NWDAF 训练数据管理的安全性分析

本报告的其他语言版本：English。

19.8.0 · ProVerif+ Agent 报告

文稿	5G System; Network Data Analytics Services; Stage 3 （协议定义）
检查的性质	13 条：11 条成立，2 条不成立（2 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

NWDAF 的 Nnwdaf_DataManagement 服务允许另一个网络功能订阅聚合后的非个人训练数据，在数据就绪时收到通知，获取数据并取消订阅。参与方有两方：NF 服务消费者持有自己的 OAuth2 访问令牌，该令牌具有单一服务范围 nnwdaf-datamanagement，并持有自己的 TLS 会话；NWDAF 持有订阅记录和已收集的数据。NRF 充当授权服务器，每个请求、通知和获取操作都通过服务化接口上双向认证的 TLS 连接传输。

交互分为四步：消费者提交订阅，携带其通知 URI 和关联标识符；NWDAF 返回已创建的订阅及其订阅标识符；NWDAF 在该 URI 上向消费者发送通知并附上获取指令；消费者使用收到的关联标识符获取数据；随后删除订阅。最终，消费者确信每条通知和载荷都携带自己订阅的关联器，并且都经由自己的 TLS 连接到达；NWDAF 则确信其所服务的对象出示了服务范围的令牌，并指明了一条它持有的订阅记录。

参与方：

- **NF 服务消费者**：订阅、接收通知、获取数据、取消订阅。持有：访问令牌（范围 nnwdaf-datamanagement）、notifCorrId、notificURI。
- **NWDAF**：服务生产者。持有：订阅记录、已收集并汇总的数据。
- **NRF**：签发访问令牌。持有：NF 配置文件，充当授权服务器。

图 1 展示了文稿所规定的交互。

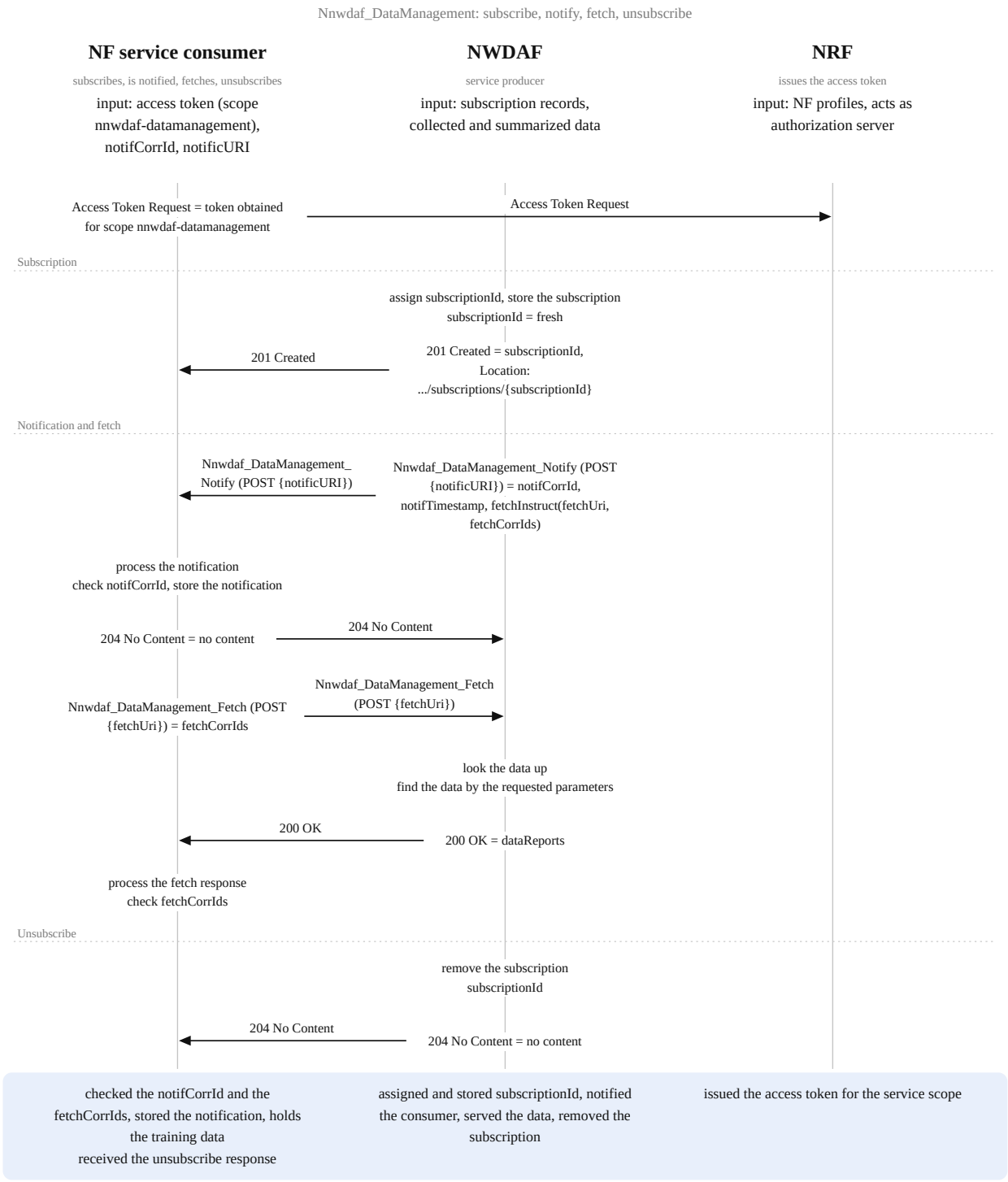


图 1. 文稿所规定的协议。

分析范围见附录 D：分析了什么、遗漏了什么，以及所依据的文稿章节。

2 发现

分析共有 1 项发现：以下内容在文稿中并未说明。其中 1 项，文稿是否说明尚不明确。

一项发现：该 API 只有单一的服务级范围，且不检查调用方是否为订阅的创建者，因此一个知道 subscriptionId 的合法 NF 可以通过 Fetch 读取另一个消费者的训练数据，并删除该消费者的订阅。在十三条性质中，六条得到证明，两条由该攻击给出答案，另有五条仅在弱于其要求的解读下得到证明：A02、

A03、A04、D01、A08 和 U01 之所以成立，是因为模型保留了来自其他 NF 的通知端点和关联器，而规范从未要求这一点；Z04、S05 和 U03 则悬而未决，因为模型从不删除订阅记录。这些结果基于 TLS 的符号模型，令牌是原子性的，没有令牌代数，因此它们无法说明计算强度。

攻击图沿用图 1 的列。红色表示攻击者的行为。黄色标记某一方在指定交互中本不该在此时发送的消息，以及攻击所破坏的结果。

2.1 一个知道 subscriptionId 的合法 NF 读取另一个消费者的训练数据并删除其订阅

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, U02 （见第 3 节）
原文	s0078, s0079, s0115, s0116, s0117, s0156, s0226, s0230, s0341, s0350
依据	ProVerif attack queries `z03`, `u02`; Figures 2 and 3

发生了什么。 NWDAF 仅以 subscriptionId 作为订阅的键：任何出示单一服务范围令牌的 NF，只要指明某个订阅，其 DELETE 或 Fetch 请求就会被处理。订阅的创建者从不与调用方进行比对，应用错误表中也没有针对外部资源的拒绝条目。在模型中，恶意 NF 删除诚实消费者的订阅，并通过 Fetch 获取其训练数据。

攻击者需要什么。 同一网络中的一个合法 NF：自己用于单一服务范围的访问令牌、自己到 NWDAF 的 TLS 连接，以及受害者的 subscriptionId——它是资源 URI 中的普通字符串，在任何地方都未声明为机密。

