Security analysis of NWDAF analytics subscription transfer control

19.8.0 · ProVerif+ Agent report

Documents	5G System; Network Data Analytics Services; Stage 3 (protocol definition)
Properties examined	11: 7 hold, 4 fail (7 attacks)
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_EventsSubscription_Transfer service operation lets one NWDAF instance hand an existing analytics subscription to another NWDAF instance. Three parties take part: the existing analytics consumer nfc, which holds one non-UE aggregate analytics subscription and the notificationURI where it wants notifications; the source NWDAF src, which serves that subscription and acts as the NF service consumer of the transfer; and the target NWDAF tgt, which receives the subscription.

The source creates a transfer request with POST and a PREPARE body, receives a 201 with the transferId, executes it with PUT and a TRANSFER body, and the target then delivers analytics notifications to the consumer. A DELETE cancels a prepared request. Every request carries an OAuth2 access token obtained from the NRF; the pre-existing subscription's notificationURI travels in nwdafEvSub and consumerId names the analytics consumer.

The parties:

- **Analytics consumer (nfc)**: existing consumer of one non-UE aggregate analytics subscription. Holds: subscriptionId, notificationURI.
- **Source NWDAF (src)**: serves the subscription, NF service consumer of the transfer. Holds: consumerId, analytics subscription, access token from the NRF.
- **Target NWDAF (tgt)**: NF service producer, receives the subscription. Holds: NRF public key.

Figure 1 shows the exchange as the documents specify it.

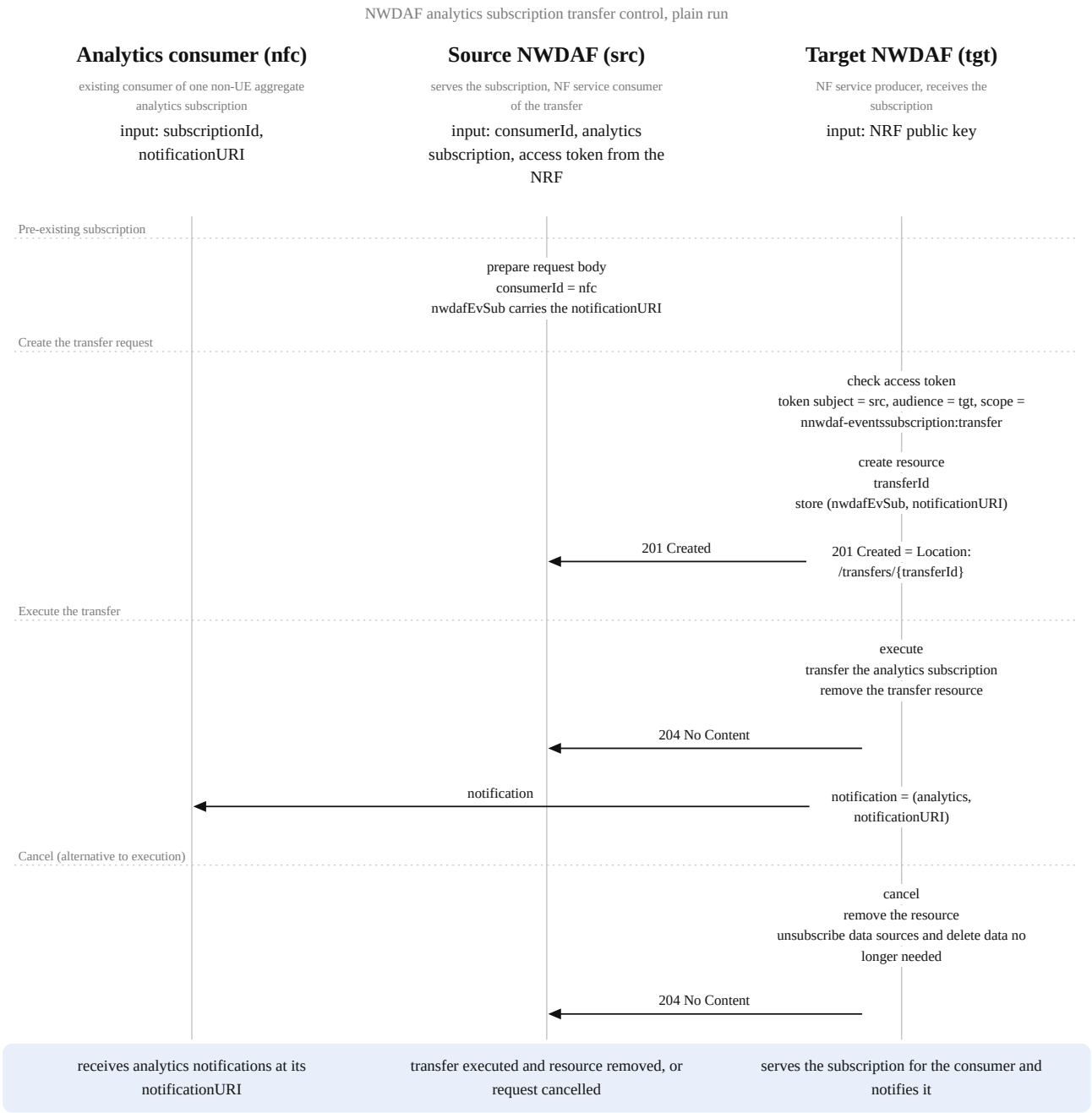


Figure 1. The protocol as specified.

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.

The protocol does not give the property it is meant to give: a legitimate NF holding its own valid transfer-scope token can make the target NWDAF accept and execute a transfer that names another consumer's subscription, so that consumer's notificationURI can be replaced by the attacker's. That is the one finding; the documents only warn in general that a service-level scope must not grant undue write access, and do not state the failure. Against the network

alone the protocol behaves correctly: no field can be changed, agreement with the source holds, the token checks hold, and the transfer resource rejects replay and retransmission. Of the eleven targets, seven are proved or bounded-verified with controls and four carry attacks; every answer rests on the model simplifications listed as limits.

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

2.1 A legitimate NF can transfer another consumer's analytics subscription to itself

Kind	Weakness of the protocol as specified
New	Unclear: the documents state it in part, or could not be searched (ts-29-501:1801, ts-33-501:8040)
Value	high
Properties	A03, Z03, Z03.10, A03.8 (section 3)
Source text	s0014, s0020, s0022, s0152, s0184
Shown by	ProVerif attack queries `z0310_nfc`, `z03b`, `a038_nfc`; Figures 2, 3, 4 and 5

What happens. The target NWDAF checks the access token's subject, audience and transfer scope, but not the consumer or subscription named in the request body. A legitimate NF instance that presents its own valid transfer-scope token therefore gets a transfer request accepted with consumerId set to another consumer, creates the transfer resource, executes it, and has the target record its own notificationURI for that consumer. The source NWDAF never sent the request.

What the attacker needs. The attacker is a legitimate NF instance with its own TLS credentials and its own valid OAuth2 access token for the nnwdaf-eventssubscription:transfer scope with the target NWDAF as audience. It needs the consumerId of the target consumer and the analytics subscription identifier, both carried in nwdafEvSub/consumerId on the service-based interface.

What the attacker gains. The attacker can create and execute a transfer for another consumer's subscription, so the target associates that consumer with the attacker's notificationURI; subsequent analytics notifications for the consumer can be directed to the attacker, and the consumer loses its analytics stream.

