SAMPLE - Stripe Webhook Failure Drill

> **SAMPLE ONLY - NOT A CLIENT REPORT**
>
> This document demonstrates the format of the $950 Stripe Webhook Failure Drill. `ExampleCo`, every identifier, timestamp, result, finding, and measurement below are fictional and synthetic. Nothing here describes a client, a production system, or a completed engagement.

Engagement Snapshot

Item	Illustrative scope
Integration	One Stripe test-mode integration
Handlers	Checkout provisioning, subscription lifecycle, invoice payment
Environment	Staging only; read-only code and log review
Business invariants	One purchase creates one entitlement; cancellation cannot be undone by an older event; one invoice grants credits once
Evidence window	Synthetic example: 2026-01-15 14:00-16:00 UTC
Overall result	**SAMPLE FAIL** - four of eight illustrative controls failed

The release gate fails in this example because concurrency and event ordering can change paid access or credits incorrectly. A real drill reports only what is observed in the agreed staging scope.

Pass/Fail Evidence Matrix

All evidence references and counts in this matrix are illustrative.

ID	Failure mode	Expected invariant	Illustrative evidence	Result
DRILL-DUP-01	Duplicate delivery	Repeating one event ID produces one entitlement write and one side effect.	`evt_sample_dup_01` delivered 3 times; processed-event rows: 1; entitlement writes: 1; trace: `trace-sample-dup-01`.	**PASS**
DRILL-ORDER-01	Out-of-order delivery	An older lifecycle event cannot overwrite newer subscription state.	`customer.subscription.deleted` applied before an older `customer.subscription.updated`; final access: `active`; expected: `canceled`.	**FAIL**
DRILL-5XX-01	Timeout and 5xx	A failed attempt makes no partial change and remains recoverable by retry.	Attempt 1 returned 500 after forced rollback; attempt 2 returned 200; final entitlement writes: 1; partial writes after attempt 1: 0.	**PASS**
DRILL-REPLAY-01	Replay	Provider or operator replay cannot repeat an irreversible side effect.	Original plus dashboard replay kept one entitlement row but emitted 2 welcome notifications for `evt_sample_replay_01`.	**FAIL**
DRILL-SIG-01	Invalid signature	Bad, stale, or malformed signatures fail closed without state change.	Three rejected requests returned 400; handler writes: 0; structured reason recorded for each rejection; no signature material retained.	**PASS**
DRILL-CONC-01	Concurrency	Simultaneous copies of one event cannot bypass idempotency or transaction boundaries.	Two synchronized `invoice.paid` requests created 2 credit grants and `+200` credits; expected: 1 grant and `+100`.	**FAIL**
DRILL-QUEUE-01	Queue failure	The endpoint acknowledges only after durable acceptance; a rejected event remains retryable.	Queue unavailable: 503 and 0 accepted rows; recovery retry: 200 and 1 queued row; worker applied the event once.	**PASS**
DRILL-RECON-01	Reconciliation	A missed event is detected and repaired from Stripe's source of truth.	One staged subscription was intentionally drifted; no scheduled comparison, repair record, or alert detected it during the sample window.	**FAIL**

Result Summary

- Passed: 4/8
- Failed: 4/8
- Highest-risk paths: concurrent invoice handling and subscription ordering
- Release recommendation: hold the affected handlers until F-01 and F-02 are fixed and the failed cases pass in staging

Ranked Illustrative Findings

SAMPLE ONLY - no client or production results

F-01 - High: Concurrent invoice events can grant credits twice

Business impact: a single paid invoice can create duplicate usage credits or fulfillment when two deliveries overlap.

Evidence: DRILL-CONC-01 produced two credit-grant rows for one synthetic event ID. The handler checked for an existing event before opening the credit transaction, leaving a race between the check and insert.

Minimal correction: enforce a database uniqueness constraint on the event and effect identity, then create the processed-event record and credit grant in one transaction. Treat a uniqueness conflict as an already-applied success.

