Security analysis of NWDAF VFL client training 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	17: 6 hold, 1 fail (3 attacks), 10 without an answer
Findings	1

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_VFLTraining service lets an NF service consumer (an AF, NEF or NWDAF) use an NWDAF as a VFL client. The consumer holds an OAuth2 access token the NRF issued it for the single scope nnwdaf-vfltraining, the NWDAF holds the training state and assigns subscription identifiers.

The consumer creates a subscription as a capability check with vflPreFlag true, updates it to a training subscription with vflPreFlag false, receives training reports on the callback URI it gave, and cancels the subscription with a DELETE. Both endpoints are NFs of the same network and reach each other over mutually authenticated TLS on the service based interface.

The parties:

- **NF service consumer (VFL server)**: AF / NEF / NWDAF that runs the VFL training. Holds: client credentials; notifUri; notifCorrId; vflCorrId.
- **NRF**: authorization server, client credentials grant. Holds: NF profiles; local authorization policy.
- **NWDAF (VFL client)**: NF service producer, keeps subscriptions and training state. Holds: its NF identity; subscription store.

Figure 1 shows the exchange as the documents specify it.

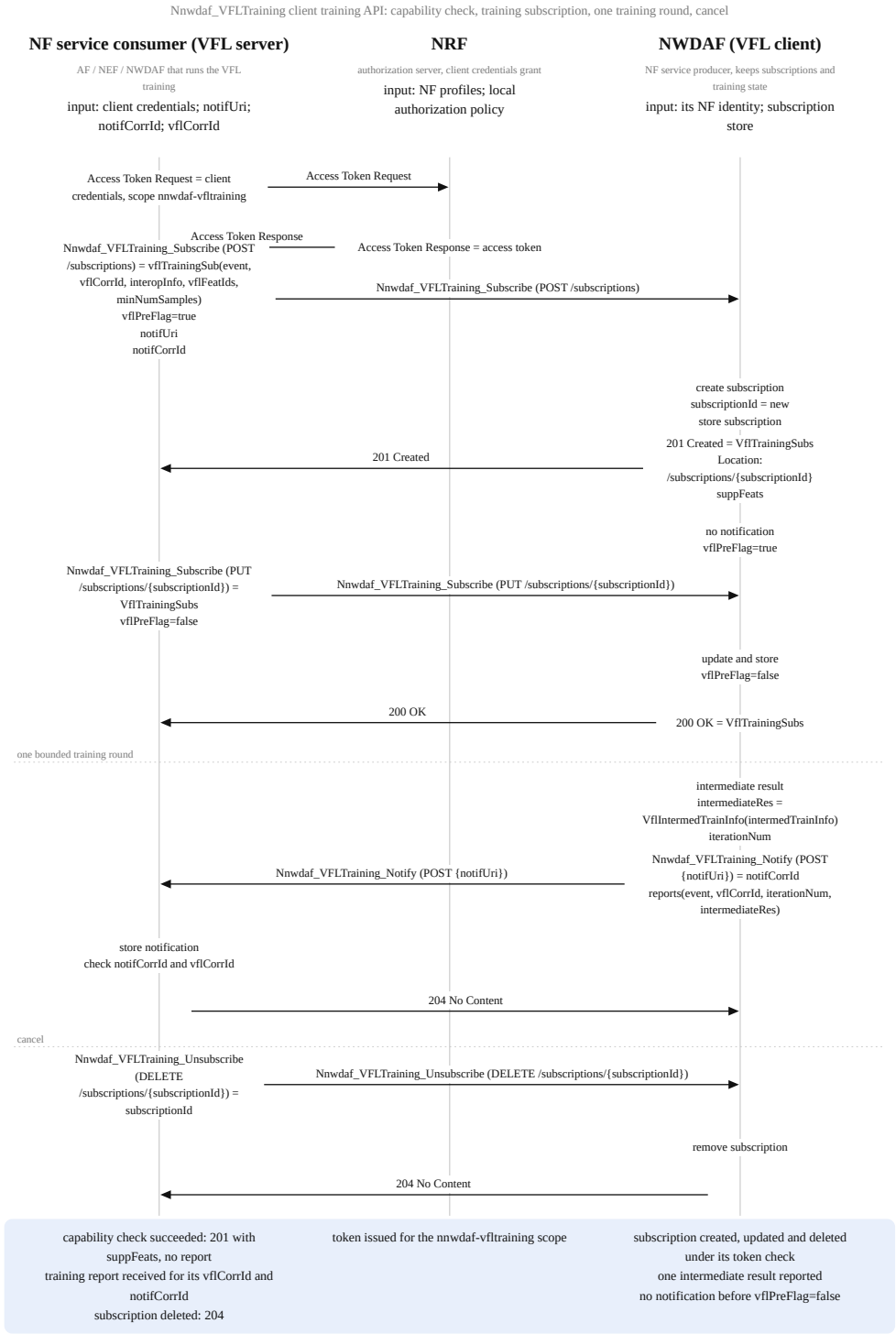


Figure 1. The protocol as specified.

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 1 finding: what follows is not stated in the documents. For 1 of them it is unclear whether the documents state it.

Authorization is sound in the model: every creation, modification and deletion the NWDAF accepts is backed by an NRF-issued token for the nnwdaf-vfltraining scope presented by the NF it names, and a token that leaks to another NF cannot be used to act as its owner. One finding stands: the API defines no resource-level scope and no operation clause requires an ownership check, so any NF with a service-scope token can modify, patch or delete another consumer's subscription once it knows the subscription URI. Notification correlation, the preparation flag and the capability check hold, but only because the consumer checks the correlation values it chose itself. Three targets about freshness and about the sender of a notification are undecided: ProVerif's replicated request handler lets one creation request build two subscriptions. Of the 17 targets, 10 are proved or reached, 4 carry an attack and 3 are undecided.

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 An NF with the service scope can hijack or cancel another consumer's VFL training subscription

Kind	Weakness of the protocol as specified
New	Unclear: the documents state it in part, or could not be searched (ts-29-500:4996, ts-29-500:5000)
Value	high
Properties	Z03 (section 3)
Source text	s0048, s0049, s0072, s0073, s0293
Shown by	ProVerif attack queries `z03_update`, `z03_delete`, `z03_patch`; Figure 2

What happens. The API defines a single OAuth2 scope for the whole service and no resource-level scope. The NWDAF validates the token, its audience and its scope, then looks a subscription up by its subscriptionId alone and applies the PUT, PATCH or DELETE without asking whether the requesting NF created it. Any NF that holds a service-scope token and knows the URI of another consumer's subscription therefore replaces its content, stops its training by setting vflPreFlag, or deletes it, and the NWDAF answers 200, 204 or 204 as it would to the owner.

What the attacker needs. An NF instance identity of the own network, an access token the NRF issued for nnwdaf-vfltraining, and the subscriptionId of the victim's subscription, which the documents do not declare secret and which is carried in a Location header and in the URI of every later request.

What the attacker gains. The attacker cancels the VFL training run of another AF or NWDAF, or redirects the subscription to its own notifUri and correlation values so that the consumer no longer receives the reports it subscribed to.

