MFAF 配置通知与获取的安全分析

本报告的其他语言版本：English。

19.7.0 · ProVerif+ Agent 报告

文稿	5G System; Messaging Framework Adaptor Services; Stage 3（协议定义）
检查的性质	13 条：3 条成立，10 条不成立（11 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

TS 29.522 规定，Messaging Framework Adaptor Function 负责保存从别处收集的数据和分析结果，并将其交给消费者。DCCF 用分析消费者的地址配置 MFAF，MFAF 通过 fetchInstruction 通知该消费者，消费者再用收到的 fetch 关联标识符获取数据。MFAF 保存配置和数据；DCCF 持有配置服务的访问令牌；分析消费者持有获取服务的令牌及其收到的通知；NRF 签发这些令牌。

交换结束后，消费者拥有与其收到的标识符对应的数据，并能核对这些数据是否属于它所知的配置。MFAF 知道通知和获取都由两个服务的令牌授权，也知道这些标识符是它自己签发的，但它没有任何规则把获取绑定到标识符所针对的那个消费者。

各方：

- **DCCF**：配置 MFAF 的 NF 服务消费者。持有：nmfaf-3dadatamanagement 的访问令牌。
- **MFAF**：两个服务的生产者。持有：已存储的配置、已收集的数据和分析结果。
- **分析消费者**：配置的通知端点。持有：nmfaf-3cadatamanagement 的访问令牌。

图 1 展示了文稿所规定的交换流程。

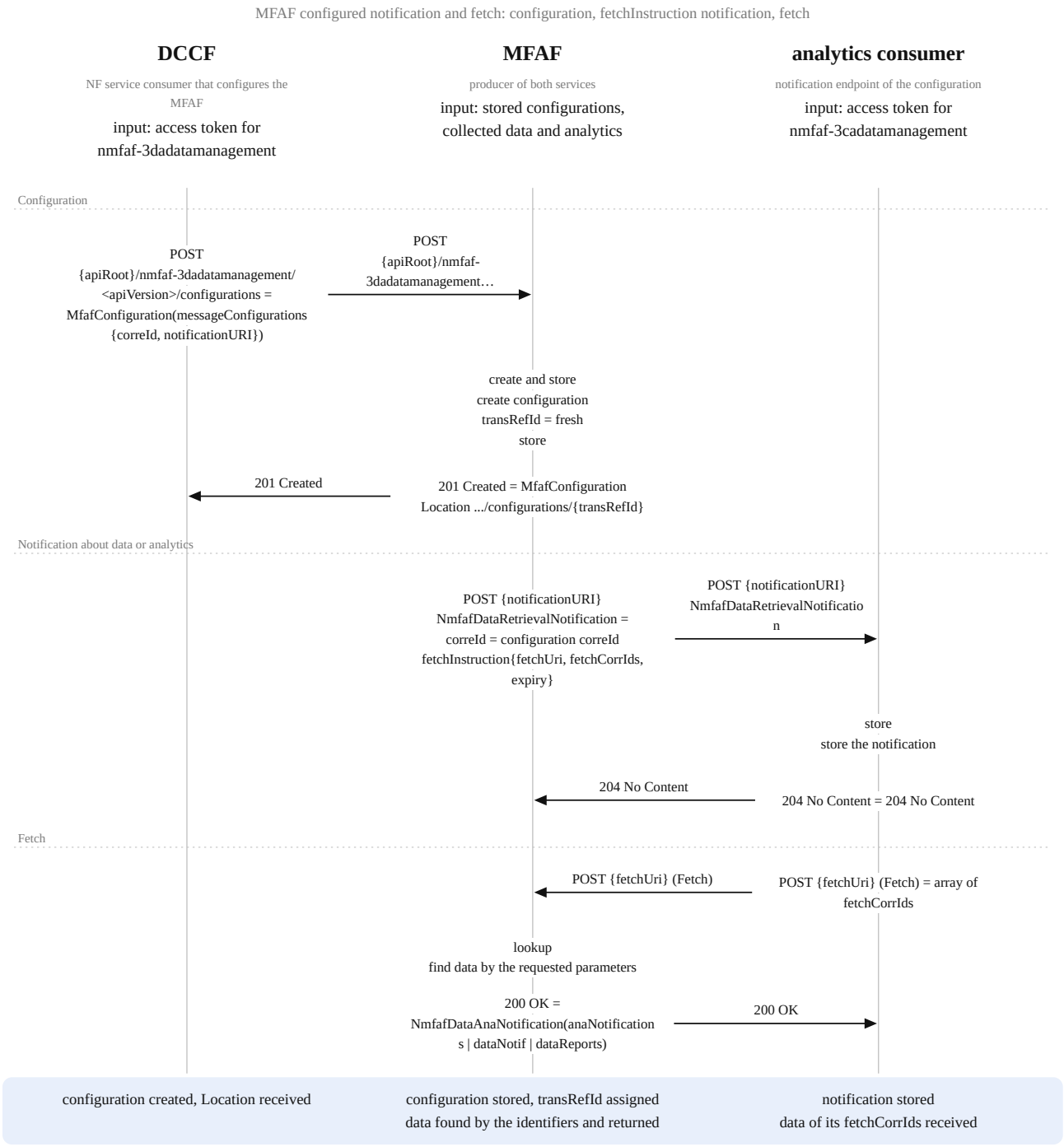


图 1. 文稿所规定的协议。

分析的范围见附录 D：分析了什么、省略了什么，以及依据的是文稿的哪些章节。

2 发现

本次分析共有 4 项发现：以下内容文稿中并未写明。其中 4 项无法确定文稿是否写明。

该流程能保护数据不被网络攻击者获取，并将通知绑定到消费者自身的配置，但没有把获取绑定到该消费者：fetchCorrIds 是持有者凭据，MFAF 只要求出示者持有服务范围任意令牌即为其服务。按文稿原文，由此产生三个弱点：只要持有某通知的标识符，就足以获取其数据（A03、A08、Z03、K02）；到期期限和 Deconfigure 不会撤销任何东西，旧标识符继续有效（Z04）；配置的更新永远不会到达消费者手中的

fetchInstruction，因此更新前的数据会被返回，并被当作当前数据接受（S03、S03.13）。在十三个目标中，Z01、Z01.12 和 R03 已由证明确立；其余目标部分确立，不成立的部分各有攻击。这些结果基于一个模型，该模型将 TLS 理想化为每对网络功能之间的一条经过认证的加密信道，并将配置的生命周期展开为三次获取；关于任何持有 transRefId 的 NF 都能改写配置这一发现未经查询确认，因此未作为发现报告。

攻击图沿用图 1 的列。红色表示攻击者的操作。黄色标记的是某一方发送的消息——尽管规定的交换在该处并不包含它——以及该攻击所破坏的结果。

2.1 任何持有某通知的 fetchCorrIds 的 NF 都能获取其数据

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 (main:2640, rfc6749:2879)
价值	中
涉及的性质	A03, A08, A08.11, Z03, K02, S04（见第 3 节）
依据	ProVerif 攻击查询 `a03_owner_ctl`、`k02_rogue_ctl`；图 2、3、4、5、6 和 7

发生了什么。 只要出示 fetchCorrIds 和服务范围的令牌，Fetch 就会被授予；MFAF 无需将请求者与标识符所属配置的消费者进行比对。因此，持有另一消费者标识符的第二个 NF 就能获得该消费者的数据；如果这些标识符日后泄露，它们仍能以同样的方式被兑换。

攻击者需要什么。 某条通知的 correId、fetchUri 和 fetchCorrIds，来自处理或重定向该通知的 SCP（所有各方都必须信任 SCP，ts-33-501:2921），或来自这些值的先前持有者；以及服务范围 nmfaf-3cadatamanagement 的任意访问令牌，任何已授权的 NF 都能获得。