攻击者能得到什么。 它读取为该订阅收集的聚合训练数据并删除该订阅，从而获知数据，并使受害者丧失其通知和获取到的数据。

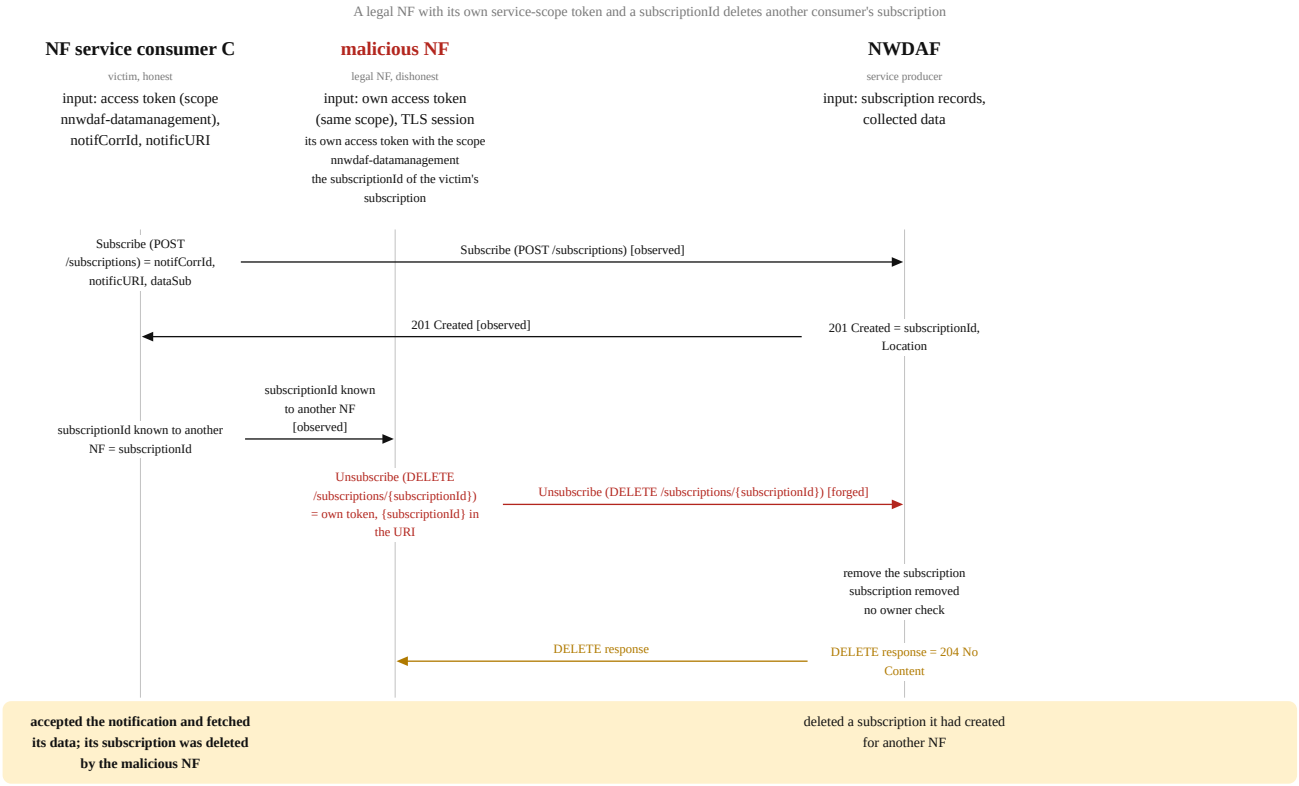


图 2. 一个持有自己服务范围令牌和 subscriptionId 的合法 NF 删除另一个消费者的订阅。性质 Z03；攻击查询 z03。

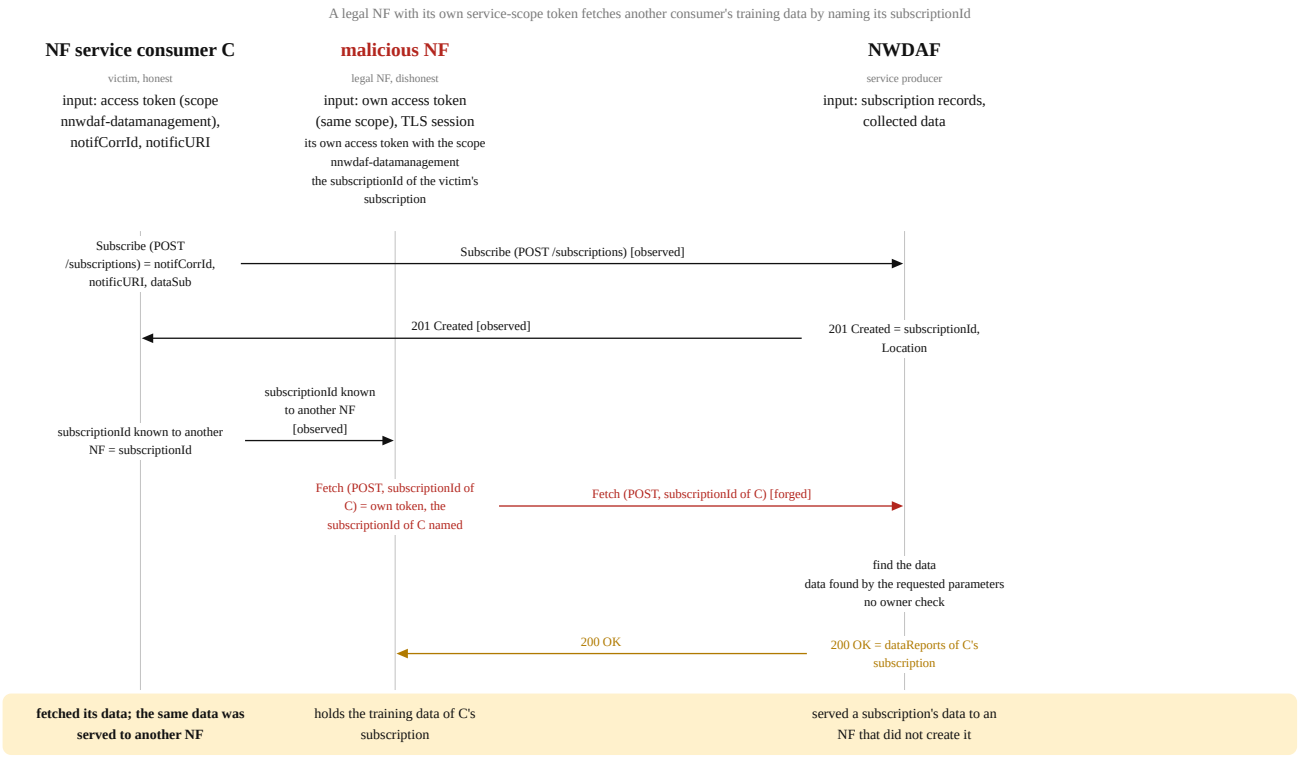


图 3. 一个持有自己服务范围令牌的合法 NF 通过指明另一个消费者的 subscriptionId 获取其训练数据。性质 U02；攻击查询 u02。

拟议修改。 为每个订阅记录访问令牌签发时所对应的身份，并将其与该 subscriptionId 上后续每次 PUT、DELETE 和 Fetch 的令牌身份进行比对，不一致时返回 403 Forbidden；按订阅划分范围则可使同一检查在授权服务器处强制执行。该修改已在模型副本上检查：两个攻击均变得不可证明，正常流程仍能被找到；在其余已记录的答案中，只有两个仅涉及网络的解读发生变化，因为在修复后的模型中，合法 NF 无需任何人泄露外部标识符即可获得自己的订阅。

