# ADRF 分析记录生命周期安全分析

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** | 11 条：6 条成立，5 条不成立（8 个攻击） |
| **发现** | 1 |

**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

ADRF 的 Nadrf_DataManagement API 允许 NF 服务消费者存储分析数据、检索数据和删除数据。参与方有三类：存储记录的 NF、读取记录的 NF，以及保存记录的 ADRF；NRF 是授权服务器，为每个 NF 签发该 API 范围的 OAuth2 访问令牌。

一次交换受到端到端保护：每个请求都用 NF 自己的凭证认证，并携带其访问令牌；ADRF 在请求方 NF 自己的连接上应答。存储方 NF 获知它所创建记录的 storeTransId；读取方用数据集标识符或该标识符来指名记录；删除操作则用 storeTransId 或数据规范来指名记录。

各方：

- **NF 服务消费者（存储方）**：存储分析数据，之后将其删除。持有：其 TLS 凭证和访问令牌。
- **NF 服务消费者（读取方）**：获批读取该记录。持有：其 TLS 凭证和访问令牌。
- **ADRF**：保存数据存储中的记录。持有：带 storeTransId 和 dataSetId 的记录存储。
- **NRF**：授权服务器。持有：NF 配置、客户端凭证。

图 1 展示了文稿所规定的交换流程。

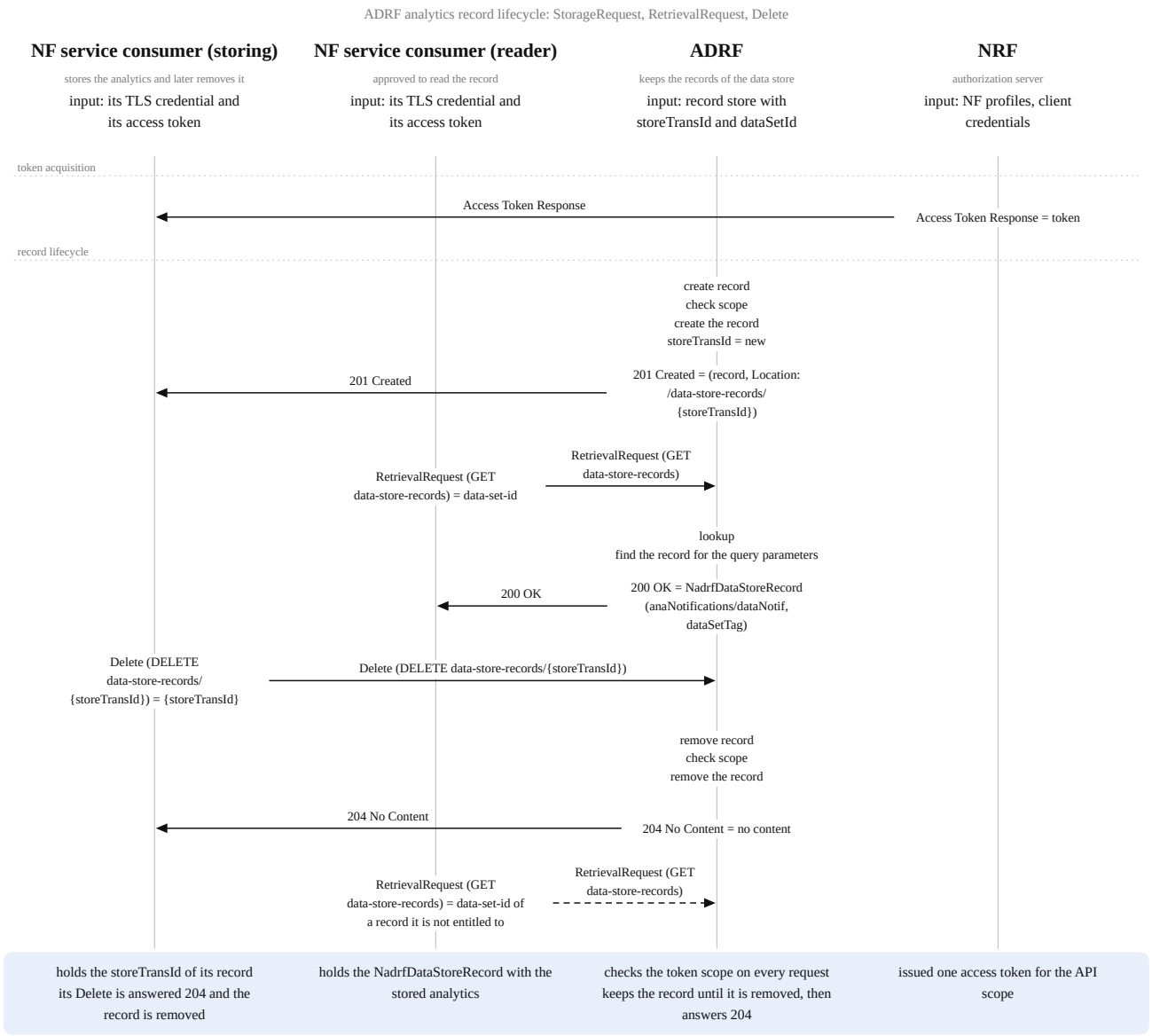



图 1. 规范规定的协议。

分析范围见附录 D：分析了哪些内容、遗漏了哪些内容，以及所依据的文稿章节。

## 2 发现

分析有 1 个发现：下文所述内容并未写在文稿中。对于其中 1 个发现，文稿是否写明尚不明确。

令牌门控和线路保护成立：只有持有自身凭证和自身令牌的 NF 才会得到服务或被执行操作，消费者接受的记录始终是 ADRF 存储的记录。协议不能把一个 NF 的记录与另一个合法 NF 隔离开来，这就是该发现：在单一 API 范围加数据集标识符的情况下，恶意 NF 可以读取另一个 NF 的分析数据并删除其记录。共回答了 11 个目标，其中三个是在展开记录状态的模型上回答的；授权目标中有一条主张（令牌未过期）无法向证明器提出。对协议的一项已检查的拟议修改——ADRF 只对存储该记录的 NF 或该 NF 批准的 NF 操作该记录——消除了另一个 NF 的三个攻击，而正常流程仍然成立。这些答案依赖于将 TLS 和 OAuth2 抽象为按端点的密钥和不可伪造的令牌，以及将记录生命周期展开到五步。

攻击图使用图 1 的列。红色表示攻击者的行为。黄色标记的是：某一方发送了规范交换流程在该点本不应有的消息，以及攻击所破坏的结果。

## 2.1 合法 NF 可以读取并抹除另一个 NF 的分析记录

| | |
|---|---|
| **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** | 高 |
| **涉及的性质** | Z05, K02, Z03, U01（见第 3 节） |
| **原文** | s0034, s0069, s0373 |
| **依据** | ProVerif 攻击查询 `z05_read`、`z05_rmspec`、`z05_del`；图 2、3、4 和 5 |

**发生了什么。** 第 5.1.9 条的范围把存储、检索、删除和订阅操作纳入同一个令牌，而 ADRF 只检查该范围。RetrievalRequest 可以用数据集标识符指名记录，remove-stored-data-analytics 会删除匹配某一数据规范的所有记录，因此一个合法 NF 只要指名另一个 NF 的数据集，就能获得该记录并将其抹除。模型同时展示了这两点：恶意 NF 收到存储方 NF 的载荷并删除该记录；一旦它获知标识符，通过按 storeTransId 删除也同样成立。

**攻击者需要什么。** 一个运营商认为是合法的 NF 凭证、NRF 为 API 范围 nadrf-datamanagement:storage-read-delete-subs 签发的访问令牌，以及该记录的数据集标识符或其 storeTransId 之一。