攻击者能得到什么。 它能收到该通知的分析数据或数据报告，因为 MFAF 会返回这些标识符对应的已存储数据，而不比对请求者与配置。

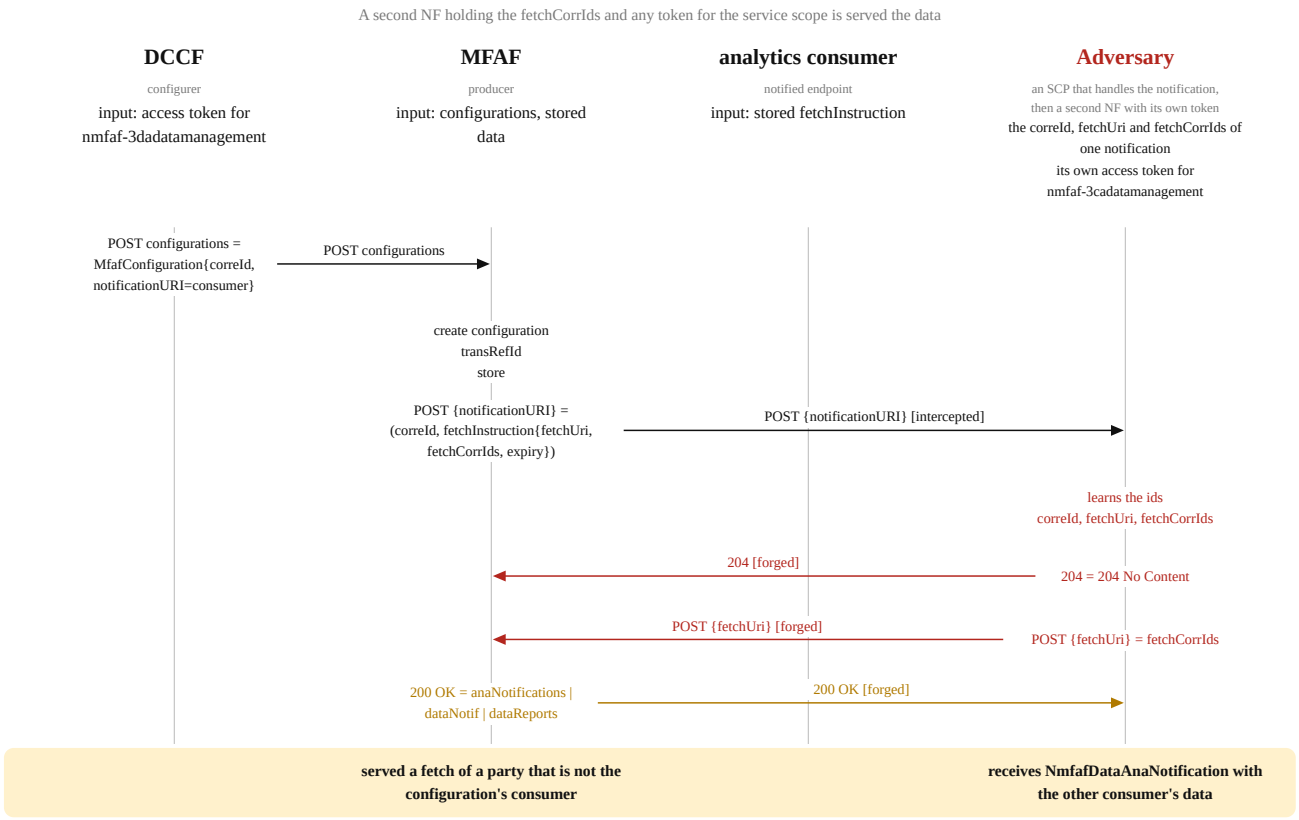


图 2. 持有 fetchCorrIds 和服务范围任意令牌的第二个 NF 会得到数据。性质 A03；攻击查询 a03_owner_ctl。

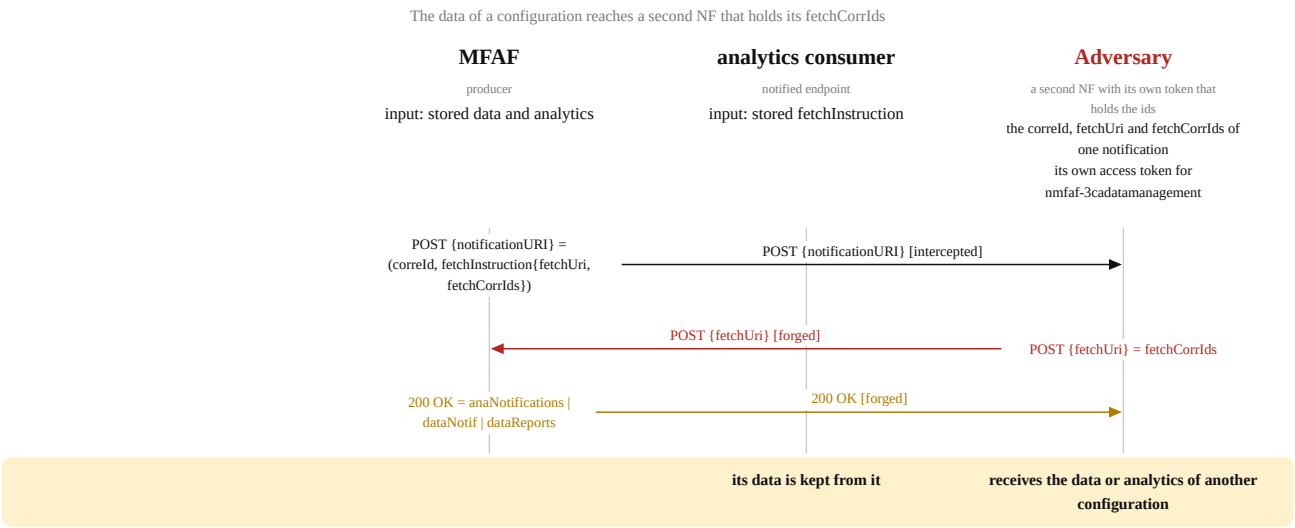


图 3. 某配置的数据会到达持有其 fetchCorrIds 的第二个 NF。性质 K02；攻击查询 k02_rogue_ctl。

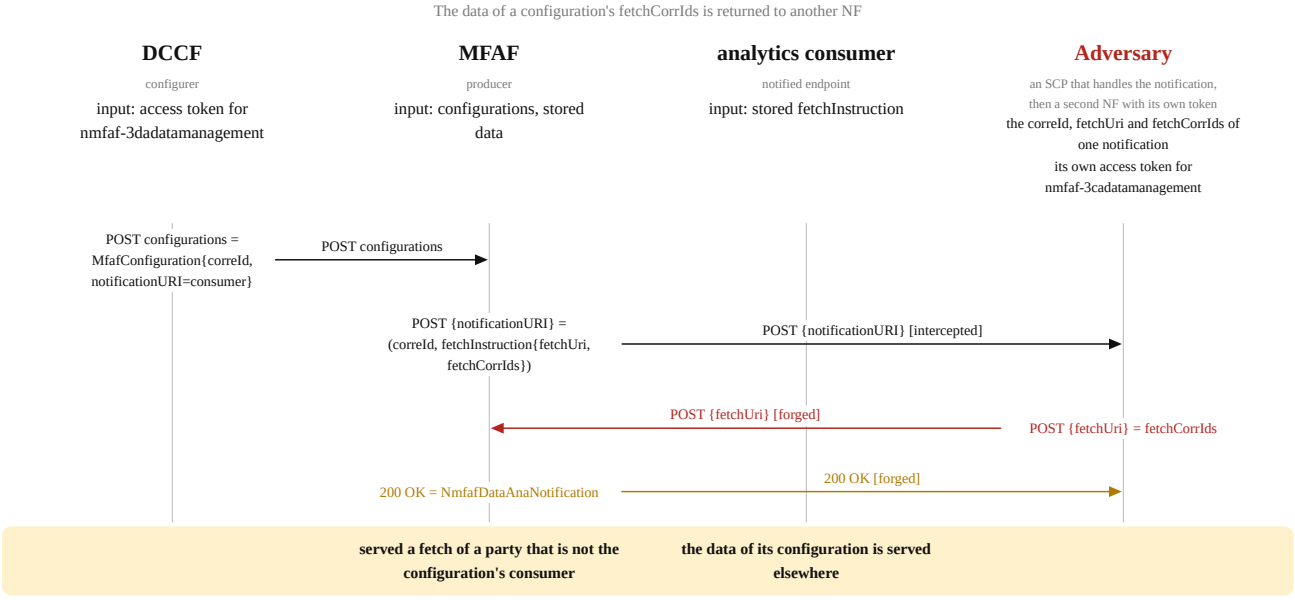


图 4. 某配置的 fetchCorrIds 对应的数据被返回给另一个 NF。性质 A08；攻击查询 a03_owner_ctl。

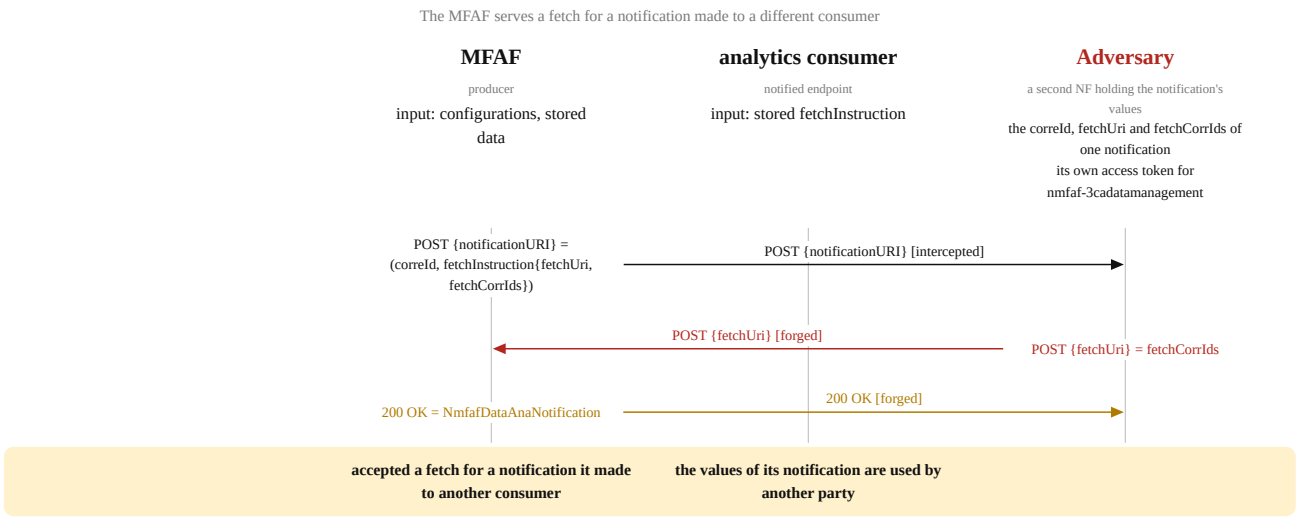


图 5. MFAF 为发往另一消费者的通知提供获取服务。性质 A08.11；攻击查询 a03_owner_ctl。

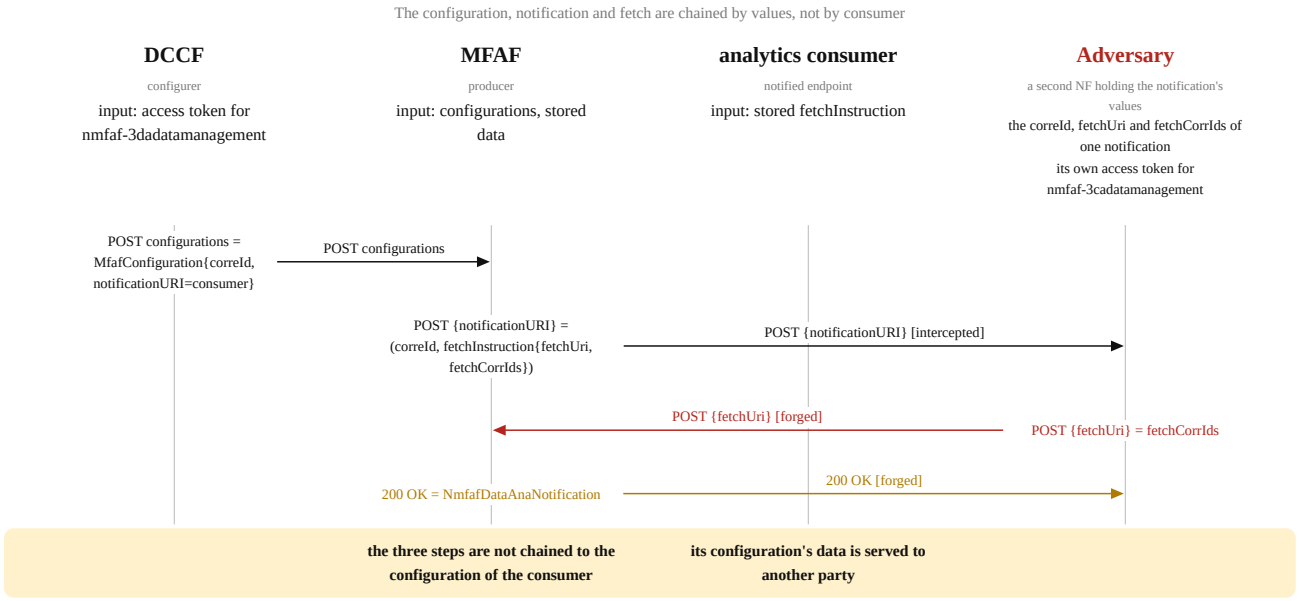


图 6. 配置、通知和获取依靠值而非消费者串联起来。性质 S04；攻击查询 a03_owner_ctl。

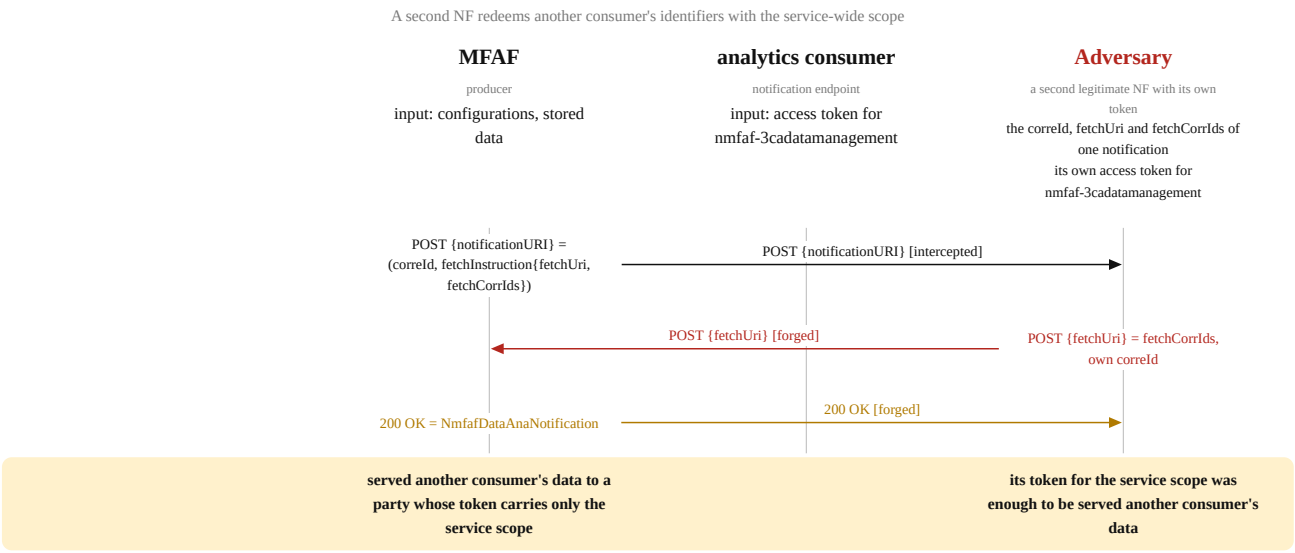


图 7. 第二个 NF 用服务范围的令牌兑换另一消费者的标识符。性质 Z03；攻击查询 a03_owner_ctl。

拟议修改。 MFAF 应在 fetchInstruction 的 fetchCorrIds 旁一并存储产生它们的配置的 NF 服务消费者，并且仅当访问令牌的主体是该消费者时才提供 Fetch 服务。另一种做法是让标识符一次性使用且有效期很短。

用 ProVerif 复核。 未检查：在会话预算内未对本建议运行任何检查，因此未经检查：模型首先需要在每个 fetchCorrIds 旁存储其所属配置的消费者，并在每次 Fetch 时将其与令牌主体比对。

2.2 fetchInstruction 的到期和 Deconfigure 不会撤销任何东西

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 (main:2461)
价值	中
涉及的性质	Z04（见第 3 节）
依据	ProVerif 攻击查询 `z04_expiry`、`z04_delete`；图 8

发生了什么。 fetchInstruction 带有一个到期时间，被描述为获取数据的截止期限，Deconfigure 会移除单个配置，但 MFAF 处理 Fetch 的步骤中既未列出与到期时间的比对，也未列出对配置是否仍然存在的检查。因此，在截止期限之后以及配置被删除之后，同样的 fetchCorrIds 仍会持续返回数据。

