ADRF 数据检索订阅的安全分析

本报告的其他语言版本：English。

19.7.0 · ProVerif+ Agent 报告

文稿	5G System; Analytics Data Repository Services; Stage 3（协议定义）
检查的性质	13 条：6 条成立，5 条不成立（5 个攻击），2 条无答案
发现	2

目录

1 协议和范围
2 发现
3 检查过的性质
4 本次分析的局限
附录 A：逐条性质的解读
附录 C：主张和证明器的答案
附录 D：分析是怎么跑的

1 协议和范围

Nadrf_DataManagement API 让 NF 服务消费者向 ADRF（分析数据存储功能）请求其已存储的分析或数据，以及新记录到达后随之而来的通知。参与方有三方：发起订阅的 NF 服务消费者、存储记录并发送通知的 ADRF，以及充当 OAuth 2.0 授权服务器并签发访问令牌的 NRF。

消费者发送一个 NadrfDataRetrievalSubscription，其中带有 notifCorrId、notificationURI、时间窗口，以及一个分析订阅或数据订阅。ADRF 创建该订阅，分配 subscriptionId，以 201 Created 响应并带上 Location 头，随后向 notificationURI 发送 NadrfDataRetrievalNotification，其中带有 notifCorrId、时间戳以及分析、数据或取回指令。消费者存储每条通知并应答 204 No Content；对订阅执行 DELETE 会将其删除。

参与方：

- **NF 服务消费者**：发起订阅并接收通知。持有：NF 身份、长期凭据、回调地址（notificationURI）。
- **ADRF**：创建订阅、发送通知、删除订阅。持有：已存储的分析记录、订阅存储。
- **NRF**：签发 OAuth 2.0 访问令牌。持有：授权服务器角色。

图 1 展示了文稿所规定的交互流程。

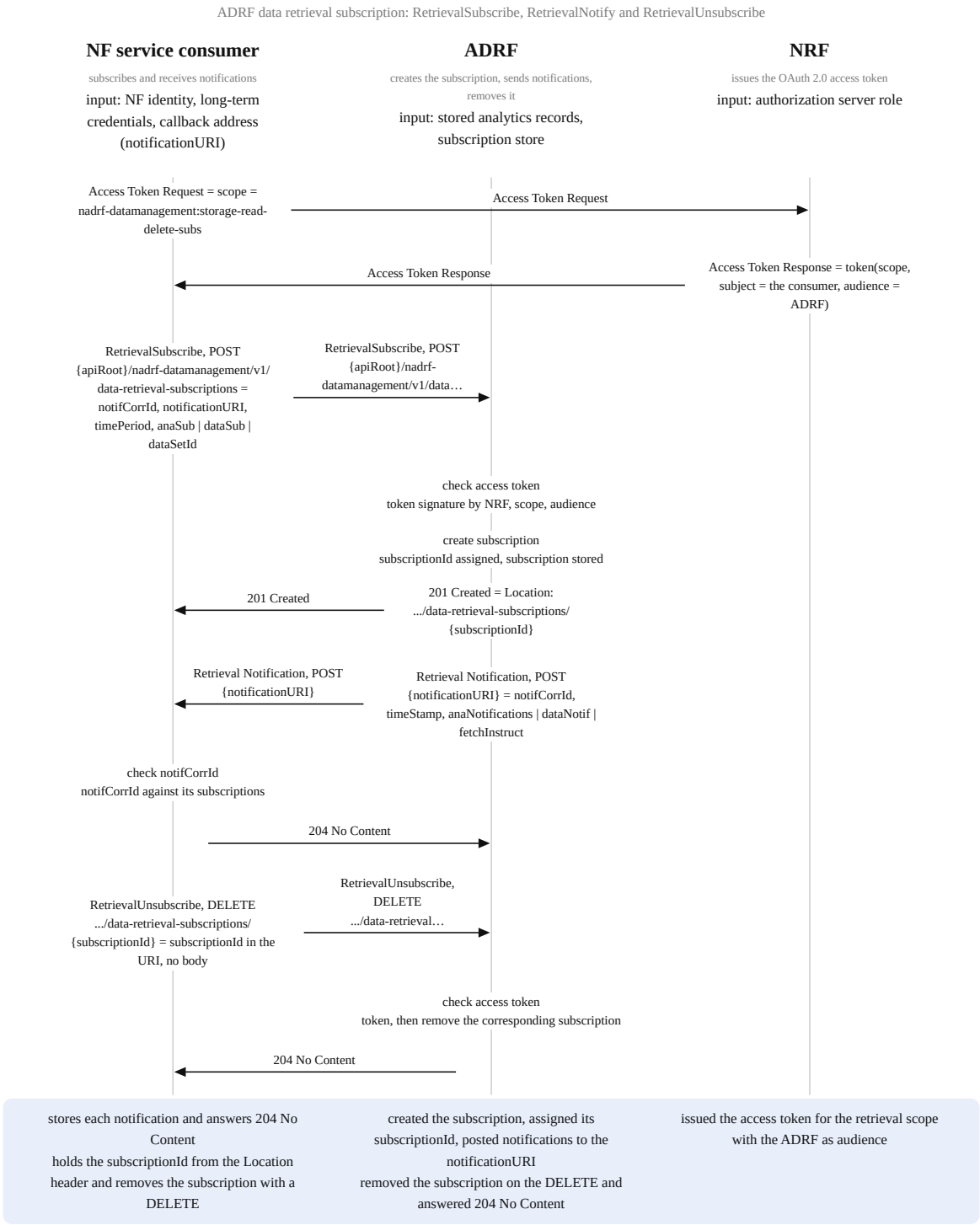


图 1. 文稿所规定的协议。

分析范围见附录 D：分析了什么、省略了什么，以及依据的是文稿的哪些章节。

2 发现

本次分析有 2 项发现：以下内容文稿并未写明。其中 2 项，文稿是否写明尚不明确。

订阅交互本身是站得住的：消费者只接受自己发送的值，ADRF 只接受 NRF 已授权的 NF 身份，通知载荷不会到达其他 NF。但发现了两处弱点。通知仅依靠每个消费者必须唯一的 notifCorrId 和订阅者指定的回调地址来标识其所属订阅，因此一个 NF 的通知可能被另一个 NF 的端点当作该 NF 自己的通知接收。另外，

只要某个持有检索范围的 NF 能说出 subscriptionId，它就能删除任意 NF 的检索订阅，因为文稿没有为 DELETE 规定任何归属规则。全部十三个目标都有答案；其中两个目标依赖的模型对传输和订阅生命周期做了理想化处理，因此删除条件与通知顺序的结论仍带有保留。

攻击图沿用图 1 的列。红色表示攻击者的行为。黄色标记的是某一方发送了规定交互在该时点本不应有的消息，以及攻击所破坏的结果。

2.1 任何持有检索范围的 NF 都能删除另一个 NF 的检索订阅

类型	协议规范本身的弱点
是否新发现	不明确：文稿只写了一部分，或没能检索
价值	中
涉及的性质	Z03, Z05（见第 3 节）
原文	s0021, s0032, s0377, s0200
依据	ProVerif 攻击查询 `z03_owner`；图 2 和图 3

发生了什么。 任何持有该 API 范围的 NF 都能删除任意其他 NF 的检索订阅，因为 DELETE 只携带 subscriptionId、没有消息体，而 ADRF 没有规则将请求者与创建该订阅的 NF 进行比对。攻击者用自己的有效令牌、带上受害者订阅的 subscriptionId 发送 DELETE；ADRF 删除该订阅并向攻击者应答 204 No Content。受害者始终不会被告知其订阅已被删除。

攻击者需要什么。 攻击者是一个合法的 NF 服务消费者，拥有自己的凭据，以及一个带有范围 "nadrf-datamanagement:storage-read-delete-subs" 的访问令牌，文稿允许该 API 的每个消费者持有此范围。攻击者还需要受害者订阅的 subscriptionId，而文稿既不要求它保密，也不要求它不可猜测。

攻击者能得到什么。 受害者的检索订阅在其不知情的情况下终止：它不再收到自己订阅的分析，而攻击者可以对其能说出标识符的任意订阅重复此操作，从而拒绝向选定的 NF 提供分析服务。