**攻击者能得到什么。** 恶意 NF 读取其他 NF 的分析数据，并删除另一个 NF 存储的记录，从而使存储方 NF 失去其数据以及它所请求的删除通知。不需要另一个 NF 的任何密钥，也没有破坏任何 TLS 会话。

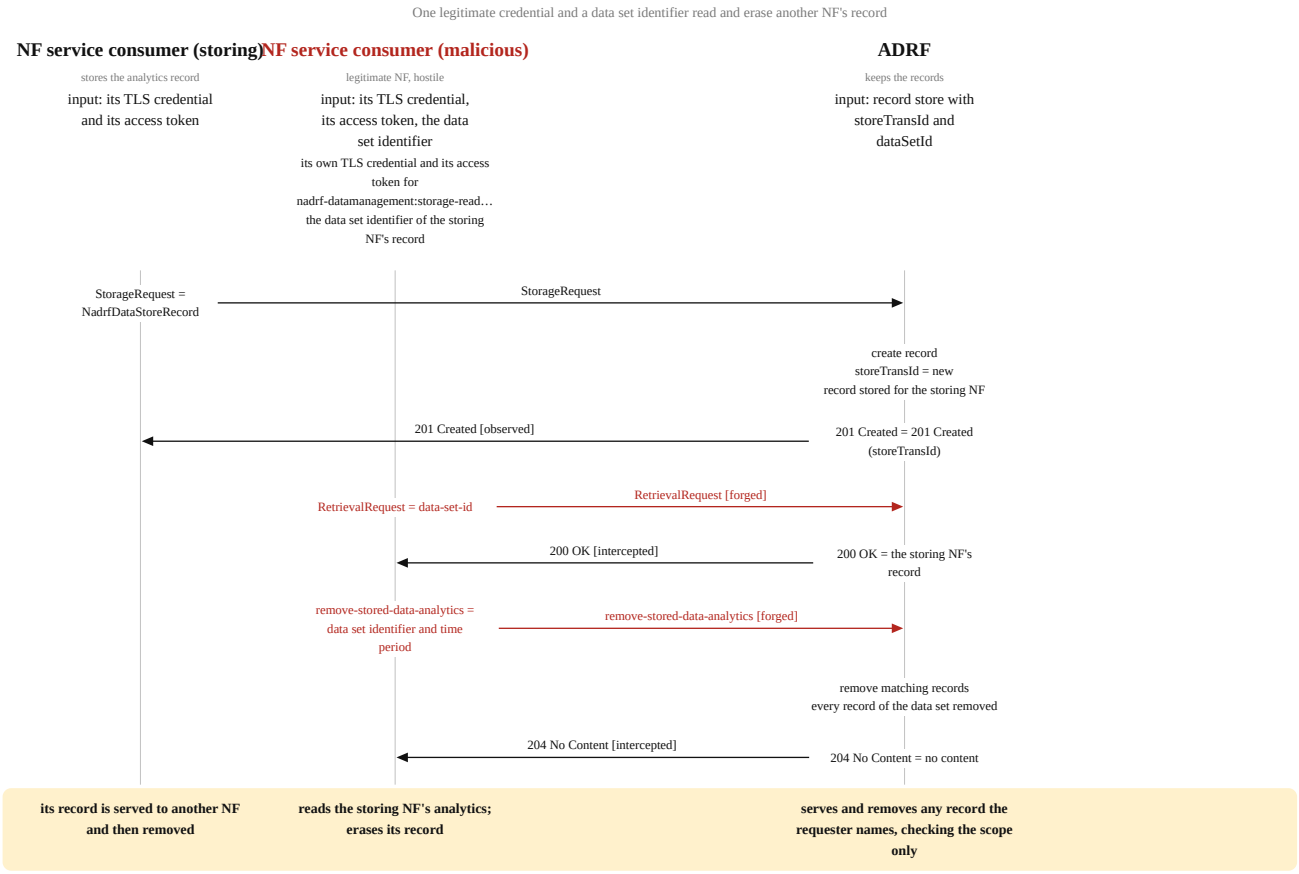



图 2. 一个合法凭证加一个数据集标识符即可读取并抹除另一个 NF 的记录。性质 Z05；攻击查询 z05_read、z05_rmspec。

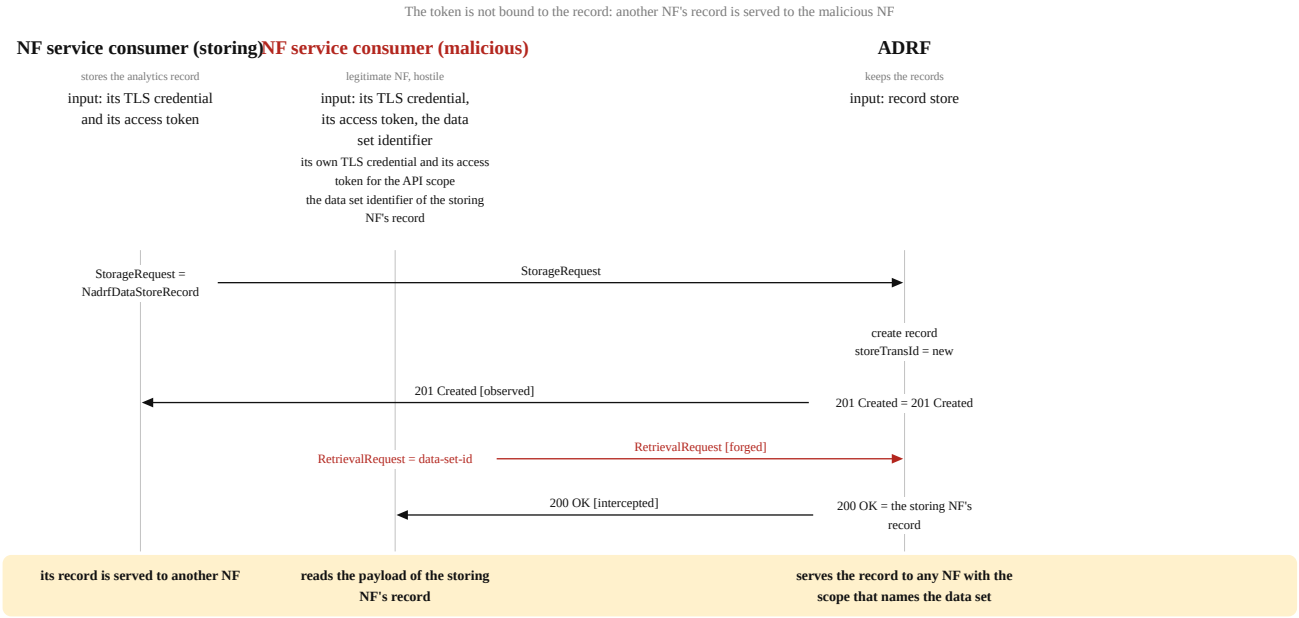



图 3. 令牌未绑定到记录：另一个 NF 的记录被提供给恶意 NF。性质 Z03；攻击查询 z05_read。

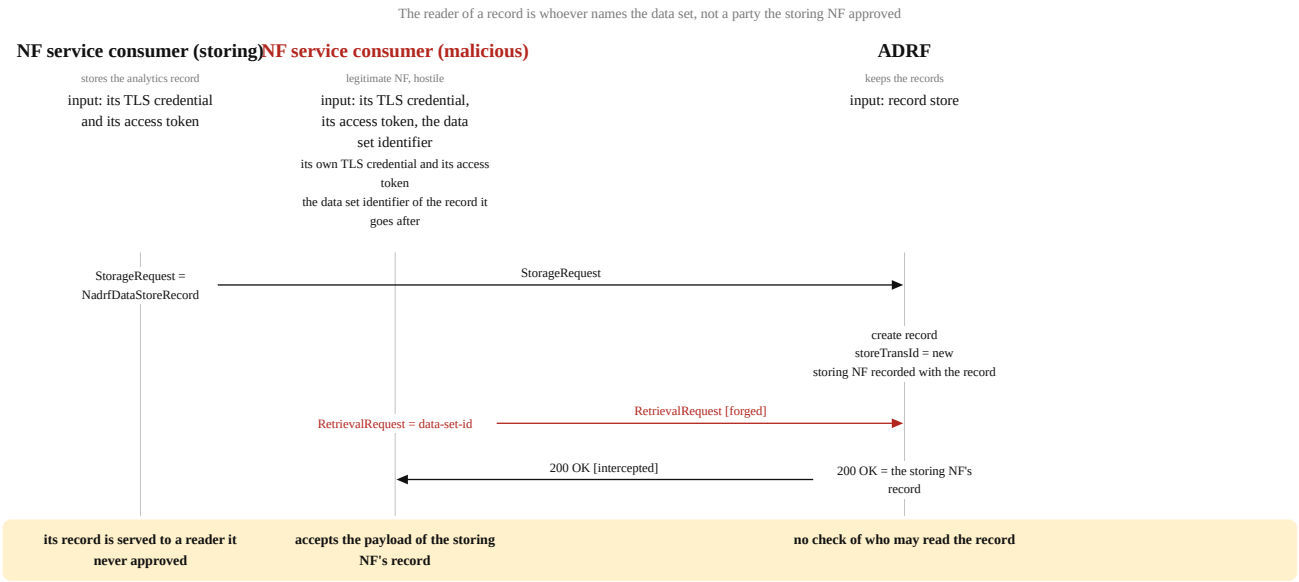



图 4. 记录的读取方是指名该数据集的一方，而不是存储方 NF 所批准的方。性质 U01；攻击查询 z05_read。

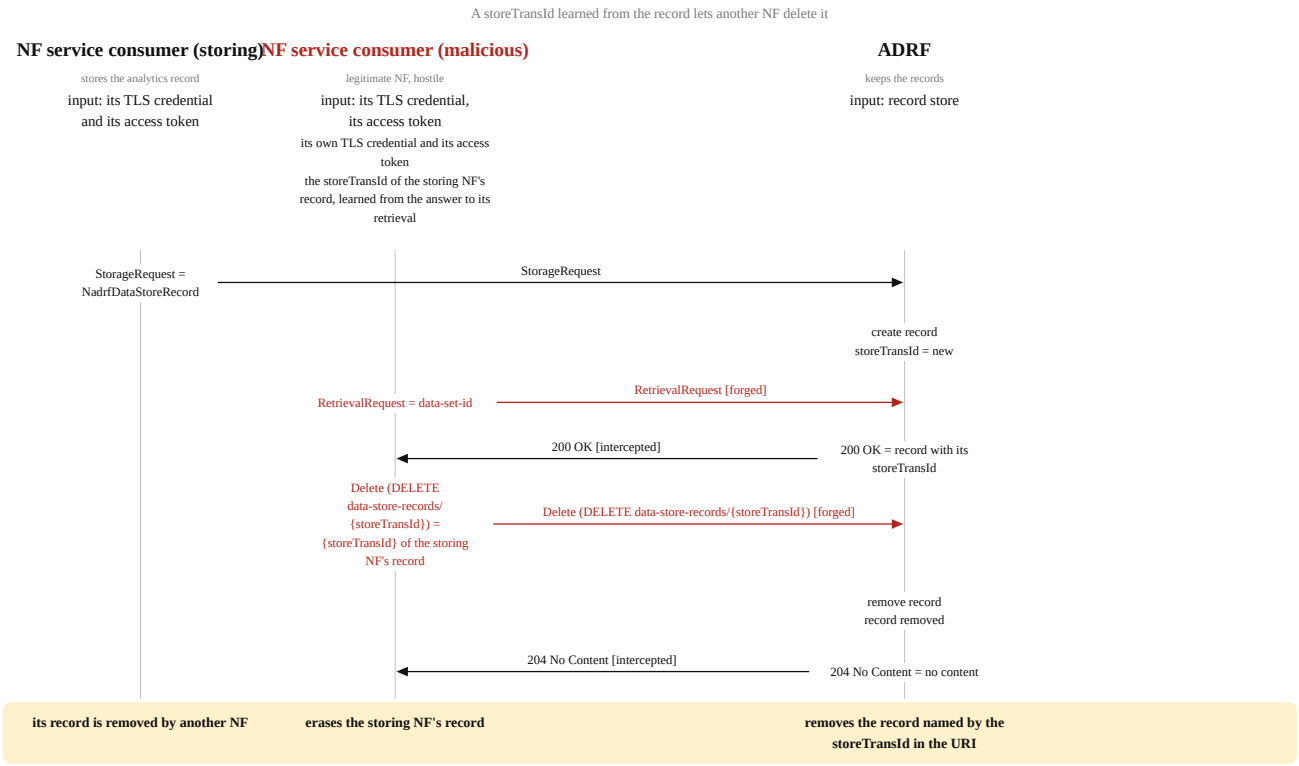



图 5. 从记录中获知的 storeTransId 使另一个 NF 可以删除该记录。性质 Z05；攻击查询 z05_del。

**拟议修改。** 将存储方 NF 的身份与记录保存在一起，并且只对该 NF 或存储方 NF 在存储请求中指名的 NF 操作该记录：检索只向这些方提供，remove-stored-data-analytics 只删除请求方 NF 存储的记录，否则返回错误。该修改已经过检查：在修改后的模型上，另一个 NF 的三个攻击不再存在，正常流程仍可找到，载荷攻击也随之消失。将单一范围拆分为存储、读取和删除范围是第二项修改，模型未对其进行检查。

**用 ProVerif 复核。** 该修改被放入模型的一个副本中：ADRF 保存存储该记录的 NF 的身份（它已经知道该身份，见第 4.2.2.2.2 条），并且在检索时只向存储方 NF 或存储方 NF 指名为允许读取的 NF 提供服务；对于按 storeTransId 删除和 remove-stored-data-analytics，它只删除请求方 NF 存储的记录，否则返回错误。在修改后的模型上，ProVerif 证明了 `z05_read`、`z05_rmspec`、`z05_del`：该攻击在那里不再可能。协议的正常流程仍可找到。在其余已记录的答案中，13 条仍然成立，1 条发生变化：`k02_nf`（K02，原为攻击）现在已验证。这是对模型的首次检查：它表明该修改在那里消除了此攻击，而不是表明它不会引入其他问题。修改后的模型：`records/fixes/0047-fix-ownership.pv`。

这些属于某一发现的攻击没有给出图：K02: z05_del、K02: z05_read、K02: z05_rmspec、U01: z05_del、U01: z05_rmspec、Z03: z05_del、Z03: z05_rmspec。

## 3 检查过的性质

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

| ID | 性质 | 假定的泄露 | 结果 | 价值 | 文稿是否已写 |
|---|---|---|---|---|---|
| Z01 | **令牌门控的检索和删除。** 每次 ADRF 接受并执行 RetrievalRequest 或 Delete（含按规范删除）时，请求方必须持有覆盖该操作的、由授权服务器签发给该请求方且未过期的许可。 | 无 | 成立 | 中 | 是 |

| ID | 性质 | 假定的泄露 | 结果 | 价值 | 文稿是否已写 |
|---|---|---|---|---|---|
| Z03 | **令牌绑定到主体和记录。** ADRF 执行读取或删除时，所用许可的主体、受众、资源（storeTransId）与操作类型必须与本次请求一致。 | 无 | 不成立（2 个攻击） | 高 | 不明确 |
| K02 | **载荷对其他NF隐藏。** 读取响应中的记录载荷只对存储该记录的 NF 与获批读取的 NF 可见，恶意合法 NF 与外部攻击者不能推导出该载荷。 | 无 | 不成立（1 个攻击） | 高 | 不明确 |
| D01 | **提供的记录与存储的记录一致。** 每次读取接受的数据对应 ADRF 真实存储的、由请求参数定位的记录，载荷与上下文一致。 | 无 | 成立 | 中 | 是 |
| S05 | **无法检索已删除的记录。** 删除生效点之后，对已删除记录的读取不得返回成功内容（应返回 204 或错误），旧上下文不能重新获得原数据。 | 无 | 在界内成立（变化值展开到 5 步） | 中 | 是 |
| R02 | **删除最多被接受一次。** 同一逻辑删除消息至多被 ADRF 接受一次，重放不改变其他记录的状态。 | 无 | 在界内成立（变化值展开到 5 步） | 中 | 不明确 |