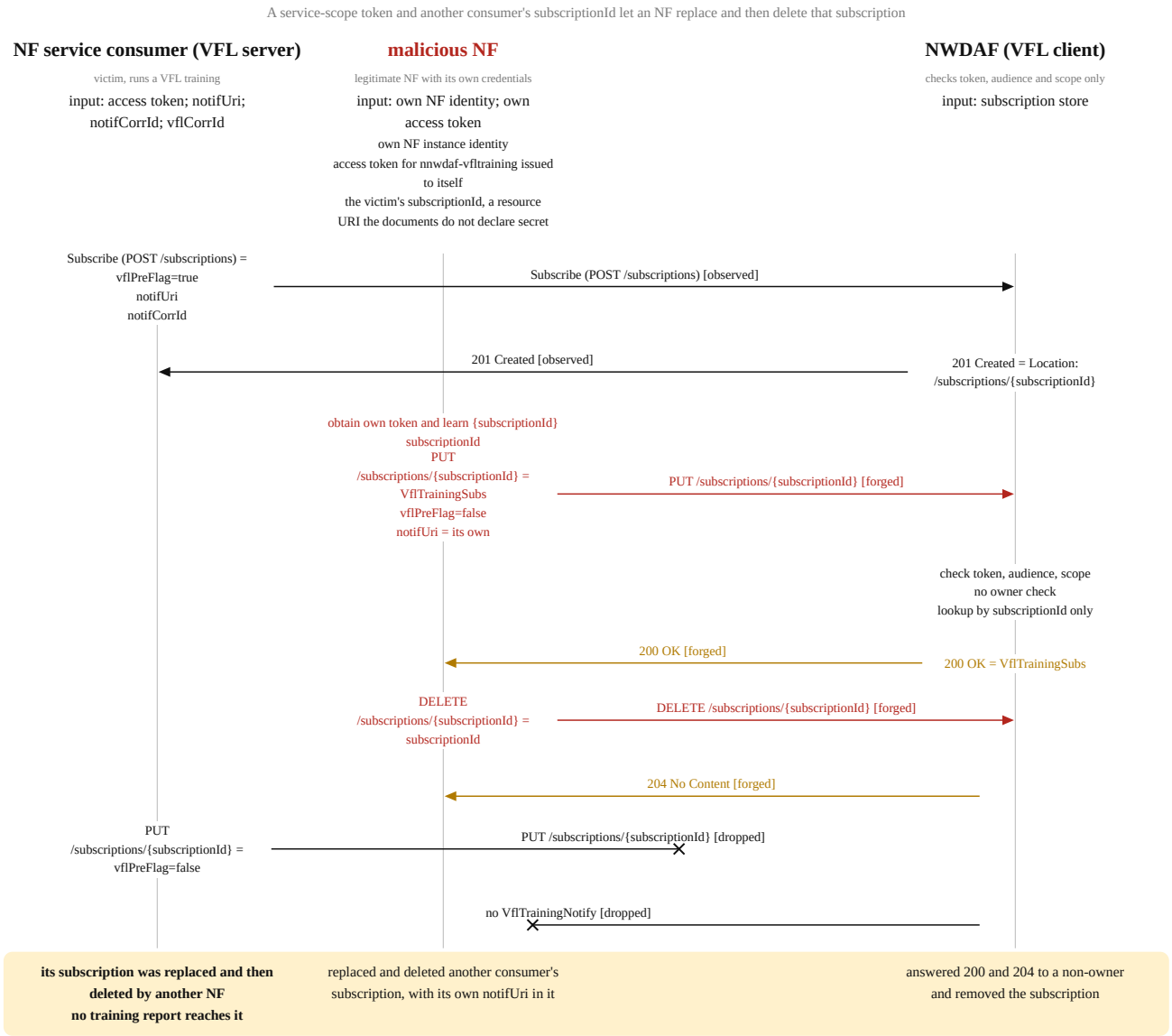


Figure 2. A service-scope token and another consumer's subscriptionId let an NF replace and then delete that subscription. Property Z03; attack queries z03_delete, z03_patch, z03_update.

Proposed change. The NWDAF must refuse a PUT, PATCH or DELETE on an individual subscription whose recorded owner is not the subject of the presented access token, answering 403 Forbidden with no side effect; state it in 4.10.2.2.3, 4.10.2.2.4 and 4.10.2.3.2, and define a resource-level scope for the individual subscription as TS 29.500 clause 6.7.3 provides for. Checked on a copy of the model in which only that check is added: the three attack queries become verified, the normal run is still found and the other recorded answers are unchanged.

Check of the change with ProVerif. The change was put into a copy of the model: The NWDAF refuses a PUT, PATCH or DELETE on an individual VFL training subscription whose recorded owner is not the subject of the presented access token: the request is answered with 403 Forbidden and no side effect is applied. On the changed model ProVerif proves `z03_update`, `z03_delete`, `z03_patch`: the attack is no longer possible there. The normal run of the protocol is still found. Of the other recorded answers, 16 still hold and 7 changed: `z01_notify` (Z01, was undecided) is now inconclusive; `x01_notify` (X01, was undecided) is now inconclusive; `x01_notify` (A03, was undecided) is now inconclusive; `a04_create` (A04, was undecided) is now inconclusive; `r02_report` (R02, was undecided) is now inconclusive; `a04_create` (R03, was undecided) is now inconclusive; `r02_report` (U02, was undecided) is now inconclusive. This is a first check on the model: it shows that the change removes this attack there, not that it introduces no other problem. Changed model: `records/fixes/0050-fix-owner.pv`.

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
Z01	**each operation covered by a valid token.** Every time a subscription creation/modification/deletion or a notification delivery is accepted by the NWDAF or the consumer, a valid, unexpired authorization issued by the NRF as the authorization server, covering the nnwdaf-vfltraining scope, must exist.	no compromise	No answer	medium	unclear
Z03	**subscription bound to its authorized subject.** Modification/deletion of a given subscriptionId, and notification for a given notifCorrId, may only be performed by the authorized subject of that subscription, within the context covered by its token; the token, the subscriptionId and the notifUri/notifCorrId must not be transplantable to another subject or another subscription.	no compromise	Fails (3 attacks)	high	unclear
A03	**consumer and NWDAF act on real runs.** When the NWDAF accepts a subscription creation, a real run initiated by the consumer with the same identity and the same VflTrainingSubs content exists; when the consumer accepts a notification, a real notification run by the NWDAF for that subscription, with the same notifCorrId and reports, exists.	no compromise	No answer	medium	yes
A04	**fresh identifier per real run.** Every successful subscription creation by the NWDAF (a new subscriptionId) corresponds to a distinct real subscription run by the consumer; every notification acceptance by the consumer corresponds to a distinct real notification run by the NWDAF.	no compromise	No answer	low	unclear
R02	**one acceptance per notification instance.** A notification instance with the same notifCorrId is accepted by the consumer at most once; a subscription creation request produces at most one subscription resource.	no compromise	No answer	low	unclear
R03	**one side effect per logical operation.** The same logical subscription creation, subscription deletion or notification storage operation commits at most one security-relevant side effect between the consumer and the NWDAF.	no compromise	No answer	low	unclear
S05	**deleted subscription stays deleted.** After a successful DELETE (204), that subscriptionId accepts no further modification and produces no further notifications; the old subscription context cannot re-trigger these operations.	no compromise	Reachable only	low	yes
S04	**preparation, rounds and report fields consistent.** A preparation-phase subscription (vflPreFlag=true) produces no notifications until it is updated to false; checkpoint recovery uses the iteration number it references; in a notification, termCause, intermediateRes and vflNotOnTimeInd appear at most one at a time and are consistent with the preceding subscription context.	no compromise	Holds	medium	yes

ID	Property	Compromise assumed	Result	Value	In the documents
A08	**notification values match the subscription.** In a notification the consumer accepts, event, vflCorrId, iterationNum and notifCorrId must correspond to the same set of values in the subscription it created or modified; the subscription parameters the NWDAF accepts must correspond to the values the consumer actually proposed.	no compromise	No answer	medium	no
K02	**training data stays with the two parties.** Intermediate training information, intermediate results and sample identifiers (SUPI/GPSI) are visible only to the subscribing consumer and the NWDAF; intermediate forwarders such as the SCP/SEPP cannot read their content.	no compromise	Reachable only	none	unclear
K01	**access token kept from the attacker.** An OAuth2 access token is known only to the NRF that issued it and the NF service consumer that obtained it; an attacker cannot derive a valid token usable to access the Nnwdaf_VFLTraining API.	no compromise	Holds	low	yes
U01	**notification correlation comes from the consumer.** In a notification the consumer accepts, vflCorrId and iterationNum must come from a subscription the consumer itself submitted.	no compromise	Holds	medium	no
U02	**training report accepted once.** The same training report (notifCorrId, vflCorrId, iterationNum) is accepted by the consumer at most once, not across sessions and not across rounds.	no compromise	No answer	low	unclear
Q3	**capability check differs from training completion.** The question concerns the semantic distinction between the capability check and training completion, and finite-state consistency after cancellation.	no compromise	Holds	medium	yes
X01	**mutual authentication of the SBI exchange.** the NWDAF acts on a subscription request only from a peer whose NF identity it has authenticated, and the NF service consumer accepts a notification only from that peer; neither party acts on a message of a third NF that holds its own credentials	an NF instance identity of its own in the same network and an access token the NRF issued to it for the nnwdaf-vfltraining scope, obtained before the run	No answer	medium	yes
X02	**invalid requests refused.** the NWDAF refuses a create, modify or delete request that names no existing subscriptionId, or that carries no token, an expired token, or a token for another audience or scope, and answers with an error instead of acting	an NF instance identity of its own in the same network and an access token the NRF issued to it for the nnwdaf-vfltraining scope, obtained before the run	Holds	medium	yes
X03	**access token not usable by another NF.** an access token that leaked to another NF cannot be used to act in the name of the NF it was issued to, because the producer requires the token subject to be the identity it authenticated on the connection	an access token the NRF issued to the legitimate NF service consumer for the nnwdaf-vfltraining scope, but not the consumer's TLS credentials	Holds	medium	yes

Goals the documents state that the drafted list of properties did not have, added by the analyst (Appendix D): X01, X02, X03.