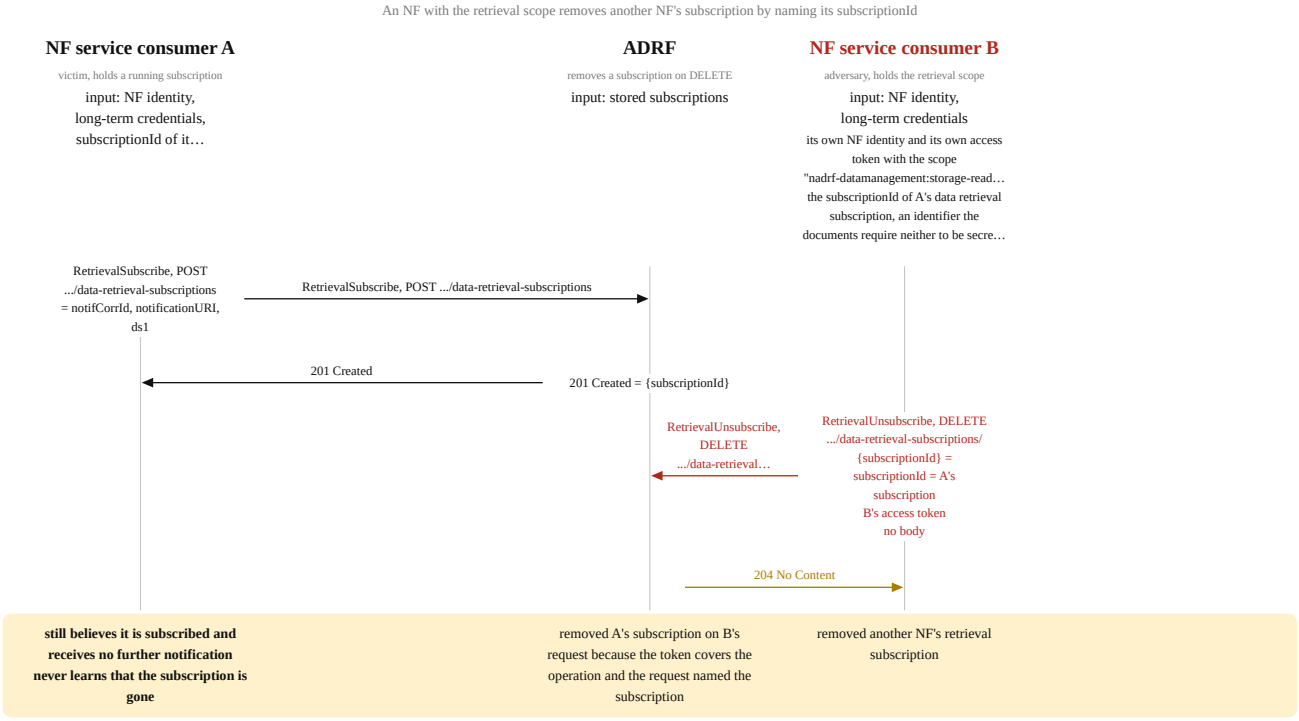


图 2. 持有检索范围的 NF 通过说出 subscriptionId 删除另一个 NF 的订阅。性质 Z03；攻击查询 z03_owner。

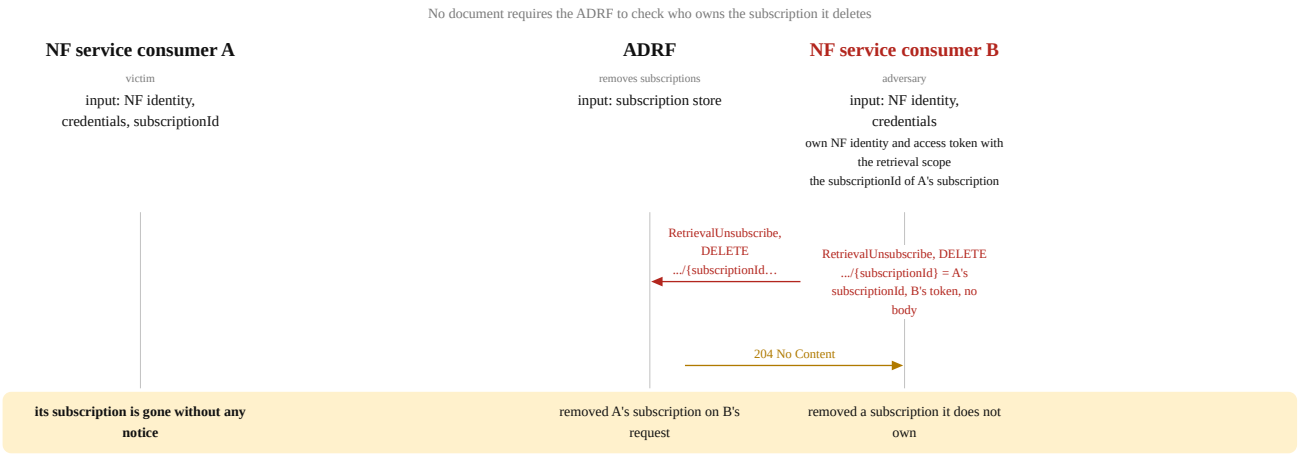


图 3. 没有任何文稿要求 ADRF 检查它所删除的订阅归谁所有。性质 Z05；攻击查询 z03_owner。

拟议修改。 创建订阅时，将请求方 NF 的身份与订阅一同存储；在删除 Individual ADRF Data Retrieval Subscription 时，除非请求者就是创建该 subscriptionId 的 NF，否则以 403 Forbidden 拒绝请求。另一种做法是为每个订阅定义资源级范围，使得持有检索范围不足以授权删除另一个消费者的订阅。

用 ProVerif 复核。 未检查：该修改是 ADRF 执行的一项检查，本可以建模，但本次会话在运行带归属检查的模型副本之前就结束了，因此该建议未经检查。

2.2 为某个 NF 的订阅生成的通知被另一个 NF 的端点接受

类型	协议规范本身的弱点
是否新发现	不明确：文稿只写了一部分，或没能检索 (ts-29-500:6152)
价值	中
涉及的性质	D01, A08, U01（见第 3 节）
原文	s0325, s0328, s0045, s0057
依据	ProVerif 攻击查询 `d01_own`；图 4、图 5 和图 6

发生了什么。 通知携带订阅者选定的 notifCorrId，但不携带 subscriptionId，也不携带订阅者的身份，ADRF 将其发送到订阅所指定的 notificationURI。若某个 NF 将另一个 NF 的回调地址指定为自己的 notificationURI，并使用相同的 notifCorrId，则其通知会被投递到那个 NF，而该 NF 的端点按 notifCorrId 匹配这些通知，并将其作为自己订阅的通知接受。接收方 NF 于是持有了它并未订阅的数据集的分析。

攻击者需要什么。 攻击者是一个合法的 NF 服务消费者，拥有自己的凭据和令牌，并且需要两样文稿并未保护的东西：另一个 NF 的回调地址，这是一个可达的网络地址；以及那个 NF 使用的 notifCorrId 值，该值只需在每个消费者之内唯一，无需不可猜测。

攻击者能得到什么。 消费者存储并据以行动的是它并未订阅的数据集的分析，它基于这些通知做出的任何自动化决策都建立在另一个 NF 的数据之上。它无法将这些通知与自己的通知区分开，并向 ADRF 应答 204 No Content，ADRF 于是将该通知记录为已投递。