| Z05 | **其他NF的记录保持受保护。** 恶意合法 NF 控制自身域时，不能读取或删除其他 NF 存储的记录，目标域的保密与授权性质仍成立。 | 无 | 不成立（3 个攻击） | 高 | 不明确 |
| U01 | **记录和读取者均真实。** 读取返回的记录由声称存储方真实存入，且请求方为获批读取方，storeTransId 与内容一致。 | 无 | 不成立（1 个攻击） | 高 | 不明确 |
| U02 | **读取和删除权限分离。** 读取与删除各自要求覆盖该操作的许可，二者不可互相替代。 | 无 | 攻击成立（文稿已写明） | 中 | 是 |
| U03 | **删除后无过期成功。** 删除生效或错误之后，旧记录不得作为新鲜成功结果被接受。 | 无 | 在界内成立（变化值展开到 5 步） | 中 | 是 |
| X01 | **令牌和凭证机密性。** 一个 NF 的 TLS 凭证和 NRF 签发给它的访问令牌只被该 NF、NRF 和 ADRF 知晓；网络攻击者和另一个合法 NF 都不能获得或伪造另一个 NF 身份的令牌，因此不能以该 NF 的身份对 ADRF 行事。 | 无 | 成立 | 中 | 是 |

“攻击成立（文稿已写明）”：证明器找到了这个攻击，而文稿自己已经写明它存在。这条性质不成立；这是已知结果，不算发现（附录 B）。

文稿写明而起草的清单里没有、由分析者补充的目标（见附录 D）：X01。

所提出的问题及分析者的回答：

| 问题 | 回答 | 依据 |
|---|---|---|
| Q1: dataStoreId/storer/allowed consumers 与数据绑定 | 读取方接受的 storeTransId 和载荷始终是 ADRF 存储的那一份，ADRF 报告的存储该记录的 NF 也始终是实际存储的那个 NF；但没有任何机制把记录绑定到被允许读取它的 NF，因此读取方就是任何指定该数据集的 NF，而不是存储该记录的 NF 所批准的一方。 | D01, U01, K02 |
| Q2: 读取许可和删除许可分别依据规范条件 | API 为存储、检索和删除只定义了一个范围，因此读取和删除并未分别授权：一个允许 NF 读取的令牌同时也允许它删除，模型显示同一个令牌被两者接受。 | U02, Z01, Z03 |
| Q3: 错误/删除后数据不可作为新鲜成功结果接受 | ADRF 删除记录后，对下一次检索返回 204，载荷不再返回；重复删除会被拒绝，而不是被执行两次；模型未覆盖的是 ADRF 的旧应答在新连接内被重放。 | S05, R02, U03 |

# 4 本次分析的局限

这里的结果是针对协议模型的结果：它说明规范定义的消息和检查给出了什么、未能给出什么，其中密码学被视为理想，不涉及任何实现。

分析者指出，本次分析不覆盖以下内容：

- TLS 和 OAuth2 被抽象化：传输层是每个端点各自的对称密钥，令牌是 NRF 从 NF 身份推导出的不可伪造值，因此对 TLS 的强度、令牌格式或任何算法都没有结论。
- 未对令牌过期建模：ProVerif 没有时间概念，因此目标中“许可未过期”的要求没有得到回答，模型中的令牌也永远不会过期。
- ADRF 自身记录状态的身份仅在 bounded.pv 中针对一条记录的五步展开建模；更长的生命周期、记录在生命周期到期时被删除、向存储 NF 发送删除通知，均不在分析范围内。
- model.pv 将记录保存在一张只增不减的表中，因此无法表达删除；关于删除之后行为的回答来自 bounded.pv，且只对那里展开的五步成立。
- 建模了存储消费者、读取方和一个恶意 NF；被攻陷的 ADRF 或 NRF、不诚实的 SCP 或漫游伙伴不在分析范围内。
- 未对 TLS 连接内已存储响应的重放建模：响应在端点密钥下绑定到每请求的 nonce，消费者如何处理迟到的应答只由展开的模型覆盖。
- 文稿条款 4.2.2.9.3 中按数据规范删除仅按数据集标识符匹配建模；时间段以及数据或分析规范未被表示。
- 文稿本身指出令牌不绑定到记录（RFC 6749 第 10.3 节），并为 API 给出单一范围（条款 5.1.9）；本发现超出了这些陈述，展示了拥有该范围的 NF 能做什么。

宿主记录的内容：

- S05、R02、U03 的回答依赖模型简化的一项操作（附录 D）。

每条性质仍有哪些未决问题、以及什么能解决它们，见附录 A。

# 附录 A：逐条性质的解读

以下是分析者本人对证明器回答的解读，而非证明器的回答：这些回答对协议意味着什么、解读依据是什么、能否按原样使用、以及它的价值如何。

### Z01: 令牌门控的检索和删除

每次 ADRF 接受并执行 RetrievalRequest 或 Delete（含按规范删除）时，请求方必须持有覆盖该操作的、由授权服务器签发给该请求方且未过期的许可。

只有用自身凭据认证并出示 NRF 签发给它的令牌的 NF 才会被服务或被执行，检索和两种形式的删除均如此。恶意 NF 会被服务，但仅以其自身身份和自身令牌；它所做的任何事都无法让 ADRF 代表另一个 NF 行动。令牌是否仍然有效不在模型能说明的范围内。