F-02 - High: Older subscription events can restore canceled access

Business impact: a canceled account can regain paid access when an older update arrives late.

Evidence: DRILL-ORDER-01 ended in `active` after the newer cancellation had already been applied. The sample handler accepted lifecycle events without a monotonic version check or source-of-truth refresh.

Minimal correction: reject stale transitions using stored event/object version data and fetch the current Stripe subscription before applying an ambiguous transition.

F-03 - Medium: Operator replay repeats a customer notification

Business impact: support-triggered replay can send duplicate onboarding or receipt messages even when database state is idempotent.

Evidence: DRILL-REPLAY-01 preserved one entitlement row but emitted two synthetic welcome-notification records.

Minimal correction: route replays through the same idempotency boundary as live delivery and record each external effect in a transactional outbox keyed by provider event ID and effect type.

F-04 - Medium: No reconciliation control detects missed entitlement events

Business impact: delivery loss or manual database drift can leave billing and access inconsistent without an alert.

Evidence: DRILL-RECON-01 introduced one synthetic mismatch and observed no comparison job, repair record, or alert during the sample window.

Minimal correction: schedule a read-only comparison against Stripe, emit a drift metric and alert, and require an explicit reviewed action before repair.

Reproduction Outline

This outline intentionally omits credentials, raw signatures, and environment-specific commands.

1. Confirm test mode, the three in-scope handlers, expected state transitions, and rollback contacts.
2. Record a baseline of processed-event, entitlement, credit, queue, and outbox rows using synthetic Stripe objects.
3. Submit duplicate and deliberately reordered fixtures while recording request IDs, response codes, handler logs, and before/after state.
4. Inject one handler rollback, one queue outage, and controlled retries; verify acknowledgement happens only after durable acceptance.
5. Submit malformed and stale-signature requests; verify rejection and zero mutation without recording secret material.
6. Release two synchronized copies of one invoice event, then replay a completed event through the normal operator path.
7. Create one known Stripe-to-local mismatch and run the documented reconciliation procedure, if one exists.
8. Restore staging, rerun the baseline checks, and package redacted traces, query results, and log excerpts under the matrix IDs above.

Minimal Fix Plan

Effort is illustrative and must be re-estimated after reviewing the real code, schema, deployment path, and change-control requirements.

Order	Change	Covers	Illustrative effort	Verification gate
1	Add database-enforced effect uniqueness and a single transaction for event plus credit state.	F-01	0.5-1.0 engineering day	DRILL-DUP-01 and DRILL-CONC-01 pass under repeated runs.
2	Add stale-event protection and source-of-truth refresh for ambiguous subscription transitions.	F-02	1-2 engineering days	DRILL-ORDER-01 converges to the current Stripe state.
3	Put replayed external effects behind the live idempotency/outbox path.	F-03	0.5-1.0 engineering day	DRILL-REPLAY-01 emits each effect once.
4	Add scheduled reconciliation, drift metrics, alerting, and reviewed repair.	F-04	1-2 engineering days	DRILL-RECON-01 detects the seeded mismatch within the agreed window.

Implementation is separate from the drill and starts only after scope and findings are approved.

Limitations And Boundaries

SAMPLE ONLY - no client or production results

- This sample contains no real customer data, system access, logs, source code, test execution, or verified results.
- A paid drill covers one Stripe integration, up to three handlers, staging, and read-only code and log review after access and inventory are ready.
- The drill does not mutate production or request production signing secrets.
- It is not a penetration test, compliance certification, migration, or uptime guarantee.
- Results are point-in-time evidence for the exercised fixtures and observed configuration, not proof that every billing path is defect-free.
- Production rollout, data repair, and implementation are outside the drill unless separately approved.
- The delivered engagement also includes a 45-minute engineering readout; this static sample does not represent that discussion.

Sample status: **fictional, redacted, and not a client report.**