Security analysis of Review ML model monitoring with SBA isolation

19.8.0 · ProVerif+ Agent report

Documents	5G System; Network Data Analytics Services; Stage 3 (protocol definition); 5G System; Principles and Guidelines for Services Definition; Stage 3 (supporting); Security architecture and procedures for 5G system (supporting)
Properties examined	15: 8 hold, 7 fail (8 attacks)
Findings	1; known results reproduced in Appendix B: 1

Contents

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

1 Protocol and scope

The Nnwdaf_MLModelMonitor service is the 5G core interface over which one Network Data Analytics Function instance tells another how well an ML model it uses is performing. The instance containing AnLF uses the model, registers that use with the instance containing MTLF, and later reports the measured accuracy and the analytics feedback; the instance containing MTLF subscribes to those reports and receives them. Access to the API is authorized with OAuth2 access tokens issued by the NRF, and requests travel over mutually authenticated TLS.

When the exchange ends, the consumer is sure that the producer which accepted its registration and subscription is the instance named in the token, that each acceptance belongs to one of its own requests, and that the identifiers it was given are its own. The producer is sure that the notifications it receives carry the correlation identifier and model identifiers of the subscription it created, that they were sent by the producer of that subscription over its TLS session, and that no other NF instance has obtained its registration or subscription identifier.

The parties:

- **NWDAF containing AnLF**: NF service consumer, uses the ML model. Holds: NF instance identity, TLS credentials, modelId.
- **NWDAF containing MTLF**: NF service producer, monitors model accuracy. Holds: NF instance identity, TLS credentials, ML models to monitor.
- **NRF**: OAuth2 authorization server. Holds: NRF instance identity, token signing key.

Figure 1 shows the exchange as the documents specify it.

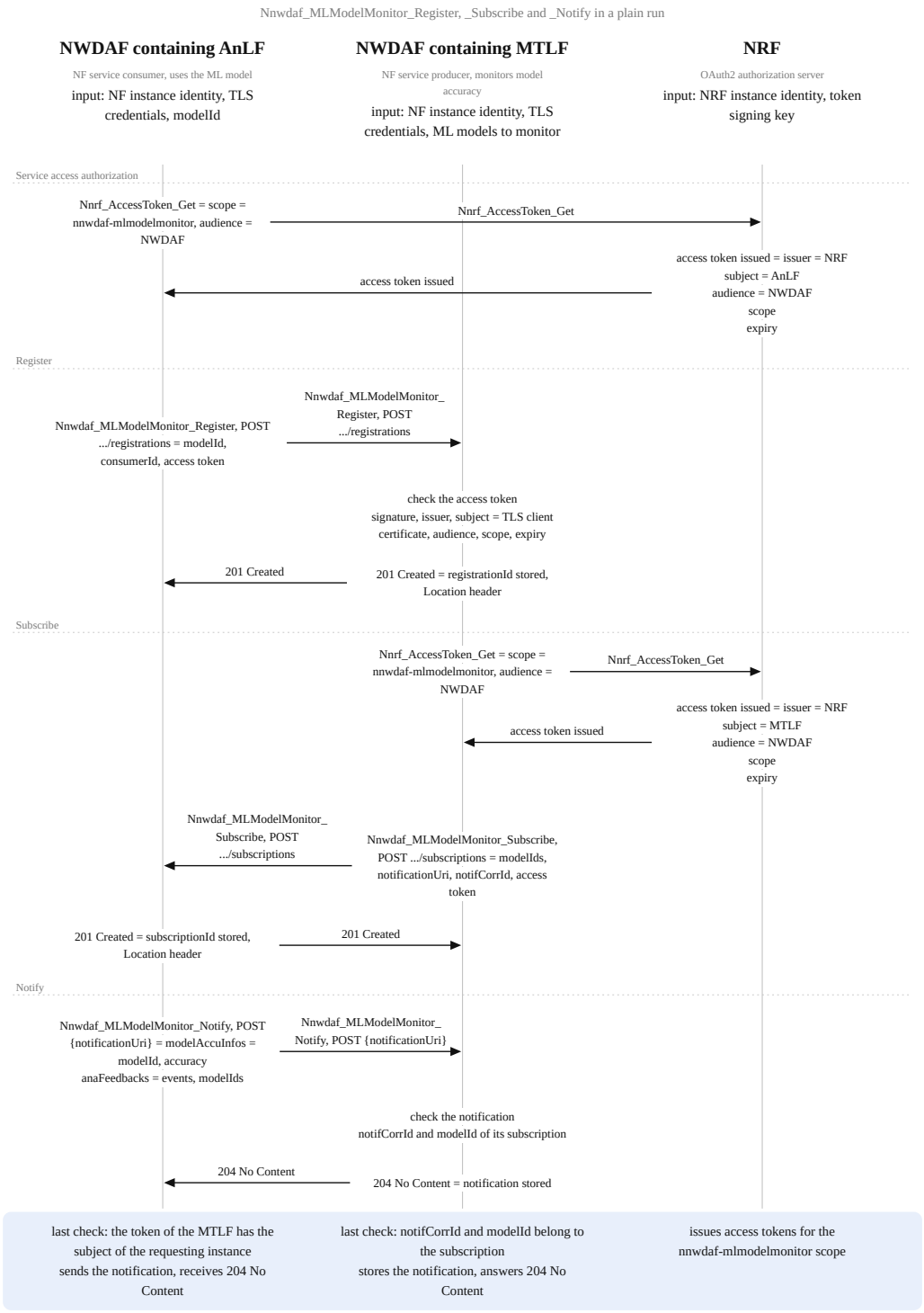


Figure 1. The protocol as specified.

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. 1 result that the documents already state is not counted and is in Appendix B.

The protocol gives what it is meant to give against the network and against a malicious NF, with one exception: the consumerId of a registration is not bound to the authenticated consumer. Every other failure is either stated in the sources or an artifact of the model. Fourteen of the fifteen targets are answered for every attacker set for them; the fifteenth, request creation, is proved in its identifier-injectivity reading and remains a model artifact in its replay reading. The notification path is protected by TLS alone, which the sources state, so it falls to a compromised SCP as the sources also state. The results rest on a simplified TLS without handshake and on an access token modelled as one signature over its claims.

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 instance can register another NF instance's consumerId

Kind	Weakness of the protocol as specified
New	Unclear: the documents state it in part, or could not be searched
Value	medium
Properties	A07, Z03, U03 (section 3)
Source text	s0040, s0041, s0042, s0480, s0347
Shown by	ProVerif attack query `a07_body_nf`; Figures 2, 3 and 4

What happens. A NWDAF instance holding a valid access token of its own invokes Nnwdaf_MLModelMonitor_Register with the consumerId of a different AnLF instance. The NWDAF containing MTLF verifies the token signature, its issuer, its audience, its scope and its subject against the TLS client certificate, then stores the registration and answers 201 Created with a registrationId. The stored registration names the other instance as the consumer.

What the attacker needs. One NF instance identity of its own with its TLS credentials and one access token issued by the NRF for the nnwdaf-mlmodelmonitor scope. No compromised key and no other credential is needed, and the instance uses its own TLS session with the producer.

What the attacker gains. The producer's registration records, and everything derived from them about which instance uses which ML model, can be made to name an instance that never asked. An operator reading the records attributes monitoring to the wrong NF.

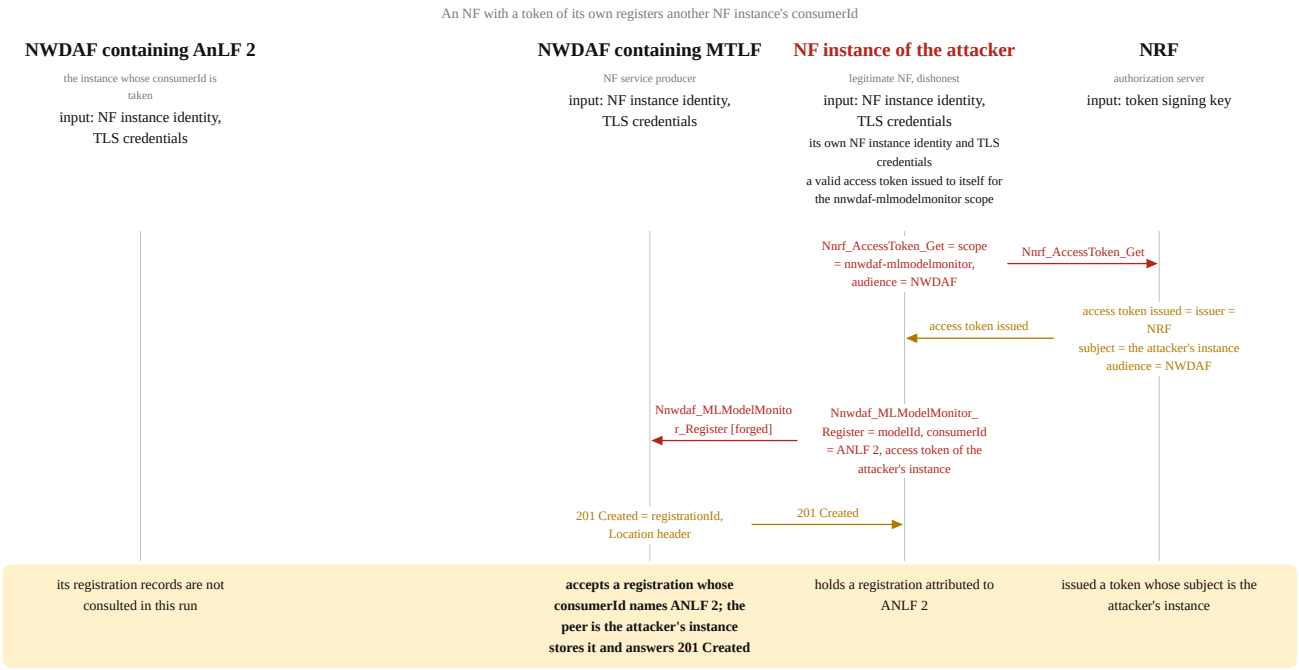


Figure 2. An NF with a token of its own registers another NF instance's consumerId. Property A07; attack query a07_body_nf.

