NWDAF 模型监测的安全性分析：注册、订阅与反馈通知

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	9 条：1 条成立，8 条不成立（10 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

5G 核心网的 Nnwdaf_MLModelMonitor 服务让一个 NWDAF 监测另一个 NWDAF 所用 ML 模型的 analytics 准确性。参与的有两个角色：包含 MTLF 的 NWDAF，它使用并训练该模型，希望被告知模型有多准确；以及包含 AnLF 的 NWDAF，它在 analytics 中运行该模型并观察其准确性。两者都持有由 NRF 签发的、属于同一服务范围的 OAuth2 访问令牌。

包含 AnLF 的 NWDAF 先在包含 MTLF 的 NWDAF 处注册该 ML 模型的使用与监测能力，并收到一个注册标识符。随后包含 MTLF 的 NWDAF 在包含 AnLF 的 NWDAF 处订阅准确性监测事件，给出 ML 模型标识符、其回调地址和通知关联标识符，并收到一个订阅标识符。包含 AnLF 的 NWDAF 把所订阅的事件通知到该回调地址，携带关联标识符、模型的准确性信息和 analytics 反馈，包含 MTLF 的 NWDAF 存储该通知并应答 204 No Content。任一方都可以对单个注册或订阅发起 DELETE 来结束这一安排。

参与方：

- **包含 MTLF 的 NWDAF**：使用并训练该 ML 模型，订阅其准确性。持有：范围为 nnwdaf-mlmodelmonitor 的 OAuth2 访问令牌。
- **包含 AnLF 的 NWDAF**：运行该模型，观察其准确性，发送通知。持有：范围为 nnwdaf-mlmodelmonitor 的 OAuth2 访问令牌；其在 analytics 中运行的 ML 模型。
- **NRF**：签发访问令牌。持有：授权服务器。

图 1 展示了文稿所规定的交互过程。

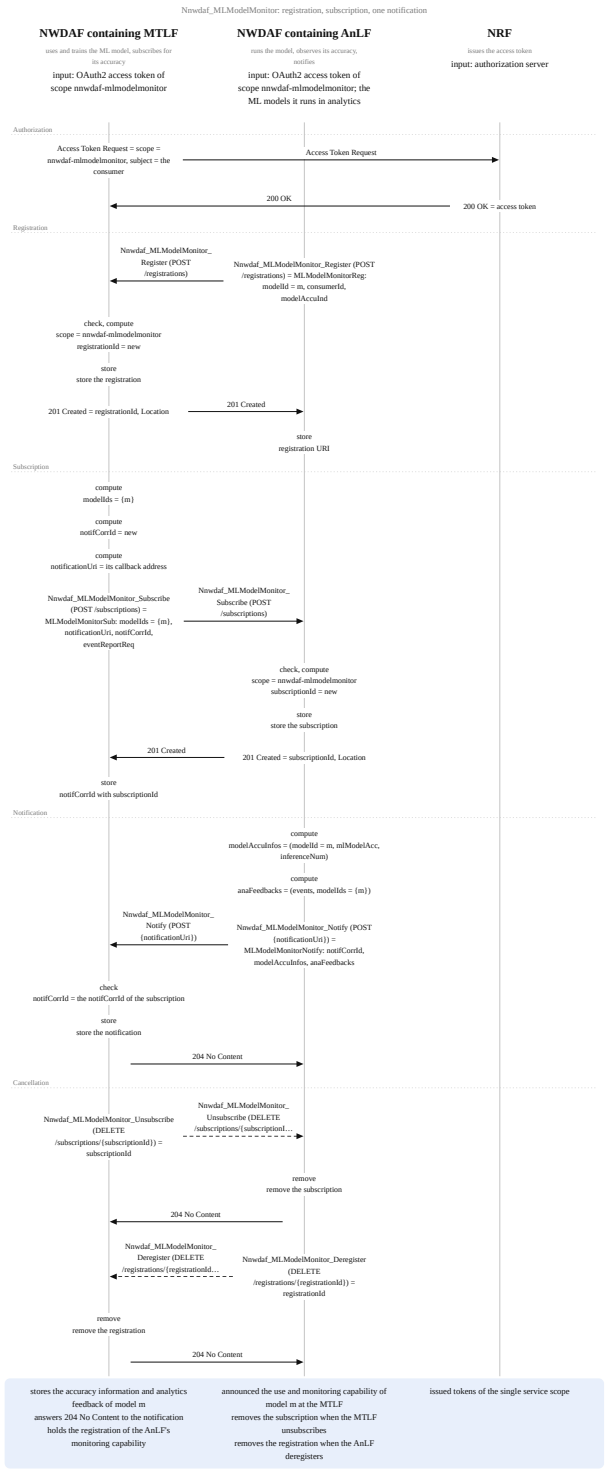


图 1. 文稿所规定的协议。

分析的范围见附录 D：分析了什么、省略了什么，以及它依据的文稿章节。

2 发现

本次分析有 2 项发现：以下内容文稿并未写明。其中 2 项是否由文稿写明并不明确。

该服务并未提供其通知流程本应提供的东西：通知仅凭一个由消费者自行选择并以明文发送的关联字符串就被接受，因为 notify 操作不携带 OAuth2 令牌，文稿也没有为它定义其他认证方式。因此，伪造的和重复的通知都会被包含 MTLF 的 NWDAF 存储，而其中携带的模型标识符从不与订阅进行比对。确实成立的

是：被接受的通知携带的是消费者自己选择的标识符；任何 Register 或 Subscribe 请求在没有该服务范围的令牌时都不会被处理；订阅被删除时生产者会替换其保存的值。这两处缺口文稿并未写明，因此属于发现；Z03 的单一范围授权缺口由 TS 29.501 第 4.10 条写明，属于复现而非发现。这些答案基于一个把 SBI 传输视为攻击者通道的模型，因此通知类攻击对处于 SBI 路径上的主体成立，并取决于 TLS 在何处终止；与状态相关的答案来自两个展开的监测区间。

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在规范交互在该处并无此消息时仍发送的消息，以及该攻击所破坏的结果。

2.1 通知仅凭关联字符串就被接受：路径上的主体可以注入准确性数据

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	中
涉及的性质	D01, A03, A03.8, A08, A08.9（见第 3 节）
原文	s0073, s0074, s0076, s0122, s0123, s0132, s0319
依据	ProVerif attack queries `payload`、`corr_bound`、`model_bound`、`fb_bound`；图 2、3、4、5 和 6

发生了什么。 包含 MTLF 的 NWDAF 仅凭通知关联标识符就接受通知。该标识符由消费者选择并在订阅请求中携带，而 TS 29.500 第 6.7.3 条规定 notify 操作在本版本中不使用 OAuth2 访问令牌；文稿也没有为通知定义签名或 MAC。能读取订阅请求的主体可以自行提交 MLModelMonitorNotify 报文体，消费者会存储它并应答 204。modelAccuInfos 和 anaFeedbacks 中的 modelIds 同样不与订阅进行比对，因此通知可以指向攻击者选择的另一个模型；文稿中唯一相关的要求是一条 NOTE，指出 mLEvent、mLEventFilter 和 tgtUe 应与 AnLF 注册的内容一致。

攻击者需要什么。 一个能读取订阅请求 HTTP 报文体、从而获知 notifCorrId 的位置，例如按 TS 29.500 第 6.7.2 和 6.10.9 条允许的方式终止 TLS 的 SCP，或者一个可被猜到的 notifCorrId。它不需要令牌，也不需要密钥。

攻击者能得到什么。 包含 MTLF 的 NWDAF 会存储包含 AnLF 的 NWDAF 从未上报过的模型的准确性信息和 analytics 反馈，并据此监测该模型、据此指示准确性要求是否满足。同一次运行还会把数据归到攻击者选择的另一个 ML 模型上。