用 ProVerif 复核。 该修改被加入模型副本：NWDAF 为每个订阅存储该订阅创建时所依据的身份（即 Nnwdaf_DataManagement_Subscribe 请求被授权的身份），并将其与该 subscriptionId 上后续每次 Fetch 和 DELETE 请求方的身份进行比对，不一致时返回 403 Forbidden，而不是执行删除或返回数据。在修改后的模型上，ProVerif 证明了 `z03`：该攻击在那里不再可能。协议的正常流程仍能被找到。在其余已记录的答案中，23 条仍然成立，3 条发生变化：`z03_net`（Z03，原为成立）现在不成立；`u02_net`（U02，原为成立）现在不成立；`u02`（U02，原为攻击）现在验证通过。这是对模型的首次检查：它表明该修改在模型中消除了此攻击，而不是表明它不会引入其他问题。修改后的模型：`records/fixes/0040-fix-owner.pv`。

对于一项发现的这些攻击，未给出图：U02：z03，Z03：u02。

3 检查过的性质

每一行是一条提交给 ProVerif 的性质。结果是 ProVerif 对分析者所建协议模型给出的答案；价值和文稿是否已说明该结果，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
A02	**通知发送方是 NWDAF。** 每当消费者接受一条来自 NWDAF 的通知或订阅响应，就存在一个真实的 NWDAF 运行，其意图身份与消费者所认为的对端身份一致。	无	成立	低	否

ID	性质	假定的泄露	结果	价值	文稿是否已写
A03	**关联器将通知绑定到订阅。** 每当消费者接受一条通知或 Fetch 响应，就存在一个匹配的 NWDAF 运行，其角色、双方身份以及 notifCorrId、fetchCorrIds、subscriptionId 等关联字段与消费者本地值一致。	无	成立	低	否
A04	**每次 NWDAF 运行仅接受一次。** 消费者每次接受订阅响应或通知，都对应一个不同的 NWDAF 运行；不存在两次接受映射到同一 NWDAF 运行的情况。	无	成立	低	否
D01	**通知载荷来自 NWDAF。** 每次消费者接受通知数据，都存在允许发送方 NWDAF 的真实发送，且载荷与获批的关联字段一致。	无	成立	低	否
A08	**通知字段与订阅匹配。** 消费者接受的通知与 Fetch 响应，其 notificURI、notifCorrId、fetchUri、fetchCorrIds 等字段对应消费者在订阅中提出的值，且属于当前安全上下文。	无	成立	低	否
K01	**访问令牌与 SBI 密钥保密。** 攻击者不能推导出消费者的 OAuth2 访问令牌或 NWDAF 与消费者之间的 SBI 会话密钥。	无	成立	低	是
Z01	**每个操作均被授权。** 每次 NWDAF 提交订阅创建/更新/删除或返回 Fetch 数据，都存在一个由授权服务器签发的、对请求方有效且覆盖 nnwdaf-datamanagement scope 的许可。	无	成立	中	是
Z03	**操作主体为订阅创建者。** 每次对某 subscriptionId 的更新、删除或 Fetch，其授权主体与该订阅的创建者一致，且操作与令牌受众、资源相符。	无	不成立（1 个攻击）	中	不明确
Z04	**终止后无通知。** 在订阅被删除或 NWDAF 请求终止之后，针对该 subscriptionId 不再产生新的通知或 Fetch 数据。	无	成立	none	否
S05	**旧标识符停止工作。** 订阅终止生效点之后，使用旧 subscriptionId、notifCorrId 或 fetchUri 不能触发新的通知投递或 Fetch 数据返回。	无	成立	none	否
U01	**关联器仅服务于其消费者。** 通知与 Fetch 中的 notifCorrId/fetchCorrIds 只对发起该订阅的消费者有效。	无	成立	低	否
U02	**其他 NF 无法读取或替换数据。** 恶意合法 NF 不能替换或读取其他订阅的数据。	无	不成立（1 个攻击）	中	不明确
U03	**取消订阅后无通知。** 取消订阅生效后，消费者不再接受该订阅的通知。	无	成立	none	否

提出的问题及分析者的回答:

问题	回答	依据
Q1: 订阅/通知/Fetch 的 correId 与请求者绑定	只有消费者一侧被绑定：消费者只接受由其自己选择的关联器所对应的通知或 Fetch 响应，这一点已经验证。在生产者一侧，两个关联器都没有与消费者身份绑定，因此关联标识符不是授权令牌：另一个知道 subscriptionId 的合法 NF 会在其自己的关联器下被服务。	U01, A08, Z03, U02
Q2: 跨订阅数据替换与恶意合法 NF 的读取边界	一个拥有自身服务范围令牌的恶意合法 NF，通过使用自己未创建的 subscriptionId 调用 Fetch 来读取另一个消费者的训练数据，并删除该订阅；这两种攻击在模型上都成立，因此 API 所能给出的边界是资源标识符的保密性，而不是令牌。在该模型上，它无法向受害者的消费者注入通知，因为受害者的通知端点和关联器对它而言是未知的。	U02, Z03, A02, A03
Q3: 有限次数通知及取消后陈旧响应处理	两者都没有定论：规范没有对通知次数设置限制，而模型也无法表明退订后不会有任何消息到达，因为在该模型中订阅记录从未被移除。两者都保持开放，需要对记录建立有状态模型。	Z04, S05, U03

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和检查给出了什么或未能给出什么，其中密码学被视为理想的，且不涉及任何实现。

正如分析者所述，本分析不覆盖以下内容：

- TLS 被建模为每个端点上的私有通道：网络攻击者读不到明文，也无法伪造或重放，且对端的证书身份从未与其预期身份进行比较。
- 通知端点和关联标识符被视为仅为其使用的 NWDAF 所知；一个获知另一消费者 notificURI 和 notifCorrId 的合法 NF 未被建模，且规范不要求此类保密性。
- 订阅记录从未从模型的存储中移除，因此关于终止后会发生什么的任何主张，都无法以目标所要求的意义得到定论；Z04、S05 和 U03 仅由较弱的解读来回答。
- OAuth2 令牌是一个原子值，要么被接受要么不被接受：TS 29.510 和 TS 33.501 的受众、有效期以及范围到资源的语义未被建模，且没有令牌被盗或被重放。
- 应用错误、静音、存储处理、待处理通知和重定向未被建模；仅建模了简单的四次交换正常流程以及合法 NF 自身的请求。
- UE 同意、个人数据、ADRF、DCCF、MFAF、多源收集以及处理功能超出任务设定的范围。
- 符号模型：无计算性主张，无算法强度，无抗量子性。

宿主记录的内容：

- 没带对照查询的证明（附录 C）：A02: a02, A03: a03, A04: a04, D01: d01, A08: a08, K01: k01, Z01: z01, Z01: z01b, Z04: term_notif, S05: term_data, U01: u01, U03: term_notif。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。

每条性质仍有哪些开放之处，以及什么能使其定论，见附录 A。

附录 A：逐条性质的解读

这是分析者自己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，该解读依据什么，它能否按原样使用，以及它的价值如何。

A02: 通知发送方是 NWDAF

每当消费者接受一条来自 NWDAF 的通知或订阅响应，就存在一个真实的 NWDAF 运行，其意图身份与消费者所认为的对端身份一致。

消费者所接受的每一条通知都来自该订阅对应的 NWDAF，并带有消费者所选择的关联器。消费者只检查关联器；发送方身份来自 TLS，且不要求与其所订阅的 NWDAF 进行比较。