- 结果：成立；是否回答了所问：部分；可在有保留的情况下使用
- 价值 中：它确认了规范要求的 OAuth2 门控，并给出了 ADRF 行动的确切条件，但对过期未作任何说明，而目标也提出了这一点。
- 文稿中是否有陈述：是。文稿中的位置：s0368, s0369, s0373
- 依据：令牌是 NRF 从 NF 身份推导出的不可伪造值，请求由 NF 自身的端点密钥认证；令牌格式、受众检查和过期未被建模。
- 未决：请求时令牌是否未过期：模型没有时间，因此过期主张未得到回答。

### Z03: 令牌绑定到主体和记录

ADRF 执行读取或删除时，所用许可的主体、受众、资源（storeTransId）与操作类型必须与本次请求一致。

令牌绑定到出示它的 NF：NF 只能以自身身份行动，NRF 将每个令牌签发给一个 NF。令牌既不绑定到请求所指的记录，也不绑定到操作：任何拥有该范围的 NF 都可以指定另一个 NF 的记录并对其操作，同一个令牌既可用于读取也可用于删除。

- 结果：不成立（2 个攻击）；是否回答了所问：是；可在有保留的情况下使用
- 价值 高：主体绑定得到了证明，而绑定到记录和操作的失败正是该发现的机制；这是目标中规范从未陈述的部分。
- 文稿中是否有陈述：不明确。所依据的原文：rfc6749:2879, s0034, s0373
- 依据：模型没有给 ADRF 任何按记录的授权，因为规范没有定义这种授权；如果部署在 API 之外添加了这种授权，回答会改变。；恶意 NF 知道记录的数据集标识符；规范将其视为描述性标识符，而非秘密。
- 未决：部署是否期望 ADRF 在规范未作说明的情况下将令牌绑定到记录。

### K02: 载荷对其他NF隐藏

读取响应中的记录载荷只对存储该记录的 NF 与获批读取的 NF 可见，恶意合法 NF 与外部攻击者不能推导出该载荷。

存储的分析数据对网络攻击者保密，网络攻击者没有 NF 凭据，无法读取交换内容。这些数据对另一个合法 NF 并不保密：任何指定该数据集标识符的 NF 都会被提供载荷并读取它。

- 结果：不成立（1 个攻击）；是否回答了所问：是；可使用

- 价值 高：它区分了协议确实保护的内容（传输线路）与未保护的内容（一个合法 NF 对另一个合法 NF），这正是该发现。
- 文稿中是否有陈述：不明确。所依据的原文：s0037, s0373
- 依据：载荷在模型中是私有值，交换由端点密钥保护，因此网络攻击者只能通过合法凭据获取它。；成功的检索路径已建模；读取方对 204 应答的处理仅在展开的模型中覆盖。
- 未决：部署中数据集标识符是否为秘密；模型假设恶意 NF 知道它。

### D01: 提供的记录与存储的记录一致

每次读取接受的数据对应 ADRF 真实存储的、由请求参数定位的记录，载荷与上下文一致。

消费者接受的始终是 ADRF 真实存储的记录，具有相同的 storeTransId、相同的存储 NF 和相同的载荷。恶意 NF 可以在另一个 NF 使用的数据集标识符下存储记录，因此请求该数据集的读取方可能得到恶意 NF 的记录——对应关系成立，但并非与读取方所期望的 NF 之间成立。

- 结果：成立；是否回答了所问：是；可使用
- 价值 中：存储内容与服务内容之间干净的对应关系；其对照查询显示了局限，即存储者不必是读取方所期望的 NF。
- 文稿中是否有陈述：是。文稿中的位置：s0019, s0036, s0037
- 依据：记录完整返回给读取方，包括存储 NF 身份，模型将其放入记录中。；模型的表只增不减，因此这里对删除后的记录没有任何说明。

### S05: 无法检索已删除的记录

删除生效点之后，对已删除记录的读取不得返回成功内容（应返回 204 或错误），旧上下文不能重新获得原数据。

一旦 ADRF 删除了记录，之后对该记录的检索会返回 204，不再向读取方或任何其他人返回载荷。该回答依赖于 ADRF 在删除记录时替换其保存的状态，并在应答前检查该状态。

- 结果：在界内成立（变化值展开至 5 步）；是否回答了所问：是；可在有保留的情况下使用
- 价值 中：它表明删除确实在 API 的应答中生效，并有对照查询确认删除后的检索会被到达；保留之处是五步的界。
- 文稿中是否有陈述：是。文稿中的位置：s0045, s0054, s0055
- 依据：生命周期针对一条记录的五步展开；更长的生命周期和按过期删除不在其中。

### R02: 删除最多被接受一次

同一逻辑删除消息至多被 ADRF 接受一次，重放不改变其他记录的状态。

到达两次的删除消息只删除记录一次；第二次会被拒绝，因为 ADRF 的状态已变为已删除。被拒绝的删除保持状态不变，因此失败不会让记录看起来已被删除。

- 结果：在某个界内成立（变化值展开到 5 步）；所问是否得到回答：是；可带保留条件使用
- 价值中：规范没有说明重复删除会怎样；模型给出了一种解读，即重复删除是无害的，并用一个对照查询表明拒绝确实会被触发。
- 文稿中的表述：不明确。所依据的原文：ts-33-501:2875, s0053, s0054, s0055, s0057
- 依据：同一条记录的五个展开步骤；错误应答被建模为一种独立的回复，而不是第 5.1.7 条中的 HTTP 状态码。

- 待定：规范对重复删除一条已不存在的记录究竟如何规定；所选文本中没有任何内容说明期望的是错误、204 还是重新创建。

### Z05: 其他NF的记录保持受保护

恶意合法 NF 控制自身域时，不能读取或删除其他 NF 存储的记录，目标域的保密与授权性质仍成立。

仅面对网络时该性质成立：没有另一个 NF 的凭证，就永远不会为其提供或删除任何记录。面对合法 NF 时则不成立：拥有该 API 范围的 NF 只要指明数据集的名称就能读取另一个 NF 的记录，并能通过数据规范删除它，或者在拿到标识符后通过 storeTransId 删除它。

- 结果：不成立（3 个攻击）；所问是否得到回答：是；可用
- 价值高：这就是本项发现：规范所定义的协议存在文稿未说明的弱点，并给出了确切的攻击者和确切的请求。
- 文稿中的表述：不明确。所依据的原文：s0034, s0063, s0069, s0373
- 依据：恶意 NF 的知识包括数据集标识符，以及在第一次删除尝试后从应答中学到的 storeTransId。；假设 ADRF 不按记录执行授权，因为 API 并未定义此类授权。
- 待定：规范是否期望 ADRF 对记录进行任何授权；所选文本中没有任何内容说明这一点。

### U01: 记录和读取者均真实

读取返回的记录由声称存储方真实存入，且请求方为获批读取方，storeTransId 与内容一致。

读取者接受的记录是真实且完整的，但读取者不必是存储 NF 所批准的一方：ADRF 会向任何指明该数据集的 NF 提供服务。应答的完整性成立；读取者的批准在 API 中并不存在。

- 结果：不成立（1 个攻击）；所问是否得到回答：是；可用
- 价值高：它准确说明了问题中哪一半成立、哪一半不成立，这正是编辑所需要的。
- 文稿中的表述：不明确。所依据的原文：s0008, s0034, s0373
- 依据：存储请求没有任何字段指明可以读取该记录的 NF，因此模型无从检查。；仅涉及成功检索路径；对读取者返回 204 的情形覆盖在展开模型中。
- 待定：部署时是否会在 API 之外自行增加对读取者的批准。

### U02: 读取和删除权限分离

读取与删除各自要求覆盖该操作的许可，二者不可互相替代。

读取和删除没有被分别授权：第 5.1.9 条的表为检索、删除和订阅操作只给出一个范围 nadrf-datamanagement:storage-read-delete-subs，因此为读取获得的令牌可以用于删除，反之亦然。工具显示了同一个令牌被用于两种操作；网络攻击者无法利用这一差异，因为它根本没有任何令牌。

- 结果：攻击成立（文稿中已说明）；所问是否得到回答：是；可用
- 价值中：规范在其范围表中说明了这单一范围，因此权限分离的缺失是该设计的一项已记录性质，而非新的发现；模型使其后果变得明确。
- 文稿中的表述：是。文稿中的说明位置：ts-33-501:2875, s0373, s0028, s0048
- 依据：该范围被建模为 NRF 按身份推导出的单一值；由于 API 未定义按操作区分的范围，ADRF 没有更细的权限可供检查。

### U03: 删除后无过期成功

删除生效或错误之后，旧记录不得作为新鲜成功结果被接受。

删除之后，读取者的下一次检索会得到 204 应答，旧载荷不会作为新鲜结果返回；应答通过读取者选择的某个新鲜值绑定到读取者自己的请求上。模型没有说明消费者会如何处理稍后再次送达的旧应答。

- 结果：在某个界内成立（变化值展开到 5 步）；所问是否得到回答：是；可带保留条件使用
- 价值中：它表明 API 在删除后的应答不是过期的成功，并用一个对照查询表明拒绝确实会被触发；保留条件在于界以及重放的抽象方式。
- 文稿中的表述：是。文稿中的说明位置：s0045, s0054, s0056
- 依据：每个应答都在读取者的密钥下绑定到按请求生成的值；TLS 连接内部的重放不在模型范围内。；同一条记录的五个展开步骤。
- 待定：在新连接内把旧应答重放给后续请求的情形未被覆盖。

### X01: 令牌和凭证机密性

An NF's TLS credential and the access token the NRF issues to it stay known to that NF, the NRF and the ADRF only; neither the network attacker nor another legitimate NF can obtain or forge a token for another NF's identity, so it cannot act towards the ADRF as that NF.