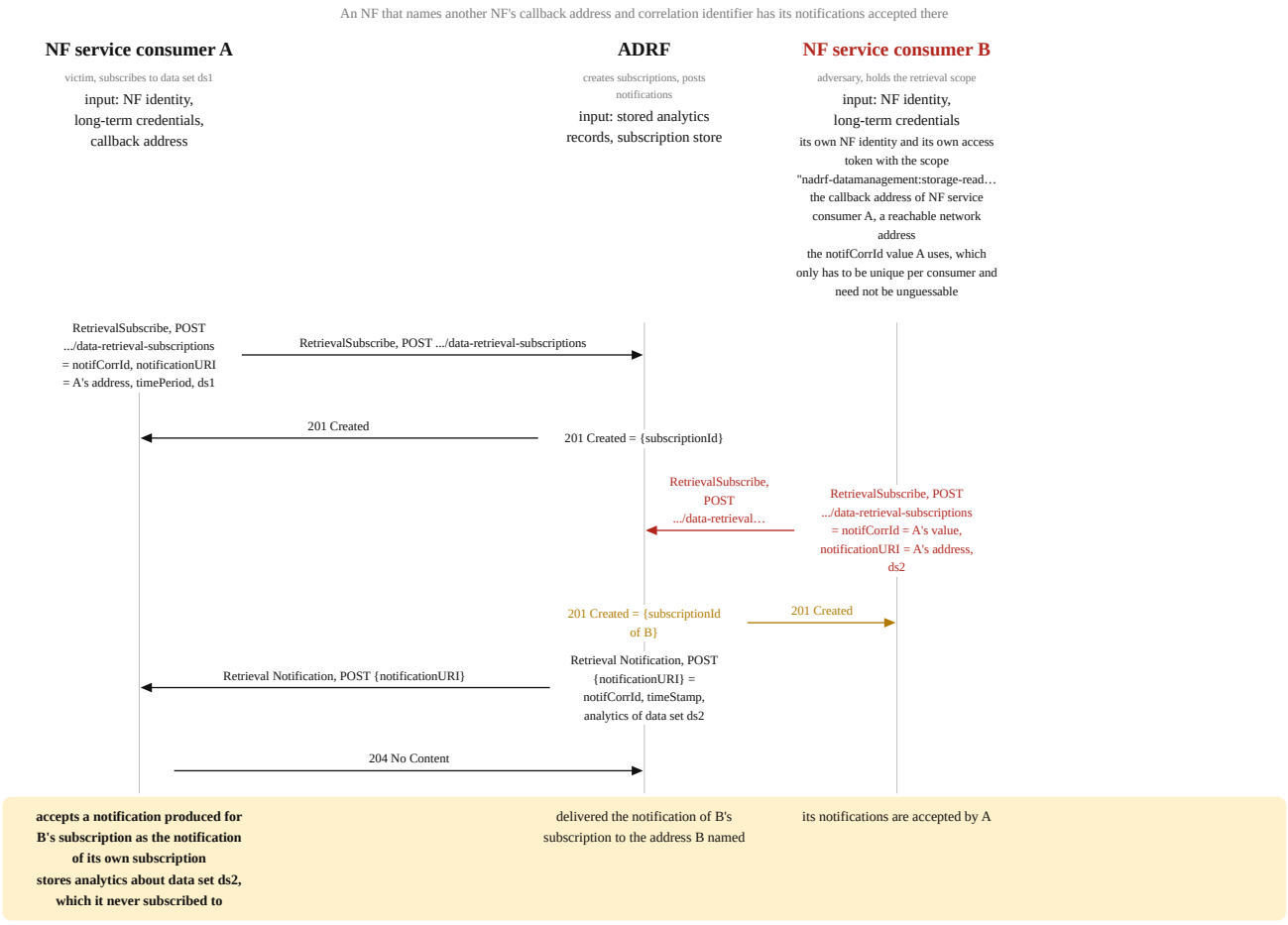


图 4. 指定另一个 NF 的回调地址和关联标识符的 NF，其通知会在那里被接受。性质 D01；攻击查询 d01_own。

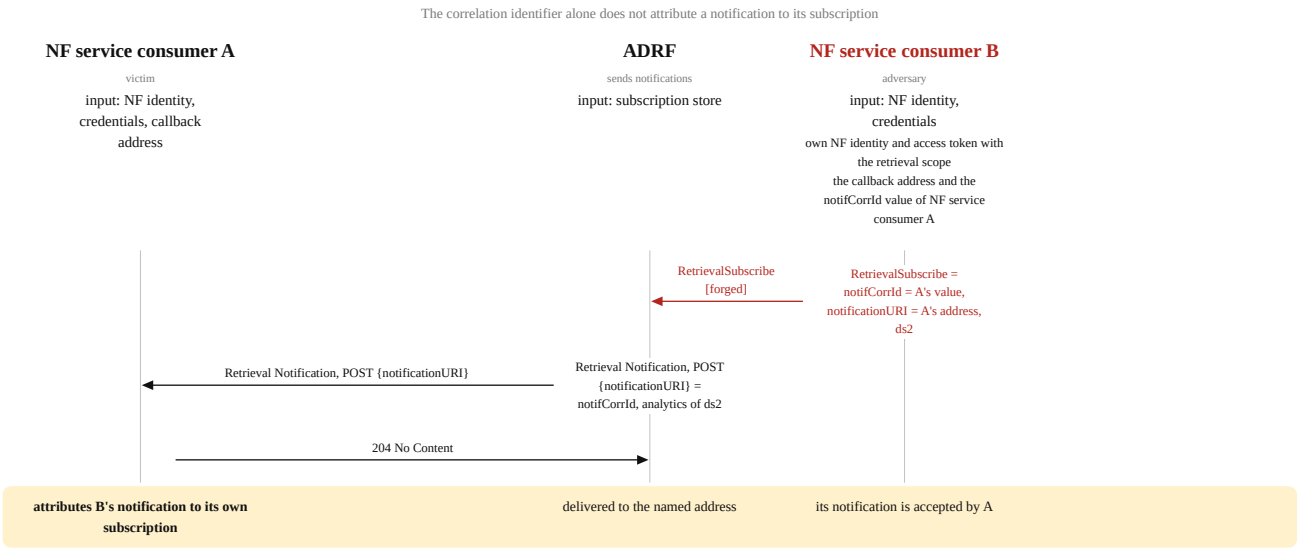


图 5. 仅凭关联标识符无法将通知归属到其订阅。性质 A08；攻击查询 d01_own。

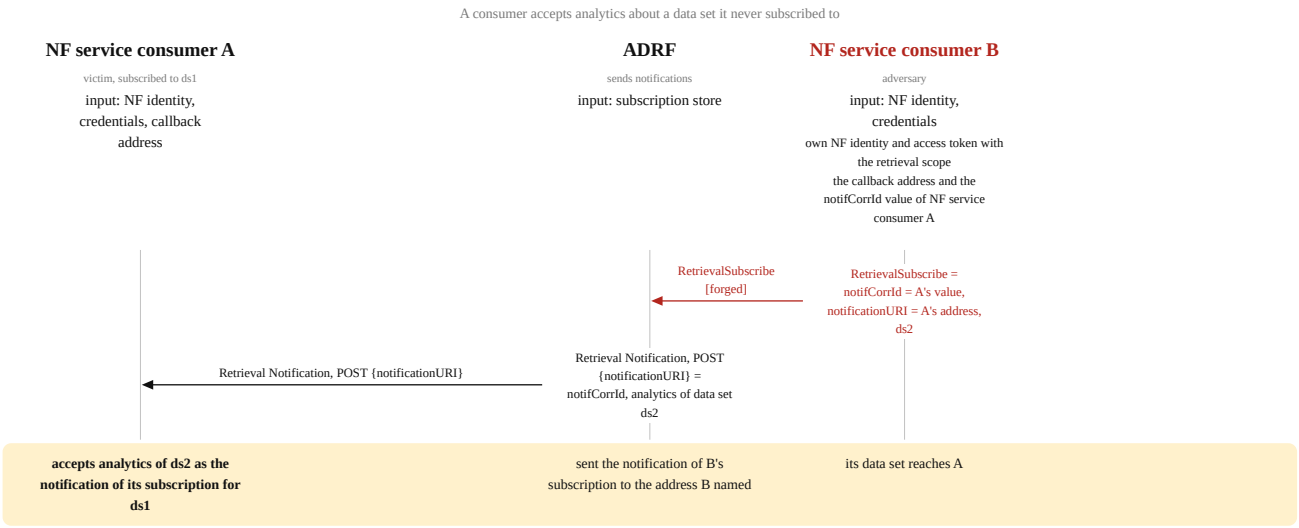


图 6. 消费者接受了它从未订阅的数据集的分析。性质 U01；攻击查询 d01_own。

拟议修改。 在 NadrfDataRetrievalNotification 中携带 subscriptionId，并要求消费者仅当该标识符属于其自己的某个订阅时才接受通知；或者在订阅时将 notificationURI 绑定到订阅方 NF，并拒绝属于另一个 NF 的 notificationURI。

用 ProVerif 复核。 未检查：该修改是通知中的一个字段加上消费者端的一项检查，因此可以建模；但并未建模，因为本次会话在运行带 subscriptionId 的通知载荷模型副本之前就结束了，因此该建议未经检查。

3 检查过的性质

每一行是一条提交给 ProVerif 的性质。结果是 ProVerif 在分析者的协议模型上给出的答案；价值以及文稿是否已经写明该结果，是分析者的判断，理由见附录 A。

ID	性质	假定的泄露	结果	价值	文稿是否已写
A03	**订阅请求与已创建的订阅匹配。** 每次消费者接受订阅创建确认（201 与 Location）时，都存在一个 ADRF 的真实运行，其收到的订阅请求中的 notifCorrId、notificationURI、订阅内容与消费者本地值一致；每次 ADRF 接受订阅请求时，都存在一个以该 ADRF 为对端的消费者真实运行。	无	成立	中	是
A04	**订阅与通知为单射。** 每次 ADRF 接受一个订阅请求都对应消费者一次不同的真实订阅运行（同一 notifCorrId 不被重复接受为两个订阅）；每次消费者接受一条通知都对应 ADRF 针对该订阅的一次不同通知发送。	无	成立	低	不明确
D01	**通知来自 ADRF。** 每次消费者接受一条 Retrieval Notification 时，都存在该 ADRF 针对该消费者该订阅的真实通知发送，notifCorrId 与订阅一致，通知载荷未被替换。	无	不成立（1 个攻击）	中	不明确
A08	**通知属于其自身的订阅。** 消费者接受的每条通知，其 notifCorrId 与 notificationURI 都对应消费者此前为该订阅提交的值，且该通知不被归入其他订阅。	无	不成立（1 个攻击）	中	不明确
Z01	**操作需要有效的访问令牌。** 每次 ADRF 提交订阅创建或取消操作、每次消费者接受通知，都必须存在授权服务器签发的、覆盖该操作的有效许可（scope 与受众匹配）。	无	成立	中	是
Z03	**仅订阅所有者可移除该订阅。** 每次以某 subscriptionId 提交的取消操作，其许可的主体、受众与资源都对应创建该订阅的消费者；许可不能被用于其他订阅或其他操作。	无	不成立（1 个攻击）	中	不明确

ID	性质	假定的泄露	结果	价值	文稿是否已写
S05	**取消订阅后无通知。** 在 ADRF 成功处理取消（或发出 terminationReq）之后，不再针对该 subscriptionId 产生新的通知，旧 subscriptionId 也不能再次触发删除副作用。	无	只确认可达	低	不明确
R02	**每条通知至多被接受一次。** 同一逻辑通知实例（同一 notifCorrId 与同一内容）在消费者处至多被接受一次。	无	无答案	无	不明确
K02	**通知载荷保留在消费者处。** 通知载荷与 fetchInstruct 取回的数据/分析，除订阅的消费者与 ADRF 外，其他实体（含其他合法 NF）不能推导。	无	成立	中	是
K01	**访问令牌不可推导。** 攻击者不能推导出授权服务器签发的有效 OAuth2 token（含 scope 与受众）。	无	成立	低	是
U01	**通知与数据集保留在订阅者处。** 通知的 notificationURI 与 dataStoreId 与创建该订阅的消费者提交值一致，且不被归入其他订阅。	无	不成立（1 个攻击）	中	不明确
U02	**通知真实且订阅已终止。** 通知来自订阅的 ADRF 且属于该订阅；取消后该订阅不再产生通知。	无	成立	中	是
Z05	**跨 NF 访问策略具有规范性依据。** 问题询问跨 NF 数据访问策略的规范依据，而非某个可判定的轨迹性质。	无	不成立（1 个攻击）	中	不明确

所提出的问题及分析者的回答：

问题	回答	依据
Q1: 订阅者与 dataStoreId/notificationURI 的绑定	否：notificationURI 由订阅者在订阅时自行选择，ADRF 从不将其与订阅者自身的身份进行比对；通知既不携带 subscriptionId，也不携带订阅者的身份，因此消费者无法检查某条通知是否属于它创建的订阅。唯一的绑定是 notifCorrId，而原文要求它仅在每个消费者内唯一，且从不要求它是保密的。	A08, U01, D01
Q2: 通知真实性、跨订阅替换与有限状态取消	通知的来源是真实的、未被修改，且携带消费者所选的 notifCorrId，其载荷也不会到达其他 NF；但通知归属于哪个订阅这一点不受保护，订阅的移除也没有与其创建者绑定，因此有限状态取消规则依赖于 ADRF 对自身移除标记的检查，而本模型并未对此给出结论。	D01, U02, S05, Z03
Q3: 对其他合法 NF 的数据访问策略必须有规范依据	不存在这样的规范依据：文稿给出的唯一规则是 OAuth 2.0 的 scope 集合，其中一个资源/操作级别的 scope 就覆盖了持有它的每个消费者的检索订阅、取消订阅和通知操作。TS 29.501 4.10 要求这种仅靠 scope 的设计不得授予不当的删除权限，这是对 API 设计者的要求，而被分析的 API 并未将其落实为按订阅区分的规则。	Z05, Z03

4 本次分析的局限

这里的结果是协议模型上的结果：它说明所规定的消息与检查提供了什么、未能提供什么，其中密码学被视为理想情况，与具体实现无关。

正如分析者所述，本次分析不覆盖以下内容：

- 传输被建模为发送方的签名加上对接收方的加密，而非 TLS 会话：因此消息保护在模型中是可转移的，这通过将接收方地址绑定在已签名的通知内来补偿。真实协议通过双向 TLS 阻止此类中继。
- 模型的 subscriptionId 空间只包含一个值，因此“一个请求不会被接受为两个订阅”这一主张是弱的；随机标识符是否能防止删除另一个 NF 的订阅，此处未给出结论。
- 订阅生命周期被建模为单调表加上移除标记：ADRF 移除订阅后不再发送通知这一顺序性主张未被证明，只证明了 ADRF 在通知前会检查该标记。

- 消费者被建模为单次运行，因此同一条通知是否会被存储两次无法判定；ProVerif 对该查询返回了无答案。
- 只分析了不含 SUPI 或 GPSI 的分支；涉及 UE 数据（s0004）所需的与 UDM 进行的用户同意检查不在模型范围内。
- 外部数据收集以及通过 Nadrf_DataManagement_RetrievalRequest 取回数据（s0052、s0062）不在被分析的子流程范围内，因此取回数据的保护问题没有得到回答。
- OAuth 2.0 token 被建模为 NRF 对 scope、主体和受众的签名；TS 29.510 的过期、撤销和 token 请求流程未被建模。
- 时序、可用性、资源耗尽以及原语的计算强度无法查询，也不在覆盖范围内。