- 结果：Holds；回答了所问：是；可在有保留的情况下使用
- 价值低：它源自端点的建模方式，而非一条明文规定的规则。
- 文稿中是否规定：否
- 依据：通知端点和关联器被建模为对其他 NF 未知，而规范并不要求这一点。
- 开放：一个获知另一消费者 notificURI 的合法 NF 未被覆盖。

A03: 关联器将通知绑定到订阅

每当消费者接受一条通知或 Fetch 响应，就存在一个匹配的 NWDAF 运行，其角色、双方身份以及 notifCorrId、fetchCorrIds、subscriptionId 等关联字段与消费者本地值一致。

被接受的 fetch 响应恰好携带该 NWDAF 为该订阅产生的关联器和载荷，且消费者在接受前会检查它们。关联在消费者一侧是可靠的，但这并不使这些标识符成为授权数据。

- 结果：Holds；回答了所问：是；可在有保留的情况下使用
- 价值低：这是端点建模的结果，而非规范规则的结果。
- 文稿中是否规定：否
- 依据：fetch 端点是 NWDAF 创建并在通知中下发的通道。
- 开放：一个获知 fetchCorrIds 的合法 NF 未被建模。

A04: 每次 NWDAF 运行仅接受一次

消费者每次接受订阅响应或通知，都对应一个不同的 NWDAF 运行；不存在两次接受映射到同一 NWDAF 运行的情况。

每次接受订阅响应都对应一次不同的 NWDAF 发送，因此被重放的消息无法产生第二次接受。这来自每次交换使用的新鲜连接，而非消息中的新鲜度值。

- 结果：Holds；回答了所问：是；可在有保留的情况下使用
- 价值低：单射性来自私有通道模型。
- 文稿中是否规定：否
- 依据：传输是线性的私有通道；规范未对这类消息规定重放保护。
- 开放：HTTP 层面的重放未被建模。

D01: 通知载荷来自 NWDAF

每次消费者接受通知数据，都存在允许发送方 NWDAF 的真实发送，且载荷与获批的关联字段一致。

消费者通过 fetch 接受的数据就是该 NWDAF 为该订阅产生的数据，路径上不可能被替换为其他东西。保护仅来自传输本身。

- 结果：Holds；回答了所问：是；可在有保留的情况下使用
- 价值低：它只是重述了传输假设。
- 文稿中是否规定：否
- 依据：SBI 保护被建模为私有通道（s0347）。
- 开放：通知正文在传输之外的完整性未被建模。

A08: 通知字段与订阅匹配

消费者接受的通知与 Fetch 响应，其 notificURI、notifCorrId、fetchUri、fetchCorrIds 等字段对应消费者在订阅中提出的值，且属于当前安全上下文。

消费者所见的关联器始终是其自己提出的那个，且通知到达的是它为该订阅创建的端点，因此被接受通知的字段属于其自己的订阅。

- 结果：Holds；回答了所问：是；可在有保留的情况下使用
- 价值低：它源自端点的建模方式。
- 文稿中是否规定：否
- 依据：通知端点是消费者创建并在订阅请求中传递的通道。
- 开放：投递到错误端点的情况不在模型范围内。

K01: 访问令牌与 SBI 密钥保密

攻击者不能推导出消费者的 OAuth2 访问令牌或 NWDAF 与消费者之间的 SBI 会话密钥。

消费者的令牌及其连接的会话密钥永远不会到达攻击者，交换过程中也没有任何东西能推导出它们。该安排是 OAuth2 客户端凭据方式，且 API 流程没有增加通往令牌的路径。

- 结果：Holds；回答了所问：是；可在有保留的情况下使用
- 价值低：该结果就是规范已经做出的假设。
- 文稿中是否规定：是。文稿中的出处：rfc6749:2859, ts-33-501:2885
- 依据：令牌是一个原子值；TLS 密钥仅通过连接是私有的这一事实来体现。
- 开放：令牌被盗、重放和有效期未被建模。

Z01: 每个操作均被授权

每次 NWDAF 提交订阅创建/更新/删除或返回 Fetch 数据，都存在一个由授权服务器签发的、对请求方有效且覆盖 nnwdaf-datamanagement scope 的许可。

每一次被服务的删除和每一次被服务的 fetch 之前，都有一个 NWDAF 为该服务范围接受的令牌，因此没有任何服务是在未授权的情况下提供的。由于一个范围覆盖整个服务，被接受的令牌只能表明调用者是某个已授权的 NF，而不能表明它可以触碰哪些记录。

- 结果：Holds；回答了所问：是；可在有保留的情况下使用

- 价值中：它确定了服务被授权的确切含义，也是 Z03 和 U02 失败所依赖的前提；该事实本身已在文稿中。
- 文稿中是否规定：是。文稿中的出处：ts-33-501:2885, main:3460
- 依据：令牌验证是一个原子检查；受众、有效期以及令牌与身份的绑定不在模型范围内。
- 开放：该范围是否应当覆盖外部资源，正是 Z03 和 U02 所回答的问题。

Z03: 操作主体为订阅创建者

每次对某 subscriptionId 的更新、删除或 Fetch，其授权主体与该订阅的创建者一致，且操作与令牌受众、资源相符。

任何出示服务范围令牌并知道 subscriptionId 的 NF 都能得到 DELETE 服务；创建者从未与调用者进行比较。仅针对网络而言，没有任何删除是可以达成的，因此该弱点的范围恰好与资源标识符的可获知程度一样大。

- 结果：不成立（1 个攻击）；是否回答了所问：部分；带保留可用
- 价值 中：服务级范围且没有资源检查，使合法 NF 能够删除同事的订阅；文稿说明了范围，但未说明这一后果。
- 文稿中是否说明：不明确。所依赖的原文：s0115、s0116、s0117、s0226、s0230、s0341、s0350
- 依赖：subscriptionId 被视为可获取：URI 中的普通字符串，未声明机密性，也没有资源级范围。
- 待定：攻击者对外部 subscriptionId 的了解是一项已声明能力，并非源自协议内部的泄露。；在范围之上额外加入所有权检查的实现未被建模。

Z04: 终止后无通知

在订阅被删除或 NWDAF 请求终止之后，针对该 subscriptionId 不再产生新的通知或 Fetch 数据。

记录的答案只表明，在终止前后被接受的通知是该订阅的真实 NWDAF 通知。NWDAF 在应答删除之后是否仍能通知，尚未确定。

- 结果：成立；是否回答了所问：部分；不可用
- 价值 无：答案讲的是通知路径，而不是终止之后发生什么，而后者正是目标所问。
- 文稿中是否说明：否。所依赖的原文：s0116、s0117、s0119、s0120、s0140
- 依赖：只展开了正常流程；订阅记录从未被清空。
- 待定：终止之后会发生什么尚未确定；模型中记录从未被移除。；带未发送的已存储事件的 200 OK 应答未被建模。

S05: 旧标识符停止工作

订阅终止生效点之后，使用旧 subscriptionId、notifCorrId 或 fetchUri 不能触发新的通知投递或 Fetch 数据返回。

记录的答案表明，消费者在其取消订阅前后所接受的载荷属于 NWDAF。旧 subscriptionId 或 fetchUri 不再返回数据这一点无法展示，因为模型从不清除它所服务的记录。

- 结果：成立；是否回答了所问：部分；不可用
- 价值 无：它回答的是相邻的主张，而不是目标所提出的主张。
- 文稿中是否说明：否。所依赖的原文：s0116、s0117、s0140
- 依赖：已存储的订阅从未从证明器的表中删除。

- 待定：旧标识符在删除后是否停止工作尚未确定。；端点与关联器的失效未被建模。

U01: 关联器仅服务于其消费者

通知与 Fetch 中的 notifCorrId/fetchCorrIds 只对发起该订阅的消费者有效。

关联标识符只在选择它的消费者中有意义：其他消费者的端点不接受它，NWDAF 也无法将其归因于其他消费者。它只是一个标签，不是凭据，因此它保护消费者，但不保护生产者。

