

WEBHOOK & EVENT-DRIVEN SECURITY.



A webhook flips the usual direction of trust — **the receiver has to authenticate a push it never asked for, and the sender has to make sure that push can't be pointed anywhere it shouldn't reach.**

- RAW-BODY SIGNATURE VERIFICATION
- REPLAY & TIMESTAMP PROTECTION
- CALLBACK URL ALLOWLISTING
- IDEMPOTENT EVENT PROCESSING

01 THE MECHANISM



Signature Verification Gaps

A webhook's HMAC signature has to be verified against the exact raw request bytes, using a constant-time comparison — **re-serializing the payload first, or comparing signatures with normal string equality, quietly breaks the control the scheme depends on.**



Callback URL as an SSRF Vector

Letting a user register any URL to receive webhook events turns the sending server into a request proxy on demand — **a registered internal address gets the sending server to make that request for the attacker from inside its own network.**



Replay of a Captured Payload

A correctly signed webhook event, once captured, stays valid and replayable indefinitely unless the receiver separately checks for it — **a signature proves the payload wasn't altered, not that it's being seen for the first time.**

02 THE FLOW

The Verified, Idempotent Receiver

CONTAINED

- Every incoming webhook's signature is recomputed over the exact raw request body and compared in constant time before the payload is trusted.
- A timestamp inside the signed payload is checked against a tolerance window, and each event's unique ID is tracked, so **a replayed or duplicate event is rejected outright.**
- Processing an event twice, whether from a legitimate retry or a replay, produces the same end state, so **duplicate delivery never doubles a financial or state-changing effect.**

The Trusting, Non-Idempotent Receiver

EXPOSED

- Incoming payloads are processed without a recomputed signature check, or the check compares against a reformatted body that no longer matches the original bytes.
- No timestamp or event-ID tracking exists, so **a captured, valid webhook payload can be resent by an attacker at any time and is processed as new.**
- Reprocessing the same event applies its effect again, so **a retried or replayed "payment succeeded" event can double-credit an account** or repeat any other state change.

03 CORE PILLARS & COVERAGE

PILLAR	WHAT IT COVERS
Signature Verification on Raw Payload	Recomputes the HMAC or signature over the exact, unmodified request body and compares it in constant time, rather than trusting a reformatted or partially checked payload.
Timestamp & Replay Protection	Rejects any webhook outside a defined age window and tracks processed event IDs, so a captured payload can't be resent and accepted as new.
Callback URL Allowlisting	Restricts registrable webhook destinations to public, non-internal addresses, closing the SSRF path that lets a callback URL reach internal infrastructure.
Idempotent Event Processing	Designs every event handler to produce the same result whether an event arrives once, twice, or out of order, matching the at-least-once delivery most event systems provide.

04 DETECTION APPROACHES COMPARED

APPROACH	SUFFICIENT ALONE?	WHAT IT MISSES
No Signature Verification	No	Accepts any payload that arrives at the endpoint, with no way to distinguish a legitimate event from a forged one.
Signature Checked Against a Re-Serialized Body	No	Fails or is silently bypassed whenever the parsed-and-reformatted payload no longer matches the bytes the sender actually signed.
Unrestricted Callback URL Registration	No	Lets a registered internal or link-local address turn the sending server into an SSRF proxy against its own network.
No Timestamp or Event-ID Deduplication	No	Leaves a captured, validly signed payload replayable indefinitely, with no way to tell a replay from a first delivery.
Raw-Body Verification + Replay Protection + Callback Allowlisting + Idempotent Handlers	Yes	Closes the forgery, SSRF, replay, and duplicate-processing paths together rather than any one alone.

WEBHOOK & EVENT-DRIVEN SECURITY.

VERIFY · RESTRICT · DEDUPLICATE

05 PROGRAM OPERATIONS

A webhook signature only proves integrity, not freshness. Verifying the signature confirms the payload wasn't altered; it says nothing about whether this exact payload has already been processed once before.

Letting users register arbitrary callback URLs is a server-side request forgery feature, not a bug waiting to be found. The sending server will faithfully deliver a request to whatever address it's given.

Event-driven systems are built on at-least-once delivery, not exactly-once. A receiver that assumes each event arrives exactly one time, in order, will eventually mishandle a duplicate or out-of-order delivery.

Signing secrets need a rotation path, not just an initial value. A scheme that can't accept two valid secrets during a transition period forces an all-at-once cutover that risks dropping legitimate events.

06 CONTROL MAPPING

OWASP SSRF PREVENTION CHEAT SHEET

Guidance on validating and restricting outbound request destinations, directly applicable to webhook callback URL registration.

STANDARD WEBHOOKS SPECIFICATION

An open, vendor-neutral specification for webhook signing, verification, and event ID conventions.

HTTP MESSAGE SIGNATURES (RFC 9421)

An IETF standard for signing HTTP requests, offering a general alternative to vendor-specific HMAC signing schemes.

CLOUDEVENTS (CNCF)

A standardized event envelope format for describing event metadata consistently across event-driven architectures.

07 WHAT THIS DOES NOT SOLVE

Receiver-Side Application Vulnerabilities

Verifying that a webhook came from the right sender doesn't fix a vulnerability in how the receiving application processes the event payload itself.

Sender-Side Compromise

No amount of receiver-side verification protects against an attacker who has compromised the legitimate sending system and can produce validly signed events.

Downstream Data Handling

Confirming a webhook's authenticity doesn't govern what the receiving system does with the data afterward, including how long it's retained or who else can access it.

Availability of the Receiving Endpoint

Signature and replay protection don't address an endpoint that's simply unreachable or overwhelmed, which is a resilience problem, not an authenticity one.

08 MYTHS VS. REALITY

MYTH

"HTTPS delivery alone makes a webhook secure."

REALITY

TLS protects the payload in transit, but **doesn't verify who actually sent it** — without a signature check, any party that can reach the endpoint can submit a forged event over HTTPS just as easily.

MYTH

"If the signature is valid, the event is safe to process."

REALITY

A valid signature only proves the payload wasn't altered — **a captured, previously valid payload replayed later still carries a valid signature.**

MYTH

"Letting customers register their own webhook URLs is just a convenience feature."

REALITY

An unrestricted callback URL registration is **a built-in SSRF vector**, letting a registered internal address turn the sending server into a proxy against its own network.

MYTH

"Retrying a failed webhook delivery is always safe."

REALITY

A retry that reprocesses an already-applied event **can double a financial or state-changing effect** unless the handler is specifically built to be idempotent.

MYTH

"Webhooks and polling APIs need the same security controls."

REALITY

A webhook flips the direction of trust — **the receiver has to authenticate an inbound push it didn't request**, a different problem than authenticating outbound calls it initiated itself.

09 SPEAK THE PROGRAM — KEY TERMS

HMAC SIGNATURE VERIFICATION

Recomputing a cryptographic signature over the **exact raw payload bytes** and comparing it to the sender's signature to confirm authenticity.

REPLAY ATTACK

Resending a **previously valid, correctly signed payload** to trick a receiver into processing it again as if it were new.

SSRF (SERVER-SIDE REQUEST FORGERY)

Tricking a server into **making a request to an unintended destination**, such as an internal address, on the attacker's behalf.

IDEMPOTENT PROCESSING

Designing an event handler so that **receiving the same event multiple times produces the same result** as receiving it once.

AT-LEAST-ONCE DELIVERY

A delivery guarantee where an event **may be delivered more than once** but never silently dropped, common to most webhook and message-queue systems.

CALLBACK URL ALLOWLISTING

Restricting which **destinations a registered webhook can point to**, blocking internal, link-local, or otherwise unintended addresses.