Security analysis of NWDAF VFL client inference API

This report in another language: 中文.

19.8.0 · ProVerif+ Agent report

Documents	5G System; Network Data Analytics Services; Stage 3 (protocol definition)
Properties examined	13: 2 hold, 7 fail (9 attacks), 4 without an answer
Findings	2

Contents

1 Protocol and scope
2 Findings
3 Properties examined
4 Limits of this analysis
Appendix A: Reading of each property
Appendix C: Claims and prover answers
Appendix D: How the analysis was run

1 Protocol and scope

The Nnwdaf_VFLInference service lets an NF service consumer acting as a VFL server subscribe to the inference events of an already trained VFL model that the NWDAF runs as VFL client. The consumers are NWDAFs, AFs and NEFs; a consumer holds a notifCorrId and a notification endpoint, and the NWDAF holds the model and the intermediate inference results. The consumer subscribes with a VflInferSub body, the NWDAF answers 201 Created with the subscriptionId in the Location header, and the NWDAF then sends VflInferNotif messages carrying the notifCorrId and the vflInferResults to the endpoint the consumer named. The consumer may update the subscription, modify it, read it and remove it.

The parties:

- **NF service consumer (NWDAF/AF/NEF)**: VFL server side, subscribes and receives notifications. Holds: OAuth2 access token for the scope nnwdaf-vflinference, notifCorrId, notifUri.
- **NWDAF (VFL client)**: service producer, runs the VFL inference. Holds: VFL model with a fixed identity.

Figure 1 shows the exchange as the documents specify it.

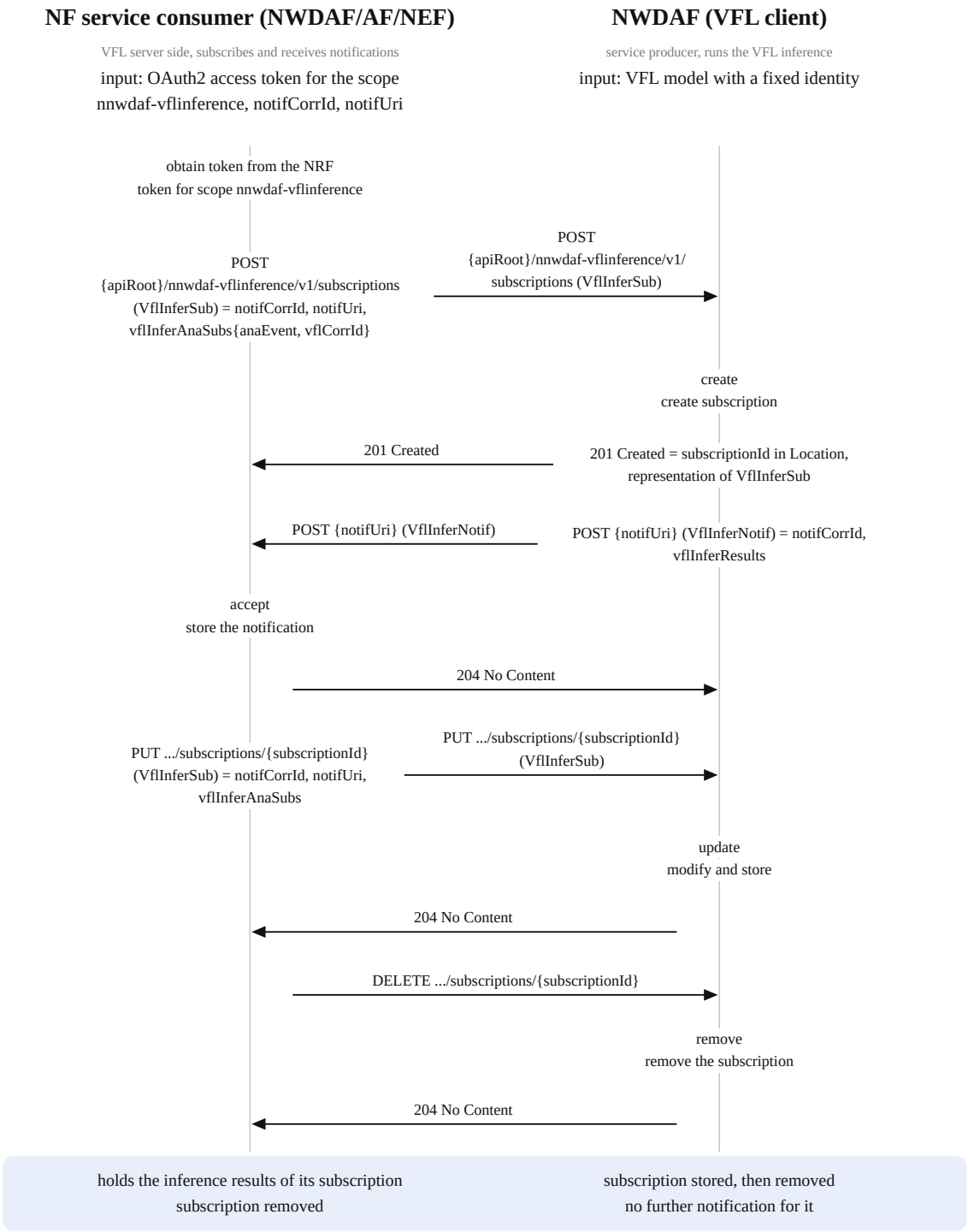


Figure 1. The protocol as specified.

The scope of the analysis is in Appendix D: what was analysed, what was left out, and the sections of the documents it is built from.

2 Findings

The analysis has 2 findings: what follows is not stated in the documents. For 2 of them it is unclear whether the documents state them.

The protocol does not give what the target set asks for: one legitimate NF can take over another consumer's subscription and receive its inference results. An NF that holds a valid token with the single scope nnwdaf-vflinference and knows another consumer's subscriptionId replaces that subscription's notifUri with its own endpoint, and the next VflInferNotif of the victim's subscription is delivered to it; the same NF can delete the victim's subscription. A second failure is that a consumer cannot tell which subscription a notification belongs to, so a notification carrying its notifCorrId but belonging to another subscription is accepted as its own. Access control itself works as the documents describe: only a request backed by a token issued to the requesting NF is accepted, and no token of another NF is derivable. These results come from a model that treats TLS as a private channel, assumes one honest consumer session and assumes the resource identifiers can be known to the attacker; the table has 13 properties, of which 4 have a proof, 4 an attack and the rest a reachability answer or no answer for the attacker they were set for.

The attack figures use the columns of Figure 1. Red is what the attacker does. Yellow marks a message that a party sends although the specified exchange does not have it at that point, and the outcome that the attack breaks.

2.1 A known subscriptionId plus one valid token diverts another consumer's inference results

Kind	Weakness of the protocol as specified
New	Unclear: the documents state it in part, or could not be searched
Value	medium
Properties	A08, Z03, K02, U03, Z04 (section 3)
Source text	s0054, s0055, s0310, s0313, s0045, s0085
Shown by	ProVerif attack queries `a08_put`, `z04_removed`, `k02_attack`; Figures 2, 3, 4, 5, 6 and 7

What happens. A legitimate NF that knows the subscriptionId of another consumer's subscription sends a PUT with its own notifUri and notifCorrId. The NWDAF checks the token but not the owner of the subscription, stores the new notifUri, and sends the next VflInferNotif of that subscription, with its vflInferResults, to the attacker's endpoint. The same request shape deletes the victim's subscription.

What the attacker needs. The attacker is a legitimate NF in the same domain. It holds its own OAuth2 token, which carries the single service-wide scope, and it knows one subscriptionId of another consumer; the source does not make the resource identifier secret and defines no per-resource scope.

What the attacker gains. The attacker receives the intermediate VFL inference results the victim subscribed to, for the analytics event and the UEs of that subscription, and can stop the victim's notifications by deleting its subscription.



Figure 2. A legitimate NF that knows one subscriptionId redirects the subscription to its own endpoint and receives its results. Property A08; attack query a08_put.



Figure 3. One PUT with the victim's subscriptionId delivers its inference results to another NF. Property K02; attack query k02_attack.