The questions asked, and the analyst's answers:

Question	Answer	Rests on
Q1: the VFL correlation ID/iteration ID is bound to the participant identity	Yes in the modelled run: the subscription is bound to the vflCorrId and notifCorrId the consumer generated, the NWDAF copies both into every report, and the consumer accepts a report only when both match. The binding rests on those values alone, not on the identity of the sender, because the notify operation is not token-authorized.	A08, U01, Z03
Q2: a training report is not accepted across sessions or across rounds	It cannot be settled on this model: a fresh notifCorrId and a fresh iteration number per report make a repeat impossible in the plain run, but ProVerif's replicated sender lets one creation request build two subscriptions, so the injective query is inconclusive and neither an attack nor a proof is claimed.	U02, R02, A04
Q3: a successful capability check is not the same as training completion, and finite-state consistency after cancellation	The capability check is a preparation subscription and produces no report until the subscription is updated to vflPreFlag false, so a successful check is not a training completion. After a delete the NWDAF refuses to notify the deleted subscription, but the documents do not say whether a report in flight is delivered, and another NF can force that delete, which is the finding.	Q3, S04, S05, Z03

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 is modelled as a private channel between the endpoints, so a relay that terminates a connection and could read or re-originate a message is not modelled; TS 33.501 5.9.2.4 states all parties must trust the SCP.
- The subscription store is monotone: a delete is recorded as a marker and removes nothing, so the count of side effects of one request and the fate of a report in flight around a delete cannot be settled.
- The notifUri is a private channel per subscription, so a third party posting a forged notify request to the consumer's callback URI is not modelled; the consumer's only protection there is the secrecy and check of notifCorrId and vflCorrId, which the specification does not require to be unguessable.
- Token expiry is not modelled, so the unexpired part of the authorization target is not tested.
- The training itself and the opaque fields intermedTrainInfo, interopInfo, vflFeatIds and accuMonInfo are not analysed; convergence, poisoning robustness and differential privacy are outside a symbolic model.
- Subscriber privacy of the SUPI and GPSI sample identifiers is outside the scope of the task.
- The model is symbolic: it says nothing about the computational strength of TLS, of the token format or of the HTTP/2 transport.

What the host records:

- No answer for: Z01, A03, A04, R02, R03, S05, A08, K02, U02, X01.
- The answers changed after the reading was written for: A08, X01.
- The prover gave no answer (Appendix C): Z01: z01_notify, A03: x01_notify, A04: a04_create, R02: r02_report, R03: a04_create, A08: a08_notify, U02: r02_report, X01: x01_notify. 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): S05 under `network`; S05 under `nf_scope_resource_uri`; K02 under `network`; K02 under `nf_scope_resource_uri`.
- Proved without a control query (Appendix C): Z01: z01_create, Z01: z01_update, Z01: z01_delete, Z03: x02_foreign, A03: z01_create, S04: s04_preflag, A08: z03_notify, K01: k01, U01: z03_notify, Q3: s04_preflag,

X01: z01_create, X01: x01_request_c, X02: x02_foreign, X03: x02_foreign. A control is a variant of the property that must fail; without one, nothing shows that the property as written can fail in this model.

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.

Z01: each operation covered by a valid token

Every time a subscription creation/modification/deletion or a notification delivery is accepted by the NWDAF or the consumer, a valid, unexpired authorization issued by the NRF as the authorization server, covering the nnwdaf-vfltraining scope, must exist.

The NWDAF acts only on requests whose token it can validate for its own NF type and the nnwdaf-vfltraining scope and whose subject is the NF it authenticated on the connection, so every accepted creation, modification and deletion is backed by a token the NRF issued. The notification delivery is the operation the baseline deliberately leaves without a token, so that half of the claim fails by design and the prover could not decide it.

- Result: No answer; answers what is asked: partly; usable with caveats
- Value medium: The proof shows the producer applies the checks the baseline requires, which is no new result, and the notification half is a documented design choice rather than a weakness.
- Stated in the documents: unclear
- Rests on: TS 29.500 clause 6.7.3 is the only authorization rule modelled; the token is opaque and only the NRF can mint it, so its format, claims and expiry are not analysed.; The notify operation is modelled without a token, as TS 29.500 clause 6.7.3 states.
- Open: whether a predictable notifCorrId or a consumer that does not check it lets a third party inject a report
- Next step: Model the callback as a public endpoint with per-connection encryption and a guessable notifCorrId.

Z03: subscription bound to its authorized subject

Modification or deletion of a given subscriptionId, and notification for a given notifCorrId, can only be performed by the authorized subject of that subscription, within the context covered by its token; the token, the subscriptionId and the notifUri/notifCorrId cannot be transplanted to another subject or another subscription.

The subject binding of a token is enforced: a token issued to one NF cannot be used to act as another, because the producer requires the token subject to equal the identity it authenticated. Ownership of the resource is not enforced: an NF with its own valid token can replace, patch or delete any subscription whose URI it knows, and the NWDAF answers as it would to the owner.

- Result: Fails (3 attacks); answers what is asked: yes; usable
- Value high: The documents state the scope model that makes the attack possible but never state the failure.
- Stated in the documents: unclear
- Rests on: The attacker is assumed to know the victim's subscriptionId; the documents neither declare it secret nor require it to be unguessable.; The producer is modelled exactly as the operation clauses describe it, with a token, audience and scope check and a lookup by subscriptionId, and without an ownership check because none is stated.; TLS gives each NF its own authenticated connection, and the identity check is modelled by the connection a request arrives on.
- Open: whether an operator is expected to configure resource-level scopes for this API
- Next step: Check a change that adds a resource-level scope and an owner check on the individual subscription.

A03: consumer and NWDAF act on real runs

When the NWDAF accepts a subscription creation, there is a real run initiated by the consumer with the same identity and the same VflTrainingSubs content; when the consumer accepts a notification, there is a real notification run of the NWDAF for that subscription, with the same notifCorrId and reports.

The creation half holds: the NWDAF records as owner only the identity whose token it validated, so no third NF can have a subscription created under the consumer's identity. The notification half could not be decided: the prover neither proves nor falsifies that an accepted report was sent by the NWDAF for that subscription.

- Result: No answer; answers what is asked: partly; usable with caveats
- Value medium: TS 33.501 clause 5.9.2.1 requires mutual authentication between consumer and producer, which is what the creation half shows.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2871
- Rests on: Mutual authentication is taken from TLS and modelled as a connection carrying the peer's identity.; The agreement on the sender of a notification is not decided.
- Open: the agreement on the sender of a notification
- Next step: Bind each notification to a session value the replicated sender cannot share and ask the injective form.

A04: fresh identifier per real run

Every successful subscription creation by the NWDAF (a new subscriptionId) corresponds to a distinct real subscription run of the consumer; every notification acceptance by the consumer corresponds to a distinct real notification run of the NWDAF.

The injective correspondence could not be decided: the prover builds one subscription from a single creation request through two copies of the replicated handler, and that over-approximation blocks the query.

- Result: No answer; answers what is asked: no; not usable
- Value low: The undecided answer comes from the replication abstraction, not from the protocol.
- Stated in the documents: unclear
- Rests on: The subscriptionId comes from a fresh per-request value, but the replicated handler still lets one request produce two subscriptions in the abstraction.
- Open: whether a retried POST creates two subscriptions, which the document does not discuss
- Next step: Model the creation with an idempotency key and ask the injective query again.