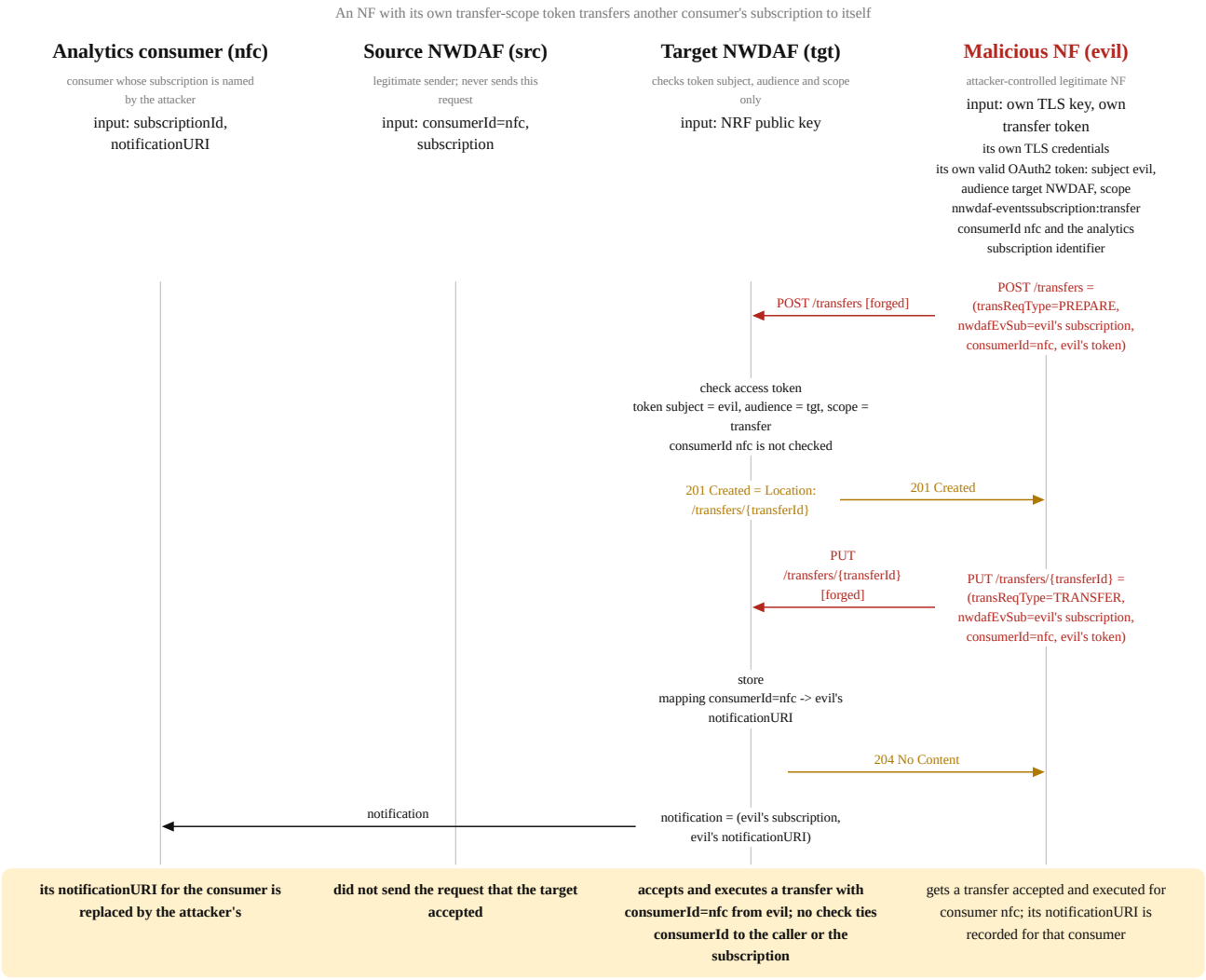


Figure 2. An NF with its own transfer-scope token transfers another consumer's subscription to itself. Property Z03.10; attack queries z0310, z0310_nfc.

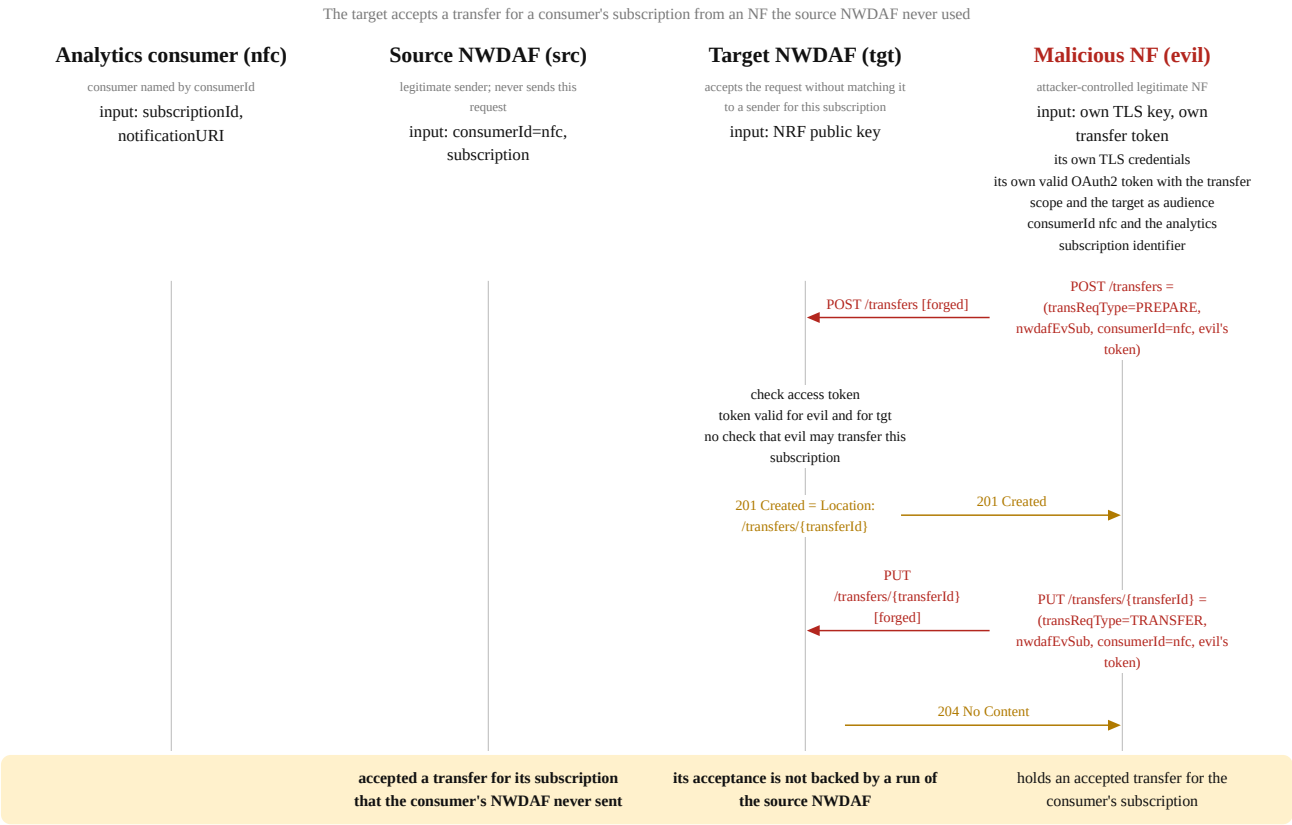


Figure 3. The target accepts a transfer for a consumer's subscription from an NF the source NWDAF never used. Property A03; attack queries a03_any, z03b.

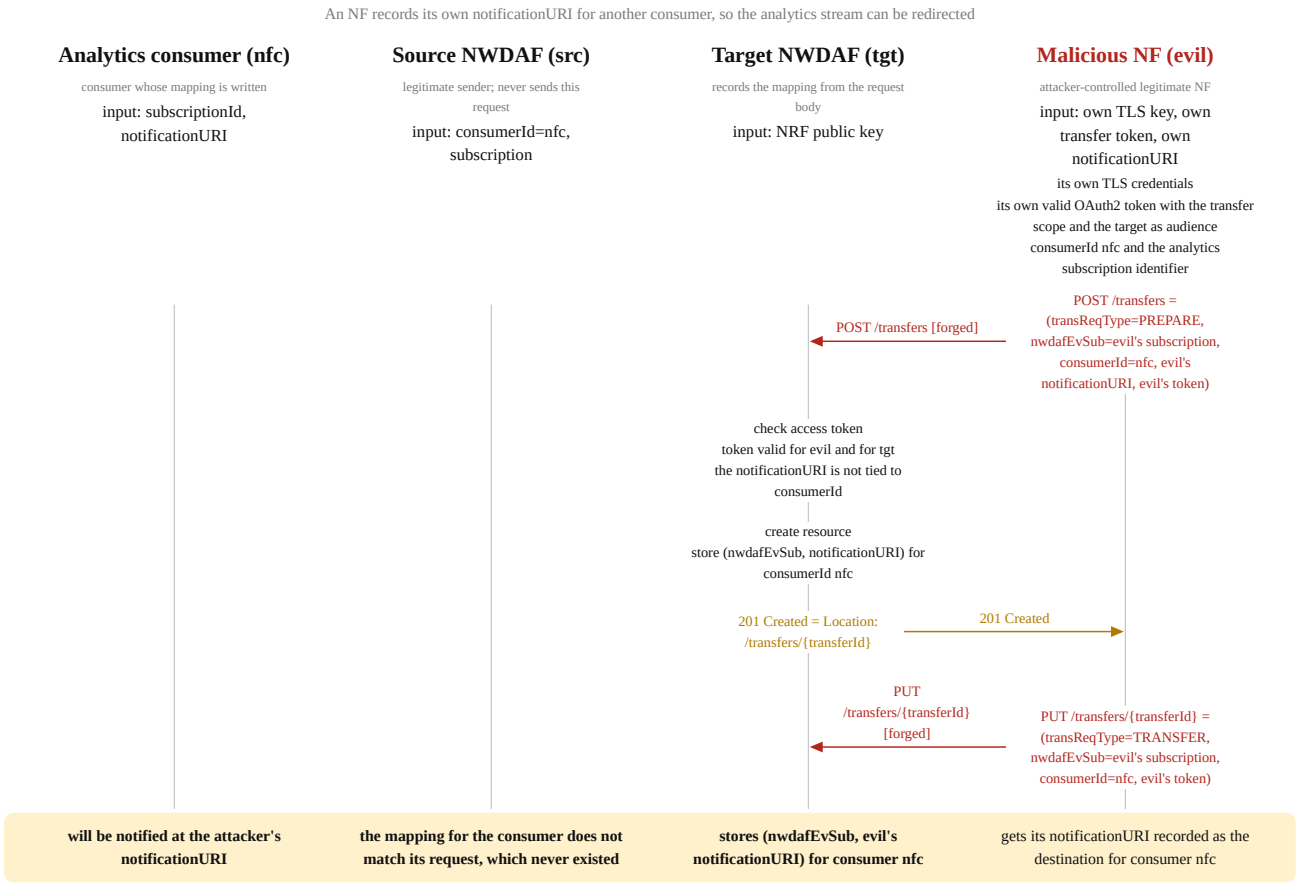


Figure 4. An NF records its own notificationURI for another consumer, so the analytics stream can be redirected. Property A03.8; attack queries a038_evil, a038_nfc.