宿主记录的内容：

- 无答案：S05、R02。
- 证明器没有给出结论的查询（附录 C）：R02: r02_once。这条性质问过了，证明器既没有证明它也没有找到攻击，所以结果写“无答案”。
- 分析者为性质设定了攻击者条件，但既没有证明也没有攻击的（附录 D）：S05 在 `network` 下。
- 没带对照查询的证明（附录 C）：A03: a03_created, A03: a03_sub, A04: a04_sub, D01: d01_origin, A08: d01_origin, Z01: a03_sub, Z01: z01_unsub, Z03: z01_unsub, K02: k02_secret, K01: a03_sub, U01: d01_origin, U02: d01_origin, Z05: a03_sub。对照查询是这条性质的一个必然不成立的变体；没有它，就没有东西表明这条性质照现在的写法在模型里有可能不成立。

每条性质还有哪些问题悬而未决，以及什么能解决它，见附录 A。

附录 A：逐条性质的解读

以下是分析者对证明器回答的解读，而非证明器自身的回答：这些回答对协议意味着什么、该解读依据什么、能否照原样使用，以及它的价值。

A03: 订阅请求与已创建的订阅匹配

每次消费者接受订阅创建确认（201 与 Location）时，都存在一个 ADRF 的真实运行，其收到的订阅请求中的 notifCorrId、notificationURI、订阅内容与消费者本地值一致；每次 ADRF 接受订阅请求时，都存在一个以该 ADRF 为对端的消费者真实运行。

消费者的 201 Created 只能携带它自己发送的订阅值，因此它不可能被诱导接受为另一个请求创建的订阅；而 ADRF 只接受来自 NRF 已为其签发检索 token 的 NF 的订阅。这两点对外部攻击者成立，对持有自身有效凭据的 NF 也成立。

- 结果：成立；回答了所问的问题：是；可用
- 价值中：订阅交换的两个方向是消费者此后收到的一切的基础，该结果是基于对 NF 身份进行认证的传输之上的证明；需要注意的是传输是被理想化的。
- 文稿是否已写：是。文稿中的位置：ts-33-501:2871
- 依据：传输被建模为发送方签名加上对接收方的加密，而非 TLS 会话；消费者只运行一个订阅；同一消费者的第二个订阅未被建模。
- 下一步：对这条主张无需再做；同一目标的通知识别由 D01 回答。

A04: 订阅与通知为单射

每次 ADRF 接受一个订阅请求都对应消费者一次不同的真实订阅运行（同一 notifCorrId 不被重复接受为两个订阅）；每次消费者接受一条通知都对应 ADRF 针对该订阅的一次不同通知发送。

一个订阅请求被接受为一个订阅，这一点是针对单个订阅标识符展示的。目标的第二半，即一条通知只被接受一次，并未得到结论：消费者被建模为单次运行，而能展示这一点的查询返回了无答案。

- 结果：成立；回答了所问的问题：部分；有保留地可用
- 价值低：答案依赖于退化的标识符空间和单次运行的消费者，因此对协议说明甚少。
- 文稿是否已写：不明确
- 依据：模型的 subscriptionId 空间是一个单值，因此单射性主张是弱的；消费者是单个进程，因此无法观察到重复接受。
- 未决：消费者是否会将重复的通知存储两次；来自同一 NF 的重复订阅请求是否会被接受两次。
- 下一步：建立一个具有真实标识符空间、且消费者会保留其已存储通知的有界模型。

D01: 通知来自 ADRF

每次消费者接受一条 Retrieval Notification 时，都存在该 ADRF 针对该消费者该订阅的真实通知发送，notifCorrId 与订阅一致，通知载荷未被替换。

被接受的通知始终是真实的、未被修改的 ADRF 通知，且携带消费者所选的关联标识符；但它未必是为该消费者的某个订阅而生成的：一个 NF 只要写出该消费者的回调地址及其 notifCorrId，就能让 ADRF 把自身的通知投递到那里，而消费者会将其当作自己的通知接受。

- 结果：不成立（1 次攻击）；回答所问：部分；有保留地可用
- 价值中：它隔离出一个特定的缺口：通知没有携带其所属订阅的标识符，而原文依赖一个只要求按消费者唯一的值。
- 在文稿中说明：不明确。所依赖的原文：ts-29-500:6152
- 依赖：攻击者知道受害者的回调地址（公开）及其 notifCorrId 值，原文只要求该值按消费者唯一。；ADRF 签名内的接收方绑定代替了真实传输中的双向 TLS 绑定。
- 待解决：在没有 subscriptionId 的情况下，通知究竟能否绑定到其订阅。
- 下一步：在模型中向通知加入 subscriptionId，并重新运行这些查询，看攻击是否被消除。

A08: 通知属于其自身的订阅

消费者接受的每条通知，其 notifCorrId 与 notificationURI 都对应消费者此前为该订阅提交的值，且该通知不被归入其他订阅。

相关标识符不足以将通知归属于某个订阅：因为它只要求对给定消费者唯一，而通知没有其他标识符，所以一条被接受的通知可能属于另一个消费者的订阅，而消费者从未请求过的数据集的分析结果会被当作自己的结果接受。

- 结果：不成立（1 次攻击）；回答所问：部分；有保留地可用
- 价值中：它展示了原文设计的相关标识符能做什么、不能做什么，并给出确切条件和发生错误归属的步骤。
- 在文稿中说明：不明确
- 依赖：攻击者知道受害者的 notifCorrId 值；随机选择的相关标识符可以阻止它，而原文没有这样要求。；消费者只按 notifCorrId 匹配，正如 s0328 对通用回调 URI 所建议的那样。
- 下一步：检查在通知中携带 subscriptionId 是否能消除攻击。

Z01: 操作需要有效的访问令牌

每次 ADRF 提交订阅创建或取消操作、每次消费者接受通知，都必须存在授权服务器签发的、覆盖该操作的有效许可（scope 与受众匹配）。

ADRF 接受的每个订阅和每次移除，都由 NRF 为该 NF 签发的、以 ADRF 为受众、覆盖检索范围的令牌所覆盖，因此没有消费者能用不是签发给它的令牌行事。令牌做不到的是区分消费者可以对哪个订阅行事，这正是其他目标所探究的授权问题。

- 结果：成立；回答所问：是；有保留地可用
- 价值中：它展示了 OAuth 2.0 检查能做什么、在哪里停止，这正是 RFC 6749 关于令牌到客户端绑定的警告所说的。
- 在文稿中说明：是。文稿中的说明位置：rfc6749:2879
- 依赖：令牌被建模为 NRF 对范围、主体和受众的签名；过期和吊销未建模。；API 的范围集被建模为覆盖检索操作的单个范围。
- 下一步：为每个订阅定义资源级范围，并询问相同的查询。

Z03: 仅订阅所有者可移除该订阅

每次以某 subscriptionId 提交的取消操作，其许可的主体、受众与资源都对应创建该订阅的消费者；许可不能被用于其他订阅或其他操作。

持有检索范围的 NF 可以移除任何它能够说出 subscriptionId 的检索订阅：DELETE 只携带该标识符，而 ADRF 没有所有权规则。后果是订阅所有者静默失去该订阅，而这只需要一个持有自己令牌的合法 NF。

- 结果：不成立（1 次攻击）；回答所问：是；有保留地可用
- 价值中：它把 TS 29.501 4.10 的要求转化为该 API 的具体失败，并给出关于标识符的确切条件。
- 在文稿中说明：不明确。所依赖的原文：ts-29-501:1801
- 依赖：攻击者能说出受害者的 subscriptionId；原文没有任何地方要求它不可猜测。；模型没有给 ADRF 所有权检查，因为原文没有规定。
- 下一步：在 ADRF 中加入所有权检查后重新运行，以确认攻击被消除。

S05: 取消订阅后无通知

在 ADRF 成功处理取消（或发出 terminationReq）之后，不再针对该 subscriptionId 产生新的通知，旧 subscriptionId 也不能再次触发删除副作用。

ADRF 在每条通知之前检查其移除标记，因此在这个模型中，对于已经移除的订阅绝不会产生通知；记录的证伪来自一个无顺序查询，它允许通知在移除之前出现，而协议允许这种情况。顺序性质本身在这里并未确定。

- 结果：仅可达；回答所问：部分；有保留地可用
- 价值低：目标的性质未被判定；只展示了 ADRF 检查的存在。
- 在文稿中说明：不明确
- 依赖：订阅移除是单调表中的标记，而不是真正的删除。；移除与通知的顺序在这个模型中无法表达。
- 待解决：作为顺序主张，ADRF 移除订阅后不再有通知。
- 下一步：一个有界模型，展开订阅生命周期，并展示针对已移除订阅的通知绝不会被接受。

R02: 每条通知至多被接受一次

同一逻辑通知实例（同一 notifCorrId 与同一内容）在消费者处至多被接受一次。

ProVerif 既不能证明也不能反驳消费者对每个订阅至多接受一个通知载荷；它产生的推导是过近似，而不是迹。协议本身在通知中没有重放检测，因此该性质取决于消费者和传输。

- 结果：无答案；回答所问：不确定；不可用
- 价值无：未获得答案，因此对该目标而言没有确立任何关于协议的东西。
- 在文稿中说明：不明确
- 依赖：消费者被建模为单次运行，因此无法观察到重复。；传输上的通知重放被建模的签名和加密阻止。
- 待解决：一条通知是否被存储两次，ProVerif 在这里无法判定。
- 下一步：一个带复制消费者和已存储通知表的模型。

K02: 通知载荷保留在消费者处

通知载荷与 fetchInstruct 取回的数据/分析，除订阅的消费者与 ADRF 外，其他实体（含其他合法 NF）不能推导。

除 ADRF 和通知所投递到的 NF 之外，任何实体都得不到被通知的分析结果，包括持有自己凭据的第二个合法 NF。保护来自传输，因此只有当通知绑定到它所投递到的地址时才成立，这正是模型中对接收方的签名所表示的。

- 结果：成立；回答所问：是；有保留地可用
- 价值中：对通知体的机密性证明，且对持有自己 NF 凭据的攻击者成立，但它依赖传输抽象。
- 在文稿中说明：是。文稿中的说明位置：ts-33-501:2871
- 依赖：载荷机密性完全依赖建模的传输；文稿没有为通知体提供端到端保护。；取回指令以及用 Nadrf_DataManagement_RetrievalRequest 取回数据不在模型范围内。
- 下一步：对通知体无需下一步；取回路径需要在模型中包含取回请求操作。

