RRaffle

AUDITOR-READY DRAWING DOCUMENTATION

Randomness and Audit Methodology

A plain-language explanation of how Raffle locks the ticket population, commits to secret randomness before a drawing, selects tickets without modulo bias, and produces evidence that can be retained and independently reviewed.

SAMPLE PACKAGE FOR COALITION FOR KIDS
Prepared for the October 1, 2026 annual reverse raffle and auction. This document and the companion workbook demonstrate the reporting format; they do not record an actual drawing.

800	795	16	5
tickets locked	eliminations logged	$125 milestone prizes	grand-prize recipients

What the organization can retain

The audit export provides a summary of the drawing identity and cryptographic evidence, the complete locked ticket population, every event and ticket elimination in exact order, and a consolidated list of milestone and grand-prize winners. The companion CSV draw log supports reviewers who prefer a flat-file format.

Important scope statement

Raffle describes this design as **NIST-aligned** because it follows the public-auditability pattern NIST describes: commit before randomness is used, specify a deterministic operation, then reveal enough evidence for others to verify the result. Raffle does not claim NIST certification, validation, or approval, and version 1.0 does not consume the NIST Randomness Beacon.

Methodology version 1.0 | Prepared August 3, 2026

How the random pull works

The process separates evidence created before the first ticket is pulled from evidence revealed after the outcome is fixed. That separation is what lets an auditor test both unpredictability and reproducibility.

Stage	What Raffle records	Why it matters
1. Lock the population	An ordered snapshot of every eligible ticket and its split election, plus a SHA-256 hash of that snapshot.	A later addition, deletion, edit, or reordering changes the hash.
2. Commit before activation	A 32-byte secret server seed generated by PHP random_bytes(), plus SHA-256(server seed). Only the commitment is exposed before the drawing.	The commitment binds the system to the seed without disclosing it early.
3. Add public ceremony input	A value entered at the event is combined with the committed seed and locked pool hash.	The seed is tied to the public ceremony and the exact ticket population.
4. Select tickets	HMAC-SHA256 derives a candidate value for each draw position and current available-pool hash.	Every selection depends on the prior state and can be reproduced later.
5. Remove modulo bias	Values outside an even multiple of the current pool size are rejected and retried.	Each remaining ticket receives an equal-size region of the accepted value space.
6. Chain the event log	Each event hash includes the previous event hash, timestamp, actor, ticket, event type, and payload.	Insertion, deletion, reordering, or modification becomes detectable.
7. Reveal after completion	The server seed is disclosed in the completion event and audit summary.	An auditor can recompute the commitment, final seed, selections, and event chain.

Version 1.0 formulas

```
ticket_pool_hash = SHA-256(canonical JSON of locked ticket snapshot)
```

```
seed_commitment = SHA-256(server_seed)
```

```
final_seed = SHA-256(canonical JSON of public_entropy, server_seed, ticket_pool_hash)
```

```
proof = HMAC-SHA256(final_seed, draw_position + available_pool_hash + attempt)
```

```
selected_index = accepted_first_52_bits(proof) modulo available_ticket_count
```

Reports and independent verification

The report set is designed to answer the questions a finance reviewer typically asks: What was eligible? What happened, in what order? Who won each prize? Could the outcome have been changed without leaving evidence? Can another party reproduce the selection?

Report	Contents	Primary audit use
Audit Summary	Draw ID, status, algorithm/version, timestamps, ticket-pool hash, seed commitment, public entropy, revealed seed, final event hash, and integrity result.	Establish the drawing identity and the evidence needed to verify it.
Locked Ticket Pool	Every imported ticket in deterministic order, including ticket number, participant name, and grand-prize split election.	Reconcile the drawing population to RallyUp or another source export.
Complete Event Log	Every ceremony event and elimination, exact sequence, draw position, ticket, milestone flag, selection proof, prior hash, current hash, and JSON details.	Reconstruct the drawing and test the append-only chain.
Prize Winners	First-draw and every-50th milestone winner, five-way split recipients or final grand-prize winner, ticket identity, and split election. The sample adds event-specific amounts and disposition status.	Reconcile prize obligations and award records.
CSV Draw Log	Flat export of the complete event log.	Archive or analyze the draw without spreadsheet software.

Recommended auditor procedure

1	Reconcile the 800 locked tickets to the final RallyUp export and investigate any missing or duplicate number.
2	Confirm the draw-committed timestamp and seed commitment precede activation and the first elimination.
3	Hash the revealed server seed with SHA-256 and compare it to the pre-draw commitment.
4	Hash the canonical locked ticket snapshot and compare it to the reported ticket-pool hash.
5	Re-run algorithm version 1.0, checking every selected ticket and selection proof in sequence.
6	Recompute every event hash and confirm each previous-hash link and the final event hash.
7	Confirm draw 1 and draws 50, 100, 150, and so on appear on the Prize Winners report.
8	Confirm the final grand-prize outcome agrees with the remaining ticket count and locked split elections.

Event-day controls and retention

Cryptographic evidence is strongest when paired with ordinary operational controls. The organization remains responsible for validating its source data, documenting prize decisions, and retaining the exports under its own policy.

When	Recommended control	Evidence to retain
Before the event	Finalize ticket sales; export RallyUp; resolve duplicate or missing ticket numbers; capture split elections; identify authorized operators.	Source-platform export, reconciliation sign-off, operator list, event rules.
At draw commitment	Display or record the draw ID, ticket count, ticket-pool hash, seed commitment, and committed timestamp.	Screenshot, signed event note, livestream/video if used.
At activation	Enter ceremony entropy that is visible to attendees and retain its exact spelling and punctuation.	Activation event, photo/video, event script.
During the draw	Use pause/resume controls rather than restarting. Record any interruption and the reason outside the application if material.	Event log plus incident note.
At milestone awards	Confirm the ticket at draw 1 and each 50th position; record whether each $125 recipient keeps or donates the award.	Prize winner report, signed prize/donation acknowledgment.
At grand-prize resolution	At five remaining tickets, apply the published split rule exactly; continue if any remaining election is No or not provided.	Locked split elections and completion event.
After completion	Export XLSX and CSV promptly, save read-only copies, calculate file checksums, and restrict later changes.	Export files, checksums, retention location, reviewer sign-off.

Source references

NIST, Interoperable Randomness Beacons - Applications: https://csrc.nist.gov/projects/interoperable-randomness-beacons/apps

NIST FIPS 180-4, Secure Hash Standard: https://csrc.nist.gov/pubs/fips/180-4/upd1/final

PHP Manual, random_bytes: https://www.php.net/random-bytes