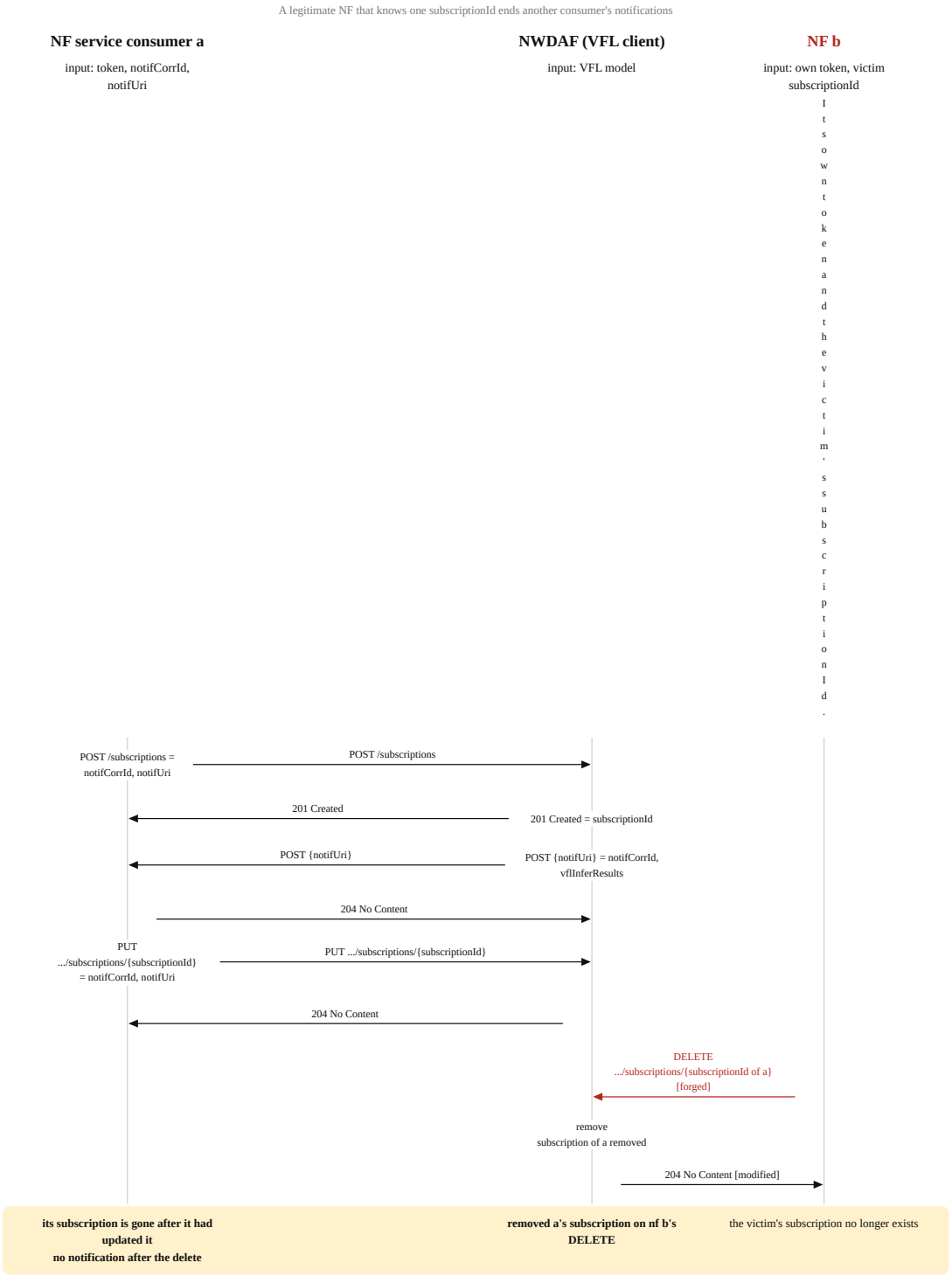


Figure 4. A legitimate NF that knows one subscriptionId ends another consumer's notifications. Property Z04; attack query z04_removed.

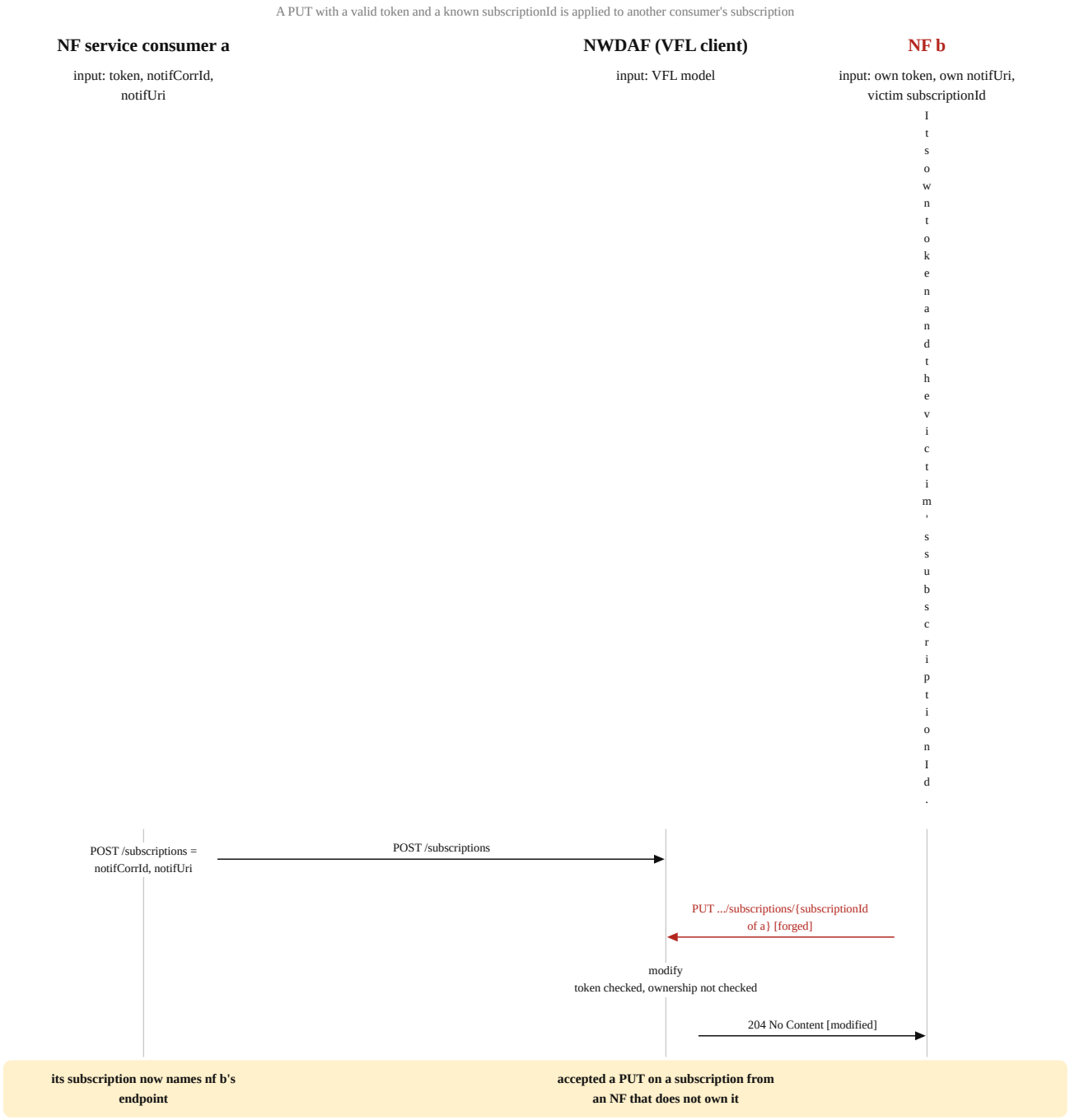


Figure 5. A PUT with a valid token and a known subscriptionId is applied to another consumer's subscription. Property Z03; attack query a08_put.