K01: 访问令牌不可推导

攻击者不能推导出授权服务器签发的有效 OAuth2 token（含 scope 与受众）。

没有任何操作会以 NRF 未签发给该 NF 的令牌被接受，因此网络攻击者和其他 NF 都无法伪造或推导出有效令牌；涉及的令牌只有 NRF 签发给其主体的那些。

- 结果：成立；回答所问：是；可用
- 价值低：它确认了 OAuth 2.0 要求已经陈述的内容，没有新的弱点。
- 在文稿中说明：是。文稿中的说明位置：rfc6749:2870
- 依赖：令牌格式被理想化为不可伪造的 NRF 签名。；只展示了不可伪造性；令牌值的不可猜测性是原语的性质，而不是这个模型的性质。
- 下一步：无。

U01: 通知与数据集保留在订阅者处

通知的 notificationURI 与 dataStoreId 与创建该订阅的消费者提交值一致，且不被归入其他订阅。

消费者可能被迫接受其从未订阅过的数据集的分析结果，作为自己订阅的通知，因为通知只用另一个 NF 可能共享的相关标识符来标识订阅，并且不携带它所请求的数据集标识符。确实成立的是通知被投递到消费者指定的地址，而这正是使错误归属成为可能的原因。

- 结果：不成立（1 次攻击）；回答所问：部分；有保留地可用
- 价值中：它把错误归属与消费者请求的具体属性联系起来，这正是编辑者需要看到的。
- 在文稿中说明：不明确
- 依赖：消费者只按 notifCorrId 匹配通知，而相关值不必不可猜测。；数据集标识符不携带在通知中，这正是原文所规定的。
- 下一步：在通知中携带 subscriptionId 并重新运行。

U02: 通知真实且订阅已终止

通知来自订阅的 ADRF 且属于该订阅；取消后该订阅不再产生通知。

每条被接受的通知都是一条真实的、未被修改的 ADRF 通知，带有消费者选择的关联标识符，因此即使面对恶意的 NF，来源与完整性也成立。消费者无法判断的是，该通知是否是为其自己的订阅而生成的；而其接受在移除之后即停止，仅仅是因为其自身的运行已经结束。

- 结果：Holds；是否回答了所问：是；可谨慎使用
- 价值：中。来源与完整性部分是对所建模传输的可靠证明；后半部分未作判定。
- 文稿中有说明：是。文稿中的位置：ts-33-501:2871
- 依据：消费者是一次运行，因此订阅的结束并未作为状态变化加以检验。；签名内部的接收方绑定代替了双向 TLS。

- 未决：目标的排序部分，即移除之后不会再有通知到达。
- 下一步：对订阅生命周期建立有界模型。

Z05: 跨 NF 访问策略具有规范性依据

问题询问跨 NF 数据访问策略的规范依据，而非某个可判定的轨迹性质。

文稿给出的唯一规则是 OAuth 2.0 范围，而它并不区分订阅的消费者：任何拥有检索范围的 NF 都可以对其能够指名的任意订阅调用 unsubscribe。因此，所分析的文稿除范围本身之外，并未给出跨 NF 限制的规范性依据，而模型显示了这留下了哪些空白。

- 结果：Fails（1 次攻击）；是否回答了所问：部分；可谨慎使用
- 价值：中。它回答了用户提出的治理问题，并指向要求 API 设计者对此进行核查的那一段。
- 文稿中有说明：不明确。所依据的原文：ts-29-501:1801
- 依据：答案是文稿的解读加上删除攻击，而非其自身的轨迹性质。；通过 Nadrf_DataManagement_RetrievalRequest 读取存储的分析数据不在模型范围内。
- 未决：哪个 NF 可以通过检索请求操作读取存储的分析数据，这一点任务中未涉及。
- 下一步：将模型扩展到检索请求操作，以同时回答读取一侧的问题。

附录 C：主张和证明器的答案

证明器对每个目标的记录：达到的级别、达到该级别的查询、是否检查了对照，以及各项攻击和分析者对每项攻击的判定。

目标	级别	界	查询	对照	攻击	主张
A03	proved		a03_created			3 proved
A04	proved		a04_sub			2 proved, 1 reachable
D01	proved		d01_origin		d01_own (holds_in_source)	2 proved, 1 attack
A08	proved		d01_origin		d01_own (holds_in_source)	1 proved, 1 attack
Z01	proved		a03_sub			3 proved
Z03	proved		z01_unsub		z03_owner (depends_on_ambiguity)	1 proved, 1 attack
S05	reachable		s05_reach		s05_none (model_artifact)	1 reachable, 1 attack
R02	reachable		k02_reach			1 reachable, 2 undecided
K02	proved		k02_secret			2 proved
K01	proved		a03_sub			2 proved
U01	proved		d01_origin		d01_own (holds_in_source)	1 proved, 1 attack
U02	proved		d01_origin			2 proved, 1 reachable
Z05	proved		a03_sub		z03_owner (holds_in_source)	1 proved, 1 attack

逐条主张，按分析者对每条主张的表述；每条主张的答案是证明器给出的：

目标	主张	攻击者	回答	查询	对照
A03	消费者接受的每个 201 Created 都携带消费者自己发送的订阅请求中的 notifCorrId、notificationURI、数据集合和 subscriptionId。	`network`	proved	a03_created	none
A03	消费者接受的每个 201 Created 都携带消费者自己发送的订阅请求中的 notifCorrId、notificationURI、数据集合和 subscriptionId，在另一个 NF 持有自己的凭据时也是如此。	`malicious_nf`	proved	a03_created	none
A03	ADRF 接受的每个订阅都是由 NRF 已向其签发具有检索范围访问令牌的 NF 请求的。	`malicious_nf`	proved	a03_sub	none
A04	同一订阅请求（相同的订阅者、notifCorrId、notificationURI 和数据集合）不会被接受为两个不同的订阅。	`malicious_nf`	proved	a04_sub	none
A04	订阅请求被 ADRF 接受，这正是目标后半部分所说的情形。	`network`	reachable	k02_reach	
A04	同一订阅请求被接受为一个订阅。	`network`	proved	a04_sub	none
D01	消费者接受的每条通知都是由 ADRF 发送的，具有相同的 notifCorrId 和相同的负载，且负载未被修改。	`malicious_nf`	proved	d01_origin	none
D01	消费者接受的每条通知都是由 ADRF 针对该消费者自己的某个订阅发送的。	`malicious_nf`	attack (holds_in_source)	d01_own	

目标	主张	攻击者	回答	查询	对照
D01	每条被接受的通知都是由 ADRF 以消费者选择的 notifCorrId 和未经修改的负载发送的。	network	proved	d01_origin	none
A08	被接受的通知只归属于消费者创建的订阅，绝不归属于另一个消费者的订阅。	malicious_nf	attack (holds_in_source)	d01_own	
A08	仅靠网络攻击者无法让消费者接受通知：每条被接受的通知都是真实的 ADRF 通知，携带消费者选择的关联标识符。	network	proved	d01_origin	none
Z01	ADRF 接受的每个检索订阅都受到 NRF 向请求 NF 签发的、针对检索范围并以该 ADRF 为受众的访问令牌的覆盖。	malicious_nf	proved	a03_sub	none
Z01	ADRF 接受的每次检索订阅移除都受到 NRF 向请求 NF 签发的、针对检索范围并以该 ADRF 为受众的访问令牌的覆盖。	malicious_nf	proved	z01_unsub	none
Z01	ADRF 接受的每个订阅都受到 NRF 向请求 NF 签发的令牌的覆盖。	network	proved	a03_sub	none
Z03	移除检索订阅的 NF 就是创建该订阅的 NF 服务消费者。	malicious_nf	attack (depends_on_ambiguity)	z03_owner	
Z03	ADRF 只有在收到网络攻击者无法发出的请求之后才会移除检索订阅并应答 204 No Content，因为该请求必须携带它无法获得的令牌。	network	proved	z01_unsub	none
S05	ADRF 向已存储订阅的 notificationURI 发送检索通知。	network	reachable	s05_reach	
S05	已移除的订阅不再产生任何通知。	network	attack (model_artifact)	s05_none	
R02	消费者接受针对其订阅的检索通知，因此达到了目标所说的情形。	network	reachable	k02_reach	
R02	消费者对每个订阅至多接受一个通知负载，因此一个通知实例不会被处理两次。	network	undecided	r02_once	
R02	消费者对每个订阅至多接受一个通知负载，在另一个 NF 持有自己的凭据时也是如此。	malicious_nf	undecided	r02_once	
K02	通知的负载不会到达 ADRF 和接收该通知的 NF 之外的任何实体；其他任何 NF 以及网络攻击者都不会得知该负载。	malicious_nf	proved	k02_secret	none
K02	除 ADRF 和被通知的 NF 之外，任何实体都不会得知通知的负载。	network	proved	k02_secret	none
K01	不会用授权服务器未向该 NF 签发的令牌接受任何操作，因此任何攻击者和任何其他 NF 都无法推导或伪造有效令牌。	malicious_nf	proved	a03_sub	none
K01	不会用授权服务器未向该 NF 签发的令牌接受任何操作。	network	proved	a03_sub	none
U01	被接受的通知报告的是消费者所订阅的数据集合，且其 notificationURI 是其自己订阅的 notificationURI。	malicious_nf	attack (holds_in_source)	d01_own	
U01	消费者接受的通知被投递到其指定的 notificationURI，且其负载不会到达任何其他实体。	network	proved	d01_origin	none
U02	消费者接受的每条通知都是 ADRF 的真实通知，未经修改，并携带消费者选择的 notifCorrId。	malicious_nf	proved	d01_origin	none
U02	消费者在接受通知后移除其订阅，因此达到了目标后半部分所说的移除情形。	network	reachable	run	

目标	主张	攻击者	回答	查询	对照
U02	每条被接受的通知都是真实的 ADRF 通知，且消费者随后移除其订阅。	`network`	proved	d01_origin	none
Z05	在所分析的文稿中存在一条规范性规则，阻止一个 NF 服务消费者移除另一个消费者的检索订阅；分析表明文稿中唯一的授权规则允许什么。	`malicious_nf`	attack (holds_in_source)	z03_owner	
Z05	仅靠网络攻击者既不能订阅也不能移除订阅，因为这两种操作都需要 NRF 向某个 NF 签发的令牌。	`network`	proved	a03_sub	none