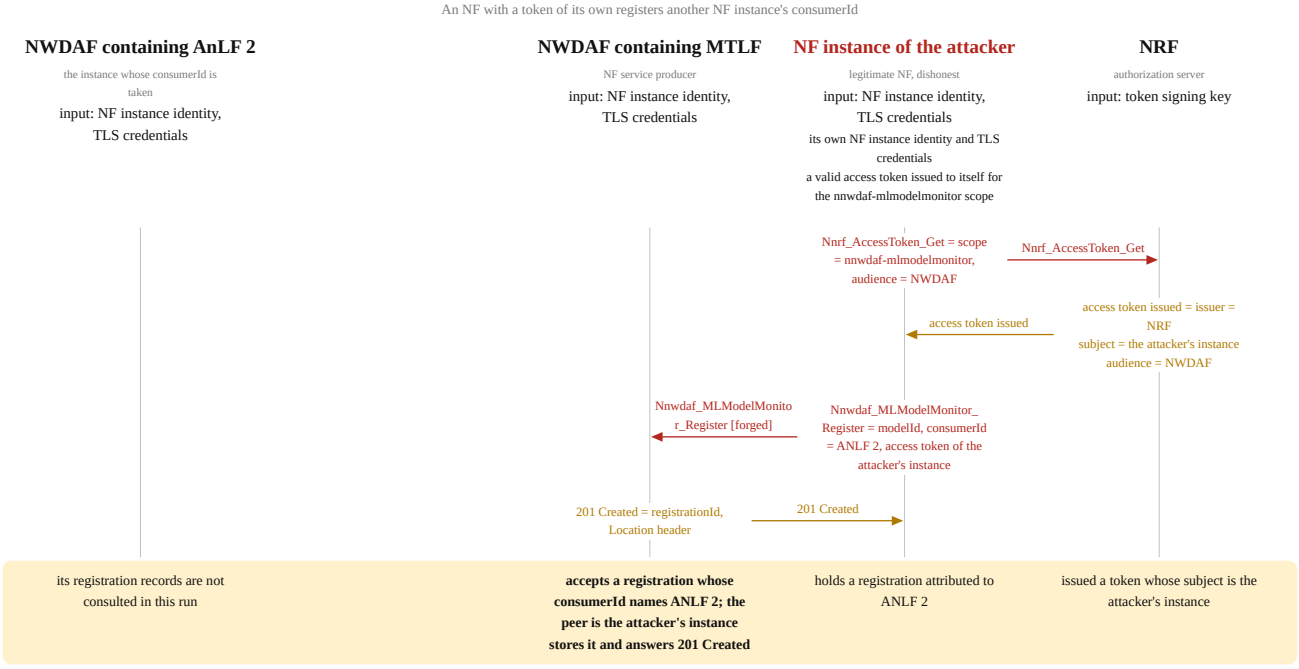


Figure 3. An NF with a token of its own registers another NF instance's consumerId. Property Z03; attack query a07_body_nf.

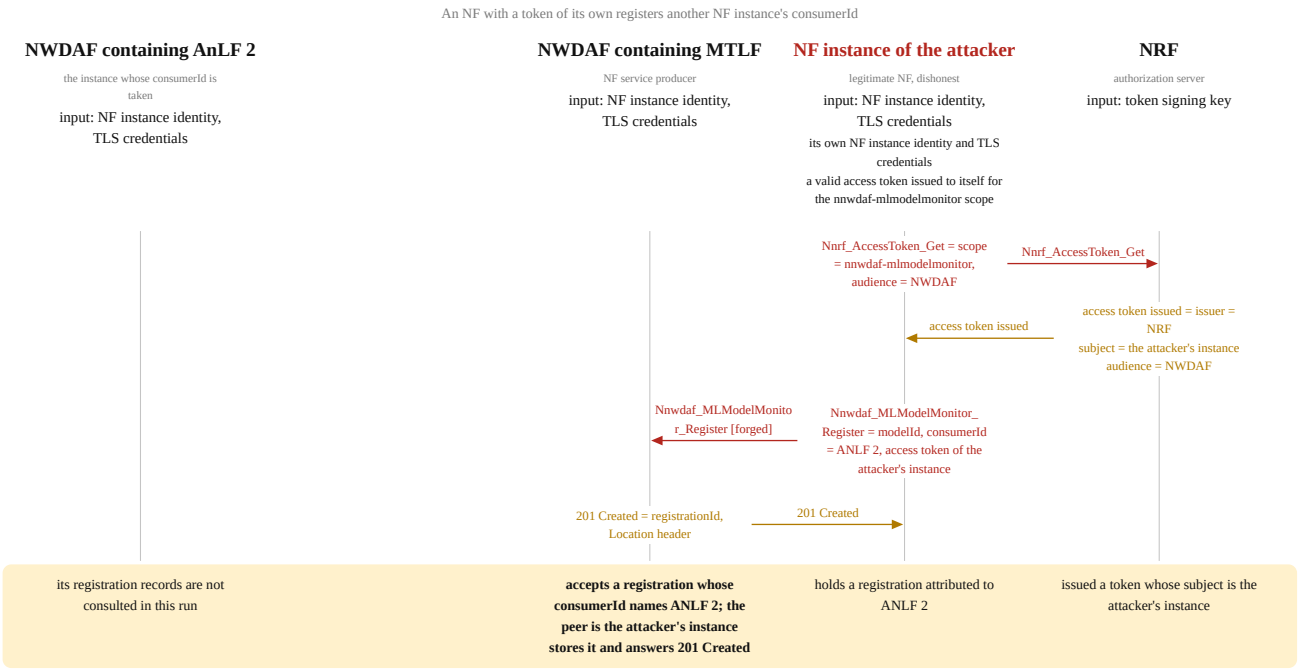


Figure 4. An NF with a token of its own registers another NF instance's consumerId. Property U03; attack query a07_body_nf.

Proposed change. In the Register handling, compare the consumerId (or consumerSetId) of the MLModelMonitorReg body with the NF Instance ID of the token subject and of the authenticated TLS client, and reject the request when they differ. The change was checked on a copy of the model: the attack query becomes verified, the normal run is still found and no other recorded answer changes.

Check of the change with ProVerif. The change was put into a copy of the model: In the Nnwdaf_MLModelMonitor_Register handling, after the token checks, the NWDAF containing MTLF compares the consumerId (respectively consumerSetId) of the MLModelMonitorReg body with the NF Instance ID of the token subject and of the authenticated TLS client, and rejects the request when they differ. On the changed model ProVerif proves `a07_body_nf`: the attack is no longer possible there. The normal run of the protocol is still found. The other 43 recorded answers are unchanged. 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/0049-fix-consumerid.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
A02	**NFinstance authentication of requests.** Whenever containedMTLFIt's...NWDAFWhen accepting a request for registration/subscription/notification/feedback, there is a real running one thatMTLFExample is the content of the opposite endAnLFIt's...NWDAFRun (and vice versa) with an identity particle size fixed toNFExample.	no compromise	Holds	medium	yes

ID	Property	Compromise assumed	Result	Value	In the documents
A03	**Notification values match the subscription.** At each timeMTLFIt's...NWDAFAt receiving notification, there is a match contentAnLFIt's...NWDAFRun, andmodelId(s)、notifCorrId、notificationUri、mLEvent/mLEventFilter/tgtUeis consistent with local values at the time of subscription; ♪ Every time ♪AnLFIt's...NWDAFA match exists when accepting subscriptionsMTLFIt's...NWDAFRun, andmodelIdsandnotifCorrIdUnanimously.	no compromise	Holds	medium	yes
A04	**Distinct resource per creation.** At each timeMTLFIt's...NWDAFAccepts the creation of a registration at a time, corresponding to the different contentAnLFIt's...NWDAFRunregistrationIdSingle-shot; ♪ Every time ♪AnLFIt's...NWDAFAccept one subscription to create with a different contentMTLFIt's...NWDAFRunsubscriptionIdSingle-shot.	no compromise	Holds	low	no
A07	**consumerIdmatches the authenticated consumer.** At each timeMTLFIt's...NWDAFOn accepting sign-up/subscription, in requestconsumerIdand token subject,TLSAmong Client CertificatesNFExamplesIDUnanimously, and token assuer allowedNRF.	no compromise	Fails (1 attack)	medium	unclear
A08	**Notification belongs to an earlier subscription.** At each timeMTLFIt's...NWDAFOn receipt of the notification,notifCorrIdandnotificationUriIt's about a subscription that was previously created by the consumer.modelAccuInfos/anaFeedbacksMediummodelId_Other OrganisermodelIds, and requests the token scope to overwrite the operation.	no compromise	Holds	medium	yes
K01	**Long-term keys stay secret.** External attackers and malicious acts with their own evidenceNFNeither can be extrapolated.NRF. . . . . .NRFShare symmetric keys with producers, or otherNFIt's...TLSSession key.	no compromise	Holds	low	yes
K02	**Accuracy and feedback confidentiality.** In the notificationmodelAccuInfos、anaFeedbacksand subscription/registrationtgtUeCreate only the content of the subscriptionMTLFIt's...NWDAFAnd Ham.AnLFIt's...NWDAFVisible; Other legalNFWith external assailants, they cannot be obtained.	no compromise	Attack holds (stated in the documents)	medium	yes
D01	**Notification authenticity and integrity.** At each timeMTLFIt's...NWDAFExistence at receipt of notificationAnLFIt's...NWDAFIt's real, and...notifCorrId、modelAccuInfos/anaFeedbacksSame as when sent.	no compromise	Attack holds (stated in the documents)	medium	yes
Z01	**Every request carries a valid token.** At each timeMTLFor includesAnLFIt's...NWDAFEnabled to submit registration, subscription, unsubscribe, update or notification operationsNRFIssued, unexpired, Audience matches this instance, scope overwrites this operation, and the token object corresponds to the identity of the applicant.	no compromise	Holds	low	yes

