NWDAF 模型提供的安全性分析：Subscribe → Notify → Unsubscribe

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	10 条：3 条成立，7 条不成立（8 个攻击）
发现	4

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

5G 核心网的 Nnwdaf_MLModelProvision 服务允许 NF 服务消费者——包含分析功能的 NWDAF、包含模型训练功能的另一 NWDAF，或 LMF——向包含模型训练逻辑功能的 NWDAF 订阅，并获得已训练 ML 模型文件的发布位置。消费者发送订阅，指明其关注的事件以及接收通知的 URI；生产者返回所创建订阅的标识符，随后发布携带该标识符、事件和模型文件地址的通知。

交换结束后，消费者持有订阅标识符以及每个已订阅事件的模型地址，生产者持有 notifUri 和需要通知的事件。消费者唯一的保证是通知中重复了此前获得的订阅标识符；流程中没有任何内容将通知与生产者绑定，而该 API 的授权是单一范围，覆盖所有订阅和所有操作。

涉及的各方：

- **NF 服务消费者**：包含 AnLF 的 NWDAF、包含 MTLF 的 NWDAF，或 LMF。持有：身份、notifUri、OAuth2 令牌。
- **包含 MTLF 的 NWDAF**：模型生产者。持有：已训练的 ML 模型文件、NRF 签名密钥。
- **NRF**：授权服务器。持有：客户端注册信息、nnwdaf-mlmodelprovision 范围。

图 1 展示了文稿所规定的交换流程。

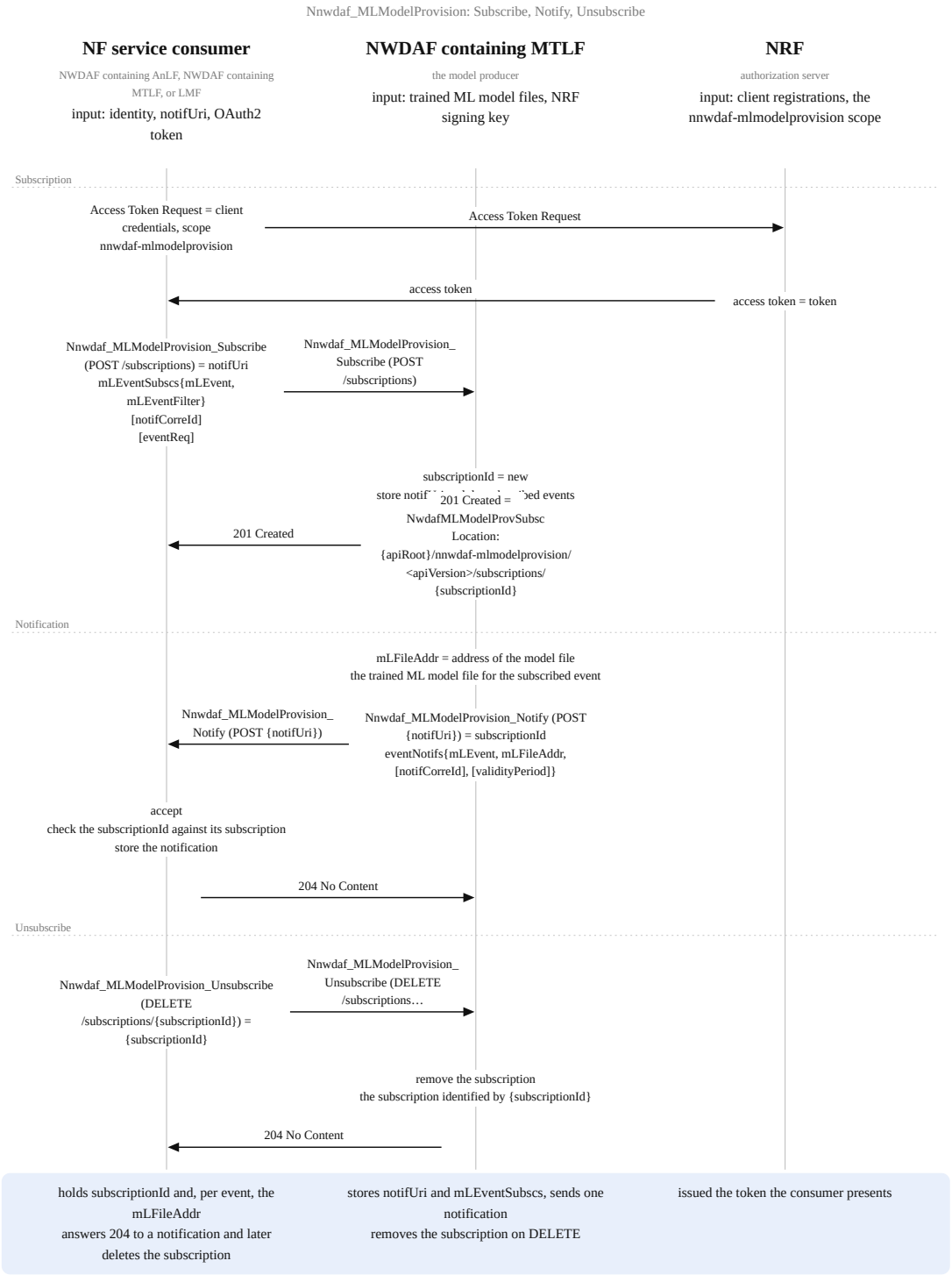


图 1. 文稿所规定的协议。

分析的范围见附录 D：分析了什么、遗漏了什么，以及它依据的文稿章节。

2 发现

分析得到 4 个发现：以下内容在文稿中并未写明。其中 1 项无法确定文稿是否写明。

该 API 并未提供流程读者会假设的保证：通知不携带来源校验、不携带新鲜度值，也不对其载荷做任何绑定，因此服务化接口上的任意 NF 都能投递一个消费者会接受的通知，其中携带自己的模型地址；能重放已录制的通知；或使某条通知在订阅被删除之后仍被存储。授权是单一的服务级范围，因此任何已授权的 NF

都能删除其他消费者的订阅。这两个核心弱点均由 ProVerif 证明，且各有一项经过检查的修改能够消除它。在十条性质中，有四条对网络实际赋予攻击者的能力不成立，有两条只能通过模型未受保护的响应通道达成，有一条只是重述了文稿已有的内容，其余则仅在消费者校验订阅标识符的解读下成立。这些结果基于一个将整个服务化接口都交给攻击者、且不建模 TLS、OAuth2 令牌校验或模型文件本身的模型。

攻击图使用图 1 的列。红色表示攻击者的行为。黄色标记某方发送了规定交换在该时点并不包含的消息，以及攻击所破坏的结果。

2.1 SBI 上的任意 NF 都能投递一个消费者会接受的通知，并携带任意模型地址

类型	协议规范本身的弱点
是否新发现	是：文稿里没有写到
价值	高
涉及的性质	A03, A03.9, D01, A08 （见第 3 节）
依据	ProVerif attack queries `a03`, `a08`; Figures 2, 3, 4 and 5

发生了什么。 Notify 操作是一个 HTTP POST，消费者在其 notifUri 上接受它。消费者将 subscriptionId 与其持有的值进行比较，但没有任何内容标识发送方，也没有任何内容绑定事件标识符或 mLFileAddr，因此观察到 subscriptionId 的 NF 可以投递一条通知，携带自选的模型地址、事件标识符和有效期，消费者会存储它并返回 204。模型显示该被接受的通知背后没有任何 MTLF 的发送。

攻击者需要什么。 服务化接口上的一个位置；不需要令牌，也不需要密钥。subscriptionId 可从 201 Created 响应或通知本身读取，notifUri 可从 Subscribe 请求读取。

攻击者能得到什么。 消费者被引导至攻击者选定的 ML 模型，并将其用于分析或定位，尽管包含 MTLF 的 NWDAF 从未发布过该模型。流程只提供地址，因此下游没有任何环节能发现这一替换。

