Mimran u. a. *Evaluating the Security of Open Radio Access Networks*. 2022. arXiv: 2201.06080 [cs.CR].

[23e] *Nessus*. https://www.tenable.com/products/nessus. Zugriff am 17. November 2023. 2023.

[23f] *OpenVAS*. https://www.openvas.org/. Zugriff am 17. November 2023. 2023.

[Pol+23] Michele Polese u. a. „Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges“. In: *IEEE Communications Surveys & Tutorials* 25.2 (2023), S. 1376–1411. DOI: 10.1109/COMST.2023.3239220.

[Rah+22] Talha F. Rahman u. a. *Network and Physical Layer Attacks and countermeasures to AI-Enabled 6G O-RAN*. 2022. arXiv: 2106.02494 [cs.CR].

[23g] *Recon-ng*. https://github.com/lanmaster53/recon-ng. Zugriff am 17. November 2023. 2023.

[Rim23] Rimedo-Labs. *Traffic Steering xApp Technical Specification*. Techn. Ber. Rimedo-Labs, 2023.

[Ros+20] Scott Rose u. a. *Zero Trust Architecture*. en. Aug. 2020. DOI: https://doi.org/10.6028/NIST.SP.800-207.

[23h] *SD-RAN Dokumentation*. https://docs.sd-ran.org/master/introduction.html. letzter Zugriff am 17. November 2023. 2023.

[SSK20] Sameer Kumar Singh, Rohit Singh und Brijesh Kumbhani. „The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities“. In: *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. 2020, S. 1–6. DOI: 10.1109/WCNCW48565.2020.9124820.

[23i] *Social-Engineer Toolkit (SET)*. https://www.trustedsec.com/social-engineer-toolkit/. Zugriff am 17. November 2023. 2023.

[Thi+23] Prabhu Kaliyammal Thiruvasagam u. a. *Open RAN: Evolution of Architecture, Deployment Aspects, and Future Directions*. 2023. arXiv: 2301.06713 [cs.NI].

[Tri20] Ulrich Trick. *5G: Eine Einführung in die Mobilfunknetze der 5. Generation*. De Gruyter Oldenbourg, 2020. ISBN: 978-3-11-069999-9.

[23j] *Wireshark User's Guide*. https://www.wireshark.org/docs/wsug_html/. Wireshark Project. 2023.

[Xav+23] Bruno Missi Xavier u. a. *Machine Learning-based Early Attack Detection Using Open RAN Intelligent Controller*. 2023. arXiv: 2302.01864 [cs.NI].

[Zel22] Anatoly Zelenin. *Apache Kafka von den Grundlagen bis zum Produktiveinsatz*. ger. Hanser eLibrary. 2022. ISBN: 9783446470460.

[Zwa23] Amy Zwarico. „O-RAN Security“. In: *Open RAN*. John Wiley I& Sons, Ltd, 2023. Kap. 8, S. 121–136. ISBN: 9781119886020. DOI: https://doi.org/10.1002/9781119886020.ch8. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119886020.ch8.

Eidesstattliche Erklärung

Hiermit versichere ich – Johannes Micketeit – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 20. November 2023

Ort, Datum Johannes Micketeit