NF 的凭证以及 NRF 签发给它的令牌只为该 NF 所知：网络攻击者和另一个合法 NF 都无法获得它们，因此二者都无法以该身份对 ADRF 行事。对照查询表明攻击者确实能获得自己的令牌和凭证，因此该查询并非空查询。

- 结果：成立；所问是否得到回答：是；可用
- 价值中：它在一个 NF 具有敌意的确切场景下，确认了 RFC 6749 对令牌所要求的机密性。
- 文稿中的表述：是。文稿中的说明位置：rfc6749:2859, s0369, s0373
- 依据：令牌和凭证被建模为 NF 身份的私有派生值，并且只会以该 NF 的密钥发送。

# 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、是否检查了对照查询，以及每个攻击连同分析者的判定。

| 目标 | 级别 | 界 | 查询 | 对照 | 攻击 | 主张 |
|---|---|---|---|---|---|---|
| Z01 | proved | | z01_get | checked | | 4 proved |
| Z03 | proved | | z01_get | checked | z05_read (holds_in_source); u02 (holds_in_source) | 2 proved, 2 attack |
| K02 | proved | | k02_net | checked | k02_nf (holds_in_source) | 1 proved, 1 attack |
| D01 | proved | | d01 | checked | | 2 proved |
| S05 | bounded | 变化值展开到 5 步 | b_s05 | checked | | 2 bounded, 1 reachable |
| R02 | bounded | 变化值展开到 5 步 | b_r02 | checked | | 2 bounded, 1 reachable |
| Z05 | proved | | z05_net_get | checked | z05_read (holds_in_source); z05_rmspec (holds_in_source); z05_del (holds_in_source) | 1 proved, 3 attack |
| U01 | proved | | d01 | checked | z05_read (holds_in_source) | 1 proved, 1 attack |
| U02 | proved | | x01_tok | checked | u02 (holds_in_source) | 1 proved, 1 attack |
| U03 | bounded | 变化值展开到 5 步 | b_u03 | checked | | 2 bounded, 1 reachable |
| X01 | proved | | x01_tok | checked | | 2 proved |

逐条主张，按分析者对每条主张的措辞；每条主张的答案来自证明器：

| 目标 | 主张 | 攻击者 | 回答 | 查询 | 对照 |
|---|---|---|---|---|---|
| Z01 | 只有当请求方出示的访问令牌是由 NRF 签发给同一 NF 的，ADRF 才会为它应答 RetrievalRequest；网络攻击者不持有任何 NF 凭据，因此永远不会被服务。 | `network` | proved | z01_get | checked |
| Z01 | 只有当请求方出示的令牌是由 NRF 签发给它本人的，ADRF 才会为它应答 RetrievalRequest；恶意 NF 会被服务，但只能使用 NRF 签发给它自己的令牌。 | `malicious_nf` | proved | z01_get | checked |
| Z01 | 按 storeTransId 删除受调用者自身令牌的限制；恶意 NF 只持有自己的令牌。 | `malicious_nf` | proved | z01_del | checked |
| Z01 | 按数据规范删除受调用者自身令牌的限制；恶意 NF 只持有自己的令牌。 | `malicious_nf` | proved | z01_rmspec | checked |
| Z03 | 出示的令牌就是调用者自己的：只有请求携带的令牌是由 NRF 签发给它声称的身份时，请求才会被应答。 | `network` | proved | z01_get | checked |
| Z03 | 主体绑定对恶意 NF 同样成立：它只能以自己的身份和自己的令牌行事。 | `malicious_nf` | proved | z01_get | checked |

| 目标 | 主张 | 攻击者 | 回答 | 查询 | 对照 |
|---|---|---|---|---|---|
| Z03 | 令牌绑定到它所指向的记录：调用者只能对有权操作的记录行事。恶意 NF 持有自己的令牌和数据集合标识符。 | `malicious_nf` | attack (holds_in_source) | z05_read（z05_net_get 的对照） | |
| Z03 | 令牌绑定到签发它所要执行的操作。恶意 NF 只有一张覆盖该 API 范围的令牌。 | `malicious_nf` | attack (holds_in_source) | u02 | |
| K02 | 存储 NF 所存储的分析负载只有在合法 NF 的凭据落入网络攻击者手中时，才会到达网络攻击者。 | `network` | proved | k02_net | checked |
| K02 | 另一个 NF 记录的负载不会对合法 NF 保密：恶意 NF 只要指明数据集合标识符，就会被服务并读到该负载。 | `malicious_nf` | attack (holds_in_source) | k02_nf（k02_net 的对照） | |
| D01 | 消费者所接受的每一条记录都确实存储在 ADRF 上，且 storeTransId、存储 NF 和负载均相同。 | `network` | proved | d01 | checked |
| D01 | 恶意 NF 活跃时同样的对应关系成立：消费者所接受的是 ADRF 存储过的记录，但不一定是它所期望的存储 NF 所创建的记录（对照查询表明了这一点）。 | `malicious_nf` | proved | d01 | checked |
| S05 | 紧跟删除之后的检索会得到 204 应答——目标所述的情形可达。 | `network` | reachable | b_s05_ctl | |
| S05 | ADRF 删除记录之后，任何检索都不会返回其内容；它在删除时替换所保持的状态，并在应答前进行核对。 | `network` | bounded（变化值展开到 5 步） | b_s05 | checked |
| S05 | 删除之后，无论谁请求，任何检索都不会返回其内容；恶意 NF 持有自己的凭据和数据集合标识符。 | `malicious_nf` | bounded（变化值展开到 5 步） | b_s05 | checked |
| R02 | 重复的删除消息到达 ADRF 后会被拒绝——目标所述的情形可达。 | `network` | reachable | b_r02_ctl | |
| R02 | 对同一条记录的第二次删除会被拒绝：ADRF 替换所保持的状态，因此同一条消息不会让它再次行动。 | `network` | bounded（变化值展开到 5 步） | b_r02 | checked |
| R02 | 来自另一个 NF 的重放删除不会第二次删除该记录，也不会改变任何其他记录。 | `malicious_nf` | bounded（变化值展开到 5 步） | b_r02 | checked |
| Z05 | 恶意的合法 NF 无法读取另一个 NF 的记录：没有其他 NF 的凭据，该记录就永远不会被提供。 | `network` | proved | z05_net_get | checked |
| Z05 | 恶意的合法 NF 无法读取另一个 NF 的记录；它持有自己的凭据和令牌、该 API 范围，以及该记录的数据集合标识符。 | `malicious_nf` | attack (holds_in_source) | z05_read（z05_net_get 的对照） | |
| Z05 | 恶意的合法 NF 无法删除另一个 NF 的记录；它持有自己的凭据和令牌，以及该记录的数据规范。 | `malicious_nf` | attack (holds_in_source) | z05_rmspec | |
| Z05 | 恶意的合法 NF 无法按 storeTransId 删除另一个 NF 的记录；它持有自己的凭据和令牌，并获知了 storeTransId。 | `malicious_nf` | attack (holds_in_source) | z05_del | |
| U01 | 读取方所接受的内容确实是由存储 NF 以相同的 storeTransId 和内容存储的；网络上的任何东西都无法让它接受其他内容。 | `network` | proved | d01 | checked |
| U01 | 读取方是存储 NF 所批准的 NF；恶意 NF 持有自己的凭据和令牌以及数据集合标识符。 | `malicious_nf` | attack (holds_in_source) | z05_read（z05_net_get 的对照） | |

| 目标 | 主张 | 攻击者 | 回答 | 查询 | 对照 |
|---|---|---|---|---|---|
| U02 | 读取和删除各自都需要覆盖该操作的权限，二者不能相互替代；恶意 NF 只有一张覆盖该 API 范围的令牌。 | `malicious_nf` | attack (holds_in_source) | u02 | |
| U02 | 读取和删除都需要权限；网络攻击者根本不持有访问令牌，因此两个操作都无法调用。 | `network` | proved | x01_tok | checked |
| U03 | 消费者只把检索结果当作对自身请求的应答来接受；该目标所述的情形可达（ADRF 对后续请求的应答）。 | `network` | reachable | b_u03_ctl | |
| U03 | 删除之后，读取方的下一次检索得到 204 应答，旧负载不会作为新结果返回。 | `network` | bounded（变化值展开到 5 步） | b_u03 | checked |
| U03 | 删除之后，任何消费者（包括恶意 NF）都无法把旧记录作为新的检索结果获得。 | `malicious_nf` | bounded（变化值展开到 5 步） | b_u03 | checked |
| X01 | 签发给某个 NF 的访问令牌只有该 NF 和 NRF 知道：网络攻击者永远无法获得另一个 NF 的令牌。 | `network` | proved | x01_tok | checked |
| X01 | 某个 NF 的 TLS 凭据始终留在该 NF 手中：另一个合法 NF 无法获得它，因此无法以该 NF 的身份与 ADRF 交互。 | `malicious_nf` | proved | x01_key | checked |

# 附录 D：分析是怎么跑的

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

ProVerif+ Agent 会话 session-20261114T230426，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：51，证明器 6 秒，实际耗时 16 分钟，费用 0.150 美元。

ProVerif：运行 127 次证明，126 次得出结论，1 次无答案，耗时 4 秒，记录 36 个回答。

## 分析了什么

| | |
|---|---|
| **协议** | ADRF analytics record lifecycle |
| **版本** | 19.7.0 |
| **分析范围** | 单个非 UE 聚合 analytics record，StorageRequest→RetrievalRequest→Delete，一名存储 NF、一名允许读取 NF、一个恶意合法 NF；固定两状态有界删除；排除上游订阅、UE consent、记录过期调度。 |
| **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。 |
| **原文** | 443 个编号原文单元（331162 字节），来自下列各节。本报告中的 s0001 之类的引用就是这些原文单元之一 |