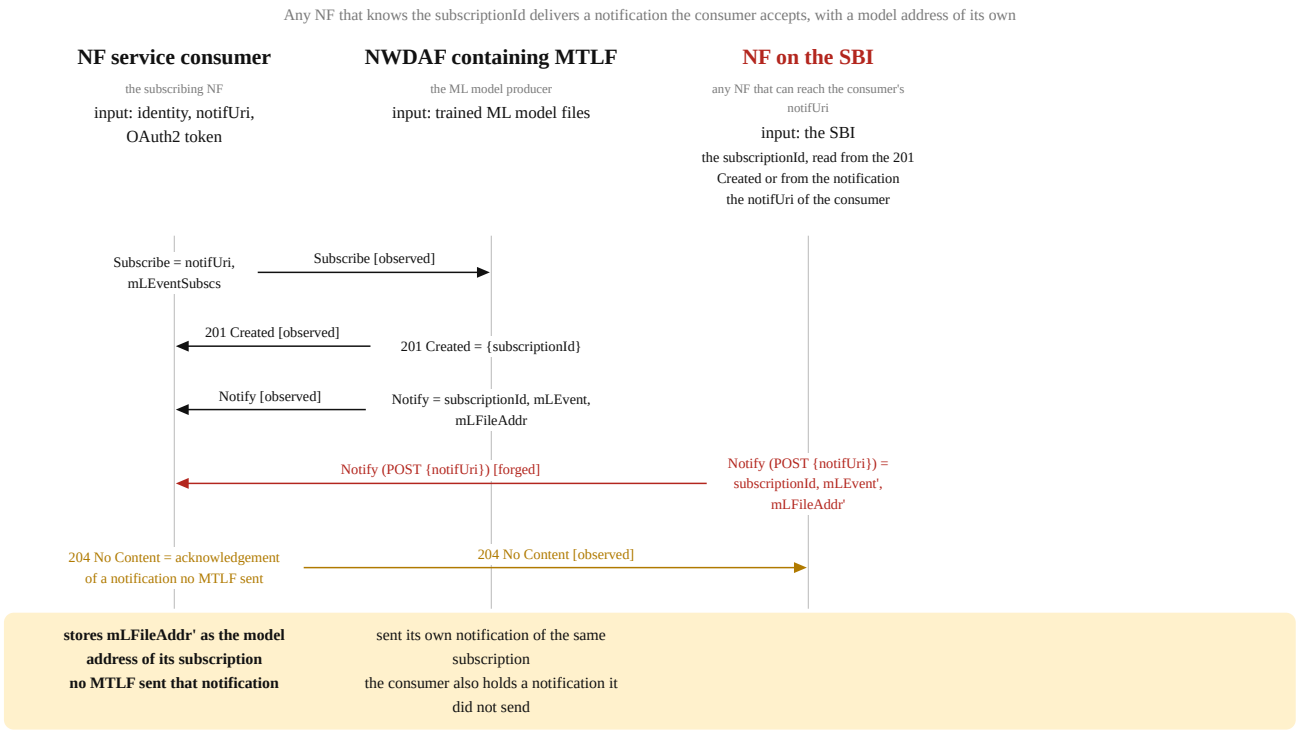


图 2. 任何知道 subscriptionId 的 NF 都能投递一个消费者会接受的通知，并携带自己的模型地址。性质 A03；攻击查询 a03。

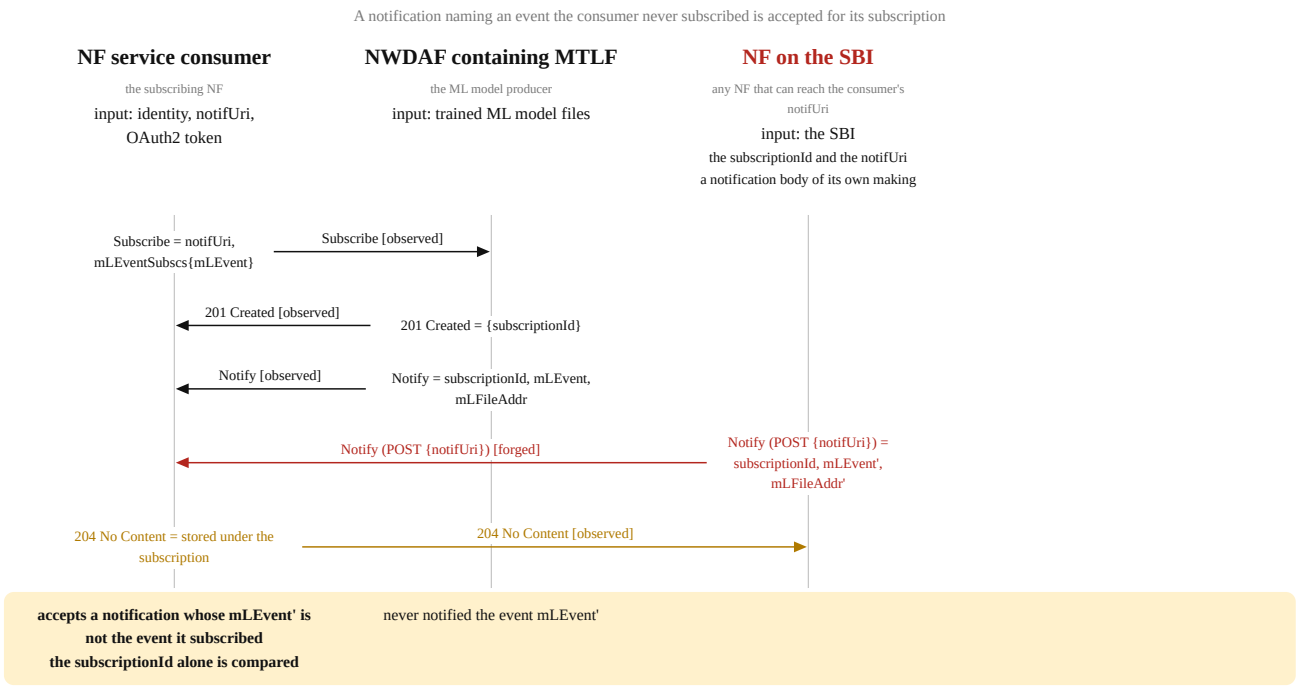


图 3. 一条指明消费者从未订阅的事件的通知，会以其订阅的名义被接受。性质 A08；攻击查询 a08。

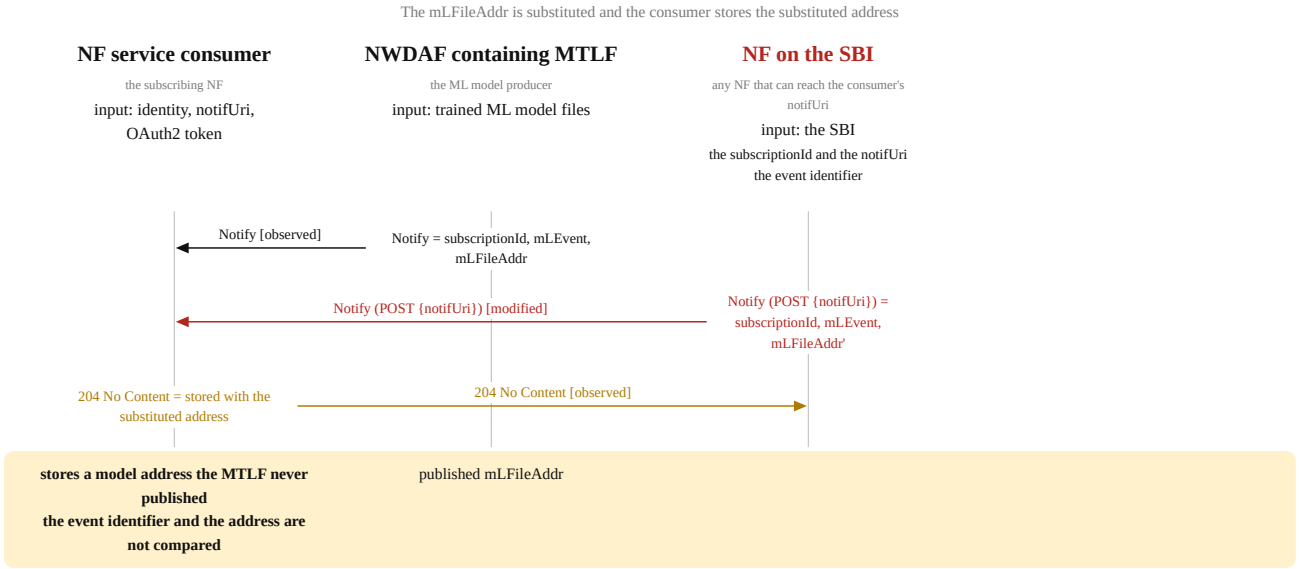


图 4. mLFileAddr 被替换，消费者存储了被替换的地址。性质 D01；攻击查询 a03。

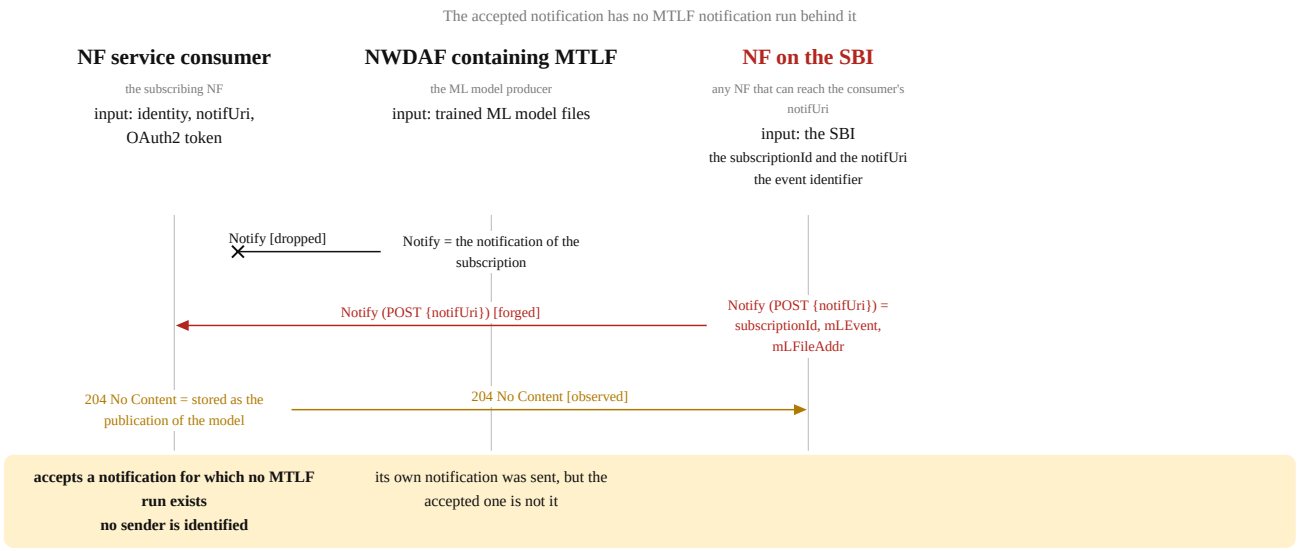


图 5. 被接受的通知背后没有 MTLF 的通知运行。性质 A03.9；攻击查询 a03。

拟议修改。 让包含 MTLF 的 NWDAF 对通知体（subscriptionId、notifUri、事件标识符、mLFileAddr）签名，并让消费者在存储通知并返回 204 之前，用其订阅所在的 MTLF 验证该签名。在该修改的模型上已检查：关于被接受的通知与 MTLF 发送之间对应关系的查询得到验证，正常流程仍可被发现，其他已记录的答案保持不变。只对通知体签名而不包含 notifUri 是不够的：此时校验仍会接受一条本应发给其他接收方的通知。