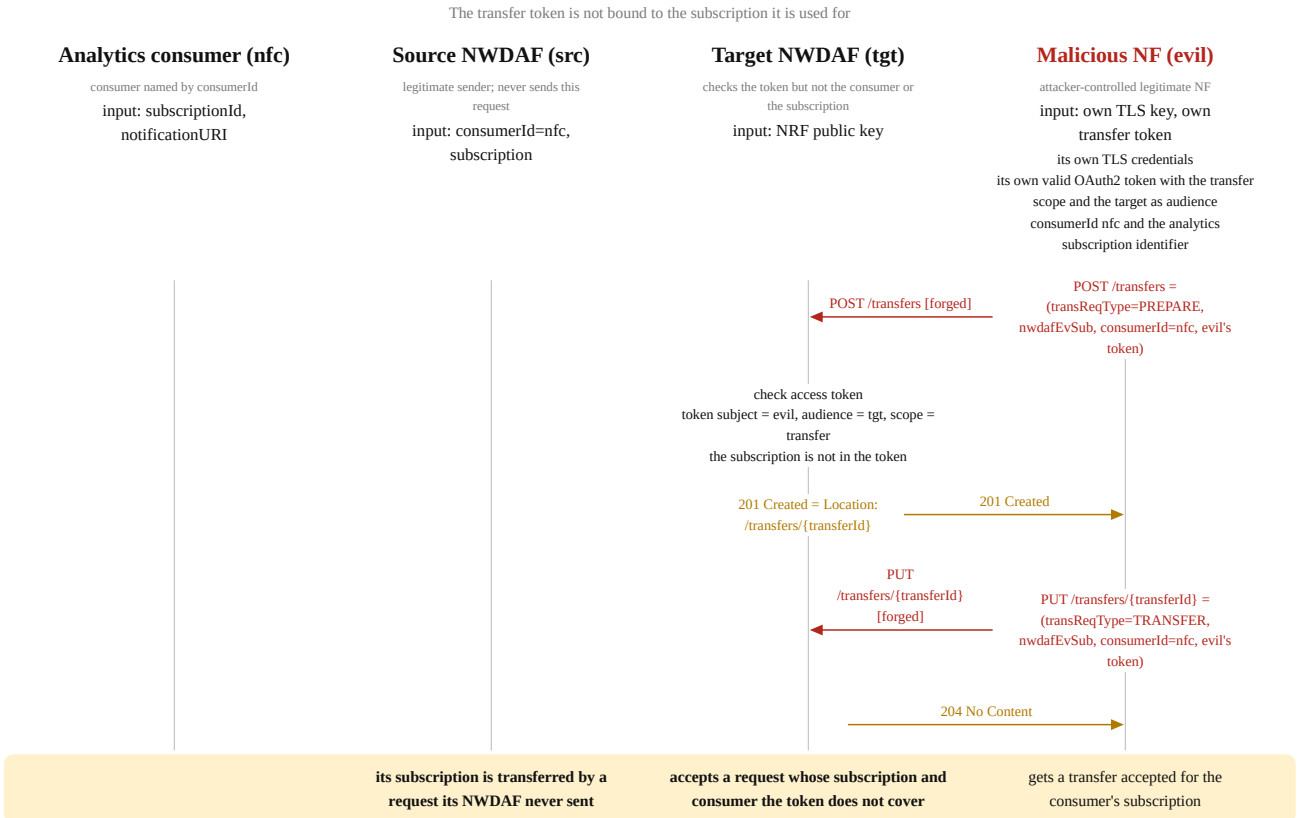


Figure 5. The transfer token is not bound to the subscription it is used for. Property Z03; attack query z03b.

Proposed change. Carry the authorized consumer in the access token (an additional resource scope naming the subscription or consumerId), have the NRF issue it only for the (subject, consumer) pair it authorizes, and have the target NWDAF reject a POST or PUT whose body consumerId differs from the token's consumer or from the resource's stored consumer, and a DELETE whose token is not for the resource's stored consumer. check_fix on fix-binding.pv verifies the attack query z0310_nfc (and z03b, a038_nfc) and keeps the normal run.

Check of the change with ProVerif. The change was put into a copy of the model: The access token carries the consumer (consumerId) its subject is authorized to transfer; the NRF issues the token only for that (subject, consumer) pair; the target NWDAF requires the consumerId in a PUT to equal the token's consumer and the resource's stored consumer, and requires a DELETE to carry a token for the resource's stored consumer. On the changed model ProVerif proves `z0310_nfc`: the attack is no longer possible there. The normal run of the protocol is still found. Of the other recorded answers, 15 still hold and 3 changed: `z03b` (A03, was attack) is now verified; `z03b` (Z03, was attack) is now verified; `a038_nfc` (A03.8, was attack) is now verified. 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-binding.pv`.

No figure was given for these attacks of a finding: A03.8: z0310_nfc, A03.8: z03b, A03: a038_nfc, A03: z0310_nfc, Z03.10: a038_nfc, Z03.10: z03b, Z03: a038_nfc, Z03: z0310_nfc.

3 Properties examined

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

ID	Property	Compromise assumed	Result	Value	In the documents
A03	**Accepted transfer request agrees with its sender.** ObjectiveNWDAFEach time a request for relocation is accepted (see below).PREPAREorTRANSFER. . . . . .NWDAF♪ A real run, both sides to subscribe to the logo ♪consumerId、notificationURIandtransReqTypeUnanimously.	no compromise	Fails (2 attacks)	medium	unclear
A04	**At most one execution per transfer request.** ObjectiveNWDAFSame request for relocation (sametransferId/ The same subscription) for up to one execution; Every time the corresponding source is acceptedNWDAFDifferent running.	no compromise	Holds within a bound (the changing value unrolled to 4 steps)	medium	unclear
Z01	**Transfer accepted only with valid authorization.** For each migration request to be accepted and submitted by the target NWDAF, a valid permit must be issued by the authorized server, overwhelming the transfer scope and the subject matches consumerId.	no compromise	Holds	medium	yes
Z03	**Token bound to consumer and transfer resource.** Licences used in migration requests, their subjects, audiences, resources (subscriptions/comments)transferId) and action (transfer) must be in the requesting bodyconsumerIdandnotificationURIUnanimously.	no compromise	Fails (1 attack)	high	unclear
S04	**No mixing across transfer contexts.** TRANSFERSubscriptions used during the implementation phaseconsumerId/notificationURIIt has to come from the same migration request.PREPAREThe stage or the same request shall not be mixed with other moving or subscription contexts.	no compromise	Holds	low	unclear

ID	Property	Compromise assumed	Result	Value	In the documents
S05	**No execution after cancel or removal.** No new migration execution or data collection action may be submitted using old transferId or the context of the migration after the resources have been cancelled or deleted.	no compromise	Holds within a bound (the changing value unrolled to 4 steps)	medium	unclear
D01	**Accepted fields are the sender's own values.** ObjectiveNWDAFAcceptance of each migration request field (%2)transReqType、nwdafEvSub、consumerId、notificationURI. . . . . .NWDAFThe real dispatch was not replaced by an intermediary.	no compromise	Holds	low	yes
A03.8	**Target stores the source's subscription andnotificationURI.** ObjectiveNWDAF_Other OrganisernotificationURIIt has to be with the source.NWDAFConsistency in request.	no compromise	Fails (2 attacks)	high	unclear
S04.9	**No notification under a stale transfer context.** Upon cancellation or renewal of the relocation request, the notification may no longer be submitted in the context of the displacement that has expired.	no compromise	Holds within a bound (the changing value unrolled to 4 steps)	low	unclear
Z03.10	**MaliciousNFcannot transfer another's subscription.** Malicious possession of his own legal badgeNFUnable to submitconsumerIdThe relocation of other consumers is requested and accepted.	no compromise	Fails (2 attacks)	high	unclear
X01	**Transfer request data confidential on the network.** An attacker holding only the network cannot obtain the access token used for a transfer request, nor the subscription identifier and notificationURI it carries.	no compromise	Holds	low	yes

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: Old Subscriptions and NewNWDAF/notificationURIAssociation	For the source NWDAF's own transfer the target records exactly the subscription and notificationURI the source sent, but the association is not protected against another NF: any NF with a valid transfer token can write a mapping for consumer nfc and replace its notificationURI (A03.8, Z03.10).	A03.8, Z03.10, Z03
Q2: Consistency between removal requests/updates and the limited status of notification in transit	With the transfer resource removed on cancel, the target does not deliver a pending notification under the removed context for the four unrolled steps (S04.9, S05); recovery from a lost 204 and notifications already queued at a data source are not modelled.	S04.9, S05, A04
Q3: Malicious and lawfulNFCan't move other consumer subscriptions to themselves	No: a legitimate NF with its own valid transfer-scope token can submit a transfer naming another consumer's consumerId and have the target accept and execute it, so the consumer's notificationURI can be redirected to the attacker (Z03.10, A03, Z03, A03.8).	Z03.10, A03, Z03, A03.8

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:

- The symbolic model fixes the cryptographic primitives; it does not prove computational or post-quantum strength (C08 and the algorithm requirements are not askable).
- Availability, timing, progress, deadlock and resource consumption are outside what a trace-property prover answers (V01, V02, V03).
- TLS and OAuth2 keys, token expiry, revocation and the NRF itself are modelled abstractly; the NRF is trusted and the token is treated as unexpired.
- The transfer resource is unrolled to four steps; real deletion, persistence, caching and a shared resource table for any number of sessions are not modelled.
- The malicious NF is assumed to know the consumerId and the analytics subscription identifier of its victim; if those were secret, the attack would need a further step to learn them.
- Indirect communication through an SCP or SEPP, roaming and the NEF path are out of scope; the model assumes direct NF-to-NF communication.
- The same-subscription race, two transfer requests for one subscription each executing, was identified in the state review but is not confirmed by a query.
- The partial-accept branch (PartialAnalyticsSubTransfer) and the no-rollback behaviour on partial failure are not modelled.
- Tamarin+ Agent, which models a changing resource table with its real deletion semantics, would settle the state-review candidates and the same-subscription race.