附录 D：分析是怎么跑的

LLM 分析者根据文稿构建协议模型，并把每个性质交给 ProVerif；第 3 节中的每个结果都是证明器的回答。

ProVerif+ Agent 会话 session-20261114T231523，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步骤：53，证明器 4 s，实际耗时 18 分钟，费用 0.114 美元。

ProVerif：运行 69 次证明，67 次已判定，2 次无答案，2 s，记录 30 个回答。

分析了什么

协议	ADRF data retrieval subscription
版本	19.7.0
分析范围	固定已存储的非 UE 聚合 analytics 记录；RetrievalSubscribe/Notify/Unsubscribe，省略外部数据采集。明确只选无 SUPI/GPSI 的分支，不能推广为无需用户同意。
不在范围内	符号模型不证明算法的计算安全或抗量子安全；攻击轨迹只在明确假设下成立，不直接等同规范漏洞。每个性质必须有可达性/正例控制。 固定 Rel-19。分别区分外部网络攻击者与持有自身凭据的恶意合法 NF；TLS 认证加密按端点建模，合法 NF 不能读取其他 NF 的 TLS 会话。 OAuth2 subject/audience/scope/expiry 及 TLS 身份绑定依照所附基线；服务 scope 不自动等于资源所有权。单调表不能模拟真实删除，有限状态生命周期必须明确界定。目标是指定 API 子流程，不是完整 5G 安全。 前次交付的 native Word 表格有跨条款定位错误；本次从不变的原稿重新解析并独立验证相同范围。旧模型是待复核工件，不能将旧的错配来源当作规范或直接据此宣称原结论为假。
原文	来自下列章节的 447 个编号原文单元（331824 字节）。本报告中的 s0001 之类的引用就是这些原文单元之一

分析所依据的文稿章节：

文稿	章节	标题
main	4.2.2.6	Nadrf_DataManagement_RetrievalSubscribe 服务操作
main	4.2.2.6.1	概述
main	4.2.2.6.2	请求检索和订阅数据或分析
main	4.2.2.7	Nadrf_DataManagement_RetrievalUnsubscribe 服务操作
main	4.2.2.7.1	概述
main	4.2.2.7.2	请求删除数据或分析的检索订阅
main	4.2.2.8	Nadrf_DataManagement_RetrievalNotify 服务操作
main	4.2.2.8.1	概述
main	4.2.2.8.2	关于已订阅数据或分析的通知
main	4.2.2.8.3	关于即将被删除的数据或分析的通知
main	5.1	Nadrf_DataManagement 服务 API
main	5.1.1	引言
main	5.1.2	HTTP 的使用
main	5.1.2.1	概述

文稿	章节	标题
main	5.1.2.2	HTTP 标准标头
main	5.1.2.2.1	概述
main	5.1.2.2.2	内容类型
main	5.1.2.3	HTTP 自定义标头
main	5.1.3	资源
main	5.1.3.1	概述
main	5.1.3.2	资源：ADRF 数据存储记录
main	5.1.3.2.1	描述
main	5.1.3.2.2	资源定义
main	5.1.3.2.3	资源标准方法
main	5.1.3.2.3.1	POST
main	5.1.3.2.3.2	GET
main	5.1.3.2.4	资源自定义操作
main	5.1.3.3	资源：单个 ADRF 数据存储记录
main	5.1.3.3.1	描述
main	5.1.3.3.2	资源定义
main	5.1.3.3.3	资源标准方法
main	5.1.3.3.3.1	DELETE
main	5.1.3.3.4	资源自定义操作
main	5.1.3.4	资源：ADRF 数据检索订阅
main	5.1.3.4.1	描述
main	5.1.3.4.2	资源定义
main	5.1.3.4.3	资源标准方法
main	5.1.3.4.3.1	POST
main	5.1.3.4.4	资源自定义操作
main	5.1.3.5	资源：单个 ADRF 数据检索订阅
main	5.1.3.5.1	描述
main	5.1.3.5.2	资源定义
main	5.1.3.5.3	资源标准方法
main	5.1.3.5.3.1	DELETE
main	5.1.3.5.4	资源自定义操作
main	5.1.4	无关联资源的自定义操作
main	5.1.4.1	概述
main	5.1.4.2	操作：request-storage-sub
main	5.1.4.2.1	描述
main	5.1.4.2.2	操作定义

文稿	章节	标题
main	5.1.4.3	操作：request-storage-sub-removal
main	5.1.4.3.1	描述
main	5.1.4.3.2	操作定义
main	5.1.4.4	操作：remove-stored-data-analytics
main	5.1.4.4.1	描述
main	5.1.4.4.2	操作定义
main	5.1.5	通知
main	5.1.5.1	概述
main	5.1.5.2	检索通知
main	5.1.5.2.1	描述
main	5.1.5.2.2	目标 URI
main	5.1.5.2.3	标准方法
main	5.1.5.2.3.1	POST
main	5.1.5.3	ADRF 告警通知
main	5.1.5.3.1	描述
main	5.1.5.3.2	目标 URI
main	5.1.5.3.3	标准方法
main	5.1.5.3.3.1	POST
main	5.1.6	数据模型
main	5.1.6.1	概述
main	5.1.6.2	结构化数据类型
main	5.1.6.2.1	引言
main	5.1.6.2.2	类型：NadrfDataStoreRecord
main	5.1.6.2.3	类型：NadrfDataStoreSubscription
main	5.1.6.2.4	类型：NadrfDataRetrievalSubscription
main	5.1.6.2.5	类型：NadrfDataRetrievalNotification
main	5.1.6.2.6	类型：NadrfDataStoreSubscriptionRef
main	5.1.6.2.7	类型：NadrfStoredDataSpec
main	5.1.6.2.8	类型：DataSubscription
main	5.1.6.2.10	类型：StorageHandlingInfo
main	5.1.6.2.11	类型：NadrfAlertNotification
main	5.1.6.2.12	类型：NadrfAlertNotificationResponse
main	5.1.6.2.13	类型：DataSetTag
main	5.1.6.3	简单数据类型和枚举
main	5.1.6.4	描述备选数据类型或数据类型组合的数据类型
main	5.1.7	错误处理

文稿	章节	标题
main	5.1.7.1	概述
main	5.1.7.2	协议错误
main	5.1.7.3	应用错误
main	5.1.8	特性协商
main	5.1.9	安全
main	5.1.6.2.9	

源文包中的源文步骤所定位的协议目的：固定已存储的非 UE 聚合 analytics 记录；RetrievalSubscribe/Notify/Unsubscribe，省略外部数据采集。明确只选无 SUPI/GPSI 的分支，不能推广为无需用户同意。

对照文稿定性质

在编写查询之前，分析者将草拟的性质清单与文稿自身对协议目的的表述逐一对照。名称、性质所涉及的内容以及文稿的说法均属分析者的解读；所指出的位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A03	订阅请求与已创建的订阅匹配	NF 服务消费者与 ADRF：消费者确信其发送的 notifCorrId、notificationURI 和订阅内容正是其所接受的 201 Created 中的那些，且 ADRF 确信其所接受的请求来自授权服务器为此操作授权的 NF。	消费者发送 NadrfDataRetrievalSubscription，携带 notifCorrId、notificationURI、timePeriod 以及 anaSub/dataSub/dataSetId 之一（s0009-s0016）；ADRF 创建订阅，分配 subscriptionId，并返回 201 Created，附带表示和 Location 头（s0019-s0023）。TS 33.501 5.9.2.1 要求消费者与生产者之间进行双向认证（ts-33-501:2871）。	s0003, s0009, s0015, s0019, s0023
A04	订阅与通知为单射	ADRF 抵御重复的订阅请求，消费者抵御重复的通知：一个请求创建一个订阅，一个通知只被接受一次。	ADRF 在收到 POST 时应创建新订阅并分配 subscriptionId（s0019-s0022）；消费者应存储通知并返回 204（s0057-s0059）。文稿未规定防重放规则；notifCorrId 的唯一性仅要求对给定消费者按订阅保持（s0325）。	s0019, s0020, s0021, s0057, s0059, s0325
D01	通知来自 ADRF	NF 服务消费者确信每个被接受的 Retrieval Notification 均由 ADRF 为其自身某个订阅发送，携带该订阅的 notifCorrId 且载荷未被修改。	ADRF 应调用 Nadrf_DataManagement_RetrievalNotify 并 POST NadrfDataRetrievalNotification，其中携带 notifCorrId、timeStamp 以及 anaNotifications/dataNotif/fetchInstruct 之一（s0040, s0045-s0051）；消费者存储该通知并返回 204（s0057-s0059）。完整性委托给 SBI 传输安全（ts-33-501:2871, s0266）。	s0040, s0045, s0046, s0057, s0059
A08	通知属于其自身的订阅	NF 服务消费者：被接受的通知仅归属于其创建的订阅，不归属于其他订阅，也不归属于其他消费者的订阅。	notifCorrId 由消费者提供，且对给定 NF 服务消费者应按订阅保持唯一（s0325）；通知携带 notifCorrId，据称在多个订阅共用同一回调 URI 时有用（s0328）；notificationURI 是消费者选择的通知目标地址（s0325）。RFC 6749 10.15 警告端点向请求方提供的地址发送凭据的风险（rfc6749:3154）。	s0015, s0045, s0046, s0325, s0328