攻击者需要什么。 某条通知的标识符，由消费者自身、过时的实例或中介持有；除服务范围令牌外无需其他令牌。

攻击者能得到什么。 在 MFAF 自己设定的截止期限之后以及配置被删除之后，数据仍然可获取，因此一组过时或已泄露的标识符只要数据仍被存储，就能一直保持访问权。

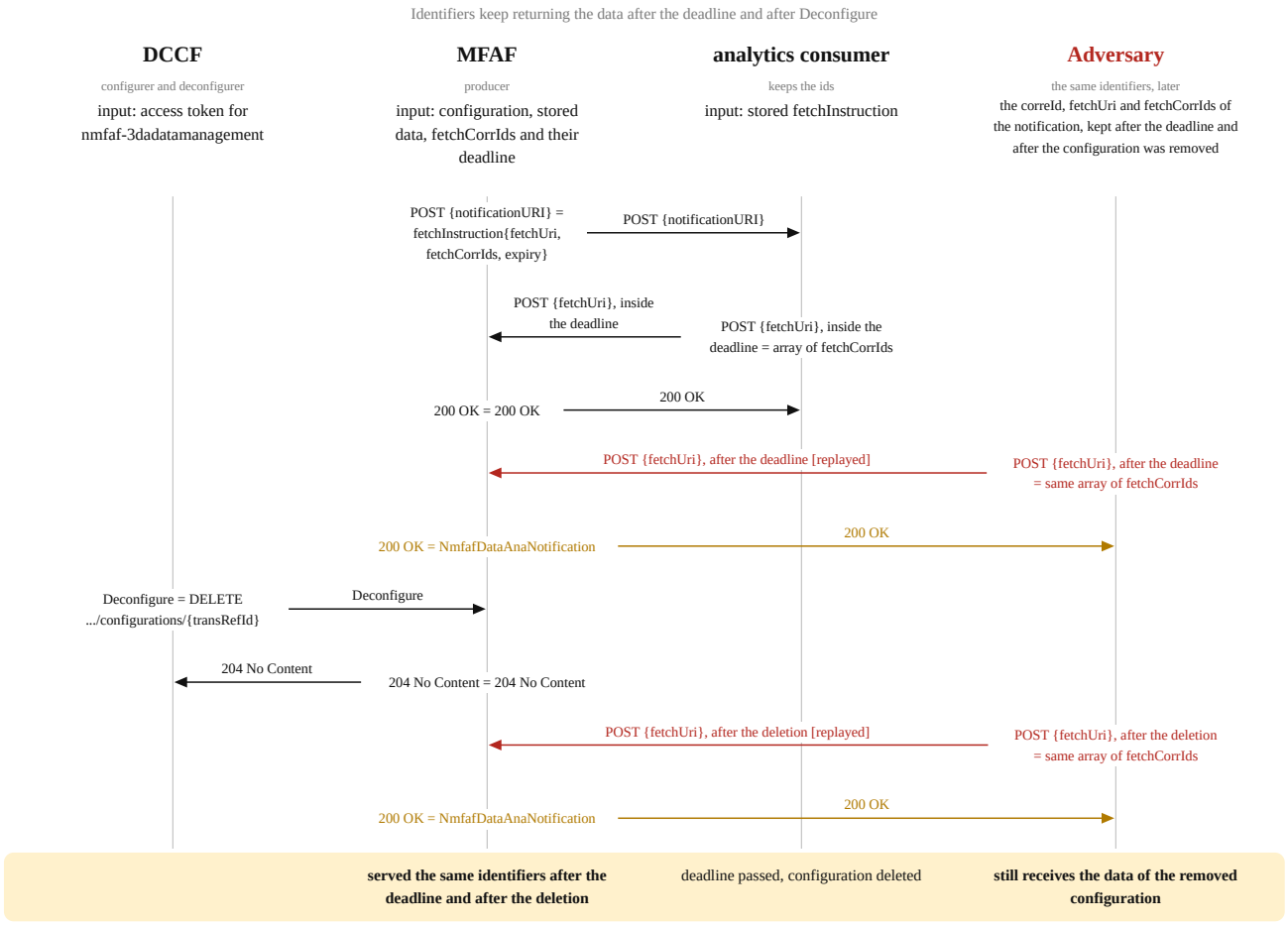


图 8. 在截止期限之后以及 Deconfigure 之后，标识符仍会持续返回数据。性质 Z04；攻击查询 z04_delete、z04_expiry。

拟议修改。 MFAF 应在每次 Fetch 时将当前时间与 fetchCorrIds 所属 fetchInstruction 的到期时间比对，并在截止期限过后返回错误。Deconfigure 应移除其删除的配置的 fetchCorrIds 和数据。

用 ProVerif 复核。 未检查：在会话预算内未对本建议运行任何检查，因此未经检查；若进行检查，版本会在 bounded.pv 中加入与到期时间的比对以及未失效标识符的移除，并重新运行两个攻击查询。

2.3 配置更新后消费者的 fetchInstruction 仍然有效

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 (main:2362)
价值	中
涉及的性质	S03, S03.13（见第 3 节）
依据	ProVerif 攻击查询 `s03_update`、`s03_accept`；图 9 和 10

发生了什么。 更新会替换已存储的配置，但没有任何东西到达消费者或已签发的标识符。消费者仍持有旧配置的 correId 和 fetchCorrIds，用它们获取，收到携带旧 correId 的更新前数据，并把它当作自己所知配置的数据接受，因为原文没有定义可供比对的配置版本。

攻击者需要什么。 除消费者已持有的内容外，攻击者不需要任何东西：更新由合法的配置方执行，消费者自己的标识就足够了。

攻击者能得到什么。 消费者无法判断自己的指令属于已被取代的配置：它把更新前的数据当作当前生效配置的数据，而 MFAF 会继续提供这些数据。

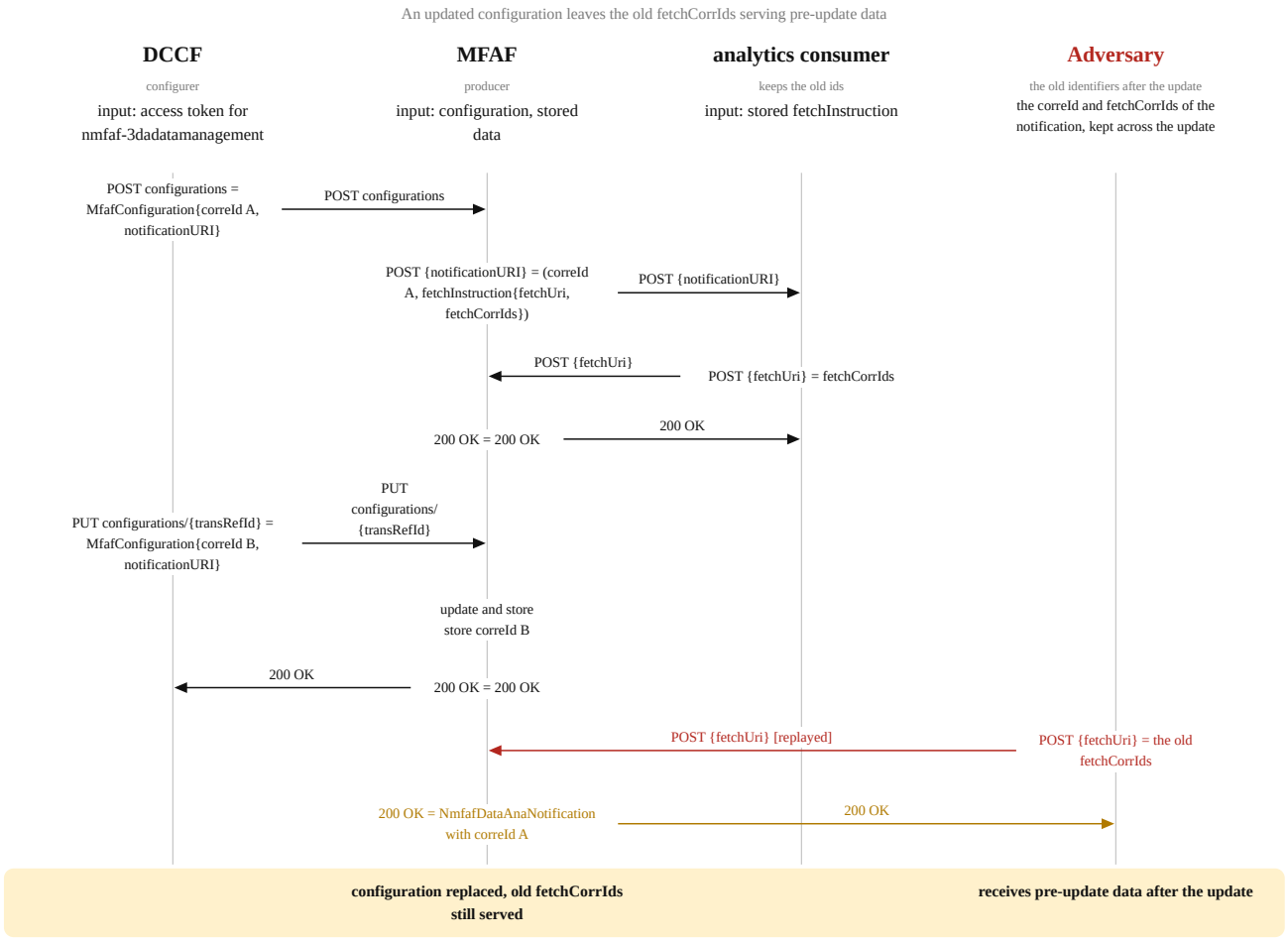


图 9. 更新后的配置使旧的 fetchCorrIds 继续提供更新前的数据。性质 S03；攻击查询 s03_update。

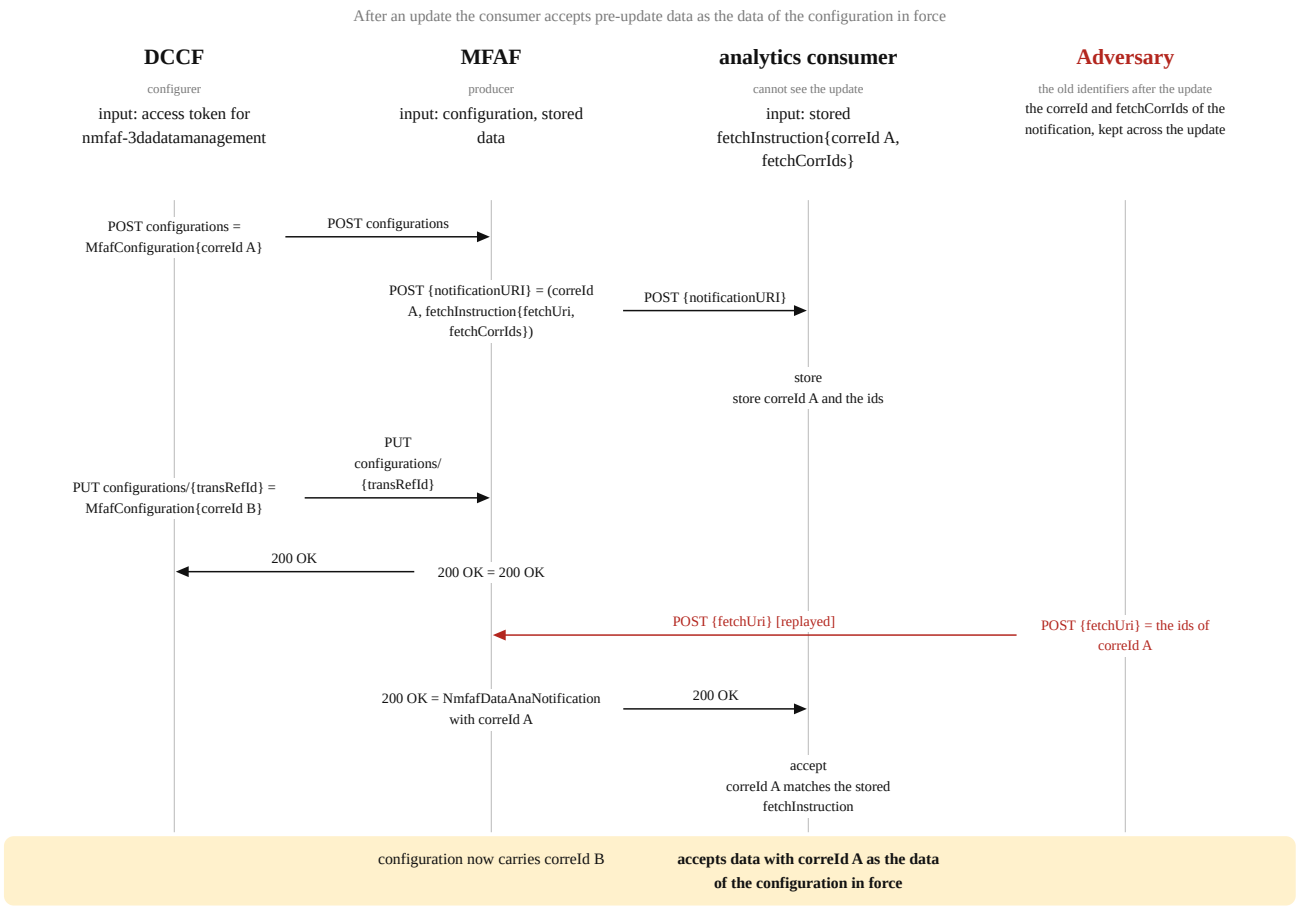


图 10. 更新之后，消费者把更新前的数据当作当前生效配置的数据接受。性质 S03.13；攻击查询 s03_accept。

拟议修改。 在 fetchInstruction 和 fetch 响应中携带配置的版本号，每次更新都递增该版本号，并让消费者只接受与其当前持有的配置版本一致的数据。若某次更新的目的在于取消先前的指令，则应写明该更新会使该配置的 fetchCorrIds 失效。

用 ProVerif 复核。 未检查：在本次会话的预算内没有对该建议运行检查；若进行检查，需在 bounded.pv 中为通知和响应加上配置版本号，并重新运行更新相关查询。

2.4 重复的通知被接受并再次存储

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 (ts-33-501:2865)
价值	低
涉及的性质	R02 （见第 3 节）
依据	ProVerif attack query `r02`; Figure 11

发生了什么。 一条通知仅由其 correId 标识，不携带消息标识，而消费者会存储它接受的每一条通知。因此同一通知的第二次投递会被接受并再次存储，与一条新通知无法区分。

攻击者需要什么。 对同一条通知的一次重复投递，例如处理该消息的 SCP 进行重试；攻击者不需要通知的内容。

攻击者能得到什么。 消费者会两次存储并处理同一份数据或 fetchInstruction；若该重复通知携带 fetchInstruction，同一份数据会被再次取回。