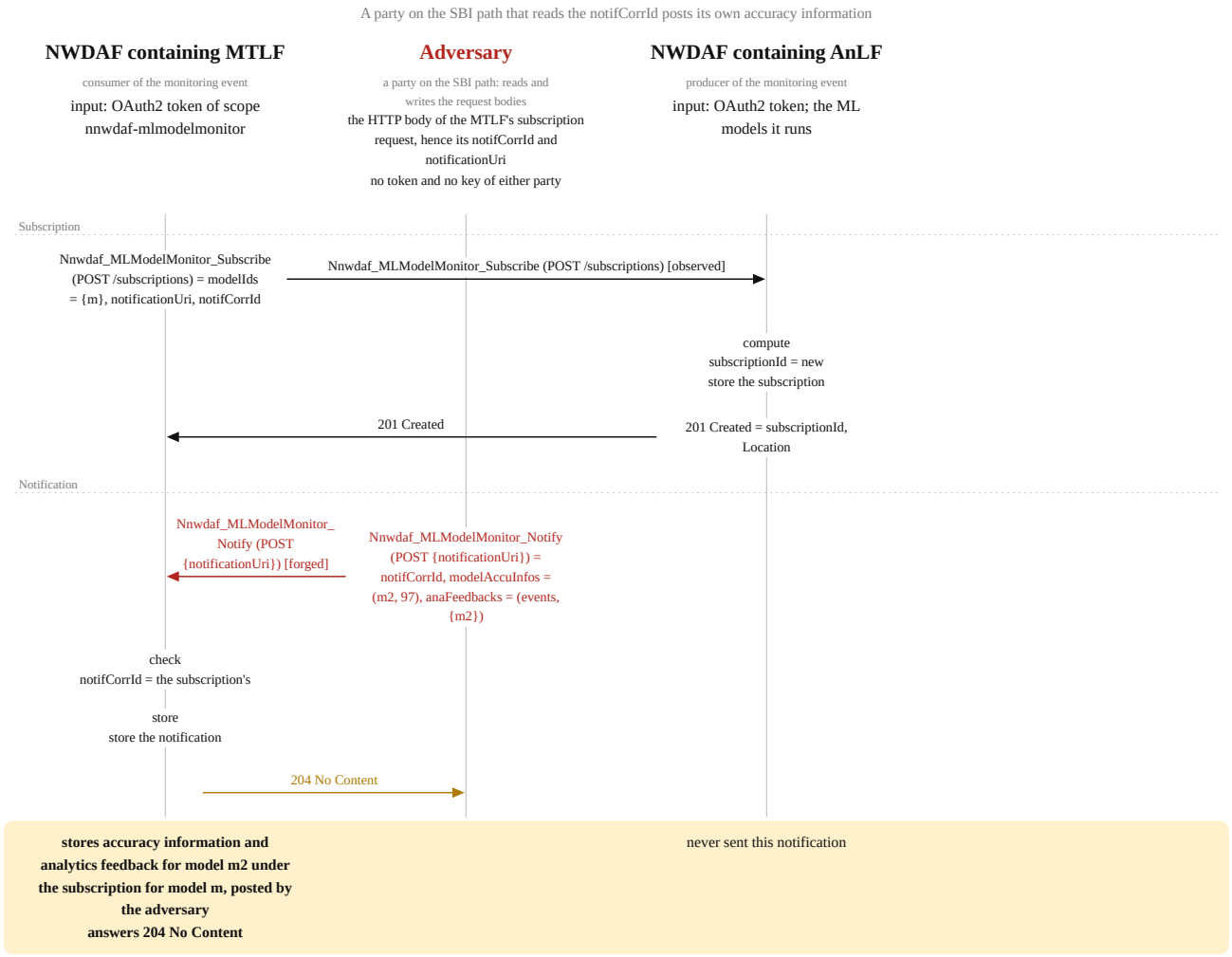


图 2. 处于 SBI 路径上、读取了 notifCorrId 的主体提交了自己的准确性信息。性质 D01；攻击查询 payload。

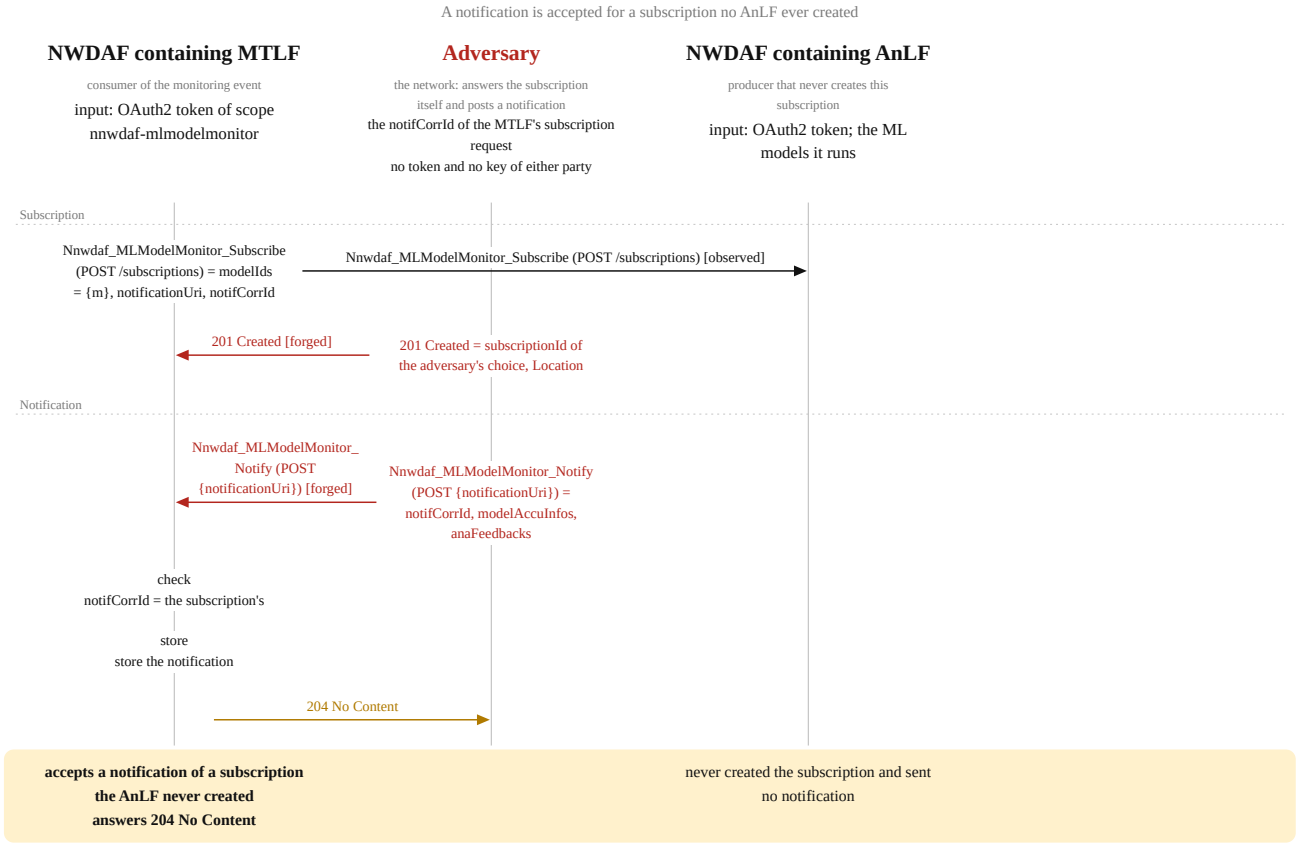


图 3. 对一个 AnLF 从未创建的订阅，通知被接受了。性质 A03；攻击查询 corr_bound。

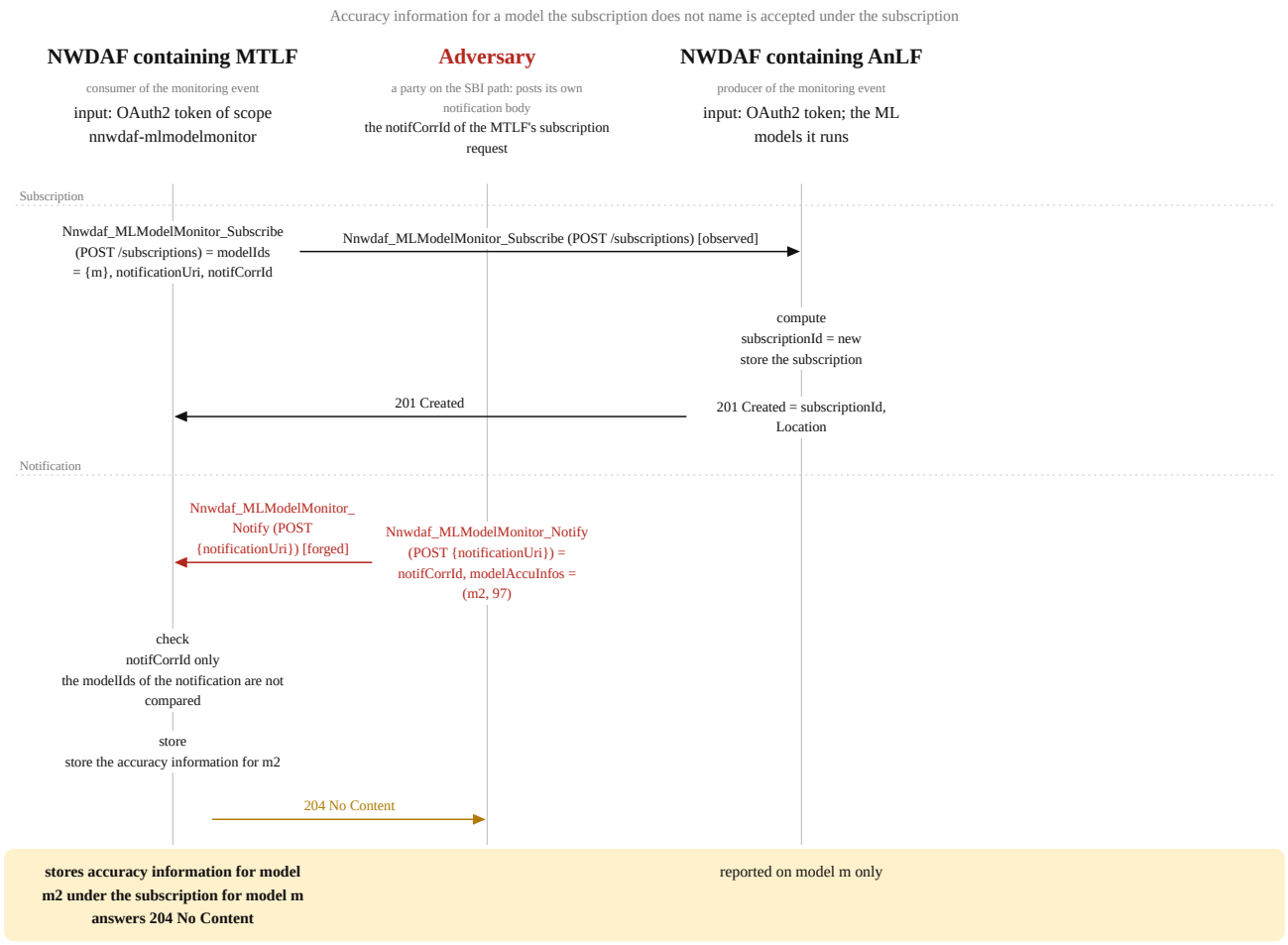


图 4. 订阅未指明的模型的准确性信息在该订阅下被接受。性质 A08；攻击查询 model_bound。

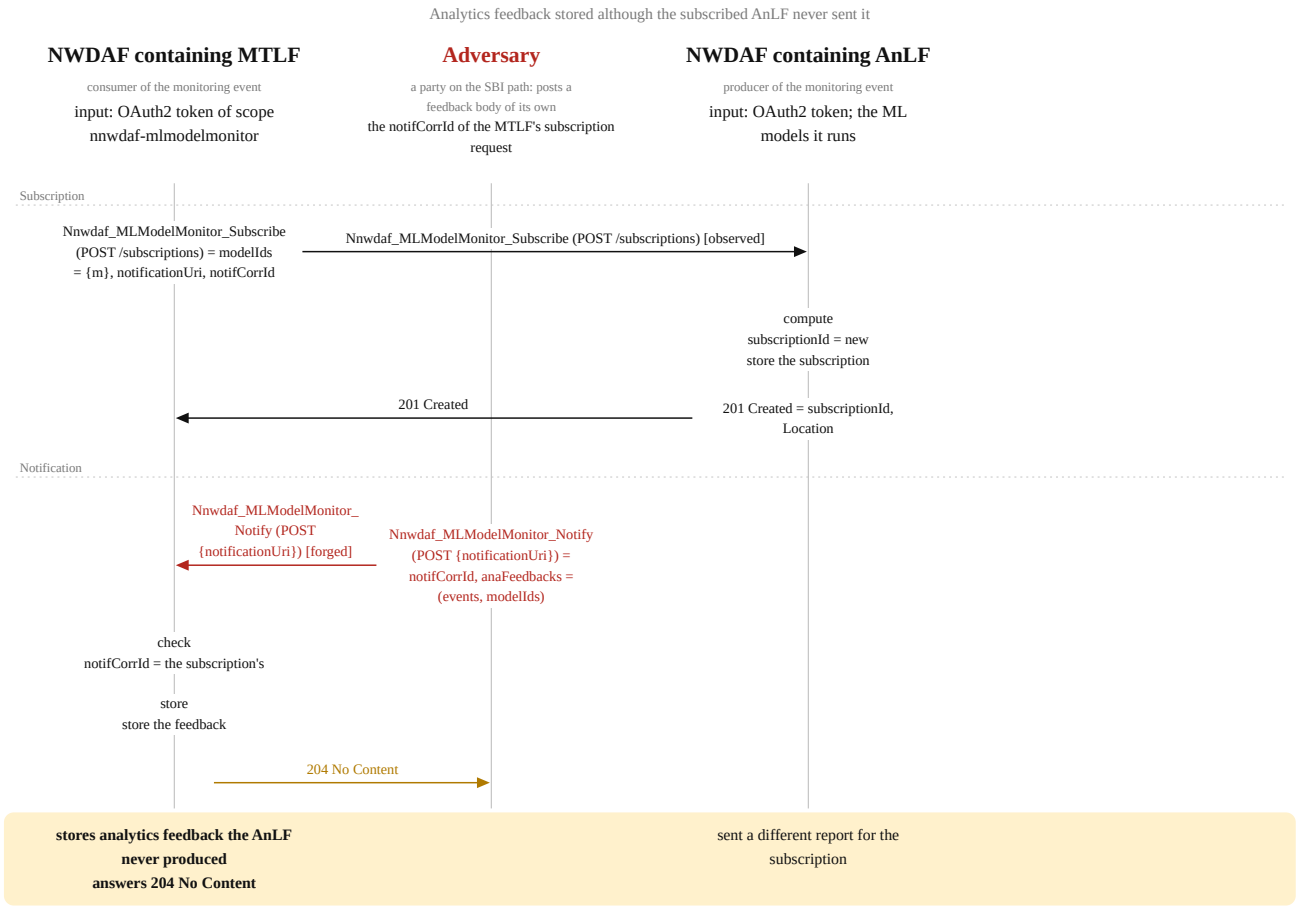


图 5. 所订阅的 AnLF 从未发送过的 analytics 反馈被存储了。性质 A03.8；攻击查询 payload。

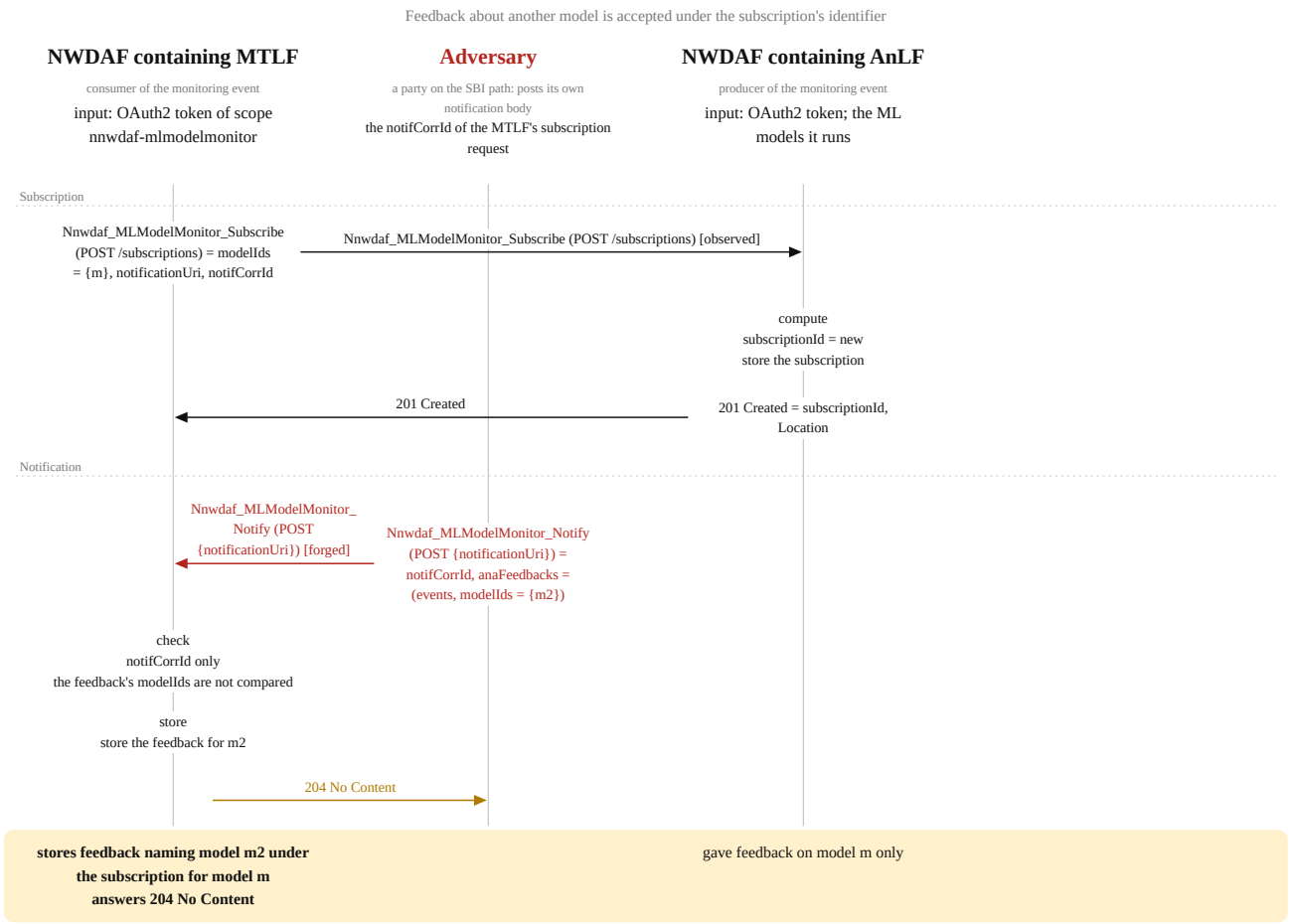


图 6. 关于另一个模型的反馈在该订阅的标识符下被接受。性质 A08.9；攻击查询 fb_bound。

拟议修改。 要求 notify 请求经过认证，确认其来自该订阅的 NF 服务生产者，并要求消费者拒绝发送方不是其所订阅生产者的通知。最省事的做法是为 notify 操作使用 OAuth2 访问令牌，如同 Subscribe 和 Unsubscribe，并写明消费者会将令牌主体与该订阅的生产者实例进行比对，因为 TS 29.500 第 6.7.3 条目前把 notify 操作排除在令牌使用之外。此外，要求消费者将 modelAccuInfos 和 anaFeedbacks 的 modelIds 与订阅的 modelIds 进行比对，而第 5.6.6.2.3 条中的 NOTE 目前只是建议这样做。

