NWDAF 模型训练事件订阅与结果通知的安全分析

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	11 条：8 条成立，3 条不成立（5 个攻击）
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

Nnwdaf_MLModelTraining 服务让 NF 服务消费者能够在包含模型训练逻辑功能的 NWDAF 上订阅机器学习模型训练事件，并让该 NWDAF 将训练结果回传给消费者。涉及三方：消费者、带 MTLF 的 NWDAF，以及 NRF，后者充当 nnwdaf-mlmodeltraining 范围的 OAuth2 授权服务器。

消费者持有其服务令牌、一个自行生成的通知识别符，以及它希望接收通知的 URI；NWDAF 持有已创建的订阅及其订阅标识符。交互过程包括一次订阅及创建订阅响应、一条或多条携带模型信息或终止请求的通知，以及一次取消订阅。

交互结束后，消费者知道它存储的每条通知都属于它自己创建的订阅，确实来自 NWDAF 且未被篡改，并且 NWDAF 是在校验了服务范围的令牌之后才创建该订阅的。它无法得知某条通知是否为重复，也无法得知终止请求与其前后通知之间的关系。

参与方：

- **NF 服务消费者**：订阅，存储其接受的通知。持有：nnwdaf-mlmodeltraining 范围的令牌、notifCorreId、notifUri、mLEvent。
- **NWDAF（MTLF）**：创建并存储订阅、训练、发送通知。持有：OAuth2 令牌验证密钥、已存储的订阅。
- **NRF**：OAuth2 授权服务器。持有：消费者的客户端凭据。

图 1 展示了文稿所规定的交互过程。



图 1. 按规范规定的协议。

分析的范围见附录 D：分析了什么、遗漏了什么，以及依据文稿中的哪些章节。

2 发现

分析共有 2 项发现：以下内容并未在文稿中写明。其中 1 项是否在文稿中写明尚不明确。

在分析的范围内，关联校验成立：被接受的通知总是属于消费者自己创建的订阅，notifCorreId、notifUri 和 mLEvent 均匹配，并且确实由 NWDAF 原样发送，因为消费者依据自己选择的标识符进行关联，且 SBA 传输保护了消息。仍存在两个缺口，二者都是发现。没有任何内容标识通知实例或对通知流排序，因此被扣

留或重复的通知会被再次接受，在终止请求之后和取消订阅之后也是如此。并且没有字段说明通知的类型，因此同时携带最终聚合模型的终止请求会被当作普通结果读取。订阅和取消订阅操作的令牌校验成立。这些结果基于一个将 SBA 传输视为经过认证的加密信道、并建模单个消费者、单个提供者和每个订阅至多两条通知的模型。

攻击图使用图 1 的列。红色表示攻击者所做的操作。黄色标记某参与方在规范规定的交互中本不应在该时点发送的消息，以及攻击所破坏的结果。

2.1 携带最终模型的终止通知被当作普通结果读取

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	U01（见第 3 节）
原文	s0128, s0131, s0133, s0321, s0337
依据	ProVerif 攻击查询 `u01_cls`、`x01_both`；图 2

发生了什么。 通知没有类型字段。消费者必须根据出现的字段来判断收到的是模型信息还是终止请求，而规范使 termTrainReq 和 mLModelInfos 可以同时合法存在：NOTE 1 要求 delayEventNotif、mLModelInfos 和 termTrainReq 至少出现其一，并且只规定 delayEventNotif 与另外两者互斥。因此，先检查 mLModelInfos 的消费者会把同时携带最终聚合模型的终止请求归类为训练结果，永远无法得知订阅已结束。该服务中其他位置也出现了同样的形式，例如 TrainingUnsubscribeInfo 将 termCause FL_FINISHED 与 mLModelInfos 一同携带。

攻击者需要什么。 无需攻击者：当 NWDAF 完成一次训练并上报其最终模型时，它自己就会发送这条格式正确的通知。网络上的攻击者只需重排这两条通知即可达到相同状态。

攻击者能得到什么。 消费者会继续在一个已经结束的订阅上训练、等待永远不会到来的通知，或者把最终模型当作中间结果处理，而存储的通知中没有任何字段能让它恢复这一区别。