用 ProVerif 复核。 该修改被放入模型的副本中：Nnwdaf_MLModelProvision_Notify 请求携带由包含 MTLF 的 NWDAF 对 (subscriptionId, notifUri, 事件标识符, mLFileAddr) 的签名，NF 服务消费者在存储通知并返回 204 之前，用其订阅所在的 MTLF 验证该签名，否则拒绝。notifUri 位于签名体之内，因此通知无法被重定向到其他消费者。在修改后的模型上，ProVerif 证明了 `a03`：该攻击在那里不再可能。协议的正常流程仍可被发现。其他 9 个已记录的答案保持不变。这是对模型的初步检查：它表明该修改在模型中消除了这一攻击，而不表明它不会引入其他问题。修改后的模型：`records/fixes/0044-fix-signed.pv`。

2.2 持有服务令牌即可删除任意消费者的订阅

类型	协议规范本身的弱点
是否新发现	不确定：文稿只写了一部分，或没能检索
价值	高
涉及的性质	Z03 （见第 3 节）
依据	ProVerif attack query `z03`; Figure 6

发生了什么。 该 API 为整个服务定义了单一的 OAuth2 范围，没有资源级或操作级的范围，而 DELETE 仅通过路径中的 {subscriptionId} 定位订阅。因此，NWDAF 会为任何出示该服务令牌的 NF 删除订阅，无论该 NF 是否创建了该订阅。模型显示：用某个消费者的令牌创建的订阅，被另一个消费者签发的令牌删除。

攻击者需要什么。 一个针对 nnwdaf-mlmodelprovision 范围的 OAuth2 访问令牌（任何已授权的 NF 服务消费者都持有），以及受害者的 subscriptionId，后者在 Location 头和每条通知中都以明文传输。

攻击者能得到什么。 任何已授权的 NF 都能终止另一消费者的 ML 模型订阅，并随之终止该消费者所依赖的模型更新流。受害者只能得知不再有通知到达。

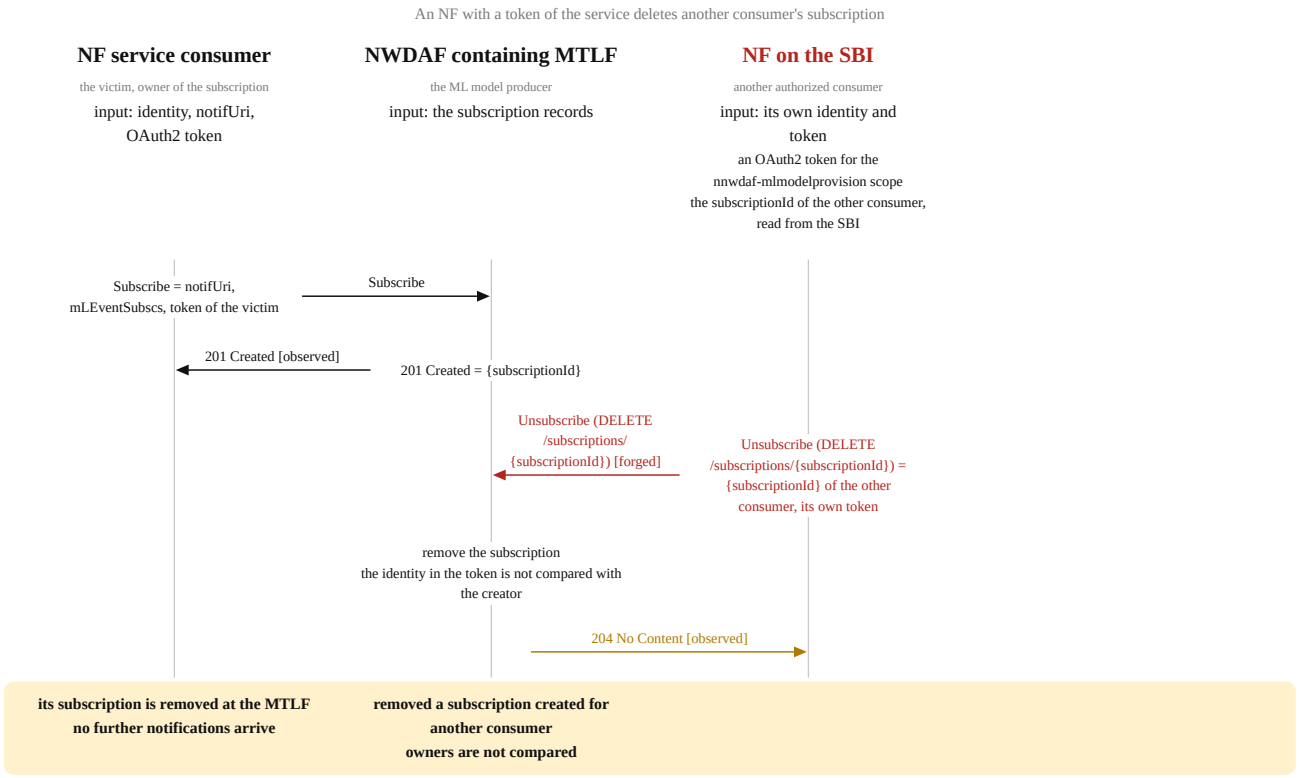


图 6. 持有服务令牌的 NF 删除了另一消费者的订阅。性质 Z03；攻击查询 z03。

拟议修改。 让包含 MTLF 的 NWDAF 针对订阅的所有者对 DELETE 进行授权：将所出示令牌中的 NF 身份与创建该订阅的身份进行比较，不同时返回 403。在该修改的模型上已检查：关于取消订阅仅被订阅所有者接受的查询得到验证，正常流程仍可被发现，其他已记录的答案保持不变。

用 ProVerif 复核。 该拟议修改已放进模型的一个副本中：包含 MTLF 的 NWDAF 针对订阅的所有者来授权 DELETE：它把所出示令牌中的 NF 身份与该订阅的创建者（即发送创建 POST 的那一方）的身份进行比较，而不是接受任何携带 nnwdaf-mlmodelprovision 作用域的令牌。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 normal run 仍能被找到。其余 11 条已记录的答案保持不变。这是对模型的首次检查：它表明该修改在模型中消除了这一攻击，并不表明它不会引入其他问题。修改后的模型：`records/fixes/0035-fix-owner.pv`。

2.3 已记录的通知可被重放并当作新通知接受

类型	协议规范本身的弱点
是否新发现	是：文稿里没有写到
价值	中
涉及的性质	A04 （见第 3 节）
依据	ProVerif attack query `a04`; Figure 7

发生了什么。 NwdafMLModelProvNotif 不携带消费者会检查的序列号、nonce 或时间戳，而 notifCorreId 是可选的，且对同一订阅的每条通知都相同。因此，在 SBI 上记录的一条通知可以被再次 POST 到同一个 notifUri，并被存储为第二条新通知。

攻击者需要什么。 在服务化接口上的一个位置，以及一条在传输中观察到的通知。

攻击者能得到什么。 消费者会反复对同一次模型发布采取行动，它执行的任何由通知驱动的步骤（模型下载、重新训练触发）都会在并非由 MTLF 选定的时间被重复。

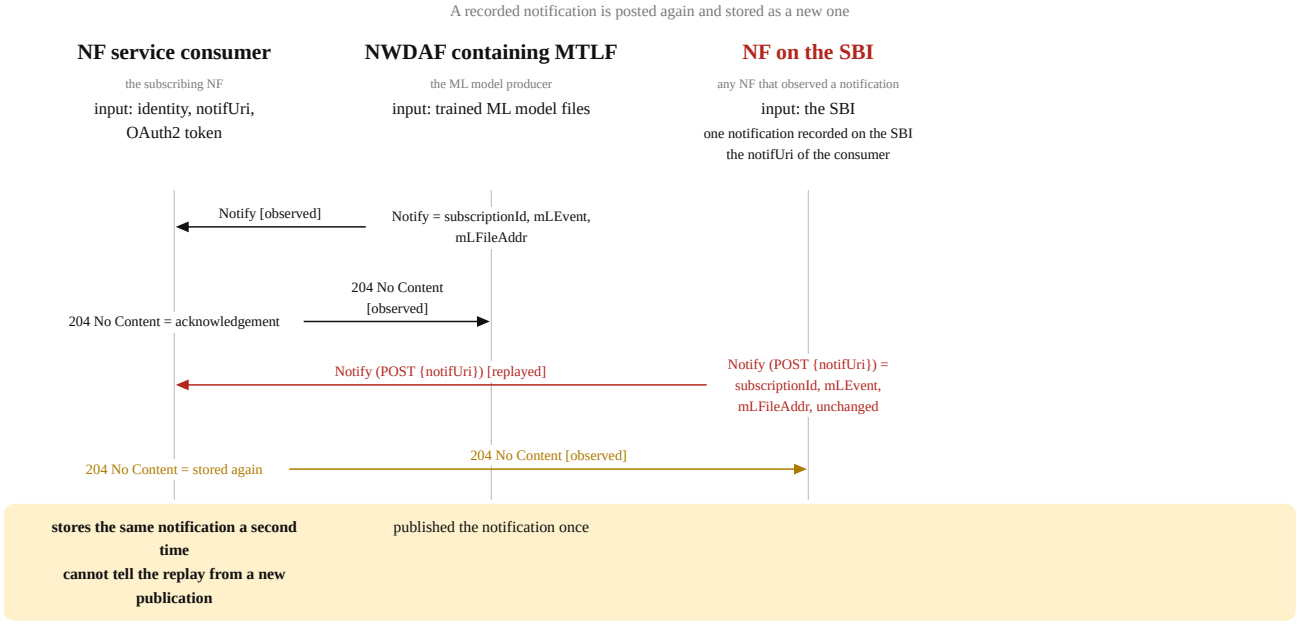


Figure 7. 已记录的通知被再次 POST 并存储为新通知。性质 A04；attack query a04。

拟议修改。 在 NwdafMLModelProvNotif 中加入一个由包含 MTLF 的 NWDAF 为每条通知生成、且对每个订阅唯一的值（通知序列号或 nonce），并让消费者拒绝其值已被接受过的通知。

用 ProVerif 复核。 未检查：该修改向 NwdafMLModelProvNotif 添加了一个新属性，而该属性超出了模型所携带的字段范围；本次分析没有构建并运行它。