用 ProVerif 复核。 未检查：该修改处于请求内容的层面，但在本模型中无法证明其有效：模型把订阅请求本身交给了攻击者，因此其中携带的任何值都为攻击者所知，在通知上添加任何应用层内容在此都无济于事。要检查它，需要一个传输层能对攻击者隐藏该请求的模型，而这超出了本任务的边界。

2.2 已取消的订阅仍在接收通知，重复的通知会被再次存储

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	低
涉及的性质	Z04, A04（见第 3 节）
原文	s0109, s0110, s0112, s0123, s0132, s0338
依据	ProVerif attack queries `late_accepted`、`replayed`、`no_replay`；图 7 和 8

发生了什么。 对消费者而言，取消并不构成边界。在包含 MTLF 的 NWDAF 取消订阅并收到 204 No Content 应答后，它仍会把收到的通知与其已删除订阅的关联标识符进行匹配，并存储到达的那一条。同样地，当完全相同的报文体被第二次投递时，它已经存储过的通知会被再次存储，因为没有定义报告标识符，也没有定义重复检测。

攻击者需要什么。 对于在途的情形，除网络之外不需要任何东西：生产者自己已经发出的通知会在取消之后被存储。对于重复的情形，同一报文体必须被投递两次；只有在该重复是伪造而非复制时才需要 notifCorrId。

攻击者能得到什么。 模型的监测记录会为已取消的订阅保留取值，也会为同一份报告保留重复的取值，因此 MTLF 用来判断模型准确性的数据、以及关于哪些分析和模型被用于动作的反馈，可能已经过时或被重复计入。