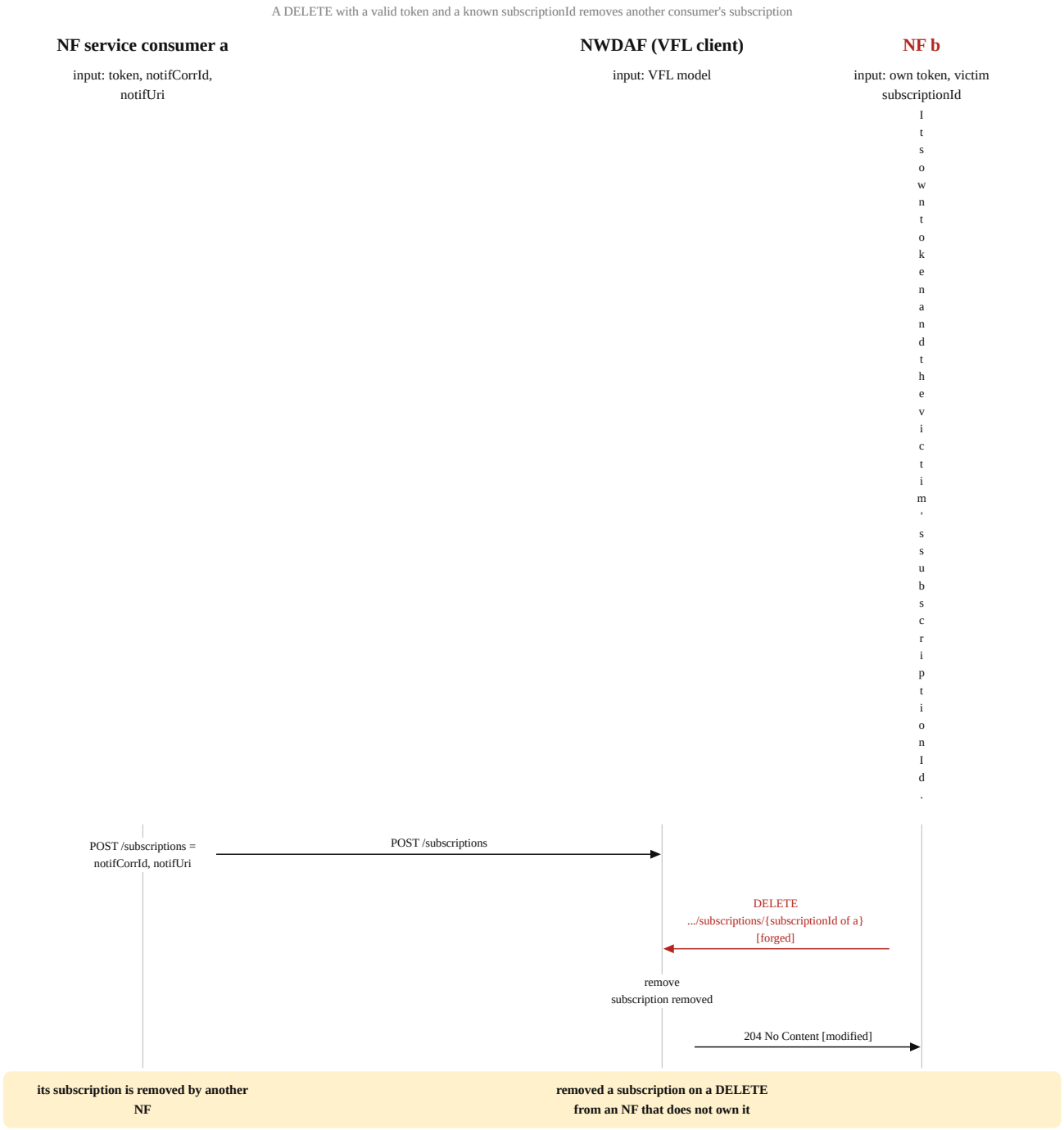


Figure 6. A DELETE with a valid token and a known subscriptionId removes another consumer's subscription. Property Z03; attack query z04_removed.

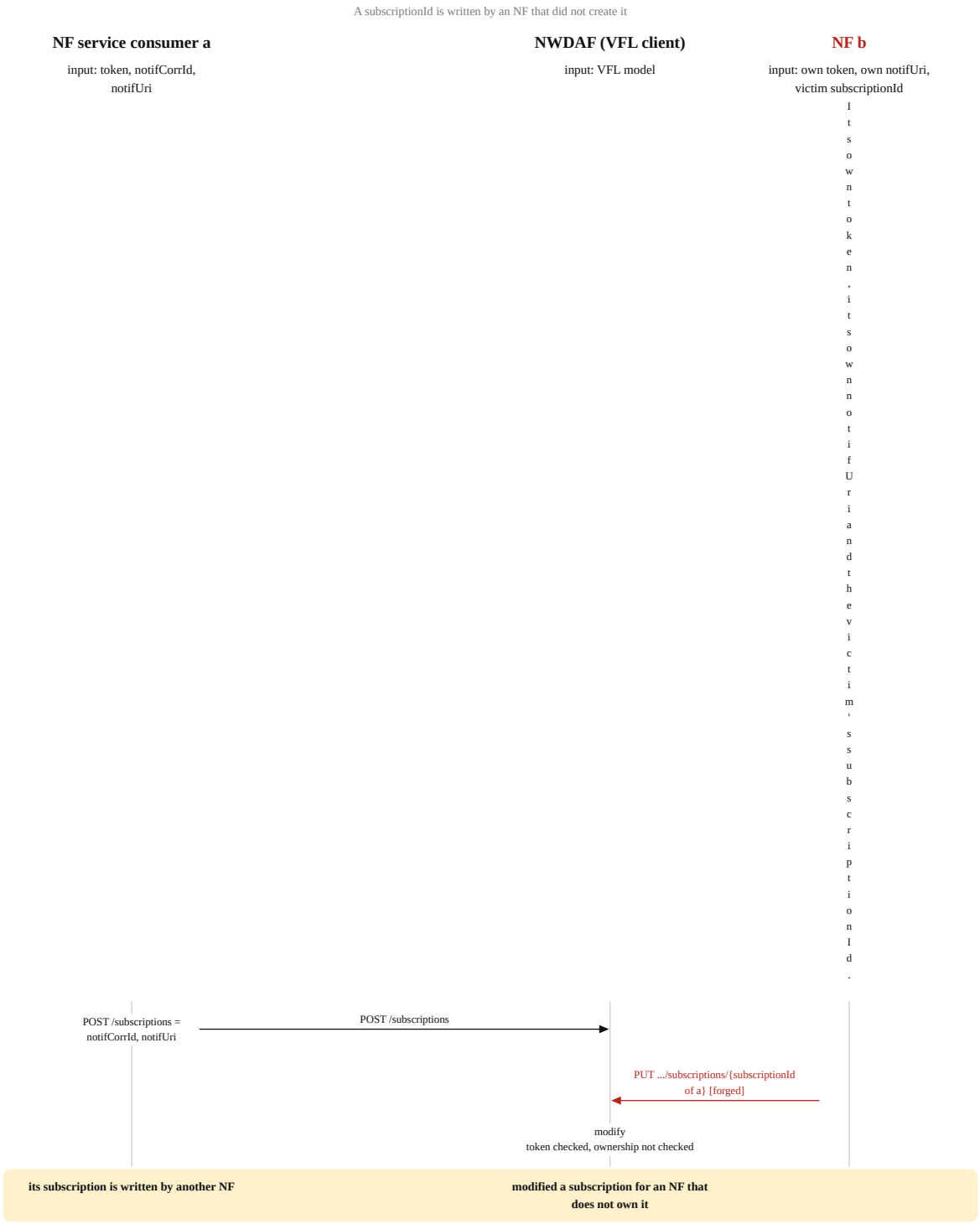


Figure 7. A subscriptionId is written by an NF that did not create it. Property U03; attack query a08_put.

Proposed change. Record the token subject with the created subscription and accept PUT, PATCH, GET and DELETE only when the subject of the presented token is that subject, answering 403 Forbidden otherwise; also define a resource-level authorization rule in clause 5.10.9 instead of a single service-wide scope.

Check of the change with ProVerif. Not checked: The check cannot be added without changing the NWDAF role of the model and re-answering every recorded query, and the budget went to the answers themselves; the fix is a check on an identity the request already carries, so it can be settled in a follow-up run.

2.2 A consumer cannot tell which subscription a notification belongs to

Kind	Weakness of the protocol as specified
New	Unclear: the documents state it in part, or could not be searched
Value	medium
Properties	A03, A08, U01 (section 3)
Source text	s0040, s0102, s0103, s0104, s0275
Shown by	ProVerif attack query `a08_own`; Figures 8, 9 and 10

What happens. The VflInferNotif body carries only the notifCorrId and the results; it names neither the sending NWDAF instance nor the subscription. A second NF therefore subscribes with the notifCorrId and the notifUri of the victim, and the NWDAF delivers that subscription's inference results to the victim's endpoint, where they are accepted and stored as if they belonged to the victim's own subscription.

What the attacker needs. The attacker is a legitimate NF. It knows the victim's notification endpoint, which is a service address, and the notifCorrId the victim uses, which the consumer itself assigns and which the source requires neither unique nor unguessable.

What the attacker gains. The victim accepts and stores inference results about analytics events and UEs that it never subscribed to, and cannot separate them from its own; the results of one subscription reach a party outside that subscription.



Figure 8. A notification about another subscription is accepted as the consumer's own because only the consumer-chosen notifCorrId links them. Property U01; attack query a08_own.



Figure 9. A consumer accepts a notification of another subscription because only the consumer-chosen notifCorrId links them. Property A03; attack query a08_own.