2.4 消费者退订后通知仍被存储

类型	协议规范本身的弱点
是否新发现	是：文稿里没有写到
价值	中
涉及的性质	S05 （见第 3 节）
依据	ProVerif attack queries `s05`, `s05b`; Figures 8 and 9

发生了什么。 DELETE 在包含 MTLF 的 NWDAF 处移除订阅，并以 204 确认，但消费者仍保持其 notifUri 开放，且该流程没有给它任何值来拒绝针对已移除订阅的通知。将消费者展开为退订前一条通知和已确认 DELETE 后一条通知时，ProVerif 到达了第二次接受，通知被存储。

攻击者需要什么。 在服务化接口上的一个位置，一条在退订前观察到的通知，以及消费者使用的 subscriptionId。

攻击者能得到什么。 消费者在请求停止接收模型发布之后仍存储并使用这些发布，且双方持有不同的状态：提供者没有订阅，消费者仍依据该标识符行动。

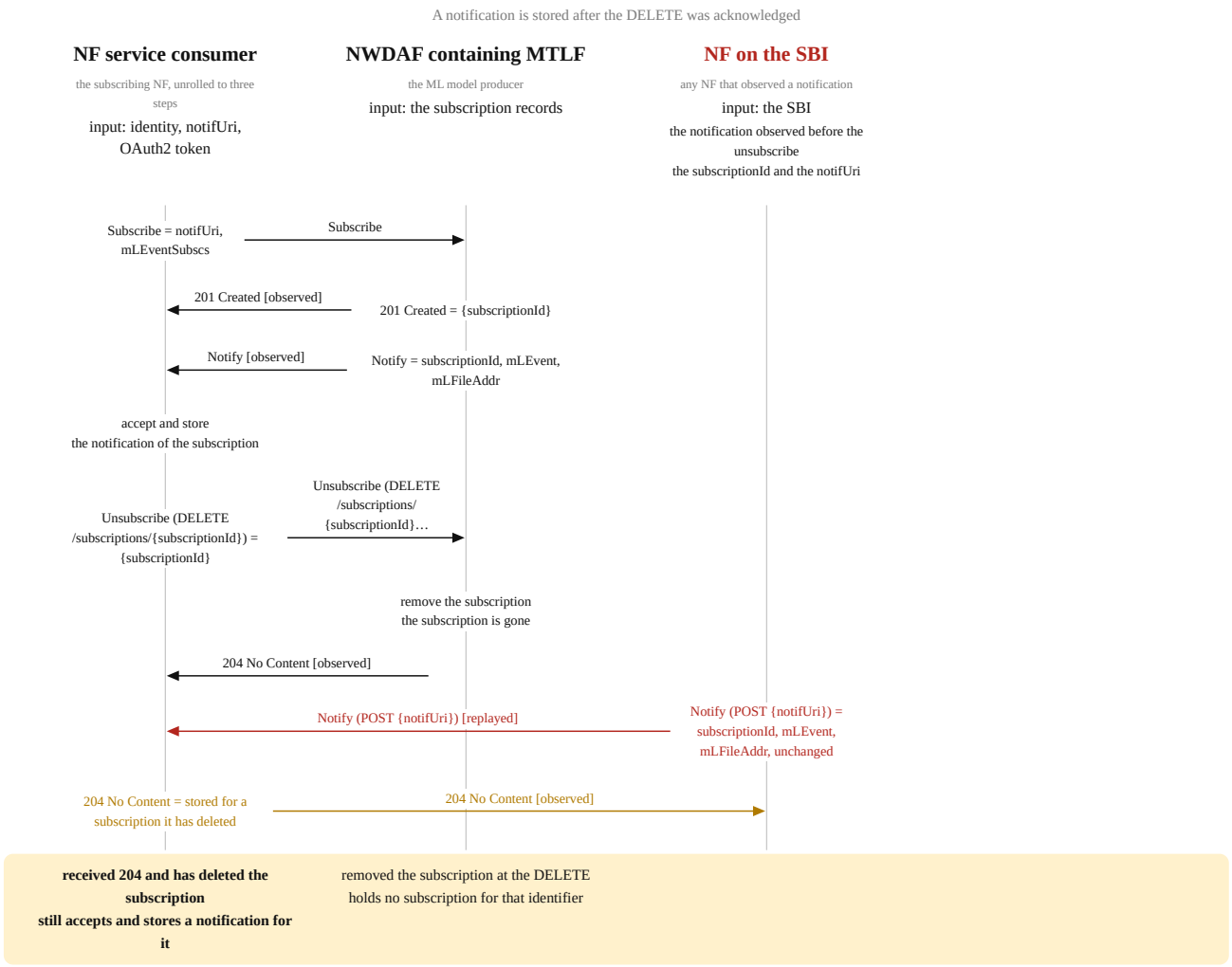


Figure 8. 在 DELETE 被确认后，一条通知被存储。性质 S05；attack query s05b。

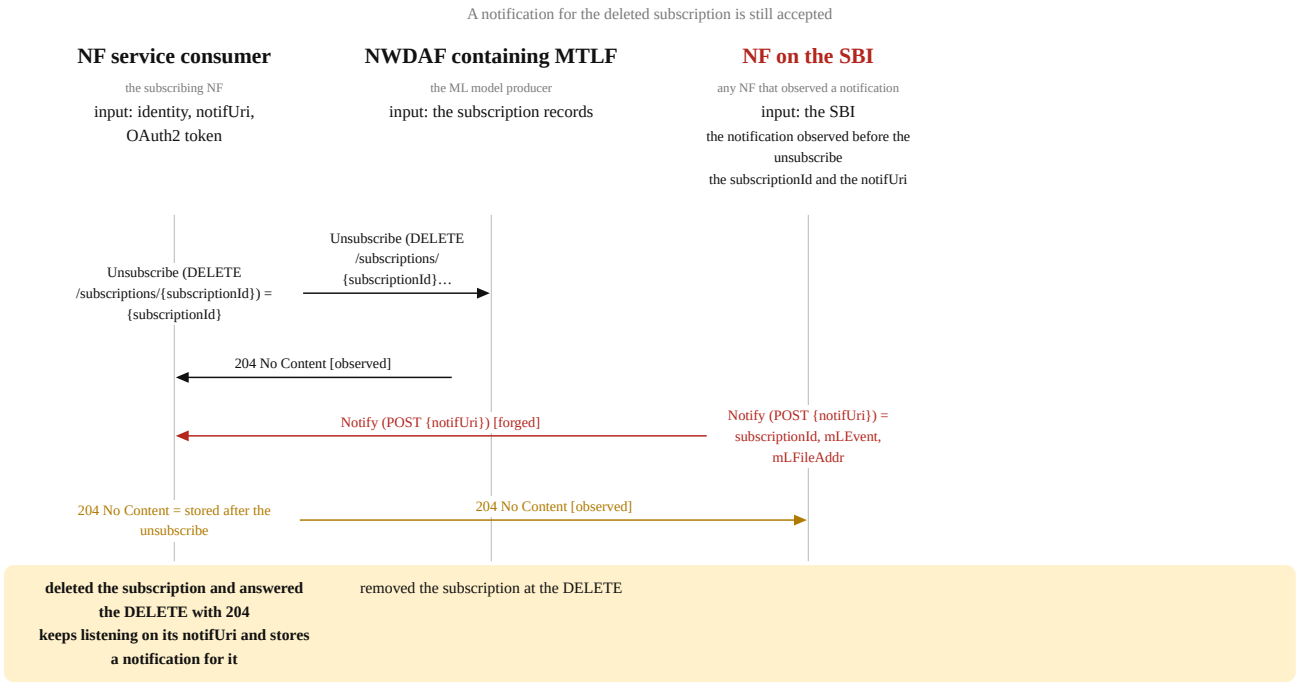


Figure 9. 针对已删除订阅的通知仍被接受。性质 S05；attack query s05。

拟议修改。 让消费者保留其 DELETE 已被确认的订阅的 subscriptionId，并拒绝任何携带该值的通知，返回错误而不是 204。

用 ProVerif 复核。 未检查：该修改向消费者添加了一个保留值和一个检查；有界模型已经展开了消费者，但本次分析没有构建并运行该变体。

对于某项发现的这些攻击，没有给出图：A03.9: a08, A03: a08, A08: a03, D01: a08。

3 检查过的性质

每一行都是一条提交给 ProVerif 的性质。结果是 ProVerif 对分析者的协议模型给出的答案；价值以及文稿是否已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	**通知与提供商发送匹配。** 消费者每次接受一条通知，都存在一个匹配的提供者运行：该运行由消费者此前的订阅请求触发，且通知中的 subscriptionId、事件标识与消费者在订阅中给出的值一致。	无	不成立（1 个攻击）	高	否
A04	**通知不被接受两次。** 消费者对同一订阅的每次通知接受，都对应提供者一次不同的通知发送运行，同一通知不能被重复接受为新的通知。	无	不成立（1 个攻击）	中	否
A08	**通知字段匹配自身订阅。** 消费者接受的通知，其 subscriptionId、事件标识与 mLFileAddr 等字段对应消费者在订阅中给出的 notifUri 与 mLEventSubscs，且未被替换为其他订阅或其他事件的值。	无	不成立（1 个攻击）	中	否
D01	**通知载荷是真实的。** 消费者接受的每条通知都对应所订阅提供者的一次真实发送，通知载荷（subscriptionId、eventNotifs、mLFileAddr）与发送时一致。	无	不成立（1 个攻击）	高	否
Z01	**操作需要有效令牌。** 订阅创建、更新、删除与通知发送等受保护操作，只有在请求方持有授权服务器签发的、覆盖 nnwdaf-mlmodelprovision 作用域的有效令牌时才被接受。	无	成立	低	是
Z03	**令牌绑定到其资源。** 针对某一订阅或某一操作获得的授权，不能被用于其他订阅、其他消费者身份或其他受保护操作。	无	不成立（1 个攻击）	高	不明确
S05	**退订后无通知。** 在提供者成功处理退订（删除订阅）之后，消费者不再接受携带该 subscriptionId 的新通知。	无	不成立（2 个攻击）	中	否
A08.8	**通知属于自身订阅。** 消费者接受的通知中，subscriptionId、notifUri 与事件标识对应其自身创建的订阅，而非其他订阅或其他事件。	无	成立	低	不明确
A03.9	**被接受的通知有真实发送。** 消费者每次接受通知，都存在所订阅提供者的一次真实通知运行，且 subscriptionId 与事件一致。	无	不成立（1 个攻击）	中	否
U01	**失败订阅无通知。** 订阅创建失败后，消费者不接受携带该 subscriptionId 的通知；接受的通知都对应成功的订阅创建。	无	成立	低	否