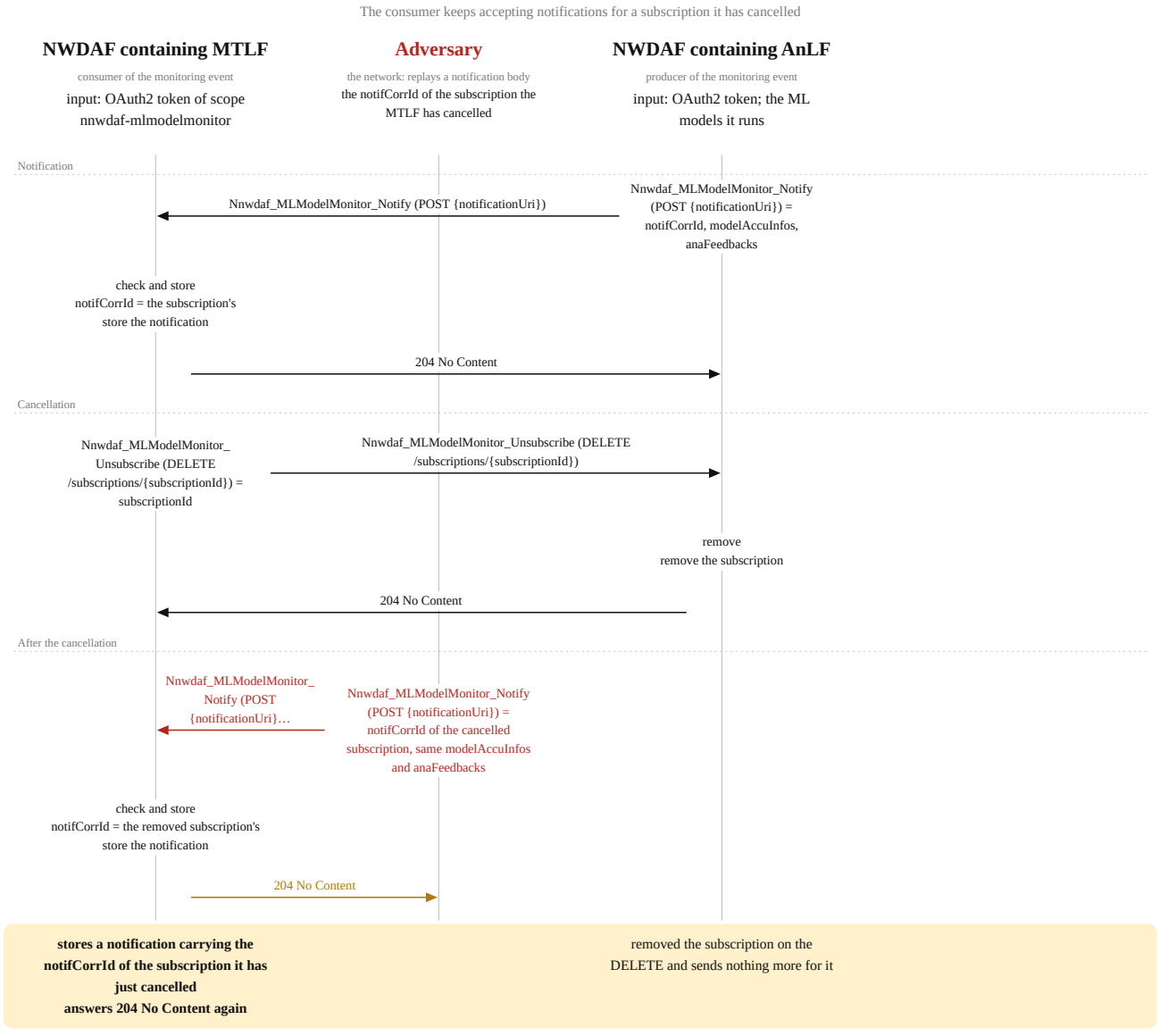


图 7. 消费者对已取消的订阅仍继续接受通知。性质 Z04；攻击查询 late_accepted。

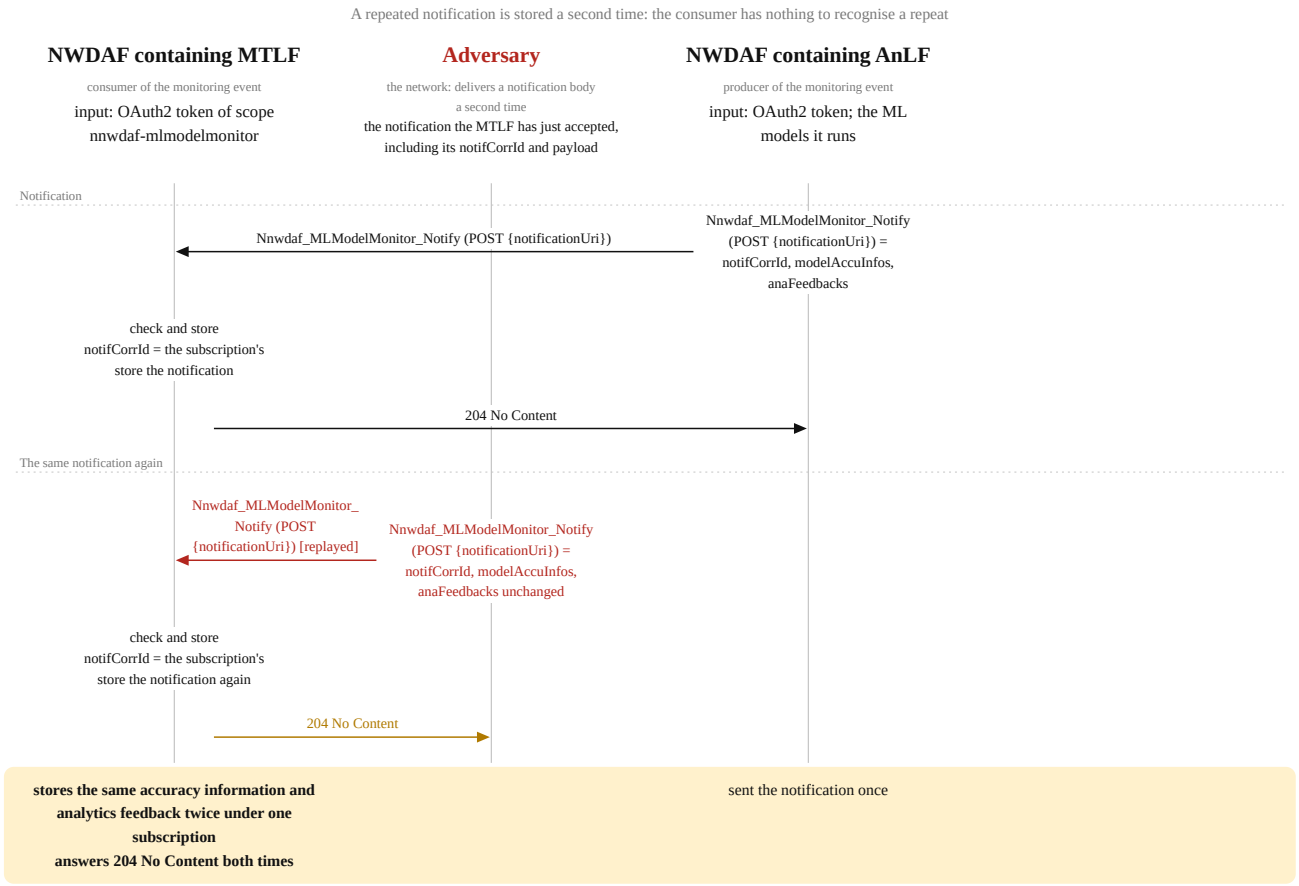


图 8. 重复的通知被再次存储：消费者没有任何手段识别重复。性质 A04；攻击查询 no_replay、replayed。

拟议修改。 让消费者在取消订阅时丢弃该订阅的 notifCorrId，并写明：对已不再活跃的订阅收到的通知予以丢弃、不予存储。为 MLModelMonitorNotify 增加一个标识符，或者写明 notifCorrId 与上报间隔一起即可标识一份报告，从而能够识别重复的报告。

用 ProVerif 复核。 未检查：该修改是一条关于消费者何时可以存储通知的规则，而检查它只能确认一个按要求丢弃的消费者确实会丢弃——这里的模型除了已经展开的两步之外，没有给消费者任何可供推理的历史。检查无法说明的是真实消费者是否实现了这种丢弃，因为文稿并未作此要求。

本次分析针对这些攻击未给出图：A03.8: corr_bound、A03.8: fb_bound、A03.8: model_bound、A03: fb_bound、A03: model_bound、A03: payload、A04: late_accepted、A08.9: corr_bound、A08.9: model_bound、A08.9: payload、A08: corr_bound、A08: fb_bound、A08: payload、D01: corr_bound、D01: fb_bound、D01: model_bound、Z04: no_replay、Z04: replayed。

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	**通知匹配真实订阅。** 每当 MTLF 接受一条通知，都存在一个 AnLF 的真实订阅运行，其订阅者身份、notificationUri、modelIds 与 notifCorrId 与所接受通知一致（非单射）。	无	攻击成立（文稿已写明）	中	是

ID	性质	假定的泄露	结果	价值	文稿是否已写
A04	**每次订阅运行仅一次接受。** 每当 MTLF 接受一条通知，都存在一个不同的 AnLF 订阅运行与之匹配（单射），同一订阅运行不被两次接受对应。	无	不成立（3 个攻击）	低	不确定
A08	**通知字段绑定到订阅。** MTLF 接受的通知中 notifCorrId、modelIds 与反馈内容对应其订阅请求中获批的取值，不被替换或错接。	无	不成立（1 个攻击）	中	不确定
D01	**通知来自所订阅的AnLF。** MTLF 接受的每条通知都对应允许发送方（订阅的 AnLF）的真实发送，载荷与关联字段一致。	无	攻击成立（文稿已写明）	中	是
Z03	**令牌授权主体和操作。** 使用某令牌提交的注册/订阅/通知操作，其主体、受众与操作与该令牌签发时的授权一致。	无	攻击成立（文稿已写明）	低	是
Z04	**取消后不生效。** 在注册被删除或订阅被取消生效之后，不再产生基于该注册/订阅的新通知接受或监测操作。	无	不成立（1 个攻击）	中	不确定
A08.7	**关联和URI匹配订阅。** MTLF 接受的通知中 notifCorrId、notificationUri 与 modelIds 对应其订阅请求中的取值。	无	成立	低	否
A03.8	**接受的反馈匹配真实发送。** 每当 MTLF 接受一条反馈通知，都存在一个 AnLF 的真实发送运行与之匹配。	无	攻击成立（文稿已写明）	中	是
A08.9	**反馈未错误连接到模型。** MTLF 接受的通知中反馈内容所属的 modelIds 与 notifCorrId 对应其订阅请求，不被跨模型或跨订阅错接。	无	不成立（1 个攻击）	中	不确定

“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）。

提出的问题，以及分析者的回答：

问题	回答	依据
Q1: notifCorrId 与订阅者、回调 URI、modelIds 的一致关联	消费者确实检查关联标识符和自身的回调地址，因此被接受的通知总是携带 MTLF 自己选择的 notifCorrId。modelIds 不被检查：命名了另一个 ML 模型的通知仍会以该订阅的名义被接受；又因为关联标识符不是秘密，与生产方订阅的一致性同样不成立。	A08.7, A03, A08
Q2: 接受反馈对应相应 AnLF 的发送事件	否。被接受的通知不必对应包含 AnLF 的 NWDAF 的一次发送：知道 notifCorrId 的一方可以自行提交通知，而 MTLF 会将其作为监测数据存储。	D01, A03.8
Q3: 跨模型/跨订阅的反馈不能错误归属（作为待验证需求，先核对接收端检查规则）	否。API 中没有任何内容要求消费者将 modelAccuInfos 或 anaFeedbacks 的 modelIds 与其订阅进行比对，因此关于另一个模型的反馈会以该订阅的关联标识符被接受；唯一相关的文字是一条 NOTE，说明 mLEvent、mLEventFilter 和 tgtUe 应与注册匹配。	A08.9, A08

4 本次分析的局限

这里的结果是针对协议模型的结果：它说明所规定的消息和检查在密码学视为理想的情况下能提供或不能提供什么，与具体实现无关。

按分析者的说法，本次分析不覆盖的内容如下：

- SBI 传输被建模为攻击者的信道：TLS（TS 29.500 6.7.2）以及 OAuth2 令牌的签发与验证（TS 33.501、TS 29.510）均未建模，只建模了令牌的范围检查。

- 部署是否会让路径上的元素读取 HTTP 正文（例如 TLS 在 SCP 处终止的情形），这一点保持开放；针对通知的攻击是作为依赖该不明确性提出的。
- 准确性取值、阈值、模型指标和反馈内容都是不透明数据；不分析它们的含义或正确性。
- 注销和取消订阅只覆盖了一个实例：注册与订阅跨 NF 实例的一致性不在范围内，如任务所述。
- 每个进程只建模一个 ML 模型和一个消费者身份；关于逐模型授权的回答来自文稿中的单一范围，而不是来自 NRF 的模型。
- 消费者在取消订阅后的行为被建模为：只要携带旧的 notifCorrId 就予以存储；文稿没有规定任何规则，因此该攻击依据的是规则的缺失，而不是某个被建模的检查。
- bounded.pv 覆盖一个订阅的两个监测间隔；更长的历史以及 Z04 的注销一侧未展开。
- 任何回答中都不涉及代数运算（异或、Diffie-Hellman）。

宿主记录的内容：

- A04、Z04 的回答依赖于模型所做的一处简化（附录 D）。

每条性质还有哪些未决之处、什么能解决它，见附录 A。

附录 A：逐条性质的解读

以下为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、解读依据什么、能否照原样使用，以及它的价值。

A03: 通知匹配真实订阅

每当 MTLF 接受一条通知，都存在一个 AnLF 的真实订阅运行，其订阅者身份、notificationUri、modelIds 与 notifCorrId 与所接受通知一致（非单射）。

MTLF 会接受针对 AnLF 从未创建过的订阅的通知：它只匹配自己选择的关联标识符，而该标识符在订阅请求中以明文传输。因此，只要有一方能够读取该请求，该性质的存在性一半——被接受的通知属于真实的订阅运行——就不成立。

- 结果：攻击成立（文稿已写明）；是否回答了所问：部分；可在有保留的情况下使用
- 价值中：该轨迹即第一项发现中的伪造通知运行；文稿写明了机制（notify 不使用令牌），但没有写明通知可以来自另一方。
- 文稿是否已写：是。文稿所在位置：s0076, s0123, s0132
- 依据：SBI 正文可被攻击者读取；模型中没有 TLS。；订阅正文不携带消费者身份，因此订阅者取自令牌主体。；只建模一个 ML 模型实例，准确性取值不透明。
- 未决：该攻击需要 notifCorrId，因此它适用于位于 SBI 路径上的一方，而不适用于仅掌握网络的攻击者。
- 下一步：建立一个传输对攻击者隐藏订阅请求的模型，可以区分这两种解读；或在规范中明确消费者除 notifCorrId 之外还必须检查什么。