What the host records:

- The answers of A04, S05, S04.9 rest on an operation the model simplifies (Appendix D).

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

Appendix A: Reading of each property

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

A03: Accepted transfer request agrees with its sender

ObjectiveNWDAFEach time a request for relocation is accepted (see below).PREPAREorTRANSFER. . . . . .NWDAF ♪ A real run, both sides to subscribe to the logo ♪ consumerId 、notificationURIandtransReqTypeUnanimously.

For the source NWDAF, the target's acceptance is backed by the source's own send and no network attacker can change a field; the accepted fields equal the source's values. For a different NF the agreement fails: an NF with its own valid transfer token gets an acceptance for the consumer's subscription although the source never sent it.

- Result: Fails (2 attacks); answers what is asked: yes; usable with caveats
- Value medium: The agreement part is a proof of a central guarantee under plain primitives; the failure for another NF is the ownership finding, which the documents mention only as a general warning (ts-29-501 1801) and not as a failure of this procedure.
- Stated in the documents: unclear. Source text it rests on: ts-29-501:1801
- Rests on: TLS is a pairwise authenticated encryption; the token is an unforgeable NRF signature.; The network-only answer is a query restricted to the source's transfers, because the malicious NF's credentials are present in the same model.; No attacker can act as an NF without that NF's TLS key and token.
- Open: Whether the target has a local policy that rejects a consumerId it does not serve.
- Next step: Model a deployment policy binding consumerId to the calling NWDAF and re-ask the second claim.

A04: At most one execution per transfer request

ObjectiveNWDAFSame request for relocation (sametransferId/ The same subscription) for up to one execution; Every time the corresponding source is acceptedNWDAFDifferent running.

With the transfer resource removed on execution, a TRANSFER retransmitted by the network, or repeated with the same transferId, is not executed a second time; the property holds for the four unrolled steps. The control that leaves the resource-state check out executes twice, so the property depends on that check.

- Result: Holds within a bound (the changing value unrolled to 4 steps); answers what is asked: yes; usable with caveats
- Value medium: The resource lifecycle of s0021, s0036 and s0047 implies the property but the documents do not state the duplicate-execution rule explicitly.
- Stated in the documents: unclear. Source text it rests on: s0021, s0036, s0047
- Rests on: The target is unrolled to four message steps; a real resource table is not modelled for any number of sessions.; The target validates a message against the resource state (ts-33-501 2875).; One transfer request per unrolled chain; the same-subscription race between two requests is not modelled.
- Open: Two transfer requests for the same subscription each executing once.
- Next step: Unroll two resources for the same subscription and ask the per-subscription reading.

Z01: Transfer accepted only with valid authorization

For each migration request to be accepted and submitted by the target NWDAF, a valid permit must be issued by the authorized server, overwhelming the transfer scope and matching the subject to consumerId.

Every accepted transfer carries an NRF-issued token with its integrity intact, with the target NWDAF as audience and the nnwdaf-eventssubscription:transfer scope; a request without such a token is not accepted, from the network or from another NF.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value medium: The baseline requires the resource server to validate the token, its scope and its expiry before allowing access, and the documents state the transfer scope.
- Stated in the documents: yes. Where the documents say it: ts-33-501:8014, ts-29-500:4994, s0179, s0180, s0184
- Rests on: The token is a signature under a fixed NRF key; revocation and expiry are not modelled.; The NRF is trusted and the scope is a fixed constant.; The NRF is assumed to issue the token only after authenticating the NF.
- Open: Expired or revoked tokens.; The audience and subject checks are modelled, the expiry check is not.
- Next step: Add token lifetime and revocation events and re-ask.

Z03: Token bound to consumer and transfer resource

Permissions used in migration requests, their subjects, audiences, resources (subscribes/subscribes)transferId) and action (transfer) must be in the requesting bodyconsumerIdandnotificationURIUnanimously.

The token is correctly bound to the calling NF and to the target NWDAF, and the transfer scope covers the operation. It is not bound to the consumer or the subscription: a token holder can name any consumerId and notificationURI, so the resource part of the property fails.

- Result: Fails (1 attack); answers what is asked: partly; usable with caveats
- Value high: The documents warn that a service-level scope must not give undue consumers write or delete access and require the additional scope to match the operation, but they do not state that this procedure fails to bind the consumer or the subscription.
- Stated in the documents: unclear. Source text it rests on: ts-29-501:1801, ts-33-501:8040
- Rests on: The source text defines no per-resource authorization; the model gives the malicious NF its own valid token.; Token expiry and revocation are not modelled.; The network-only resource claim is answered by restricting the query to the source's transfer.
- Open: Whether an operator's local policy adds the per-resource check.
- Next step: The checked fix in fix-binding.pv shows that a per-consumer token removes the failure.

S04: No mixing across transfer contexts

TRANSFERSubscriptions used during the implementation phaseconsumerId/notificationURIIt has to come from the same migration request.PREPAREThe stage or the same request shall not be mixed with other moving or subscription contexts.

A transfer is executed only on the resource created for that transferId by the same NF, so the executed context belongs to that transfer request and not to another; no cross-context mixing was found against the network or against another NF.

- Result: Holds; answers what is asked: yes; usable with caveats

- Value low: The resource represents one request to transfer subscriptions and a PUT acts on the resource identified by transferId, but the documents do not state the cross-context rule explicitly.
- Stated in the documents: unclear. Source text it rests on: s0086, s0087, s0088, s0032
- Rests on: model.pv binds each transfer to one linear session; the resource is not a shared table.; A cross-resource race between two transfer requests of the same consumer is not modelled.
- Open: A second transfer request for the same subscription.
- Next step: Unroll two resources and test whether the same subscription is executed twice.

S05: No execution after cancel or removal

After the resources have been cancelled or deleted, the old transferId or the old context of the migration cannot be used to submit new migration execution or data collection actions.

With the resource removed on execution or on cancel, a TRANSFER replayed with the old transferId is rejected, so no execution happens after removal for the four unrolled steps. The control without the resource-state check executes the replay, so the property depends on that check.

- Result: Holds within a bound (the changing value unrolled to 4 steps); answers what is asked: yes; usable with caveats
- Value medium: The documents remove the resource on execution and on cancel and the baseline requires validation against network state, but the rule for a later reuse of the transferId is not stated.
- Stated in the documents: unclear. Source text it rests on: ts-33-501:2875, s0036, s0045, s0047
- Rests on: The target is unrolled to four message steps; real deletion, persistence and caching are not modelled.; The target validates an incoming message against network state.
- Open: A lost 204 makes the source retransmit and be rejected.
- Next step: Model the lost response and the source's recovery.

D01: Accepted fields are the sender's own values

ObjectiveNWDAFAcceptance of each migration request field (%2)transReqType、 nwdafEvSub、 consumerId、 notificationURI. . . . . .NWDAFThe real dispatch was not replaced by an intermediary.

Every field of an accepted transfer request equals what the source NWDAF sent; a man in the middle on the network cannot replace transReqType, nwdafEvSub, consumerId or notificationURI.

- Result: Holds; answers what is asked: yes; usable
- Value low: The baseline requires mutual authentication between NF service consumer and producer and TLS protection of the request; the documents define the fields.
- Stated in the documents: yes. Where the documents say it: ts-33-501:8012, s0010, s0012, s0013, s0014
- Rests on: TLS is a pairwise authenticated encryption; the token is an unforgeable signature.; Neither the source's nor the target's long-term key is compromised.
- Next step: None needed within the stated model.

A03.8: Target stores the source's subscription and notificationURI

ObjectiveNWDAF_Other OrganisernotificationURIIt has to be with the source.NWDAFConsistency in request.

For the source NWDAF's transfer, the target records exactly the source's subscription and notificationURI. But any NF with a valid transfer token can make the target record its own notificationURI for consumer nfc, so the consumer's mapping can be taken over.