所提的问题及分析者的回答：

问题	回答	依据
Q1: subscriptionId 与创建者、notifUri、订阅事件的关联	消费者确实将通知绑定到它自己持有的订阅标识符，这部分对任意数量的会话都成立。除此之外，这种关联不受保护：服务化接口上的一个 NF 可以让消费者相信它创建了一个实际上从未创建的订阅，并可以为其发送通知；与订阅不同的事件标识或模型地址也会被接受。	A08.8, A08, A03

问题	回答	依据
Q2: 接受通知对应本订阅的合法提供者发送事件	否。通知是发往消费者 notifUri 的 HTTP POST，流程不携带来源检查，因此消费者接受的通知根本不必由 MTLF 发送；已记录的通知还可以被重放为新通知。	A03.9, A03, A04
Q3: 区分订阅创建失败与成功后通知的事件路径	失败响应 403 与 500 是区分开的，诚实的生产者在这之后不发送通知，模型确认消费者只对它在创建响应中收到的订阅标识符接受通知。模型无法确定的是该响应是否真的来自 MTLF，因为它把请求和响应放在同一条不受保护的通道上。	U01, A08.8

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在密码学视为理想的情况下给出了什么、未能给出什么，而不涉及任何实现。

正如分析者所述，本次分析不覆盖以下内容：

- 服务化接口被建模为攻击者拥有的一条通道。传输保护（TLS，TS 29.500 第 13.1 条）在被分析的文本之外，未被建模；需要攻击者应答消费者自己发出的请求（伪造的 201 Created）的攻击是模型的产物，并被如此标记。
- 模型没有 NF 身份，NRF 向任何在服务化接口上请求的一方签发令牌，因此在模型中网络攻击者与已授权 NF 是同一个攻击者；答案对两种解读中较弱的一种成立。
- OAuth2 令牌的签发、校验和客户端认证由一个携带身份和单一服务作用域的签名令牌表示；TS 29.510、29.500 和 33.501 的细节未被建模。
- 模型文件本身未被建模：只携带 mLFileAddr。内容、哈希和下载完整性在任务边界之外，由 TS 23.288 第 6.2A.0 条放在别处。
- 只建模了一个消费者与一个包含 MTLF 的 NWDAF 的流程。ADRF、联邦学习、模型共享、定位、漫游、PUT 更新、重定向响应 307 和 308、expiryTime 以及失败的问题详情都被略去。
- 订阅状态是一个会变化的值；model.pv 省略了它的检查，bounded.pv 把消费者展开到三步。那里所示的结果只对这些步骤成立，状态审查记录了这些步骤之外的情况。

宿主记录的内容：

- S05 的答案依赖于模型简化的一项操作（附录 D）。

每条性质仍有哪些未决之处，以及什么能解决它，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答案的解读，而不是证明器的答案：这些答案对协议意味着什么、该解读依据什么、能否按原样使用，以及它有多大价值。

A03: 通知与提供商发送匹配

消费者每次接受一条通知，都存在一个匹配的提供者运行：该运行由消费者此前的订阅请求触发，且通知中的 subscriptionId、事件标识与消费者在订阅中给出的值一致。

消费者接受了某条并非由任何 MTLF 发送的通知。服务化接口上知道 subscriptionId 的 NF 可以向消费者的 notifUri 投递一条带有任意事件标识和任意模型地址的通知，消费者会存储该通知并返回 204。

- 结果：不成立（1 个攻击）；回答了所问：是；可用
- 价值 高：这是整个服务赖以交付的价值的未受保护接受路径；文稿未陈述此类弱点，也未陈述此类检查。
- 文稿中已陈述：否
- 依据：服务化接口是攻击者拥有的一个通道，模型未为其提供传输保护。；消费者被建模为仅将通知中的 subscriptionId 与其持有的 subscriptionId 进行比较，别无其他；文本未说明它比较什么。
- 未决：通知端点是否被要求检查发送方身份尚未说明；文稿依赖传输层，而传输层将对方认证为某个 NF，而非所订阅的 MTLF。
- 下一步：与编辑确认通知端点是否应针对订阅的 MTLF 检查发送方；若传输与网络配置意在完成此事，应在规范中说明。

A04: 通知不被接受两次

消费者对同一订阅的每次通知接受，都对应提供者一次不同的通知发送运行，同一通知不能被重复接受为新的通知。

记录在接口上的通知可以被再次投递，并被存储为新通知，因为消息体不携带消费者用于比较新鲜度的任何值。消费者的每个通知驱动步骤都会在生产方未选择的时间被重复执行。

- 结果：不成立（1 个攻击）；回答了所问：是；可用但有保留
- 价值 中：消息设计中存在真实缺口，已通过机械方式展示；影响取决于消费者如何处理通知，而模型未表示这一点。
- 文稿中已陈述：否
- 依据：攻击者可以读取并重发通知，模型在单一通道上允许这一点。；notifCorreId 在文本中是可选的，按范围要求被建模为缺失。
- 未决：同一模型发布的重复投递对消费者是否可接受尚未说明。
- 下一步：添加每条通知的唯一值及消费者检查，然后重新运行单射性查询。

A08: 通知字段匹配自身订阅

消费者接受的通知，其 subscriptionId、事件标识与 mLFileAddr 等字段对应消费者在订阅中给出的 notifUri 与 mLEventSubscs，且未被替换为其他订阅或其他事件的值。

subscriptionId 将通知绑定到某订阅，但事件标识和模型地址均未与订阅进行核对，因此一条指明不同事件和不同模型地址的通知仍会被该订阅接受。

- 结果：不成立（1 个攻击）；回答了所问：是；可用但有保留
- 价值 中：服务所依赖的绑定只存在一半：标识符被检查，载荷未被检查。
- 文稿中已陈述：否
- 依据：消费者仅比较 subscriptionId；这是文本沉默处的建模选择。；服务化接口在模型中未受保护。
- 未决：文本未说明消费者是否必须将事件标识和 mLFileAddr 与订阅进行比较；模型让其仅比较 subscriptionId。
- 下一步：在规范中说明消费者必须将通知的哪些字段与订阅进行比较，并检查发送方。

D01: 通知载荷是真实的

消费者接受的每条通知都对应所订阅提供者的一次真实发送，通知载荷（subscriptionId、eventNotifs、mLFileAddr）与发送时一致。

通知中的模型地址在到达消费者的途中可以被替换，消费者会存储被替换的值。该流程只提供一个地址，别无其他，因此消费者无法区分由 MTLF 发布的模型地址与由 NF 引入的模型地址。

- 结果：不成立（1 个攻击）；回答了所问：是；可用
- 价值 高：服务存在的目的就是交付此地址；未受保护的替换会将消费者重定向到攻击者选择的模型。
- 文稿中已陈述：否
- 依据：通知上未建模任何完整性保护；文本在 API 层面也未陈述任何保护。；模型文件本身超出范围：仅携带地址。
- 未决：mLFileAddr 是否应在 API 层受到完整性保护尚不明确：TS 23.288 条款 6.2A.0 将模型文件本身的保护交由 TS 33.501 附录 X。
- 下一步：按已检查的建议，为通知消息体和接收方指定签名。

Z01: 操作需要有效令牌

订阅创建、更新、删除与通知发送等受保护操作，只有在请求方持有授权服务器签发的、覆盖 nnwdaf-mlmodelprovision 作用域的有效令牌时才被接受。

仅当请求携带授权服务器为该服务作用域签发的令牌时，生产方才会接受 Subscribe 或 Unsubscribe；不带此类令牌的请求不会被处理。这是该流程唯一的保护，而任何 NF 都能获得此类令牌。

- 结果：成立；回答了所问：是；可用但有保留
- 价值 低：它确认了文稿所述内容（s0339 至 s0342），并完全依赖令牌模型，而这是一种简化。
- 文稿中已陈述：是。文稿所述位置：main:19218、s0339、s0340、s0342
- 依据：NRF 向接口上的任何请求方签发令牌，因此模型不区分已授权 NF 与网络。；令牌被建模为携带身份和作用域的签名值；TS 33.501 的证书或客户端认证未被建模。
- 未决：令牌验证细节（过期、受众、客户端认证）不在所分析的文本范围内。
- 下一步：此主张无需进一步工作；令牌绑定到什么由 Z03 回答。

Z03: 令牌绑定到其资源

针对某一订阅或某一操作获得的授权，不能被用于其他订阅、其他消费者身份或其他受保护操作。

服务令牌可被任何订阅接受：持有令牌的 NF 会删除另一个 NF 创建的订阅。API 没有资源级授权，因此接口上观察到的标识符足以终止另一个消费者的订阅。

- 结果：不成立（1 个攻击）；回答了所问：是；可用

- 价值 高：这是授权设计的弱点，攻击者用网络照例发放的凭据即可触及；文稿陈述了规则，但未陈述后果。
- 文稿中已陈述：不明确
- 依据：攻击者是能够获得令牌的已授权 NF；模型的 NRF 向任何请求方签发。；令牌与 NF 实例的客户端认证和证书绑定未被建模，尽管它们会将令牌绑定到身份而非订阅。
- 未决：单一作用域是否为所有消费者同等受信网络的有意简化尚未说明。
- 下一步：针对 PUT 检查建议中的所有权检查，并针对持有第二个订阅令牌的消费者进行检查。