ID	Property	Compromise assumed	Result	Value	In the documents
Z03	**Token subject bound to the caller.** Every time a token is used for a service request, the token subject corresponds to the identity of the applicant 's TLS certificate, the audience matches the requested producer example, the scope covers the requested operation, and consumerId of the requested body corresponds to the object.	no compromise	Fails (1 attack)	medium	unclear
Z05	**No access to otherNFresources.** Allowing control of a legitimateNFFields (including their own documents and tokens) cannot read or modify other fieldsNFof the United Nations and otherNFIt's...notificationUri(b) Delivery notice; They're under attack.TLSThe termination agent cannot read any other endpointTLSSession.	no compromise	Holds	medium	yes
U01	**Feedback bound to model and subscription.** At each timeMTLFIt's...NWDAFAccept ContentsanaFeedbacksIn notification, in feedbackmodelIds, the identity of the sender andnotifCorrIdConsistent with one previously established subscription.	no compromise	Holds	medium	yes
U02	**External vs compromised intermediary.** Check the integrity of the notification and the authorization of the subscription, respectively, under the configuration of external assailants and the configuration of the attacked intermediate entity, and compare the conclusions of both.	TLSTermination of agent orSCP(after subscriptions are established, before notification is delivered)	Attack holds (stated in the documents)	medium	yes
U03	**MaliciousNFacross resources.** In bad faith and legalityNFIf you have a valid token of your own, check if you can handle otherNFis registered, subscribed or notified and is attributed to the normative text or local strategy.	no compromise	Fails (1 attack)	medium	unclear
X01	**Notification replay protection.** Every notification accepted twice by the NWDAF containing MTLF was sent twice by the NWDAF containing AnLF: a replayed notification POST is not accepted a second time.	no compromise	Attack holds (stated in the documents)	low	yes

"Attack holds (stated in the documents)": the prover found the attack, and the documents themselves already state it. The property fails; this is a known result, not a finding (Appendix B).

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: Feedback bound to models, reporters, monitoring subscriptions	The feedback is bound to the model identifiers of the subscription and to the producer instance of that subscription, as far as the notification fields go: the feedback element has no separate binding beyond travelling under the same notifCorrId, and its events array is not modelled.	U01, A08, D01
Q2: A clear distinction is made between external manipulation and the result of the attack on the intermediate entity	The two configurations differ exactly as the sources say: from the network the notification cannot be read, altered or replayed, while a compromised TLS terminating proxy can read it, substitute the payload and deliver a notification twice, because the notification operation carries no token.	U02, K02, D01, X01

Question	Answer	Rests on
Q3: Malicious and lawfulNFWhether cross-resource operations constitute normative gaps or local tactical conditions only	The part that is a gap in the specification is the missing comparison of consumerId with the authenticated consumer, which lets an instance register under another instance's identity; the rest is local policy, since the API defines a single service-level scope and the identifiers of other instances' resources stay unknown to a malicious NF.	U03, A07, Z05, 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 an authenticated confidential channel under a static per-pair key; the handshake, the certificate chain and the key confirmation are not modelled.
- Replay of a request is not representable: the model's table of request nonces over-approximates, so A04 is answered only in its identifier reading.
- The access token is one signature over its claims with a validity tag instead of time; expiry and the absence of revocation are not exercised.
- Audience is modelled as the NF type only; instance-level audience and the slicing and NF Set claims are not modelled.
- consumerSetId, mLEvent, mLEventFilter and tgtUe are not modelled, and the notificationUri is treated as the subscriber's own endpoint.
- The notification payload is one opaque value; numerical correctness of the accuracy is outside the model, as the task states.
- Compromise of a party's stored keys and of the NRF is not modelled; the attackers are the network, a malicious NF with its own credentials, and a TLS terminating proxy on the AnLF-MTLF link.
- Only the API sub-flow: NRF authorization decisions, discovery, SCP forwarding rules and indirect communication without mutual TLS are outside the model.
- Availability, timing and the computational strength of the primitives are not asked.

What the host records:

- Proved without a control query (Appendix C): A03: a03_bind, A04: a04_reg_id, A04: a04_sub_id, A07: a07_tls, A08: a08_bind_net, A08: a08_bind_nf, Z01: z01_reg, Z01: z01_sub, Z03: a07_tls, Z05: a08_bind_nf, U01: a08_bind_net, U01: a08_bind_nf, U03: d01_nf, U03: z05_regid. 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.

A02: NF instance authentication of requests

Every time you've got it,MTLFIt's...NWDAFWhen accepting a request for registration/subscription/notification/feedback, there is a real running one thatMTLFExample is the content of the opposite endAnLFIt's...NWDAFRun (and vice versa) with an identity particle size fixed toNFExample.

Against the network alone, every registration, subscription and notification the consumer or the producer accepts corresponds to a real run of the NF instance that authenticated on that request, so an attacker with the network alone cannot produce an acceptance. The property holds at NF instance granularity, so a party that does hold its own credentials is treated as a real instance, which is what the service is for.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The central authentication guarantee of the service, proved for any number of sessions: not new, but the property the rest of the review rests on.
- Stated in the documents: yes. Where the documents say it: s0472, s0480, s0412
- Rests on: TLS is modelled as an authenticated confidential channel under a key private to the pair of instances; the handshake is not modelled; the token is a signature of the NRF over its claims; the disjunction in the query expresses the network-only attacker by excluding traces in which a malicious NF or the proxy acted

A03: Notification values match the subscription

♪ Every time ♪MTLFIt's...NWDAFAt receiving notification, there is a match contentAnLFIt's...NWDAFRun, andmodelId(s)、notifCorrId、notificationUri、mLEvent/mLEventFilter/tgtUeis consistent with local values at the time of subscription; ♪ Every time ♪ AnLFIt's...NWDAFA match exists when accepting subscriptionsMTLFIt's...NWDAFRun, andmodelIdsandnotifCorrIdUnanimously.

The notification the consumer accepts carries the notifCorrId and the model identifiers of the subscription that consumer created, so nobody on the network can attach a notification to the wrong subscription. Agreement on mLEvent, mLEventFilter and tgtUe is only a should in the source and was not modelled, so nothing is said about it.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It is the binding that keeps notifications attached to their subscription; the exclusion of the optional attributes is a real limit.
- Stated in the documents: yes. Where the documents say it: s0076, s0122, s0126, s0319
- Rests on: mLEvent, mLEventFilter and tgtUe are not modelled; the source states their agreement with a should (s0319); modelIds is one value, not an array; the control query for this property holds as well, so no failing variant exists

A04: Distinct resource per creation

♪ Every time ♪MTLFIt's...NWDAFAccepts the creation of a registration at a time, corresponding to the different contentAnLFIt's...NWDAFRunregistrationIdSingle-shot; ♪ Every time ♪AnLFIt's...NWDAFAccept one subscription to create with a different contentMTLFIt's...NWDAFRunsubscriptionIdSingle-shot.

The identifiers the producer and the consumer assign are injective: two acceptances carrying the same registrationId or subscriptionId are the same acceptance. The stronger reading, that one request is never accepted twice, fails in the model by a replayed request, which the TLS record layer prevents in the protocol and the model cannot express, so that failure is not offered as a protocol weakness.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value low: The proved part follows from resource creation semantics; the interesting part depends on replay protection the model cannot represent.
- Stated in the documents: no. Source text it rests on: s0049, s0050, s0051, s0085, s0087
- Rests on: replay protection is modelled by a table of request nonces which ProVerif over-approximates, so a repeated request is accepted in the model; the proved part is about the identifier, not about the request

A07: consumerId matches the authenticated consumer

♪ Every time ♪ MTLFIt's...NWDAFOn accepting sign-up/subscription, in requestconsumerIdand token subject,TLSAmong Client CertificatesNFExamplesIDUnanimously, and token assuer allowedNRF. . . . . .

The token subject is always the identity on the caller's TLS client certificate, so a stolen token cannot be used by another instance. The consumerId of the request body is not bound to that identity: an instance with a valid token of its own registers under another instance's consumerId and the producer stores it.

- Result: Fails (1 attack); answers what is asked: yes; usable
- Value medium: A malicious legal NF obtains a false attribution of monitoring to another instance without breaking any cryptographic step, and the change that removes it is small and was checked.
- Stated in the documents: unclear. Source text it rests on: s0040, s0041, s0042, s0480
- Rests on: the source does not require the comparison, so the attack is on the specification as written; audience is modelled as the NF type only; expiry is a validity tag; no expired token exists in the model

A08: Notification belongs to an earlier subscription

♪ Every time ♪ MTLFIt's...NWDAFOn receipt of the notification,notifCorrIdandnotificationUriIt's about a subscription that was previously created by the consumer.modelAccuInfos/anaFeedbacksMediummodelId_Other OrganisermodelIds, and requests the token scope to overwrite the operation.