R02: one acceptance per notification instance

A notification instance with the same notifCorrId is accepted by the consumer at most once; the same subscription creation request produces at most one subscription resource.

The uniqueness of a notification instance could not be decided for the same abstraction reason: the replicated sender and the duplicated subscription let the prover reach one report twice.

- Result: No answer; answers what is asked: no; not usable
- Value low: The answer is about the model's multiplicity, not about the protocol.
- Stated in the documents: unclear
- Rests on: No retry or deduplication rule is modelled, because the document defines none.
- Open: whether the NWDAF retries a notification whose 204 was lost
- Next step: Unroll the notification exchange to one report per round and ask the injective query there.

R03: one side effect per logical operation

The same logical subscription creation, subscription deletion or notification storage operation commits a security-relevant side effect between the consumer and the NWDAF at most once.

The count of security-relevant side effects per logical operation could not be decided; the model lets one creation request be consumed twice by the replicated handler.

- Result: No answer; answers what is asked: no; not usable
- Value low: The answer says nothing dependable about the protocol.
- Stated in the documents: unclear
- Rests on: The store is monotone and the request handlers are replicated, so multiplicities are over-approximated.
- Open: retry semantics of the POST and of the DELETE
- Next step: Model one logical operation per fresh session value and count its side effects.

S05: deleted subscription stays deleted

After a successful DELETE (204), that subscriptionId no longer accepts modifications and no longer produces notifications; the old subscription context cannot trigger these operations again.

The situation is reached: after a subscription is deleted the NWDAF is asked to notify it and refuses, because it records the deleted subscriptionId and checks it before every report. Nothing is shown about a report already in flight when the delete arrives, and the finding shows another NF can bring about that delete.

- Result: Reachable only; answers what is asked: partly; usable with caveats
- Value low: TS 29.520 clause 4.10.2.3.2 states the subscription is removed when the DELETE is accepted, which is what the reached situation checks.
- Stated in the documents: yes. Where the documents say it: main:32691
- Rests on: The deletion marker is monotone: the model records the delete but cannot remove entries.; In-flight reports are not modelled as lost, so their delivery after a delete is not settled.
- Open: whether a report in flight survives the delete
- Next step: Model the delete as removing the entry, add a report in flight, and ask whether the consumer accepts it.

S04: preparation, rounds and report fields consistent

A preparation subscription (vflPreFlag=true) produces no notification until it is updated to false; checkpoint recovery uses the iteration number it references; in a notification, termCause, intermediateRes and vflNotOnTimeInd appear at most one at a time, and are consistent with the preceding subscription context.

A preparation subscription never produces a report before it is updated to vflPreFlag false, and a subscription created directly as a training subscription notifies immediately. The exclusivity of the three report fields is a static well-formedness rule that no trace can violate in this model.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The NOTE in the VflTrainingSubs table states exactly the rule the proof checks.
- Stated in the documents: yes. Where the documents say it: main:32691
- Rests on: The check that vflPreFlag is false before notifying is written in the NWDAF as the NOTE of the data type requires.; The report is modelled as carrying exactly one of the three optional fields by construction.
- Next step: None for the flag; the field exclusivity needs a model of the body rather than of traces.

A08: notification values match the subscription

In a notification accepted by the consumer, event, vflCorrId, iterationNum and notifCorrId must correspond to the same set of values in the subscription it created or modified; the subscription parameters accepted by the NWDAF must correspond to the values the consumer actually proposed.

The answers of this target changed after this reading was written.

An accepted report belongs to a subscription the consumer itself created and updated to training, and its event, vflCorrId, notifCorrId and iteration number are those of that subscription, because the consumer accepts a report only when the values it chose itself match. The agreement that the NWDAF actually sent it could not be decided.

- Result: No answer; answers what is asked: partly; usable with caveats
- Value medium: I searched the security clause and the data model for a binding rule for vflCorrId and found none; the document only defines the field, so the fact that the consumer's own check is the only binding is not stated.
- Stated in the documents: no
- Rests on: The consumer's check on notifCorrId and vflCorrId is the protection; the model gives both values to the consumer only.; The agreement on the sender of the report is not decided.
- Open: the agreement on the sender of a notification
- Next step: Bind the notification to the session in the model as for A03.

K02: training data stays with the two parties

Intermediate training information, intermediate results and sample identifiers (SUPI/GPSI) are visible only to the subscribing consumer and the NWDAF; intermediate forwarders such as the SCP and SEPP cannot read their contents.

The model shows only that a malicious NF receives the results of a subscription it created and updated itself, which the API allows by design; the consumer's own reports stay on its TLS callback because the attacker cannot supply the consumer's notifCorrId or vflCorrId. The recorded attack is an artifact of the query, and confidentiality of the consumer's training data against another NF is not shown to fail.

- Result: Reachable only; answers what is asked: no; not usable
- Value none: The attack trace is the attacker's own subscription and says nothing about the protocol's confidentiality.
- Stated in the documents: unclear
- Rests on: The callback is modelled as a channel only the subscriber can read, which is stronger than TLS for confidentiality.; A relay that terminates TLS is not modelled, so nothing is said about an SCP or SEPP reading the reports.
- Open: whether a relay can read a notification, and whether the hijack of Z03 lets the attacker obtain the consumer's reports
- Next step: Model the callback as a public endpoint reachable by any NF and ask the confidentiality query again.

K01: access token kept from the attacker

An OAuth2 access token is known only to the NRF that issued it and the NF service consumer that obtained it; an attacker cannot derive a valid token usable to access the Nnwdaf_VFLTraining API.

An attacker on the network never learns the consumer's access token: tokens travel only on the connections of the two endpoints that hold them, and the token the malicious NF holds is its own.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: RFC 6749 clause 10.3 requires access tokens to be kept confidential in transit and in storage.
- Stated in the documents: yes. Where the documents say it: rfc6749:2859
- Rests on: Tokens are opaque and only the NRF can create them; their format, signature and expiry are not modelled.; TLS is modelled as a private channel, so confidentiality in transit is assumed rather than shown.
- Next step: None.

U01: notification correlation comes from the consumer

In a notification it accepts, the consumer requires that vflCorrId and iterationNum come from the subscription the consumer itself submitted.

The vflCorrId and iteration number of an accepted report come from the consumer's own subscription: the consumer inserts a fresh vflCorrId and accepts a report only when the same value returns, so a report about another subscription is not accepted.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The data model defines vflCorrId as a correlation value but states no generation or binding rule, and the security clause does not mention it.
- Stated in the documents: no
- Rests on: The consumer checks its own vflCorrId and notifCorrId; the document does not require the check, it only defines the fields.; The vflCorrId is treated as a value the consumer chooses and keeps from other NFs.
- Open: whether a consumer that reuses one vflCorrId across subscriptions is still protected
- Next step: Model a consumer that reuses one vflCorrId and ask the agreement again.

U02: training report accepted once

The same training report (notifCorrId, vflCorrId, iterationNum) is accepted by the consumer at most once, not across sessions and not across rounds.

The uniqueness of a training report across sessions and rounds could not be decided; the replicated sender and the duplicated subscription make the prover reach the same report twice.

- Result: No answer; answers what is asked: no; not usable
- Value low: The answer comes from the model's multiplicity, not from the protocol.
- Stated in the documents: unclear
- Rests on: The model sends one report per subscription in the modelled run, so the answer is dominated by the replication abstraction.
- Open: whether reports of two rounds can be confused when the iteration number repeats
- Next step: Unroll two rounds with distinct iteration numbers and ask the injective query there.

Q3: capability check differs from training completion

The question concerns the semantic distinction between a capability check and the completion of training, and the consistency of the finite state after a cancellation.

A successful capability check is a preparation subscription and produces no report until the consumer updates vflPreFlag to false, so the check is not mistaken for a training completion; after a delete the NWDAF refuses to notify the deleted subscription. The document defines no further state machine, and another NF can force the delete.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The purpose of the capability check is stated in 4.10.1.1 and 4.10.2.2.1 and the no-notification rule in the VflTrainingSubs NOTE.
- Stated in the documents: yes. Where the documents say it: main:32691
- Rests on: The only state rules modelled are the vflPreFlag NOTE and the deletion marker.; The model bounds the training to one report and does not model a checkpoint resume.
- Open: checkpoint resume and whether a resumed round can be reported with a stale iteration number
- Next step: Model chkPoint and resIterNum and ask about the resumed round.