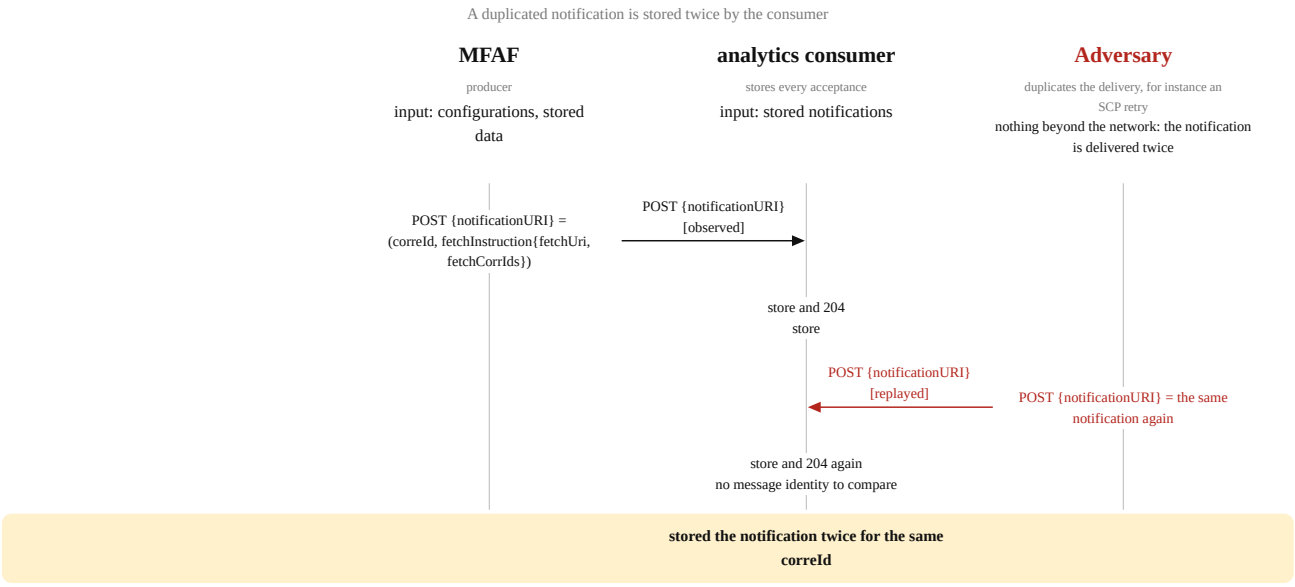


图 11. 重复的通知被消费者存储两次。性质 R02；攻击查询 r02。

拟议修改。 为 MFAF 通知加上消息标识，由消费者将其与该 correId 下已存储的通知进行比较，并写明重复通知以 204 确认且不再存储。

用 ProVerif 复核。 未检查：在本次会话的预算内没有对该建议运行检查；若进行检查，需在 model.pv 中为通知加上消息标识并在消费者中与已存储的通知进行比较，并在 bounded.pv 中跨多轮运行。

以下发现的这些攻击没有给出图： A03: k02_rogue_ctl, A08.11: k02_rogue_ctl, A08: k02_rogue_ctl, K02: a03_owner_ctl, S03.13: s03_update, S03: s03_accept, S04: k02_rogue_ctl, Z03: k02_rogue_ctl。

3 检查过的性质

每一行是一条交给 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	**通知与取回的相关性。** 每当 analytics consumer 接受一条 MFAF 通知，都存在 MFAF 的一次真实运行，其 correId 与该 consumer 配置中的 correId 一致；每当 MFAF 接受一次 Fetch，都存在该 consumer 的一次真实运行，其 fetchCorrIds 与 MFAF 此前在该通知中发出的 fetchCorrIds 一致。	无	不成立（1 个攻击）	中	不确定
A08	**取回使用其自身通知的 id。** consumer 接受的 fetchInstruction 中的 fetchUri 与 fetchCorrIds 来自它自己配置所对应的那次通知；MFAF 返回的数据对应请求中携带的 fetchCorrIds，而不是其他配置或其他 consumer 的 fetchCorrIds。	无	不成立（1 个攻击）	中	不确定
Z01	**每个操作都携带授权。** 每次配置、通知或 Fetch 操作被提交，都必须存在授权主体（NRF 作为授权服务器）签发的、对该操作有效的许可；仅有身份认证不足以产生受保护操作。	无	成立	中	是
Z03	**不传递令牌或相关 id。** 一个 consumer 的令牌与 correId/fetchCorrIds 只能用于它自己的配置与取数操作；攻击者不能把某 consumer 的许可或关联标识移到另一个 consumer 或另一个操作上使用。	无	不成立（1 个攻击）	中	不确定
Z04	**过期或撤销的 id 取不到任何内容。** 在 Deconfigure 生效之后，或 fetchInstruction 的 expiry 到期之后，使用旧 correId/fetchCorrIds 的新 Fetch 不再返回数据。	无	不成立（2 个攻击）	中	不确定

ID	性质	假定的泄露	结果	价值	文稿是否已写
S03	**重配置后无陈旧数据。** 配置更新生效之后，基于旧配置的 fetchInstruction 或旧 fetchCorrIds 不能使 MFAF 回到旧配置状态或返回旧配置下的数据。	无	不成立（1 个攻击）	中	不确定
S04	**配置、通知与取回链式关联。** 取数阶段使用的 fetchCorrIds 来自通知阶段为同一 consumer 生成的那次通知；通知阶段的 correId 来自配置阶段为同一 consumer 建立的配置；三个阶段之间不存在跨 consumer 或跨配置的错接。	无	不成立（1 个攻击）	中	不确定
R02	**通知至多被接受一次。** 同一逻辑 MFAF 通知或同一逻辑 Fetch 请求，在选定范围内至多被接收方接受一次。	无	不成立（1 个攻击）	低	不确定
R03	**取回至多被执行一次。** 同一逻辑取数操作（同一组 fetchCorrIds 对应的一次取数）在选定范围内至多提交一次安全相关副作用；请求可以重传，也可以返回缓存响应。	无	成立	低	否
K02	**数据仅对其消费者可见。** MFAF 通知与 Fetch 响应中携带的数据或分析结果，只对配置中指定的 consumer 或通知端点可见，其他合法 NF 与外部攻击者不能获得。	无	不成立（1 个攻击）	中	不确定
A08.11	**id 绑定到被通知的消费者。** consumer 接受的通知中的 correId 与 fetchInstruction 中的 fetchCorrIds，对应配置阶段为该 consumer 指定的 notificationURI 与 correId；MFAF 接受的 Fetch 中的 fetchCorrIds 对应它为同一 consumer 生成的那次通知。	无	不成立（1 个攻击）	中	不确定
Z01.12	**取回要求授权。** 每次 Fetch 操作被提交，都必须存在授权主体签发的、对该操作有效的许可；仅有身份认证不足以取数。	无	成立	中	是
S03.13	**旧指令不能存活超过更新。** 配置更新生效之后，基于旧配置的 fetchInstruction 或旧 fetchCorrIds 不能使 MFAF 返回旧配置下的数据，也不能使 consumer 把旧数据当作新配置的数据接受。	无	不成立（1 个攻击）	中	不确定

所提出的问题及分析者的回答：

问题	回答	依据
Q1: notificationURI/correId/fetchCorrIds 与数据接收者的绑定	在单个 consumer 内部，绑定成立：它接受的通知携带指定它的那个配置的 correId，它取数所用的 fetchCorrIds 来自该通知。跨 consumer 则不成立：MFAF 会向任何出示这些标识的 NF 提供数据，因此绑定的是这些值，而不是数据接收者。	A03, A08, A08.11, K02
Q2: 其他合法 NF 的 fetch 权限是否来自明确规范规则	否：唯一的规则是服务级范围 nmfaf-3cadatamanagement，每个获得授权的 NF 都能取得该范围，而 MFAF 没有义务将请求方与 fetchCorrIds 的 consumer 进行比较。只要持有该范围的有效令牌并掌握这些标识，Fetch 就能成功。	Z01.12, Z03, A08
Q3: 旧 fetchInstruction 与更新后配置的数据不可混淆	否：更新会替换已存储的配置，但旧 fetchCorrIds 仍然可用，而消费者无法看出自己的指令属于已被取代的配置，因此旧数据会被返回并被当作当前数据接受。	S03, S03.13

4 本次分析的局限

这里的结果是对协议模型的结果：它说明所规定的消息和检查提供了什么、未能提供什么，其中密码学被视为理想化的，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：

- TLS 被建模为每一对网络功能之间的一条经过认证的加密信道；SBI 自身的重放保护、证书处理以及 SCP 和 SEPP 路径属于 TS 29.500 和 TS 33.501 的基线，不在模型范围内。

- OAuth2 令牌被建模为授权服务器的一个签名，用其预置公钥进行校验；令牌生存期、受众以及令牌请求不在模型范围内。
- 在 bounded.pv 中，配置生存期被展开为 create、notify、fetch、update、fetch、deconfigure、fetch；超过三轮取数之后的情况不在覆盖范围内，而 model.pv 完全省略了更新和停用。
- MFAF 内部在配置、通知与已存储数据之间的关联并未由源文规定；模型为每个配置给出一条通知，并按标识查找数据。
- 第三方（SCP、重定向、陈旧实例）获取这些标识被建模为一个泄露它们的角色；这种泄露在部署中是否可能，源文并未给出结论。
- Individual MFAF Configuration 的更新与删除仅由 transRefId 标识，源文未列出任何所有权检查；这一来自状态审查的候选问题未经查询确认，因此不作为发现报告。
- 未分析任何可用性、时序、成本或活性性质，也未考虑算法的密码学强度。

宿主记录的内容：

- Z04、S03、S03.13 的答案依赖于模型所做的一处简化（附录 D）。
- 没带对照查询的证明（附录 C）：A03: a03_notify, A08: a08_cfg, Z01: z01_cfg, Z03: a03_owner, S04: a08_cfg, A08.11: a03_notify, A08.11: a08_cfg。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、什么能解决它们，见附录 A。

附录 A：逐条性质的解读

以下是分析者自己对证明器回答的解读，而不是证明器的回答：这些回答对协议意味着什么、这一解读依据什么、它能否照现在的样子使用，以及它有多大价值。

A03: 通知与取回的相关性

每当 analytics consumer 接受一条 MFAF 通知，都存在 MFAF 的一次真实运行，其 correId 与该 consumer 配置中的 correId 一致；每当 MFAF 接受一次 Fetch，都存在该 consumer 的一次真实运行，其 fetchCorrIds 与 MFAF 此前在该通知中发出的 fetchCorrIds 一致。

通知与取回由它们的值绑定：consumer 接受的通知，总是来自 MFAF 为某份以该 consumer 为 notificationURI 的配置真实发出的 fetchInstruction；而 MFAF 接受的 Fetch，也总是指向这样一条通知。未被绑定的是参与方的身份：谁持有 fetchCorrIds，谁就能被服务；模型中产生了一条轨迹，其中第二个 NF 因另一条 consumer 的通知而被服务。

- 结果：不成立（1 次攻击）；是否回答了所问：部分；可用但有保留
- 价值中：三个操作之间的对应关系在网络攻击者面前成立，但持有 fetchCorrIds 的参与方没有被识别，而这正是目标相关的原因。
- 文稿中是否有明确表述：不明确。所依据的原文：ts-33-501:2873, ts-33-501:2921, s0120, s0121, s0122, s0124, s0140, s0395, s0398
- 依据：TLS 被建模为每对 NF 之间的一条认证加密信道，因此网络攻击者既不能读取也不能伪造发给某个 NF 的消息；通知仅由 MFAF 选择的值来识别；未建模消息身份；第二个 NF 只能通过建模的泄露角色持有这些 id
- 未决：通知的 correId、fetchUri 和 fetchCorrIds 能否到达被通知 consumer 以外的参与方，原文未作定论
- 下一步：TS 29.522 或 TS 33.501 中关于谁可以出示 fetchCorrIds 的表述，加上 SCP 路径的 Tamarin+Agent 模型，可以定论

A08: 取回使用其自身通知的 id

consumer 接受的 fetchInstruction 中的 fetchUri 与 fetchCorrIds 来自它自己配置所对应的那次通知；MFAF 返回的数据对应请求中携带的 fetchCorrIds，而不是其他配置或其他 consumer 的 fetchCorrIds。

consumer 可以依赖其 fetchUri 与 fetchCorrIds 来自自身配置的通知，也可以依赖响应携带它所发送 id 的数据。但 MFAF 会把一组 id 的数据返回给任何持有这些 id 的 NF，因此该指令并未绑定到它为之生成的 consumer。

- 结果：不成立（1 次攻击）；是否回答了所问：部分；可用但有保留
- 价值中：consumer 一侧成立；只要持有这些 id 的参与方不是该 consumer，MFAF 一侧就不成立。
- 文稿中是否有明确表述：不明确。所依据的原文：rfc6749:2879, ts-33-501:2921, s0366, s0374, s0376, s0398, s0429
- 依据：TLS 按每对 NF 建模；MFAF 按原文所述仅依据标识符查找数据；向第二个 NF 泄露这些 id 被建模为一个角色，而非由原文推导
- 未决：原文是否意图让 fetchCorrIds 一次性使用或绑定到某个 consumer，未作说明
- 下一步：原文中的所有权规则，或一个让 MFAF 存储每个 fetchCorrIds 所属 consumer 的模型，可以定论

A08.11: id 绑定到被通知的消费者

consumer 接受的通知中的 correId 与 fetchInstruction 中的 fetchCorrIds，对应配置阶段为该 consumer 指定的 notificationURI 与 correId；MFAF 接受的 Fetch 中的 fetchCorrIds 对应它为同一 consumer 生成的那次通知。

consumer 接受的通知携带的是以它为 notificationURI 的那份配置的 correId，因此在通知路径上配置与 consumer 不会错配。然而 MFAF 会接受任何持有这些标识符的 NF 对其通知发起的 Fetch。

- 结果：不成立（1 次攻击）；是否回答了所问：部分；可用但有保留
- 价值中：主张的后半部分，即 MFAF 只服务它所通知的 consumer，被证伪。
- 文稿中是否有明确表述：不明确。所依据的原文：ts-33-501:2921, s0140, s0395, s0268, s0398
- 依据：TLS 按每对 NF 建模；consumer 除保存这些值之外不存储通知；端点或中间节点泄露这些 id
- 未决：第三方如何获得 correId 与 fetchCorrIds 不在所选原文单元范围内
- 下一步：对照此流程检查 TS 29.500 第 6.10.9 条的 SCP 与重定向路径，可以定论