- Result: Fails (2 attacks); answers what is asked: yes; usable with caveats
- Value high: The documents require nwdafEvSub to contain the notificationURI and consumerId to name the consumer, but they do not state that the target binds the notificationURI to that consumer.
- Stated in the documents: unclear. Source text it rests on: s0013, s0014, s0152
- Rests on: The target routes notifications by consumerId; the ownership of a notificationURI is not checked, as the source text names only consumerId.; The subscription identifier and consumerId are treated as available to the malicious NF.
- Open: Whether the target has a prior consumer-to-URI binding.
- Next step: Add a consumer-to-notificationURI binding at the target and re-ask.

S04.9: No notification under a stale transfer context

After cancellation or renewal of the relocation request, the notification may no longer be submitted in the context of the displacement that has lapsed.

After a cancel removes the transfer resource, the target does not deliver a pending notification for that subscription; the control without the resource-state check does deliver it, so the property depends on the removal.

- Result: Holds within a bound (the changing value unrolled to 4 steps); answers what is asked: yes; usable with caveats
- Value low: A cancelled transfer removes the resource and the data no longer needed, but the documents do not state a rule for in-flight notifications.
- Stated in the documents: unclear. Source text it rests on: s0045, s0046, s0047
- Rests on: The target is unrolled to four steps and the in-flight notification is modelled as one pending notification step.; Notifications already queued at a data source are not modelled.
- Open: Notifications already in flight at a data source.
- Next step: Model a queue at the data source and the cancel ordering.

Z03.10: Malicious NF cannot transfer another's subscription

The malice of holding his own legal badge.NFUnable to submitconsumerIdThe relocation of other consumers is requested and accepted.

The attack succeeds: with its own valid transfer-scope token a legitimate NF gets the target to accept and execute a transfer naming consumerId nfc, and the target records the attacker's notificationURI for that consumer. The token checks bind the caller and the scope but not the consumer or the subscription.

- Result: Fails (2 attacks); answers what is asked: yes; usable
- Value high: The documents warn that a service-level scope must not grant undue write or delete access and that the additional scope must match the operation, but they do not state that this procedure fails to bind the consumer or describe the attack.
- Stated in the documents: unclear. Source text it rests on: ts-29-501:1801, ts-33-501:8040, s0014, s0020, s0022
- Rests on: The malicious NF has its own valid token, as any legitimate NF does.; consumerId and the subscription identifier are known or inferable to the attacker.; The network alone cannot do it, because it lacks an NF's TLS key and token.

- Open: Whether a local target policy would reject a consumerId it does not serve.
- Next step: Check the per-consumer token fix: check_fix shows z0310_nfc verified on fix-binding.pv.

X01: Transfer request data confidential on the network

An attacker holding only the network cannot obtain the access token used for a transfer request, nor the subscription identifier and notificationURI it carries.

An attacker holding only the network does not obtain the access token, the subscription identifier or the notificationURI of the source's transfer request.

- Result: Holds; answers what is asked: yes; usable with caveats
- Value low: OAuth2 access tokens must be kept confidential in transit over TLS and the service request is protected by TLS in the baseline.
- Stated in the documents: yes. Where the documents say it: rfc6749:2859, ts-33-501:8012, s0013, s0152
- Rests on: TLS is modelled as encryption under a pairwise key; no computational or quantum claim is made.; No endpoint is compromised.
- Next step: None needed within the stated model.

Appendix C: Claims and prover answers

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

target	level	bound	lemma	control	attacks	claims
A03	proved		a03_src	checked	a03_any (holds_in_source); z03b (holds_in_source)	1 proved, 2 attack
A04	bounded	the changing value unrolled to 4 steps	a04	checked		2 bounded
Z01	proved		z01	checked		1 proved
Z03	proved		z01	checked	z03b (holds_in_source)	3 proved, 1 attack
S04	proved		s04	checked		1 proved
S05	bounded	the changing value unrolled to 4 steps	s05	checked		2 bounded
D01	proved		a03_src	checked		1 proved
A03.8	proved		a038_src	checked	a038_evil (holds_in_source); a038_nfc (holds_in_source)	1 proved, 2 attack
S04.9	bounded	the changing value unrolled to 4 steps	s049	checked		2 bounded
Z03.10	none				z0310 (holds_in_source); z0310_nfc (holds_in_source)	2 attack
X01	proved		x01_tok	checked		3 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
A03	Every transfer the target NWDAF accepts from the source NWDAF carries the subscription identifier, consumerId, notificationURI and transReqType that the source sent; with the network alone the source is the only NF whose transfer can be accepted.	`network`	proved	a03_src	checked
A03	Every accepted transfer is backed by a real run of the source NWDAF: only the source NWDAF can make the target accept a transfer for its subscription.	`malicious_nf`	attack (holds_in_source)	a03_any (the control of a03_src)	
A03	Every accepted transfer naming the consumer's subscription (consumerId = nfc) is backed by a real run of the source NWDAF for that subscription.	`malicious_nf`	attack (holds_in_source)	z03b (the control of a03_src)	
A04	At most one execution per transfer request (same transferId): a TRANSFER retransmitted by the network after the resource was executed and removed is not executed a second time.	`network`	bounded (the changing value unrolled to 4 steps)	a04	checked
A04	At most one execution per transfer request (same transferId): the resource-state check rejects a repeated TRANSFER with the same transferId, whatever NF sends it.	`malicious_nf`	bounded (the changing value unrolled to 4 steps)	a04	checked

target	claim	attacker	answer	query	control
Z01	Every accepted and committed transfer request carries an access token issued by the NRF, with its integrity intact, whose audience is the target NWDAF and whose scope is nnwdaf-eventssubscription:transfer.	`malicious_nf`	proved	z01	checked
Z03	The subscription and notificationURI in a transfer request are ones the token subject is authorized to transfer (the request names the consumer's subscription and a notificationURI belonging to that consumer).	`malicious_nf`	attack (holds_in_source)	z03b (the control of a03_src)	
Z03	The token's subject is the authenticated NF service consumer and its audience is the target NWDAF (with the network alone the token can only be the source's).	`network`	proved	z01	checked
Z03	The token's subject is the authenticated NF service consumer and its audience is the target NWDAF.	`malicious_nf`	proved	z01	checked
Z03	The subscription and notificationURI in the request are ones the request's sender is authorized to transfer; with the network alone the only accepted transfer for the consumer's subscription is the source NWDAF's own.	`network`	proved	a03_src	checked
S04	A transfer is executed only on the transfer resource created for that same transferId by the same NF; the executed context belongs to that transfer request, not to another.	`malicious_nf`	proved	s04	checked
S05	After a transfer resource is cancelled, a TRANSFER replayed with the old transferId is not executed.	`network`	bounded (the changing value unrolled to 4 steps)	s05	checked
S05	After a transfer resource is cancelled, a TRANSFER replayed with the old transferId is not executed, whether the replay comes from the network or from an NF that holds the transferId.	`malicious_nf`	bounded (the changing value unrolled to 4 steps)	s05	checked
D01	Every field of an accepted transfer request (transReqType, nwdafEvSub, consumerId, notificationURI) equals what the source NWDAF sent; a man in the middle on the network cannot replace a value.	`network`	proved	a03_src	checked
A03.8	For the source NWDAF's transfer, the subscription identifier and notificationURI the target records are exactly the ones the source sent in nwdafEvSub.	`network`	proved	a038_src	checked
A03.8	Only the source NWDAF can cause the target to record a notificationURI for the consumer; the target's mapping for the consumer always matches the source's request.	`malicious_nf`	attack (holds_in_source)	a038_evil (the control of a038_src)	
A03.8	Only the source NWDAF can cause the target to record a notificationURI for consumer nfc; the target's mapping for the consumer always matches the source's request.	`malicious_nf`	attack (holds_in_source)	a038_nfc	
S04.9	After a transfer resource is cancelled, no in-flight notification is delivered under the removed transfer context.	`network`	bounded (the changing value unrolled to 4 steps)	s049	checked
S04.9	After a transfer resource is cancelled, no in-flight notification is delivered under the removed transfer context, whether the cancellation came from the network or from an NF.	`malicious_nf`	bounded (the changing value unrolled to 4 steps)	s049	checked

target	claim	attacker	answer	query	control
Z03.10	An NF instance holding its own valid credentials and transfer scope cannot get the target NWDAF to accept a transfer naming another consumer's subscription/consumerId.	`malicious_nf`	attack (holds_in_source)	z0310	
Z03.10	An NF instance holding its own valid credentials and transfer scope cannot get the target NWDAF to accept a transfer naming the consumer's subscription with consumerId = nfc.	`malicious_nf`	attack (holds_in_source)	z0310_nfc	
X01	The access token used for the source's transfer request is not obtained by an attacker holding only the network.	`network`	proved	x01_tok	checked
X01	The subscription identifier carried in the source's transfer request is not obtained by an attacker holding only the network.	`network`	proved	x01_sub	checked
X01	The notificationURI carried in the source's transfer request is not obtained by an attacker holding only the network.	`network`	proved	x01_uri	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-20261114T234529, 2026-11-14. Status: finished. Model: deepseek-v4-1-flash-261017. Steps: 64, prover 7 s, wall 25 min, cost 0.221 USD.