X01: mutual authentication of the SBI exchange

The NWDAF acts on a subscription request only from a peer whose NF identity it has authenticated, and the NF service consumer accepts a notification only from that peer; neither party acts on a message from a third NF that holds its own credentials.

The answers of this target changed after this reading was written.

The NWDAF acts only on requests from a peer whose identity it authenticated and to which the presented token was issued, so a third NF cannot drive it under another NF's identity. The corresponding agreement for the consumer, that it accepts a notification only from the NWDAF it subscribed to, could not be decided.

- Result: No answer; answers what is asked: partly; usable with caveats
- Value medium: TS 33.501 clause 5.9.2.1 is the requirement shown for the request direction.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2871
- Rests on: Mutual authentication is taken from TLS and modelled as the connection identity.; The notification side is undecided.
- Open: the notification direction
- Next step: Model the callback as a public endpoint and ask whether a report from another NF is accepted.

X02: invalid requests refused

The NWDAF refuses a create, modify or delete request that names no existing subscriptionId, or that carries no token, an expired token, or a token for another audience or scope, and answers with an error instead of acting.

The NWDAF never acts on a request whose token subject differs from the identity it authenticated, so a request claiming another NF's identity with an own token is refused; a request naming a subscriptionId that does not exist never matches the store.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: TS 33.501 clause 5.9.2.1 requires each NF to validate incoming messages, and TS 29.500 clause 6.7.3 requires the token checks.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2875
- Rests on: The unknown-subscription case is structural: the lookup cannot match and the process stops, which ProVerif models as a refusal without an answer.; Token expiry is not modelled.
- Open: the error answer to a refused request, which the model does not produce
- Next step: Model the 401 and 403 answers and check that a refusal does not leak which check failed.

X03: access token not usable by another NF

An access token that leaked to another NF cannot be used to act in the name of the NF it was issued to, because the producer requires the token subject to be the identity it authenticated on the connection.

An access token that leaks to another NF cannot be used to act in the name of the NF it was issued to, because the NWDAF requires the token subject to be the identity it authenticated on the connection; the attacker keeps the token but has no connection under the owner's identity.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: TS 29.500 clause 6.7.5 requires a 403 when the subject claim does not match the client credentials assertion, and the model shows the equivalent check.
- Stated in the documents: yes. Where the documents say it: ts-29-500:5068
- Rests on: The model gives the leaked-token attacker only the token, not the consumer's TLS credentials, which is what RFC 6749 10.3 and TS 29.500 6.7.5 assume.; The subject-claim check is modelled as a comparison in the NWDAF.
- Open: the case where the NF's TLS credentials also leak, outside the attacker set
- Next step: Add an attacker holding both the token and the client credentials.

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
Z01	proved		z01_create			4 proved, 1 undecided
Z03	proved		x02_foreign		z03_update (holds_in_source); z03_delete (holds_in_source); z03_patch (holds_in_source)	2 proved, 3 attack
A03	proved		z01_create			1 proved, 1 undecided
A04	none					1 undecided
R02	none					1 undecided
R03	none					1 undecided
S05	reachable		s05_delete			3 reachable
S04	proved		s04_preflag			1 proved
A08	proved		z03_notify			1 proved, 1 undecided
K02	reachable		k02_reach		k02_result (model_artifact)	2 reachable, 1 attack
K01	proved		k01			1 proved
U01	proved		z03_notify			1 proved
U02	none					1 undecided
Q3	proved		s04_preflag			1 proved
X01	proved		z01_create			3 proved, 1 undecided
X02	proved		x02_foreign			2 proved
X03	proved		x02_foreign			1 proved

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
Z01	every subscription creation the NWDAF accepts (AcceptedCreate) is backed by a token the NRF issued to the subject it records; the attacker holds an own NF identity and an own token for the nnwdaf-vfltraining scope	`nf_service_scope`	proved	z01_create	none
Z01	every modification (PUT/PATCH) the NWDAF accepts is backed by an NRF-issued token for the service scope, presented by the subject it names	`nf_service_scope`	proved	z01_update	none
Z01	every deletion the NWDAF accepts is backed by an NRF-issued token for the service scope, presented by the subject it names	`nf_service_scope`	proved	z01_delete	none

target	claim	attacker	answer	query	control
Z01	every notification delivery the consumer accepts is authorized by a valid token: the notify operation carries no token (TS 29.500 clause 6.7.3), so what can be asked is that the accepted report was delivered by the NWDAF for the consumer's own subscription	`nf_service_scope`	undecided	z01_notify	
Z01	every subscription creation the NWDAF accepts is backed by a token the NRF issued to the subject it records, also against an attacker on the network alone	`network`	proved	z01_create	none
Z03	only the subject a subscription belongs to may modify it: a full PUT of another NF's subscriptionId is accepted	`nf_scope_resource_uri`	attack (holds_in_source)	z03_update	
Z03	only the subject a subscription belongs to may delete it: a DELETE of another NF's subscriptionId is accepted	`nf_scope_resource_uri`	attack (holds_in_source)	z03_delete	
Z03	only the subject a subscription belongs to may partially modify it: a PATCH of another NF's subscriptionId is accepted	`nf_scope_resource_uri`	attack (holds_in_source)	z03_patch	
Z03	a token issued to one NF cannot be used to act as another: the producer rejects a request whose subject claim differs from the authenticated peer	`nf_scope_resource_uri`	proved	x02_foreign	none
Z03	a token issued to one NF cannot be used to act as another; also against an attacker on the network alone the producer rejects a request whose subject claim differs from the authenticated peer	`network`	proved	x02_foreign	none
A03	the NWDAF records as the owner of a created subscription the identity the consumer authenticated with, so a third party cannot have a subscription created under the consumer's identity	`network`	proved	z01_create	none
A03	an accepted notification was sent by the NWDAF for that subscription with the same notifCorrId and reports	`network`	undecided	x01_notify	
A04	each accepted creation with a new subscriptionId corresponds to a distinct real subscription run of the consumer	`network`	undecided	a04_create	
R02	one notification instance (notifCorrId, vflCorrId, iteration number) is accepted at most once	`network`	undecided	r02_report	
R03	one logical create, delete or notification produces at most one security-relevant side effect at the other party	`network`	undecided	a04_create	
S05	after the consumer's subscription was deleted, the NWDAF is asked to notify it and refuses (the situation the target speaks about is reached)	`network`	reachable	s05_delete	
S05	after a subscription is deleted the NWDAF is asked to notify it and refuses (situation reached for the deletion of the consumer's own subscription)	`network`	reachable	s05_delete	

target	claim	attacker	answer	query	control
S05	after a subscription is deleted the NWDAF is asked to notify it and refuses, also when another NF with the service scope and the subscription URI deleted it	`nf_scope_resource_uri`	reachable	s05_delete	
S04	a preparation subscription (vflPreFlag=true) produces no report before it is updated to vflPreFlag=false	`network`	proved	s04_preflag	none
A08	an accepted notification belongs to a subscription the consumer itself created and updated to training, so event, vflCorrId, notifCorrId and iteration number are those of its own subscription	`network`	proved	z03_notify	none
A08	an accepted report is accepted only for a subscription the consumer had updated to training, so the report agrees with its subscription context	`network`	undecided	a08_notify	
K02	intermediate training results are known only to the consumer and the NWDAF: a subscription's results never reach another NF	`nf_scope_resource_uri`	attack (model_artifact)	k02_result	
K02	the situation the claim protects is reached: a training result is produced for a subscription and delivered to its subscriber	`network`	reachable	k02_reach	
K02	the situation the claim protects is reached: a training result is produced for a subscription owned by a consumer and delivered to that subscriber	`nf_scope_resource_uri`	reachable	k02_reach	
K01	the consumer's access token is not known to an attacker on the network	`network`	proved	k01	none
U01	the vflCorrId and iteration number of an accepted notification come from the consumer's own subscription	`network`	proved	z03_notify	none
U02	one training report (notifCorrId, vflCorrId, iteration number) is accepted at most once, not across sessions or rounds	`network`	undecided	r02_report	
Q3	a successful capability check (a preparation subscription) is not a training completion: it produces no report before the subscription is updated	`network`	proved	s04_preflag	none
X01	the NWDAF records as the owner of a new subscription only the identity whose token it was given and which is the identity it authenticated on the connection (AcceptedCreate carries an NRF-issued subject)	`network`	proved	z01_create	none
X01	every notification the consumer accepts (GotReport) was sent by the NWDAF for that subscription (SentReport with the same subscriptionId and iteration number)	`network`	undecided	x01_notify	
X01	the NWDAF records as owner only the identity whose token it validated, so a malicious NF cannot have a subscription created under another NF's identity	`nf_service_scope`	proved	z01_create	none