A04: 每次订阅运行仅一次接受

每当 MTLF 接受一条通知，都存在一个不同的 AnLF 订阅运行与之匹配（单射），同一订阅运行不被两次接受对应。

一条通知可以被接受多次，而订阅持续有效，因此多次接受属于同一个订阅运行。该目标的单射解读并不是本协议的性质；重要的是重复会被再次存储，以及消费者没有标识符来区分重复和新报告。

- 结果：不成立（3 个攻击）；回答所问：部分；可用但有注意事项
- 值低：重复的监测数据在这里是数据质量问题，原文既未禁止重复，也未要求检测。
- 文稿中的陈述：不明确。所依据的原文：s0123、s0132、s0338
- 依据：bounded.pv 将消费者展开为两次接受，将生产者展开为两个间隔。；只有当 notifCorrId 是编造而非复制时，重复才需要它。
- 未决：重复的通知是否应被拒绝，文稿并未确定。
- 下一步：至多一次的问题需要在规范中给出规则，例如报告标识符，或明确规定取消后丢弃；应从目标中删除单射性解读。

A08: 通知字段绑定到订阅

MTLF 接受的通知中 notifCorrId、modelIds 与反馈内容对应其订阅请求中获批的取值，不被替换或错接。

准确度信息中携带的 modelIds 不与订阅进行比对，因此关于另一个 ML 模型的通知会在该订阅的关联标识符下被接受。文稿不要求进行这种比对；只有一个 NOTE 建议相关字段应与注册信息匹配。

- 结果：不成立（1 个攻击）；回答所问：否；可用但有注意事项
- 值中：缺少 modelIds 检查是要求消费者执行的检查中的真实缺口，文稿并未说明这一点。
- 文稿中的陈述：不明确。所依据的原文：s0074、s0125、s0319
- 依据：攻击者读取订阅请求以获知 notifCorrId。；准确度取值是不透明的；只比对模型标识符。
- 未决：消费者是否应检查 modelIds 留给实现决定。
- 下一步：一个隐藏订阅请求的传输模型，或消费者显式地将 modelIds 与订阅进行比对，将能确定哪些模型标识符可以出现。

D01: 通知来自所订阅的AnLF

MTLF 接受的每条通知都对应允许发送方（订阅的 AnLF）的真实发送，载荷与关联字段一致。

被接受的通知并不对应所订阅 AnLF 的一次发送：知道关联标识符的一方提交自己的正文，消费者将其存储。文稿对 notify 操作提供的唯一保护是传输层，因为没有为其使用任何令牌。

- 结果：攻击成立（文稿中有陈述）；回答所问：否；可用但有注意事项
- 值中：这是第一项发现的核心缺口，TS 29.500 第 6.7.3 条指出 notify 操作不使用令牌，却未说明其代价。
- 文稿中的陈述：是。文稿中的陈述位置：s0117、s0122、s0132
- 依据：未对 notify 请求的应用层保护建模，因为文稿未定义任何此类保护。；攻击者读取订阅请求；模型在 SBI 上把每个正文都给了它。
- 未决：这里没有任何机制能区分合法的在途通知与注入的通知；模型表明两者都会被接受。
- 下一步：需要经过认证的 notify 请求，或使用不在订阅中携带的密钥计算的 MAC，才能使其成立；后者超出本模型所能展示的范围。

A08.7：关联和URI匹配订阅

MTLF 接受的通知中 notifCorrId、notificationUri 与 modelIds 对应其订阅请求中的取值。

被接受通知的关联标识符和回调地址就是 MTLF 自己使用的那些，因此消费者对标识符的检查按规范工作。这就是协议定义的全部绑定。

- 结果：成立；回答所问：是；可用
- 值低：这是规范所述的机制，在其成立之处得到了证明；对照查询表明，当加入一个消费者不检查的字段时，同一查询会失败。
- 文稿中的陈述：否。所依据的原文：s0075、s0076、s0123
- 依据：回调地址是消费者自己的端点，因此真正被比对的只有 notifCorrId。；模型未考虑在多个订阅间复用一个标识符的消费者。
- 下一步：无需进一步工作；就模型所能询问的范围而言，该性质成立。

A03.8：接受的反馈匹配真实发送

每当 MTLF 接受一条反馈通知，都存在一个 AnLF 的真实发送运行与之匹配。

MTLF 存储的分析反馈不一定由 AnLF 产生：知道关联标识符的一方提交自己的反馈正文，消费者将其存储。因此模型的反馈记录可以被生产者以外的人填充。

- 结果：攻击成立（文稿中有陈述）；回答所问：否；可用但有注意事项
- 值中：与第一项发现相同的缺口，针对反馈载荷陈述；文稿未说明这一点。

- 文稿中的陈述：是。文稿中的陈述位置：s0117、s0126、s0132
- 依据：未对通知的发送方认证建模，因为文稿未定义任何此类认证。；攻击者读取订阅请求。
- 未决：是否应拒绝来自 AnLF 的在途通知，尚未确定。
- 下一步：经过认证的 notify 请求是能使其成立的修改。

A08.9：反馈未错误连接到模型

MTLF 接受的通知中反馈内容所属的 modelIds 与 notifCorrId 对应其订阅请求，不被跨模型或跨订阅错接。

命名非订阅模型的反馈会在该订阅的标识符下被接受，因此监测数据可能被归因到错误的 ML 模型。API 中没有任何内容要求消费者比对两组模型标识符。

- 结果：不成立（1 个攻击）；回答所问：否；可用但有注意事项
- 值中：缺少 modelIds 检查是文稿未说明的缺口；只有一个 NOTE 暗示了这一点。
- 文稿中的陈述：不明确。所依据的原文：s0074、s0126、s0328
- 依据：攻击者读取订阅请求。；反馈内容是不透明的；只对模型标识符建模。
- 未决：文稿仅在 NOTE 中建议相关字段与生产者注册的内容匹配。
- 下一步：要求消费者将反馈的 modelIds 与订阅进行比对，并对发送方进行认证，将能解决此问题。

Z03: 令牌授权主体和操作

使用某令牌提交的注册/订阅/通知操作，其主体、受众与操作与该令牌签发时的授权一致。

只有持有服务范围令牌的请求才会对 Register 或 Subscribe 请求执行操作，而这正是授权的全部：API 为整个服务定义了一个范围，因此任何持有该范围的 NF 都可以为任何 ML 模型注册和订阅，包括另一个 NF 使用的模型。文稿自身陈述了这一后果。

- 结果：攻击成立（文稿中有陈述）；回答所问：是；可用
- 值低：文稿明确陈述了这一点：只有服务级范围的 API 仅将所有操作授予所有被允许访问该服务的消费者。
- 文稿中的陈述：是。文稿中的陈述位置：ts-29-501:1801、main:19224、s0344、s0345、s0347
- 依据：令牌被建模为不可伪造的项，其范围由生产者检查；签发、过期和受众未建模。；恶意 NF 获得了它的令牌，每个 NF 都从 NRF 获取令牌；这不是泄露的秘密。
- 下一步：授权结构方面无需进一步工作；单范围答案就是 TS 29.501 第 4.10 条给出的结论。

Z04: 取消后不生效

在注册被删除或订阅被取消生效之后，不再产生基于该注册/订阅的新通知接受或监测操作。

在包含 MTLF 的 NWDAF 取消订阅并收到 204 应答之后，它仍然接受并存储携带已移除订阅标识符的通知，重复的通知会被再次存储。生产者侧则相反：一旦 DELETE 被处理，AnLF 就不再为该订阅发送任何内容，这取决于它所保持的值被替换。

- 结果：不成立（1 个攻击）；回答所问：部分；可用但有注意事项
- 值中：生命周期缺口真实存在且未被说明，但其影响是过时或重复的监测数据，而非机密性丧失；该性质的生产者侧确实成立。
- 文稿中的陈述：不明确。所依据的原文：s0109、s0111、s0112、s0132、s0338
- 依据：展开的模型覆盖一个订阅的两个监测间隔；注销侧未展开。；消费者继续匹配旧的 notifCorrId 是因为文稿未陈述任何规则而做出的建模决定。；消费者侧攻击需要 notifCorrId；合法的在途通知也会被接

受，没有任何查询能将其隔离。

- 未决：订阅取消后是否必须丢弃通知，文稿并未确定。；NF 取消另一个消费者订阅的解读在状态复核中记录为未确认的候选。
- 下一步：展开注销侧并给出在途通知的规则将回答其余问题；在恶意 NF 下，答案需要一个模型，其中 DELETE 携带调用者身份，而订阅正文不提供该身份。

附录 C：主张和证明器的答案

证明器为每个目标记录：达到的层级、达到该层级的查询、是否检查了对照，以及每个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		corr_own	checked	corr_bound (depends_on_ambiguity)	1 proved, 1 attack
A04	reachable		reach_accepted		no_replay (depends_on_ambiguity); inj_sub (holds_in_source); replayed (holds_in_source)	3 attack
A08	reachable		reach_accepted		model_bound (depends_on_ambiguity)	1 attack
D01	reachable		reach_accepted		payload (depends_on_ambiguity)	1 attack
Z03	proved		no_bad_scope_sub	checked	rogue_sub (holds_in_source)	2 proved, 1 attack
Z04	bounded	变化值展开到 2 步	late_sent	checked	late_accepted (holds_in_source)	1 bounded, 1 attack
A08.7	proved		corr_own	checked		1 proved
A03.8	reachable		reach_accepted		payload (depends_on_ambiguity)	1 attack
A08.9	reachable		reach_accepted		fb_bound (depends_on_ambiguity)	1 attack

逐条列出主张，按分析者对各条主张的措辞；每条的回答来自证明器：

目标	主张	攻击者	回答	查询	对照
A03	一条被接受的 notification 的 notifCorrId 和 notificationUri 就是 MTLF 自己发出的订阅请求中的那对值。	on_path	proved	corr_own	checked
A03	被接受的 notification 属于一个确实在 AnLF 上创建的订阅流程，其订阅者身份、notificationUri、modelIds 和 notifCorrId 都与该流程一致。攻击者除网络之外还掌握 SBI 请求的报文主体（它能读到订阅请求以及发往 notificationUri 的 POST）。	on_path	attack (depends_on_ambiguity)	corr_bound	
A04	同一条 notification 不会被接受两次：每次接受都对应 AnLF 的一次不同发送。攻击者除网络之外还掌握 SBI 请求的报文主体。	on_path	attack (depends_on_ambiguity)	no_replay	
A04	每次接受都对应一个不同的订阅流程，因此一个订阅流程不会对应两次接受。	on_path	attack (holds_in_source)	inj_sub	
A04	同一条 notification 不会被接受两次：重复的 notification 不会在同一订阅下再次存储。	on_path	attack (holds_in_source)	replayed	
A08	被接受的 notification 中准确度信息的 modelIds 就是 MTLF 订阅时的 modelIds，没有被替换。攻击者除网络之外还掌握 SBI 请求的报文主体。	on_path	attack (depends_on_ambiguity)	model_bound (the control of corr_own)	
D01	被接受的 notification 对应 MTLF 所订阅的、包含 AnLF 的 NWDAF 的一次真实发送，其 payload 和关联字段都一致。攻击者除网络之外还掌握 SBI 请求的报文主体。	on_path	attack (depends_on_ambiguity)	payload	