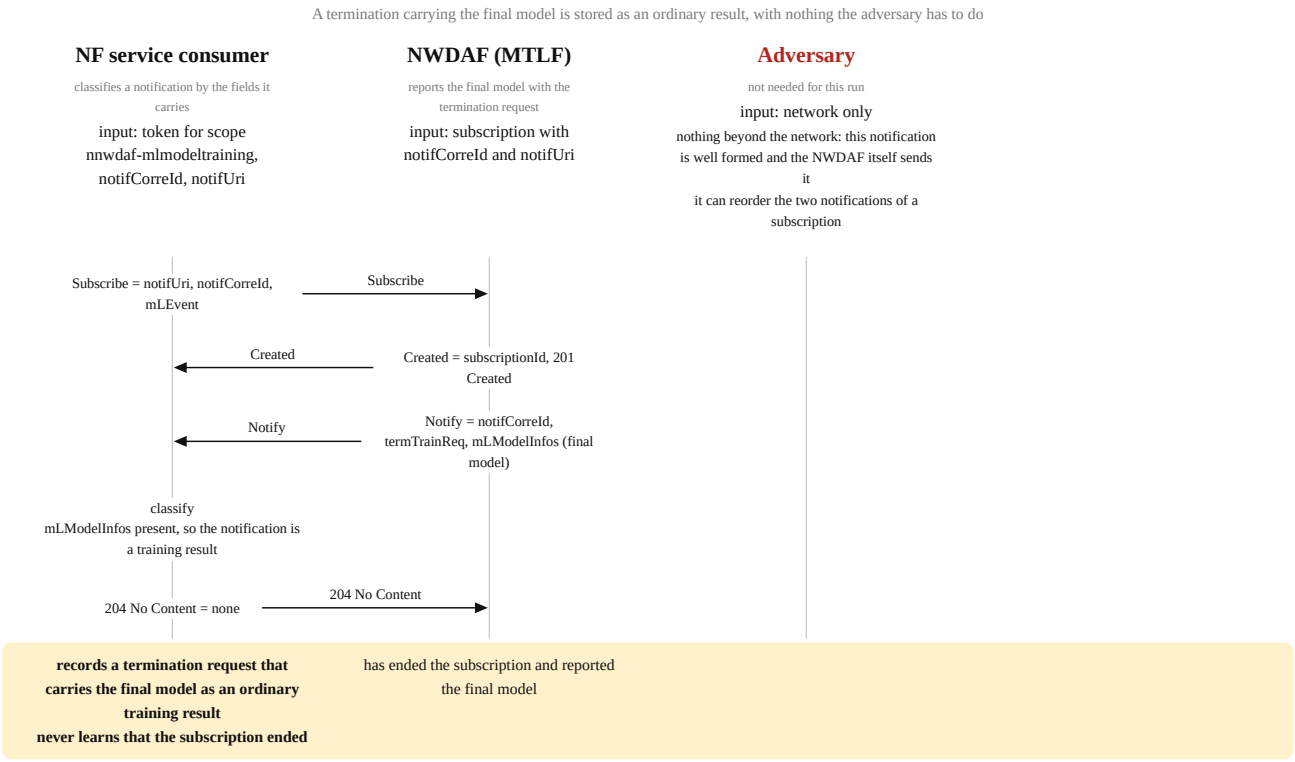


图 2. 携带最终模型的终止请求被当作普通结果存储，攻击者无需做任何操作。性质 U01；攻击查询 u01_cls、x01_both。

拟议修改。 为 NwdafMLModelTrainNotif 增加显式的通知类型，或者规定 termTrainReq 在出现时具有决定性，即携带它的通知无论还携带什么都属于终止请求，并在 NOTE 1 中说明 termTrainReq 与 mLModelInfos 的组合适用于哪些训练类型。

用 ProVerif 复核。 未检查：要检查这一点，需要每条通知都携带一个通知类型，这意味着修改消息及其所有生产者；该模型只能检查消费者侧的优先级规则，无法检查通知是否变得可区分。

2.2 重复和被扣留的通知会被再次接受，在终止和取消订阅之后也是如此

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	R02, S05（见第 3 节）
原文	s0128, s0129, s0138, s0321
依据	ProVerif 攻击查询 `r02`、`s05_term`、`s05_unsub`；图 3 和图 4

发生了什么。 通知中没有任何内容标识通知实例或其在流中的位置：notifCorreId 标识的是订阅，而 roundInd 是可选的且不受约束。当消费者收到一条它已经接受过的通知时，会再次存储它。被攻击者扣留、并在终止请求之后或取消订阅得到响应之后才释放的通知，会被当作新通知接受，尽管 termTrainReq 承诺不再提供该订阅的后续通知。

攻击者需要什么。 攻击者只需要网络：它读取一条自己无法伪造的通知，将其保留，之后再投递或投递两次。它不需要任何密钥，既没有破坏传输保护，也没有破坏令牌校验。

攻击者能得到什么。 消费者存储的订阅记录中可能包含重复的同一结果，也可能包含在其被告知订阅已结束或已取消订阅之后才到达的结果，因此根据存储的通知计算的分析结果可能将一次训练结果重复计数，或在事后才计入。