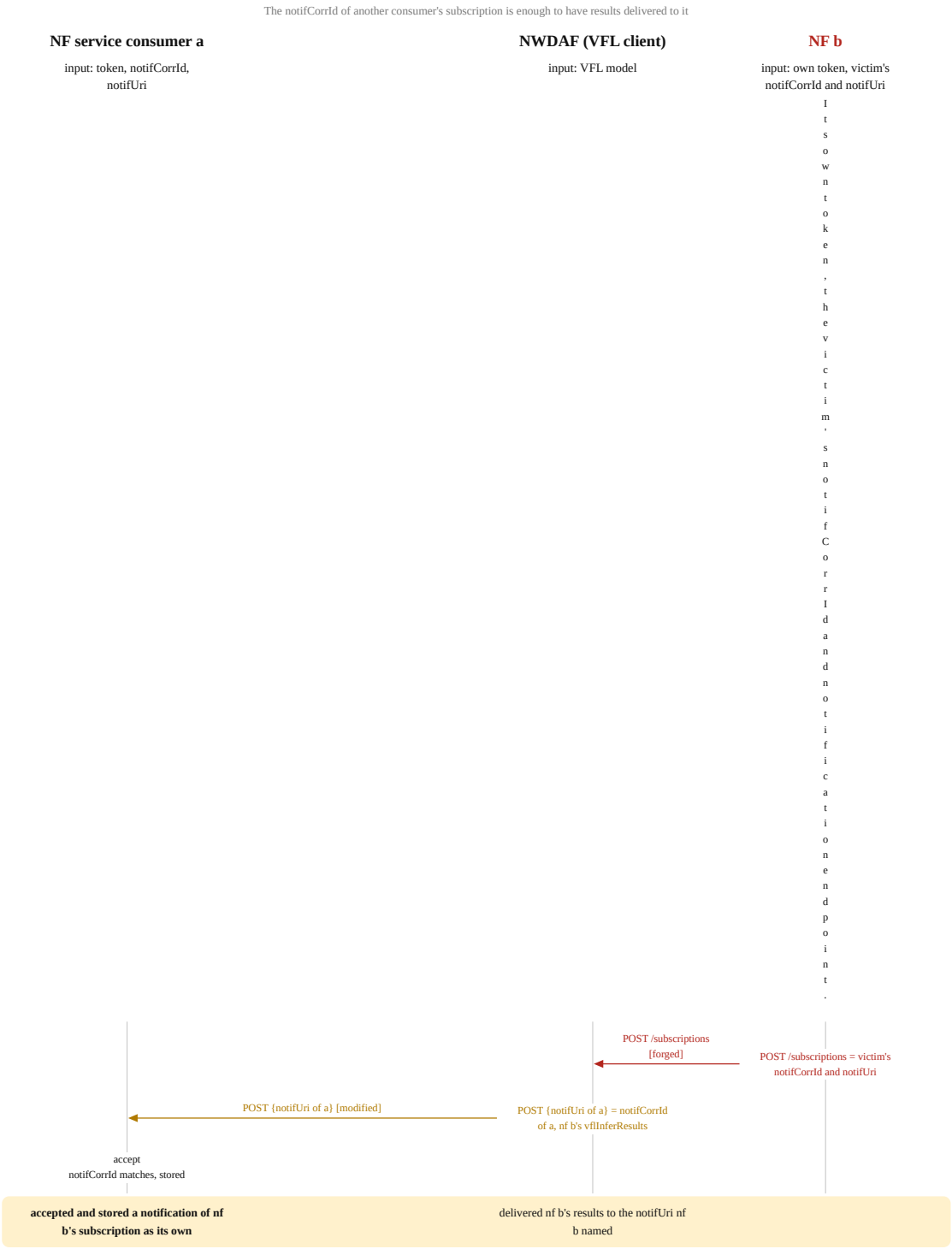


Figure 10. The notifCorrId of another consumer's subscription is enough to have results delivered to it. Property A08; attack query a08_own.

Proposed change. Put the subscriptionId and the identity of the notifying NWDAF instance into the VflInferNotif body, and require the consumer to accept a notification only when the subscriptionId belongs to a subscription it created and the NWDAF is the one it subscribed to.

Check of the change with ProVerif. Not checked: Changing the notification body changes the data model the model was built from; no check was run.

No figure was given for these attacks of a finding: A08: k02_attack, A08: z04_removed, K02: a08_put, K02: z04_removed, U03: k02_attack, U03: z04_removed, Z03: k02_attack, Z04: a08_put, Z04: k02_attack.

3 Properties examined

Each row is one property put to ProVerif. The result is ProVerif's answer on the analyst's model of the protocol; value and whether the documents already state the result are the analyst's judgement, with the reasons in Appendix A.

ID	Property	Compromise assumed	Result	Value	In the documents
A03	**Genuine subscription and notification content.** Whenever the NWDAF accepts a subscription request, there is a run genuinely initiated by a consumer whose consumer identity, notifUri, notifCorrId, vflCorrId and anaEvent match the values the NWDAF accepts; whenever a consumer accepts a notification, there is a genuine send by the NWDAF for that consumer's subscription, with the notifCorrId matching the one at subscription time.	no compromise	Fails (1 attack)	medium	unclear
A08	**Correlation ids bound to own subscription.** The notification correlation identifier, VFL correlation identifier and analysis event of a notification the NWDAF accepts must come from that consumer's currently valid subscription; the notifCorrId of a notification a consumer accepts must correspond to a subscription the consumer itself created; PUT/PATCH/DELETE act only on that consumer's own subscriptionId.	no compromise	Fails (2 attacks)	medium	unclear
Z01	**Token authorizes every accepted operation.** Whenever a subscription creation, update, cancellation or notification delivery is accepted, there must be a valid, unexpired token issued by the authorization server that covers the nnwdaf-vflinference scope, and the token subject must match the requester.	no compromise	Holds	low	yes
Z03	**Operations limited to own subscription.** When a token is used to perform GET/PUT/PATCH/DELETE on a subscriptionId or to receive a notification, that subscriptionId and notifCorrId must belong to a subscription created by the token subject.	no compromise	Fails (2 attacks)	medium	unclear
Z04	**Nothing after unsubscribe or expiry.** After a DELETE succeeds or the NWDAF sends a termination reason, that subscriptionId produces no new notifications; after a token expires it no longer authorizes new subscription creation or modification.	no compromise	Fails (1 attack)	medium	unclear
S05	**Removed subscription correlation unusable.** After a subscription is deleted or terminated, the old subscriptionId or old notifCorrId cannot be used to trigger or accept new notifications.	no compromise	Reachable only	low	unclear
R02	**One acceptance per notification.** The same logical notification, identified by the same notifCorrId, is accepted by the consumer at most once; replaying the same subscription request does not produce a second independent subscription.	no compromise	Reachable only	low	unclear
K02	**Inference results kept from others.** The vflInferResults and the UE identity in a subscription are known only to the consumer that initiated the subscription and to the NWDAF; other legitimate NFs and external attackers cannot derive them.	no compromise	Fails (1 attack)	medium	unclear
K01	**Access tokens not derivable.** An attacker cannot derive a valid access token that another NF obtained from the NRF and that covers the nnwdaf-vflinference scope.	no compromise	Holds	low	yes
U01	**Accepted notification matches own subscription.** A notification a consumer accepts must correspond to its own subscription for the same notifCorrId, vflCorrId and anaEvent.	no compromise	Fails (1 attack)	medium	unclear

ID	Property	Compromise assumed	Result	Value	In the documents
U02	**Results come from inference only.** The results and correlation identifiers in an inference notification come only from the inference process, not from the training process.	no compromise	Reachable only	none	yes
U03	**Subscription ownership and staleness.** A subscriptionId is read and written only by its creator; after cancellation or termination, stale notifications of that subscription are no longer accepted.	no compromise	Fails (1 attack)	medium	unclear
X01	**No redirection of a notification to another consumer.** When the consumer answers a notification with a redirect response, the NWDAF does not deliver the notified vflInferResults to a party other than the consuming NF instance.	a legitimate NF that answers or influences a redirect response	No answer	none	unclear

Goals the documents state that the drafted list of properties did not have, added by the analyst (Appendix D): X01.

The questions asked, and the analyst's answers:

Question	Answer	Rests on
Q1: Model of inference inputs and results / VFL correlation ID / consumer binding	The binding is weak: a consumer correlates a notification with its subscription by the notifCorrId alone, which the consumer itself chooses and the notification repeats without naming the subscription or the sender, so a second NF can have its own results delivered to and accepted by the victim; the VFL correlation identifier of the subscription plays no part in the notification.	A03, A08, U01
Q2: Training and inference contexts must not be mixed	The data model already separates them: the API is the inference service of an already trained model, the notification carries only VflInferResult, and no training message exists in the source units, so no failure is shown and no target depends on it.	U02
Q3: Cross-subscription reads and writes by other legitimate NFs, and handling of stale results	Another legitimate NF can read and replace a subscription as soon as it knows the subscriptionId, because the only authorization is one service-wide scope and no ownership of a resource is checked; and results of a subscription already removed are still accepted by the consumer after it unsubscribes.	Z03, U03, Z04, S05

4 Limits of this analysis

A result here is a result on a model of the protocol: it says what the specified messages and checks give or fail to give, with cryptography taken as ideal, and nothing about an implementation.

What this analysis does not cover, as the analyst states it:

- TLS between an NF and the NWDAF is a private channel: the attacker on the network alone reads and forges nothing, so the network-only rows of the table are the rows of an honest-channel model and say nothing about a broken TLS.
- One honest consumer session and two NWDAF service instances: a proof here is a proof for one consumer, not for any number of consumer sessions.
- subscriptionId and notifCorrId are modelled as values another NF can know; if they were unguessable and never disclosed, the cross-consumer attacks would not apply and the ownership question would not arise.
- The OAuth2 token is an unforgeable term; its scope value, subject format, lifetime and expiry are not modelled, so nothing is said about token expiry or scope narrowing.
- PATCH, GET and the redirect answers of a notification are not modelled, and neither are the SUPI and group identifiers of a subscription.

- The VFL inference itself and the content of vflInferResults are opaque values; nothing is said about the correctness or the privacy of the computation.
- A notification the NWDAF sent before it handled a DELETE is accepted by the consumer afterwards; the source gives no rule either way, so this is reported in the readings of S05, Z04 and U03 and not as a finding, because no recorded attack query shows it.
- Several rows of the table have no recorded answer for the attacker they were set for: where the model has no query that exercises the property for that attacker, the report leaves the row open rather than record a weaker claim in its place.
- ProVerif gave no answer for the last claim of U01; that one is a case for Tamarin+ Agent.

What the host records:

- No answer for: S05, R02, U02, X01.
- The prover gave no answer (Appendix C): U01: u01_late. The property was asked and the prover neither proved it nor found an attack, so the property is shown without an answer.
- Set by the analyst and without a proof or an attack (Appendix D): A08 under `network`; Z01 under `malicious_nf`; Z03 under `network`; Z04 under `network`; S05 under `network`; S05 under `malicious_nf`; R02 under `network`; R02 under `malicious_nf`; U02 under `network`; U02 under `malicious_nf`; U03 under `network`; X01 under `malicious_nf`.

What stays open for each property, and what would settle it, is in Appendix A.