Z01: 每个操作都携带授权

每次配置、通知或 Fetch 操作被提交，都必须存在授权主体（NRF 作为授权服务器）签发的、对该操作有效的许可；仅有身份认证不足以产生受保护操作。

MFAF 存储的每次 Configure 和它服务的每次 Fetch，之前都必须有授权服务器为该 NF 签发的、针对该服务范围的访问令牌；持有其他服务令牌的 NF 永远不会被服务。仅靠 TLS 上的身份认证是不够的。

- 结果：成立；是否回答了所问：是；可用
- 价值中：在一个把令牌理想化为签名的模型上，确认了原文描述的机制；值得注意的限制是范围是服务级的，这属于 Z03 的议题。
- 文稿中是否有明确表述：是。文稿中的位置：main:1982, main:2634, s0304, s0305, s0426, s0427
- 依据：令牌被建模为授权服务器的签名，其公钥预置在 MFAF 中；令牌的主体与请求携带的身份进行比较；令牌有效期与受众未建模
- 下一步：无；令牌自身的新鲜度与受众属于 OAuth2 基线，任务未涉及

Z01.12: 取回要求授权

每次 Fetch 操作被提交，都必须存在授权主体签发的、对该操作有效的许可；仅有身份认证不足以取数。

Fetch 只对令牌携带 nmfaf-3cadatamanagement 的参与方执行；经 TLS 认证但持有其他服务令牌的 NF 会被拒绝，而持有服务级范围的合法 NF 会被服务。该范围就是 Fetch 所拥有的全部授权。

- 结果：成立；是否回答了所问：是；可用
- 价值中：它表明 Fetch 的唯一条件是持有该服务的令牌，这正是单一范围所提供的机制。
- 文稿中是否有明确表述：是。文稿中的位置：main:2634, main:2640, s0426, s0429
- 依据：令牌是用授权服务器密钥验证的签名；未建模令牌过期、受众或复用规则；成对 TLS 信道给出发送方身份
- 下一步：授权问题无需进一步工作；范围是否应更细是设计问题，原文给出的答案是否定的

Z03: 不传递令牌或相关 id

一个 consumer 的令牌与 correId/fetchCorrIds 只能用于它自己的配置与取数操作；攻击者不能把某 consumer 的许可或关联标识移到另一个 consumer 或另一个操作上使用。

令牌与 correId 都不能移到另一个 consumer 上：没有 fetchCorrIds 就得不到任何数据，而另一个持有服务级范围的合法 NF 若未持有这些 id 也不会被服务。一旦它持有另一条 consumer 通知的 id，它就会得到该 consumer 的数据，因为 MFAF 从不将请求方与配置进行比较。

- 结果：不成立（1 次攻击）；是否回答了所问：部分；可用但有保留
- 价值中：通过转移 fetchCorrIds 把权限转给另一个 NF，是规范中该流程的弱点；文稿给出了要素但未明说。
- 文稿中是否有明确表述：不明确。所依据的原文：main:2640, rfc6749:2879, s0040, s0395, s0398, s0429
- 依据：这些 id 只能通过建模的泄露到达攻击者；TLS 按每对 NF 建模；原文中没有按资源的范围，因此未建模更细的授权
- 未决：原文没有说明 MFAF 是否必须将请求方与 fetchCorrIds 所属的 consumer 进行比较，也没有说明这些值是秘密还是一次性的
- 下一步：一条将 fetchCorrIds 绑定到 consumer 的规则，或接受它们为 bearer 值，可以定论；Tamarin+ Agent 可以建模泄露它们的 SCP 路径

Z04: 过期或撤销的 id 取不到任何内容

在 Deconfigure 生效之后，或 fetchInstruction 的 expiry 到期之后，使用旧 correId/fetchCorrIds 的新 Fetch 不再返回数据。

fetchInstruction 中携带的期限在实践中并不是期限：MFAF 按请求的标识符找到数据并应答，因此同样的标识符在期限之后、在 Deconfigure 删除配置之后仍会持续返回数据。原文中没有任何步骤使已经发出的 fetchCorrIds 失效。

- 结果：不成立（2 次攻击）；是否回答了所问：是；可用
- 价值中：它表明原文定义为期限的字段没有任何步骤去执行它，并给出了数据仍可取回的确切条件。
- 文稿中是否有明确表述：不明确。所依据的原文：main:2461, s0398, s0076, s0122, s0123, s0124
- 依据：配置生命周期展开到三次取回（bounded.pv）；超出部分不作主张；授权不在 bounded.pv 中；期限被建模为一步，因此第二次和第三次取回落在其之后
- 未决：原文从未说明当 Fetch 在 expiry 之后或配置被删除之后到达时 MFAF 必须做什么
- 下一步：第 4.3.2.2 条中关于 expiry 检查以及 Deconfigure 对未到期 fetchCorrIds 影响的表述可以定论；更多取回轮次的有界模型随后可以展示修复

S03: 重配置后无陈旧数据

配置更新生效之后，基于旧配置的 fetchInstruction 或旧 fetchCorrIds 不能使 MFAF 回到旧配置状态或返回旧配置下的数据。

更新会替换已存储的配置，但不会替换已经发放的标识符：旧的 fetchInstruction 仍可继续使用，并返回更新前的数据以及更新前的 correId。只有一条新的通知，携带新的 fetchCorrIds，才能让 consumer 向前推进。

- 结果：不成立（1 次攻击）；回答了所问：是；可用
- 价值 中：更新路径正是陈旧值可能存活的地方，而原文对此没有给出任何规则。
- 文稿中是否陈述：不明确。所依据的原文：main:2362, s0059, s0060, s0062, s0124, s0395
- 依据：bounded.pv 将一份配置依次展开为 create、notify、fetch、update、fetch、deconfigure、fetch；原文中不存在配置版本，因此没有对其建模；授权被排除在 bounded.pv 之外
- 待定：更新是否意在取消尚未完成的 fetchInstruction，原文未作说明
- 下一步：要么将配置版本带入通知和 Fetch，要么明确规定更新会使先前的 fetchCorrIds 失效，才能解决这一问题

S03.13: 旧指令不能存活超过更新

配置更新生效之后，基于旧配置的 fetchInstruction 或旧 fetchCorrIds 不能使 MFAF 返回旧配置下的数据，也不能使 consumer 把旧数据当作新配置的数据接受。

更新之后，consumer 保留旧配置的 correId 和 fetchCorrIds，用它们取数，收到带有旧 correId 的旧数据，并把它当作自己所知配置的数据接受。响应携带的是这些标识符所属配置的 correId，而这正是 consumer 已经持有的值。

- 结果：不成立（1 次攻击）；回答了所问：是；可用
- 价值 中：它表明对 consumer 而言更新前的数据与当前数据无法区分，而原文并未涉及这一点。
- 文稿中是否陈述：不明确。所依据的原文：main:2362, s0395, s0124, s0059
- 依据：展开了一次更新和一次后续取数；consumer 将响应与其存储的 fetchInstruction 的 correId 进行比较，如 s0395 所述；授权被排除在 bounded.pv 之外
- 待定：consumer 没有任何途径得知其 fetchInstruction 背后的配置已经改变
- 下一步：引入配置版本，或在每次更新后发送新通知，才能让 consumer 分辨出差异

S04: 配置、通知与取回链式关联

取数阶段使用的 fetchCorrIds 来自通知阶段为同一 consumer 生成的那次通知；通知阶段的 correId 来自配置阶段为同一 consumer 建立的配置；三个阶段之间不存在跨 consumer 或跨配置的错接。

在单个 consumer 内部，三个步骤是链式关联的：通知携带为其命名的配置的 correId，数据则跟随 consumer 发送的标识符。跨 consumer 时，这条链只是值的传递：持有另一 consumer 标识符的一方也能得到服务，因此这条链并不绑定到配置上。

- 结果：不成立（1 次攻击）；回答了所问：部分；有保留地可用
- 价值 中：这条链在单份配置内部成立，在配置之间断裂，这正是 Z03 和 A08 所指出的缺口。
- 文稿中是否陈述：不明确。所依据的原文：main:2362, s0035, s0047, s0052, s0122, s0124, s0395
- 依据：每对 NF 之间使用 TLS；MFAF 按配置建模，每份配置一条通知；标识符被泄露
- 待定：配置、其通知与其数据之间的内部关联未被规定
- 下一步：用一句话说明 MFAF 如何将 fetchCorrIds 与配置关联起来即可解决；模型随后可以存储该关联

R02: 通知至多被接受一次

同一逻辑 MFAF 通知或同一逻辑 Fetch 请求，在选定范围内至多被接收方接受一次。

通知除了其 correId 之外没有其他身份，因此同一通知的第二次投递会被再次接受并存储。原文并未确定这种重复是否可能发生：SBI 信令应支持重放保护，这能阻止网络重放，但无法阻止处理该消息的一方的重试。

- 结果：不成立（1 次攻击）；回答了所问：是；有保留地可用
- 价值 低：该流程本身没有定义重复检测，但该失败能否被触发取决于传输层，而任务未将其纳入。
- 文稿中是否陈述：不明确。所依据的原文：ts-33-501:2865, s0140, s0395, s0148, s0149
- 依据：模型的 TLS 抽象未提供原文所要求的 SBI 重放保护，因此该轨迹通过模型信道上的重放到达第二次接受；consumer 存储通知且不做任何比较
- 待定：重复投递是否会在部署中发生（SCP 重试）不在所选原文单元范围内
- 下一步：对 SCP 或 HTTP 层建模可以说明重复是否能被引发；API 本身不提供重复检测

R03: 取回至多被执行一次

同一逻辑取数操作（同一组 fetchCorrIds 对应的一次取数）在选定范围内至多提交一次安全相关副作用；请求可以重传，也可以返回缓存响应。

取数读取已存储的数据，不改变任何东西，因此重复执行会返回相同数据，MFAF 的状态不会移动。原文未给该操作赋予任何副作用，这正是该主张所要求的。

- 结果：成立；回答了所问：是；有保留地可用
- 价值 低：该性质成立是因为该操作是一次读取；仅作为过期缺口的对照记录才有价值。
- 文稿中是否陈述：否。所依据的原文：s0122, s0124, s0374, s0376
- 依据：model.pv 中每份配置只建模一次取数，因此至多一次的解读在该模型中按构造成立；重复取数属于 bounded.pv 中 Z04 和 S03 的讨论范围
- 待定：取数可以重复多少次未被说明
- 下一步：就所写的性质而言无须处理；截止时间之后的重复取数属于 Z04 的范围

K02: 数据仅对其消费者可见

MFAF 通知与 Fetch 响应中携带的数据或分析结果，只对配置中指定的 consumer 或通知端点可见，其他合法 NF 与外部攻击者不能获得。

网络始终无法获得数据或分析结果：通知和取数响应在 MFAF 与 consumer 之间端到端保护。持有该通知标识符的第二个具有服务范围的合法 NF 一旦拿到标识符也能获得这些数据；没有标识符，服务范围对它毫无用处。

- 结果：不成立（1 次攻击）；回答了所问：部分；有保留地可用
- 价值 中：数据的机密性完全取决于 fetchCorrIds 的机密性，而原文从未对此提出要求。
- 文稿中是否陈述：不明确。所依据的原文：ts-33-501:2915, s0124, s0140, s0376, s0429
- 依据：每对 NF 之间使用 TLS，且无其他信道；数据是不透明的，因此不对其内容进行推断建模；标识符被泄露给第二个 NF
- 待定：原文对谁可以兑换 fetchCorrIds 只字未提，因此数据的机密性取决于这些标识符的机密性
- 下一步：将标识符绑定到 consumer，或使其一次性使用，即可使答案为是

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，以及每次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_notify	1 of 2	a03_owner_ctl (depends_on_ambiguity)	2 proved, 1 attack
A08	proved		a08_cfg	1 of 2	a03_owner_ctl (depends_on_ambiguity)	2 proved, 1 attack
Z01	proved		z01_fetch	2 of 3		3 proved
Z03	proved		a03_owner	1 of 2	a03_owner_ctl (depends_on_ambiguity)	2 proved, 1 attack
Z04	none				z04_expiry (holds_in_source); z04_delete (holds_in_source)	2 attack
S03	none				s03_update (holds_in_source)	1 attack
S04	proved		a08_cfg	1 of 2	a03_owner_ctl (depends_on_ambiguity)	2 proved, 1 attack
R02	none				r02 (depends_on_ambiguity)	1 attack
R03	proved		r03	checked		1 proved
K02	proved		k02_secret	checked	k02_rogue_ctl (depends_on_ambiguity)	2 proved, 1 attack
A08.11	proved		a03_notify		a03_owner_ctl (depends_on_ambiguity)	2 proved, 1 attack
Z01.12	proved		z01_fetch	checked		2 proved
S03.13	none				s03_accept (holds_in_source)	1 attack