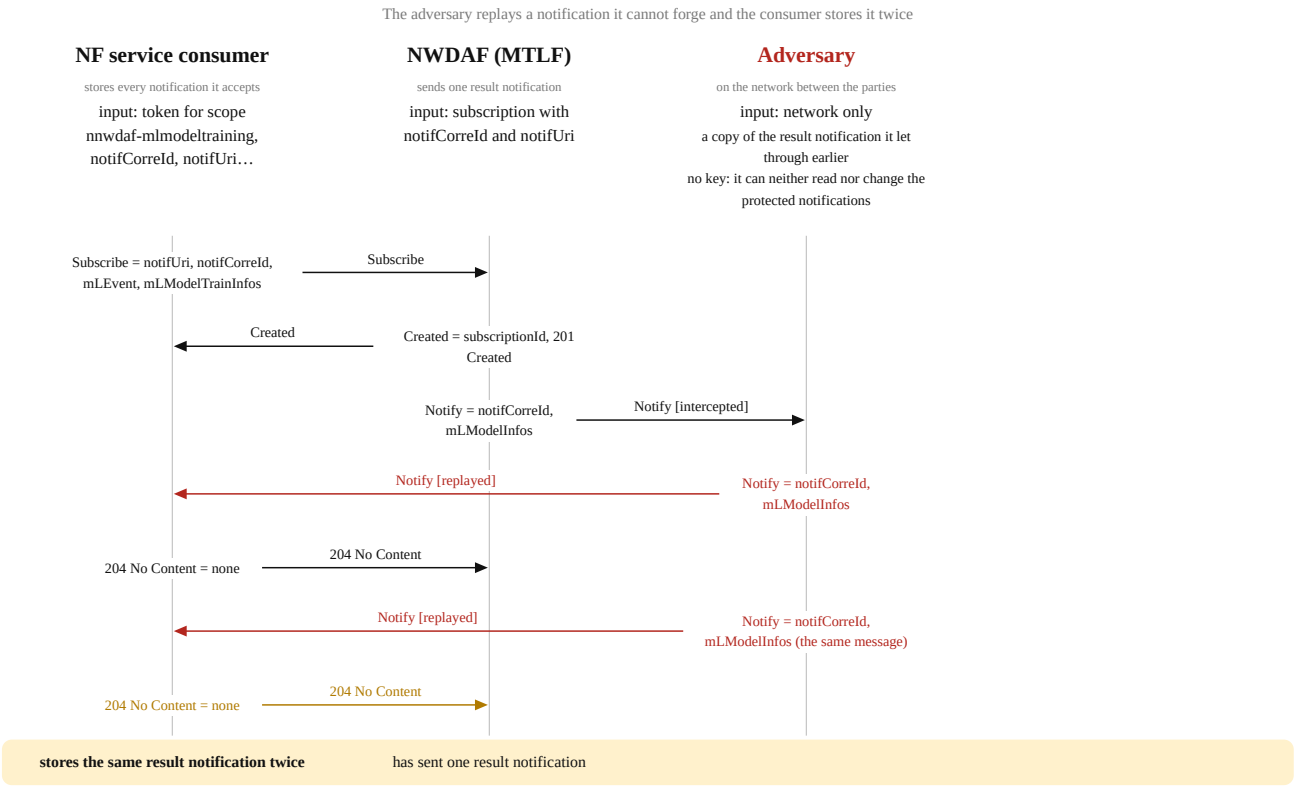


图 3. 攻击者重放一条它无法伪造的通知，消费者将其存储了两次。性质 R02；攻击查询 r02。

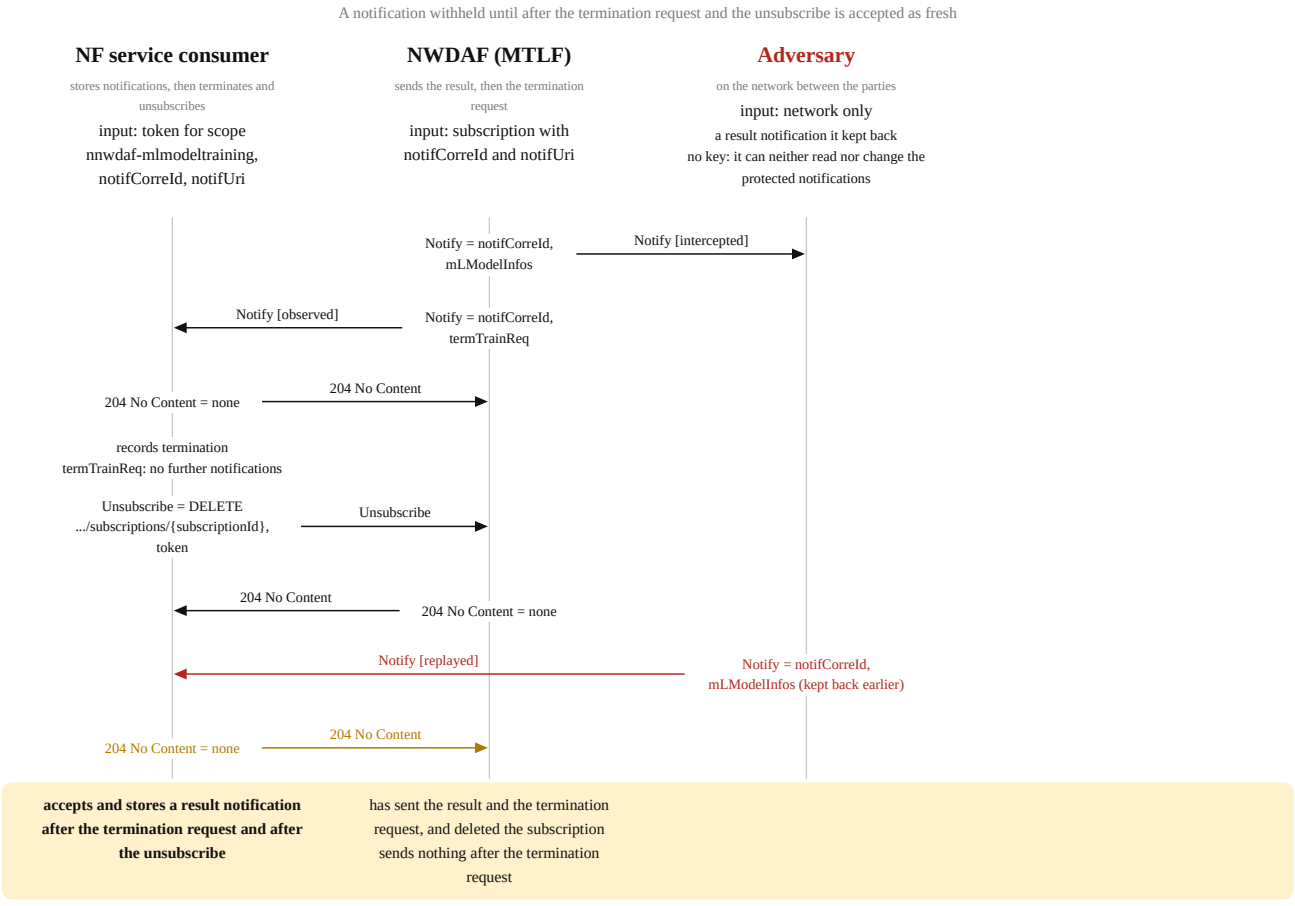


图 4. 一条被扣留到终止请求和取消订阅之后的通知被当作新通知接受。性质 S05；攻击查询 s05_term、s05_unsub。

拟议修改。 在 NwdafMLModelTrainNotif 中携带每订阅的通知序列号，在存在轮次的地方将 roundInd 设为必填，并要求消费者存储并检查某条通知的编号大于它为该订阅存储的上一个编号，丢弃其他所有通知。

用 ProVerif 复核。 未检查：该校验需要一个按订阅变化、并在每条通知上比较的值，而本模型没有这样的值：ProVerif 会遗忘某一步执行了多少次以及以何种顺序执行，因此该性质必须在展开的有界模型上或用 Tamarin+ Agent 来证明。

本发现中这些攻击没有提供图：R02：s05_term、R02：s05_unsub、S05：r02。

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的回答；价值和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
A03	**通知绑定到自身订阅。** 消费者每次接受一条训练事件通知，都存在一个由该消费者真实发起、且 notifCorreId 与 notifUri 与之匹配的订阅。	无	成立	低	是
A08	**通知上下文与订阅匹配。** 消费者接受的每条通知，其 notifCorreId、事件标识与轮次等上下文字段都对应同一消费者此前获批的订阅及其事件描述。	无	成立	低	不确定
D01	**通知的真实性与完整性。** 消费者接受的每条训练事件通知，都对应 NWDAF 针对该订阅真实发送的一条通知，且通知载荷与发送时一致。	无	成立	中	不确定
R02	**每个通知实例仅接受一次。** 对同一订阅的同一逻辑通知实例，消费者至多接受一次。	无	不成立（1 个攻击）	中	否
S05	**终止后不再接受任何内容。** 在终止通知或退订生效之后，消费者不再接受与该已终止订阅关联的新训练事件通知。	无	不成立（2 个攻击）	中	不确定
Z01	**令牌覆盖的服务操作。** 每次订阅、更新、退订或通知操作被接受，都存在授权服务器签发的、覆盖 nnwdaf-mlmodeltraining scope 的有效 token。	无	成立	中	是
Z03	**令牌绑定到其订阅。** 一个消费者持有的 token 只能用于其自身发起的订阅与操作，不能被用于其他消费者的订阅或通知。	无	成立	中	不确定
A03.8	**notifCorreId 等于自身订阅。** 消费者接受的每条通知，其 notifCorreId 等于该消费者真实发起的那次订阅中的 notifCorreId。	无	成立	低	是
A08.9	**订阅不混用。** 消费者接受的每条通知，其订阅关联标识与事件参数都对应同一消费者自己的订阅，不会来自其他订阅。	无	成立	中	不确定
U01	**终止与结果区分。** 消费者接受的每条通知，termTrainReq 与 mLModelInfos 不同时出现，且消费者按实际字段将其归类为终止事件或结果事件。	无	不成立（2 个攻击）	中	不确定
X01	**通知至少携带一个报告字段。** 消费者接受的每条通知至少携带 delayEventNotif、mLModelInfos 或 termTrainReq 之一，且绝不同时携带 delayEventNotif 与 mLModelInfos，也不同时携带 delayEventNotif 与 termTrainReq。	无	成立	低	是

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.