Appendix A: Reading of each property

The analyst's own reading of the prover's answers, not an answer of the prover: what the answers mean for the protocol, what the reading rests on, whether it can be used as it stands, and what it is worth.

A03: Genuine subscription and notification content

Whenever the NWDAF accepts a subscription request, there is a run genuinely initiated by a consumer whose consumer identity, notifUri, notifCorrId, vflCorrId and anaEvent match the values the NWDAF accepts; whenever a consumer accepts a notification, there is a genuine send by the NWDAF for that consumer's subscription, with the notifCorrId matching the one at subscription time.

The subscription direction holds: the NWDAF only creates a subscription for a request a consumer really sent, with the same notifCorrId, and the request is protected by TLS and by the token. The notification direction does not hold: a consumer accepts a notification of another consumer's subscription because nothing but the consumer-chosen notifCorrId links a notification to a subscription.

- Result: Fails (1 attack); answers what is asked: partly; usable with caveats
- Value medium: The failing half follows from the notification data model, which the documents give without saying what it costs; the holding half is what the source requires of the create operation.
- Stated in the documents: unclear
- Rests on: TLS as a private channel; one honest consumer session; notifCorrId treated as known to another NF
- Open: the claim about a notification accepted after the unsubscribe has no answer

A08: Correlation ids bound to own subscription

The notification correlation identifier, VFL correlation identifier and analysis event of a notification the NWDAF accepts must come from that consumer's currently valid subscription; the notifCorrId of a notification a consumer accepts must correspond to a subscription the consumer itself created; PUT/PATCH/DELETE act only on that consumer's own subscriptionId.

Both bindings fail. A PUT on another consumer's subscriptionId is accepted, so the correlation identifiers of a subscription can be replaced by a third party, and a consumer accepts a notification that carries its notifCorrId although the subscription behind it belongs to another NF.

- Result: Fails (2 attacks); answers what is asked: yes; usable with caveats
- Value medium: The documents give one service-wide scope and no ownership rule, and say nothing about a consumer being fed another subscription's notification.
- Stated in the documents: unclear
- Rests on: subscriptionId treated as knowable by another NF; TLS as a private channel

Z01: Token authorizes every accepted operation

Every accepted subscription creation, update, cancellation or notification delivery must be backed by a valid, unexpired token issued by the authorization server that covers the nnwdaf-vflinference scope, and the token subject must match the requester.

Every accepted create, update or delete is backed by a token whose subject is the requesting NF, and the control shows that accepted requests do occur. What the documents leave open, and the model therefore assumes, is the strength of the token itself.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: The security consideration states exactly this: access to the API may be authorized with OAuth2 and the NRF as authorization server.
- Stated in the documents: yes. Where the documents say it: main:310
- Rests on: the token as an unforgeable term; scope and lifetime not modelled
- Open: token expiry and scope narrowing

Z03: Operations limited to own subscription

When a GET/PUT/PATCH/DELETE is performed on a subscriptionId with a token, or a notification is received, that subscriptionId and notifCorrId must belong to a subscription created by the subject of that token.

An operation on another consumer's subscription is accepted whenever the caller holds a token with the service-wide scope and knows the subscriptionId; neither a PUT nor a DELETE is refused for lack of ownership. The results a consumer subscribed to can therefore be read by the party that took its subscription over.

- Result: Fails (2 attacks); answers what is asked: yes; usable with caveats
- Value medium: The documents state the service-wide scope as a rule and name no consequence for the ownership of a subscription; they never say one NF can operate on another's subscription.
- Stated in the documents: unclear
- Rests on: subscriptionId treated as knowable; the single service-wide scope as the only authorization

Z04: Nothing after unsubscribe or expiry

After a successful DELETE, or after the NWDAF sends a termination reason, that subscriptionId produces no new notifications; after the token expires, it no longer authorizes new subscription creation or modification.

A third party can end another consumer's notifications by deleting its subscription, so nothing after unsubscribe is not guaranteed. Whether the consumer still accepts a notification that was already in flight is reachable too. Token expiry is not modelled at all.

- Result: Fails (1 attack); answers what is asked: partly; usable with caveats
- Value medium: The unsubscribe clause states removal and 204 only; the documents state no rule about in-flight notifications and no ownership check on the DELETE.
- Stated in the documents: unclear
- Rests on: subscriptionId treated as knowable; no token lifetime in the model; no ordering rule for the DELETE against a notification in flight
- Open: token expiry

S05: Removed subscription correlation unusable

After a subscription is deleted or terminated, the old subscriptionId or old notifCorrId cannot trigger or accept new notifications.

A notification of a subscription the consumer has asked to remove is still accepted, because the source gives no rule for what happens to what is in flight. Reuse of a removed subscription's identifiers by a third party was not shown separately.

- Result: Reachable only; answers what is asked: partly; usable with caveats
- Value low: A rule the documents omit rather than a stated weakness; nothing in the source says the consumer must drop it.
- Stated in the documents: unclear
- Rests on: the consumer's acceptance of a late notification is modelled from s0104 without a rule; TLS as a private channel
- Open: whether the same notifCorrId can be reused for a new subscription

R02: One acceptance per notification

The same logical notification for the same notifCorrId is accepted by the consumer at most once; replaying the same subscription request does not produce a second independent subscription.

A notification is delivered and accepted once; a second acceptance would need a second send, and the network alone cannot replay a notification because endpoint traffic is under TLS. The model does not show a duplicate delivery by the NWDAF itself.

- Result: Reachable only; answers what is asked: partly; usable with caveats
- Value low: The source states no replay rule for notifications and the model shows no failure, so there is nothing to report beyond a reachability answer.
- Stated in the documents: unclear
- Rests on: TLS as a private channel, which removes replay by the network; one notification pair per subscription in the model
- Open: a repeated send by the NWDAF after a retry

K02: Inference results kept from others

The vflInferResults and the UE identifier in a subscription may only be known to the consumer that initiated that subscription and to the NWDAF; other legitimate NFs and external attackers cannot derive them.

An attacker with the network alone learns no inference result, and the control shows the same query fails once a legitimate NF acts. A legitimate NF that takes over a subscription receives its inference results in clear at its own endpoint, so the results of a subscription are not kept to the subscribing consumer.

- Result: Fails (1 attack); answers what is asked: yes; usable with caveats
- Value medium: The documents state no confidentiality goal for vflInferResults beyond the OAuth2 protection of the API and name no attacker for it.
- Stated in the documents: unclear
- Rests on: TLS as a private channel; subscriptionId treated as knowable; results abstracted to an opaque value

K01: Access tokens not derivable

An attacker cannot derive a valid access token that another NF obtained from the NRF and that covers the nnwdaf-vflinference scope.

No access token of another NF can be derived from the traffic, by the network or by a legitimate NF, and the control shows that accepted requests with tokens do occur. The strength of the token itself is assumed, not shown.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: OAuth2 requires that access tokens cannot be generated, modified or guessed by unauthorized parties.
- Stated in the documents: yes. Where the documents say it: rfc6749:2870
- Rests on: the token as an unforgeable term
- Open: token guessing against a low-entropy scope value

U01: Accepted notification matches own subscription

A notification the consumer accepts must correspond to its own subscription for the same notifCorrId, vflCorrId and anaEvent.

A consumer accepts a notification for another consumer's subscription when the notifCorrId matches, so the correlation identifier alone does not tie a notification to its own subscription. The variant where a notification in flight arrives after the unsubscribe could not be answered: ProVerif found the derivation but no trace.

- Result: Fails (1 attack); answers what is asked: partly; usable with caveats
- Value medium: The documents give the notification no sender and no subscription identifier, and do not say how a consumer is to tell two subscriptions apart.
- Stated in the documents: unclear
- Rests on: notifCorrId treated as knowable; one honest consumer session
- Open: the last-notification claim is undecided

U02: Results come from inference only

The results and correlation identifiers in an inference notification come only from the inference process, not from the training process.

The model contains one notification kind only, the VflInferNotif of the inference, and no training message at all, so a confusion between training and inference cannot arise and the reachability answer shows only that inference results are delivered and accepted. The question is settled by the data model, not by a trace property.

- Result: Reachable only; answers what is asked: no; not usable
- Value none: The service is defined as the inference service of an already trained model with a fixed identity, so the separation is structural and the documents state it.
- Stated in the documents: yes. Where the documents say it: main:4
- Rests on: the model has no training-phase message; results abstracted to an opaque value
- Open: nothing about the inference itself

U03: Subscription ownership and staleness

A subscriptionId is read and written only by its creator; after cancellation or termination, stale notifications for that subscription are no longer accepted.

A subscriptionId is read and written by any NF that holds the service-wide token and knows the identifier, so ownership is not enforced, and a notification in flight is accepted after the unsubscribe. Both parts of the target fail, but the first rests on the identifier being knowable.

- Result: Fails (1 attack); answers what is asked: partly; usable with caveats
- Value medium: The documents give the resource methods without an ownership rule and say nothing about stale results.
- Stated in the documents: unclear
- Rests on: subscriptionId treated as knowable; no ordering rule between the DELETE and a notification in flight

X01: No redirection of a notification to another consumer

When the consumer answers a notification with a redirect response, the NWDAF does not deliver the notified vflInferResults to a party other than the consuming NF instance.