S05: 退订后无通知

在提供者成功处理退订（删除订阅）之后，消费者不再接受携带该 subscriptionId 的新通知。

DELETE 被确认后，生产方已无该订阅，但消费者仍接受并存储该标识符的通知。双方最终处于不同状态，消费者会依据其已请求停止的模型发布行事。

- 结果：不成立（2 个攻击）；回答了所问：是；可用但有保留
- 价值 中：这是文本未决的状态问题，已在展开运行中展示；影响是取消后仍继续使用模型。
- 文稿中已陈述：否
- 依据：消费者在 bounded.pv 中被展开为三步；该性质针对这些步骤展示。；迟到的通知由攻击者在未受保护接口上投递。
- 未决：文本未说明消费者对已删除订阅的通知如何处理。
- 下一步：确认消费者是否应保留并执行已删除订阅的状态。

A08.8: 通知属于自身订阅

消费者接受的通知中，subscriptionId、notifUri 与事件标识对应其自身创建的订阅，而非其他订阅或其他事件。

消费者仅接受其在创建响应中收到的 subscriptionId 对应的通知，且这对任意数量的会话都成立。但这并不推出该订阅在 MTLF 处存在：创建响应本身未受保护，因此 NF 可以让消费者持有 MTLF 从未签发的标识符。

- 结果：Holds；是否回答了所问：部分；带保留地可用
- 价值低：成立的部分是任何实现对给定标识符的处理方式；不成立的部分是传输层面的产物。
- 文稿中是否说明：不明确
- 依据：模型把请求和响应放在同一信道上，因此应答可以被伪造；在 TS 29.500 的传输保护下，响应被绑定到消费者所寻址的 MTLF。；消费者被建模为只比较 subscriptionId，别无其他。
- 未决：要求 subscriptionId 必须属于 MTLF 所创建订阅的那种解读尚未确定：那里唯一的答案来自模型未受保护的请求和响应这一产物。
- 下一步：说明消费者是否必须将通知与其创建的订阅关联，并保护创建响应。

A03.9：被接受的通知有真实发送

消费者每次接受通知，都存在所订阅提供者的一次真实通知运行，且 subscriptionId 与事件一致。

被接受的通知未必对应 MTLF 的任何通知运行：接口上的某个 NF 为观察到的 subscriptionId 发布通知，消费者便将其接受为其模型的发布。

- 结果：不成立（1 次攻击）；是否回答了所问：是；可用

- 价值中：与 A03 相同的缺失来源检查，这里表述为与真实通知运行的对应关系；影响是接受了从未发生的发布。
- 文稿中是否说明：否
- 依据：模型中的服务化接口未受保护，API 层面不检查发送方身份。
- 未决：是否期望通知端点检查发送方，文稿未说明。
- 下一步：为通知规定来源检查，如 A03 的建议所述。

U01: 失败订阅无通知

订阅创建失败后，消费者不接受携带该 subscriptionId 的通知；接受的通知都对应成功的订阅创建。

消费者只接受其从 201 Created 中收到的订阅标识符对应的通知，因此被它处理为失败的创建永远不会导致被接受的通知；这对任意数量的会话都成立。标识符是否真的属于 MTLF 创建的订阅属于另一种解读，在那里模型只展示了其未受保护响应信道的产物。

- 结果：Holds；是否回答了所问：部分；带保留地可用
- 价值低：已证明的部分是实现对其处理过的创建响应的处理方式；不成立的部分是传输层面的产物，且文稿中的失败路径未被建模。
- 文稿中是否说明：否
- 依据：失败响应 403 和 500 未被建模；只建模了成功的创建。；第二种解读中的攻击是模型对请求和响应只用单一未受保护信道这一产物。
- 未决：失败响应 403 和 500 未被建模，因此该主张只对成功路径给出了回答；第二种解读依赖未受保护响应信道的产物。
- 下一步：对 403 和 500 响应及其对消费者订阅记录的影响进行建模。

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，以及各次攻击及分析者对每次攻击的判定。

目标	级别	界	查询	对照	攻击	主张
A03	none				a03 (holds_in_source)	1 attack
A04	none				a04 (holds_in_source)	1 attack
A08	none				a08 (holds_in_source)	1 attack
D01	none				a03 (holds_in_source)	1 attack
Z01	proved		z01	checked		2 proved
Z03	none				z03 (holds_in_source)	1 attack
S05	none				s05 (holds_in_source); s05b (holds_in_source)	2 attack
A08.8	proved		a088_own	checked	a088_real (model_artifact)	1 proved, 1 attack
A03.9	none				a03 (holds_in_source)	1 attack
U01	proved		a088_own	checked	a088_real (model_artifact)	2 proved, 1 attack

逐条主张，按分析者对每条主张的措辞；每条的回答均出自证明器：

目标	主张	攻击者	回答	查询	对照
A03	消费者接受的每条通知，都是由 MTLF 为其创建的订阅发送的，且 subscriptionId、事件标识符和模型地址均相同	`network`	attack (holds_in_source)	a03	
A04	订阅的每条被接受通知都是 MTLF 的一条不同通知；已记录的通知不能被重放并作为新通知再次被接受	`network`	attack (holds_in_source)	a04	
A08	被接受通知的 subscriptionId、事件标识符和 mLFileAddr 属于消费者为其 notifUri 和 mLEventSubscs 创建的订阅，而不是其他订阅或其他事件的	`network`	attack (holds_in_source)	a08	
D01	消费者接受的通知就是 MTLF 发送的那条，subscriptionId、eventNotifs 和 mLFileAddr 均未改变	`network`	attack (holds_in_source)	a03	
Z01	主张 1：MTLF 接受的 Subscribe 携带 NRF 为 nnwdaf-mlmodelprovision 范围签发的访问令牌	`network`	proved	z01	checked
Z01	主张 2：MTLF 接受的 Unsubscribe 携带 NRF 为 nnwdaf-mlmodelprovision 范围签发的访问令牌	`network`	proved	z01b	checked
Z03	为一个订阅、一个消费者身份或一个操作获得的授权，不能用于另一个订阅、另一个消费者或另一个操作	`network`	attack (holds_in_source)	z03	
S05	在 MTLF 成功处理了移除某订阅的 DELETE 之后，不再接受任何携带该 subscriptionId 的通知	`network`	attack (holds_in_source)	s05	
S05	在 MTLF 成功处理了移除某订阅的 DELETE 之后，不再接受任何携带该 subscriptionId 的通知（展开：退订前一条通知，确认 DELETE 后一条）	`sbi_nf`	attack (holds_in_source)	s05b	
A08.8	解读 2：被接受通知中的 subscriptionId 属于 MTLF 确实为该消费者创建的订阅，而不是属于另一消费者的订阅，也不属于从未创建的订阅	`network`	attack (model_artifact)	a088_real	

目标	主张	攻击者	回答	查询	对照
A08.8	解读 1：消费者接受的通知携带的 subscriptionId 是消费者本身从其自己的订阅创建中获得的	`sbi_nf`	proved	a088_own	checked
A03.9	每条被接受通知都对应消费者所订阅的 MTLF 的一次实际通知运行，且 subscriptionId 和事件标识符相同	`network`	attack (holds_in_source)	a03	
U01	主张：消费者接受的每条通知都属于创建成功的订阅；创建返回 403 或 500 的订阅永远不会产生被接受的通知	`sbi_nf`	attack (model_artifact)	a088_real	
U01	解读 1：消费者接受的每条通知都属于其在 201 Created 中收到的 subscriptionId，即属于它处理为成功的创建，而不属于创建返回 403 或 500 的订阅	`sbi_nf`	proved	a088_own	checked
U01	解读 1（网络攻击者）：消费者接受的每条通知都属于其在 201 Created 中收到的 subscriptionId，即属于它处理为成功的创建	`network`	proved	a088_own	checked

附录 D：分析是怎么跑的

一个 LLM 分析者根据文稿构建协议模型，并将每条性质交给 ProVerif；第 3 节中的每个结果都是证明器的答案。

ProVerif+ Agent 会话 session-20261114T182235，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步骤：63，证明器 4 秒，实际耗时 12 分钟，费用 0.108 美元。

ProVerif：运行 100 次证明，100 次判定，0 次无答案，3 秒，记录 29 个答案。

分析了什么

协议	NWDAF 模型提供：Subscribe → Notify → Unsubscribe
版本	19.8.0
分析范围	固定 Rel-19；一个消费者和一个 MTLF 提供者。使用直接 mLFileAddr 通知，不启用 ADRF、FL、漫游、模型共享及定位扩展。先验证创建订阅与一次通知，退订作为独立阶段扩展。
不在范围内	该流程提供模型地址，不保证文件内容/哈希完整性，模型下载不在首轮边界；不把可选 notifCorreId 或 modelUniqueId 强行设为必选字段；退订后的全局通知禁止、到期与重订阅需要状态展开或更适合状态的工具。仅证明明确模型及攻击者条件下的性质，不能把缺失检查、完整安全信道或全诚实 NF 当作待证结论的替代。
原文	342 numbered passages (87586 bytes) from the sections listed below. A reference such as s0001 in this report is one of these passages

The sections of the documents the analysis is built from:

文稿	章节	标题
main	4.5	Nnwdaf_MLModelProvision 服务
main	4.5.1	服务描述
main	4.5.1.1	概述
main	4.5.1.2	服务架构
main	4.5.1.3	网络功能
main	4.5.1.3.1	网络数据分析功能（NWDAF）
main	4.5.1.3.2	NF 服务消费者
main	4.5.2	服务操作
main	4.5.2.1	引言
main	4.5.2.2	Nnwdaf_MLModelProvision_Subscribe 服务操作
main	4.5.2.2.1	概述
main	4.5.2.2.2	订阅事件通知
main	4.5.2.2.3	更新事件通知订阅
main	4.5.2.3	Nnwdaf_MLModelProvision_Unsubscribe 服务操作
main	4.5.2.3.1	概述
main	4.5.2.3.2	取消事件通知订阅
main	4.5.2.4	Nnwdaf_MLModelProvision_Notify 服务操作