Context: the host removed stale parts from the conversation 1 time(s), 544666 characters in all (earlier reasoning, and what showed the model before its latest change). The trajectory file keeps everything.

ProVerif: 127 proofs run, 127 decided, 0 without an answer, 5 s, 38 answers recorded.

What was analysed

Protocol	NWDAF analytics subscription transfer control
Version	19.8.0
Analysed	Two.NWDAF♪ With one that already exists ♪NFconsumer； Fixed UniqueUEAggregation of subscriptions, creation/update/cancellation of a transfer request and associated confirmation; Retain original subscription with targetnotificationURIMap to exclude the user moving with the complete process.
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	209 numbered passages (162115 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	4.2.2.5	Nnwdaf_EventsSubscription_Transfer service operation
main	4.2.2.5.1	General
main	4.2.2.5.2	Creation of request for analytics subscription transfer
main	4.2.2.5.3	Update a request for analytics subscription transfer
main	4.2.2.5.4	Cancel a request for analytics subscription transfer
main	5.1.1	Introduction
main	5.1.2	Usage of HTTP
main	5.1.2.1	General
main	5.1.2.2	HTTP standard headers
main	5.1.2.2.1	General
main	5.1.2.2.2	Content type

Document	Section	Title
main	5.1.2.3	HTTP custom headers
main	5.1.3.1	Resource Structure
main	5.1.3.4	Resource: NWDAF Event Subscription Transfers
main	5.1.3.4.1	Description
main	5.1.3.4.2	Resource definition
main	5.1.3.4.3	Resource Standard Methods
main	5.1.3.4.3.1	POST
main	5.1.3.4.4	Resource Custom Operations
main	5.1.3.5	Resource: Individual NWDAF Event Subscription Transfer
main	5.1.3.5.1	Description
main	5.1.3.5.2	Resource definition
main	5.1.3.5.3	Resource Standard Methods
main	5.1.3.5.3.1	DELETE
main	5.1.3.5.3.2	PUT
main	5.1.3.5.4	Resource Custom Operations
main	5.1.6.2.40	Type AnalyticsSubscriptionsTransfer
main	5.1.6.2.41	Type SubscriptionTransferInfo
main	5.1.6.2.43	Type AnalyticsContextIdentifier
main	5.1.6.2.49	Type ConsumerNfInformation
main	5.1.6.3.17	Enumeration: TransferRequestType
main	5.1.7	Error handling
main	5.1.7.1	General
main	5.1.7.2	Protocol Errors
main	5.1.7.3	Application Errors
main	5.1.8	Feature negotiation
main	5.1.9	Security

The promise of the program as the source step listed it in the documents:NWDAF♪ With one that already exists ♪NFconsumer； Fixed UniqueUEAggregation of subscriptions, creation/update/cancellation of a transfer request and associated confirmation; Retain original subscription with targetnotificationURIMap to exclude the user moving with the complete process.

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
A03	Accepted transfer request agrees with its sender	The target NWDAF and the NF service consumer of the transfer (the source NWDAF): that the target's acceptance of a transfer request (PREPARE or TRANSFER) is backed by a real send of that consumer with the same subscription identifier, consumerId, notificationURI and transReqType.	The message flow has the source NWDAF send the AnalyticsSubscriptionsTransfer body and the target NWDAF act on it (s0007, s0010, s0020, s0022); the fields are defined in s0012..s0014 and s0152. The documents do not state an explicit agreement property; it is implied by the request/response flow over authenticated TLS (ts-33-501 8012, ts-29-500 4994).	s0003, s0007, s0010, s0014, s0020, s0022, s0152
A04	At most one execution per transfer request	The target NWDAF: that one Individual NWDAF Event Subscription Transfer resource (one transferId, one subscription) is executed at most once, each execution matching a distinct run of the NF service consumer.	A PREPARE creates the resource once (s0021); a TRANSFER (POST or PUT) executes it and removes the resource (s0024, s0036, s0037); a DELETE removes it (s0045, s0047). The documents do not spell out the duplicate-execution rule; the resource lifecycle implies it.	s0021, s0024, s0036, s0037, s0045, s0047
Z01	Transfer accepted only with valid authorizatio n	The target NWDAF (resource server): that every accepted and committed transfer request carries an access token issued by the NRF (authorization server), with integrity intact, covering the nnwdaf-eventssubscription:transfer scope.	The documents authorize the Nnwdaf_EventsSubscription API through OAuth2 client credentials with the NRF as authorization server and define the transfer scope (s0179, s0180, s0182, s0184); the baseline has the resource server verify the token, its scope and expiry (ts-33-501 8014, 8038, 8042; ts-29-500 4994).	ts-33-501:8014, ts-29-500:4994, s0179, s0180, s0182, s0184
Z03	Token bound to consumer and transfer resource	The target NWDAF and the analytics consumer named by consumerId: that the token's subject is the authenticated NF service consumer, its audience is the target NWDAF, and the subscription and notificationURI in the request are the ones that subject is authorized to transfer.	The token checks named in the baseline cover subject vs TLS identity, audience vs producer and scope/additional scope vs the requested operation (ts-33-501 8028, 8032, 8038, 8040); the documents also warn that a service-level scope must not grant inappropriate write or delete access to undue consumers (ts-29-501 1801, 1795). consumerId identifies the consumer of the subscription (s0014, s0152).	ts-33-501:8040, ts-29-501:1801, s0014, s0117, s0152, s0184
S04	No mixing across transfer contexts	The target NWDAF and the source NWDAF: that the subscription content, consumerId and notificationURI used when a transfer is executed belong to the same transfer request (same transferId), not to another transfer or subscription.	The resource represents one request to transfer subscription(s) (s0086, s0087, s0088, s0111); a PUT updates the resource identified by transferId and executes it (s0030, s0032, s0036). The documents do not state the cross-context rule explicitly.	s0086, s0087, s0088, s0032, s0036, s0046, s0047
S05	No execution after cancel or removal	The target NWDAF: that after a transfer resource is removed (by cancel DELETE or by an executed TRANSFER), an old transferId or old transfer context is not accepted again for execution or data collection.	The target removes the resource on execution and on cancel (s0036, s0037, s0045, s0047) and deletes data/unsubscribes when no longer needed (s0046). The baseline requires each NF to validate incoming messages against the protocol and network state (ts-33-501 2875).	ts-33-501:2875, s0036, s0037, s0045, s0046, s0047, s0048
D01	Accepted fields are the sender's own values	The target NWDAF: that every field of an accepted transfer request (transReqType, nwdafEvSub, consumerId, notificationURI) equals what the authenticated sender sent; no man in the middle replaces a value.	The body fields are defined in s0010..s0014 and s0020; the baseline protects the request with TLS and token-based authorization (ts-33-501 8010, 8012, 2871; ts-29-500 4990, 4994).	ts-33-501:8012, s0010, s0012, s0013, s0014, s0020, s0022

ID	Name	About	What the documents say	Where
A03.8	Target stores the source's subscription and notification URI	The target NWDAF and the analytics consumer: that after the transfer is accepted, the subscription the target holds and the notificationURI it will notify are the ones the source NWDAF sent in nwdafEvSub, which must contain the original notificationURI.	The transfer body carries nwdafEvSub, which shall contain the notificationURI (s0013, s0152 NOTE) and the target executes the transfer (s0022); the documents do not state a rule binding the notificationURI to the consumer beyond consumerId (s0014).	s0013, s0152, s0022, s0014
S04.9	No notification under a stale transfer context	The target NWDAF and the analytics consumer: that after a transfer request is cancelled or updated, in-flight notifications are not delivered under the removed or replaced transfer context.	A cancelled transfer removes the resource and the data no longer needed and unsubscribes data sources (s0045, s0046, s0047); the documents do not state a rule for in-flight notifications.	s0032, s0036, s0045, s0047
Z03.10	Malicious NF cannot transfer another's subscription	The analytics consumer named by consumerId and the target NWDAF: that an NF instance holding its own valid credentials and transfer scope cannot submit a transfer request naming another consumer's subscription/consumerId and have the target accept it.	consumerId is the NF instance identifier of the consumer of the subscription to be transferred (s0014); the transfer scope is a service-level scope granting the transfer operations (s0184), and the documents warn that a service-level scope must not give undue consumers write or delete access (ts-29-501 1795, 1801) and that the resource server must check the additional scope against the operation (ts-33-501 8040).	ts-29-501:1801, ts-33-501:8040, s0014, s0184
X01	Transfer request data confidential on the network	The source NWDAF, the target NWDAF and the analytics consumer: that the transfer request body (subscription, consumerId, notificationURI) and the access token that travels with it are not readable by an attacker who has only the network.	Access token credentials must be kept confidential in transit over TLS (rfc6749 10.3, P011-P015); the transfer body carries the analytics subscription and its notificationURI (s0013, s0152) and the baseline protects the service request with TLS (ts-33-501 8012, ts-29-500 4994).	rfc6749:2859, ts-33-501:8012, s0013, s0152

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: 29. Not taken: 369. 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	Introductory text of the OAuth2 security considerations; it states that the measures depend on the client type and grant type and gives no requirement on this protocol.
P003 to P006	rfc6749:2790-2823	Z01, Z03, `malicious_nf`	
P007 to P010	rfc6749:2827-2850	Z03, Z03.10, `malicious_nf`	
P011 to P015	rfc6749:2859-2881	Z01, Z03, X01, `network`, `malicious_nf`	
P016 to P033	rfc6749:2885-3003	Outside the scope	Refresh tokens, authorization codes, redirection-URI manipulation and resource-owner password credentials belong to grant types the NWDAF API does not use; it uses the client-credentials grant (s0179).