An accepted notification is bound to a subscription the consumer created earlier: its notifCorrId and model identifiers are those of that subscription, against the network, against a malicious NF and against the compromised proxy. The notification operation carries no access token, so no scope is asked of it; what that leaves open is payload integrity, asked separately.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It shows which part of the notification binding survives a compromised intermediary and which does not.
- Stated in the documents: yes. Where the documents say it: s0076, s0122, s0126, s0347, s0412
- Rests on: the notificationUri is modelled as the subscriber's own endpoint; redirection is not modelled; the payload content is not bound by this property, only the identifiers

K01: Long-term keys stay secret

External assailants and self-inflicted maliceNFNeither can be extrapolated.NRF. . . . . .NRFShare symmetric keys with producers, or otherNFIt's...TLSSession key.

Neither the NRF signing key, nor the key shared between the NRF and a producer, nor the TLS keys of an endpoint pair are derivable by the network attacker or by a malicious NF with its own credentials. A proxy compromised on one link does not expose the sessions of the links it does not terminate.

- Result: Holds; answers what is asked: yes; usable
- Value low: It follows from modelling keys as unguessable; its worth is the isolation part.
- Stated in the documents: yes. Where the documents say it: s0405, s0439, s0475
- Rests on: keys are idealised as unguessable private values; reading a party's key store is not modelled

K02: Accuracy and feedback confidentiality

In the notificationmodelAccuInfos、anaFeedbacksand subscription/registrationtgtUeCreate only the content of the subscriptionMTLFIt's...NWDAFAnd Ham.AnLFIt's...NWDAFVisible; Other legalNFWith external assailants, they cannot be obtained.

The accuracy information and the analytics feedback are visible only to the two NWDAF instances of the subscription against the network and against a malicious NF. An attacker who terminates the TLS of that link reads them, which the sources say must be accepted because parties have to trust the SCP.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value medium: It draws the confidentiality boundary of monitoring data where the sources draw it.
- Stated in the documents: yes. Where the documents say it: s0125, s0126, s0472
- Rests on: one payload value stands for modelAccuInfos and anaFeedbacks; tgtUe is not carried in the model's messages

D01: Notification authenticity and integrity

♪ Every time ♪ MTLFIt's...NWDAFExistence at receipt of notificationAnLFIt's...NWDAFIt's real, and...notifCorrId、modelAccuInfos/anaFeedbacksSame as when sent.

Against the network and against a malicious NF, a notification the consumer accepts was sent by the producer of that subscription with that correlation identifier and that payload, and the notifCorrId keeps a replayed notification from being accepted twice. Against an attacker who terminates the TLS of the pair, the consumer accepts a notification the producer never sent, because the notification has no token and no signature of its own.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value medium: The positive side is expected; the negative side gives the exact conditions under which monitoring data can be forged.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2921, s0122, s0123, s0126, s0412
- Rests on: a proxy that terminates TLS is modelled by giving the attacker both directions of that link; the notification carries no token, which the source states

Z01: Every request carries a valid token

♪ Every time ♪MTLFor includesAnLFIt's...NWDAFEnabled to submit registration, subscription, unsubscribe, update or notification operationsNRFIssued, unexpired, Audience matches this instance, scope overwrites this operation, and the token object corresponds to the identity of the applicant.

Every accepted registration and subscription carries a token the NRF issued to the instance that presented it, with the audience of a NWDAF and the single service scope, and the producer checks issuer, signature, subject and audience. The expiry check exists in the model but no expired token can be constructed in it, and the source states that revocation is not supported.

- Result: Holds; answers what is asked: partly; usable with caveats
- Value low: The token checks are as specified; the two parts that would be interesting, expiry and scope granularity, are the parts the model cannot exercise.
- Stated in the documents: yes. Where the documents say it: s0344, s0345, s0450, s0474, s0485, s0487, s0412
- Rests on: audience as NF type, which the source permits; expiry is a validity tag and is not exercised; one service-level scope means the scope check never distinguishes operations

Z03: Token subject bound to the caller

Every time a token is used for a service request, the token subject corresponds to the identity of the applicant 's TLS certificate, the audience matches the requested producer example, the scope covers the requested operation, and consumerId of the requested body corresponds to the object.

The producer is assured that the token was issued to the party presenting it: the subject matches the TLS client certificate, the audience is its own NF type and the scope covers the service. The body consumerId is not covered by that binding, so the identity a registration records can be chosen by the caller, as in A07.

- Result: Fails (1 attack); answers what is asked: yes; usable
- Value medium: The positive part confirms the subject binding the source requires; the negative part is the substantive gap of the review.
- Stated in the documents: unclear. Source text it rests on: s0474, s0480, s0482, s0485, s0487
- Rests on: audience as NF type, which the source allows; the consumerSetId alternative is not modelled

Z05: No access to other NF resources

Allowing control of a legitimate one.NFFields (including their own documents and tokens) cannot read or modify other fieldsNFof the United Nations and otherNFIt's...notificationUri(b) Delivery notice; They're under attack.TLSThe termination agent cannot read any other endpointTLSSession.

A malicious NF cannot obtain another NF's registration or subscription identifier and cannot learn the correlation identifier of another NF's subscription, so a notification reaches only the consumer that created the subscription; those identifiers are the only thing standing between instances. A proxy compromised on one link does not expose the other links. What is not assured is authorization at resource level: with one service-level scope every allowed instance may invoke every operation.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It shows that the single service-level scope is not exploitable through a guessable handle in the model, and marks where a resource-level scope would be needed.
- Stated in the documents: yes. Where the documents say it: s0347, s0397, s0485

- Rests on: resource-level authorization is not modelled; protection of another NF's resource rests on its identifier being unguessable; the notificationUri is the subscriber's own endpoint

U01: Feedback bound to model and subscription

♪ Every time ♪MTLFIt's...NWDAFAccept ContentsanaFeedbacksIn notification, in feedbackmodelIds, the identity of the sender andnotifCorrIdConsistent with one previously established subscription.

The analytics feedback in an accepted notification refers to the model identifiers of the consumer's subscription and comes from the producer of that subscription, against the network and against a malicious NF. The events array of the feedback is not modelled, so its binding to the analytics identifiers is not covered.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: It answers the user's question: feedback cannot be attached to another model or arrive from another instance.
- Stated in the documents: yes. Where the documents say it: s0122, s0126, s0076, s0328
- Rests on: one payload value stands for anaFeedbacks; its events element is not modelled

U02: External vs compromised intermediary

Check the integrity of the notification and the authorization of the subscription, respectively, under the configuration of the external attackers and the configuration of the captured intermediate entity, and compare the conclusions of both.

The two configurations come out differently. With the network alone the notification cannot be read, altered or forged. With a compromised TLS terminating proxy the payload can be read, a notification the producer never sent can be delivered, and a recorded one can be delivered again. The difference is exactly the trust the sources place in the SCP.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value medium: It answers the question with the exact boundary and reproduces a consequence the sources state, so it is a confirmed known result rather than a new weakness.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2921, ts-33-501:2873, s0412, s0122, s0126
- Rests on: the proxy holds both directions of the link it terminates, which is what terminating TLS means; the proxy is compromised after the subscription and before the notification

U03: Malicious NF across resources

It's illegal to be malicious.NFIf you have a valid token of your own, check if you can handle otherNFis registered, subscribed or notified and is attributed to the normative text or local strategy.

Part of the answer is a gap in the text and part is local policy. The gap: any instance with a valid token can register under another instance's consumerId, because the body attribute is never compared with the authenticated identity. The policy: with one service-level scope, which instance may invoke which operation is decided by NRF authorization and local configuration, and the identifiers of other instances' resources stay unknown to a malicious NF, so it cannot operate them in the model.

- Result: Fails (1 attack); answers what is asked: yes; usable
- Value medium: It separates what an editor must fix in the text from what an operator must configure.
- Stated in the documents: unclear. Source text it rests on: s0347, s0394, s0397, s0485

- Rests on: resource-level authorization is left to local policy and is not modelled; identifiers are fresh unguessable values

X01: Notification replay protection

Every notification accepted twice by the NWDAF containing MTLF was sent twice by the NWDAF containing AnLF: a replayed notification POST is not accepted a second time.

Against the network, a notification accepted twice was sent twice: the correlation identifier and the model identifier tie each acceptance to the subscription, and a replayed record does not pass. Against an attacker who terminates the TLS of the pair, a second notification can be delivered for one the producer sent, since nothing in the notification distinguishes a re-delivery made by the proxy.

- Result: Attack holds (stated in the documents); answers what is asked: yes; usable
- Value low: It is a consequence of D01: the same protection and the same failure, so it adds the replay reading rather than a new result.
- Stated in the documents: yes. Where the documents say it: ts-33-501:2921, s0122, s0123, s0412
- Rests on: replay protection of TLS is modelled by a table of request nonces, so the property is asked through the notifCorrId binding instead; the notification carries no token

Appendix B: Known results reproduced

A property that fails as the documents already state, or that they set aside, is not a finding. What follows is of that kind: the analysis reproduces it on the model, which gives the exact conditions.

B.1 A compromised SCP rewrites the accuracy notifications it carries

Kind	Weakness of the protocol as specified
Stated in the documents	ts-33-501:2921, ts-33-501:2873
Value	medium
Properties	D01, K02, U02, X01 (section 3)
Shown by	ProVerif attack queries `d01_ctl`, `k02_ctl`, `x01_ctl`

What happens. A TLS terminating proxy on the link between the two NWDAF instances reads the accuracy and feedback payloads, and re-encrypts to the consumer a notification with the notifCorrId and modelId of its subscription but a payload the producer never sent. The consumer accepts it and answers 204 No Content, because the notification operation carries no access token and its only protection is the TLS session the proxy terminates. A recorded notification can also be delivered a second time.