提出的问题及分析者的回答：

问题	回答	依据
Q1: 通知的 notifCorreId 与请求任务绑定	是，但仅靠标识符相等来实现：通知的 notifCorreId 必须是消费者为该订阅选定的值；由于消息受保护且该值是新生成的，消费者可以把一条通知绑定到它提交的任务上；通知中没有其他内容把它与请求关联起来。	A03, A03.8, D01
Q2: 训练结果/失败通知不被混入其他订阅	训练结果或终止事件不会被投递到其他订阅的处理程序：关联标识符和订阅标识符都必须与消费者自己的订阅匹配，因此结果不可能在错误的订阅下到达。	A08.9, A08, A03
Q3: 把终止通知与训练结果通知区分为不同事件	否，二者不能可靠区分：终止通知也可能携带最终聚合模型，而且由于通知没有类型字段，先读取模型信息的消费者会把终止当作普通结果处理。不带模型信息的终止是可区分的，这也是模型表明唯一安全的情形。	U01

4 本次分析的局限

这里的结果是针对协议模型的结果：它说明在密码学被视为理想的前提下，规范所定义的消息与检查提供了什么、未能提供什么，而对实现不作任何说明。

按分析者的说法，本次分析不覆盖以下内容：

- SBA 传输被建模为每一对参与方（消费者到 NWDAF、消费者到 NRF）之间使用一个长期密钥的认证加密；TLS 细节、SEPP/SCP 路径以及传输层自身的重放处理均未建模，传输密钥和令牌密钥也从不被攻破。
- 只建模一个消费者身份、一个 NWDAF 和一个令牌密钥：跨消费者使用订阅标识符无法表达，因此关于令牌范围的回答只涉及网络攻击者。
- 每个订阅至多建模两条通知（一条结果、一条终止请求），轮次编号、statusReport 以及多轮训练状态均未包含，因此对一轮之内或跨多轮的通知顺序不作任何说明。
- NWDAF 在退订时不删除任何已存储的订阅：模型只表明诚实的提供方不再发送任何内容，因此不覆盖不诚实提供方在退订后的行为。
- OAuth2 令牌被建模为对服务范围和消费者身份的 MAC；令牌生命周期、受众、按资源的范围以及 NRF 侧的授权策略不在范围内。
- 更新与部分更新操作（PUT、PATCH）、带 failEventReports 的部分成功路径、延迟通知以及退订信息操作均未建模。
- 训练模型的正确性以及模型信息的任何机密性均未建模，且不在任务范围内。

宿主记录的内容：

- 没带对照查询的证明（附录 C）：A03: a03, A08: a03, D01: d01, Z01: z01_unsub, Z03: z03, A03.8: a03, A08.9: a08_9, X01: x01_empty。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、需要什么才能定论，见附录 A。

附录 A：逐条性质的解读

以下是分析者自己对证明器回答的解读，不是证明器的回答：这些回答对协议意味着什么、解读依据是什么、能否按现状使用，以及它的价值如何。

A03: 通知绑定到自身订阅

消费者每次接受一条训练事件通知，都存在一个由该消费者真实发起、且 notifCorreId 与 notifUri 与之匹配的订阅。

消费者存储的每条通知都属于消费者自己发出的某个订阅请求，具有相同的 notifCorreId、相同的 mLEvent 以及自己的 notifUri。这条性质对网络攻击者成立，因为消费者检查的是自己选定的标识符，且传输保证了消息完整。

- 结果：成立；是否回答了所问：是；可带保留地使用
- 价值低：规范本身规定通知关联标识符应设置为与订阅中相同的值，因此该结果只是在明确的传输假设下确认了一条已写明的规则，而非发现了新问题。
- 文稿是否已写明：是。文稿中的位置：main:27562
- 依据：SBA 传输被建模为消费者与 NWDAF 共享密钥下的认证加密，因此攻击者无法伪造或篡改通知；消费者被建模为在接受前对照自己的订阅检查 notifCorreId、notifUri 和 mLEvent；一个消费者身份，每个订阅至多两条通知
- 下一步：无；只有对令牌范围的跨订阅解读才需要双消费者模型。

A08: 通知上下文与订阅匹配

消费者接受的每条通知，其 notifCorreId、事件标识与轮次等上下文字段都对应同一消费者此前获批的订阅及其事件描述。

被接受通知的事件标识符，是消费者在同一关联标识符下订阅的事件。轮次和状态上下文字段未建模，因此对通知的轮次是否与订阅的预期相符不作任何说明。

- 结果：成立；是否回答了所问：部分；可带保留地使用
- 价值低：文稿说通知报告的是 notifCorreId 所标识订阅的模型信息，但从未规定所订阅的事件标识符与所通知的事件标识符之间的对应关系，而轮次上下文又不在建模范围内。
- 文稿是否已写明：不明确
- 依据：通知上下文中只建模了事件标识符 mLEvent；roundInd 和 statusReport 缺失；传输保护通知不被篡改
- 未决：通知的轮次上下文（roundInd、statusReport）
- 下一步：在有界模型中展开多轮训练状态，并考察通知的轮次能否被乱序接受。

A03.8: notifCorreId 等于自身订阅

消费者接受的每条通知，其 notifCorreId 等于该消费者真实发起的那次订阅中的 notifCorreId。

每条被接受通知的关联标识符，都等于消费者真实发出的订阅请求中的关联标识符；其他方无法产生该值，因为它由消费者重新生成，且只在受保护的消息中传递。

- 结果：成立；是否回答了所问：是；可带保留地使用

- 价值低：该值的相等正是属性描述所规定的；证明只补充了使其不可猜测的传输假设。
- 文稿是否已写明：是。文稿中的位置：main:27562
- 依据：传输凭据从未被攻破；消费者在接受前检查该标识符
- 下一步：无。

A08.9: 订阅不混用

消费者接受的每条通知，其订阅关联标识与事件参数都对应同一消费者自己的订阅，不会来自其他订阅。

通知不会从其他订阅混入：它所携带的订阅标识符是 NWDAF 为在同一关联标识符下创建的订阅所分配的，因此持有多个订阅的消费者靠标识符而非任何密码学绑定将它们区分开。

- 结果：成立；是否回答了所问：是；可带保留地使用
- 价值中：文稿将 notifCorreId 定义为标识通知所关联的订阅，但对通知与其订阅之间的密码学绑定只字未提。
- 文稿是否已写明：不明确
- 依据：绑定依靠受保护消息内部的标识符相等，而非按订阅的密钥或签名；只有一个消费者身份，因此不同消费者的两个订阅未建模
- 未决：两个不同消费者的订阅之间的混用
- 下一步：一个包含两个消费者身份和共享通知端点的模型可以定论跨消费者的解读。