分析所依据的文稿各节如下：

| 文稿 | 章节 | 标题 |
|---|---|---|
| main | 4.2.2.2 | Nadrf_DataManagement_StorageRequest 服务操作 |
| main | 4.2.2.2.1 | 概述 |
| main | 4.2.2.2.2 | 请求存储数据或分析结果 |
| main | 4.2.2.5 | Nadrf_DataManagement_RetrievalRequest 服务操作 |
| main | 4.2.2.5.1 | 概述 |
| main | 4.2.2.5.2 | 请求并从 ADRF 数据存储获取已存储的数据或分析结果 |
| main | 4.2.2.9 | Nadrf_DataManagement_Delete 服务操作 |
| main | 4.2.2.9.1 | 概述 |
| main | 4.2.2.9.2 | 请求移除已存储的数据或分析结果 |
| main | 4.2.2.9.3 | 使用数据或分析结果规范请求移除已存储的数据或分析结果 |
| 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 |
| main | 5.1.2.2 | HTTP 标准头 |

| 文稿 | 章节 | 标题 |
| --- | --- | --- |
| main | 5.1.2.2.1 | 概述 |
| main | 5.1.2.2.2 | 内容类型 |
| main | 5.1.2.3 | HTTP 自定义头 |
| main | 5.1.3 | 资源 |
| main | 5.1.3.1 | 概览 |
| 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 | 概览 |
| 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 |
| main | 5.1.4.3 | 操作：request-storage-sub-removal |

| 文稿 | 章节 | 标题 |
|---|---|---|
| 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 | 简单数据类型与枚举 |
| 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 |
| main | 5.1.7.1 | 概述 |

| 文稿 | 章节 | 标题 |
|---|---|---|
| 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 | |

源文包在文稿中定位到的协议目的：单个非 UE 聚合分析记录，StorageRequest→RetrievalRequest→Delete，一名存储 NF、一名允许读取 NF、一个恶意合法 NF；固定两状态有界删除；排除上游订阅、UE 同意、记录过期调度。

## 对照文稿定性质

在编写任何查询之前，分析者已将草拟的性质清单对照文稿本身对协议用途的说明。名称、性质所涉及的内容以及文稿的说明均出自分析者的解读；所引位置确实存在于文稿中。

| ID | 简称 | 对象 | 文稿怎么说 | 出处 |
|---|---|---|---|---|
| Z01 | 令牌门控的检索和删除 | ADRF 只为一个 NF 服务消费者行事，该消费者出示的访问令牌是 NRF 签发给同一个 NF、用于 Nadrf_DataManagement 范围的；该令牌覆盖所请求的操作。 | 第 5.1.9 条说，对 API 的访问可以由 OAuth2 授权，NRF 作为授权服务器，消费者从它获取令牌，并列出 RetrievalRequest 和 Delete 操作的范围 nadrf-datamanagement:storage-read-delete-subs。 | s0368, s0369, s0371, s0373 |
| Z03 | 令牌绑定到主体和记录 | ADRF 检查令牌属于调用者本人、其受众是 ADRF，并且请求所指的记录是调用者可以操作的记录。 | API 为检索、删除和订阅操作定义了一个范围（5.1.9），对令牌绑定到 storeTransId 或绑定到资源所有者只字未提；RFC 6749 第 10.3 节指出，OAuth 没有为资源服务器提供任何方法来检查令牌是否签发给了出示它的客户端，第 10.16 节则指向受众限制。 | rfc6749:2879, s0373, s0034, s0053 |
| K02 | 载荷对其他NF隐藏 | 检索到的 NadrfDataStoreRecord 中携带的分析载荷只有存储它的 NF 和获准的读取者知道，另一个合法 NF 不知道，网络上的攻击者也不知道。 | 检索响应在 anaNotifications/dataNotif 中携带存储的分析数据（4.2.2.5.2）；第 5.1.9 条和 TS 33.501 5.9.2.1 要求 SBI 交换双向认证并受保护。 | s0037, s0039, s0040, s0368 |
| D01 | 提供的记录与存储的记录一致 | 读取者确信，它接受的内容就是 ADRF 在其请求的标识符下真正存储的记录，storeTransId 相同，载荷也相同。 | 检索按请求的参数找到数据并返回 NadrfDataStoreRecord（4.2.2.5.2）；TS 33.501 5.9.2.1 要求消费者与生产者之间双向认证，并要求每个 NF 校验传入的消息。 | ts-33-501:2871, s0034, s0036, s0037 |
| S05 | 无法检索已删除的记录 | ADRF 删除 Individual ADRF Data Store Record 之后，任何消费者，无论合法还是恶意，都无法再次获得其内容；ADRF 回答 204 或一个错误。 | 4.2.2.9.2 说 ADRF 应删除对应的存储记录并回答 204，4.2.2.5.2 说检索不存在的数据时回答 204。 | s0054, s0055, s0056, s0045 |

| ID | 简称 | 对象 | 文稿怎么说 | 出处 |
|---|---|---|---|---|
| R02 | 删除最多被接受一次 | 网络或另一个 NF 再次发送的已记录的删除请求不会让 ADRF 再次执行操作，也不会改变任何其他记录。 | 删除请求通过 storeTransId 或数据规范指定记录，ADRF 删除匹配的记录并回答 204（4.2.2.9.2, 4.2.2.9.3）；TS 33.501 5.9.2.1 要求每个 NF 按照协议规范和网络状态校验传入的消息。 | ts-33-501:2875, s0053, s0054, s0055, s0069 |
| Z05 | 其他NF的记录保持受保护 | 一个恶意但合法的 NF 服务消费者，持有自己的凭证和该 API 范围的令牌，无法读取或删除另一个 NF 存储的记录。 | 5.1.9 的单一范围覆盖整个 API 的存储、检索和删除；文稿没有说明任何按记录的所有权检查，因此保护一个 NF 的记录免受另一个合法 NF 侵害的只有范围和记录标识符。 | s0373, s0034, s0053, s0069 |
| U01 | 记录和读取者均真实 | 读取者接受的内容确实是由存储它的 NF 存储的，读取者是存储该记录的 NF 所批准的 NF；storeTransId 和内容与存储时一致。 | ADRF 分配 storeTransId 并返回记录（4.2.2.2.2 第 2 步），检索返回按请求参数找到的记录（4.2.2.5.2）；文稿对将存储的记录绑定到可以读取它的 NF 只字未提。 | s0019, s0022, s0034, s0037 |
| U02 | 读取和删除权限分离 | 读取和删除记录各自需要覆盖该操作的权限，一个不能代替另一个。 | 第 5.1.9 条为 RetrievalRequest 和 Delete 操作定义了同一个范围 nadrf-datamanagement:storage-read-delete-subs，因此文稿没有区分这两种权限。 | s0028, s0048, s0373 |
| U03 | 删除后无过期成功 | 删除或出错之后，旧的成功响应不能作为新的检索结果呈现给消费者。 | 数据不存在时 ADRF 回答 204，删除成功后也回答 204（4.2.2.5.2, 4.2.2.9.2）；文稿对响应重放或缓存只字未提。 | s0045, s0054, s0055, s0056 |
| X01 | 令牌和凭证机密性 | NF 服务消费者的访问令牌和客户端凭证，针对另一个 NF 和针对网络。 | RFC 6749 第 10.3 节要求访问令牌凭证保密，禁止未授权方生成、修改或猜测有效令牌，第 10.10 节限定了猜测概率；TS 33.501 5.9.2.2 规定 NRF 为授权服务器，并要求它和各个 NF 之间双向认证。 | rfc6749:2859, rfc6749:2870, rfc6749:3032, ts-33-501:2883 |

分析者依据文稿补充的性质：X01.

## 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 30 段，没选 368 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 | 关于 OAuth 安全性取决于部署方式的泛泛说明；没有陈述本协议必须提供的任何性质。 |
| P003 至 P006 | rfc6749:2790-2823 | X01, `malicious_nf` | |
| P007 至 P010 | rfc6749:2827-2850 | Z03, `malicious_nf` | |
| P011 至 P015 | rfc6749:2859-2881 | Z01, Z03, X01, `malicious_nf` | |
| P016 至 P021 | rfc6749:2885-2917 | 不在范围内 | 刷新令牌；此 API 仅使用客户端凭据授予（5.1.9）。 |