目标	主张	攻击者	回答	查询	对照
Z03	针对某个 ML 模型的 Register 或 Subscribe 操作，只有在令牌的授权覆盖该主体对该模型的权限时才会执行。攻击者除网络之外还持有一个有效的、单一服务范围的 OAuth2 访问令牌。	`rogue_nf`	attack (holds_in_source)	rogue_sub (the control of no_bad_scope)	
Z03	只有当 Subscribe 请求携带的 OAuth2 访问令牌的范围是单一服务范围 nnwdaf-mlmodelmonitor 时，该请求才会被执行。	`network`	proved	no_bad_scope_sub	checked
Z03	只有当 Register 请求携带的 OAuth2 访问令牌的范围是单一服务范围 nnwdaf-mlmodelmonitor 时，该请求才会被执行。	`network`	proved	no_bad_scope	checked
Z04	在 MTLF 取消订阅之后（Unsubscribe 返回 204），基于该订阅的任何 notification 都不再被接受。	`on_path`	attack (holds_in_source)	late_accepted	
Z04	在 AnLF 处理完删除某订阅的 DELETE 之后，它不再为该订阅发送任何 notification。	`rogue_nf`	bounded (the changing value unrolled to 2 steps)	late_sent	checked
A08.7	被接受的 notification 的 notifCorrId 和 notificationUri 就是 MTLF 自己订阅请求中的那对值。	`on_path`	proved	corr_own	checked
A03.8	被接受的反馈 notification 对应被订阅的 AnLF 的一次真实发送，因此 MTLF 存储的分析反馈和准确度信息确实由该生产者产生。攻击者除网络之外还掌握 SBI 请求的报文主体。	`on_path`	attack (depends_on_ambiguity)	payload	
A08.9	被接受的 notification 中分析反馈的 modelIds 就是 MTLF 订阅时的 modelIds，因此反馈不会被归到另一个模型上。攻击者除网络之外还掌握 SBI 请求的报文主体。	`on_path`	attack (depends_on_ambiguity)	fb_bound	

附录 D：分析是怎么跑的

LLM 分析者根据文稿构建协议模型，并把每条性质交给 ProVerif 验证；第 3 节的每个结果都是证明器给出的回答。

ProVerif+ Agent 会话 session-20261114T182230，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：57，证明器耗时 5 秒，实际耗时 17 分钟，费用 0.136 美元。

ProVerif：运行了 92 个证明，92 个已有结论，0 个无答案，耗时 3 秒，记录了 27 个回答。

分析了什么

协议	NWDAF 模型监测：注册、订阅与反馈通知
版本	19.8.0
分析范围	固定 MTLF/AnLF 两角色；先做固定 modelIds 的监测订阅与一次反馈，注册关系单独建模，不能无依据假定每次订阅必先注册。准确度值作不透明数据。
不在范围内	不验证准确度数值真假、模型精度、收敛或阈值统计；API 把部分检查留给实现/策略时，不能擅自加检查后声称原协议安全；撤销注册和取消订阅的全局一致性另外处理。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	347 个带编号的原文单元（63012 字节），取自下列各节。本报告中的 s0001 这类引用就是这些原文单元之一

构建该分析所依据的文稿各节如下：

文稿	章节	标题
main	4.7	Nnwdaf_MLModelMonitor Service
main	4.7.1	Service Description
main	4.7.1.1	Overview
main	4.7.1.2	Service Architecture
main	4.7.1.3	Network Functions
main	4.7.1.3.1	Network Data Analytics Function (NWDAF)
main	4.7.1.3.2	NF Service Consumers
main	4.7.2	Service Operations
main	4.7.2.1	Introduction
main	4.7.2.2	Nnwdaf_MLModelMonitor_Register service operation
main	4.7.2.2.1	General
main	4.7.2.2.2	Registering the monitoring of the analytics accuracy of an ML Model
main	4.7.2.3	Nnwdaf_MLModelMonitor_Deregister service operation
main	4.7.2.3.1	General
main	4.7.2.3.2	Deregistering the monitoring of the analytics accuracy of an ML Model
main	4.7.2.4	Nnwdaf_MLModelMonitor_Subscribe service operation
main	4.7.2.4.1	General

文稿	章节	标题
main	4.7.2.4.2	Subscription for monitoring notifications
main	4.7.2.4.3	Update of subscription for monitoring notifications
main	4.7.2.5	Nnwdaf_MLModelMonitor_Unsubscribe service operation
main	4.7.2.5.1	General
main	4.7.2.5.2	Unsubscribe from monitoring notifications
main	4.7.2.6	Nnwdaf_MLModelMonitor_Notify service operation
main	4.7.2.6.1	General
main	4.7.2.6.2	Notification about subscribed event
main	5.6	Nnwdaf_MLModelMonitor Service API
main	5.6.1	Introduction
main	5.6.2	Usage of HTTP
main	5.6.2.1	General
main	5.6.2.2	HTTP standard headers
main	5.6.2.2.1	General
main	5.6.2.2.2	Content type
main	5.6.2.3	HTTP custom headers
main	5.6.3	Resources
main	5.6.3.1	Resource Structure
main	5.6.3.2	Resource: NWDAF ML model monitoring registrations
main	5.6.3.2.1	Description
main	5.6.3.2.2	Resource Definition
main	5.6.3.2.3	Resource Standard Methods
main	5.6.3.2.3.1	POST
main	5.6.3.2.4	Resource Custom Operations
main	5.6.3.3	Resource: Individual NWDAF ML model monitoring registration
main	5.6.3.3.1	Description
main	5.6.3.3.2	Resource definition
main	5.6.3.3.3	Resource Standard Methods
main	5.6.3.3.3.1	DELETE
main	5.6.3.3.4	Resource Custom Operations
main	5.6.3.4	Resource: NWDAF ML model monitoring Subscriptions
main	5.6.3.4.1	Description
main	5.6.3.4.2	Resource Definition
main	5.6.3.4.3	Resource Standard Methods
main	5.6.3.4.3.1	POST
main	5.6.3.4.4	Resource Custom Operations

文稿	章节	标题
main	5.6.3.5	Resource: Individual NWDAF ML model monitoring Subscription
main	5.6.3.5.1	Description
main	5.6.3.5.2	Resource definition
main	5.6.3.5.3	Resource Standard Methods
main	5.6.3.5.3.1	PUT
main	5.6.3.5.3.2	DELETE
main	5.6.3.5.4	Resource Custom Operations
main	5.6.4	Custom Operations without associated resources
main	5.6.5	Notifications
main	5.6.5.1	General
main	5.6.5.2	Event Notification
main	5.6.5.2.1	Description
main	5.6.5.2.2	Operation Definition
main	5.6.6	Data Model
main	5.6.6.1	General
main	5.6.6.2	Structured data types
main	5.6.6.2.1	Introduction
main	5.6.6.2.2	Type MLModelMonitorReg
main	5.6.6.2.3	Type MLModelMonitorSub
main	5.6.6.2.4	Type MLModelMonitorNotify
main	5.6.6.2.5	Type MLModelAccuracyInfo
main	5.6.6.2.6	Type AnalyticsFeedback
main	5.6.7	Error handling
main	5.6.7.1	General
main	5.6.7.2	Protocol Errors
main	5.6.7.3	Application Errors
main	5.6.8	Feature negotiation
main	5.6.9	Security

源文步在文稿中定位到的协议目的：固定 MTLF/AnLF 两角色；先做固定 modelIds 的监测订阅与一次反馈，注册关系单独建模，不能无依据假定每次订阅必先注册。准确度值作不透明数据。

对照文稿定性质

在撰写查询之前，分析者把草拟的性质清单与文稿本身对协议目的的说明做了对照。性质的名称、性质所针对的对象以及文稿给出的说法，均属分析者的解读；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
A03	通知匹配真实订阅	含 MTLF 的 NWDAF 得到保证：它接受的每条通知都属于它在含 AnLF 的 NWDAF 上发起的一次订阅运行，且订阅者身份、notificationUri、modelIds 和 notifCorrId 均相同；非单射。	消费者在订阅请求中发送 modelIds、notificationUri 和 notifCorrId（s0074、s0075、s0076），AnLF 携带 notifCorrId 进行通知（s0122、s0123），消费者存储该通知（s0132）。文稿未说明除 notifCorrId 之外还有任何检查能把已接受的通知绑定回该订阅，也未定义针对未知 notifCorrId 的应用错误（s0338）。	s0008, s0069, s0073, s0076, s0084, s0117, s0123, s0132
A04	每次订阅运行仅一次接受	含 MTLF 的 NWDAF 对一条通知的接受，与含 AnLF 的 NWDAF 的一次不同订阅运行相对应，从而同一条通知不会在一个订阅下被接受两次。	订阅以 subscriptionId 创建（s0086），通知携带 notifCorrId（s0123）；文稿未定义消息级重放保护、重复检测，也未定义针对重复通知的应用错误（s0338），并把防重放交给传输层（TS 29.500 第 6.7.2 条）。	s0076, s0086, s0123, s0132
A08	通知字段绑定到订阅	含 MTLF 的 NWDAF 得到保证：一条已接受通知的 notifCorrId、modelIds 和 notificationUri，以及它携带的反馈，都来自它自己的订阅请求，既未被替换也未被错接。	订阅固定了 modelIds、notificationUri 和 notifCorrId（s0074、s0075、s0076）；通知携带 notifCorrId 以及 modelAccuInfos 和 anaFeedbacks，各自指明 modelIds（s0123、s0125、s0126、s0322、s0328）。文稿仅在一条 NOTE 中建议 mLEvent、mLEventFilter 和 tgtUe 应与已登记的内容一致（s0319），未说明对通知的 modelIds 与订阅进行强制校验。	s0074, s0075, s0076, s0123, s0124, s0125, s0126, s0319
D01	通知来自所订阅的AnLF	含 MTLF 的 NWDAF 得到保证：它接受的每条通知都对应它所订阅的含 AnLF 的 NWDAF 的一次真实发送，且负载与关联字段相同。	Notify 操作由含 AnLF 的 NWDAF 用于通知订阅者（s0117、s0288），消费者存储该通知并应答 204（s0132）。文稿未为通知定义签名、MAC 或令牌；TS 29.500 第 6.7.3 条指出本版本中 notify 操作不使用 OAuth2 访问令牌，TS 29.500 第 6.7.2 条把消息保护交给 TLS。	s0117, s0122, s0126, s0132, s0288
Z03	令牌授权主体和操作	含 MTLF 的 NWDAF 与含 AnLF 的 NWDAF 得到保证：Register、Subscribe 或 Notify 请求的主体、受众和操作，正是 OAuth2 令牌为其签发的内容。	对 API 的访问可由 OAuth2 授权，以 NRF 作为授权服务器（s0344），消费者在消费服务前应获取令牌（s0345），且该 API 为整个服务定义单一 scope，在资源级或操作级不定义额外 scope（s0347）。TS 29.501 第 4.10 条指出，仅有服务级 scope 的 API 会向所有被允许访问该服务的消费者授予对其全部服务操作的访问权，TS 29.500 第 6.7.3 条补充说明 notify 操作不使用令牌。	main:19224, ts-29-501:1801, ts-29-500:5032, s0344, s0345, s0347
Z04	取消后不生效	含 MTLF 的 NWDAF 与含 AnLF 的 NWDAF 得到保证：一旦订阅被取消（Unsubscribe，s0109 至 s0112）或注册被删除（Deregister，s0061 至 s0064），基于该订阅或注册的任何通知或监控步骤都不再生效。	NWDAF 应在 DELETE 时移除订阅并应答 204（s0110、s0111、s0112），并应在 DELETE 时移除注册（s0062、s0063、s0064）。文稿未对移除时已在途或移除后重复的通知规定任何规则，并把注销与取消订阅的全局一致性置于一旁（任务边界）。	s0061, s0063, s0064, s0109, s0111, s0112
A08.7	关联和URI匹配订阅	含 MTLF 的 NWDAF 得到保证：一条已接受通知的 notifCorrId 和 notificationUri 来自它自己的订阅请求。	消费者在订阅中提供 notificationUri 和 notifCorrId（s0075、s0076），通知携带 notifCorrId 被投递到 {notificationUri}（s0122、s0123）。	s0073, s0074, s0075, s0076, s0123