target	claim	attacker	answer	query	control
X01	every creation the NWDAF accepts under the consumer's identity corresponds to a creation request the consumer itself sent	`network`	proved	x01_request_c	none
X02	no request is acted on whose token subject differs from the identity authenticated on the connection (no ForeignIdentity event is ever reached)	`nf_service_scope`	proved	x02_foreign	none
X02	no request is acted on whose token subject differs from the identity authenticated on the connection, also against an attacker on the network alone	`network`	proved	x02_foreign	none
X03	a request that names a subscription and carries the access token of another NF is refused: the producer requires the token subject to equal the authenticated connection identity (ForeignIdentity unreachable)	`nf_token_leaked`	proved	x02_foreign	none

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: 58, prover 9 s, wall 16 min, cost 0.168 USD.

Context: the host removed stale parts from the conversation 1 time(s), 473732 characters in all (earlier reasoning, and what showed the model before its latest change). The trajectory file keeps everything.

ProVerif: 119 proofs run, 94 decided, 25 without an answer, 5 s, 27 answers recorded.

What was analysed

Protocol	NWDAF VFL client training API
Version	19.8.0
Analysed	One trusted AF/NWDAF VFL server and one NWDAF VFL client; one capability check, subscription, a bounded training round report and cancellation; raw features/gradients are opaque, so machine learning convergence, poisoning robustness or differential privacy cannot be proved.
Not analysed	The symbolic model does not prove computational or post-quantum security of the algorithms; attack traces hold only under explicit assumptions and are not directly equivalent to specification vulnerabilities. Each property must have a reachability/positive control. Rel-19 is fixed. A distinction is made between an external network attacker and a malicious legitimate NF holding its own credentials; 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 explicitly defined. The target is the specified API sub-flow, not complete 5G security.
Source text	293 numbered passages (68550 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	4.10	Nnwdaf_VFLTraining Service
main	4.10.1	Service Description
main	4.10.1.1	Overview
main	4.10.1.2	Service Architecture
main	4.10.1.3	Network Functions
main	4.10.1.3.1	Network Data Analytics Function (NWDAF)
main	4.10.1.3.2	NF Service Consumers
main	4.10.2	Service Operations
main	4.10.2.1	Introduction
main	4.10.2.2	Nnwdaf_VFLTraining_Subscribe service operation
main	4.10.2.2.1	General

Document	Section	Title
main	4.10.2.2.2	Subscription for VFL training information
main	4.10.2.2.3	Update of subscription for VFL training information
main	4.10.2.2.4	Partial update subscription for VFL training information
main	4.10.2.3	Nnwdaf_VFLTraining_Unsubscribe service operation
main	4.10.2.3.1	General
main	4.10.2.3.2	Unsubscribe from VFL training information
main	4.10.2.4	Nnwdaf_VFLTraining_Notify service operation
main	4.10.2.4.1	General
main	4.10.2.4.2	Notification about VFL training information
main	5.9	Nnwdaf_VFLTraining Service API
main	5.9.1	Introduction
main	5.9.2	Usage of HTTP
main	5.9.2.1	General
main	5.9.2.2	HTTP standard headers
main	5.9.2.2.1	General
main	5.9.2.2.2	Content type
main	5.9.2.3	HTTP custom headers
main	5.9.3	Resources
main	5.9.3.1	Overview
main	5.9.3.2	Resource: VFL Training Subscriptions
main	5.9.3.2.1	Description
main	5.9.3.2.2	Resource Definition
main	5.9.3.2.3	Resource Standard Methods
main	5.9.3.2.3.1	POST
main	5.9.3.2.4	Resource Custom Operations
main	5.9.3.3	Resource: Individual VFL Training Subscription
main	5.9.3.3.1	Description
main	5.9.3.3.2	Resource Definition
main	5.9.3.3.3	Resource Standard Methods
main	5.9.3.3.3.1	PUT
main	5.9.3.3.3.2	DELETE
main	5.9.3.3.3.3	GET
main	5.9.3.3.3.4	PATCH
main	5.9.3.3.4	Resource Custom Operations
main	5.9.4	Custom Operations without associated resources
main	5.9.5	Notifications

Document	Section	Title
main	5.9.5.1	General
main	5.9.5.2	VFL Training Notification
main	5.9.5.2.1	Description
main	5.9.5.2.2	Target URI
main	5.9.5.2.3	Standard Methods
main	5.9.5.2.3.1	POST
main	5.9.6	Data Model
main	5.9.6.1	General
main	5.9.6.2	Structured data types
main	5.9.6.2.1	Introduction
main	5.9.6.2.2	Type VflTrainingSubs
main	5.9.6.2.3	Type VflTrainingSub
main	5.9.6.2.4	Type VflCheckpointInfo
main	5.9.6.2.5	Type VflInteropInfo
main	5.9.6.2.6	Type VflTrainingNotify
main	5.9.6.2.7	Type VflIntermedTrainInfo
main	5.9.6.2.8	Type VflTrainingSubsPatch
main	5.9.6.2.9	Type: VflTrainingReport
main	5.9.6.2.10	Type: VflSampleId
main	5.9.7	Error Handling
main	5.9.7.1	General
main	5.9.7.2	Protocol Errors
main	5.9.7.3	Application Errors
main	5.9.8	Feature negotiation
main	5.9.9	Security

The purpose of the protocol, as located in the documents in the source step: a trusted AF/NWDAF VFL server and an NWDAF VFL client; a capability check, a subscription, a bounded report of training rounds and a cancellation; the raw features and gradients are opaque, and machine-learning convergence, poisoning robustness and differential privacy cannot be proven.

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
Z01	each operation covered by a valid token	the NWDAF as NF service producer: every subscription creation, modification or deletion it accepts is backed by an access token the NRF issued, for the nnwdaf-vfltraining scope, unexpired; the notification delivery is the operation the baseline leaves without a token	TS 29.520 5.9.9 says access to the API may be authorized with OAuth2 client credentials, NRF as authorization server; TS 29.500 6.7.3 requires the NF service to validate the token, its expiry and its access scope before allowing access to the requested resource, and says the notify operation uses no OAuth2 token.	ts-29-500:4994, ts-29-500:5032, s0290, s0291, s0293
Z03	subscription bound to its authorized subject	the NWDAF: modification, deletion and notification of one subscriptionId are reserved to the subject the subscription belongs to, within the context its token covers; a token, a subscriptionId or a correlation identifier is not transplantable to another subject or another subscription	TS 29.520 5.9.9 defines one scope for the whole service and no resource- or operation-level scopes; TS 29.500 6.7.3 says a service scope grants generic access to the API, and that resource/operation-level scopes exist for operations that require a specific authorization. The service operation clauses state no check that the requester owns the subscription.	ts-29-500:4996, ts-29-501:1795, s0042, s0072, s0085, s0143, s0293
A03	consumer and NWDAF act on real runs	the NWDAF when it accepts a subscription creation, and the NF service consumer when it accepts a notification: the accepted values correspond to a real run of the other party	TS 29.520 4.10.2.2.2 and 4.10.2.4.2 describe the request and the notification; TS 33.501 5.9.2.1 requires mutual authentication for direct communication between consumer and producer and that each NF validate all incoming messages.	ts-33-501:2871, s0036, s0042, s0085, s0086
A04	fresh identifier per real run	the NWDAF's subscriptionId and the consumer's notification acceptance: each corresponds to a distinct run of the other party	TS 29.520 4.10.2.2.2 has the NWDAF assign a subscriptionId and store the subscription; 4.10.2.4.2 has the consumer store the notification and answer 204; replay protection is required on the SBI by TS 33.501 5.9.2.1.	ts-33-501:2865, s0038, s0039, s0040, s0085, s0086
R02	one acceptance per notification instance	the NF service consumer: a notification instance with one notifCorrId and one set of reports is accepted at most once	TS 29.520 5.9.5.2.3.1 and 4.10.2.4.2 give no deduplication or replay window rule; the property rests on the correlation identifier.	ts-33-501:2915, s0085, s0086, s0262
R03	one side effect per logical operation	the NWDAF's subscription store and the consumer's stored notification: one logical create, delete or notification produces at most one security-relevant side effect	TS 29.520 4.10.2.2.2 (create, assign subscriptionId, store), 4.10.2.3.2 (remove the subscription, answer 204), 4.10.2.4.2 (store the notification, answer 204); no retry semantics are defined.	s0038, s0039, s0040, s0073, s0074, s0086
S05	deleted subscription stays deleted	the NWDAF: after a successful DELETE (204) the subscriptionId accepts no modification and produces no notification; the old subscription context cannot re-trigger them	TS 29.520 4.10.2.3.2: on acceptance the NWDAF removes the subscription and answers 204. Nothing is said of in-flight notifications or of reuse of the identifier.	s0073, s0074, s0075
S04	preparation, rounds and report fields consistent	the NWDAF and the consumer: a preparation subscription (vflPreFlag=true) produces no notification before it is updated to false; a report carries at most one of intermediateRes, termCause, vflNotOnTimeInd, consistent with the subscription context	TS 29.520 5.9.6.2.2 NOTE: with vflPreFlag=true the producer sends no notification until the subscription is updated to vflPreFlag=false and trainReports is absent from the POST response; 5.9.6.2.9 NOTE 1: only one of intermediateRes, termCause, vflNotOnTimeInd is provided.	main:32691, s0250, s0271