What the attacker needs. Control of the SCP or TLS terminating proxy on that link, which means both directions of the TLS sessions it terminates, obtained after the subscription and before the notification. The network attacker alone cannot do this.

What the attacker gains. The consumer's view of how well a model performs, and the feedback it acts on, are under the proxy's control, and one notification can be delivered twice.

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
A02	proved		a02_reg	checked		3 proved
A03	proved		a03_bind			1 proved
A04	proved		a04_reg_id		a04_reg (model_artifact); a04_sub (model_artifact)	2 proved, 2 attack
A07	proved		a07_tls	1 of 3	a07_body_nf (holds_in_source)	3 proved, 1 attack
A08	proved		a08_bind_net			3 proved
K01	proved		k01_nrfkey	checked		5 proved
K02	proved		k02_net	checked	k02_ctl (holds_in_source)	2 proved, 1 attack
D01	proved		d01_net	checked	d01_ctl (holds_in_source)	2 proved, 1 attack
Z01	proved		z01_reg			2 proved
Z03	proved		a07_tls	1 of 3	a07_body_nf (holds_in_source)	3 proved, 1 attack
Z05	proved		z05_regid	4 of 5		5 proved
U01	proved		a08_bind_net	2 of 4		4 proved
U02	proved		d01_net	checked	d01_ctl (holds_in_source); k02_ctl (holds_in_source)	2 proved, 2 attack
U03	proved		d01_nf	3 of 5	a07_body_nf (holds_in_source)	5 proved, 1 attack
X01	proved		x01_net	checked	x01_ctl (holds_in_source)	1 proved, 1 attack

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
A02	Every acceptance of a registration by the NWDAF containing MTLF corresponds to a real run of the NF instance that authenticated on that request; the attacker has the network only.	`network`	proved	a02_reg	checked
A02	Every acceptance of a subscription by the NWDAF containing AnLF corresponds to a real run of the NF instance that authenticated on that request; the attacker has the network only.	`network`	proved	a02_sub	checked
A02	Every acceptance of a notification by the NWDAF containing MTLF corresponds to a real run of the NWDAF containing AnLF that authenticated; the attacker has the network only.	`network`	proved	a02_not	checked
A03	The notifCorrId and the modelIds of an accepted notification are those of the subscription this consumer created earlier; the attacker has the network only.	`network`	proved	a03_bind	none
A04	Each accepted registration creation corresponds to a distinct registration request run, so no two acceptances come from one request; the attacker has the network only.	`network`	attack (model_artifact)	a04_reg	

target	claim	attacker	answer	query	control
A04	Each accepted subscription creation corresponds to a distinct subscription request run, so no two acceptances come from one request; the attacker has the network only.	network	attack (model_artifact)	a04_sub	
A04	The registrationId is injective: two acceptances that carry the same registrationId are the same acceptance, so no two registration requests receive the same identifier; the attacker has the network only.	network	proved	a04_reg_id	none
A04	The subscriptionId is injective: two acceptances that carry the same subscriptionId are the same acceptance, so no two subscription requests receive the same identifier; the attacker has the network only.	network	proved	a04_sub_id	none
A07	The token subject is the TLS client certificate identity of the caller; the attacker has the network only.	network	proved	a07_tls	none
A07	The token subject is the TLS client certificate identity of the caller; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	a07_tls	none
A07	The consumerId in the request body is the authenticated consumer of the token; the attacker has its own NF credentials and a valid token.	malicious_nf	attack (holds_in_source)	a07_body_nf (the control of a07_body_net)	
A07	The consumerId in the request body is the authenticated consumer of the token; the attacker has the network only.	network	proved	a07_body_net	checked
A08	The notifCorrId and modelIds of an accepted notification belong to a subscription this MTLF created earlier, and no token scope is asked because OAuth2 does not apply to the notification operation; the attacker has the network only.	network	proved	a08_bind_net	none
A08	The notifCorrId and modelIds of an accepted notification belong to a subscription this MTLF created earlier, and no token scope is asked because OAuth2 does not apply to the notification operation; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	a08_bind_nf	none
A08	The notifCorrId and modelIds of an accepted notification belong to a subscription this MTLF created earlier, and no token scope is asked because OAuth2 does not apply to the notification operation; the attacker holds both directions of the TLS links it terminates on that pair.	scp_compromised	proved	a08_bind_nf	none
K01	The NRF token signing key is not derivable; the attacker has the network only.	network	proved	k01_nrfkey	checked
K01	The NRF token signing key is not derivable; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	k01_nrfkey	checked
K01	The TLS session keys of an endpoint pair are not derivable by a third party; the attacker has the network only.	network	proved	k01_tlskey	checked
K01	The TLS session keys of an endpoint pair are not derivable by a third party; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	k01_tlskey2	checked
K01	The key shared between the NRF and the producer is not derivable; the attacker has the network only.	network	proved	k01_tlskey2	checked
K02	The accuracy and feedback payload is not learned by an attacker who terminates the TLS of the pair.	scp_compromised	attack (holds_in_source)	k02_ctl (the control of k02_net)	

target	claim	attacker	answer	query	control
K02	The accuracy and feedback payload of a notification is not learned by the network attacker.	network	proved	k02_net	checked
K02	The accuracy and feedback payload of a notification is not learned by a malicious NF holding its own credentials.	malicious_nf	proved	k02_nf	checked
D01	An accepted notification was sent by the NWDAF containing AnLF of the subscription, with that notifCorrId and that payload; the attacker has the network only.	network	proved	d01_net	checked
D01	An accepted notification was sent by the NWDAF containing AnLF of the subscription, with that notifCorrId and that payload, even against an attacker who terminates the TLS of the pair.	scp_compromised	attack (holds_in_source)	d01_ctl (the control of d01_net)	
D01	An accepted notification was sent by the NWDAF containing AnLF of the subscription, with that notifCorrId and that payload; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	d01_nf	checked
Z01	Every accepted registration carries a token issued by the NRF to the NF instance that presented it, with the audience of this producer, the single service scope and a valid (unexpired) validity; the attacker has the network only.	network	proved	z01_reg	none
Z01	Every accepted subscription carries a token issued by the NRF to the NF instance that presented it, with the audience of this producer, the single service scope and a valid (unexpired) validity; the attacker has the network only.	network	proved	z01_sub	none
Z03	The token subject is the identity of the caller on its TLS client certificate; the attacker has the network only.	network	proved	a07_tls	none
Z03	The token subject is the identity of the caller on its TLS client certificate; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	a07_tls	none
Z03	The consumerId in the request body is the subject of the token that authorized the request; the attacker has its own NF credentials and a valid token.	malicious_nf	attack (holds_in_source)	a07_body_nf (the control of a07_body_net)	
Z03	The consumerId in the request body is the subject of the token that authorized the request; the attacker has the network only.	network	proved	a07_body_net	checked
Z05	A notification is delivered only to the endpoint of the consumer that created the subscription, never to another NF's notificationUri; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	a08_bind_nf	none
Z05	A malicious NF cannot learn another NF's registrationId and so cannot operate that registration; the attacker has the network only.	network	proved	z05_regid	checked
Z05	A malicious NF cannot learn another NF's registrationId and so cannot operate that registration; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	z05_regid	checked
Z05	A malicious NF cannot learn the notifCorrId of another NF's subscription and so cannot inject a notification into it; the attacker has its own NF credentials and a valid token.	malicious_nf	proved	z05_corr	checked
Z05	A proxy compromised on the AnLF-MTLF link does not expose the TLS sessions of other endpoints, here the links to the NRF; the attacker holds both directions of the TLS links it terminates on that pair.	scp_compromised	proved	k01_tlskey	checked

target	claim	attacker	answer	query	control
U01	The modelIds of the analytics feedback in an accepted notification belong to the subscription of this consumer; the attacker has the network only.	`network`	proved	a08_bind_net	none
U01	The modelIds of the analytics feedback in an accepted notification belong to the subscription of this consumer; the attacker has its own NF credentials and a valid token.	`malicious_nf`	proved	a08_bind_nf	none
U01	The feedback in an accepted notification comes from the NWDAF containing AnLF of the subscription and from no other sender; the attacker has the network only.	`network`	proved	d01_net	checked
U01	The feedback in an accepted notification comes from the NWDAF containing AnLF of the subscription and from no other sender; the attacker has its own NF credentials and a valid token.	`malicious_nf`	proved	d01_nf	checked
U02	Against an attacker who terminates the TLS of the pair the notification content can be modified: the consumer accepts a notification the producer never sent.	`scp_compromised`	attack (holds_in_source)	d01_ctl (the control of d01_net)	
U02	Against an attacker who terminates the TLS of the pair the notification payload can be read.	`scp_compromised`	attack (holds_in_source)	k02_ctl (the control of k02_net)	
U02	Against the external network attacker the notification content cannot be modified or forged: an accepted notification was sent by the producer with that payload.	`network`	proved	d01_net	checked
U02	Against the external network attacker the notification payload cannot be read.	`network`	proved	k02_net	checked
U03	A malicious NF cannot read or forge the notifications of another NF's subscription; the attacker has its own NF credentials and a valid token.	`malicious_nf`	proved	d01_nf	none
U03	A malicious NF cannot obtain another NF's registration or subscription handle; the attacker has its own NF credentials and a valid token.	`malicious_nf`	proved	z05_regid	none
U03	A malicious NF holding its own valid token can operate another NF's resource: it can register with that NF's consumerId.	`malicious_nf`	attack (holds_in_source)	a07_body_nf (the control of a07_body_net)	
U03	With the network alone no request is accepted whose consumerId is not the authenticated consumer, so the network attacker cannot make an NF operate another NF's resource.	`network`	proved	a07_body_net	checked
U03	With the network alone no notification of another NF's subscription can be forged or read.	`network`	proved	d01_net	checked
U03	With the network alone no registration handle of another NF can be obtained, so another NF's registration cannot be operated.	`network`	proved	z05_regid	checked
X01	A notification accepted twice by the consumer was sent twice by the producer, even against an attacker who terminates the TLS of the pair.	`scp_compromised`	attack (holds_in_source)	x01_ctl (the control of x01_net)	
X01	A notification accepted twice by the consumer was sent twice by the producer: the notifCorrId binding rejects a replayed notification; the attacker has the network only.	`network`	proved	x01_net	checked

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-20261115T083926, 2026-11-15. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 62, prover 28 s, wall 19 min, cost 0.179 USD.