逐条主张，按分析者对每条主张的措辞；每条主张的答案即证明器的答案：

目标	主张	攻击者	回答	查询	对照
A03	当 analytics consumer 接受一条带有 correId 和 fetchCorrIds 的 MFAF 通知时，MFAF 确实向它发送过带有这些值的该通知。	`network`	proved	a03_notify	none
A03	Fetch 所服务的对象是该配置的 consumer；攻击者还持有一条通知的 correId、fetchUri 和 fetchCorrIds。	`ids_disclosed`	attack (depends_on_ambiguity)	a03_owner_ctl（a03_owner 的对照查询）	
A03	当 MFAF 接受一个带有 fetchCorrIds 的 Fetch 时，它此前为某个配置发送过一条携带这些 fetchCorrIds 的通知，而该配置的 consumer 正是它通知的对象。	`network`	proved	a03_fetch	checked
A08	consumer 所接受的通知携带的 correId 来自一个将该 consumer 命名为 notificationURI 的配置。	`network`	proved	a08_cfg	none
A08	MFAF 只把数据返回给这些 fetchCorrIds 所属配置中指定的 consumer；攻击者还持有一条通知的 correId、fetchUri 和 fetchCorrIds。	`ids_disclosed`	attack (depends_on_ambiguity)	a03_owner_ctl（a03_owner 的对照查询）	
A08	一方收到的数据对应于该方在自己的 Fetch 中放入的 fetchCorrIds，而不是另一方的。	`network`	proved	a08_data	checked

目标	主张	攻击者	回答	查询	对照
Z01	MFAF 接受的每个 Fetch 之前，授权服务器都向请求者签发了针对 fetch 服务范围（scope）的访问令牌。	`network`	proved	z01_fetch	checked
Z01	MFAF 存储的每个 Configure 之前，都向配置方 NF 签发了针对配置服务范围（scope）的访问令牌。	`network`	proved	z01_cfg	none
Z01	持有自身令牌的合法 NF 无法在令牌未经检查操作范围（scope）及其自身主体的情况下进行配置或获取数据。	`rogue_nf`	proved	z01_fetch	checked
Z03	另一个持有自身凭据和服务范围（scope）的合法 NF 无法将这些凭据变为另一个 consumer 的数据：它只会被提供它曾收到的那条通知的 id。	`rogue_nf`	proved	a03_owner	none
Z03	一个 consumer 的 correId 和 fetchCorrIds 无法被另一个 NF 兑换以获取相同数据；攻击者还持有一条通知的 correId、fetchUri 和 fetchCorrIds。	`ids_disclosed`	attack (depends_on_ambiguity)	a03_owner_ctl（a03_owner 的对照查询）	
Z03	令牌、correId 或 fetchCorrIds 无法被用于另一个 consumer 的配置或另一个操作：没有这些值，一方就得不到服务。	`network`	proved	a03_owner	checked
Z04	在 fetchInstruction 携带的截止时间之后，使用旧 fetchCorrIds 的 Fetch 不返回数据。	`network`	attack (holds_in_source)	z04_expiry	
Z04	配置被删除后，使用旧 correId 和 fetchCorrIds 的 Fetch 不返回数据；攻击者还持有这些旧标识符。	`ids_disclosed`	attack (holds_in_source)	z04_delete	
S03	配置被更新为返回其更早的数据后，旧 fetchInstruction 的 id 无法再被使用；攻击者还持有这些旧标识符。	`ids_disclosed`	attack (holds_in_source)	s03_update	
S04	通知携带的 correId 来自将该 consumer 命名为 notificationURI 的配置，因此配置步骤和通知步骤不会在 consumer 之间交叉。	`network`	proved	a08_cfg	none
S04	与配置指定的 consumer 不同的 consumer 永远不会被提供该配置的数据；攻击者还持有一条通知的 correId、fetchUri 和 fetchCorrIds。	`ids_disclosed`	attack (depends_on_ambiguity)	a03_owner_ctl（a03_owner 的对照查询）	
S04	收到的数据遵循同一方发送的 fetchCorrIds，因此 fetch 阶段不会与另一个 consumer 的 fetch 交叉。	`network`	proved	a08_data	checked
R02	一条逻辑 MFAF 通知至多被接受一次：第二次投递就是第二次接受。	`network`	attack (depends_on_ambiguity)	r02（r03 的对照查询）	
R03	一次逻辑 fetch 至多有一个与安全相关的效果：它读取已存储的数据且不改变任何东西，因此重传返回相同的数据。	`network`	proved	r03	checked
K02	外部攻击者永远无法获知通知和 fetch 响应中携带的 analytics 数据或分析结果。	`network`	proved	k02_secret	checked
K02	其他合法 NF 都无法获取某个 fetchCorrIds 的数据；攻击者还持有一条通知的 correId、fetchUri 和 fetchCorrIds。	`ids_disclosed`	attack (depends_on_ambiguity)	k02_rogue_ctl（a03_owner 的对照查询）	

目标	主张	攻击者	回答	查询	对照
K02	另一个合法 NF 只有在某条通知的 fetchCorrIds 被泄露给它时才能获取数据：仅有服务范围（scope）并不能让它得到数据。	`rogue_nf`	proved	a03_owner	checked
A08.11	一个 consumer 所接受的 correId 和 fetchCorrIds 来自 MFAF 针对该 consumer 作为 notificationURI 的那份配置所发送的通知。	`network`	proved	a03_notify	none
A08.11	所接受的通知对应于一份将该 consumer 和该 correId 命名为对象的配置。	`network`	proved	a08_cfg	none
A08.11	MFAF 只为它向同一个 consumer 发出的那条通知接受 Fetch；攻击者还持有一条通知的 correId、fetchUri 和 fetchCorrIds。	`ids_disclosed`	attack (depends_on_ambiguity)	a03_owner_ctl（a03_owner 的对照查询）	
Z01.12	通过 TLS 认证但只持有另一个服务令牌的 NF 永远不会被提供 Fetch：仅凭身份认证无法检索数据。	`network`	proved	z01_noauth	checked
Z01.12	只有当一方的令牌携带 fetch 服务范围（scope）时才会为其执行 Fetch：NRF 签发的服务范围（scope）既是必要的也是充分的。	`rogue_nf`	proved	z01_fetch	checked
S03.13	旧数据无法被当作更新后配置的数据来接受；攻击者还持有旧的 correId 和 fetchCorrIds。	`ids_disclosed`	attack (holds_in_source)	s03_accept	

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，并把每个性质交给 ProVerif；第 3 节中的每个结果都是证明器给出的回答。

ProVerif+ Agent 会话 session-20261114T230127，2026-11-14。状态：已完成。模型：deepseek-v4-1-flash-261017。步数：43，证明器 2 秒，实际耗时 16 分钟，成本 0.130 USD。

ProVerif：运行了 59 个证明，58 个已判定，1 个无答案，2 秒，记录了 38 个回答。

分析了什么

协议	MFAF 配置的通知与获取
版本	19.7.0
分析范围	一个 DCCF 配置者、MFAF、一个 analytics consumer；已收集聚合数据，配置→fetchInstruction 通知→带 fetchCorrIds 的 Fetch→数据响应，一个有界传输。配置过程必须纳入，不能发明不存在的 3ca Subscribe 操作；排除上游采集及复杂处理。
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	477 个编号原文单元（195817 字节），来自下面列出的章节。本报告中的 s0001 之类的引用就是其中一个原文单元

分析所依据的文稿章节：

文稿	章节	标题
main	4.2	Nmfaf_3daDataManagement Service
main	4.2.1	服务描述
main	4.2.1.1	概述
main	4.2.1.2	服务架构
main	4.2.1.3	网络功能
main	4.2.1.3.1	消息框架适配功能（MFAF）
main	4.2.1.3.2	NF 服务消费者
main	4.2.2	服务操作
main	4.2.2.1	引言
main	4.2.2.2	Nmfaf_3daDataManagement_Configure 服务操作
main	4.2.2.2.1	概述
main	4.2.2.2.2	MFAF 的初始配置
main	4.2.2.2.3	更新 MFAF 的配置
main	4.2.2.3	Nmfaf_3daDataManagement_Deconfigure 服务操作
main	4.2.2.3.1	概述

文稿	章节	标题
main	4.2.2.3.2	移除 MFAF 配置
main	4.3	Nmfaf_3caDataManagement Service
main	4.3.1	服务描述
main	4.3.1.1	概述
main	4.3.1.2	服务架构
main	4.3.1.3	网络功能
main	4.3.1.3.1	消息框架适配功能（MFAF）
main	4.3.1.3.2	NF 服务消费者
main	4.3.2	服务操作
main	4.3.2.1	引言
main	4.3.2.2	Nmfaf_3caDataManagement_Fetch 服务操作
main	4.3.2.2.1	概述
main	4.3.2.2.2	从 MFAF 检索数据或分析
main	4.3.2.2A	Nmfaf_3caDataManagement_Subscribe 服务操作
main	4.3.2.3	Nmfaf_3caDataManagement_Notify 服务操作
main	4.3.2.3.1	概述
main	4.3.2.3.2	关于已订阅数据或分析的通知
main	5.1	Nmfaf_3daDataManagement Service API
main	5.1.1	引言
main	5.1.2	HTTP 的使用
main	5.1.2.1	概述
main	5.1.2.2	HTTP 标准标头
main	5.1.2.2.1	概述
main	5.1.2.2.2	内容类型
main	5.1.2.3	HTTP 自定义标头
main	5.1.3	资源
main	5.1.3.1	概述
main	5.1.3.2	资源：MFAF 配置
main	5.1.3.2.1	描述
main	5.1.3.2.2	资源定义
main	5.1.3.2.3	资源标准方法
main	5.1.3.2.3.1	POST
main	5.1.3.2.4	资源自定义操作
main	5.1.3.3	资源：单个 MFAF 配置
main	5.1.3.3.1	描述
main	5.1.3.3.2	资源定义

文稿	章节	标题
main	5.1.3.3.3	资源标准方法
main	5.1.3.3.3.1	PUT
main	5.1.3.3.3.2	DELETE
main	5.1.4	无关联资源的自定义操作
main	5.1.5	通知
main	5.1.6	数据模型
main	5.1.6.1	概述
main	5.1.6.2	结构化数据类型
main	5.1.6.2.1	引言
main	5.1.6.2.2	类型：MfafConfiguration
main	5.1.6.2.3	类型：MessageConfiguration
main	5.1.6.2.4	类型：MfafNotiInfo
main	5.1.6.2.5	类型：MfafTransferInfo
main	5.1.6.2.6	类型：TargetMfafNotiInfo
main	5.1.6.3	简单数据类型与枚举
main	5.1.6.3.1	引言
main	5.1.6.3.2	简单数据类型
main	5.1.6.4	描述备选数据类型或数据类型组合的数据类型
main	5.1.6.5	二进制数据
main	5.1.7	错误处理
main	5.1.7.1	概述
main	5.1.7.2	协议错误
main	5.1.7.3	应用错误
main	5.1.8	特性协商
main	5.1.9	安全
main	5.2	Nmfaf_3caDataManagement Service API
main	5.2.1	引言
main	5.2.2	HTTP 的使用
main	5.2.2.1	概述
main	5.2.2.2	HTTP 标准标头
main	5.2.2.2.1	概述
main	5.2.2.2.2	内容类型
main	5.2.2.3	HTTP 自定义标头
main	5.2.3	资源
main	5.2.4	无关联资源的自定义操作
main	5.2.5	通知

文稿	章节	标题
main	5.2.5.1	概述
main	5.2.5.2	MFAF 通知
main	5.2.5.2.1	描述
main	5.2.5.2.2	目标 URI
main	5.2.5.2.3	标准方法
main	5.2.5.2.3.1	POST
main	5.2.5.3	提取通知
main	5.2.5.3.1	描述
main	5.2.5.3.2	目标 URI
main	5.2.5.3.3	标准方法
main	5.2.5.3.3.1	POST
main	5.2.6	数据模型
main	5.2.6.1	概述
main	5.2.6.2	结构化数据类型
main	5.2.6.2.1	引言
main	5.2.6.2.2	类型：NmfafDataRetrievalNotification
main	5.2.6.2.3	类型：FetchInstruction
main	5.2.6.2.4	类型：NmfafDataAnaNotification
main	5.2.6.3	简单数据类型与枚举
main	5.2.6.3.1	引言
main	5.2.6.3.2	简单数据类型
main	5.2.6.4	描述备选数据类型或数据类型组合的数据类型
main	5.2.6.5	二进制数据
main	5.2.7	错误处理
main	5.2.7.1	概述
main	5.2.7.2	协议错误
main	5.2.7.3	应用错误
main	5.2.8	特性协商
main	5.2.9	安全

协议的目的，如源文步骤在文稿中所定位的那样：一个 DCCF 配置者、MFAF、一个 analytics consumer；已收集聚合数据，配置→fetchInstruction 通知→带 fetchCorrIds 的 Fetch→数据响应，一个有界传输。配置过程必须纳入，不能发明不存在的 3ca Subscribe 操作；排除上游采集及复杂处理。