D01: 通知的真实性与完整性

消费者接受的每条训练事件通知，都对应 NWDAF 针对该订阅真实发送的一条通知，且通知载荷与发送时一致。

消费者收到的每条通知确实都是 NWDAF 针对该订阅发送的，且其载荷未被篡改；这一保证完全来自传输层凭证，因为通知本身不携带签名或 MAC。

- 结果：成立；是否回答了所问：是；可用但有保留
- 价值 中：29.500 和 33.501 文稿要求服务化接口在传输层使用 TLS，这也是模型所假设的，但这两份文稿以及服务规范都没有把通知真实性表述为一项性质，而且 TS 33.501/29.500 不在所选原文单元之内。
- 文稿中是否陈述：不明确
- 依据：全部保护就是所建模的 SBA 传输；没有它，载荷就未经认证；NWDAF 对模型信息的内容是诚实的
- 待解决：通过 SEPP 或 SCP 的间接通信终止传输时的保护
- 下一步：对 SEPP/SCP 跳建模，看通知是否得到端到端保护。

R02: 每个通知实例仅接受一次

对同一订阅的同一逻辑通知实例，消费者至多接受一次。

一条被投递两次的通知会被存储两次。通知中没有任何内容能区分它的两个实例，因此消费者无法察觉重复，而该攻击除了网络之外不需要任何其他条件。

- 结果：不成立（1 个攻击）；是否回答了所问：是；可用但有保留
- 价值 中：我在文稿中检索了重放、重复通知和排序规则，没有找到关于通知重复或去重的任何陈述；唯一提到的保护是传输层的 TLS，而它并不对两个通知请求排序。
- 文稿中是否陈述：否

- 依据：假设传输不会拒绝重复的应用消息；原文对通知重复没有任何说明；消费者对它所接受的每条通知都予以存储，不做去重
- 待解决：真实部署中的传输是否会拒绝重复请求
- 下一步：在类型中声明通知序列号并在消费者处校验，然后重新运行该性质。

S05: 终止后不再接受任何内容

在终止通知或退订生效之后，消费者不再接受与该已终止订阅关联的新训练事件通知。

终止请求会结束通知、退订会结束通知，这一承诺在消费者处并未得到强制执行：被攻击者扣留、之后才释放的通知仍会被接受并存储，因此存储记录中可能包含在订阅结束之后才到达的结果。

- 结果：不成立（2 个攻击）；是否回答了所问：是；可用但有保留
- 价值 中：termTrainReq 声明不再提供与该订阅相关的通知，退订会删除订阅，但文稿对已在途的通知或消费者如何识别这类通知只字未提；弱点在于文本没有说明的内容。
- 文稿中是否陈述：不明确
- 依据：消费者不保存任何标记订阅已结束的状态，而这一点正是原文未作规定之处；NWDAF 在模型中不删除任何内容，只是不再发送；假设传输不会拒绝被重新投递的通知
- 待解决：不诚实的 NWDAF 在终止后仍继续发送通知的行为
- 下一步：增加按订阅的结束标记或序列号，并要求消费者丢弃其后的任何内容。

Z01: 令牌覆盖的服务操作

每次订阅、更新、退订或通知操作被接受，都存在授权服务器签发的、覆盖 nnwdaf-mlmodeltraining scope 的有效 token。

NWDAF 在检查 NRF 为 nnwdaf-mlmodeltraining scope 签发的 token 之后才创建订阅并接受退订，而该性质正是依赖这一检查：去掉检查后就会出现从未验证 token 的接受。通知方向未被覆盖，因为规范没有为 Notify 操作定义 token，更新操作也未建模。

- 结果：成立；是否回答了所问：部分；可用但有保留
- 价值 中：安全章节声明访问可用 OAuth2 客户端凭证授权，且消费者在消费服务之前应获取 token；该结果针对订阅和退订重述了这条规则，并明确指出通知不携带任何 token。
- 文稿中是否陈述：是。文稿陈述位置：main:27946, main:27948
- 依据：token 被建模为对服务 scope 和消费者身份的 MAC，密钥由 NRF 和 NWDAF 持有；Notify 操作在建模时不带任何 token，因为原文未定义；更新和部分更新操作未建模
- 待解决：Notify 操作是否本就打算由 token 授权；更新和部分更新操作
- 下一步：查阅 TS 29.500 中针对 Notify 方向的 API 访问控制条款。

Z03: 令牌绑定到其订阅

一个消费者持有的 token 只能用于其自身发起的订阅与操作，不能被用于其他消费者的订阅或通知。

只有当订阅是由同一消费者建立的，针对该订阅的操作才会被接受，因为标识符只在受保护的交换内部传递。token 绑定到服务而非订阅，因此任何获知订阅标识符的已授权消费者都可以用自己的 token 对其操作；模型中只有一个消费者，所以该解读只对网络上的攻击者给出了回答。

- 结果：成立；是否回答了所问：部分；可用但有保留
- 价值 中：文稿陈述了单一的服务级 scope 以及不存在资源级 scope，这正是保留意见的成因，但从未说明持有已知订阅标识符的消费者可以对其做什么。

- 文稿中是否陈述：不明确
- 依据：只有一个消费者身份，因此跨消费者使用无法表达；订阅标识符不可猜测且只在受保护的消息内部传递；token 只携带服务 scope，正如规范所述
- 待解决：订阅标识符的跨消费者使用
- 下一步：对两个持有同一 scope 有效 token 的消费者建模，并考察其中一个在知道标识符后能否退订或修改另一个的订阅。

U01: 终止与结果区分

消费者接受的每条通知，termTrainReq 与 mLModelInfos 不同时出现，且消费者按实际字段将其归类为终止事件或结果事件。

消费者无法可靠地区分终止与训练结果。一条通知可能同时携带 termTrainReq 和 mLModelInfos，而按模型信息是否存在来分类的消费者会把这样的终止解读为普通结果，从而永远不知道订阅已结束。

- 结果：不成立（2 个攻击）；是否回答了所问：是；可用但有保留
- 价值 中：通知类型的 NOTE 1 陈述了字段规则，并让 termTrainReq 与 mLModelInfos 可以共存，TrainingUnsubscribeInfo 在其他地方也展示了同样的组合，但文稿从未说明消费者应如何区分终止与结果，因此成因虽已陈述，后果却未陈述。
- 文稿中是否陈述：不明确
- 依据：假设消费者的分类规则先检查 mLModelInfos 再检查 termTrainReq，这是字段规则的两种可能解读之一；每个订阅只建模一个结果通知和一个终止通知
- 待解决：合规的消费者应以哪个字段为准
- 下一步：询问规范编辑者 termTrainReq 是否具有决定作用，或要求显式的通知类型。

X01: 通知至少携带一个报告字段

消费者接受的每条通知至少携带 delayEventNotif、mLModelInfos 或 termTrainReq 之一，且绝不将 delayEventNotif 与 mLModelInfos 或 termTrainReq 同时携带。

消费者接受的每条通知至少携带一个报告字段，因此不带任何报告的通知不会出现。该规则的另一半，即 delayEventNotif 绝不与其他两个字段同时出现，未建模，因为延迟通知属于范围之外的联邦学习部分。