ID	简称	对象	文稿怎么说	出处
A03.8	接受的反馈匹配真实发送	含 MTLF 的 NWDAF 得到保证：一条已接受的反馈通知对应含 AnLF 的 NWDAF 的一次真实发送，从而它存储的分析反馈和准确度信息是由所订阅的生产者产生的。	AnLF 携带 anaFeedbacks 和 modelAccuInfos 通知所订阅的事件（s0117、s0122、s0126），消费者存储该通知（s0132）；未定义对 notify 请求的逐消息认证（TS 29.500 第 6.7.3 条：notify 不使用令牌）。	s0117, s0118, s0122, s0126, s0132
A08.9	反馈未错误连接到模型	含 MTLF 的 NWDAF 得到保证：它为某一模型接受的分析反馈，不会被归到它未请求的另一个模型或另一个订阅上。	订阅固定了 modelIds（s0074）和 notifCorrId（s0076），而一条通知携带的 anaFeedbacks 按元素指明 modelIds（s0126、s0328）；文稿不要求将这些 modelIds 与订阅进行比较，仅在 NOTE 层面建议 mLEvent、mLEventFilter 和 tgtUe 应一致（s0319）。	s0074, s0076, s0123, s0126, s0328

分析者没有补充性质：起草的清单已覆盖文稿写明的目标。

性质目录，逐条去向

性质是从一份安全和隐私性质目录起草的。起草时有 36 条没选，分析者把它们逐条过了一遍：作为性质在这里检查，或者写明原因不检查。检查了 8 条，没检查 28 条。“文稿没有提到”不能作为不检查的原因。

条目	目录里的性质	作为哪条性质检查	不检查的原因
K01	密钥／凭据保密性	不在范围内	本 API 未定义长期密钥或秘密凭据；OAuth2 访问令牌是一种持有者凭据，其保护属于传输层，由 TS 33.501 和 TS 29.500 定义，而任务未纳入这两份规范。令牌的授权含义由 Z03 覆盖。
K02	载荷与端到端保密性	不在范围内	SBI 上精度信息和反馈载荷的保密性如果有，也是由 TS 33.501 的传输安全提供的，而任务未纳入该规范；对谁可以获得监控数据的唯一应用层控制是 Z03 所检查的授权。
K03	强保密／秘密值不可区分性	不适用	本 API 未定义任何要求不可区分性的秘密值。
K04	密钥用途、方向与会话分离	不适用	本 API 不派生也不使用任何密钥。
A01	对端曾参与性（Aliveness）	A03	
A02	弱一致性认证	A03	
A05	隐式密钥认证	不适用	该协议不建立任何密钥。
A06	显式密钥确认	不适用	不建立也不确认任何密钥。
A07	身份—凭据—密钥绑定	Z03	
R01	认证／挑战新鲜性	不适用	本 API 未定义任何挑战、计数器、时间戳或票据新鲜性机制。
R02	消息级抗重放	A04	
R03	业务操作至多一次	A04	
R04	消息接受顺序／窗口安全	不适用	本 API 未定义任何序列号、排序规则或接受窗口。
R05	抗降级／安全策略保持	不适用	本 API 的支持特性表为空（s0341、s0342），因此没有可降级的协商。
C01	前向保密性（FS）	不适用	不建立密钥，因此前向保密不适用。
C02	抗密钥泄露冒充（KCI）	不适用	本 API 不对消息做长期密钥认证。
C03	已知密钥安全／泄露隔离	不适用	本 API 中没有密钥层次结构或会话密钥。
C04	临时秘密／部分状态泄露韧性	不适用	没有临时秘密，也没有本 API 要防护其泄露的组件状态。

条目	目录里的性质	作为哪条性质检查	不检查的原因
C05	泄露后安全恢复（PCS）	不适用	没有棘轮或密钥更新机制。
C06	注册凭据泄露后的增强认证	不适用	没有口令认证的密钥交换，也没有认证客户端的注册记录。
C07	抗离线字典攻击	不适用	不使用口令或其他人类选择的秘密。
C08	指定原语失效下的目标保持	不在范围内	本 API 未定义自己的密码原语；所用的原语属于 TLS 和 OAuth2 框架，而任务未纳入这些内容。
Z01	授权执行正确性	Z03	
Z02	委托权限不扩张	不适用	本 API 没有委派或代理链；SCP 转发请求时不带委派授权。
Z05	跨租户／跨服务安全隔离	不适用	本 API 没有自己的租户或域隔离语义；PLMN 分离不属于所建模的条款。
S01	加密 nonce 唯一性	不适用	本 API 不使用任何对 nonce 敏感的密码原语。
S02	重传／密钥重安装安全	不适用	本 API 中没有密钥安装或重传状态机。
S03	抗回滚与状态单调性	Z04	
S04	跨阶段安全上下文一致性	不在范围内	任务声明注册关系单独建模，且不能假定任何订阅都需要先有注册，并把去注册与取消订阅的全局一致性放在一边；因此这类跨阶段上下文一致性不在范围内。
S05	终止与旧状态失效	Z04	
P01	身份匿名性	不适用	各方是已标识的 NF 实例；没有陈述或隐含任何匿名性目标。
P02	会话不可关联性	不适用	没有陈述任何不可关联性目标；NF 身份在 SBI 中按设计是可见的。
P03	用户／凭据存在性隐藏	不适用	本 API 没有可枚举用户或凭据的登录或发现查询。
V01	条件性进展／错误恢复	证明器问不了	无法向证明器询问进展和终止：它回答的是迹的性质，而不是一次运行总能完成。
V02	无非预期死锁	证明器问不了	无死锁是分支时间性质，无法向证明器询问。
V03	受限资源滥用／抗放大	证明器问不了	资源消耗和放大是定量性质，无法向证明器询问。

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 13 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 6 项，没设的 7 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 containing MTLF	参与方	`rogue_nf`	
S02	NWDAF containing AnLF	参与方	攻击者得不到更多	AnLF 是它所通知的精度信息和分析反馈的来源；控制它只能得到它本来就会产生的数据。通知绑定目标仍然成立，因为它们要求该参与方真实发送，而精度值的真实性已被任务排除。
S03	NRF	参与方	按定义全部失守	NRF 是授权服务器和发现的信任锚；一旦它被控制，所有授权性质按构造即失效，而且文稿没有要求在 NRF 不诚实时对本 API 提出任何性质。
S04	notifCorrId	值	`on_path`	
S05	notificationUri	值	`on_path`	

ID	参与方或值	类别	攻击者条件	不设的原因
S06	modelIds	值	攻击者得不到更多	ML 模型标识符是订阅中和每条通知中都明文携带的小整数；知道它们对读取请求的攻击者没有任何增益，它们本身也打不开任何东西。
S07	订阅者身份	值	攻击者得不到更多	订阅体不携带 consumerId：订阅者只能从令牌主体和传输对端得知，而这两者对 AnLF 来说按设计都是可见的。NF 身份在 SBI 中不是秘密。
S08	subscriptionId	值	`on_path`	
S09	anaFeedbacks	值	攻击者得不到更多	分析反馈就是通知载荷；没有目标要求其保密性，而读取它需要载荷，on_path 攻击者已经拥有载荷。任务把反馈的含义排除在范围之外。
S10	modelAccuInfos	值	攻击者得不到更多	精度信息就是通知载荷；按任务陈述其值是不透明数据，且没有对其提出任何保密性目标。
S11	OAuth2 访问令牌	值	`rogue_nf`	
S12	scope nnwdaf-mlmodelmonitor	值	攻击者得不到更多	该 scope 是公开的服务名；没有令牌时持有该名称不授予任何东西，而令牌是 S11 中的攻击者持有物。
S13	registrationId	值	`on_path`	

每条性质在各攻击者条件下的答案

攻击者由分析者设定，且在编写查询之前就已设定；每个答案都是在其中某一个攻击者下记录的。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”是为该性质设定的、证明器在其下让某条主张悬而未决或什么都没记录的攻击者；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、修改和发送消息，不持有任何一方的秘密		
`on_path`	读取并可改写 NWDAF containing MTLF 与 NWDAF containing AnLF 之间交换的 HTTP 请求体和响应体，因此它获知 notifCorrId、notificationUri、modelIds、subscriptionId 和 registrationId，并且可以向两个 NF 的资源 URI 和回调 URI 发送请求。它自己不持有任何 OAuth2 令牌。	本 API 自身的消息不带任何应用层保护：TLS 是 SBI 消息唯一声明的保护（TS 29.500 第 6.7.2 条），SCP 中介请求并可能终止 TLS（TS 29.500 第 6.10.9 条），而本版本中 notify 操作完全不使用 OAuth2 访问令牌（TS 29.500 第 6.7.3 条），同时文稿没有规定对通知的签名或 MAC。	ts-29-500:4980, ts-29-500:5032, s0344, s0122
`rogue_nf`	持有单一 scope nnwdaf-mlmodelmonitor 的有效 OAuth2 访问令牌（每个 NF 都从 NRF 获得），以及服务地址；它可以用自选参数（任意 modelIds、自己的 notificationUri 和 notifCorrId）调用 Register、Deregister、Subscribe 和 Unsubscribe，并接收它订阅的通知。它不读取其他 NF 受传输保护流量。	每个 NF 服务消费者在消费服务前都应获得令牌（s0345），且本 API 为整个服务定义单一 scope，没有资源级或操作级 scope（s0347），因此服务级 scope 授予对所有操作的通用访问（TS 29.501 第 4.10 条）。	ts-29-501:1801, main:19224, s0344, s0345, s0347