| 段落 | 出处 | 定为 | 不选的原因 |
|---|---|---|---|
| P022 至 P025 | rfc6749:2921-2943 | 不在范围内 | 授权码；客户端凭据授予不使用。 |
| P026 至 P029 | rfc6749:2947-2980 | 不在范围内 | 授权码重定向；此流程中没有用户代理或重定向。 |
| P030 至 P033 | rfc6749:2984-3003 | 不在范围内 | 资源所有者口令凭据授予；不使用。 |
| P034 至 P035 | rfc6749:3013-3019 | X01 | |
| P036 | rfc6749:3023-3028 | Z01, D01 | |
| P037 至 P039 | rfc6749:3032-3041 | X01 | |
| P040 至 P042 | rfc6749:3045-3066 | 不在范围内 | 对最终用户的钓鱼；此 API 子流程中没有最终用户参与。 |
| P043 至 P048 | rfc6749:3070-3110 | 不在范围内 | 针对用户代理重定向端点的 CSRF；此处没有用户代理。 |
| P049 至 P050 | rfc6749:3119-3138 | 不在范围内 | 授权页面的点击劫持；没有浏览器交互。 |
| P051 至 P052 | rfc6749:3142-3150 | 没有陈述目标 | 建议实现者对输入做净化；没有陈述交换的任何性质。 |
| P053 至 P054 | rfc6749:3154-3167 | 不在范围内 | 开放重定向；此流程中没有重定向。 |
| P055 至 P062 | rfc6749:3170-3210 | 不在范围内 | 隐式流程和冒充资源所有者；此 API 使用客户端凭据授予，没有资源所有者。 |
| P063 | ts-33-501:2491-2491 | 证明器问不了 | 协商安全特性的降级；模型没有协商，也没有就算法选择询问 ProVerif。 |
| P064 至 P072 | ts-33-501:2495-2511 | 不在范围内 | UE 与服务网络的认证和授权；此 API 子流程没有 UE。 |
| P073 至 P074 | ts-33-501:2515-2517 | 不在范围内 | AS/NAS 保护的密钥长度和密钥派生；不是 ADRF API。 |
| P075 至 P078 | ts-33-501:2523-2529 | 不在范围内 | 对 UE 的要求。 |
| P079 至 P089 | ts-33-501:2533-2553 | 不在范围内 | UE 用户数据和信令数据在无线接口上的机密性。 |
| P090 至 P103 | ts-33-501:2557-2583 | 不在范围内 | UE 用户数据和信令数据在无线接口上的完整性。 |
| P104 至 P110 | ts-33-501:2587-2599 | 不在范围内 | 签约凭据的存储；此有界子流程中没有签约数据。 |
| P111 至 P125 | ts-33-501:2603-2631 | 不在范围内 | 签约用户隐私；所分析的记录是非 UE 的聚合分析记录，且任务排除了 UE 同意和签约。 |
| P126 | ts-33-501:2637-2637 | 不在范围内 | gNB 要求的通用标题。 |
| P127 至 P136 | ts-33-501:2641-2659 | 不在范围内 | UE 与 gNB 之间用户数据和信令数据的机密性。 |
| P137 至 P147 | ts-33-501:2663-2683 | 不在范围内 | UE 与 gNB 之间用户数据和信令数据的完整性。 |

| 段落 | 出处 | 定为 | 不选的原因 |
|---|---|---|---|
| P148 至 P178 | ts-33-501:2687-2761 | 不在范围内 | 对 gNB 设置、密钥管理、F1/E1 接口以及 ng-eNB 的要求；都与 ADRF API 无关。 |
| P179 至 P192 | ts-33-501:2767-2795 | 不在范围内 | 接入网侧的信令机密性和完整性，不是此 API 的服务化接口。 |
| P193 至 P198 | ts-33-501:2799-2811 | 不在范围内 | 签约用户隐私和 SEAF 要求。 |
| P199 至 P213 | ts-33-501:2819-2853 | 不在范围内 | 对 UDM、SIDF、AUSF 和 UDR 的要求；所分析的子流程排除了签约数据和用户同意。 |
| P214 | ts-33-501:2859-2859 | Z01, K02, D01, `malicious_nf` | |
| P215 至 P217 | ts-33-501:2865-2869 | 不在范围内 | 服务注册与发现以及拓扑隐藏；任务排除了发现，且模型假定消费者知道 ADRF。 |
| P218 | ts-33-501:2871-2871 | Z01, Z03, D01, `malicious_nf` | |
| P219 至 P220 | ts-33-501:2873-2875 | D01, S05, R02 | |
| P221 至 P227 | ts-33-501:2879-2891 | Z01, X01, `malicious_nf` | |
| P228 至 P234 | ts-33-501:2895-2907 | 不在范围内 | NEF 安全要求；NEF 不是此子流程的参与方。 |
| P235 至 P241 | ts-33-501:2911-2923 | 不在范围内 | 服务通信代理；模型假定消费者与 ADRF 之间直接通信。 |
| P242 至 P254 | ts-33-501:2929-2953 | 不在范围内 | 端到端核心网互联和漫游；场景位于单一运营商域内。 |
| P255 至 P292 | ts-33-501:2957-3031 | 不在范围内 | SEPP 对 N32 的要求；此子流程没有漫游接口。 |
| P293 至 P303 | ts-33-501:3035-3055 | 不在范围内 | N32 上的漫游中间节点。 |
| P304 至 P312 | ts-33-501:3059-3075 | 不在范围内 | N32-f 上的属性保护；此处没有 SEPP。 |
| P313 至 P314 | ts-33-501:3079-3081 | 不在范围内 | IPUPS 包过滤；不属于 ADRF API。 |
| P315 至 P318 | ts-33-501:3085-3091 | 没有陈述目标 | 3gpp-Sbi-Originating-Network-Id 头的部署规则，文稿描述它时没有涉及记录生命周期的任何性质。 |
| P319 至 P327 | ts-33-501:3097-3115 | 不在范围内 | 信令流量监控；是另一项特性。 |
| P328 至 P337 | ts-33-501:3121-3141 | 不在范围内 | 安全特性对 UE 和用户可见性与可配置性。 |
| P338 至 P369 | ts-33-501:3149-3215 | 证明器问不了 | 算法标识、算法强度和选择；没有就密码强度询问 ProVerif。 |
| P370 至 P375 | ts-33-501:3219-3231 | 不在范围内 | 对 5G-RG 和 NSSAAF 的要求。 |
| P376 至 P380 | ts-33-501:6540-6548 | 不在范围内 | N3 上的安全。 |
| P381 至 P394 | ts-33-501:9656-10917 | 不在范围内 | EAP-TLS 附录示例和 IAB 要求。 |

| 段落 | 出处 | 定为 | 不选的原因 |
|---|---|---|---|
| P395 至 P398 | ts-33-501:11439-11901 | 不在范围内 | 附录中 MBSF/MBSTF 和 NEF 的要求。 |

## 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 23 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 11 项，没设的 12 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 service consumer | 参与方 | `malicious_nf` | |
| S02 | 恶意合法 NF | 参与方 | `malicious_nf` | |
| S03 | ADRF | 参与方 | 按定义全部失守 | ADRF 是存储并提供记录的参与方；一旦它落入攻击者之手，这里所问的每条性质都按定义失守，而文稿在不诚实的 ADRF 下不问其中任何一条。 |
| S04 | NRF | 参与方 | 按定义全部失守 | 不诚实的授权服务器会为任何身份签发令牌，因此这里所问的每条授权性质都按定义失守；文稿在被攻陷的 NRF 下不问其中任何一条。 |
| S05 | 允许读取 NF | 参与方 | `malicious_nf` | |
| S06 | 存储方 NF | 参与方 | `malicious_nf` | |
| S07 | OAuth2 访问令牌及其 scope | 值 | `malicious_nf` | |
| S08 | 存储记录 storeTransId | 值 | `malicious_nf` | |
| S09 | 记录中的数据/analytics | 值 | 按定义全部失守 | analytics 内容本身；攻击者持有它就是持有机密性目标所保护的那个秘密，这就是泄露本身，而不是进一步的攻击者。 |
| S10 | 访问令牌 | 值 | `malicious_nf` | |
| S11 | storeTransId | 值 | `malicious_nf` | |
| S12 | 记录内容 | 值 | 按定义全部失守 | 攻击者持有的记录内容就是机密性目标本身的失守。 |
| S13 | NadrfDataStoreRecord 中的 dataNotif/anaNotifications 内容 | 值 | 按定义全部失守 | 攻击者持有的检索到的载荷就是机密性目标本身的失守。 |
| S14 | 记录载荷 | 值 | 按定义全部失守 | 攻击者持有的记录载荷就是机密性目标本身的失守。 |
| S15 | 已删除记录 | 值 | 攻击者得不到更多 | 已删除记录的副本不能告诉攻击者 ADRF 是否仍在提供它；删除目标说的是 ADRF 返回什么。 |
| S16 | 删除请求 | 值 | `malicious_nf` | |
| S17 | 记录状态 | 值 | 按定义全部失守 | 记录状态保存在 ADRF 内部；控制它的攻击者就掌控了 ADRF，而在这种情况下这里所问的任何性质都无法成立。 |
| S18 | 其他 NF 的记录 | 值 | 按定义全部失守 | 持有其他 NF 的记录正是机密性和授权目标所描述的失守。 |
| S19 | 读取许可 | 值 | `malicious_nf` | |
| S20 | 删除许可 | 值 | `malicious_nf` | |
| S21 | 记录 | 值 | 按定义全部失守 | 持有记录就是失守本身。 |

| ID | 参与方或值 | 类别 | 攻击者条件 | 不设的原因 |
|---|---|---|---|---|
| S22 | 错误响应 | 值 | 攻击者得不到更多 | 错误响应由 ADRF 产生并保护给需要它的 NF；产生或重放一个都不需要超出网络的能力，而网络正是回答这些目标所依据的基础攻击者。 |
| S23 | 缓存响应 | 值 | 攻击者得不到更多 | 重放缓存响应不需要超出网络的能力，而网络是基础攻击者；由另一个 NF 持有的存储副本已由恶意 NF 覆盖。 |