- 结果：成立；是否回答了所问：是；带保留可用
- 价值 低：它由端点建模推出；规范未规定绑定规则。
- 文稿中是否说明：否
- 依赖：每个消费者的通知端点只有该订阅的 NWDAF 知道。
- 待定：消费者故意复用另一消费者的关联器未被建模。

U02: 其他 NF 无法读取或替换数据

恶意合法 NF 不能替换或读取其他订阅的数据。

持有服务范围令牌和另一消费者 subscriptionId 的合法 NF 通过 Fetch 获得该订阅的训练数据，且没有检查它是否创建了该订阅。仅针对网络而言不会提供任何数据，因此资源标识符的保密性就是数据的全部保护。

- 结果：不成立（1 个攻击）；是否回答了所问：部分；带保留可用
- 价值 中：缺少资源级授权使合法 NF 能够读取另一消费者的训练数据；文稿说明了范围设计，而非这一损失。
- 文稿中是否说明：不明确。所依赖的原文：s0156、s0157、s0158、s0159、s0230、s0341、s0350
- 依赖：subscriptionId 被视为可获取（s0226、s0230、s0350）。；如 4.4.2.5.2 所要求，NWDAF 服务于其持有的 subscriptionId 对应的任何 fetch。
- 待定：合法 NF 如何获得外部 subscriptionId 是假设的，并非推导得出。；该数据在运营商策略中是否敏感属于部署问题。

U03: 取消订阅后无通知

取消订阅生效后，消费者不再接受该订阅的通知。

在正常流程中，消费者接受通知、执行 fetch、删除并结束，因此取消订阅之后没有后续动作；记录的答案只补充说明，该时刻前后的通知仍是 NWDAF 的。在途通知之后是否仍会被接受，没有得到回答。

- 结果：成立；是否回答了所问：部分；不可用
- 价值 无：它并未确立目标所提出的主张。
- 文稿中是否说明：否。所依赖的原文：s0116、s0117、s0119、s0120
- 依赖：每个订阅一条通知，消费者流程是线性的。
- 待定：取消订阅后不再接受通知，需要对删除与投递之间建立顺序，而模型未构建该顺序。

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，以及各攻击及分析者对其的判定。

目标	级别	界	查询	对照	攻击	主张
A02	proved		a02			2 proved
A03	proved		a03			2 proved
A04	proved		a04			2 proved
D01	proved		d01			2 proved
A08	proved		a08			2 proved
K01	proved		k01			2 proved
Z01	proved		z01			4 proved
Z03	proved		z03_net	checked	z03 (holds_in_source)	1 proved, 1 attack
Z04	proved		term_notif			2 proved
S05	proved		term_data			2 proved
U01	proved		u01			2 proved
U02	proved		u02_net	checked	u02 (holds_in_source)	1 proved, 1 attack
U03	proved		term_notif			2 proved

逐条主张，按分析者对每条主张的措辞；每条的答案是证明器的：

目标	主张	攻击者	回答	查询	对照
A02	只要消费者接受了一条通知，就说明某个真实的 NWDAF 运行针对同一订阅和同一 notifCorrId 发送了它。	`network`	proved	a02	none
A02	只要消费者接受了一条通知，就说明某个真实的 NWDAF 运行针对同一订阅和同一 notifCorrId 发送了它，即使另一个合法 NF 知道该消费者的 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	a02	none
A03	被接受的 fetch 响应携带的正是 NWDAF 为该订阅发送的 fetchCorrId 和载荷。	`network`	proved	a03	none
A03	被接受的 fetch 响应携带的正是 NWDAF 为该订阅发送的 fetchCorrId 和载荷，即使另一个合法 NF 知道该消费者的 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	a03	none
A04	每一次接受订阅响应，都对应 NWDAF 一次不同的订阅响应发送；重放一次 NWDAF 发送不能产生两次接受。	`network`	proved	a04	none
A04	每一次接受订阅响应，都对应 NWDAF 一次不同的订阅响应发送，即使另一个合法 NF 知道该消费者的 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	a04	none
D01	消费者通过 fetch 接受的数据、dataNotification/dataReports 是由 NWDAF 为该订阅发送的。	`network`	proved	d01	none

目标	主张	攻击者	回答	查询	对照
D01	消费者通过 fetch 接受的数据、dataNotification/dataReports 是由 NWDAF 为该订阅发送的，即使另一个合法 NF 知道该消费者的 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	d01	none
A08	被接受通知的 notifCorrId 是消费者自己在为该订阅的 subscribe 请求中提出的那个。	`network`	proved	a08	none
A08	被接受通知的 notifCorrId 是消费者自己在为该订阅的 subscribe 请求中提出的那个，即使另一个合法 NF 知道该消费者的 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	a08	none
K01	攻击者无法推导出消费者的 OAuth2 访问令牌以及消费者与 NWDAF 之间的 SBI 会话密钥。	`network`	proved	k01	none
K01	消费者的 OAuth2 访问令牌以及消费者与 NWDAF 之间的 SBI 会话密钥，对持有自身令牌的合法 NF 仍保持保密。	`malicious_nf_resource_id`	proved	k01	none
Z01	只有在请求出示了 NWDAF 为 nnwdaf-datamanagement 服务 scope 接受的令牌之后，才会处理订阅删除。	`network`	proved	z01	none
Z01	只有在请求出示了 NWDAF 为 nnwdaf-datamanagement 服务 scope 接受的令牌之后，才会处理订阅删除；当请求者是持有自身令牌的合法 NF 时同样如此。	`malicious_nf_resource_id`	proved	z01	none
Z01	只有在请求出示了 NWDAF 为 nnwdaf-datamanagement 服务 scope 接受的令牌之后，才会返回 fetch 响应。	`network`	proved	z01b	none
Z01	只有在请求出示了 NWDAF 为 nnwdaf-datamanagement 服务 scope 接受的令牌之后，才会返回 fetch 响应；当请求者是持有自身令牌的合法 NF 时同样如此。	`malicious_nf_resource_id`	proved	z01b	none
Z03	订阅删除只对该订阅的创建者 NF 处理；仅凭网络，攻击者无法触达任何订阅。	`network`	proved	z03_net	checked
Z03	订阅删除只对该订阅的创建者 NF 处理；持有自身服务 scope 令牌的合法 NF 不能删除另一个消费者的订阅。	`malicious_nf_resource_id`	attack (holds_in_source)	z03 (the control of z03_net)	
Z04	消费者为正在终止的订阅所接受的通知，仍然是 NWDAF 为该订阅发送的；已终止订阅的通知不会从别处到达消费者。	`network`	proved	term_notif	none
Z04	消费者为正在终止的订阅所接受的通知，仍然是 NWDAF 为该订阅发送的，即使另一个合法 NF 知道该 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	term_notif	none
S05	消费者为正在终止的订阅所接受的 fetch 载荷，仍然是 NWDAF 为该订阅发送的载荷。	`network`	proved	term_data	none
S05	消费者为正在终止的订阅所接受的 fetch 载荷，仍然是 NWDAF 为该订阅发送的载荷，即使另一个合法 NF 知道该 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	term_data	none
U01	消费者在通知上接受的 notifCorrId 是它自己选定的那个；其他任何消费者都不可能使其被接受。	`network`	proved	u01	none