The redirect answers of a notification were added as a goal because the source lets a consumer or an SCP answer a notification with a 307 or 308 and a Location, but the model does not contain that branch, so nothing is established about where the results go after a redirect.

- Result: No answer; answers what is asked: undetermined; not usable
- Value none: No prover answer was asked for it; the goal was added from the operation definition of the notification.
- Stated in the documents: unclear
- Rests on: the redirect path is not modelled
- Open: the whole question

Appendix C: Claims and prover answers

The prover's record for each target: the level reached, the query that reached it, whether a control was checked, and the attacks with the analyst's verdict on each.

target	level	bound	lemma	control	attacks	claims
A03	proved		a03_sub	checked	a08_own (holds_in_source)	1 proved, 1 attack
A08	none				a08_own (holds_in_source); a08_put (holds_in_source)	2 attack
Z01	proved		z01_token	checked		1 proved
Z03	none				a08_put (holds_in_source); z04_removed (holds_in_source)	2 attack
Z04	reachable		s05_late		z04_removed (holds_in_source)	1 reachable, 1 attack
S05	reachable		s05_late			1 reachable
R02	reachable		r02_acc			1 reachable
K02	proved		k02_net	checked	k02_attack (holds_in_source)	1 proved, 1 attack
K01	proved		k01_tok	checked		2 proved
U01	none				a08_own (holds_in_source)	1 attack, 1 undecided
U02	reachable		r02_acc			1 reachable
U03	reachable		s05_late		a08_put (holds_in_source)	1 reachable, 1 attack
X01	none					

Claim by claim, in the analyst's wording of each claim; the answer to each is the prover's:

target	claim	attacker	answer	query	control
A03	Every subscription the NWDAF accepts was requested by a consumer run with that notifCorrId (the attacker has the network only; TLS and the token protect the request)	`network`	proved	a03_sub	checked
A03	Every notification the consumer accepts corresponds to the NWDAF's send for that consumer's own subscription (attacker: a legitimate NF that subscribes with another consumer's notifCorrId and endpoint)	`malicious_nf`	attack (holds_in_source)	a08_own	
A08	The consumer accepts a notification only for its own subscription, with its own notifCorrId (attacker: a legitimate NF choosing another consumer's notifCorrId and endpoint)	`malicious_nf`	attack (holds_in_source)	a08_own	
A08	A PUT acts only on the subscriptionId of the requesting consumer's own subscription (attacker: a legitimate NF with its own token that knows another consumer's subscriptionId)	`malicious_nf`	attack (holds_in_source)	a08_put	
Z01	Every accepted subscription, update or delete is backed by an access token whose subject is the requesting NF (attacker: the network, which holds no token)	`network`	proved	z01_token	checked

target	claim	attacker	answer	query	control
Z03	A PUT on {subscriptionId} acts only on the subscription of the token subject (attacker: a legitimate NF with a valid token that knows another consumer's subscriptionId)	`malicious_nf`	attack (holds_in_source)	a08_put	
Z03	A DELETE on {subscriptionId} acts only on the subscription of the token subject (attacker: a legitimate NF with a valid token that knows another consumer's subscriptionId)	`malicious_nf`	attack (holds_in_source)	z04_removed	
Z04	After a successful DELETE the subscription produces no further notification that the consumer accepts; the situation is reachable in the model	`network`	reachable	s05_late	
Z04	Only the consumer that owns a subscription can delete it, so that a third party cannot stop its notifications (attacker: a legitimate NF that knows the subscriptionId)	`malicious_nf`	attack (holds_in_source)	z04_removed	
S05	A notification that arrives after the consumer has sent the DELETE for its subscription is still accepted; the situation is reachable in the model	`network`	reachable	s05_late	
R02	A notification of a subscription is delivered to and accepted by the consumer once (at most one acceptance per notification in this model)	`network`	reachable	r02_acc	
K02	The vflInferResults of a subscription are not learned by an attacker that holds only the network	`network`	proved	k02_net	checked
K02	The vflInferResults of a subscription reach only the subscribing consumer and the NWDAF (attacker: a legitimate NF that replaces the notifUri of another consumer's subscription with its own)	`malicious_nf`	attack (holds_in_source)	k02_attack (the control of k02_net)	
K01	An attacker cannot derive the access token of an honest NF from the traffic	`network`	proved	k01_tok	checked
K01	A legitimate NF cannot derive the access token of another NF; each token authorizes only its own subject	`malicious_nf`	proved	k01_tok	checked
U01	A consumer accepts a notification only when it corresponds to its own subscription with the same notifCorrId, vflCorrId and anaEvent (attacker: a legitimate NF that requests a subscription with another consumer's notifCorrId and endpoint)	`malicious_nf`	attack (holds_in_source)	a08_own	
U01	A notification accepted after the consumer has asked to unsubscribe still corresponds to the consumer's own subscription	`network`	undecided	u01_late	
U02	A notification carrying VFL inference results is delivered to and accepted by the subscribing consumer; the situation is reachable in the model	`network`	reachable	r02_acc	
U03	Results of a removed subscription are not accepted afterwards; the situation is reachable in the model	`network`	reachable	s05_late	
U03	A subscriptionId is written only by the consumer that created it (attacker: a legitimate NF with a valid token that knows another consumer's subscriptionId)	`malicious_nf`	attack (holds_in_source)	a08_put	

Appendix D: How the analysis was run

An LLM analyst builds the model of the protocol from the documents and puts each property to ProVerif; every result in section 3 is an answer of the prover.

ProVerif+ Agent session session-20261114T221340, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 42, prover 20 s, wall 14 min, cost 0.119 USD.

ProVerif: 57 proofs run, 49 decided, 8 without an answer, 9 s, 19 answers recorded.

What was analysed

Protocol	NWDAF VFL client inference API
Version	19.8.0
Analysed	A pre-trained VFL model with a fixed identity; a trusted AF/NWDAF server and one NWDAF client, subscription, one input/output notification, update and cancellation, with an explicit stage identity; model training and numerical prediction correctness are not included.
Not analysed	The symbolic model does not prove the computational security or post-quantum security of the algorithms; attack traces hold only under explicit assumptions and are not directly equivalent to specification vulnerabilities. Every property must have a reachability/positive-example control. Fixed at Rel-19. The external network attacker and the malicious legitimate NF holding its own credentials are distinguished separately; TLS authenticated encryption is modelled per endpoint, and a legitimate NF cannot read another NF's TLS session. OAuth2 subject/audience/scope/expiry and TLS identity binding follow the attached baseline; a service scope does not automatically equal resource ownership. Monotonic tables cannot model real deletion, and the finite-state lifecycle must be defined explicitly. The goal is the specified API sub-flow, not complete 5G security.
Source text	313 numbered passages (64041 bytes) from the sections listed below. A reference such as s0001 in this report is one of these passages

The sections of the documents the analysis is built from:

Document	Section	Title
main	4.11	Nnwdaf_VFLInference Service
main	4.11.1	Service Description
main	4.11.1.1	Overview
main	4.11.1.2	Service Architecture
main	4.11.1.3	Network Functions
main	4.11.1.3.1	Network Data Analytics Function (NWDAF)
main	4.11.1.3.2	NF Service Consumers
main	4.11.2	Service Operations
main	4.11.2.1	Introduction
main	4.11.2.2	Nnwdaf_VFLInference_Subscribe service operation
main	4.11.2.2.1	General
main	4.11.2.2.2	Subscription for VFL inference event notifications
main	4.11.2.2.3	Update subscription for event notifications

Document	Section	Title
main	4.11.2.2.4	Partial update subscription for event notifications
main	4.11.2.3	Nnwdaf_VFLInference_Unsubscribe service operation
main	4.11.2.3.1	General
main	4.11.2.3.2	Unsubscribe from VFL inference notifications
main	4.11.2.4	Nnwdaf_VFLInference_Notify service operation
main	4.11.2.4.1	General
main	4.11.2.4.2	Notification about subscribed event
main	5.10	Nnwdaf_VFLInference Service API
main	5.10.1	Introduction
main	5.10.2	Usage of HTTP
main	5.10.2.1	General
main	5.10.2.2	HTTP standard headers
main	5.10.2.2.1	General
main	5.10.2.2.2	Content type
main	5.10.2.3	HTTP custom headers
main	5.10.3	Resources
main	5.10.3.1	Resource Structure
main	5.10.3.2	Resource: NWDAF VFL Inference Subscriptions
main	5.10.3.2.1	Description
main	5.10.3.2.2	Resource definition
main	5.10.3.2.3	Resource Standard Methods
main	5.10.3.2.3.1	POST
main	5.10.3.2.4	Resource Custom Operations
main	5.10.3.3	Resource: Individual VFL Inference Subscription
main	5.10.3.3.1	Description
main	5.10.3.3.2	Resource definition
main	5.10.3.3.3	Resource Standard Methods
main	5.10.3.3.3.1	PUT
main	5.10.3.3.3.2	PATCH
main	5.10.3.3.3.3	DELETE
main	5.10.3.3.3.4	GET
main	5.10.3.3.4	Resource Custom Operations
main	5.10.4	Custom Operations without associated resources
main	5.10.5	Notifications
main	5.10.5.1	General
main	5.10.5.2	VFL Inference Event Notification

Document	Section	Title
main	5.10.5.2.1	Description
main	5.10.5.2.2	Operation Definition
main	5.10.6	Data Model
main	5.10.6.1	General
main	5.10.6.2	Structured data types
main	5.10.6.2.1	Introduction
main	5.10.6.2.2	Type VflInferSub
main	5.10.6.2.3	Type VflInferSubPatch
main	5.10.6.2.4	Type VflInferAnaSub
main	5.10.6.2.5	Type VflInferNotif
main	5.10.6.2.6	Type VflInferReq
main	5.10.6.2.7	Type VflInferResult
main	5.10.6.3	Simple data types and enumerations
main	5.10.6.3.1	Introduction
main	5.10.6.3.2	Simple data types
main	5.10.6.3.3	Enumeration: VflTermCause
main	5.10.7	Error handling
main	5.10.7.1	General
main	5.10.7.2	Protocol Errors
main	5.10.7.3	Application Errors
main	5.10.8	Feature negotiation
main	5.10.9	Security

The purpose of the protocol as the source step located it in the documents: a pre-trained VFL model with a fixed identifier; a trusted AF/NWDAF server and one NWDAF client; subscription, a single input/output notification, update and cancellation, with an explicit stage identifier; does not cover model training or the correctness of the numerical predictions.