Paragraphs	Where	Taken up as	Why not
P034 to P036	rfc6749:3013-3028	D01, Z01, X01, `network`	
P037 to P039	rfc6749:3032-3041	The prover cannot be asked	Guessing of client credentials is about the strength of the credential and of the primitive, which the symbolic model does not measure.
P040 to P054	rfc6749:3045-3167	Outside the scope	Phishing, cross-site request forgery, clickjacking, code injection and open redirectors concern end-user browsers; this procedure has no end user and no browser.
P055 to P062	rfc6749:3170-3210	Outside the scope	Misuse of an access token to impersonate a resource owner applies to the implicit grant, which the NWDAF API does not use.
P063	ts-33-501:2491-2491	The prover cannot be asked	Bidding down of security features concerns algorithm negotiation; the symbolic model fixes the primitives and does not measure their strength.
P064 to P213	ts-33-501:2495-2853	Outside the scope	These are requirements on UE subscription authentication, signalling confidentiality/integrity, subscriber privacy and RAN/UDM/AUSF/UDR behaviour (clauses 5.1-5.8); this task models only the NWDAF-to-NWDAF service operation, not the 5G access network.
P214	ts-33-501:2859-2859	Outside the scope	Trust zones and NDS/IP protection of messages crossing trust boundaries are a deployment property; the model assumes direct NF-to-NF communication protected by TLS.
P215 to P220	ts-33-501:2865-2875	Z01, Z03, A03, A04, D01, S05, `malicious_nf`	
P221 to P227	ts-33-501:2879-2891	Z01, `malicious_nf`	
P228 to P314	ts-33-501:2895-3081	Outside the scope	Requirements on the NEF, SCP, SEPP, roaming intermediaries, attribute protection and IPUPS concern entities and paths this task leaves out; the model assumes direct communication without SCP/SEPP.
P315 to P318	ts-33-501:3085-3091	No goal stated	The 3gpp-Sbi-Originating-Network-Id header and its configured handling state no property that must hold or can be attacked for the transfer procedure.
P319 to P337	ts-33-501:3097-3141	No goal stated	Signalling-monitoring and security-visibility/configuration requirements are operational capabilities, not a property of the transfer messages.
P338 to P369	ts-33-501:3149-3215	The prover cannot be asked	Ciphering and integrity algorithm identifiers and algorithm selection are the strength of primitives, which the symbolic model does not measure.
P370 to P398	ts-33-501:3219-11901	Outside the scope	Requirements on 5G-RG, NSSAAF, the N3 interface, EAP-TLS, IAB, MBSF/MBSTF and NEF concern other subsystems, not the NWDAF analytics subscription transfer.

The catalogue of properties, entry by entry

The properties were drafted from a catalogue of security and privacy properties. The draft left out 35 of its entries, and the analyst went through each of them: examined here as a property, or left out with the reason. Examined: 14. Left out: 21. That the documents do not speak of a property is not a reason the analyst may give.

Entry	Property of the catalogue	Examined as	Why not
K01	Key/evidence confidentiality	X01	

Entry	Property of the catalogue	Examined as	Why not
K02	Load and End-to-End Confidentiality	X01	
K03	Indistinguishability of confidentiality/secretity	Not applicable	The transfer control exposes consumerId and notificationURI as identifiers the target must use; no indistinguishability between alternative secret values is claimed or needed, and the symbolic model would only restate the encryption.
K04	Key uses, directional statements separated	Not applicable	The transfer procedure derives no key of its own; TLS and OAuth2 token keys are per-pair baseline material, and the scope boundary places them outside this analysis.
A01	(Aliveness)	A03	
A02	Weak Consistency Certification	A03	
A05	Secret Key Authentication	Not applicable	The procedure establishes no new session key, so implicit key authentication does not apply.
A06	Visible Key Confirmation	Not applicable	The procedure performs no key confirmation; the 201/204 responses are HTTP acknowledgements, not proof of possession of a key.
A07	Identity - Certificate - Key binding	Z03	
A08	Parameters for consultations tied to security context	S04	
R01	Certification/challenging novelty	Not applicable	The transfer procedure has no challenge, nonce or timestamp exchange; freshness is carried by the transferId resource and by the token expiry.
R02	Message Resume	A04	
R03	Operations at most once	A04	
R04	Message acceptance order/window security	S05	
R05	Anti-degrading/security strategy maintained	Not applicable	The transfer procedure has no version, suite or security-level negotiation that could be lowered.
C01	Forward confidentiality (FS)	Not applicable	The procedure establishes no session key, so forward secrecy does not apply.
C02	Anti-key disclosure fake (KCI)	Not applicable	There is no key establishment between the parties in the transfer procedure, so key-compromise impersonation does not apply.
C03	Known key security/dissemination quarantine	Not applicable	The procedure has no key hierarchy whose keys could be isolated.
C04	Temporary secret/partial state leaks resilience	Not applicable	No ephemeral secret is established by the procedure; the only stored state is the transfer resource, covered by S04/S05.
C05	Security restored after leak (PCS)	Not applicable	The procedure has no ratchet or rekeying, so post-compromise security does not apply.
C06	Enhanced authentication after disclosure of a certificate of registration	Not applicable	No PAKE or registration record is involved.
C07	Anti-offline dictionary attack	Not applicable	No password or human-chosen secret is used; NF credentials are cryptographic keys, and offline dictionary attacks do not apply.
C08	Specify the target to be maintained under the inactivation	The prover cannot be asked	Resilience when a named primitive fails concerns the strength of the cryptographic algorithms, which the symbolic model fixes and does not measure.

Entry	Property of the catalogue	Examined as	Why not
Z02	Delegation does not extend	S04	
Z04	Avoidance and expiry	Z01, S05	
Z05	Cross Tenant/Interservice Safety	Z03.10	
S01	Encryption uniqueness	Not applicable	The transfer procedure uses no nonce-sensitive primitive of its own; TLS record protection is baseline.
S02	Re-transmit/key weight installation security	A04	
S03	Anti-roll and state monotony	S05	
P01	Identity anonymity	Not applicable	NF identities are authenticated and visible in this procedure; no anonymity is claimed.
P02	Interconnectedness of sessions	Not applicable	The transfer procedure makes no unlinkability claim between sessions.
P03	User/evidence Existence Hide	Not applicable	The transfer resource is addressed by an opaque transferId that its creator already holds; there is no user or credential existence lookup that a response could enumerate.
V01	Conditional progress/ error recovery	The prover cannot be asked	Conditional progress and recovery are liveness questions; ProVerif answers trace properties, not whether a run always completes.
V02	No unexpected death locks.	The prover cannot be asked	Deadlock freedom is a liveness question that a trace-property prover cannot be asked.
V03	Abuse of restricted resources/anti-magnification	The prover cannot be asked	Resource abuse and amplification are cost questions beyond counting events, which this prover does not measure.

The parties and the protected values, one by one

The host listed the 20 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: 6. Without: 14. 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	Source NWDAF	party	Nothing gained	The source NWDAF is the legitimate sender and already holds the subscription; for the properties asked its compromise changes nothing (the target still stores what the source sends and the request is still authorized), and any redirection it performs is already reachable through the malicious NF.
S02	Objective NWDAF	party	Everything lost by definition	Every property asked about acceptance, token checking, resource state and stored mapping is a property of the target NWDAF; an attacker controlling it can accept and record anything, and the documents ask none of them under a compromised producer.
S03	analytics consumer	party	Nothing gained	The analytics consumer is the resource owner: a compromised consumer transfers its own subscription, which the properties do not forbid.
S04	NRF	party	Everything lost by definition	The NRF is the authorization server; an attacker controlling it can issue any token, so every authorization property is lost by definition and the documents ask none of them without a trusted authorization server.
S05	Malicious and lawful NF	party	`malicious_nf`	

ID	Party or value	Kind	Attacker	Why none
S06	Subscriptions	value	Nothing gained	The analytics subscription is the very object the target must hold and use; reading it does not break any property asked (the added confidentiality goal is about the network attacker, and the malicious NF brings its own subscription content).
S07	consumerId	value	Nothing gained	consumerId is a public NF instance identifier used for routing.
S08	notificationURI	value	Nothing gained	The notificationURI is metadata the target must use; learning it does not by itself break the agreement or authorization properties, and the network confidentiality goal covers its transit.
S09	transReqType	value	Nothing gained	transReqType is a public enumeration value (PREPARE/TRANSFER).
S10	Individual NWDAF Event Subscription Transfer Resource	value	`malicious_nf`	
S11	transferId	value	`malicious_nf`	
S12	OAuth2 Access tokens	value	`malicious_nf`	
S13	scope nnwdaf-eventssubscription:transfer	value	Nothing gained	The scope name is public and stated in the specification (s0184).
S14	Access tokens	value	`malicious_nf`	
S15	Migration of resources	value	`malicious_nf`	
S16	Subscriptions	value	Nothing gained	The subscription content is what the target must hold in order to serve it; the attacker brings its own content in its own request.
S17	Data Sources Subscription	value	Nothing gained	Data-source subscriptions belong to the target's data collection and are outside the modelled transfer flow; reading them breaks no property asked.
S18	Move Request Fields	value	Nothing gained	The transfer request body fields are the message content the target must process; the attacker can put its own values in its own request, which is the malicious NF exposure already named.
S19	Subscriptions	value	Nothing gained	The subscription is the object the target must hold; this is the same exposure as S06.
S20	Notice in transit	value	Nothing gained	In-flight notifications are protected to the consumer by TLS; the property asked about them is which transfer context they belong to (S04.9), not their secrecy.