目标	主张	攻击者	回答	查询	对照
U01	消费者在通知上接受的 notifCorrId 是它自己选定的那个；其他任何消费者，即使知道该 subscriptionId，也不可能使其被接受。	`malicious_nf_resource_id`	proved	u01	none
U02	仅凭网络时，没有其他 NF 能读取某个订阅的数据：NWDAF 为另一个 NF 提供的每次读取，都要求该 NF 知道该 subscriptionId。	`network`	proved	u02_net	checked
U02	持有自身服务 scope 令牌的恶意合法 NF 不能读取或替换另一个 NF 的训练数据。	`malicious_nf_resource_id`	attack (holds_in_source)	u02 (the control of u02_net)	
U03	消费者在其 unsubscribe 前后所接受的通知，仍然是 NWDAF 为该订阅发送的，因此 unsubscribe 响应之后不会出现来自别处的通知。	`network`	proved	term_notif	none
U03	消费者在其 unsubscribe 前后所接受的通知，仍然是 NWDAF 为该订阅发送的，即使另一个合法 NF 知道该 subscriptionId 也是如此。	`malicious_nf_resource_id`	proved	term_notif	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 节的每个结果都是证明器的回答。

ProVerif+ Agent 会话 session-20261114T213347，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步骤：43，证明器 4 s，总耗时 11 min，费用 0.097 USD。

ProVerif：运行 87 次证明，87 次得出结论，0 次无答案，3 s，记录 39 个回答。

分析了什么

协议	NWDAF training data management
版本	19.8.0
分析范围	两 NWDAF 直接 Subscribe/Notify/Fetch/Unsubscribe；固定聚合非个人训练数据、单一订阅，数据处理函数为 opaque，排除 ADRF、DCCF、多源转发、UE consent 分支。
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	来自下列章节的 350 个带编号的原文单元（73904 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	4.4	Nnwdaf_DataManagement Service
main	4.4.1	服务描述
main	4.4.1.1	概述
main	4.4.1.2	服务架构
main	4.4.1.3	网络功能
main	4.4.1.3.1	网络数据分析功能（NWDAF）
main	4.4.1.3.2	NF 服务消费者
main	4.4.2	服务操作
main	4.4.2.1	引言
main	4.4.2.2	Nnwdaf_DataManagement_Subscribe 服务操作
main	4.4.2.2.1	概述
main	4.4.2.2.2	订阅数据通知
main	4.4.2.2.3	更新数据通知订阅
main	4.4.2.3	Nnwdaf_DataManagement_Unsubscribe 服务操作
main	4.4.2.3.1	概述
main	4.4.2.3.2	取消数据通知订阅

文稿	章节	标题
main	4.4.2.4	Nnwdaf_DataManagement_Notify 服务操作
main	4.4.2.4.1	概述
main	4.4.2.4.2	已订阅数据的通知
main	4.4.2.5	Nnwdaf_DataManagement_Fetch 服务操作
main	4.4.2.5.1	概述
main	4.4.2.5.2	从 NWDAF 检索数据
main	5.3	Nnwdaf_DataManagement Service API
main	5.3.1	引言
main	5.3.2	HTTP 的使用
main	5.3.2.1	概述
main	5.3.2.2	HTTP 标准头部
main	5.3.2.2.1	概述
main	5.3.2.2.2	内容类型
main	5.3.2.3	HTTP 自定义头部
main	5.3.3	资源
main	5.3.3.1	资源结构
main	5.3.3.2	资源：NWDAF 数据管理订阅
main	5.3.3.2.1	描述
main	5.3.3.2.2	资源定义
main	5.3.3.2.3	资源标准方法
main	5.3.3.2.3.1	POST
main	5.3.3.2.4	资源自定义操作
main	5.3.3.3	资源：单个 NWDAF 数据管理订阅
main	5.3.3.3.1	描述
main	5.3.3.3.2	资源定义
main	5.3.3.3.3	资源标准方法
main	5.3.3.3.3.1	PUT
main	5.3.3.3.3.2	DELETE
main	5.3.3.3.4	资源自定义操作
main	5.3.4	无关联资源的自定义操作
main	5.3.5	通知
main	5.3.5.1	概述
main	5.3.5.2	事件通知
main	5.3.5.2.1	描述
main	5.3.5.2.2	操作定义
main	5.3.5.3	Fetch 通知

文稿	章节	标题
main	5.3.5.3.1	描述
main	5.3.5.3.2	目标 URI
main	5.3.5.3.3	标准方法
main	5.3.5.3.3.1	POST
main	5.3.6	数据模型
main	5.3.6.1	概述
main	5.3.6.2	结构化数据类型
main	5.3.6.2.1	引言
main	5.3.6.2.2	类型 NnwdafDataManagementSubsc
main	5.3.6.2.3	类型 NnwdafDataManagementNotif
main	5.3.6.3	简单数据类型与枚举
main	5.3.6.3.1	引言
main	5.3.6.3.2	简单数据类型
main	5.3.6.3.3	枚举：PendingNotificationCause
main	5.3.7	错误处理
main	5.3.7.1	概述
main	5.3.7.2	协议错误
main	5.3.7.3	应用错误
main	5.3.8	特性协商
main	5.3.9	安全

源文包步骤在文稿中定位的协议目的：两 NWDAF 直接 Subscribe/Notify/Fetch/Unsubscribe；固定聚合非个人训练数据、单一订阅，数据处理函数为 opaque，排除 ADRF、DCCF、多源转发、UE consent 分支。

对照文稿定性质

在编写任何查询之前，分析者将草拟的性质清单对照文稿自身所述的协议目的进行了核对。名称、性质所针对的内容以及文稿所述内容均为分析者的解读；所提及的位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A02	通知发送方是 NWDAF	NF 服务消费者确信，它在其 notificURI 上接受的作为通知或订阅响应的 POST 来自它所订阅的 NWDAF，而不是来自另一个 NF。	消费者订阅，在 notificURI 上收到通知，并接受和存储该通知（s0006-s0008、s0125、s0130、s0142）；唯一写明的保护是 SBI 双向认证（s0347、ts-33-501:2871），它认证的是对端 NF，而不是特定的 NWDAF。除此之外，没有写明消费者端对发送方的任何检查。	ts-33-501:2871、s0006、s0007、s0125、s0130、s0142、s0347