Context: the host removed stale parts from the conversation 1 time(s), 465071 characters in all (earlier reasoning, and what showed the model before its latest change). The trajectory file keeps everything.

ProVerif: 229 proofs run, 228 decided, 1 without an answer, 21 s, 75 answers recorded.

What was analysed

Protocol	Review ML model monitoring with SBA isolation
Version	19.8.0
Analysed	First round of reviewNnwdaf_MLModelMonitor：Register/Subscribe/Notify/accuracy feedback；Isolate strictly by endpoint.TLS, identity bound to token subject; External attacker, malicious consumer, captured.TLSThe termination agent must be divided into clear, independent configurations. Precision is opaque data, which does not prove the value correct.
Not analysed	Symbolic models do not support the calculation of algorithms as safe or as resistant to quantum security; The trajectory of the attack is established only under clear assumptions and is not directly equivalent to a normative loophole. Each nature must be subject to accessibility/proportional controls. FixedRel-19。 Distinguishing between an outside cyber attacker and an ill-intentioned and legitimate person holding his own evidence.NF； TLSAuthenticate encryption modeling by endpoint, validNFCan not open messageNFIt's...TLSSession.OAuth2Subject/udience/scope/expiry andTLS(a) The identity is bound in accordance with the attached baseline; Service scope does not automatically equal ownership of resources. Single-modules cannot simulate real deletions, and the life cycle of a limited state must be clearly defined. Targets are specified.APISubprocess, not complete 5G secure.
Source text	661 numbered passages (251766 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	4.7	Nnwdaf_MLModelMonitor Service
main	4.7.1	Service Description
main	4.7.1.1	Overview
main	4.7.1.2	Service Architecture
main	4.7.1.3	Network Functions
main	4.7.1.3.1	Network Data Analytics Function (NWDAF)
main	4.7.1.3.2	NF Service Consumers
main	4.7.2	Service Operations
main	4.7.2.1	Introduction
main	4.7.2.2	Nnwdaf_MLModelMonitor_Register service operation
main	4.7.2.2.1	General

Document	Section	Title
main	4.7.2.2.2	Registering the monitoring of the analytics accuracy of an ML Model
main	4.7.2.3	Nnwdaf_MLModelMonitor_Deregister service operation
main	4.7.2.3.1	General
main	4.7.2.3.2	Deregistering the monitoring of the analytics accuracy of an ML Model
main	4.7.2.4	Nnwdaf_MLModelMonitor_Subscribe service operation
main	4.7.2.4.1	General
main	4.7.2.4.2	Subscription for monitoring notifications
main	4.7.2.4.3	Update of subscription for monitoring notifications
main	4.7.2.5	Nnwdaf_MLModelMonitor_Unsubscribe service operation
main	4.7.2.5.1	General
main	4.7.2.5.2	Unsubscribe from monitoring notifications
main	4.7.2.6	Nnwdaf_MLModelMonitor_Notify service operation
main	4.7.2.6.1	General
main	4.7.2.6.2	Notification about subscribed event
main	5.6	Nnwdaf_MLModelMonitor Service API
main	5.6.1	Introduction
main	5.6.2	Usage of HTTP
main	5.6.2.1	General
main	5.6.2.2	HTTP standard headers
main	5.6.2.2.1	General
main	5.6.2.2.2	Content type
main	5.6.2.3	HTTP custom headers
main	5.6.3	Resources
main	5.6.3.1	Resource Structure
main	5.6.3.2	Resource: NWDAF ML model monitoring registrations
main	5.6.3.2.1	Description
main	5.6.3.2.2	Resource Definition
main	5.6.3.2.3	Resource Standard Methods
main	5.6.3.2.3.1	POST
main	5.6.3.2.4	Resource Custom Operations
main	5.6.3.3	Resource: Individual NWDAF ML model monitoring registration
main	5.6.3.3.1	Description
main	5.6.3.3.2	Resource definition
main	5.6.3.3.3	Resource Standard Methods
main	5.6.3.3.3.1	DELETE
main	5.6.3.3.4	Resource Custom Operations

Document	Section	Title
main	5.6.3.4	Resource: NWDAF ML model monitoring Subscriptions
main	5.6.3.4.1	Description
main	5.6.3.4.2	Resource Definition
main	5.6.3.4.3	Resource Standard Methods
main	5.6.3.4.3.1	POST
main	5.6.3.4.4	Resource Custom Operations
main	5.6.3.5	Resource: Individual NWDAF ML model monitoring Subscription
main	5.6.3.5.1	Description
main	5.6.3.5.2	Resource definition
main	5.6.3.5.3	Resource Standard Methods
main	5.6.3.5.3.1	PUT
main	5.6.3.5.3.2	DELETE
main	5.6.3.5.4	Resource Custom Operations
main	5.6.4	Custom Operations without associated resources
main	5.6.5	Notifications
main	5.6.5.1	General
main	5.6.5.2	Event Notification
main	5.6.5.2.1	Description
main	5.6.5.2.2	Operation Definition
main	5.6.6	Data Model
main	5.6.6.1	General
main	5.6.6.2	Structured data types
main	5.6.6.2.1	Introduction
main	5.6.6.2.2	Type MLModelMonitorReg
main	5.6.6.2.3	Type MLModelMonitorSub
main	5.6.6.2.4	Type MLModelMonitorNotify
main	5.6.6.2.5	Type MLModelAccuracyInfo
main	5.6.6.2.6	Type AnalyticsFeedback
main	5.6.7	Error handling
main	5.6.7.1	General
main	5.6.7.2	Protocol Errors
main	5.6.7.3	Application Errors
main	5.6.8	Feature negotiation
main	5.6.9	Security
ts-29-501	4.10	Scopes definition for OAuth2.0 access token
ts-33-501	13.4	Authorization of NF service access

Document	Section	Title
ts-33-501	13.4.1	OAuth 2.0 based authorization of Network Function service access
ts-33-501	13.4.1.0	General
ts-33-501	13.4.1.1	Service access authorization within the PLMN
ts-33-501	13.4.1.1.1	OAuth 2.0 roles
ts-33-501	13.4.1.1.2	Service Request Process
ts-33-501	13.4.1.1.3	Access token requests in deployments with several NRFs
ts-33-501	13.4.1.1A	Service access authorization in interconnect scenarios
ts-33-501	13.4.1.2	Service access authorization in roaming scenarios
ts-33-501	13.4.1.2.1	OAuth 2.0 roles
ts-33-501	13.4.1.2.2	Service Request Process
ts-33-501	13.4.1.3	Service access authorization in indirect communication scenarios
ts-33-501	13.4.1.3.1	Authorization for indirect communication without delegated discovery procedure
ts-33-501	13.4.1.3.1.1	With mutual authentication between NF Service Consumer and NRF at the transport layer
ts-33-501	13.4.1.3.1.2	Without mutual authentication between NF and NRF at the transport layer
ts-33-501	13.4.1.3.2	Authorization for indirect communication with delegated discovery procedure
ts-33-501	13.4.1.4	Service access authorization in inter NF mobility scenario
ts-33-501	13.4.1.5	Service access authorization in interconnect and roaming scenarios with indirect communication
ts-33-501	13.4.1.5.1	General
ts-33-501	13.4.1.5.2	Authorization with NF selection at source network
ts-33-501	13.4.1.5.2.1	General
ts-33-501	13.4.1.5.2.2	Authorization for indirect communication with or without delegated discovery procedure
ts-33-501	13.4.1.5.3	Authorization with NF selection at target network
ts-33-501	13.4.1.5.3.1	General
ts-33-501	13.4.1.5.3.2	Authorization for indirect communication without delegated discovery procedure
ts-33-501	13.4.1.5.3.3	Authorization for indirect communication with delegated discovery procedure

The process of the review as the source step listed it in the documents:Nnwdaf_MLModelMonitor：Register/Subscribe/Notify/accuracy feedback； Isolate strictly by endpoint.TLS, identity bound to token subject; External attacker, malicious consumer, captured.TLSThe termination agent must be divided into clear, independent configurations. Precision is opaque data, which does not prove the value correct.

The properties against the documents

Before a query was written, 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 exist in the documents.