The properties against the documents

Before writing a query, the analyst set the drafted list of properties against what the documents themselves say the protocol is for. The name, what a property is about and what the documents say are the analyst's reading; the places named do exist in the documents.

ID	Name	About	What the documents say	Where
A03	Genuine subscription and notification content	The NWDAF is assured that an accepted subscription comes from a real NF service consumer run with matching notifUri, notifCorrId, vflCorrId and anaEvent, and the consumer is assured that an accepted notification is a genuine NWDAF send for its notifCorrId.	Derived from the flow: the POST carries notifCorrId, notifUri and vflInferAnaSubs and the NWDAF creates and stores the subscription, and the notification carries notifCorrId and vflInferResults; the source states no security goal for it.	s0037, s0038, s0039, s0040, s0041, s0044, s0101, s0102, s0103, s0104

ID	Name	About	What the documents say	Where
A08	Correlation ids bound to own subscription	The NWDAF is assured that the notifCorrId, vflCorrId and anaEvent it accepts, and the subscriptionId it updates or deletes, belong to the requesting consumer's own subscription; the consumer is assured that an accepted notification carries its own notifCorrId.	notifCorrId is assigned by the NF service consumer and identifies the corresponding notification, vflCorrId identifies the VFL process, and the subscriptionId is the resource identifier of the update and delete; the source states no ownership rule.	s0040, s0054, s0085, s0102, s0266, s0272
Z01	Token authorizes every accepted operation	Every subscription creation, update, deletion and notification delivery the API accepts is backed by a valid OAuth2 access token with the nnwdaf-vflinference scope whose subject is the requesting NF.	Access to the API may be authorized with OAuth2 client credentials, the NF obtains a token from the NRF acting as authorization server, and the API defines a single scope for the whole service.	s0310, s0311, s0313
Z03	Operations limited to own subscription	A GET, PUT, PATCH or DELETE on {subscriptionId}, and the delivery of a notification to a notifUri, act only on the subscription of the token subject.	The API defines a single service-wide scope and no scope at resource or operation level, and the source states no check that the token subject owns the addressed subscription.	s0313, s0045, s0172
Z04	Nothing after unsubscribe or expiry	After a successful DELETE or a VflTermCause termination no further notification is produced or accepted for that subscription, and an expired token authorizes nothing.	The DELETE removes the corresponding subscription and answers 204, and vflTermCause means the VFL client sends no further notifications for the subscription.	s0086, s0087, s0088, s0281, s0291
S05	Removed subscription correlation unusable	An old subscriptionId or notifCorrId of a deleted or terminated subscription cannot trigger or be used to accept a new notification.	The source describes removal and termination but states nothing about notifications already in flight or about reuse of the identifiers.	s0086, s0087, s0088, s0291
R02	One acceptance per notification	The same logical notification is accepted by the consumer at most once, and a replayed subscription request does not create a second independent subscription.	The notification is acknowledged with 204 No Content and carries only the notifCorrId as a correlation value; the source states no replay rule.	s0101, s0102, s0103, s0104
K02	Inference results kept from others	The vflInferResults of a subscription and the UE identifiers it concerns are known only to the subscribing consumer and the NWDAF, not to other legitimate NFs.	The notification carries the intermediate VFL inference results for the subscribed analytics event; the source states no confidentiality goal for them beyond the OAuth2 protection of the API.	s0103, s0275, s0281, s0272
K01	Access tokens not derivable	An attacker cannot derive a valid nnwdaf-vflinference access token issued to another NF.	The consumer obtains a token from the NRF acting as authorization server; OAuth2 requires tokens to be unguessable and confidential.	s0310, s0311
U01	Accepted notification matches own subscription	A consumer accepts a notification only when it corresponds to its own subscription with the same notifCorrId, vflCorrId and anaEvent.	The notifCorrId is assigned by the consumer for the requested notifications and repeated in the notification, which is what lets the consumer correlate it.	s0040, s0101, s0102, s0103, s0275
U02	Results come from inference only	The results and correlation identifiers in a notification come from the VFL inference of the subscribed analytics event and not from a training step.	The service is the VFL client inference API of an already trained, fixed model, and the notification carries the intermediate VFL inference results.	s0004, s0103, s0275, s0281
U03	Subscription ownership and staleness	A subscriptionId is read and written only by the consumer that created it, and results of a removed subscription are not accepted later.	PUT, GET, PATCH and DELETE address the individual subscription by subscriptionId; the source states no ownership check and no rule for results of a removed subscription.	s0085, s0086, s0087, s0088, s0172, s0214, s0281

ID	Name	About	What the documents say	Where
X01	No redirection of a notification to another consumer	The NWDAF and the consuming NF: the inference results must not follow a Location header to a foreign endpoint.	The consumer may answer a notification with a 307 or 308 redirect whose Location names an alternative NF consumer instance, and an SCP may provide it.	s0105, s0247, s0249, s0250

Added from the documents by the analyst: X01.

The security statements of the documents, paragraph by paragraph

The host listed the 398 paragraphs of the sections in which the documents say what the protocol protects and against what, and the analyst said of each what it made of it: a property, an attacker, or why neither. Taken up: 36. Not taken: 362. Which paragraphs are listed follows from the layout of the standard; where each went is the analyst's judgement.

Paragraphs	Where	Taken up as	Why not
P001 to P002	rfc6749:2778-2786	No goal stated	Overview of the OAuth2 security considerations and a pointer to the threat model document.
P003 to P006	rfc6749:2790-2823	Z01, K01, `malicious_nf`	
P007 to P010	rfc6749:2827-2850	Z01, Z03, K01, `malicious_nf`	
P011 to P014	rfc6749:2859-2877	K01, `malicious_nf`	
P015	rfc6749:2879-2881	Z03, A08, `malicious_nf`	
P016 to P021	rfc6749:2885-2917	Outside the scope	Refresh tokens; the API uses the client credentials grant only.
P022 to P025	rfc6749:2921-2943	Outside the scope	Authorization codes are not used by the client credentials grant.
P026 to P029	rfc6749:2947-2980	Outside the scope	Redirection URI of the authorization endpoint; no end user is involved here.
P030 to P033	rfc6749:2984-3003	Outside the scope	The resource owner password credentials grant is not used.
P034 to P035	rfc6749:3013-3019	K01, K02, `network`	
P036	rfc6749:3023-3028	A03, A08, `network`	
P037 to P039	rfc6749:3032-3041	K01, Z01, `malicious_nf`	
P040 to P042	rfc6749:3045-3066	Outside the scope	Phishing of end users; this machine-to-machine API has no resource owner.
P043 to P048	rfc6749:3070-3110	Outside the scope	Browser based cross-site request forgery; no user agent takes part.
P049 to P050	rfc6749:3119-3138	Outside the scope	Clickjacking of user-facing pages.
P051 to P052	rfc6749:3142-3150	A03, U01, `network`, `malicious_nf`	