ID	Name	About	What the documents say	Where
A08	notification values match the subscription	the NF service consumer: event, vflCorrId, iterationNum and notifCorrId in an accepted notification correspond to the values of its own subscription; the NWDAF accepted the values the consumer proposed	TS 29.520 5.9.6.2.3 and 5.9.6.2.9 define the correlated attributes; 4.10.2.4.2 has the NWDAF use the notifUri and the report carry notifCorrId. No generation or binding rule for vflCorrId is given.	s0253, s0262, s0271, s0085
K02	training data stays with the two parties	the NF service consumer and the NWDAF: intermediate training information, intermediate results and sample identifiers (SUPI/GPSI) are readable only by them, not by another NF nor by a relay	TS 29.520 5.9.6.2.7 and 5.9.6.2.9 carry the intermediate results, 5.9.6.2.10 carries SUPI/GPSI; TS 33.501 5.9.2.4 requires confidentiality and integrity between SCP and NFs; the task scope states that a legitimate NF cannot read another NF's TLS session.	ts-33-501:2915, s0265, s0271, s0274
K01	access token kept from the attacker	the NRF and the NF service consumer: an access token usable on the Nnwdaf_VFLTraining API is known only to the NRF and the NF it was issued to	TS 29.520 5.9.9 delegates token issue and protection to the OAuth2 baseline; RFC 6749 10.3 requires access tokens to be kept confidential in transit and storage and to be unguessable, and 10.8 forbids sending them in the clear.	rfc6749:2859, rfc6749:3013, s0290, s0291
U01	notification correlation comes from the consumer	the NF service consumer: the vflCorrId and iterationNum of an accepted notification come from the subscription it submitted	TS 29.520 5.9.6.2.3 has the vflCorrId in the subscription, 5.9.6.2.9 has it in the report; no binding rule is given.	s0253, s0271, s0085
U02	training report accepted once	the NF service consumer: one training report (notifCorrId, vflCorrId, iterationNum) is accepted at most once, and not across sessions or rounds	TS 29.520 5.9.6.2.6 and 5.9.6.2.9 define the report identifiers; 29.520 gives no deduplication rule and TS 33.501 requires replay protection on the SBI.	ts-33-501:2865, s0262, s0271, s0086
Q3	capability check differs from training completion	the NF service consumer and the NWDAF: a successful capability check (a preparation subscription, vflPreFlag=true) is not a training completion, and a cancelled subscription leaves a consistent finite state	TS 29.520 4.10.1.1 and 4.10.2.2.1 give the capability check as its own purpose; 5.9.6.2.2 NOTE forbids notifications for a preparation subscription; 4.10.2.3.2 removes the subscription on DELETE. The document defines no state machine beyond these.	s0006, s0031, s0075, s0250
X01	mutual authentication of the SBI exchange	the NF service consumer and the NWDAF, against any other NF of the network	TS 33.501 5.9.2.1 requires mutual authentication for direct communication between NF service consumer and producer and has every NF validate all incoming messages; TS 29.500 6.7.2 requires TLS for the service based interfaces.	ts-33-501:2871, ts-29-500:4980
X02	invalid requests refused	the NWDAF, against any NF that holds its own credentials	TS 33.501 5.9.2.1: each NF shall validate all incoming messages and reject or discard messages that are not valid; TS 29.500 6.7.3 requires the validation of token, expiry and scope and the 401/403 answers.	ts-33-501:2875, ts-29-500:4994
X03	access token not usable by another NF	the NWDAF, holding the token of a legitimate consumer	RFC 6749 10.3 states that OAuth 2.0 gives the resource server no method to verify that the token presented by a client was issued to that client; TS 29.500 6.7.5 requires the subject claim of the access token to match the client credentials assertion of the requesting NF.	rfc6749:2879, ts-29-500:5068

Added from the documents by the analyst: X01, X02, X03.

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: 39. Not taken: 359. 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	the introduction to the RFC 6749 security considerations names the client profiles the sections are written for and points to the OAuth threat model; it states no property of this API
P003 to P006	rfc6749:2790-2823	Z01, K01	
P007 to P010	rfc6749:2827-2850	Z01, Z03, `nf_service_scope`	
P011 to P015	rfc6749:2859-2881	K01, Z03	
P016 to P033	rfc6749:2885-3003	Outside the scope	refresh tokens, authorization codes, redirection URIs and resource owner password credentials do not exist in a client credentials grant over the SBI
P034 to P035	rfc6749:3013-3019	K01, K02	
P036	rfc6749:3023-3028	A03	
P037 to P039	rfc6749:3032-3041	K01	
P040 to P062	rfc6749:3045-3210	Outside the scope	phishing, cross-site request forgery, clickjacking, code injection, open redirectors and the implicit grant concern user-agent flows; this API is consumed by NFs with client credentials
P063	ts-33-501:2491-2491	Outside the scope	bidding down of AS/NAS security features concerns the radio interface, not this SBI API
P064 to P074	ts-33-501:2495-2517	Outside the scope	subscription and serving network authentication, UE authorization and 5G key requirements are access-layer properties outside the APIs analysed here
P075 to P213	ts-33-501:2523-2853	Outside the scope	requirements on the UE, gNB, ng-eNB, SEAF, UDM, SIDF, UDR and subscriber privacy are access and subscriber layers the task leaves out
P214	ts-33-501:2859-2859	No goal stated	the trust boundary paragraph only says which requirements apply to messages that cross trust zones; it states no property of this API
P215 to P220	ts-33-501:2865-2875	A03, A04, Z01, R02, `nf_service_scope`	
P221 to P227	ts-33-501:2879-2891	Z01, K01	
P228 to P234	ts-33-501:2895-2907	Outside the scope	these requirements concern the NEF-AF interface, which is not the interface analysed here
P235 to P241	ts-33-501:2911-2923	K02	
P242 to P303	ts-33-501:2929-3055	Outside the scope	end-to-end core network interconnection, SEPP and roaming intermediaries concern roaming across PLMNs, which this single-PLMN API analysis leaves out

Paragraphs	Where	Taken up as	Why not
P304 to P318	ts-33-501:3059-3091	Outside the scope	protection of attributes for roaming and the IPUPS requirements concern interconnection between networks, outside this API
P319 to P375	ts-33-501:3097-3231	Outside the scope	signalling monitoring, security visibility, algorithm selection, 5G-RG and NSSAAF requirements concern other parts of the system
P376 to P398	ts-33-501:6540-11901	Outside the scope	N3 security, the EAP-TLS annexes, IAB, MBS and the NEF requirements concern the user plane, the access layer or other interfaces