ID	Name	About	What the documents say	Where
A02	NF instance authenticati on of requests	The NWDAF containing MTLF is assured that every registration, subscription or notification it accepts comes from a real run of the NF instance named in the request; the NWDAF containing AnLF is assured likewise for subscriptions.	Requests are exchanged over mutually authenticated TLS between NF Service Consumer and Producer (TS 33.501 13.4.1.1.2, s0472; TS 33.501 5.9.2.1), and the token subject must match the NF Instance ID in the TLS client certificate (s0480).	s0004, s0022, s0023, s0472, s0480
A03	Notification values match the subscription	The NWDAF containing MTLF is assured that notifCorrId, modelIds and the payload of an accepted notification match the subscription it created; the NWDAF containing AnLF is assured the subscription matches its registration.	The notification must carry the notifCorrId and the model IDs of the subscription (s0076, s0122, s0126); the values of mLEvent, mLEventFilter and tgtUe should match the registration (s0319).	s0076, s0122, s0126, s0319
A04	Distinct resource per creation	The NWDAF containing MTLF assigns a distinct registrationId to each accepted registration, and the NWDAF containing AnLF a distinct subscriptionId to each accepted subscription; two acceptances are never two deliveries of one request.	The producer creates a new registration and assigns a registrationId (s0049, s0050, s0051) or a new subscription and a subscriptionId (s0085, s0086, s0087).	s0049, s0050, s0051, s0085, s0086, s0087
A07	consumerId matches the authenticate d consumer	The NWDAF containing MTLF is assured that the consumerId or consumerSetId in the request body identifies the NF instance that authenticated, i.e. the token subject and the TLS client certificate identity.	The body carries the NF instance ID or NF set ID of the consumer (s0040, s0041, s0042); the producer checks the token subject against the TLS client certificate (s0480) and the issuer, audience and scope (s0474, s0482, s0485).	s0040, s0041, s0042, s0480
A08	Notification belongs to an earlier subscription	The NWDAF containing MTLF is assured that an accepted notification is bound to a subscription it created earlier with this consumer: the notifCorrId, the notificationUri and the modelIds belong to that subscription, and the operation is covered by the token scope.	Subscription carries modelIds, notificationUri and notifCorrId (s0073, s0075, s0076); the notification echoes notifCorrId and carries modelAccuInfos/anaFeedbacks for the subscribed models (s0122, s0123, s0126).	s0073, s0075, s0076, s0122, s0126
K01	Long-term keys stay secret	The NRF token signing key, the key shared between the NRF and the producer, and the TLS keys of other endpoints are kept from the network attacker and from a malicious NF holding its own credentials.	Tokens are secured with a digital signature or MAC of the NRF (s0405, s0475); the NRF and the producer share credentials (s0439); NF communication is protected by mutually authenticated TLS (s0472).	s0405, s0439, s0475, s0472
K02	Accuracy and feedback confidential ity	The monitored accuracy information, the analytics feedback and the target UE information are visible only to the two NWDAF instances of the subscription, not to other NFs or the network attacker.	modelAccuInfos and anaFeedbacks are carried in the notification (s0125, s0126) and tgtUe in the registration and subscription (s0047, s0083); NF communication is confidentiality protected by TLS (s0472, TS 33.501 X.6).	s0125, s0126, s0047, s0083, s0472
D01	Notification authenticity and integrity	The NWDAF containing MTLF is assured that an accepted notification was sent by the NWDAF containing AnLF of its subscription, with the notifCorrId and the payload it sent.	The notification is an HTTP POST to the notificationUri carrying notifCorrId and the accuracy/feedback payload (s0122, s0123, s0126); OAuth2 does not apply to the notification operation (s0412), so authenticity rests on the transport session alone (s0472).	s0122, s0123, s0126, s0412, s0472
Z01	Every request carries a valid token	The NWDAF containing MTLF and the NWDAF containing AnLF are assured that each service request they accept carries a token issued by an allowed NRF, unexpired, with an audience matching this producer and a scope covering the operation.	OAuth2 with the client credentials grant may authorize the API (s0344, s0345) and the producer verifies issuer, signature, audience, scope and expiry (s0474, s0475, s0482, s0485, s0487); the notification operation is outside OAuth2 (s0412).	s0344, s0345, s0450, s0474, s0485, s0487, s0412

ID	Name	About	What the documents say	Where
Z03	Token subject bound to the caller	The producer is assured that the token it accepts was issued to the party presenting it: the subject matches the TLS client certificate identity, the audience this producer and the scope this operation, and the body consumerId matches the subject.	The producer checks the issuer (s0474), the subject against the TLS client certificate in the direct communication case (s0480), the audience (s0482), the scope (s0485) and the expiry (s0487).	s0474, s0480, s0482, s0485, s0487
Z05	No access to other NF resources	A malicious NF holding its own credentials and token cannot read or change the registrations and subscriptions of other NFs, cannot post notifications to another NF's notificationUri, and a TLS terminating proxy compromised on one link does not expose the TLS sessions of other endpoints.	The API defines a single service-level scope for the whole service (s0347); TS 29.501 warns that a service-level scope alone grants all operations to all allowed consumers (s0397); parties must trust the SCP that handles their messages (s2921).	s0347, s0397, s0485
U01	Feedback bound to model and subscription	The NWDAF containing MTLF is assured that the analytics feedback in a notification (anaFeedbacks: modelIds, events) comes from the NWDAF containing AnLF of its subscription and refers to the models of that subscription.	anaFeedbacks carries the analytics IDs and modelIds used for actions (s0328) and travels in the notification with notifCorrId (s0122, s0126); the notification echoes the notifCorrId of the subscription (s0076).	s0122, s0126, s0076, s0328
U02	External vs compromised intermediary	Notification integrity and subscription binding are answered separately for the network attacker and for an attacker who holds the plaintext of a TLS terminating proxy (SCP) on the AnLF-MTLF link.	OAuth2 does not apply to the notification operation (s0412) and there is no mutual authentication between consumer and producer when an SCP is used (s2873); all parties communicating through the SCP must trust it (s2921).	s0412, s0472
U03	Malicious NF across resources	Whether a malicious legal NF, holding its own valid token, can operate the registrations, subscriptions or notifications of other NFs, and whether that gap is in the specification text or left to local policy.	The API defines only the service-level scope (s0347) while resource and operation level scopes are optional and left to the API group (s0394); the producer checks only that the scope matches the operation (s0485).	s0347, s0394, s0397, s0485
X01	Notification replay protection	the NWDAF containing MTLF is assured that a repeated notification it accepts is a new notification of its producer, not a replay by an intermediary or by the network.	NF service based discovery, registration and request exchanges shall support confidentiality, integrity and replay protection (TS 33.501 5.9.2.1, s2865); the token is protected with JWS (s0405); HTTP/2 over TLS is used (s0149).	s0405, s0122

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: 35. Not taken: 363. 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	they describe the scope of the RFC's security considerations and point to a separate threat model document; they state no property of this protocol
P003 to P004	rfc6749:2790-2809	Z01, Z03	
P005 to P006	rfc6749:2811-2823	Outside the scope	they are about authorisation-code and redirection-URI based client authentication; this API uses the client credentials grant only (s0393)

Paragraphs	Where	Taken up as	Why not
P007 to P010	rfc6749:2827-2850	Z03, A07	
P011	rfc6749:2859-2865	K01, Z01	
P012	rfc6749:2867-2868	Outside the scope	it concerns the implicit grant, which this API does not use
P013	rfc6749:2870-2872	K01, Z01	
P014	rfc6749:2874-2877	Z03	
P015	rfc6749:2879-2881	Z03, A07	
P016 to P021	rfc6749:2885-2917	Outside the scope	they concern refresh tokens; the client credentials grant used here issues none
P022 to P029	rfc6749:2921-2980	Outside the scope	they concern authorisation codes and redirection URIs, which this API does not use
P030 to P033	rfc6749:2984-3003	Outside the scope	they concern the resource owner password credentials grant, which this API does not use
P034 to P035	rfc6749:3013-3019	K02, D01, Z05	
P036	rfc6749:3023-3028	A02	
P037 to P039	rfc6749:3032-3041	Z01	
P040 to P042	rfc6749:3045-3066	Outside the scope	they concern phishing of resource owners; there is no resource owner in the client credentials grant
P043 to P048	rfc6749:3070-3110	Outside the scope	they concern cross-site request forgery in browser-based clients; NF clients are not browsers
P049 to P050	rfc6749:3119-3138	Outside the scope	clickjacking concerns user agents, which this API does not involve
P051 to P052	rfc6749:3142-3150	No goal stated	advice on input validation in the authorisation server's error handling; it states no property that can be checked about this API
P053 to P054	rfc6749:3154-3167	Outside the scope	they concern open redirectors in browser flows
P055 to P062	rfc6749:3170-3210	Outside the scope	they concern misuse of tokens in the implicit grant, which is not used here
P063	ts-33-501:2491-2491	The prover cannot be asked	bidding-down attacks depend on the negotiation of algorithms and security features, which the symbolic model does not represent
P064 to P072	ts-33-501:2495-2511	Outside the scope	they concern UE and serving-network authentication over NAS, outside the API sub-flow in scope
P073 to P074	ts-33-501:2515-2517	Outside the scope	they concern AS and NAS key lengths, outside the API sub-flow in scope

Paragraphs	Where	Taken up as	Why not
P075 to P213	ts-33-501:2523-2853	Outside the scope	they are the requirements on the UE, the gNB, the ng-eNB, the SEAF, the UDM/SIDF, the AUSF and the UDR for radio and subscription security, none of which this API sub-flow uses
P214	ts-33-501:2859-2859	Z05	
P215	ts-33-501:2865-2865	X01, Z01	
P216	ts-33-501:2867-2867	Z01	
P217	ts-33-501:2869-2869	Outside the scope	it requires hiding the topology of one trust domain from another, which concerns inter-PLMN deployment and not this intra-PLMN API
P218	ts-33-501:2871-2871	A02, Z03	
P219	ts-33-501:2873-2873	A02, D01, Z05, `scp_compromised`	
P220	ts-33-501:2875-2875	A08, D01	
P221 to P227	ts-33-501:2879-2891	Z01, `network`	
P228 to P234	ts-33-501:2895-2907	Outside the scope	they are the security requirements of the NEF towards application functions, which this API does not traverse
P235 to P239	ts-33-501:2911-2919	Z05, `scp_compromised`	
P240	ts-33-501:2921-2921	Z05, D01, `scp_compromised`	
P241	ts-33-501:2923-2923	No goal stated	it is a rule on filling the 3gpp-Sbi-Originating-Network-Id header by the SCP; it states no property of this API that can be checked
P242 to P254	ts-33-501:2929-2953	Outside the scope	they concern interconnection between core networks, outside this intra-PLMN API
P255 to P312	ts-33-501:2957-3075	Outside the scope	they concern the SEPP, roaming and interconnect protection, outside the analysed sub-flow
P313 to P318	ts-33-501:3079-3091	Outside the scope	they concern IPUPS and NF requirements for inter-PLMN traffic, outside this sub-flow