Paragraphs	Where	Taken up as	Why not
P053 to P054	rfc6749:3154-3167	Outside the scope	Open redirectors of the authorization endpoint.
P055 to P062	rfc6749:3170-3210	Outside the scope	Misuse of a token in the implicit grant; the implicit grant is not used.
P063	ts-33-501:2491-2491	The prover cannot be asked	Bidding down of a security feature is a property of feature negotiation and implementation strength, not of traces of this message flow.
P064 to P072	ts-33-501:2495-2511	Outside the scope	Authentication and authorization of the UE and of the serving network; this API has neither UE nor access network.
P073 to P074	ts-33-501:2515-2517	Outside the scope	Key lengths for AS and NAS protection.
P075 to P089	ts-33-501:2523-2553	Outside the scope	UE requirements on ciphering and integrity protection over the air interface.
P090 to P103	ts-33-501:2557-2583	Outside the scope	Integrity protection requirements between UE and gNB.
P104 to P110	ts-33-501:2587-2599	Outside the scope	Storage of subscription credentials in the UE and the UICC.
P111 to P125	ts-33-501:2603-2631	Outside the scope	Subscriber privacy over the radio interface.
P126	ts-33-501:2637-2637	Outside the scope	General statement about ng-eNB requirements.
P127 to P136	ts-33-501:2641-2659	Outside the scope	Confidentiality of user and signalling data on the access network.
P137 to P147	ts-33-501:2663-2683	Outside the scope	Integrity of user and signalling data on the access network.
P148 to P168	ts-33-501:2687-2735	Outside the scope	Requirements on the gNB, its secure environment and its key management.
P169 to P178	ts-33-501:2739-2761	Outside the scope	Requirements on the F1, E1 and ng-eNB interfaces.
P179 to P198	ts-33-501:2767-2811	Outside the scope	Signalling protection and subscriber privacy for the AMF, SEAF and access stratum.
P199 to P201	ts-33-501:2819-2823	Z01, `network`	
P202 to P213	ts-33-501:2827-2853	Outside the scope	Requirements on AUSF, UDM and UDR; they take no part in this API.
P214	ts-33-501:2859-2859	Outside the scope	Trust zones between operators and protection of messages crossing trust boundaries; this analysis stays inside one operator domain.
P215	ts-33-501:2865-2865	R02, `network`	
P216	ts-33-501:2867-2867	No goal stated	NRF authorizes discovery and registration; that is NRF behaviour, not a goal of this API's message flow.
P217 to P218	ts-33-501:2869-2871	A03, Z01, `network`, `malicious_nf`	
P219 to P220	ts-33-501:2873-2875	U01, A08, `network`, `malicious_nf`	
P221 to P227	ts-33-501:2879-2891	Z01, K01, `malicious_nf`	

Paragraphs	Where	Taken up as	Why not
P228 to P234	ts-33-501:2895-2907	Outside the scope	NEF and Application Function exposure requirements; this API is consumed by NWDAF, AF and NEF as NFs, and the NEF-af interface is not part of the flow.
P235 to P241	ts-33-501:2911-2923	Outside the scope	Requirements on the SCP, which is not modelled; it is a routing intermediary of the SBI.
P242 to P254	ts-33-501:2929-2953	Outside the scope	End-to-end core network interconnection for roaming; this flow stays inside one PLMN.
P255 to P303	ts-33-501:2957-3055	Outside the scope	SEPP and roaming intermediary requirements.
P304 to P312	ts-33-501:3059-3075	Outside the scope	Protection of 3gpp-Sbi message attributes between networks.
P313 to P314	ts-33-501:3079-3081	Outside the scope	IPUPS functionality.
P315 to P318	ts-33-501:3085-3091	Outside the scope	NF requirements for end-to-end interconnection security between networks.
P319 to P327	ts-33-501:3097-3115	Outside the scope	Signalling monitoring configuration and streaming.
P328 to P334	ts-33-501:3121-3133	Outside the scope	Security visibility for operators, not a property of this API.
P335 to P337	ts-33-501:3137-3141	The prover cannot be asked	Security configurability is an operational property, not a trace property.
P338 to P369	ts-33-501:3149-3215	The prover cannot be asked	Algorithm identifiers, algorithm selection and negotiation: the strength of a primitive is outside symbolic analysis.
P370 to P375	ts-33-501:3219-3231	Outside the scope	Requirements on the 5G-RG and NSSAAF, which take no part in this API.
P376 to P380	ts-33-501:6540-6548	Outside the scope	Security on the N3 interface.
P381 to P385	ts-33-501:9656-9666	Outside the scope	EAP-TLS authentication procedures for the UE.
P386 to P394	ts-33-501:10893-10917	Outside the scope	IAB node and donor requirements.
P395 to P396	ts-33-501:11439-11443	Outside the scope	MBSF and MBSTF requirements.
P397 to P398	ts-33-501:11899-11901	Outside the scope	NEF security requirements.

The parties and the protected values, one by one

The host listed the 17 parties and protected values that the drafted properties name, and the analyst said of each which attacker controls the party or holds the value, or why no such attacker was set. With an attacker: 9. Without: 8. The list follows from the properties; what was made of each entry is the analyst's judgement.

ID	Party or value	Kind	Attacker	Why none
S01	NF service consumer	party	`malicious_nf`	

ID	Party or value	Kind	Attacker	Why none
S02	NWDAF	party	Everything lost by definition	The NWDAF is the trusted VFL client and the producer of the results; a dishonest NWDAF can fabricate any result, and no target asks a property of the protocol under a dishonest producer.
S03	NRF	party	Everything lost by definition	The NRF is the authorization server and the trust anchor for every token; with it the whole authorization baseline is gone and no target is worded for that case.
S04	subscription content	value	`malicious_nf`	
S05	notification content	value	`malicious_nf`	
S06	subscriptionId	value	`malicious_nf`	
S07	notifCorrId	value	`malicious_nf`	
S08	vflCorrId	value	`malicious_nf`	
S09	anaEvent	value	Nothing gained	The analytics event is a public enumeration value of the API.
S10	notifUri	value	Nothing gained	A notification endpoint is a service address that any NF in the domain can address.
S11	OAuth2 access token	value	`malicious_nf`	
S12	nnwdaf-vflinference scope	value	Nothing gained	The scope name is public and covers the whole service, so holding it separates no consumers.
S13	token subject	value	Nothing gained	The token subject is the NF's own public identity.
S14	VflTermCause	value	Nothing gained	The termination cause is a public enumeration value.
S15	token validity period	value	Outside the scope	Token lifetime handling is in the NRF and the token profile of the baseline, which the model does not represent.
S16	vflInferResults	value	`malicious_nf`	
S17	gpsis/supis/intGroupIds	value	`malicious_nf`	

The properties under each attacker

The attackers are the analyst's, set before a query was written; each answer was recorded under one of them. A cell is the prover's answer for the property under that attacker; "No answer" is an attacker set for the property under which the prover left a claim undecided or nothing was recorded; an empty cell is an attacker the analyst did not set for the property.

Attacker	Has	Basis	Where
`network`	the network only: it reads, drops, changes and sends messages, and holds no secret of a party		
`malicious_nf`	a legitimate NF's own OAuth2 access token with the nnwdaf-vflinference scope and its own TLS credentials, the ability to address other NFs' notification endpoints (they are service addresses), and a notifCorrId or subscriptionId value of another consumer, neither of which the source requires to be secret or unguessable	The API defines one service-wide scope and no resource-level scope, so a token does not by itself separate consumers; OAuth2 states that a malicious client can impersonate another one and that the resource server has no method to check that a token was issued to the presenting client; the source states no ownership check on {subscriptionId} and no rule making notifCorrId or subscriptionId confidential.	rfc6749:2827, rfc6749:2879, main:313, s0313, s0045, s0040

ID	network	malicious_nf
A03	Holds	Fails
A08	No answer	Fails
Z01	Holds	No answer
Z03	No answer	Fails
Z04	No answer	Fails
S05	No answer	No answer
R02	No answer	No answer
K02	Holds	Fails
K01	Holds	Holds
U01	No answer	Fails
U02	No answer	No answer
U03	No answer	Fails
X01		No answer

Source and targets

Source: 313 units, 64041 bytes, frozen in source-delivery-01.json. Targets: 12 approved (A03, A08, Z01, Z03, Z04, S05, R02, K02, K01, U01, U02, U03); questions asked: 3. The list of targets was approved as drafted, in advance, with --approve-targets.

Decisions the host took by its documented default (task `decisions: auto`), for you to check:

- approve_package: approved source package protocol-acf2a3558d66d3a6b61883de8847

How this step reports itself: 1 request (1 failed), 0.000 USD, 46105 tokens in and 5233 out, 266 s.

Notes on the prover and the model

The targets the analyst named:

step	targets	features	reason
29	A03, A08, Z01, Z03, Z04, S05, R02, K02, K01, U01, U02, U03	plain_primitives, many_sessions, other	TLS is modelled as a private channel with per-endpoint encryption under a private credential function, and the token as an unforgeable private term, so only hashing/encryption style primitives are involved. The model has one honest consumer session, one NWDAF service instance per subscription and one malicious NF, so the properties concern a single run of each party rather than any number of sessions. The token's scope, subject and expiry are abstracted, and subscriptionId and notifCorrId are treated as values another NF may know; both matter to the failing targets.

The analyst's closing note

The Nnwdaf_VFLInference flow is modelled and answered. Findings: (1) a legitimate NF that holds a token with the single scope nnwdaf-vflinference and knows another consumer's subscriptionId takes over that subscription with a PUT naming its own notifUri, receives its vflInferResults, and can delete it (recorded attacks a08_put, z04_removed, k02_attack): the source defines one service-wide scope, no resource-level scope and no ownership check on {subscriptionId}. (2) A consumer cannot tell which subscription a notification belongs to, because VflInferNotif

carries only the consumer-chosen notifCorrId, so results of another subscription are accepted as its own (attack a08_own). Proved, each with a control: only requests backed by a token of the requesting NF are accepted; no other NF's token is derivable; the attacker with the network alone learns no inference result. The reachability answers cover the unsubscribe race and a repeated acceptance; U01's last claim is undecided (derivation without a trace, a case for Tamarin+ Agent). Limits that matter: TLS is a private channel, the model has one honest consumer session, subscriptionId and notifCorrId are treated as knowable (the two findings rest on that), the token lifetime and the redirect path are absent, so X01 has no answer. Several rows have no answer for the attacker they were set for and are left open; the ownership fix was not checked with check_fix within the budget.