ID	简称	对象	文稿怎么说	出处
A03	关联器将通知绑定到订阅	被接受的通知或获取响应携带的是消费者自己订阅的 notifCorrId/fetchCorrIds 和 subscriptionId，而不是另一个消费者的。	请求携带 notificURI 和 notifCorrId（s0053、s0054），NWDAF 分配 subscriptionId（s0079、s0081），通知重复 notifCorrId（s0132），获取使用 fetchCorrIds（s0137、s0156）。没有写明任何生成、唯一性或消费者绑定规则。	s0053、s0054、s0079、s0081、s0132、s0137、s0156
A04	每次 NWDAF 运行仅接受一次	对订阅响应或通知的两次接受不会映射到同一次 NWDAF 运行：重放不会产生第二次接受。	每创建一个订阅就分配一个新的 subscriptionId（s0079、s0081），取消订阅会删除该订阅（s0116、s0117）；notifTimestamp 只是准备时间（s0133）。没有写明这些消息的任何重放保护。	s0079、s0081、s0116、s0117、s0133
D01	通知载荷来自 NWDAF	消费者接受的每个 dataNotification/dataReports 值，都是由 NWDAF 针对该订阅和关联标识符发送的。	NWDAF 发送带有 dataNotification/dataReports 的通知（s0125、s0130、s0135、s0136），如果处理成功，消费者就接受它（s0142）；除 SBI 之外，没有写明任何消息级完整性保护（s0347）。	s0125、s0130、s0135、s0136、s0142
A08	通知字段与订阅匹配	消费者接受的通知和获取响应，与其在自己的订阅上下文中提出的 notificURI、notifCorrId、fetchUri 和 fetchCorrIds 相对应。	消费者提出 notificURI、notifCorrId、anaSub/dataSub 和处理指令（s0053-s0057、s0060、s0061）；NWDAF 用该 notifCorrId 向该 URI 发送通知（s0130-s0132），并给出 fetchUri/fetchCorrIds（s0137、s0156）。没有写明任何绑定规则。	s0053、s0054、s0056、s0057、s0060、s0061、s0130、s0132、s0137、s0156
K01	访问令牌与 SBI 密钥保密	消费者的 OAuth2 访问令牌以及消费者与 NWDAF 之间的 SBI 会话密钥对攻击者保密。	访问可使用从 NRF 获取的 OAuth2 客户端凭据进行授权（s0347、s0348），SBI 按 TS 29.500/TS 33.501 的要求受到保护（s0347）；令牌和密钥格式不在所选原文单元范围内。	rfc6749:2859、s0347、s0348
Z01	每个操作均被授权	每次订阅创建、删除和获取响应都由授权服务器签发的令牌覆盖，该令牌对请求 NF 有效，并携带 nnwdaf-datamanagement 范围。	访问可由以 NRF 作为授权服务器的 OAuth2 客户端凭据授权（s0347），在消费服务之前获取令牌（s0348），一个范围 nnwdaf-datamanagement 覆盖整个服务（s0350）。	ts-33-501:2885、s0347、s0348、s0350
Z03	操作主体为订阅创建者	对某个 subscriptionId 执行 DELETE 或在其下执行 Fetch 的主体，是创建该订阅的 NF；否则 NWDAF 拒绝该操作。	NWDAF 创建订阅并分配 subscriptionId（s0078、s0079）；取消订阅按 subscriptionId 删除（s0115、s0117），获取按 fetchCorrIds 进行（s0156）。单一服务范围（s0350）和错误表（s0341）都没有写明所有权规则，也没有写明对外来资源的拒绝。	s0078、s0079、s0115、s0117、s0156、s0341、s0350
Z04	终止后无通知	订阅被删除后，该 subscriptionId 的任何后续通知都不会到达消费者。	收到 DELETE 时，NWDAF 删除订阅并应答 204（s0116-s0119）；终止请求可以在通知中指示（s0140）。未发送的已存储事件可以在 DELETE 响应中返回（s0120）；在途通知的时序没有规定。	s0116、s0117、s0119、s0120、s0140
S05	旧标识符停止工作	终止后，旧的 subscriptionId、notifCorrId 或 fetchUri 不再产生通知投递或获取的数据。	删除会移除订阅（s0116、s0117），终止请求可以被指示（s0140）；没有写明关于使这些标识符失效或关于缓存数据的任何规则。	s0116、s0117、s0140

ID	简称	对象	文稿怎么说	出处
U01	关联器仅服务于其消费者	一个消费者选择的 notifCorrId 只对该消费者有意义：另一个消费者既不能使 NWDAF 使用它，也不能接受携带它的通知。	notifCorrId 由消费者选择（s0054），并由 NWDAF 重复（s0132）；fetchCorrIds 由 NWDAF 给出（s0156）。没有写明这些标识符与消费者身份的任何绑定。	s0054、s0132、s0156
U02	其他 NF 无法读取或替换数据	持有自己服务范围令牌的恶意合法 NF，无法读取或替换另一个 NF 订阅的训练数据。	该服务为订阅消费者收集训练数据，并通过通知或获取返回（s0006-s0008、s0125、s0159-s0161）；该 API 只有一个服务级范围（s0350），且没有为外来资源命名任何错误（s0341）。	ts-33-501:2859、s0006、s0007、s0008、s0125、s0159、s0161、s0341、s0350
U03	取消订阅后无通知	消费者的取消订阅完成后，消费者不再接受该订阅的任何通知。	收到 DELETE 时，NWDAF 删除订阅（s0116、s0117），并以 204 或 200 应答，同时返回未发送的已存储事件（s0119、s0120）；没有规则限制通知的数量，也没有规则规定通知相对于 DELETE 响应的顺序。	s0116、s0117、s0119、s0120

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 48 段，没选 350 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	RFC 6749 安全考虑的引言：给实现者的建议，不涉及此 API 的任何性质，也没有攻击者。
P003 至 P006	rfc6749:2790-2823	Z01, `network`	
P007 至 P010	rfc6749:2827-2850	Z03, U02, `malicious_nf_resource_id`	
P011 至 P015	rfc6749:2859-2881	K01, Z01, `network`	
P016 至 P021	rfc6749:2885-2917	不在范围内	刷新令牌；此 API 使用客户端凭据授予，不签发刷新令牌（s0347）。
P022 至 P025	rfc6749:2921-2943	不在范围内	授权码授予；未使用（s0347）。
P026 至 P029	rfc6749:2947-2980	不在范围内	授权码流程中的重定向 URI 操纵；不属于此授予。
P030 至 P033	rfc6749:2984-3003	不在范围内	资源所有者口令凭据授予；未使用（s0347）。
P034 至 P035	rfc6749:3013-3019	K01, `network`	
P036	rfc6749:3023-3028	A02, `network`	
P037 至 P039	rfc6749:3032-3041	K01, Z01, `network`	

段落	出处	定为	不选的原因
P040 至 P042	rfc6749:3045-3066	不在范围内	对资源所有者的钓鱼；客户端凭据授予中没有资源所有者。
P043 至 P048	rfc6749:3070-3110	不在范围内	针对浏览器用户代理的跨站请求伪造；此 NF 到 NF 子流程中没有浏览器。
P049 至 P050	rfc6749:3119-3138	不在范围内	点击劫持；属于浏览器框架问题，不适用于 SBI 请求。
P051 至 P052	rfc6749:3142-3150	A08	
P053 至 P054	rfc6749:3154-3167	不在范围内	开放重定向：此 API 的重定向响应在 SCP 层面（s0122、s0148、s0163），未建模。
P055 至 P062	rfc6749:3170-3210	不在范围内	隐式授予中访问令牌被滥用；此 API 使用客户端凭据，且没有用户代理呈递令牌。
P063	ts-33-501:2491-2491	证明器问不了	降级安全特性：算法强度和特性协商（s0343）无法向符号证明器提问。
P064 至 P072	ts-33-501:2495-2511	不在范围内	归属网络对 UE 和服务网络的认证与授权；此子流程中没有 UE。
P073 至 P125	ts-33-501:2515-2631	不在范围内	接入网和 gNB 密钥、用户面与信令的机密性和完整性，以及用户隐私；本任务排除个人数据、UE 同意和无线密钥。
P126 至 P196	ts-33-501:2637-2805	不在范围内	对 gNB、ng-eNB 及其接口的要求，属于本任务排除的系统部分。
P197 至 P213	ts-33-501:2809-2853	不在范围内	对 SEAF、UDM、AUSF 和 UDR 在用户认证中的要求；此子流程不请求订阅数据（UE 同意已排除）。
P214	ts-33-501:2859-2859	U02, `malicious_nf_resource_id`	
P215 至 P220	ts-33-501:2865-2875	Z01, A02, A03, `network`	
P221 至 P227	ts-33-501:2879-2891	Z01, K01	
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；本任务建模的是不使用 SCP 的直接 NF 到 NF 通信。
P242 至 P254	ts-33-501:2929-2953	A02, D01, K01, `network`	
P255 至 P314	ts-33-501:2957-3081	不在范围内	SEPP、漫游中间节点、跨信任边界的属性保护以及 IPUPS：此子流程没有漫游环节，也没有 SEPP。
P315 至 P318	ts-33-501:3085-3091	没有陈述目标	originating-network-id 头及其配置处理：一项运行性的头颅要求，不涉及此子流程的任何性质。
P319 至 P337	ts-33-501:3097-3141	不在范围内	信令流量监控和安全可见性/可配置性：属于运行监控，在子流程之外，无法提问。
P338 至 P369	ts-33-501:3149-3215	证明器问不了	加密和完整性算法标识符及算法选择：计算强度无法向符号证明器提问。
P370 至 P380	ts-33-501:3219-6548	不在范围内	5G-RG、NSSAAF 和 N3 要求：本任务排除的系统部分。
P381 至 P394	ts-33-501:9656-10917	不在范围内	EAP-TLS 用户认证和 IAB 要求：在被分析的子流程之外。