Paragraphs	Where	Taken up as	Why not
P319 to P327	ts-33-501:3097-3115	Outside the scope	they concern signalling monitoring and streaming of monitoring data, a separate function
P328 to P337	ts-33-501:3121-3141	Outside the scope	they concern operator security visibility and configurability of the network, not a property of this API
P338 to P369	ts-33-501:3149-3215	The prover cannot be asked	they are the ciphering and integrity algorithm identifiers and the algorithm selection requirements; the strength of a primitive is outside what the prover can be asked
P370 to P375	ts-33-501:3219-3231	Outside the scope	they are requirements on the 5G-RG and the NSSAAF, outside this sub-flow
P376 to P380	ts-33-501:6540-6548	Outside the scope	they concern the N3 user-plane interface
P381 to P385	ts-33-501:9656-9666	Outside the scope	they concern EAP-TLS variants for UE authentication
P386 to P394	ts-33-501:10893-10917	Outside the scope	they concern IAB nodes and their interfaces
P395 to P396	ts-33-501:11439-11443	Outside the scope	they concern the MBSF and MBSTF functions
P397 to P398	ts-33-501:11899-11901	Outside the scope	they are NEF security requirements, and this API is not traversed through the NEF

The parties and the protected values, one by one

The host listed the 37 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: 13. Without: 24. 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	NWDAF with AnLF	party	`malicious_nf`	
S02	NWDAF with MTLF	party	`malicious_nf`	
S03	NRF	party	Everything lost by definition	the NRF is the authorisation server; if the attacker is the NRF every token property asked here is lost by definition and the documents ask none of them under that attacker
S04	External assailants	party	Nothing gained	the external attacker is the attacker on the network, already present in the analysis
S05	Malicious and lawful NF	party	`malicious_nf`	
S06	Other legal NF	party	`malicious_nf`	
S07	The captured, TLS, the termination agent.	party	`scp_compromised`	
S08	External cyber attacker	party	Nothing gained	the external network attacker is the attacker on the network, already present

ID	Party or value	Kind	Attacker	Why none
S09	Sold TLS Terminator or SCP	party	`scp_compromised`	
S10	NF Example Identity	value	Nothing gained	NF instance identities are published by the NRF through discovery
S11	Service name nnwdaf-mlmodelmontor	value	Nothing gained	the service name is public in every request URI
S12	modelId / modelIds	value	Nothing gained	the ML model identifiers are not secret; the property asked about them is agreement, not confidentiality
S13	notifCorrId	value	Nothing gained	the correlation identifier is created by the subscriber and travels only on its own TLS link; holding it does not let an attacker speak as the producer
S14	notificationUri	value	Nothing gained	the notification URI is the address of the subscriber's endpoint and is not a secret
S15	mLEvent、mLEventFilter、tgtUe	value	Everything lost by definition	if the attacker already holds the target UE information the confidentiality question about it is answered by the assumption itself
S16	registrationId	value	`malicious_nf`	
S17	subscriptionId	value	`malicious_nf`	
S18	consumerId / consumerSetId	value	Nothing gained	consumerId and consumerSetId are identities published by the NRF; the property asked is that they are bound to the authenticated consumer, not that they are secret
S19	tokens subject, issuer, audience	value	Nothing gained	the subject, issuer and audience of a token are read by every producer the token is presented to
S20	TLS Client Certificate Identity	value	Nothing gained	the NF Instance ID in the TLS client certificate is the public identity of the NF
S21	modelIds	value	Nothing gained	the model IDs are not secret; the property asked about them is agreement with the subscription
S22	token	value	Nothing gained	the scope is a public service name carried in the token
S23	NRF token signature private key	value	Everything lost by definition	with the NRF signing key every token can be forged, so all token properties asked here are lost by definition; the documents ask none of them under that attacker
S24	NRF Share symmetric keys with producers	value	Everything lost by definition	with the NRF-producer shared key tokens can be forged by definition; the documents ask none of them under that attacker
S25	Peers TLS Session Keys	value	`scp_compromised`	
S26	modelAccuInfos	value	Everything lost by definition	the monitored accuracy information is exactly the value whose confidentiality is asked; an attacker who holds it has already won that claim
S27	anaFeedbacks	value	Everything lost by definition	the feedback is exactly the value whose confidentiality is asked
S28	tgtUe	value	Everything lost by definition	the target UE information is exactly the value whose confidentiality is asked
S29	Access tokens	value	Nothing gained	the token is a bearer credential, but the producer checks its subject against the TLS client certificate identity (s0480); a stolen token presented over the attacker's own TLS session is rejected, which the model shows
S30	token object, object, scope	value	Nothing gained	the subject, audience and scope are read by the producer on every request
S31	consumerId	value	Nothing gained	consumerId is the public identity of the consumer

ID	Party or value	Kind	Attacker	Why none
S32	Registration and Subscription Resources per NF	value	`malicious_nf`	
S33	TLS Sessions	value	`scp_compromised`	
S34	Consumer identity	value	Nothing gained	the consumer identity is the public NF instance identity carried in the TLS certificate and the token
S35	Notify Load	value	Everything lost by definition	the notification payload is the value whose confidentiality and integrity are asked; holding it answers the question by the assumption
S36	Subscription and Registration Resources	value	`malicious_nf`	
S37	Sign up and subscribe to resources	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`	One NF instance identity of its own, its own TLS credentials for that identity, and a valid access token issued by the NRF for the nnwdaf-mlmodelmonitor scope, obtained before the run.	The API defines only a service-level scope, so every consumer allowed to access the service may invoke all its service operations (s0347, s0397); the token subject is checked against the TLS certificate (s0480), so the attacker gets no other NF's identity.	s0347, s0397, s0480, s0485
`scp_compromised`	The plaintext and key material of both directions of the TLS connections it terminates on the AnLF-MTLF link, from the moment it is compromised, which is after the subscription and before the notification.	All parties communicating through the SCP need to trust the SCP to correctly handle the messages passing through it (s2921); there is no mutual authentication between consumer and producer in indirect communication via SCP (s2873).	

ID	`network`	`malicious_nf`	`scp_compromised`
A02	Holds		
A03	Holds		
A04	Holds		
A07	Holds	Fails	
A08	Holds	Holds	Holds
K01	Holds	Holds	
K02	Holds	Holds	Fails
D01	Holds	Holds	Fails
Z01	Holds		

ID	network	malicious_nf	scp_compromised
Z03	Holds	Fails	
Z05	Holds	Holds	Holds
U01	Holds	Holds	
U02	Holds		Fails
U03	Holds	Fails	
X01	Holds		Fails

Notes on the prover and the model

What the analyst named for targets:

step	targets	features	reason
46	A02, A03, A07, A08, K01, K02, D01, Z01, Z03, Z05, U01, U02, U03	plain_primitives, many_sessions	All targets rest on ordinary primitives only: TLS modelled as an authenticated confidential channel under a private per-pair key, the access token as a signature of the NRF, and a synchronous authenticated encryption for the payload. The model covers an unbounded number of sessions of every role, so the agreement, secrecy and injectivity answers are about any number of sessions. The TLS handshake itself is not modelled: no key confirmation and no transcript, which is what makes the replay questions of A04 and X01 depend on the model.
46	A04, X01	other	A04 and X01 ask whether one request or notification can be accepted twice. That property rests on the replay protection of the TLS record layer, which the model represents by a monotonically growing table of request nonces; ProVerif over-approximates tables, so a repeated nonce is still accepted in the model even though the protocol's transport rejects it.

The analyst's closing note

All fifteen targets are answered under every attacker set for them. The plain run of Register, Subscribe and Notify, the access-token issuance by the NRF and two attackers beyond the network (a malicious NF with its own credentials, a TLS terminating proxy on the AnLF-MTLF link) are in model.pv; no target rests on a value that changes, so there is no bounded.pv.

Proved for the network and for a malicious NF: NF instance authentication of every acceptance, the correlation-identifier and model binding of a notification to its subscription, the token subject matched against the TLS client certificate, the confidentiality of accuracy and feedback, notification authenticity, the token checks, the isolation of the links a compromised proxy does not terminate, and the unguessability of other instances' resource identifiers. Controls were recorded for the proofs that have a failing variant (a02_reg_ctl, a02_sub_ctl, k01_tlskey_ctl, z05_regid_ctl, z05_corr_ctl, d01_ctl, k02_ctl, x01_ctl).

One gap of the specification text: the consumerId of a registration is never compared with the authenticated consumer, so any instance holding a valid token of its own registers under another instance's consumerId (a07_body_nf, under malicious_nf). The change that closes it was checked on fix-consumerid.pv: the attack query becomes verified, the run survives and no other recorded answer changes.

Recorded known results reproduced: the accuracy and feedback payloads are readable and rewritable by a compromised SCP, and a notification can be delivered twice, exactly as TS 33.501 says of the SCP trust and of OAuth2 not applying to notifications.

What blocks the rest: A04 is answered only in its identifier-injectivity reading; its request-replay reading is falsified in the model because ProVerif over-approximates the table of request nonces, and a per-connection-ticket variant lost the normal run. Expiry, revocation, resource-level authorization, consumerSetId and the TLS handshake itself are

outside the model, as listed in the limits.