文稿	章节	标题
main	4.5.2.4.1	概述
main	4.5.2.4.2	关于已订阅事件的通知
main	5.4	Nnwdaf_MLModelProvision 服务 API
main	5.4.1	引言
main	5.4.2	HTTP 的使用
main	5.4.2.1	概述
main	5.4.2.2	HTTP 标准头
main	5.4.2.2.1	概述
main	5.4.2.2.2	内容类型
main	5.4.2.3	HTTP 自定义头
main	5.4.3	资源
main	5.4.3.1	资源结构
main	5.4.3.2	资源：NWDAF ML 模型提供订阅
main	5.4.3.2.1	描述
main	5.4.3.2.2	资源定义
main	5.4.3.2.3	资源标准方法
main	5.4.3.2.3.1	POST
main	5.4.3.2.4	资源自定义操作
main	5.4.3.3	资源：单个 NWDAF ML 模型提供订阅
main	5.4.3.3.1	描述
main	5.4.3.3.2	资源定义
main	5.4.3.3.3	资源标准方法
main	5.4.3.3.3.1	PUT
main	5.4.3.3.3.2	DELETE
main	5.4.3.3.4	资源自定义操作
main	5.4.4	无关联资源的自定义操作
main	5.4.5	通知
main	5.4.5.1	概述
main	5.4.5.2	事件通知
main	5.4.5.2.1	描述
main	5.4.5.2.2	操作定义
main	5.4.6	数据模型
main	5.4.6.1	概述
main	5.4.6.2	结构化数据类型
main	5.4.6.2.1	引言
main	5.4.6.2.2	类型 NwdafMLModelProvSubsc

文稿	章节	标题
main	5.4.6.2.3	类型 MLEventSubscription
main	5.4.6.2.4	空
main	5.4.6.2.5	类型 NwdafMLModelProvNotif
main	5.4.6.2.6	类型 MLEventNotif
main	5.4.6.2.7	类型 FailureEventInfoForMLModel
main	5.4.6.2.8	类型 MLModelAddr
main	5.4.6.2.9	空
main	5.4.6.2.10	空
main	5.4.6.2.11	类型 MLRepEventCondition
main	5.4.6.2.12	类型 InputDataInfo
main	5.4.6.2.13	类型 ModelProvisionParamsExt
main	5.4.6.2.14	类型 AdditionalMLModelInformation
main	5.4.6.2.15	类型 MLModelAdrf
main	5.4.6.2.16	类型 TrainInputDataInfo
main	5.4.6.2.17	类型 InferenceDataForModelTrain
main	5.4.6.2.18	类型 VflInfo
main	5.4.6.3	简单数据类型与枚举
main	5.4.6.3.1	引言
main	5.4.6.3.2	简单数据类型
main	5.4.6.3.3	枚举：FailureCode
main	5.4.6.3.4	枚举：MLModelMetric
main	5.4.6.3.5	枚举：VflTrainingStatus
main	5.4.6.3.6	枚举：PositioningCase
main	5.4.7	错误处理
main	5.4.7.1	概述
main	5.4.7.2	协议错误
main	5.4.7.3	应用错误
main	5.4.8	特性协商
main	5.4.9	安全

源文步骤在文稿中确定该协议的目的：固定 Rel-19；一个消费者和一个 MTLF 提供者。使用直接 mLFileAddr 通知，不启用 ADRF、FL、漫游、模型共享及定位扩展。先验证创建订阅与一次通知，退订作为独立阶段扩展。

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陈述进行了对照。名称、某条性质所针对的内容以及文稿的陈述均为分析者的解读；所列位置在文稿中确实存在。

ID	简称	对象	文稿怎么说	出处
A03	通知与提供商发送匹配	NF 服务消费者确信：它接受的每一条通知，都是由包含 MTLF 的 NWDAF 针对它所创建的订阅发送的，且带有它订阅时使用的 subscriptionId 和事件标识符。	通知携带 subscriptionId（s0109），并按事件携带事件标识符（s0111）；对订阅的响应在 Location 头中携带 subscriptionId（s0065）。文稿陈述了这种对应关系，但没有给出核对手段。	s0065, s0108, s0109, s0110, s0111, s0123
A04	通知不被接受两次	消费者把某订阅的每条通知都视为不同的通知；已记录的通知不能被重放并作为新通知接受。	文稿要求消费者存储通知并回复 204（s0123），并声明了一次性、周期性或事件检测通知（s0024），但在 NwdafMLModelProvNotif（s0109, s0110）中没有定义通知序列号、nonce 或任何其他新鲜性值。	s0024, s0109, s0110, s0123
A08	通知字段匹配自身订阅	消费者确信：通知中的 subscriptionId、事件标识符和 mLFileAddr 属于它自己订阅中放入的 notifUri 和 mLEventSubscs，而不是另一个订阅或另一个事件的。	notifUri 标识通知的接收方（s0041），mLEventSubscs 标识所订阅的事件（s0042），通知重述 subscriptionId（s0109）和事件（s0111）；文稿没有说明消费者比较什么。	s0041, s0042, s0108, s0109, s0111, s0112
D01	通知载荷是真实的	消费者确信：它接受的通知就是 MTLF 发送的那一条，subscriptionId、eventNotifs 和 mLFileAddr 均未改变；流程提供模型地址，但不提供内容或哈希完整性。	通知正文在 s0108 至 s0112 中逐字段定义，消费者存储它（s0123）；所选文本没有陈述通知的完整性或来源保护。TS 29.500 第 13.1 条（第 948 行）把消息保护放在传输层，不在本文稿范围内。	s0108, s0109, s0110, s0112, s0123
Z01	操作需要有效令牌	只有当请求方持有 NRF 为 nnwdaf-mlmodelprovision 作用域签发的有效 OAuth2 访问令牌时，NF 服务消费者对 Subscribe、Update、Unsubscribe 和 Notify 的访问才被授予。	s0339 指出访问“可借助 OAuth2 协议授权”，授权服务器为 NRF，s0340 至 s0342 给出令牌请求和单一服务作用域。令牌签发与校验的细节在 TS 29.510、29.500 和 33.501 中，不在本文稿范围内。	s0339, s0340, s0341, s0342
Z03	令牌绑定到其资源	为某一个订阅、某一个消费者身份或某一个操作获得的授权，不能用于 Nnwdaf_MLModelProvision API 上的另一个订阅、另一个消费者或另一个操作。	s0342 指出该 API 为整个服务定义单一作用域，“不在资源级或操作级定义任何附加作用域”；DELETE 资源在 URI 中仅由 {subscriptionId} 标识（s0188, s0192, s0095）。	s0342, s0188, s0192, s0095
S05	退订后无通知	NWDAF 成功处理了删除订阅的 DELETE 之后，不再有携带该 subscriptionId 的通知被处理，尤其是在曾持有该订阅的消费者处。	s0096 至 s0098 要求 NWDAF 删除订阅并回复 204 No Content；文稿没有提及已在途的通知，也没有提及消费者自己对已删除订阅的记录。	s0095, s0096, s0097, s0098, s0109
A08.8	通知属于自身订阅	消费者确信：它接受的通知携带的是它自己创建的订阅的 subscriptionId 和该订阅的事件标识符，而不是另一个消费者订阅的。	Location 头把 subscriptionId 返回给创建订阅的消费者（s0065），通知重述它（s0109）；文稿不要求消费者将其与已存储的订阅进行比较。	s0065, s0108, s0109, s0111
A03.9	被接受的通知有真实发送	消费者接受的每一条通知，都对应于它所订阅的 MTLF 的一次实际通知运行，且 subscriptionId 和事件标识符相同。	Notify 服务操作由 NWDAF 调用，以通知订阅的消费者实例（s0103, s0108）；消费者在收到时存储通知（s0123）。没有规定来源检查。	s0103, s0108, s0109, s0111, s0123
U01	失败订阅无通知	消费者不接受任何携带创建失败的 subscriptionId 的通知（VFL 为 403 Forbidden，ML 模型不可用为 500 Internal Server Error），且每一条被接受的通知都对应一次成功的创建。	s0065 使带 subscriptionId 的 201 Created 成为唯一成功结果；s0068 和 s0069 给出失败响应。文稿没有说明消费者如何处理针对它从未收到的 subscriptionId 的通知。	s0065, s0066, s0068, s0069, s0123

分析者没有补充性质：起草的清单已覆盖文稿写明的目标。

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 16 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 4 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 服务消费者	参与方	`sbi_nf`	
S02	NWDAF（含 MTLF）提供者	参与方	按定义全部失守	包含 MTLF 的 NWDAF 是它所通知模型的权威；如果攻击者控制它，模型地址和它报告的事件就由攻击者决定，而这正是通知性质假定排除的失守情形。所选文本没有对不诚实的提供者提出任何流程要求；模型文件本身在别处受保护（TS 33.501 附录 X，引自 TS 23.288 第 6.2A.0 条）。
S03	NRF	参与方	按定义全部失守	NRF 是授权服务器（s0339）。如果攻击者控制它，每个令牌都是它签发的，因此 Z01 和 Z03 按定义失守，且没有其他目标涉及令牌签发。
S04	subscriptionId	值	攻击者得不到更多	subscriptionId 在 201 响应的 Location 头和每条通知中传输（s0065, s0109）；网络已经读取它，因此持有它不会增加任何东西。
S05	notifUri	值	攻击者得不到更多	notifUri 在 Subscribe 请求体中（s0041），并在 SBI 上被读取；持有它不会给网络已有的信息增加任何东西。
S06	mLEvent 事件标识	值	攻击者得不到更多	mLEvent 在 Subscribe 请求和通知中（s0043, s0111）；它在 SBI 上是公开的。
S07	mLFileAddr	值	攻击者得不到更多	mLFileAddr 是通知的载荷（s0112），并在 SBI 上被读取；知道它正是 D01 和 A08 所检验的，而 D01 已经假定攻击者能看到它。
S08	通知内容	值	攻击者得不到更多	通知内容是 mLFileAddr 和事件标识符（s0110 至 s0112）；它在 SBI 上，因此网络攻击者已经持有它。
S09	mLEvent	值	攻击者得不到更多	mLEvent 标识符在 SBI 上是公开的（s0043, s0111）。
S10	通知载荷	值	攻击者得不到更多	通知载荷是 SBI 上 HTTP POST 的正文（s0108）；网络会读取它。
S11	OAuth2 访问令牌	值	`sbi_nf`	
S12	nnwdaf-mlmodelprovision 作用域	值	攻击者得不到更多	作用域名称 nnwdaf-mlmodelprovision 是固定的公开字符串（s0342）；持有它等于什么都没持有。
S13	订阅资源	值	`sbi_nf`	
S14	访问令牌	值	`sbi_nf`	
S15	订阅状态	值	攻击者得不到更多	提供者持有的订阅状态（存在哪些 subscriptionId，以及删除它们的 DELETE，s0064, s0096 至 s0098）不会给网络已从 SBI 读取的 subscriptionId 和 notifUri 增加任何东西；S05 问的是该状态如何表现，而不是谁持有它。
S16	错误响应	值	攻击者得不到更多	带有 problemDetails 原因的失败响应 403 和 500（s0068, s0069）不携带网络尚不拥有的任何值；没有目标要求失败不可区分。