- 结果：成立；是否回答了所问：部分；可用但有保留
- 价值 低：NOTE 1 正是陈述了这条字段规则，因此该结果只是复述了一项已陈述的要求，而非发现任何新内容。
- 文稿中是否陈述：是。文稿陈述位置：s0321
- 依据：延迟通知未建模，因此其互斥性未被检验；NWDAF 对所填写的字段是诚实的
- 待解决：delayEventNotif 的互斥性
- 下一步：对联邦学习情形的延迟通知建模，以检验该规则的另一半。

附录 C：主张和证明器的答案

证明器对每个目标的记录：达到的层级、达到该层级的查询、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
A03	proved		a03			1 proved
A08	proved		a03			1 proved
D01	proved		d01			1 proved
R02	none				r02 (depends_on_ambiguity)	1 attack
S05	none				s05_term (depends_on_ambiguity); s05_unsub (depends_on_ambiguity)	2 attack
Z01	proved		z01	1 of 2		2 proved
Z03	proved		z03			1 proved
A03.8	proved		a03			1 proved
A08.9	proved		a08_9			1 proved
U01	none				x01_both (holds_in_source); u01_cls (holds_in_source)	2 attack
X01	proved		x01_empty			1 proved

逐条主张，按分析者对每条主张的表述给出；每条的回答来自证明器：

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属于该消费者确实发送过的订阅请求，且 notifCorreId 与 mLEvent 均相同。	network	proved	a03	none
A08	被接受通知的 mLEvent 是该消费者在同一 notifCorreId 下订阅的事件；轮次上下文（roundInd、statusReport）未建模。	network	proved	a03	none
D01	消费者收到的每条通知确实由 NWDAF 为该订阅发送，且其载荷未被篡改（传输凭据是唯一的保护手段）。	network	proved	d01	none
R02	消费者对同一通知至多接受一个实例；实际上它会再次接受重复的通知，因为通知中没有任何内容能标识该实例。	network	attack (depends_on_ambiguity)	r02	
S05	消费者接受某订阅的终止通知（termTrainReq）之后，不再接受该订阅的任何通知；实际上它会再次接受同一终止通知。	network	attack (depends_on_ambiguity)	s05_term	
S05	某订阅的取消订阅被接受之后，消费者不再接受该订阅的任何通知；实际上在取消订阅之前发送的通知会在取消订阅之后被接受。	network	attack (depends_on_ambiguity)	s05_unsub	
Z01	NWDAF 接受的每个订阅请求都由 NRF 为 nnwdaf-mlmodeltraining 范围签发的令牌授权。	network	proved	z01	checked
Z01	NWDAF 接受的每次取消订阅都由 NRF 为 nnwdaf-mlmodeltraining 范围签发的令牌授权。	network	proved	z01_unsub	none
Z03	对订阅的操作只有在订阅由该消费者建立时才被接受；其他方不知道其标识符就无法对其操作。	network	proved	z03	none
A03.8	被接受通知的 notifCorreId 等于消费者确实发送的订阅请求的 notifCorreId。	network	proved	a03	none

目标	主张	攻击者	回答	查询	对照
A08.9	被接受通知携带的订阅标识符是 NWDAF 为同一 notifCorreId 下创建的订阅所分配的标识符，因此通知不会从其他订阅混入。	`network`	proved	a08_9	none
U01	通知绝不会同时携带 termTrainReq 和 mLModelInfos；实际上原文允许这样做，NWDAF 可以发送同时携带最终模型的终止通知。	`network`	attack (holds_in_source)	x01_both	
U01	消费者根据通知携带的字段对其进行分类，因此同时携带最终模型的终止通知会被归类为训练结果，消费者不会得知订阅已结束。	`network`	attack (holds_in_source)	u01_cls	
X01	消费者接受的每条通知至少携带一个报告字段，因此消费者从不需要处理完全没有报告的通知。	`network`	proved	x01_empty	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，并将每条性质提交给 ProVerif；第 3 节中的每个结果都是证明器的回答。

ProVerif+ Agent 会话 session-20261114T184451，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：56，证明器 5 s，实际耗时 13 min，费用 0.119 USD。

ProVerif：运行 101 次证明，93 次判定，8 次无答案，3 s，记录 14 个回答。

分析了什么

协议	NWDAF 模型训练事件订阅与结果通知
版本	19.8.0
分析范围	单训练提供者、单消费者、单训练任务；模型训练为不透明内部计算，先验证普通事件订阅/通知，不启用完整水平/垂直联邦学习。
不在范围内	不是完整 HFL/VFL 协议，也不证明训练结果正确；重复订阅、终止后重订阅及轮次状态需要单独展开。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	来自下列章节的 379 个编号原文单元（88264 字节）。本报告中诸如 s0001 的引用即为其中之一的原文单元

分析所依据的文稿章节：

文稿	章节	标题
main	4.6	Nnwdaf_MLModelTraining Service
main	4.6.1	Service Description
main	4.6.1.1	Overview
main	4.6.1.2	Service Architecture
main	4.6.1.3	Network Functions
main	4.6.1.3.1	Network Data Analytics Function (NWDAF)
main	4.6.1.3.2	NF Service Consumers
main	4.6.2	Service Operations
main	4.6.2.1	Introduction
main	4.6.2.2	Nnwdaf_MLModelTraining_Subscribe service operation
main	4.6.2.2.1	General
main	4.6.2.2.2	Subscription for event notifications
main	4.6.2.2.3	Update subscription for event notifications
main	4.6.2.2.4	Partial update subscription for event notifications
main	4.6.2.3	Nnwdaf_MLModelTraining_Unsubscribe service operation
main	4.6.2.3.1	General
main	4.6.2.3.2	Unsubscribe from event notifications

文稿	章节	标题
main	4.6.2.3.3	Unsubscribe from event notifications providing additional information
main	4.6.2.4	Nnwdaf_MLModelTraining_Notify service operation
main	4.6.2.4.1	General
main	4.6.2.4.2	Notification about subscribed event
main	5.5	Nnwdaf_MLModelTraining Service API
main	5.5.1	Introduction
main	5.5.2	Usage of HTTP
main	5.5.2.1	General
main	5.5.2.2	HTTP standard headers
main	5.5.2.2.1	General
main	5.5.2.2.2	Content type
main	5.5.2.3	HTTP custom headers
main	5.5.3	Resources
main	5.5.3.1	Resource Structure
main	5.5.3.2	Resource: NWDAF ML Model Training Subscriptions
main	5.5.3.2.1	Description
main	5.5.3.2.2	Resource definition
main	5.5.3.2.3	Resource Standard Methods
main	5.5.3.2.3.1	POST
main	5.5.3.2.4	Resource Custom Operations
main	5.5.3.3	Resource: Individual NWDAF ML Model Training Subscription
main	5.5.3.3.1	Description
main	5.5.3.3.2	Resource definition
main	5.5.3.3.3	Resource Standard Methods
main	5.5.3.3.3.1	PUT
main	5.5.3.3.3.2	PATCH
main	5.5.3.3.3.3	DELETE
main	5.5.3.3.4	Resource Custom Operations
main	5.5.3.3.4.1	Overview
main	5.5.3.3.4.2	Operation: unsubscribe-info
main	5.5.3.3.4.2.1	Description
main	5.5.3.3.4.2.2	Operation Definition
main	5.5.4	Custom Operations without associated resources
main	5.5.5	Notifications
main	5.5.5.1	General
main	5.5.5.2	Event Notification