ID	简称	对象	文稿怎么说	出处
Z01	操作需要有效的访问令牌	ADRF 确信其接受的每个检索订阅请求均已由授权服务器为 Nadrf_DataManagement API 以所需范围授权；消费者在调用该操作前持有此类令牌。	访问可由 OAuth2 客户端凭据授予方式授权，NRF 充当授权服务器；消费者应在使用服务前获取令牌（s0372-s0374）；范围为 "nadrf-datamanagement" 和 "nadrf-datamanagement:storage-read-delete-subs"（s0375-s0377）。RFC 6749 10.3 指出资源服务器无法检查令牌是否颁发给出示它的客户端（rfc6749:2879）。	rfc6749:2879, s0372, s0373, s0375, s0377
Z03	仅订阅所有者可移除该订阅	创建检索订阅的 NF 服务消费者：另一个 NF 服务消费者，即使持有具有检索范围的有效访问令牌，也不能移除该订阅。	对 Individual ADRF Data Retrieval Subscription 的 DELETE 仅在 URI 中携带 subscriptionId，无请求体（s0032, s0205-s0208）；文稿为该操作给出的唯一授权是 API 范围，该范围整体涵盖 RetrievalUnsubscribe 操作（s0377）；TS 29.501 4.10 指出仅服务级范围将授予所有有权访问该服务的消费者全部操作，并警告这不得授予不当的删除权限（ts-29-501:1801）。	ts-29-501:1801, s0032, s0033, s0377, s0200
S05	取消订阅后无通知	NF 服务消费者：在 ADRF 移除订阅（204 No Content）或请求终止后，不再收到该订阅的通知。	收到 DELETE 后，ADRF 应移除相应订阅并返回 204（s0033-s0035）；terminationReq 表示不再提供与此订阅相关的通知（s0328）。	s0027, s0033, s0034, s0035, s0328
R02	每条通知至多被接受一次	NF 服务消费者抵御同一 Retrieval Notification 的重复投递：同一通知实例不会被存储或处理两次。	消费者应存储通知并返回 204（s0057-s0059）；通知携带 timeStamp，但文稿未规定重放检测规则（s0046, s0047）。	s0045, s0046, s0047, s0057, s0059
K02	通知载荷保留在消费者处	被通知的数据或分析（anaNotifications、dataNotif、fetchInstruct）对除 ADRF 和已订阅的 NF 服务消费者之外的任何实体保密，包括其他合法 NF 和网络攻击者。	通知体携带分析、数据或获取指令（s0045-s0051）；文稿未提供端到端保护，依赖 SBI 传输完整性和机密性（s0266, ts-33-501:2871）。	s0045, s0049, s0050, s0051, s0062
K01	访问令牌不可推导	授权服务器的令牌：网络攻击者或其他 NF 均无法推导出其未被颁发的有效令牌。	消费者应在使用服务前从 NRF 获取令牌（s0372-s0374）；RFC 6749 10.3 要求令牌不能被未授权方生成、修改或猜测（rfc6749:2870），10.10 要求防止猜测（rfc6749:3032）。	rfc6749:2870, s0372, s0373
U01	通知与数据集保留在订阅者处	NF 服务消费者：通知的 notificationURI 和所请求的数据集属于其创建的订阅，且该通知不会被归到其他订阅。	notificationURI 在订阅中为必填项，是通知目标地址（s0325）；ADRF 向收到的 notificationURI 执行 POST（s0045）；dataSetId 选择所存储的数据或分析记录（s0325, s0014）。	s0014, s0015, s0045, s0325
U02	通知真实且订阅已终止	NF 服务消费者：被接受的通知由 ADRF 为其订阅产生，且订阅一旦移除，该订阅的通知不再被接受。	ADRF 向 notificationURI 发送通知（s0045）；消费者存储该通知（s0057）；DELETE 移除订阅（s0033-s0035）；terminationReq 终止后续通知（s0328）。	s0045, s0057, s0033, s0034, s0328
Z05	跨 NF 访问策略具有规范性依据	ADRF 与其他 NF 服务消费者：哪些 NF 可读取或删除存储在 ADRF 中的分析以及其他 NF 的订阅，以及该规则来自何处。	文稿为该 API 给出的唯一授权规则是 OAuth2 范围集合（s0372-s0377）；TS 29.501 4.10 指出仅服务级范围将授予所有有权访问该服务的消费者每一项操作，且在必须限制访问时应定义额外范围（ts-29-501:1795, ts-29-501:1801）。	ts-29-501:1795, ts-29-501:1801, s0372, s0375, s0377

分析者没有补充性质：起草的清单已覆盖文稿写明的目标。

文稿里的安全表述，逐段去向

宿主把文稿里说明协议保护什么、防什么的章节逐段列出，共 398 段；分析者对每一段说明了去向：定为性质、定为攻击者条件，或者为什么都不选。选了 42 段，没选 356 段。列哪些段落由标准的章节结构决定；每一段的去向是分析者的判断。

段落	出处	定为	不选的原因
P001 至 P002	rfc6749:2778-2786	没有陈述目标	OAuth 2.0 安全考虑的引言；它把问题转交给其他文稿，未为本 API 陈述任何目标。
P003 至 P006	rfc6749:2790-2823	Z01, K01	
P007 至 P010	rfc6749:2827-2850	Z01, `malicious_nf`	
P011 至 P015	rfc6749:2859-2881	K01, Z01, Z03	
P016 至 P021	rfc6749:2885-2917	没有陈述目标	刷新令牌；本 API 使用客户端凭据授权，所分析的流程中没有刷新令牌。
P022 至 P025	rfc6749:2921-2943	没有陈述目标	授权码；不属于本 API 使用的客户端凭据授权。
P026 至 P029	rfc6749:2947-2980	没有陈述目标	授权码流程中的重定向 URI 操纵；此处没有该流程，也没有重定向端点。
P030 至 P033	rfc6749:2984-3003	没有陈述目标	资源所有者口令授权；NRF 通过客户端凭据授权签发令牌，没有资源所有者参与。
P034 至 P035	rfc6749:3013-3019	K01	
P036	rfc6749:3023-3028	Z01	
P037 至 P039	rfc6749:3032-3041	K01	
P040 至 P042	rfc6749:3045-3066	没有陈述目标	针对最终用户口令的网络钓鱼；本 API 中没有最终用户，也没有口令。
P043 至 P048	rfc6749:3070-3110	没有陈述目标	浏览器端点上的跨站请求伪造；此处没有浏览器，也没有用户代理会话。
P049 至 P050	rfc6749:3119-3138	没有陈述目标	授权页面上的点击劫持；不属于本机器对机器 API。
P051 至 P052	rfc6749:3142-3150	证明器问不了	输入清洗和代码注入属于实现缺陷；无法向符号协议模型询问这些问题。
P053 至 P054	rfc6749:3154-3167	A08, U01, `malicious_nf`	
P055 至 P062	rfc6749:3170-3210	Z01, Z03	
P063	ts-33-501:2491-2491	不在范围内	UE 与网络之间在 AS/NAS 上的降级；所分析的 API 没有算法协商。
P064 至 P072	ts-33-501:2495-2511	不在范围内	UE 与服务网络之间的认证和授权；任务固定为非 UE、已存储分析的分支。

段落	出处	定为	不选的原因
P073 至 P074	ts-33-501:2515-2517	不在范围内	AS/NAS 保护的密钥长度和算法依赖性；不在所分析的 API 内。
P075 至 P078	ts-33-501:2523-2529	不在范围内	对 UE 在 Uu 上的要求；建模的子过程中没有 UE 参与。
P079 至 P089	ts-33-501:2533-2553	不在范围内	UE 与 gNB 之间的用户数据和信令机密性；在所分析的 API 之外。
P090 至 P103	ts-33-501:2557-2583	不在范围内	UE 与 gNB 之间的完整性保护；在所分析的 API 之外。
P104 至 P110	ts-33-501:2587-2599	不在范围内	订阅凭证存储在 UE 的防篡改硬件中；范围内没有 UE 或 USIM。
P111 至 P125	ts-33-501:2603-2631	不在范围内	UE 处的订户隐私和 SUPI/SUCI 处理；任务固定为不含 SUPI 或 GPSI 的分支。
P126	ts-33-501:2637-2637	不在范围内	关于 gNB 要求的通用陈述；范围内没有 gNB。
P127 至 P136	ts-33-501:2641-2659	不在范围内	UE 与 gNB 之间的加密；在所分析的 API 之外。
P137 至 P147	ts-33-501:2663-2683	不在范围内	UE 与 gNB 之间的完整性保护；在所分析的 API 之外。
P148 至 P156	ts-33-501:2687-2703	不在范围内	gNB 的建立与 O&M；在所分析的 API 之外。
P157 至 P158	ts-33-501:2707-2709	不在范围内	gNB 内部的密钥管理；在所分析的 API 之外。
P159 至 P162	ts-33-501:2713-2721	不在范围内	gNB 处的用户面和控制面处理；在所分析的 API 之外。
P163 至 P168	ts-33-501:2725-2735	不在范围内	gNB 的安全环境；在所分析的 API 之外。
P169 至 P176	ts-33-501:2739-2755	不在范围内	gNB 的 F1 和 E1 接口；在所分析的 API 之外。
P177 至 P178	ts-33-501:2759-2761	不在范围内	对 ng-eNB 的要求；在所分析的 API 之外。
P179 至 P196	ts-33-501:2767-2805	不在范围内	ng-eNB 与 5GC 之间在 NG-RAN 接口上的信令机密性、完整性和订户隐私，并非此处所分析的 SBI API。
P197 至 P198	ts-33-501:2809-2811	不在范围内	对 SEAF 的要求；不是本 API 的参与方。
P199 至 P207	ts-33-501:2819-2837	不在范围内	对网络的通用订户隐私和 SUPI 保护要求；任务固定为不含 SUPI 或 GPSI 的分支。
P208 至 P213	ts-33-501:2841-2853	不在范围内	对 AUSF 和 UDR 在认证数据方面的要求；不属于所分析的 API。
P214	ts-33-501:2859-2859	没有陈述目标	关于运营商信任区的范围陈述；它未陈述任何目标，除了网络及其自身 NF 凭证之外没有给攻击者任何东西。
P215 至 P220	ts-33-501:2865-2875	A03, D01, Z01, A04, `malicious_nf`	
P221 至 P227	ts-33-501:2879-2891	Z01, K01, `malicious_nf`	
P228 至 P234	ts-33-501:2895-2907	不在范围内	NEF 对应用功能的暴露；所分析的子过程中没有 AF，也没有 NEF。