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所述的协议目的进行了对照。名称、某一性质所涉及的内容以及文稿所述内容，均是分析者的解读；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	通知与取回的相关性	the analytics consumer, assured that a notification it accepts carries the correId of its configuration, and the MFAF, assured that a Fetch it accepts carries fetchCorrIds of a fetchInstruction it really sent	通知携带 correId 以及 fetchInstruction（s0140-s0143、s0395）；Fetch 携带该通知中先前提供的 fetchCorrIds（s0120-s0121）；MFAF 根据请求的参数找到数据（s0122-s0124）。	s0139, s0140, s0141, s0143, s0120, s0121, s0122, s0395, s0398
A08	取回使用其自身通知的 id	the analytics consumer, assured that the fetchUri and fetchCorrIds it accepts come from the notification for its own configuration, and that the MFAF returns the data of the fetchCorrIds it requested and not another configuration's	fetchUri 由 MFAF 在 FetchInstruction 中分配（s0366、s0398），响应是与这些取回关联标识符相关的已存储数据（s0376、s0124）。	s0366, s0398, s0374, s0376, s0124, s0120, s0121, s0395
Z01	每个操作都携带授权	the MFAF, assured that each Configure, Notify and Fetch it performs or accepts is authorized by a token the NRF issued for that operation's service	两个 API 的访问都可以用 OAuth2 客户端凭据授权，NRF 是授权服务器，消费者在消费该服务之前先获取令牌（s0304-s0306、s0426-s0428）；每个 NF 都应校验所有传入消息（ts-33-501:2875）。	main:1982, main:2634, ts-33-501:2875, s0304, s0305, s0306, s0426, s0427, s0428
Z03	不传递令牌或相关 id	the analytics consumer and the MFAF: a consumer's token, correId and fetchCorrIds cannot be used for another consumer's configuration or for another operation	correId 由 NF 服务消费者选择，通知仅凭它来标识（s0040、s0395）；fetchCorrIds 是 MFAF 生成的字符串，未写明归属者（s0398）；该服务只有一个范围，没有按资源划分的范围（s0429）；RFC 6749 指出，它没有提供任何方法让资源服务器确保客户端出示的令牌是签发给该客户端的（rfc6749:2879）。	main:2640, rfc6749:2879, s0040, s0395, s0398, s0121, s0429
Z04	过期或撤销的 id 取不到任何内容	the MFAF and the analytics consumer: after the configuration is deleted or the fetchInstruction expires, a Fetch with the old correId and fetchCorrIds returns no data	fetchInstruction 携带一个可选的过期时间，作为取回数据的截止时间（s0398）；Deconfigure 删除该个体资源（s0070、s0075-s0076）；MFAF 的 Fetch 步骤没有列出任何对过期时间或配置是否存在的检查（s0122-s0124）。	s0398, s0070, s0075, s0076, s0122, s0123, s0124
S03	重配置后无陈旧数据	the analytics consumer and the MFAF: after an update of the configuration takes effect, an old fetchInstruction or old fetchCorrIds cannot return the data of the old configuration nor make the consumer accept old data as new	更新操作替换由 transRefId 标识的 Individual MFAF Configuration 并将其存储（s0059-s0065）；原文没有定义配置版本，也没有定义在更新时使未完成的 fetchInstruction 失效的规则（s0398）。	s0059, s0060, s0061, s0062, s0064, s0065, s0398

ID	简称	对象	文稿怎么说	出处
S04	配置、通知与取回链式关联	the analytics consumer: the fetchCorrIds it uses come from the notification generated for its configuration, that notification's correId comes from that configuration, and no consumer or configuration is crossed with another	消费者创建配置（s0035），MFAF 将其存储（s0047-s0052），用该配置的 correId 发送通知（s0140、s0395），消费者存储该通知（s0148-s0149），并用其中的 id 执行取回（s0120-s0124）。	s0035, s0047, s0052, s0120, s0121, s0122, s0124, s0140, s0148, s0149, s0395
R02	通知至多被接受一次	the analytics consumer: one logical MFAF notification is accepted at most once	通知仅由 correId 标识，不携带消息标识或序号（s0395）；消费者在接受时存储该通知（s0148-s0149）；SBI 信令应支持重放保护（ts-33-501:2865）。	ts-33-501:2865, s0140, s0141, s0395, s0148, s0149
R03	取回至多被执行一次	the MFAF and the consumer: one logical fetch operation has at most one security-relevant effect; a retransmission or a cached response is allowed	Fetch 只携带取回关联标识符（s0374），MFAF 根据这些标识符找到已存储的数据（s0122-s0124）；原文没有说明任何副作用，也没有说明这些标识符只能使用一次。	s0115, s0122, s0124, s0374, s0376
K02	数据仅对其消费者可见	the analytics data the MFAF holds: the anaNotifications, dataNotif and dataReports of a fetchCorrIds are visible only to the consumer named in the configuration, not to other legitimate NFs nor to the external attacker	通知和取回响应把数据送到配置的 notificationURI 以及请求方（s0140、s0144-s0147、s0376）；通知应符合 SBI 通知要求（s0339），SBI 通信受机密性保护（ts-33-501:2915）。	ts-33-501:2915, s0124, s0125, s0126, s0127, s0140, s0144, s0145, s0146, s0147
A08.11	id 绑定到被通知的消费者	the analytics consumer, assured that the correId and fetchCorrIds in a notification correspond to the notificationURI and correId the configuration names for it, and that the MFAF accepts a Fetch only for the notification it made to that consumer	通知发送到配置的 notificationURI，并携带 MessageConfiguration 的 correId（s0140、s0395、s0268）；消费者存储该通知（s0148-s0149）。	s0039, s0040, s0140, s0141, s0395, s0398
Z01.12	取回要求授权	the MFAF, assured that a Fetch is performed only for a party holding a token issued for the fetch service, so that identity authentication alone does not retrieve data	Fetch 属于 Nmfaf_3caDataManagement API，其访问可以用 OAuth2 授权（s0426-s0427）；该 API 只定义一个范围，没有资源级或操作级的范围（s0429）。	main:2634, main:2640, s0426, s0427, s0429
S03.13	旧指令不能存活超过更新	the analytics consumer and the MFAF: after a configuration update takes effect, an old fetchInstruction or old fetchCorrIds cannot make the MFAF return the data of the old configuration, nor make the consumer accept old data as the data of the new configuration	更新操作替换由 transRefId 标识的 Individual MFAF Configuration 并将其存储（s0059-s0062），以 200 OK 或 204 No Content 应答（s0064-s0065）；原文没有定义配置版本或纪元，也没有定义在更新时使未完成的 fetchInstruction 失效的规则（s0398）。	s0059, s0060, s0061, s0062, s0064, s0065, s0398

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 16 段，没选 382 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	OAuth2 安全考虑的引言指向一份单独的威胁模型，它对本文协议没有提出任何要求。
P003 至 P006	rfc6749:2790-2823	不在范围内	客户端认证和客户端凭据的保密性属于 OAuth2 基线（TS 33.501、TS 29.510），任务已将其固定，并不要求分析；模型从该基线取用令牌签发和 TLS 客户端认证。
P007	rfc6749:2827-2829	Z01, Z03, `rogue_nf`	
P008 至 P010	rfc6749:2831-2850	不在范围内	针对授权服务器和资源所有者交互的措施，不在所分析的流程内。
P011	rfc6749:2859-2865	不在范围内	访问令牌在传输和存储中的保密性属于 OAuth2 与 TLS 基线，不是本通知与获取流程的性质。
P012	rfc6749:2867-2868	不在范围内	本 API 不使用隐式授权；它使用客户端凭据授权。
P013	rfc6749:2870-2872	Z01	
P014	rfc6749:2874-2877	Z03, Z01.12	
P015	rfc6749:2879-2881	Z03, A08, `rogue_nf`, `ids_disclosed`	
P016 至 P033	rfc6749:2885-3003	不在范围内	刷新令牌、授权码、重定向 URI 操纵、资源所有者口令凭据：本 API 不使用的授权类型和流程。
P034 至 P036	rfc6749:3013-3028	不在范围内	请求保密性和端点真实性是任务所固定的 OAuth2 与 TLS 基线的性质。
P037 至 P039	rfc6749:3032-3041	不在范围内	凭据猜测针对口令；本流程使用客户端凭据和令牌，其中不选低熵秘密值。
P040 至 P062	rfc6749:3045-3210	不在范围内	网络钓鱼、跨站请求伪造、点击劫持、代码注入、开放重定向和隐式授权冒充属于 Web 客户端威胁，不在本 NF 到 NF 的 API 流程内。
P063	ts-33-501:2491-2491	没有陈述目标	针对算法协商的降级攻击；本流程不协商任何算法或保护级别，因此该段没有可在此被攻击的内容。
P064 至 P072	ts-33-501:2495-2511	不在范围内	5G AKA 中 UE 与网络之间的认证和授权，不是本 API 流程。
P073 至 P212	ts-33-501:2515-2851	不在范围内	密钥、用户数据和信令保护、签约凭据、用户隐私、gNB 与 ng-eNB 要求：5G 安全的接入网和签约部分，任务已排除。
P213	ts-33-501:2853-2853	不在范围内	对 UDR 的要求，UDR 是所分析流程之外的数据存储。
P214	ts-33-501:2859-2859	不在范围内	信任区划分；模型假定 DCCF、MFAF 和使用方处于彼此只能通过 SBI 互通的信任区。
P215	ts-33-501:2865-2865	R02, R03	

段落	出处	定为	不选的原因
P216	ts-33-501:2867-2867	Z01	
P217	ts-33-501:2869-2869	不在范围内	信任域之间的拓扑隐藏，不是本流程的性质。
P218	ts-33-501:2871-2871	A03, Z01	
P219	ts-33-501:2873-2873	A03, Z03, `ids_disclosed`	
P220	ts-33-501:2875-2875	Z01, A03	
P221 至 P222	ts-33-501:2879-2881	不在范围内	NF 配置文件与证书的匹配以及 NF 发现，不在所分析的操作内。
P223 至 P224	ts-33-501:2883-2885	Z01	
P225 至 P227	ts-33-501:2887-2891	Z01, Z03, `rogue_nf`	
P228 至 P234	ts-33-501:2895-2907	不在范围内	NEF 向应用功能暴露，不是本流程。
P235 至 P237	ts-33-501:2911-2915	不在范围内	SCP 与 NF 之间的双向认证和保护是任务所固定的 SBI 传输基线。
P238	ts-33-501:2917-2917	K02, R02	
P239 至 P240	ts-33-501:2919-2921	不在范围内	SCP 端点的同址部署和 SCP 内部接口的保护，属于本流程之外的部署事项。
P241	ts-33-501:2923-2923	`ids_disclosed`	
P242 至 P303	ts-33-501:2929-3055	不在范围内	漫游互联、SEPP 与 N32-f 要求，不在所分析的 PLMN 内流程内。
P304 至 P312	ts-33-501:3059-3075	不在范围内	SEPP 之间 N32-f 接口上的属性保护，不在本流程内。
P313 至 P318	ts-33-501:3079-3091	不在范围内	IPUPS 和始发网络标识头，不在本流程内。
P319 至 P337	ts-33-501:3097-3141	不在范围内	信令监控以及安全可见性和可配置性，不是本 API 流程的性质。
P338 至 P375	ts-33-501:3149-3231	没有陈述目标	RAN 和 NAS 的加密与完整性算法标识值及算法选择：本流程不选择任何算法，因此这些段落没有可在此被攻击的内容。
P376 至 P398	ts-33-501:6540-11901	不在范围内	N3、EAP-TLS 签约标识隐私、IAB 和 MBSF 要求，不在所分析的流程内。

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 32 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 10 条，没检查 22 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	本流程仅有的密钥和凭据是 TLS 会话和 OAuth2 访问令牌，其保护已被任务固定为 TS 29.500、TS 33.501 和 TS 29.510 基线；流程本身不定义任何密钥或凭据。
K03	强保密／秘密值不可区分性	不在范围内	强保密和不可区分性属于等价路线；文稿没有声称任何值的不可区分性，而本流程的保密性主张以 K02 提出。
K04	密钥用途、方向与会话分离	不适用	本流程不定义密钥层次、方向密钥或会话密钥派生。
D01	消息完整性与数据来源真实性	A03, A08	
A01	对端曾参与性（Aliveness）	A03	
A02	弱一致性认证	A03	
A04	单射一致性认证	R02, R03	
A05	隐式密钥认证	不适用	本流程不建立会话密钥，因此隐式密钥认证无从谈起。
A06	显式密钥确认	不适用	本流程不建立会话密钥，因此密钥确认无从谈起。
A07	身份—凭据—密钥绑定	Z03	
R01	认证／挑战新鲜性	R02	
R04	消息接受顺序／窗口安全	不适用	本流程不定义序号、接受窗口或排序规则。
R05	抗降级／安全策略保持	不适用	本流程不协商任何可被降低的版本、套件或保护级别。
C01	前向保密性（FS）	不适用	不建立会话密钥，因此前向保密无从谈起。
C02	抗密钥泄露冒充（KCI）	不适用	不建立会话密钥，因此密钥泄露冒充无从谈起。
C03	已知密钥安全／泄露隔离	不适用	本流程不定义会话之间的密钥派生，因此泄露隔离没有内容。
C04	临时秘密／部分状态泄露韧性	不适用	本流程不定义临时秘密或部分状态分量。
C05	泄露后安全恢复（PCS）	不适用	本流程不定义棘轮或密钥更新，因此泄露后安全没有内容。
C06	注册凭据泄露后的增强认证	不适用	本流程不使用注册记录或口令认证密钥交换。
C07	抗离线字典攻击	不适用	本流程不使用低熵口令；凭据是客户端凭据和令牌。
C08	指定原语失效下的目标保持	不适用	本流程没有声称在特定原语失效下仍保持保护，任务已将原语固定为 TLS 与 OAuth2 基线。
Z02	委托权限不扩张	A08	
Z05	跨租户／跨服务安全隔离	Z03	
S01	加密 nonce 唯一性	不适用	本流程不为任何加密模式指定 nonce。
S02	重传／密钥重安装安全	R02, R03	
S05	终止与旧状态失效	Z04	
P01	身份匿名性	不在范围内	身份匿名性属于等价路线；文稿没有声称匿名性，且各参与方是已标识的 NF。
P02	会话不可关联性	不在范围内	会话不可关联性属于等价路线；文稿没有声称，且 correId 按设计关联各步骤。
P03	用户／凭据存在性隐藏	不在范围内	隐藏用户或凭据是否存在属于等价路线，文稿未作此声称。
V01	条件性进展／错误恢复	证明器问不了	条件性进展和恢复是活性性质，无法向证明器提出。
V02	无非预期死锁	证明器问不了	无死锁是活性性质，无法向证明器提出。
V03	受限资源滥用／抗放大	证明器问不了	资源滥用和放大是成本性质，无法向证明器提出。