文稿	章节	标题
main	5.5.5.2.1	Description
main	5.5.5.2.2	Operation Definition
main	5.5.6	Data Model
main	5.5.6.1	General
main	5.5.6.2	Structured data types
main	5.5.6.2.1	Introduction
main	5.5.6.2.2	Type NwdafMLModelTrainSubsc
main	5.5.6.2.3	Type NwdafMLModelTrainSubscPatch
main	5.5.6.2.4	Void
main	5.5.6.2.5	Type MLModelTrainInfo
main	5.5.6.2.6	Type MLTrainReportInfo
main	5.5.6.2.7	Type FailureEventInfoForMLModelTrain
main	5.5.6.2.8	Type NwdafMLModelTrainNotif
main	5.5.6.2.9	Void
main	5.5.6.2.10	Type DataAvReq
main	5.5.6.2.11	Type DelayEventNotif
main	5.5.6.2.12	Type StatusReportInfo
main	5.5.6.2.13	Type TrainDataInfo
main	5.5.6.2.14	Type TrainingUnsubscribeInfo
main	5.5.6.3	Simple data types and enumerations
main	5.5.6.3.1	Introduction
main	5.5.6.3.2	Simple data types
main	5.5.6.3.3	Enumeration: FailureCodeTrain
main	5.5.6.3.4	Enumeration: TermTrainCause
main	5.5.6.3.5	Enumeration: DelayCause
main	5.5.6.3.6	Enumeration: FLServerTermCause
main	5.5.7	Error handling
main	5.5.7.1	General
main	5.5.7.2	Protocol Errors
main	5.5.7.3	Application Errors
main	5.5.8	Feature negotiation
main	5.5.9	Security

源文步骤在文稿中定位到的协议目的：单训练提供者、单消费者、单训练任务；模型训练为不透明内部计算，先验证普通事件订阅/通知，不启用完整水平/垂直联邦学习。

对照文稿定性质

分析者在写查询之前，先把起草的性质清单与文稿本身对协议用途的说明做了对照。名称、性质针对什么、文稿怎么说，都是分析者的解读；其中点名的地方在文稿中都存在。

ID	简称	对象	文稿怎么说	出处
A03	通知绑定到自身订阅	保证 NF 服务消费者接受的每条通知都属于它自己创建的订阅，notifCorreId 相同，并且发往它自己的 notifUri。	notifCorreId 由消费者为所请求的通知分配，通知须携带相同的值（s0042、s0129、s0321）；NWDAF 向订阅中收到的 notifUri 发送（s0128）。	s0042, s0128, s0129, s0321
A08	通知上下文与订阅匹配	保证 NF 服务消费者所接受通知的事件标识符和轮次/上下文字段对应它所订阅的事件，而不是另一个订阅或另一个事件。	订阅用 mLEvent 描述所订阅的事件（s0038 至 s0040），通知为 notifCorreId 所标识的订阅报告 mLModelInfos 和 roundInd（s0131、s0136、s0321）。	s0038, s0039, s0040, s0131, s0136, s0321
D01	通知的真实性与完整性	保证 NF 服务消费者所接受的通知确实由包含 MTLF 的 NWDAF 为该订阅发送，并且其载荷在传输途中未被篡改。	NWDAF 应调用 Notify 操作通知所订阅的事件（s0128），消费者存储所接受的内容并应答 204（s0138）；文稿把消息保护留给 TS 29.500 和 TS 33.501 的传输层。	s0128, s0138
R02	每个通知实例仅接受一次	NF 服务消费者对同一订阅的同一逻辑通知至多接受一个实例。	文稿没有提到通知的重复或去重：notifCorreId 标识的是订阅，不是通知实例（s0129、s0321），而消费者只是存储通知（s0138）。	s0129, s0138, s0321
S05	终止后不再接受任何内容	在终止通知或有效退订之后，NF 服务消费者不再接受该订阅的任何通知。	termTrainReq 表示请求终止该订阅，并且不再提供与该订阅相关的任何通知（s0133、s0321）；退订会删除订阅（s0101 至 s0110）。	s0105, s0108, s0133, s0321
Z01	令牌覆盖的服务操作	保证 NWDAF 所接受的每个订阅操作都由 NRF 为 nnwdaf-mlmodeltraining 范围签发的令牌授权，消费者可以依赖这一点。	对 API 的访问可以用 OAuth2 客户端凭据授权，NRF 是授权服务器，消费者在消费该服务前应获取令牌（s0375、s0376）；该 API 定义了唯一的范围 nnwdaf-mlmodeltraining（s0378）。	s0375, s0376, s0378
Z03	令牌绑定到其订阅	一个 NF 服务消费者持有的令牌不能被用来操作另一个消费者的订阅或接收其通知。	该 API 定义了单一的服务级范围，在资源级或操作级没有额外范围（s0378），因此文稿把令牌绑定到服务，而不是绑定到订阅。	s0206, s0378
A03.8	notifCorreId 等于自身订阅	保证 NF 服务消费者所接受通知的通知关联标识符等于它确实发送的订阅请求中的 notifCorreId。	notifCorreId 由消费者分配，通知须携带与订阅相同的值（s0042、s0129、s0321）。	s0042, s0129, s0321
A08.9	订阅不混用	保证 NF 服务消费者所接受的通知及其载荷属于它自己的某个订阅，不会与另一个订阅混在一起。	notifCorreId 标识通知所涉及的订阅（s0321），mLModelInfos 报告所订阅的事件（s0131）。	s0129, s0131, s0321
U01	终止与结果区分	NF 服务消费者能区分终止通知与训练结果通知，并且通知绝不会同时携带 termTrainReq 和 mLModelInfos。	通知至少携带 delayEventNotif、mLModelInfos 或 termTrainReq 之一；delayEventNotif 与另外两者分别互斥，但 termTrainReq 和 mLModelInfos 并未声明互斥（s0321 NOTE 1）；通知没有显式的类型字段（s0128 至 s0137）。	s0128, s0131, s0133, s0321

ID	简称	对象	文稿怎么说	出处
X01	通知至少携带一个报告字段	NF 服务消费者对所存储通知体的良构性假设。	NwdafMLModelTrainNotif 定义的 NOTE 1 陈述了这条规则，并且只让 delayEventNotif 的两种组合互斥。	s0321

分析者依据文稿补充的性质：X01.