The parties and the protected values, one by one

The host listed the 28 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: 7. Without: 21. 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	`nf_service_scope`	
S02	NWDAF	party	Everything lost by definition	the NWDAF computes and holds all training results, so its own dishonesty voids the confidentiality properties by definition and the documents ask no property of it under that assumption
S03	NRF	party	Everything lost by definition	an attacker that controls the authorization server can mint any token, so every authorization property asked here is lost by definition
S04	SCP/SEPP	party	Outside the scope	the model takes TLS per endpoint, so a relay that terminates the connection of another NF is not modelled; TS 33.501 5.9.2.4 states that all parties must trust the SCP
S05	attacker	party	Nothing gained	the attacker entry of the target list is the network attacker of the model, which has nothing beyond the network
S06	OAuth2 access token	value	`nf_token_leaked`	
S07	nnwdaf-vfltraining scope	value	Nothing gained	the scope name is public and carried in the token by design
S08	subscription resource	value	`nf_scope_resource_uri`	
S09	access token	value	`nf_token_leaked`	
S10	subscriptionId	value	`nf_scope_resource_uri`	
S11	notifUri	value	Nothing gained	the callback URI is an address, not a secret; the model reaches the callback only over the TLS connection of the subscriber
S12	notifCorrId	value	Nothing gained	the notifCorrId travels only inside the TLS-protected subscription request and the consumer checks it on every notification; a third party that holds it gets no message path in the model
S13	VflTrainingSubs	value	Nothing gained	the subscription body travels inside the TLS-protected request and its VFL contents are opaque to the model
S14	VflTrainingNotify reports	value	Nothing gained	the notification reports travel only to the subscriber's TLS callback
S15	stored notification	value	Nothing gained	a stored notification stays at the consumer that accepted it
S16	subscription state	value	`nf_scope_resource_uri`	
S17	vflPreFlag	value	Nothing gained	vflPreFlag is a public boolean carried in the subscription
S18	iterationNum	value	Nothing gained	the iteration number is a public counter carried in the report

ID	Party or value	Kind	Attacker	Why none
S19	chkPoint	value	Nothing gained	the checkpoint information is a scheduling attribute of the subscription, not a secret
S20	VflTrainingReport fields	value	Nothing gained	the report fields other than the results are public correlation and status values
S21	vflCorrId	value	Nothing gained	the vflCorrId is the correlation value the consumer itself chose; holding it does not reach the callback
S22	event	value	Nothing gained	the analytics event identifier is public
S23	VflIntermedTrainInfo	value	Everything lost by definition	the intermediate training information is the value K02 protects; once another party holds it the confidentiality property is broken and no further property of this API is asked about it
S24	intermediateRes	value	Everything lost by definition	intermediateRes is the protected training result; holding it is the violation K02 speaks of
S25	VflSampleId	value	Outside the scope	the SUPI and GPSI sample identifiers raise subscriber privacy, which the task leaves out
S26	VflTrainingReport	value	Everything lost by definition	the report carries the training results; holding it is the violation K02 speaks of
S27	VFL capability check result	value	Nothing gained	the capability check result is returned to the requesting consumer only and is a public boolean outcome
S28	subscription state	value	`nf_scope_resource_uri`	

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		
`nf_service_scope`	an NF instance identity of its own in the same network and an access token the NRF issued to it for the nnwdaf-vfltraining scope, obtained before the run	TS 29.520 5.9.9 defines one scope for the whole service and no resource- or operation-level scopes; TS 29.500 6.7.3 lets any NF service consumer use OAuth2 with client credentials, and the task scope separates the external network attacker from a malicious legitimate NF that holds its own credentials.	ts-29-500:4990, ts-29-500:4998, s0290, s0293
`nf_scope_resource_uri`	the credentials and the token of nf_service_scope, and the subscriptionId (the resource URI) of a subscription of another consumer, which the documents do not declare secret	TS 29.500 6.7.3 has the producer validate the token, its expiry and its access scope before allowing access to the requested resource, and says the service scope grants generic access to the API; 29.520 5.9.9 defines no resource-level scope, so the resource URI is the only thing that selects the resource and 29.520 4.10.2.3.2 states no ownership check.	ts-29-500:4994, ts-29-500:5000, s0042, s0143, s0293

Attacker	Has	Basis	Where
`nf_token_leaked`	an access token the NRF issued to the legitimate NF service consumer for the nnwdaf-vfltraining scope, but not the consumer's TLS credentials	RFC 6749 10.3 treats a leaked access token as the classic case and states that the resource server has no method to check that the token was issued to the presenting client.	rfc6749:2859, rfc6749:2879

ID	`network`	`nf_service_scope`	`nf_scope_resource_uri`	`nf_token_leaked`
Z01	Holds	No answer		
Z03	Holds		Fails	
A03	No answer			
A04	No answer			
R02	No answer			
R03	No answer			
S05	No answer		No answer	
S04	Holds			
A08	No answer			
K02	No answer		No answer	
K01	Holds			
U01	Holds			
U02	No answer			
Q3	Holds			
X01	No answer	Holds		
X02	Holds	Holds		
X03				Holds

Source and targets

Source: 293 units, 68550 bytes, frozen in source-delivery-01.json. Targets: 13 approved (Z01, Z03, A03, A04, R02, R03, S05, S04, A08, K02, K01, U01, U02); the questions asked: 3. The list of targets was approved as drafted, beforehand, with --approve-targets.

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

- approve_package: approved source package protocol-1a1c8dc25e8c702e2c5626af7852

This step as it reports itself: 1 requests (1 failed), 0.000 USD, 46821 tokens in and 5489 out, 266 s.

Notes on the prover and the model

What the analyst named for targets:

step	targets	features	reason
40	Z01, Z03, A03, A04, R02, R03, S05, S04, A08, K02, K01, U01, U02, Q3, X01, X02, X03	many_sessions, plain_primitives, other	The model covers any number of concurrent consumer sessions and of malicious-NF requests on one shared subscription store, and it uses only idealised MAC-free primitives: opaque tokens, tagged tuples and private (TLS) channels. Nothing is modelled of counting, exclusive-or or Diffie-Hellman; TLS is modelled as a private channel between the endpoints and each callback as a per-subscription private channel, which is stronger than TLS for confidentiality and says nothing about a relay that terminates TLS.

Falsified and not submitted

The prover falsified these properties and they are in no recorded answer.

- `a04_create` (model.pv, on an earlier version of the model, not run since): sid: subid, v: bitstring, w: bitstring; inj-event(AcceptedCreate(sid, idc, v, w)) ==> inj-event(ConsumerCreate(v, w))
- `a08_notify` (model.pv, on an earlier version of the model, not run since): sid: subid, it: bitstring; event(GotReport(sid, it)) ==> event(TrainingReady(sid))

The analyst's closing note

The model covers the whole Nnwdaf_VFLTraining client flow: token issue at the NRF, the capability check as a preparation subscription, the training subscription, one intermediate-result notification and the delete, with the single service scope, the audience/scope/subject check of the token, a subscription store, a malicious legitimate NF holding its own token, and an attacker that later holds the consumer's token. Of the 17 targets, 11 are proved or reached, 4 carry an attack and 3 are undecided.

The one finding is that the API defines no resource-level scope and no service operation clause requires an ownership check, so an NF with a service-scope token and another consumer's subscription URI can PUT, PATCH or DELETE that subscription and the NWDAF answers as it would to the owner. A change adding the ownership check was checked on a copy of the model: the three attack queries become verified, the normal run stays and no other answer changes.

What holds: every creation, modification and deletion the NWDAF accepts is backed by an NRF-issued token presented by the NF it names; a leaked token cannot be used to act as another NF; the preparation flag blocks notifications until the subscription is updated; a creation under the consumer's identity corresponds to a request it sent; the consumer accepts only its own vflCorrId and notifCorrId. What is blocked: the freshness of a notification instance and the sender agreement on notifications cannot be decided, because ProVerif's replicated request handler lets one creation request build two subscriptions - a case for an unrolled model or for Tamarin+ Agent. The attacks recorded on notification confidentiality and on deletion are artefacts of the query and the model, not protocol failures. TLS is modelled as a private channel per endpoint, so a relay that terminates a connection, token expiry, and any counting or algebraic detail lie outside what was shown.