## 每条性质在各攻击者条件下的答案

攻击者是分析者在编写查询之前设定的；每个答案都是在其中一个攻击者下记录的。单元格是证明器在该攻击者下对相应性质的答案；"无答案"是分析者为该性质设定的攻击者，在该攻击者下证明器让某条主张悬而未决，或者没有记录任何答案；空单元格是分析者没有为该性质设定的攻击者。

| 攻击者 | 持有什么 | 依据 | 出处 |
|---|---|---|---|
| `network` | 只有网络：它读取、丢弃、修改并发送消息，不持有任何一方的秘密 | | |
| `malicious_nf` | 运营商视为合法的一个 NF 服务消费者的凭据：它自己的 TLS 端点凭据、NRF 为它签发的、范围为 nadrf-datamanagement:storage-read-delete-subs 的访问令牌，以及它所针对的记录的标识符（它收到的 storeTransId，或描述分析数据的 dataSetId）。它不持有任何其他 NF 的密钥，也不读取其他 NF 的 TLS 会话。 | 5.1.9 的范围表为每个被授权访问该 API 的 NF 发放一个同时用于存储、检索和删除的令牌；检索请求可以通过 store-trans-id 或 data-set-id（4.2.2.5.2）指定记录，删除请求则通过数据规范（4.2.2.9.3）指定，因此消费者可以寻址并非由它存储的记录。TS 33.501 5.9.1 将信任区内的每个 NF 都置于边界之内，因此敌对的 NF 是持有自己凭据的合法内部人员。 | s0373, s0034, s0069, s0368 |

| ID | `network` | `malicious_nf` |
|---|---|---|
| Z01 | 成立 | 成立 |
| Z03 | 成立 | 不成立 |
| K02 | 成立 | 不成立 |
| D01 | 成立 | 成立 |
| S05 | 成立 | 成立 |
| R02 | 成立 | 成立 |
| Z05 | 成立 | 不成立 |
| U01 | 成立 | 不成立 |
| U02 | 成立 | 不成立 |
| U03 | 成立 | 成立 |
| X01 | 成立 | 成立 |

## 原文和目标

Source: 443 units, 331162 bytes, frozen in source-delivery-01.json. Targets: 10 approved (Z01, Z03, K02, D01, S05, R02, Z05, U01, U02, U03); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

宿主按文档规定的默认值（任务 `decisions: auto`）做出的决定，供你核对：

- approve_package: 批准源文包 protocol-f1f52cf258d8c7809ece111ccf09

本步骤的自我报告：1 次请求（1 次失败），0.000 USD，输入 152430 个 token，输出 4129 个，耗时 202 秒。

## 关于证明器和模型的说明

- S05：在 ProVerif 中作答，分析者为该目标指定了 changing_state
- R02：在 ProVerif 中作答，分析者为该目标指定了 changing_state
- U03：在 ProVerif 中作答，分析者为该目标指定了 changing_state

一个对代数做了化简（xor、dh_algebra、other_algebra）的 ProVerif 模型，给攻击者的能力弱于真实运算：在那里得到的证明是对更弱攻击者的证明，在那里得到的攻击则是真实的攻击。一个省略了对变化值检查（changing_state）的 ProVerif 模型，会让各方接受更多：在那里得到的证明成立，在那里得到的攻击可能只是缺少了该检查。Tamarin+ Agent 按原样对这些进行建模。

以步数给出的界是 bounded.pv 上的答案：它是分析者的第二个 ProVerif 模型，其中保存变化值的一方带着检查被展开到该步数。该模型所做的少于协议本身，因此该性质仅在这些步数内成立，不再延伸，且在那里发现的攻击是真实的攻击。步数由分析者给出。

## 分析者的状态审查

这部分是分析者对原文的解读，不是证明器的答案。ProVerif 所示下的“是”表示 ProVerif 在展开模型中找到了带有这些步骤的轨迹，因此这些步骤是可能的；它们是否破坏了某个目标，只由答案表中的一次攻击说明，它们还付出了什么代价则是分析者的解读。“未发现”说明查看了哪些模式，并不证明任何东西。

| 值 | 保存方 | 何时变化 | 何时检查 | refs |
|---|---|---|---|---|
| 一条 Individual ADRF Data Store Record 连同其载荷及其 storeTransId 的存在 | ADRF | StorageRequest 创建它，ADRF 分配一个 storeTransId（s0018, s0019）；ADRF 处理并接受的 Delete 会将其移除，通过 storeTransId（s0054, s0055）或通过数据或分析规范（s0069）。 | 在每个 RetrievalRequest 上，ADRF 查找与请求参数匹配的记录时（s0036），以及在每个 Delete 上，ADRF 在移除任何东西之前（s0054）。 | s0018, s0019, s0036, s0045, s0054, s0055, s0057, s0069 |
| 一条已存储记录的 storeTransId | ADRF 和存储该记录的 NF 服务消费者 | 每个 StorageRequest 都会重新分配，因此每条已存储记录都有自己的 storeTransId（s0019），存储的 NF 从 Location 头获知它（s0022）。 | ADRF 在 Delete 中用它查找记录（s0053），消费者可以在 RetrievalRequest 中指定它（s0034）。 | s0019, s0022, s0034, s0053 |

| 模式 | 结果 | 目标 | ProVerif 所示 | 步骤 |
|---|---|---|---|---|
| replay | 未发现 | | | bounded.pv St3 先测试所保存的状态，然后才触发移除，将 gone0 传给下一步；到达 ADRF 的第二次删除被拒绝，因此不会发生第二次接受（b_r02 verified，b_r02_ctl 显示拒绝已到达）。 |
| repeat_one_time_action | 未发现 | | | 移除是 ADRF 对每条记录执行一次的步骤；此后状态被替换，重复或重放的删除会得到错误应答，而不会再次移除任何东西。 |
| desynchronisation | 未发现 | | | 记录状态仅由 ADRF 持有；存储的 NF 除了 storeTransId 之外不保存任何东西，因此不存在可能漂移的第二份副本，且在五步内的搜索找不到可用的不一致。 |
| update_before_check | 未发现 | | | 在 bounded.pv St3 中，状态在执行移除之前先被测试，新状态仅在检查之后才传给下一步；没有任何轨迹会在检查失败时移除记录。 |
| no_rollback_on_failure | 未发现 | | | 找不到记录的删除会让所保存的状态保持原样：拒绝分支将未改变的状态传下去（bounded.pv St3 和 St5），因此失败或中止的运行不会假装记录已不存在。 |

| 模式 | 结果 | 目标 | ProVerif 所示 | 步骤 |
|---|---|---|---|---|
| race_between_sessions | 未发现 | | | model.pv 将记录保存在一个只增不减的表中，因此在那里对同一记录的两次删除都会被接受；那是单调表，不是协议。在状态被替换的 bounded.pv 中，第二次删除被拒绝，且规范规定 ADRF 在每一步只读取并改变一个记录状态，因此两次会话的任何顺序都不会以某个目标所禁止的方式移除记录两次。 |
| distinguishable_failure | 候选 | | 是：ProVerif 在 bounded.pv 上否证了 `b_s05_ctl` | 恶意 NF 持有自己的凭据和范围为 nadrf-datamanagement:storage-read-delete-subs 的令牌（s0373）。/ 它发送一个指定数据集标识符的 RetrievalRequest（s0034）。/ ADRF 找到记录，以 200 返回该记录；或找不到，以 204 应答（s0037, s0045）。/ 两种应答不同，因此该 NF 得知另一 NF 的记录是否存在。 |
| wrap_or_window | 未发现 | | | 记录状态不使用计数器、序列号或接受窗口；storeTransId 对每个 StorageRequest 重新分配，且从不与更早的相比较，因此没有可回绕的东西。 |

分析者为各目标指定的内容：

| 步骤 | 目标 | 特征 | 理由 |
|---|---|---|---|
| 38 | S05, R02, U03 | changing_state | ADRF 保存一条 Individual ADRF Data Store Record 的状态——它在 StorageRequest 之后存在，并被 Delete 移除——而这三个目标关乎该变化带来的后果：此后没有检索会返回载荷，第二次删除不被接受，且没有旧载荷会被当作新结果接受。model.pv 无法表达移除（表只增不减），因此答案来自 bounded.pv，其中状态被展开为五步，对所保存值的检查按条款 4.2.2.9.2 和 4.2.2.5.2 的陈述编写。 |
| 38 | Z01, Z03, K02, D01, Z05, U01, U02, X01 | plain_primitives, many_sessions | NF 与 ADRF 或 NRF 之间的 TLS 被建模为每端点的对称密钥，OAuth2 令牌被建模为 NF 身份的单向私有函数，因此答案仅依赖于哈希、MAC 和加密，不含任何证明器可能出错的代数定律。角色被复制，因此每个性质都针对存储 NF、读取方和恶意 NF 的任意数量会话作答。 |

## 分析者的结语

十一个目标（Z01、Z03、K02、D01、S05、R02、Z05、U01、U02、U03 以及新增的目标 X01）已针对网络攻击者和恶意但合法的 NF 作答，解读和图已记录。

成立的部分：只有用自己的凭据认证并出示 NRF 为其签发的令牌的 NF 才会被服务或遵从，检索和两种形式的删除皆然（Z01 已证明）；NF 的凭据和令牌留在它自己手中（X01 已证明）；消费者接受的是 ADRF 确实存储的记录，且 storeTransId、存储 NF 和载荷都相同（D01 已证明，一个被否证的对照查询表明存储方不必是读取方所预期的 NF）；移除之后 ADRF 应答 204，重复删除被拒绝（S05、R02、U03 在展开模型上成立，对照查询被否证）；载荷从不到达网络攻击者（K02 已证明）。

这条发现：由于第 5.1.9 条为存储、检索和删除定义同一个范围，而 ADRF 只检查该范围，因此一个知道数据集标识符的合法 NF 可以读取另一个 NF 的分析数据并删除其记录：可以按数据规范，也可以按 storeTransId（Z05、K02、Z03、U01 攻击；另外还有读权限与删权限之间失去分离，U02，以及由 200/204 应答构成的存在性预言机）。一项经检查的修改——ADRF 只对存储该记录的 NF 或经其批准的 NF 操作该记录——消除了针对另一个 NF 的三种攻击，并保留了正常流程。

还有哪些阻碍：无法向 ProVerif 询问令牌过期（没有时间），因此 Z01 的那条主张仍只得到部分回答；记录生命周期只展开五步，而且执行删除的是模型，不是无界模型中的单调表。TLS 和 OAuth2 被抽象为按端点的密钥和不可伪造的令牌，因此对原语本身没有任何说明；受损的 ADRF 或 NRF、SCP 和漫游不在分析范围内。