每条性质在各攻击者条件下的答案

攻击者条件由分析者设定，且在编写查询之前就已设定；每个答案都记录在其中一个攻击者条件下。单元格是证明器在该攻击者条件下对相应性质的答案；“无答案”是为该性质设定的攻击者条件，在该条件下证明器对某条主张未作决定，或者没有记录任何内容；空单元格是分析者未为该性质设定的攻击者条件。

攻击者	持有什么	依据	出处
`network`	仅网络：它读取、丢弃、更改和发送消息，不持有任何一方的秘密		
`sbi_nf`	在服务化接口上作为 NF 服务消费者的一个位置，以及从 NRF 获得的、仅覆盖单一范围 nnwdaf-mlmodelprovision 的有效 OAuth2 访问令牌。它读取、注入、重放和修改 SBI 消息，并且可以自行运行 Subscribe/Unsubscribe。	s0339 将 NRF 作为授权服务器的 OAuth2 定为授权机制，s0342 指出该 API 为整个服务定义了单一范围，没有资源级或操作级范围，因此任何已授权的 NF 持有的令牌都覆盖每一个订阅和每一个操作。	s0339, s0340, s0342

ID	`network`	`sbi_nf`
A03	不成立	不成立
A04	不成立	不成立
A08	不成立	不成立
D01	不成立	不成立
Z01	成立	成立
Z03	不成立	不成立
S05	不成立	不成立
A08.8	成立	成立
A03.9	不成立	不成立
U01	成立	成立

原文和目标

Source: 342 units, 87586 bytes, frozen in source-delivery-01.json. Targets: 8 approved (A03, A04, A08, D01, Z01, Z03, S05, U01); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

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

- approve_package: 批准源文包 protocol-8c6f560bedbfa26c000a198dd5f3

这一步的自我报告：1 次请求（1 次失败），0.000 USD，输入 55291 个 token，输出 4125 个 token，耗时 156 秒。

关于证明器和模型的说明

- S05：在 ProVerif 中作答，分析者为此目标指定了 changing_state

一个简化代数（xor、dh_algebra、other_algebra）的 ProVerif 模型给攻击者的能力弱于真实操作：在那里得到的证明是针对更弱攻击者的证明，在那里得到的攻击就是攻击。一个遗漏了对变化值（changing_state）检查的 ProVerif 模型让各方接受更多：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+ 按原样对这些进行建模。

分析者的状态审查

这部分是分析者对原文的自身解读，不是证明器的答案。ProVerif 下“是”表示 ProVerif 在展开的模型中找到了包含这些步骤的轨迹，因此这些步骤是可能的；它们是否破坏某个目标，只由答案表中的攻击来断言，它们还带来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	由谁保持	何时变化	何时检查	参考
订阅记录：subscriptionId、notifUri 和已订阅的 mLEvent 集合	包含 MTLF 的 NWDAF（提供方）	由对 /subscriptions 的 POST 创建（s0064）；由对 /subscriptions/{subscriptionId} 的 PUT 修改并再次存储（s0077、s0078）；由 DELETE 移除（s0096、s0097）	在选择要通知的订阅时，提供方使用所存储订阅的 notifUri 和事件（s0108、s0110）；当 DELETE 到达时，它按 {subscriptionId} 查找该订阅（s0095）	s0064, s0077, s0078, s0095, s0096, s0097, s0100, s0108
消费者对其活动订阅的记录	NF 服务消费者	在返回带 {subscriptionId} 的 Location 的 201 Created 时（s0065）；文本未说明 DELETE 的 204（s0098）是否会改变它	未规定：消费者在收到通知时存储它并回答 204（s0123），没有说明将 subscriptionId 与某个活动订阅进行比较	s0065, s0098, s0123
subscriptionId	提供方作为资源标识符，消费者作为其订阅的句柄	由提供方在创建资源时签发（s0065）	在每条通知中重复（s0109），并用作 DELETE 和 PUT 的 {subscriptionId} 路径段（s0095、s0075）	s0065, s0095, s0098, s0109

模式	结果	目标	ProVerif 所示	步骤
replay	候选	A04	否：未用 ProVerif 尝试	1. 消费者订阅，MTLF 通过 SBI 发送其通知（s0108）。 / 2. 攻击者记录通知 POST，其中携带 subscriptionId、event 和 mLFileAddr，没有序列号、随机数或其他新鲜性值（s0109 至 s0112）。 / 3. 攻击者将相同的正文再次 POST 到相同的 notifUri。 / 4. 消费者将其存储为新通知并回答 204（s0123）；所记录的正文中没有任何东西能区分第二次接收与第一次。
desynchronisation	候选	S05	是：ProVerif 在 bounded.pv 上否证了 `s05b`	1. 消费者订阅，收到带 {subscriptionId} 的 201，并接受一条通知（s0065、s0123）。 / 2. 消费者发送 DELETE .../subscriptions/{subscriptionId} 并收到 204 No Content；提供方已移除该订阅（s0095 至 s0098）。 / 3. 双方现在持有不同的状态：提供方没有订阅，消费者没有任何状态来拒绝一条通知。 / 4. 一条携带相同 subscriptionId 的通知被 POST 到该 notifUri；消费者接受并存储它（s0123）。ProVerif 在消费者展开到三步的情况下找到了这一运行。
no_rollback_on_failure	未发现			提供方不接受的 DELETE 会以错误作答（s0100），订阅保留；文本没有给出记录被移除同时返回错误的路径，因此不产生回滚问题。
update_before_check	未发现			提供方仅在 POST 被接受后才存储订阅（s0064、s0065），且 PUT 仅在请求被成功处理时才修改它（s0076 至 s0078）。令牌在记录创建之前被检查，正如 s0339 和 s0340 所要求。
repeat_one_time_action	未发现			对已不存在的订阅重复执行 DELETE，会按第 5.4.7 条以 HTTP 错误作答（s0100）；文本没有给出使移除或创建发生两次的消息。
race_between_sessions	未发现			两个消费者持有由提供方签发的不同 subscriptionId（s0065），且提供方按 {subscriptionId} 查找订阅（s0095），因此一个会话的 DELETE 不会触及另一会话的记录。会话之间共享的值是令牌，那是 Z03，而不是对记录的竞争。
distinguishable_failure	未发现			提供方确实会回答不同的失败原因（403 VFL_MODEL_TO_BE_USED_FOR_ALL_EVENTS 和 500 UNAVAILABLE_ML_MODEL_FOR_ALLEVENTS，s0068、s0069，带有 5.4.7 的问题详情），但没有目标询问谁能从中得知什么，且该差异除请求本身外不泄露任何被保持的值。
wrap_or_window	未发现			subscriptionId 没有顺序，也没有接受窗口；expiryTime（s0050）是绝对时间，没有目标依赖它，且文本未规定任何回绕或窗口检查。

分析者为目标指定的内容：

步骤	目标	特征	原因
30	S05	changing_state, many_sessions	S05 询问在提供方移除订阅之后会发生什么：订阅状态（存在/已删除）是一个跨步骤变化的值，model.pv 遗漏了它，而 bounded.pv 将其展开。其他目标涉及在任意数量的消费者和提供方会话中通知与订阅之间的对应关系，ProVerif 对任意数量的会话都能覆盖。

分析者的结语

模型运行：model.pv 包含 NF 服务消费者、包含 MTLF 的 NWDAF 和 NRF，以及简单的 Subscribe -> 201 Created -> Notify -> 204 -> DELETE -> 204 运行；bounded.pv 将消费者展开到三个通知步骤，以应对变化的订阅状态。全部十个目标都有答案，且每一个记录的答案都在最终文本上重新核对过。

已确立：Z01 对两个主张均被证明（Subscribe 或 Unsubscribe 只有在持有 NRF 为该服务范围签发的令牌时才被接受），每个都配有一个失败的对照查询。A08.8 和 U01 在这样一种解读下被证明：消费者将通知绑定到它在 201 Created 中收到的 subscriptionId。其他所有目标均不成立：A03、A03.9 和 D01（消费者接受了一条背后没有 MTLF 发送的通知，且 mLFileAddr 任意），A08（事件标识符未与订阅比较），A04（记录的通知被重放并存储为新通知），S05（在展开的模型上也是如此），以及 Z03（服务令牌删除了另一消费者的订阅）。A08.8 和 U01 的第二种解读仅通过针对消费者自身请求的伪造响应而不成立，而传输层会阻止这种响应；它被记录为模型产物，不是发现。

有两项发现附带经证明器检查的修改：对包含 notifUri 的通知正文进行签名可消除通知类发现，针对订阅所有者授权 DELETE 可消除授权类发现；两种情况下正常流程均保留，且其他答案不变。重放和退订后两类发现仅给出建议，未在模型上检查。

其余被阻塞：模型将整个基于服务的接口交给攻击者，且不建模 TLS、OAuth2 令牌验证或模型文件，因此需要攻击者响应消费者自身发出的请求的攻击无法与传输伪影区分；到期、PUT 更新、重定向以及 403/500 失败路径均未建模；两项未检查的建议需要各自的模型运行。