ID	`network`	`on_path`	`rogue_nf`
A03		不成立	
A04		不成立	
A08		不成立	
D01		不成立	
Z03	成立		不成立

ID	network	on_path	rogue_nf
Z04		不成立	成立
A08.7		成立	
A03.8		不成立	
A08.9		不成立	

原文和目标

原文：347 个原文单元，63012 字节，冻结于 source-delivery-01.json。 目标：6 个已批准（A03、A04、A08、D01、Z03、Z04）；所提问题：3 个。 目标清单事先按草稿批准，使用 --approve-targets。

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

- approve_package: 批准源文包 protocol-188d5da04c35c21d9e19487f1a09

本步骤自报：1 次请求（1 次失败），0.000 USD，输入 46859 个令牌，输出 3348 个，151 秒。

关于证明器和模型的说明

- A04：已在 ProVerif 中作答，分析者为该目标命名 changing_state
- Z04：已在 ProVerif 中作答，分析者为该目标命名 changing_state

一个简化了代数（xor、dh_algebra、other_algebra）的 ProVerif 模型给攻击者的能力低于真实运算：在此得到的证明是针对更弱攻击者的证明，在此找到的攻击就是攻击。一个略去对变化值（changing_state）检查的 ProVerif 模型让各方接受得更多：在此得到的证明成立，在此找到的攻击可能只是缺少检查。Tamarin+Agent 按原样对这些建模。

以步数给出的界是在 bounded.pv 上的答案：这是分析者的第二个 ProVerif 模型，其中保存变化值的一方连同其检查一起展开到该步数。该模型做得比协议少，因此该性质只对这些步成立，不再往前；在此找到的攻击是攻击。步数是分析者的陈述。

分析者的状态审查

这一部分是分析者自己对原文的解读，不是证明器的答案。ProVerif 所示一栏中的“yes”表示 ProVerif 在展开模型中找到了带有这些步的轨迹，因此这些步是可能的；这些步是否破坏某个目标，只由答案表中的一次攻击说明，它们还付出什么代价则是分析者的解读。“nothing found”说明查看了哪些模式，不证明任何东西。

值	由谁保存	何时变化	何时检查	参考文献
ML 模型监测订阅的集合，每个订阅带有其 subscriptionId、modelIds、notificationUri 和 notifCorrId	包含 AnLF 的 NWDAF	在 Subscribe 时（创建新订阅，分配 subscriptionId 并存储）和在 Unsubscribe 时（删除订阅）；该删除正是 Z04 所依赖的步骤。	当 AnLF 为某个订阅发送通知时（s0122），以及当 DELETE 指名某个 subscriptionId 时（s0110）。	s0073, s0085, s0086, s0087, s0109, s0110, s0111, s0114
ML 模型监测注册的集合，每个注册带有其 registrationId、modelId 和消费者标识符	包含 MTLF 的 NWDAF	在 Register 时创建（分配 registrationId，存储注册），在 Deregister 时删除。	当 DELETE 指名某个 registrationId 时（s0061），以及当该注册是监测该模型的理由时。	s0038, s0048, s0049, s0050, s0051, s0061, s0062, s0063, s0064, s0066

值	由谁保存	何时变化	何时检查	参考文献
消费者归属于其订阅的 notifCorrId	包含 MTLF 的 NWDAF	在建立订阅时选定；原文没有给出在 Unsubscribe 时使其失效的步骤。	在每次通知时：消费者将 MLModelMonitorNotify 中的 notifCorrId 与其订阅的 notifCorrId 比较（s0123, s0132）。	s0076, s0110, s0111, s0112, s0123, s0132, s0338

模式	结果	目标	ProVerif 所示	步骤
replay	候选	A04	是：ProVerif 在 bounded.pv 上否证了 `replayed`	MTLF 已为其订阅接受了一条通知并应答 204（s0132）。/ 同一 MLModelMonitorNotify 正文，带有相同的 notifCorrId、modelAccuInfos 和 anaFeedbacks（s0122, s0123），再次被发往 notificationUri。/ MTLF 在其建立的订阅中找到相同的 notifCorrId，第二次存储该通知并再次应答 204（s0132）；文稿没有定义对重复通知的重复检测，也没有定义应用错误（s0338）。
repeat_one_time_action	候选	A04	否：未用 ProVerif 尝试	MTLF 用其 modelIds、notificationUri 和 notifCorrId 发送 POST /subscriptions（s0073），响应在返回途中丢失，于是它再次发送同一请求。/ 在第二次 POST 时，AnLF 再次创建新订阅，分配新的 subscriptionId 并存储（s0085, s0086, s0087）；文稿不要求幂等键，也不要求重复检测。/ 两个订阅随后用相同的 notifCorrId 通知同一 notificationUri（s0122），于是每份监测报告都在一个标识符下被投递两次，MTLF 无法区分这两个订阅。
desynchronisation	候选	Z04	是：ProVerif 在 bounded.pv 上否证了 `late_accepted`	MTLF 取消订阅：对其订阅发送 DELETE，AnLF 删除该订阅并应答 204（s0110, s0111, s0112）。/ AnLF 此前已为该订阅发送过一条通知（s0122），该通知在其 204 之后到达 MTLF；此时双方处于不同状态，文稿中没有任何东西使它们重新同步。/ MTLF 仍将收到的通知与其已删除订阅的 notifCorrId 匹配，因此接受并存储了迟到的通知（s0132）；它没有保留任何可用于拒绝的记录，也没有规则告诉它要拒绝。
update_before_check	nothing found			查看了令牌检查和删除：订阅只在范围检查之后创建，且只在生产者接受的 DELETE 之后删除。在改变某个值的消息被验证之前，没有值被改变。
no_rollback_on_failure	nothing found			AnLF 在接受 DELETE 时替换其保存的值（在 bounded.pv 上验证为 late_sent），在拒绝时保持不变（s0114），这与原文所述一致。未能同步的消费者一侧记录在 desynchronisation 下。
race_between_sessions	候选		否：未用 ProVerif 尝试	MTLF 的订阅被创建，其 subscriptionId 出现在 201 响应的 Location 头中（s0088）；SBI 路径上的某一方读到了它。/ 该方用自己的 OAuth2 访问令牌对 /subscriptions/{subscriptionId} 发送 DELETE：该 API 只有一个服务范围（s0347），因此 AnLF 对调用者能施加的唯一检查就是该范围。/ 订阅正文不携带 consumerId（s0073 至 s0076），因此 AnLF 没有存储任何可用于比较调用者与订阅者的东西；它删除 MTLF 的订阅并应答 204（s0111, s0112），MTLF 对该模型的监测随之停止。
distinguishable_failure	候选		否：未用 ProVerif 尝试	调用者对 /subscriptions/{subscriptionId} 发送 DELETE，一次使用有效令牌和存在的标识符，一次使用有效令牌和不存在的标识符。/ 第一次应答 204 No Content（s0112），第二次应答 HTTP 错误（s0114），若令牌不被接受则为 401/403（TS 29.500 6.7.3）。/ 这一差异告诉调用者该 subscriptionId 在 AnLF 处是否存在，而这正是想要取消他人监测的 NF 所需要的；subscriptionId 是 AnLF 分配的 URI 片段（s0088），因此收益受其可预测性限制，且没有目标要求隐藏存在性。
wrap_or_window	nothing found			查看了每个值：没有一个是计数器、序列号或窗口。notifCorrId 是消费者选定的不透明字符串，subscriptionId 和 registrationId 只分配一次，该 API 没有定义接受窗口或排序规则（s0338, s0342）。

分析者为各目标命名的内容：

步骤	目标	特征	理由
31	Z04	changing_state	Z04 涉及活动订阅和注册的集合，AnLF 和 MTLF 保存并改变该集合：订阅在 Subscribe 时创建，在 Unsubscribe 时删除，而后续通知是否仍然生效取决于该值。model.pv 略去了删除（只增不减的存储），因此在 bounded.pv 中把删除展开为两个监测间隔。
31	A03, A04, A08, D01, A08.7, A03.8, A08.9	plain_primitives, many_sessions	通知绑定的答案不依赖任何代数原语：模型中只有不可伪造的 OAuth2 令牌项和不透明的准确度数据，SBI 传输被建模为 Dolev-Yao 信道，双方各有任意数量的会话。被简化的是传输和令牌框架，不是带有法则的运算。
31	Z03	plain_primitives	Z03 仅依赖 OAuth2 访问令牌，将其建模为不可伪造的项，生产者检查其范围；令牌框架本身（签发、过期、受众）未被建模，且不涉及任何带代数法则的运算。
42	A04	plain_primitives, changing_state, many_sessions	A04 的重放答案依赖于消费者为其订阅保存的状态：是否识别出对已存储通知的重复，取决于它保留了什么，而 model.pv 没有展开这一点，因此答案取自 bounded.pv，其中该状态展开为两步。

分析者的结束语

我按照 TS 29.503 第 4.7 条，对包含 AnLF 的 NWDAF 与包含 MTLF 的 NWDAF 之间的 Nnwdaf_MLModelMonitor 交互进行了建模：Register；携带 modelIds、notificationUri 和 notifCorrId 的 Subscribe；一条携带准确性信息与分析反馈的 Notify；以及用于退订和注销的 DELETE 操作。注册单独建模，跟踪一次注册不需要先有订阅，生产者只按文稿的要求检查 OAuth2 服务范围。九个目标对每一项主张在为其设定的每一种攻击者集合下都有答案，每个证明都带有对照查询，图和解读均已记录。

已确立：被接受的通知确实携带消费者自己选择的 notifCorrId 和回调地址（A08.7 以及 A03 的一项主张，已证明；对照查询显示，对于消费者不检查的字段，同一查询不成立）；没有服务范围的令牌，任何 Register 或 Subscribe 请求都不会被执行（Z03，已证明）；一旦 AnLF 处理完 DELETE，它就不再为该订阅发送任何内容（Z04，在订阅存储展开到两个区间的情况下有界，对照查询将其证伪）。

有两项发现，文稿均未陈述。第一，通知仅凭一个关联字符串就被接受：notify 操作不携带 OAuth2 令牌（TS 29.500 第 6.7.3 条），没有定义签名或 MAC，通知中的 modelIds 也从不与订阅比对，因此 SBI 路径上能读取订阅请求的一方可以为任意模型注入准确性信息和反馈（D01、A03、A03.8、A08、A08.9 被证伪）。第二，取消不是边界：消费者会继续接受其已取消的订阅的通知，并再次存储重复的通知（Z04，A04 在普通模型和展开模型中均被证伪）。Z03 的单范围授权缺口按文稿所述复现，并非新发现（TS 29.501 第 4.10 条）。

局限：在模型中，SBI 传输是攻击者的通道，因此每一个需要 notifCorrId 的攻击都被提交为取决于 TLS 在何处终止，以及路径上的元素是否在威胁模型中；针对每个模型的授权问题无法向 NRF 的模型提出；展开模型覆盖一个订阅的两个监控区间，不覆盖注销一侧；没有任何答案依赖异或或 Diffie-Hellman。有三种状态审查模式是候选，各带一个查询（重放、去同步）或一份书面轨迹（重试的 Subscribe、由另一 NF 取消、可区分的失败）。