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 22 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 15 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	DCCF	参与方	按定义全部失守	配置器决定它创建的每个配置的 notificationURI 和 correId；如果它不诚实，它自己配置的数据就会流向它指定的位置，而且文稿没有针对不诚实配置器对本流程提出任何性质要求。
S02	MFAF	参与方	按定义全部失守	如果攻击者控制 MFAF，它本身就持有数据和 fetchCorrIds；本流程关于 MFAF 的每条性质按定义都失效。
S03	analytics consumer	参与方	`ids_disclosed`	
S04	NRF	参与方	按定义全部失守	被攻陷的授权服务器可以为任意主体和 scope 签发令牌，因此 Z01 和 Z01.12 按定义失效，文稿也没有对它提出其他性质要求。
S05	其他合法 NF	参与方	`rogue_nf`	
S06	correId	值	`ids_disclosed`	
S07	fetchCorrIds	值	`ids_disclosed`	
S08	fetchUri	值	`ids_disclosed`	
S09	notificationURI	值	攻击者得不到更多	notificationURI 是配置指定的端点地址；攻击者拿到它，除了网络已经通过 SBI 提供的信息之外，得不到更多。
S10	transRefId	值	`ids_disclosed`	
S11	OAuth2 访问令牌	值	按定义全部失守	持有某 NF 访问令牌的一方可以冒充该 NF，因此授权主张对它按定义失效；文稿只对没有令牌的一方提出这些主张。
S12	服务 scope nmfaf-3dadatamanagement	值	`rogue_nf`	
S13	服务 scope nmfaf-3cadatamanagement	值	`rogue_nf`	
S14	expiry	值	攻击者得不到更多	expiry 是 fetchInstruction 中携带的截止时间；持有它除了让攻击者知道同一条通知之外，在网络之外得不到更多，这一点已由 ids_disclosed 覆盖。
S15	配置版本	值	攻击者得不到更多	原文没有定义配置版本；更新和去配置主张所依赖的值已由 ids_disclosed 覆盖。
S16	通知消息实例	值	`ids_disclosed`	
S17	Fetch 请求实例	值	`ids_disclosed`	
S18	取数操作实例	值	`ids_disclosed`	
S19	anaNotifications	值	`ids_disclosed`	
S20	dataNotif	值	`ids_disclosed`	
S21	dataReports	值	`ids_disclosed`	
S22	fetchCorrIds 对应的数据	值	`ids_disclosed`	

每条性质在各攻击者条件下的答案

攻击者由分析者在编写查询之前设定；每个答案都在其中一种攻击者条件下记录。单元格是证明器在该攻击者条件下对该性质的答案；“无答案”表示该性质在该攻击者集合下，证明器留下了未决的主张或没有记录任何内容；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：它读取、丢弃、修改和发送消息，不持有任何参与方的秘密		
`rogue_nf`	同一网络中的另一个合法 NF：它自己的 TLS 凭据，以及 NRF 为服务级 scope nmfaf-3cadatamanagement（以及 nmfaf-3dadatamanagement）签发的访问令牌，但不持有其他消费者的 fetchCorrIds	TS 29.522 为每个服务的整体定义一个 scope，没有资源级或操作级 scope（main:1988, main:2640）；NRF 向 NF 服务消费者签发令牌（ts-33-501:2883-2885）；任务边界要求的是拥有自己凭据的恶意合法 NF。	main:2640, s0307, s0429, s0304, s0426
`ids_disclosed`	一条通知的 correId、fetchUri 和 fetchCorrIds，是在 fetch 操作之外获得的：来自处理或重定向该通知的 SCP，或来自这些值的先前持有者或过期持有者	SCP 处理它转发的消息，每个参与方都必须信任它正确处理这些消息（ts-33-501:2915-2921），而且使用 SCP 时消费者与生产者之间没有双向认证（ts-33-501:2873）；TS 29.522 没有对 fetchCorrIds 和 fetchUri 的机密性、唯一性或一次性使用作出规定，也没有禁止另一个 NF 出示它们（s0398, s0120-s0123）。	ts-33-501:2921, s0398, s0121, s0122, s0123, s0355, s0376

ID	`network`	`rogue_nf`	`ids_disclosed`
A03	成立		不成立
A08	成立		不成立
Z01	成立	成立	
Z03	成立	成立	不成立
Z04	不成立		不成立
S03	不成立		不成立
S04	成立		不成立
R02	不成立		
R03	成立		
K02	成立	成立	不成立
A08.11	成立		不成立
Z01.12	成立	成立	
S03.13	不成立		不成立

原文和目标

来源：477 个原文单元，195817 字节，冻结在 source-delivery-01.json 中。 目标：10 个已批准（A03, A08, Z01, Z03, Z04, S03, S04, R02, R03, K02）；提出的问题：3 个。 目标列表事先按草稿批准，使用 --approve-targets。

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

- approve_package: 批准源文包 protocol-86e6555558a8460ca7956850a3e1

本步骤自报：1 次请求（1 次失败），0.000 USD，输入 101532 tokens，输出 5580 tokens，265 秒。

关于证明器和模型的说明

- Z04：在 ProVerif 中回答，分析者为该目标指定了 changing_state
- S03：在 ProVerif 中回答，分析者为该目标指定了 changing_state
- S03.13：在 ProVerif 中回答，分析者为该目标指定了 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：在那里得到证明，就是针对更弱攻击者的证明；在那里发现攻击，就是真实的攻击。一个省略了对变化值检查（changing_state）的 ProVerif 模型让参与方接受更多：在那里得到证明即成立，那里的攻击可能只是缺少了该检查。Tamarin+ Agent 按原样对这些建模。

分析者的状态审查

这部分是分析者自己对原文的解读，不是证明器的答案。ProVerif 显示下的“是”表示 ProVerif 在展开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说明，它们还带来什么代价则是分析者的解读。“未找到”说明查看了哪些模式，并不证明任何东西。

值	由谁保存	何时变化	何时检查	出处
单个 MFAF 配置：transRefId、correId、notificationURI	MFAF	Configure POST 创建它（s0047-s0052），对 /configurations/{transRefId} 的 PUT 替换它（s0059-s0062），DELETE 删除它（s0075-s0076）	MFAF 读取它，以便将通知发送到其 notificationURI 并设置通知的 correId（s0140, s0395）；在更新和删除时，仅通过请求 URI 中的 transRefId 识别它	s0047, s0048, s0049, s0050, s0051, s0052, s0059, s0060, s0061, s0062, s0075, s0076
一条通知的 fetchCorrIds，以及 fetchInstruction 的 expiry 截止时间	MFAF	MFAF 为 fetchInstruction 生成它们（s0398），当 fetchInstruction 的 expiry 过去（s0398）或配置被移除（s0076）时，它们应失效	每次 Fetch 时：MFAF 根据请求者携带的标识符查找数据（s0122-s0124, s0374）	s0398, s0120, s0121, s0122, s0123, s0124, s0374, s0376
存储的 fetchInstruction：correId、fetchUri、fetchCorrIds	analytics consumer	MFAF 通知它，消费者存储该通知（s0148-s0149）；当这些值背后的配置被更新或移除时，没有任何机制通知它	当数据或分析结果出现在 fetch 响应中时：消费者可以将响应的 correId 与其存储的 fetchInstruction 的 correId 进行比较（s0395）	s0140, s0148, s0149, s0153, s0395

模式	结果	目标	ProVerif 显示	步骤
wrap_or_window	候选	Z04	是：ProVerif 在 bounded.pv 上否证 `z04_expiry`	MFAF 创建该配置，并发送带有 fetchCorrIds 和到期期限的 fetchInstruction（s0140、s0398）。/ 使用方在期限内拉取；MFAF 以 200 返回数据（s0122-s0124）。/ 同一使用方在期限过后再次拉取相同标识符；MFAF 的步骤不将任何内容与期限比较，因此再次以 200 返回相同数据。
desynchronisation	候选	S03.13	是：ProVerif 在 bounded.pv 上否证 `s03_accept`	DCCF 以 correId A 创建该配置；MFAF 以 correId A 和 fetchCorrIds I 通知使用方，使用方存储它们（s0140、s0148-s0149）。/ DCCF 通过 PUT /configurations/{transRefId} 以新内容更新同一配置；MFAF 存储新配置（s0059-s0062）。/ 使用方没有收到新通知，保留 A 和 I；它用 I 拉取，MFAF 以 correId A 提供旧配置的数据，使用方将其接受为它所知配置的数据。
replay	候选	R02	否：未用 ProVerif 尝试	MFAF 发送 correId A 的通知，带有 fetchCorrIds I。/ 使用方接受它，存储它并返回 204（s0148-s0150）。/ 同一通知被再次投递，例如由 SCP 重试；原文中没有任何内容标识通知实例，因此使用方再次接受并存储它。

模式	结果	目标	ProVerif 显示	步骤
update_before_check	候选		否：未用 ProVerif 尝试	具有配置服务范围的 NF 获知一个 transRefId，例如从它曾见过的 Location 头或日志中获知。/ 它用自己的 MfafConfiguration 发送 PUT /nmfaf-3dadatamanagement/<apiVersion>/configurations/{transRefId}（s0059）。/ MFAF 更新并存储该配置，而不将请求者与该配置的所有者比较，因此另一个使用方的 notificationURI 和 correId 被替换。
race_between_sessions	候选		否：未用 ProVerif 尝试	MFAF 持有配置 A 的 fetchCorrIds I 的数据。/ 两方，或使用方的两个副本，同时发送带有 I 的 Fetch。/ 两者都被提供相同数据，而原文中没有任何内容说明第二个必须被拒绝。
distinguishable_failure	候选		否：未用 ProVerif 尝试	使用未知标识符的拉取得到错误响应（s0417）；使用已删除配置的标识符的拉取则被提供数据（模型中为 s0122-s0124）。/ 这一差异告诉发送方给定标识符集的数据是否仍然存在，而这是关于另一个使用方配置的信息。
repeat_one_time_action	未发现			原文中没有声明某个值只安装一次且不得再次安装，而存储的配置仅由原文定义的操作创建、替换或移除。
no_rollback_on_failure	未发现			失败的 Fetch 收到错误响应（s0417），且不改变任何存储值；被拒绝的 Configure、更新或 Deconfigure 使配置保持原样；原文未声明回滚步骤，也不需要回滚。

分析者为目标所命名的内容：

步骤	目标	特征	原因
29	Z04、S03、S03.13	changing_state	这三个目标依赖于 MFAF 保持并改变的状态：创建的配置、由 PUT 更新并由 DELETE 移除的配置，以及 MFAF 发出的 fetchCorrIds 的生命周期，该生命周期受 fetchInstruction 的可选到期限制。ProVerif 无法在任意数量的会话中跟踪这样的值，因此 model.pv 省略了更新、删除和期限；bounded.pv 将单个配置按创建、通知、拉取、更新、拉取、取消配置、拉取的顺序展开。Tamarin+ Agent 按原样对配置存储和期限建模，将能确定在三个拉取之外的情况。
29	A03、A08、A08.11、S04、Z01、Z01.12、Z03、K02、R02、R03	plain_primitives、many_sessions	所有其他目标仅依赖于对称加密、签名、无 MAC 的令牌检查以及成对的 TLS 密钥，除理想原语外没有代数定律，并在三个角色的任意数量会话上提问。答案依赖于端点 TLS 抽象（每对 NF 一条经过认证的加密通道）以及 OAuth2 令牌是只有授权服务器才能生成的签名。

分析者的结尾说明

十三个目标在 TS 29.522 的配置通知与拉取的 ProVerif 模型上得到回答（model.pv：DCCF、MFAF、分析使用方、NRF 令牌、第二个合法 NF 以及一个泄露信息的中间人；bounded.pv：三个拉取轮次内的配置生命周期）。已证明：通知与拉取的对应关系（A03、A08、A08.11）；每次 Configure 和 Fetch 都由针对该操作范围的 NRF 令牌授权，仅持有其他服务令牌的外部者被拒绝（Z01、Z01.12）；拉取至多产生一次效果（R03）；以及数据对网络攻击者保密（K02）。被否证的各项均记录了其攻击并绘制出来：持有通知的 fetchCorrIds 的第二个 NF，用任意针对服务全局范围的令牌即可获得该使用方的数据（A03、A08、A08.11、Z03、K02、S04）；fetchInstruction 的到期和 Deconfigure 不撤销任何内容，因此旧标识符持续返回数据（Z04）；更新永远不会到达使用方的 fetchInstruction，因此更新前的数据被返回并被当作当前数据接受（S03、S03.13）；重复的通知被存储两次，并作为歧义报告，因为模型的 TLS 抽象缺少原文所要求的 SBI 重放保护（R02）。共报告四项发现，其中三项是原文所写规范的弱点，并附有其所需的确切攻击者；在会话预算内，三项建议均未用 check_fix 检查，每项发现都说明了这一点。阻塞项：所有权缺口取决于通知的标识符能否到达第三方，原文未对此作出定论，而 Tamarin+ Agent 可以用 SCP 路径建模；在无所有权检查的情况下通过 transRefId 更新/删除配置这一情况在状态审查中被发现，但未被查询确认，因此它是限制而非发现。