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		

Attacker	Has	Basis	Where
`malicious_nf`	A legitimate NF instance (a different instance from the source NWDAF, the target NWDAF and the analytics consumer) that holds its own TLS credentials and its own valid OAuth2 access token for the nnwdaf-eventssubscription:transfer scope with the target NWDAF as audience; it can open TLS to the target and send service requests.	Task scope boundaries require distinguishing a malicious legitimate NF with its own credentials; the transfer scope grants access to the transfer operations (s0184) and the baseline issues tokens from the NRF (ts-33-501 8008, 8010).	

ID	`network`	`malicious_nf`
A03	Holds	Fails
A04	Holds	Holds
Z01	Holds	Holds
Z03	Holds	Fails
S04	Holds	Holds
S05	Holds	Holds
D01	Holds	
A03.8	Holds	Fails
S04.9	Holds	Holds
Z03.10		Fails
X01	Holds	

Notes on the prover and the model

- A04: answered in ProVerif, and the analyst names changing_state for this target
- S05: answered in ProVerif, and the analyst names changing_state for this target
- S04.9: answered in ProVerif, and the analyst names changing_state for this target

A ProVerif model that simplifies algebra (xor, dh_algebra, other_algebra) gives the attacker less than the real operation: a proof there is a proof against a weaker attacker, an attack there is an attack. A ProVerif model that leaves out a check on a value that changes (changing_state) lets parties accept more: a proof there holds, an attack there may be nothing but the missing check. Tamarin+ Agent models these as they are.

A bound in steps is an answer on bounded.pv: the analyst's second ProVerif model, in which the party that keeps a value that changes is unrolled to that many steps with its check in place. That model does less than the protocol does, so the property holds for those steps and no further, and an attack found there is an attack. The number of steps is the analyst's statement.

State review by the analyst

This part is the analyst's own reading of the source, not an answer of the prover. "yes" under shown by ProVerif means that ProVerif found a trace with these steps in the unrolled model, so the steps are possible; that they break a target is said only by an attack in the table of answers, and what else they cost is the analyst's reading. "nothing found" says which patterns were looked at and proves nothing.

value	kept by	changes when	checked when	refs
Individual NWDAF Event Subscription Transfer resource, identified by transferId (with its stored subscription, consumerId and notificationURI)	target NWDAF	created on POST PREPARE (s0021), replaced on PUT PREPARE (s0032), removed on PUT TRANSFER execution (s0036, s0037) and on DELETE cancel (s0045, s0047)	on every PUT and DELETE addressed to /transfers/{transferId}, the target must find the resource it is asked to update or remove (s0030, s0044); ts-33-501 2875 requires validating an incoming message against network state	s0021, s0030, s0032, s0036, s0037, s0038, s0044, s0045, s0047, s0049
Mapping from the analytics subscription (nwdafEvSub) to the notificationURI to notify	target NWDAF	written from the transfer request body when the resource is created or updated (s0013, s0152 NOTE); no longer needed after a cancel, which deletes data and unsubscribes (s0046)	when the target delivers notifications to the consumer at the transferred notificationURI (s0152 NOTE)	s0013, s0014, s0046, s0152
Position in the request lifecycle kept by the source NWDAF (prepared, executed or cancelled) together with the transferId it received	source NWDAF	when it receives the 201 for a PREPARE (s0021), the 204 for a TRANSFER execution (s0036), or the 204 for a cancel (s0048)	before it sends the next request for the same transfer request (s0030, s0044)	s0021, s0030, s0036, s0044, s0048

pattern	result	target	shown by ProVerif	steps
replay	nothing found			A TRANSFER replayed after the resource was executed or cancelled is rejected by the resource check: bounded.pv verifies s05 (no execution after removal) and its control s05_ctl falsifies it, and a04 shows at most one execution per transferId. A replayed POST PREPARE is accepted, but it simply creates a new transfer request with a new transferId, which is the POST's normal semantics and breaks no target.
repeat_one_time_action	nothing found			bounded.pv a04 verifies that one transfer request is executed at most once; the control a04_ctl, with the check left out, shows two executions, so the property depends on the resource-state check.
desynchronisation	candidate		no: not tried with ProVerif	The source sends PUT /transfers/{transferId} with transReqType TRANSFER (s0035). / The target executes the transfer and removes the resource (s0036), then sends 204 No Content. / The network drops the 204. / The source retransmits the same PUT; the target finds no resource and rejects it (s0038, ts-33-501 2875). / The source now believes the transfer did not execute while the target executed it and removed the resource; the source cannot tell an execution from a cancellation with the same transferId (s0045, s0047, s0048).
update_before_check	nothing found			The target verifies the access token and the resource before it changes the resource: ts-33-501 8014, 8038, 8046 verify the token before the service is executed, and s0031-s0036 act on the resource identified by transferId. bounded.pv models the token check before the create and before each change.
no_rollback_on_failure	candidate		no: not tried with ProVerif	A TRANSFER request is accepted only partly: the PartialAnalyticsSubTransfer feature is supported and not all analytics events are accepted (s0024, s0034, s0037). / The target still removes the Individual NWDAF Event Subscription Transfer resource after sending the 201/200 with failTransEventReports (s0024, s0037). / The resource is gone, so the consumer cannot retry or complete the failed events under the same transfer request; the source text defines no rollback to the prepared resource.

pattern	result	target	shown by ProVerif	steps
race_between_sessions	candidate	A04	no: not tried with ProVerif	Two transfer requests for the same analytics subscription are created, each with its own transferId (s0021, s0022). / Both are executed, and the target removes each resource after its own execution (s0036). / The target now runs two data collections and two notification streams for the same subscription; nothing in s0020-s0024 or s0031-s0037 rejects the second request because the check is per transferId, not per subscription.
distinguishable_failure	nothing found			The target does return different codes (201/204, 307/308 redirects, and the error codes of 5.1.7), but a transferId is an opaque handle returned only inside the creator's TLS session (s0105, s0117), so a failure code does not reveal a consumer's identity or a value a target asks about. No target depends on it.
wrap_or_window	nothing found			The changing values are an opaque transferId string (s0117) and the request lifecycle; there is no numeric counter or acceptance window, so nothing can wrap or fall outside a window.

What the analyst named for targets:

step	targets	features	reason
33	A04, S05, S04.9	changing_state	These three targets rest on the Individual NWDAF Event Subscription Transfer resource, a value that changes across steps: it is created by PREPARE, removed by a TRANSFER execution or a DELETE cancel, and in-flight notifications may still be delivered afterwards. ProVerif cannot track a resource table for any number of sessions, so model.pv binds each transfer to one linear session and these targets are unrolled in bounded.pv.
33	A03, Z01, Z03, Z03.10, D01, A03.8, S04, X01	plain_primitives, many_sessions	TLS is modelled as a pairwise authenticated encryption with only its plain behaviour, the access token as an unforgeable NRF signature, and the answer is asked for any number of sessions. No algebraic law beyond a shared-key encryption is involved.

The analyst's closing note

The 11 targets are answered. Against the network alone the transfer control does what it should: field integrity and agreement with the source are proved (D01, A03 first claim), the OAuth2 token checks are proved (Z01), the token is bound to the caller and the target (Z03 first claim), the executed context belongs to its own transfer resource (S04), the token/subscription/notificationURI are confidential (X01), and the unrolled target rejects a retransmitted or replayed transfer after execution or cancel (A04, S05, S04.9, each bounded to 4 steps with a falsified control).

The finding: a legitimate NF with its own valid transfer-scope token gets the target NWDAF to accept and execute a transfer naming another consumer's subscription, and the target records the attacker's notificationURI for that consumer (Z03.10, Z03 second claim, A03 second claim, A03.8, all holds_in_source under the malicious_nf attacker). The target checks the token's subject, audience and scope but nothing ties consumerId or the subscription to the caller. The documents only warn in general that a service-level scope must not grant undue write access (ts-29-501 1801, ts-33-501 8040), so the novelty is unclear rather than recorded. check_fix on fix-binding.pv shows the proposed fix (per-consumer scope in the token plus consumerId comparison at the target) removes the attack and keeps the normal run.

Unconfirmed findings from the state review: a lost 204 desynchronises the source from the target, a partial accept still removes the resource, and two transfer requests for the same subscription each execute once. Everything rests on the listed simplifications: fixed primitives, no token expiry or revocation, a trusted NRF, and a target unrolled to four steps — Tamarin+ Agent would settle the state questions.