段落	出处	定为	不选的原因
P395 至 P398	ts-33-501:11439-11901	不在范围内	MBSF、MBSTF 和 NEF 要求：这些功能不是此子流程的参与方。

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 27 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 22 项，没设的 5 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	NWDAF	参与方	按定义全部失守	不诚实的 NWDAF 自行生成并递交数据；此处所问的每条性质按定义都不成立，而且文稿在生产者不诚实的情况下没有提出其中任何一条。
S02	NF 服务消费者	参与方	`malicious_nf_resource_id`	
S03	NRF	参与方	`malicious_nf_resource_id`	
S04	消费者 C	参与方	`malicious_nf_resource_id`	
S05	其他消费者 C'	参与方	`malicious_nf_resource_id`	
S06	恶意消费者 C'	参与方	`malicious_nf_resource_id`	
S07	订阅资源与 subscriptionId	值	`malicious_nf_resource_id`	
S08	通知内容	值	`malicious_nf_resource_id`	
S09	notifCorrId 与 notificURI 绑定	值	`malicious_nf_resource_id`	
S10	notifCorrId	值	`malicious_nf_resource_id`	
S11	fetchCorrIds	值	`malicious_nf_resource_id`	
S12	subscriptionId	值	`malicious_nf_resource_id`	
S13	通知数据内容	值	`malicious_nf_resource_id`	
S14	notifTimestamp	值	攻击者得不到更多	notifTimestamp 只是 NWDAF 准备数据的时间（s0133）；没有任何目标依赖其保密性。
S15	dataNotification	值	`malicious_nf_resource_id`	
S16	dataReports	值	`malicious_nf_resource_id`	
S17	notificURI	值	`malicious_nf_resource_id`	
S18	fetchUri	值	`malicious_nf_resource_id`	
S19	dataSub/anaSub	值	`malicious_nf_resource_id`	
S20	OAuth2 访问令牌	值	按定义全部失守	令牌持有者按定义被授权访问单一服务范围；令牌保密性另行提问（K01），在给定令牌下不提出任何目标。
S21	SBI 会话密钥	值	按定义全部失守	持有某个消费者 TLS 会话的会话密钥后，攻击者在那个会话中就是该消费者；该会话的性质按定义都不成立，而且文稿在其下没有提出其中任何一条。
S22	订阅资源	值	`malicious_nf_resource_id`	
S23	采集数据与汇总数据	值	`malicious_nf_resource_id`	

ID	参与方或值	类别	攻击者条件	不设的原因
S24	OAuth2 令牌	值	按定义全部失守	同 S20：令牌持有者按定义被授权访问服务范围。
S25	已存储未发送事件	值	malicious_nf_resource_id	
S26	通知数据	值	malicious_nf_resource_id	
S27	订阅数据	值	malicious_nf_resource_id	

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定好；每个答案都记录在其中一个攻击者之下。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”表示该性质设定了一个攻击者，但证明器让某条主张悬而未决，或者没有记录任何内容；空单元格表示分析者没有为该性质设定该攻击者。

攻击者	持有什么	依据	出处
network	仅网络：它能读取、丢弃、修改和发送消息，不持有任何参与方的秘密		
malicious_nf_resource_id	一个合法的 NF 及其自己的凭据：它自己的访问令牌，带有单一服务范围 nnwdaf-datamanagement（s0350），它自己到 NWDAF 的 TLS 会话，并且知道另一个 NF 的 subscriptionId（资源 URI 中的普通字符串，s0226/s0230）。	s0350 为整个服务给出一个范围，在资源或操作层面没有范围；s0230 使 subscriptionId 成为标识订阅的字符串；s0341 没有为外来资源列出任何应用错误。没有任何条款声明 subscriptionId 或 fetchCorrIds 保密。	s0350, s0230, s0226, s0341

ID	network	malicious_nf_resource_id
A02	成立	成立
A03	成立	成立
A04	成立	成立
D01	成立	成立
A08	成立	成立
K01	成立	成立
Z01	成立	成立
Z03	成立	不成立
Z04	成立	成立
S05	成立	成立
U01	成立	成立
U02	成立	不成立
U03	成立	成立

原文和目标

出处：350 个原文单元，73904 字节，冻结在 source-delivery-01.json 中。 目标：13 个已批准（A02、A03、A04、D01、A08、K01、Z01、Z03、Z04、S05、U01、U02、U03）；提出的问题：3 个。 目标清单按草稿事先获得批准，使用 --approve-targets。

宿主按其文档化默认值（任务 `decisions: auto`）做出的决定，供你核对：

- approve_package：批准源文包 protocol-dc1ce5568bcfec991ad738d2901c

本步骤的自报结果：1 次请求（1 次失败），0.000 美元，输入 48677 个 token，输出 5240 个，256 秒。

关于证明器和模型的说明

分析者为目标点名的内容：

步骤	目标	特性	原因
29	A02, A03, A04, D01, A08, K01, Z01, Z03, Z04, S05, U01, U02, U03	plain_primitives	所有主张都只依赖不透明令牌、会话通道和相关标识符；模型给攻击者的是 TLS 的符号视图（每条连接一个私有通道），除名称相等外没有代数定律，因此没有任何原语被简化到会影响答案的程度。
29	A02, A03, A04, D01, A08, K01, Z01, Z03, U01, U02	many_sessions	这些主张针对任意数量的运行：恶意的 NF、它的订阅以及 fetch/unsubscribe 的副本都被复制，因此 ProVerif 覆盖无界的会话和客户端，而正常流程是任务所界定的单次订阅。

分析者的结束说明

在 model.pv 上（ProVerif，符号模型，任意数量的会话）已确立：证明器判定为 proved——消费者接受的每条通知都是 NWDAF 针对该订阅、用其自己的 notifCorrId 发出的（A02、A08、U01），所获取的载荷携带 NWDAF 签发的相关标识符（A03、D01），每次 NWDAF 发送对应一次接受（A04），消费者的令牌保持机密（K01），每次被服务的 Fetch 和 DELETE 之前都先有一个被接受的服务范围令牌（Z01）。每个目标也都有可达性答案。证明器判定为 attack——一个持有自己令牌、知道某个 subscriptionId 的合法 NF 会被服务以另一个消费者的训练数据（U02），并删除该消费者的订阅（Z03）；这是唯一的发现（新颖性不明确：s0350 只陈述了单一的服务范围，没有条款陈述其后果），而一个经过检验的修复（存储并比较创建者身份）可以消除这两个攻击，同时保持正常流程不变。尚未定论：Z04、S05 和 U03 只能由较弱的解读来回答，因为证明器的表在这里无法清空，所以 unsubscribe 之后的一切都无法判定；而 A02、A03、A04、D01、A08、U01 之所以成立，只是因为模型保留了来自其他 NF 的通知端点和相关标识符，而规范从未要求这一点。阻碍其余部分的是：一个针对终止后主张的订阅记录有状态模型（Tamarin+ Agent），以及一个针对发送者身份主张的、让合法 NF 获知另一个消费者的 notificURI 和 notifCorrId 的模型。