段落	出处	定为	不选的原因
P235 至 P241	ts-33-501:2911-2923	不在范围内	服务通信代理的行为；所分析的过程是两个 NF 之间的直接通信。
P242 至 P254	ts-33-501:2929-2953	不在范围内	漫游的端到端互连安全；所分析的过程停留在单个运营商网络内，没有 SEPP。
P255 至 P303	ts-33-501:2957-3055	不在范围内	对 N32 的 SEPP 和漫游中间节点要求；所分析的过程中没有 SEPP，也没有漫游参与。
P304 至 P312	ts-33-501:3059-3075	不在范围内	SEPP 之间在 N32-f 上的属性保护；所分析的过程不跨越该接口。
P313 至 P314	ts-33-501:3079-3081	不在范围内	IPUPS 处对 GTP-U 的包过滤；不属于所分析的 API。
P315 至 P318	ts-33-501:3085-3091	不在范围内	漫游信令中的始发网络标识符；所分析的过程中没有漫游。
P319 至 P327	ts-33-501:3097-3115	不在范围内	信令流量监控要求；监控功能不属于所分析的过程。
P328 至 P337	ts-33-501:3121-3141	不在范围内	面向 UE 应用的 AS/NAS 安全可见性和可配置性；范围内没有 UE。
P338 至 P357	ts-33-501:3149-3189	证明器问不了	NAS/AS 加密和完整性的算法标识符；原语的强度不是符号证明器能判定的问题，而且这些算法并不保护所分析的 API。
P358 至 P369	ts-33-501:3193-3215	证明器问不了	UE 与网络之间的算法选择和降级；符号模型无法判定原语强度，且所分析的 API 自身没有算法协商。
P370 至 P375	ts-33-501:3219-3231	不在范围内	对 5G-RG 和 NSSAAF 的要求；二者都不是所分析 API 的参与方。
P376 至 P380	ts-33-501:6540-6548	不在范围内	N3 接口上的安全；在所分析的 API 之外。
P381 至 P385	ts-33-501:9656-9666	不在范围内	带订阅标识符隐私和不带的 EAP-TLS；不属于所分析的 API。
P386 至 P394	ts-33-501:10893-10917	不在范围内	IAB 附录要求；在所分析的 API 之外。
P395 至 P396	ts-33-501:11439-11443	不在范围内	MBSF 和 MBSTF 要求；在所分析的 API 之外。
P397 至 P398	ts-33-501:11899-11901	不在范围内	NEF 安全要求；所分析的过程中没有 NEF。

参与方和受保护的值，逐项是否设为攻击者条件

宿主把起草的性质里点到的参与方和受保护的值逐项列出，共 21 项；分析者对每一项说明了哪个攻击者条件控制这个参与方或持有这个值，或者为什么不设。设了攻击者条件的 5 项，没设的 16 项。清单来自性质本身；每一项怎么处理是分析者的判断。

ID	参与方或值	类别	攻击者条件	不设的原因
S01	NF 服务消费者	参与方	`malicious_nf`	
S02	ADRF	参与方	按定义全部失守	ADRF 是创建、通知和删除订阅的一方；一旦它被控制，这里所问的每条性质都按定义丧失，并且文稿在不诚实的 ADRF 下对这些性质一概不作要求（TS 33.501 5.9.1 假定同一运营商的 NF 处于同一信任区，ts-33-501:2859）。

ID	参与方或值	类别	攻击者条件	不设的原因
S03	NRF	参与方	按定义全部失守	NRF 是授权服务器；如果攻击者控制了它，它就会为任意主体签发 token，token 性质便按定义丧失。
S04	notifCorrId	值	`malicious_nf`	
S05	notificationURI	值	攻击者得不到更多	notificationURI 是 ADRF 必须向其投递的回调地址；它是一个可达的网络地址，并非秘密，所以攻击者知道它也只不过获得了向该地址发送消息的能力。
S06	subscriptionId	值	`malicious_nf`	
S07	订阅内容	值	攻击者得不到更多	所请求的分析订阅、数据集和时间窗口不受任何目标保护；受保护的只有被通知的载荷（K02）。
S08	通知实例	值	`malicious_nf`	
S09	通知载荷	值	按定义全部失守	载荷正是 K02 所保护的值；攻击者持有它就意味着 K02 按定义丧失，因此它不能成为此处所问性质的攻击者。
S10	timeStamp	值	攻击者得不到更多	通知的时间戳属于 K02 所覆盖的载荷的一部分；没有目标要求它单独保密。
S11	OAuth2 token	值	`malicious_nf`	
S12	scope	值	攻击者得不到更多	scope 名称是公开的，并随 token 携带；知道它并不能赋予攻击者任何其尚未拥有的能力。
S13	受众	值	攻击者得不到更多	token 的受众是 ADRF 的公开身份。
S14	token 受众	值	攻击者得不到更多	token 的受众是 ADRF 的公开身份。
S15	订阅状态	值	攻击者得不到更多	读取订阅状态并不能让攻击者获得超出其可调用操作之外的东西；S05 问的是通知，而非状态的保密性。
S16	anaNotifications	值	按定义全部失守	分析通知是 K02 所保护的载荷；攻击者持有它们就意味着该性质按定义丧失。
S17	dataNotif	值	按定义全部失守	数据通知是 K02 所保护的载荷；攻击者持有它就意味着该性质按定义丧失。
S18	fetchInstruct 取回的数据/分析	值	不在范围内	用 fetch 指令取回数据使用的是 RetrievalRequest 操作，任务已将其排除在所分析的子流程之外。
S19	dataStoreId	值	攻击者得不到更多	数据集标识符为已存储的记录命名，但在本 API 中不授予任何访问权；载荷保护是 K02。
S20	订阅	值	按定义全部失守	攻击者持有的订阅正是 A03 和 Z03 所问性质的订阅；已无剩余可问。
S21	数据/分析记录	值	不在范围内	已存储的分析记录本身是通过 RetrievalRequest 操作读取的，在所分析的子流程之外。

每条性质在各攻击者条件下的答案

攻击者由分析者设定，且在编写查询之前就已设定；每个答案都在其中一个攻击者下记录。单元格是证明器在该攻击者下对该性质给出的答案；“无答案”是分析者为该性质设定的一个攻击者，在该攻击者下证明器未对某项主张作出判定或没有记录任何内容；空单元格是分析者未为该性质设定的攻击者。

攻击者	持有什么	依据	出处
`network`	仅网络：它读取、丢弃、修改和发送消息， 不持有任何一方的秘密		

攻击者	持有什么	依据	出处
`malicious_nf`	一个合法 NF 服务消费者的长期凭据（其 NF 身份及其自有的、带有 scope “nadrf-datamanagement:storage-read-delete-subs” 的 OAuth2 访问 token），自运行开始即持有；它可以用该 token 调用每一个 Nadrf_DataManagement 服务操作，并选择其创建的订阅的 notifCorrId、数据集和 notificationURI。它还知道另一个 NF 的回调地址（一个可达的网络地址）以及该 NF 使用的 notifCorrId 和 subscriptionId，因为这些都是文稿要求唯一但从不要求不可猜测的字符串。	文稿允许任何持有该 API scope 的 NF 服务消费者调用 RetrievalSubscribe、RetrievalUnsubscribe 和 RetrievalNotify 操作（s0372-s0377）；TS 29.501 4.10 规定，仅服务级 scope 将 API 的所有操作授予所有被允许访问该服务的消费者（ts-29-501:1801）。RFC 6749 10.2 规定，持有自身凭据的恶意客户端可以行动，而 TS 33.501 5.9.2.1 对哪个消费者可以访问哪个资源不作任何限制（ts-33-501:2871）。关联标识符和订阅标识符是文稿最多要求唯一、从不要求不可猜测的字符串：notifCorrId“对于给定的 NF 服务消费者，每个订阅应唯一”（s0325），而 subscriptionId 仅“标识一个订阅”（s0200）。	ts-29-501:1801, ts-33-501:2871, s0325, s0200, s0372, s0375, s0377

ID	`network`	`malicious_nf`
A03	成立	成立
A04	成立	成立
D01	成立	不成立
A08	成立	不成立
Z01	成立	成立
Z03	成立	不成立
S05	无答案	
R02	无答案	无答案
K02	成立	成立
K01	成立	成立
U01	成立	不成立
U02	成立	成立
Z05	成立	不成立

原文和目标

Source: 447 units, 331824 bytes, frozen in source-delivery-01.json. Targets: 12 approved (A03, A04, D01, A08, Z01, Z03, S05, R02, K02, K01, U01, U02); the questions asked: 3. 目标清单事先按草稿获得批准，采用 --approve-targets。

宿主按其文档化默认值（任务 `decisions: auto`）所作的决定，供你核查：

- approve_package: 批准源文包 protocol-beba40bbab10b6aa5ff4a5b20e6f

本步骤的自报情况：1 次请求（1 次失败），0.000 USD，输入 152803 tokens、输出 5151 tokens，202 秒。

关于证明器和模型的说明

分析者为目标命名的内容：

step	targets	features	reason
40	A03, A04, D01, A08, Z01, Z03, U01, U02, S05, R02, K02, K01, Z05	plain_primitives, many_sessions, other	该模型只使用签名、公钥加密和不含哈希的元组，除此之外没有任何代数定律；它覆盖任意数量的会话，因为 ADRF、NRF 和通知处理器都被复制。传输被表示为发送方的签名加上对接收方的加密，而非 TLS 会话，且 subscriptionId 空间被简化为单一值，因此 A04 的答案建立在一个简化的标识符空间之上。订阅生命周期（移除后不再发送通知，即 S05 和 U02 的后半部分）用一个单调表加一个移除标记来表示，而非真正的状态机，因此该处的顺序性主张无法由本模型确定：那需要一个有界状态模型（Tamarin+ Agent）。

分析者的结语

正常流程（订阅 → 返回 201 及 subscriptionId → 通知 → 204 → DELETE → 204）最先被建模并全程保留；最终模型加入了 NRF token、一个其凭据被攻击者持有的第二个 NF，以及一张订阅表。十三个目标中有十二个在其攻击者下有了答案，其中大多数是证明：消费者的 201 与其发送的值相匹配，ADRF 只接受经 NRF 授权的 NF，每一条被接受的通知都是真实的且未被修改，通知载荷不会到达任何其他 NF，且没有任何操作会在 NRF 未签发的 token 下被接受。

通过轨迹确立了两个弱点。第一，通知只携带一个须对每个消费者唯一的 notifCorrId，而 ADRF 将它投递到订阅者指定的 notificationURI，因此一个 NF 只要指定了另一个 NF 的回调地址及其关联值，其通知就会在那里被接受，并归于受害者的订阅、涉及受害者从未请求过的数据集（D01、A08、U01）。第二，DELETE 除了 subscriptionId 之外不携带任何内容，且没有文稿要求 ADRF 将请求者与创建者进行比较，因此任何持有检索 scope 的 NF，只要能够说出该标识符，就能移除另一个 NF 的订阅（Z03、Z05）。

仍悬而未决的：R02 未从 ProVerif 得到答案（无定论）；S05 和 U02 的顺序性一半依赖于一个单调移除标记，A04 的单射性依赖于单值 subscriptionId；传输是对双向 TLS 的签名加密理想化。生命周期和标识符空间的问题需要有界状态模型和 Tamarin+ Agent。两项建议（对 DELETE 进行所有权检查、在通知中携带 subscriptionId）均以未核查状态给出。