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 15 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 0 项，没设的 15 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	攻击者得不到更多	不诚实的 NF 服务消费者恰好持有服务范围授予每个已注册消费者的东西：它自己的令牌、它自己的订阅标识符和传输凭据。它从协议中学不到其他消费者的 notifCorreId、notifUri 或 subscriptionId，因此这里所问的性质都不会改变。
S02	NWDAF	参与方	攻击者得不到更多	不诚实的 NWDAF 只能通过消费者自己的订阅了解到该消费者的标识符，而训练模型的正确性不在任务范围内。这里所问的性质针对 NWDAF 确实发送的消息；不诚实的提供方发送它自己的通知并不会破坏这些性质。
S03	NRF	参与方	攻击者得不到更多	被攻陷的授权服务器可以为服务范围签发令牌，但它不持有用于保护消息的传输凭据，而且任何已注册消费者本来都能获得同一范围的令牌；这里所问的任何性质都不会只在它参与时才变得可回答。
S04	notifCorreId	值	攻击者得不到更多	notifCorreId 由消费者选择，并且只在受保护的消息中传输；持有它并不能让人在没有传输凭据的情况下使通知被接受。
S05	notifUri	值	攻击者得不到更多	notifUri 由消费者选择，并携带在受保护的订阅请求中；没有传输凭据，知道它也毫无用处。
S06	subscriptionId	值	攻击者得不到更多	在模型中，持有 subscriptionId 并不能让人操作该订阅：请求仍然需要有效令牌和受保护信道。它能让人做什么，已作为令牌范围目标的注意事项写明，因为该 API 没有定义资源级范围。
S07	mLEvent	值	攻击者得不到更多	mLEvent 是双方都合法知晓的公开事件标识符；它标识订阅了什么，而不标识谁可以接收。
S08	roundInd	值	不在范围内	轮次编号属于多轮训练状态，而任务边界将其排除在外；没有目标要求轮次的状态。
S09	mLModelInfos	值	攻击者得不到更多	没有目标、也没有文稿陈述要求训练模型的机密性；在模型中载荷受传输保护，其正确性不在范围内。
S10	termTrainReq	值	攻击者得不到更多	termTrainReq 是携带在受保护通知中的原因码；持有它并不能让人终止或继续某个订阅。
S11	通知载荷	值	攻击者得不到更多	通知载荷只有在消费者已将其关联到自身订阅的通知中才有意义；仅持有载荷不会破坏这里所问的任何性质。
S12	通知实例	值	攻击者得不到更多	除了订阅关联之外，没有任何东西标识通知实例，而这正是去重目标所问的内容；持有实例不会给出任何密钥材料。
S13	OAuth2 token	值	攻击者得不到更多	令牌只授权服务范围，而且每个已注册消费者都能获得同一范围的令牌；请求仍然需要传输凭据，因此持有令牌并不会增加什么。
S14	scope nnwdaf-mlmodeltraining	值	攻击者得不到更多	该范围是公开的，并且对该服务的每个消费者都相同。

ID	参与方或值	类别	攻击者条件	不设的原因
S15	failEventReports	值	不在范围内	failEventReports 属于 subscribe 操作的部分成功路径，而这一路径不属于所建模的正常流程。

每条性质在各攻击者条件下的答案

攻击者是分析者设定的，设定于写查询之前；每个答案都记录在其中一种攻击者之下。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”是指分析者为该性质设定、但证明器在该条件下使某项主张未决或没有记录任何内容的攻击者；空单元格是指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`	仅网络：它可以读取、丢弃、修改和发送消息，不持有任何一方的秘密		

ID	`network`
A03	成立
A08	成立
D01	成立
R02	不成立
S05	不成立
Z01	成立
Z03	成立
A03.8	成立
A08.9	成立
U01	不成立
X01	成立

原文和目标

Source: 379 units, 88264 bytes, frozen in source-delivery-01.json. Targets: 8 approved (A03, A08, D01, R02, S05, Z01, Z03, U01); the questions asked: 3. 目标清单事先按草稿批准，使用的是 --approve-targets。

宿主按其文档化的默认设置（任务 `decisions: auto`）所做的决定，供你核对：

- approve_package: 批准源文包 protocol-b33fef4f0dbf4eb9d2c66b76d4e6

本步骤自报：2 个请求（2 个失败），0.000 USD，输入 116167 tokens，输出 7679，耗时 169 s。

关于证明器和模型的说明

分析者为目标标注的内容：

step	targets	features	reason
49	A03, A08, A03.8, A08.9, D01, R02, S05, Z01, Z03, U01, X01	plain_primitives, many_sessions	所有答案只使用对称认证加密和用于 SBA/HTTP 层的 MAC，以及基于 MAC 的 OAuth2 令牌，会话数为任意多个消费者/NWDAF。协议不保留任何计数器、序列号或安装后会被替换的密钥，因此没有哪条性质依赖会变化的值；失败恰恰是因为缺少这样的值（没有通知标识符/序列号）。轮次编号（roundInd）本可以是这样的值，但它在任务范围之外，未建模。

分析者的结语

回答了十一个目标，其中六个被证明，三个被证伪，两项发现。

在模型上得到证明：消费者接受的每条通知都属于它自己创建的订阅，notifCorreId、notifUri 和 mLEvent 均匹配，并且确实由 NWDAF 原样发送（A03、A03.8、A08 针对事件标识符、A08.9、D01）；NWDAF 只接受持有 NRF 为 nnwdaf-mlmodeltraining 范围签发的令牌的订阅和取消订阅（Z01，两条主张，并附有一个对照，其中不检查的变体在没有验证令牌的情况下也会接受）；对订阅的操作只对消费者自己的订阅被接受（Z03）；每条被接受的通知至少携带一个报告字段（X01，前半部分）。

被证伪并作为攻击报告：同一条通知投递两次会被存储两次（R02）；通知被扣留、在终止请求或取消订阅之后才释放时，会被当作新通知接受（S05）；同时携带最终聚合模型的终止通知会被归类为普通训练结果（U01）。前两项是一项发现（没有任何东西标识通知实例或对消息流排序），第三项是第二项发现（通知没有类型字段，且 NOTE 1 使 termTrainReq 和 mLModelInfos 兼容）。

这些结论都基于一个模型：它将 SBA 传输视为经过认证的加密信道，并且只保留一个消费者、一个提供者，每个订阅至多两条通知。仍未解决的问题：订阅标识符的跨消费者使用（需要双消费者模型）、通知排序与去重（需要每订阅计数器，因此需要有界模型或 Tamarin+ Agent）、联邦学习场景的多轮状态与延迟通知，以及更新操作。