# Anti Hunter Staking & Locking Whitepaper (Draft v0.2)

Receipts-funded staking for $ANTIHUNTER: 30/60/90/120-day locks, linear streaming, explicit reward sources, and deterministic control rules.

### TL;DR

- **Lock terms:** 30 / 60 / 90 / 120 days (bonus weight for longer locks)
- **Rewards:** streamed **linearly** (per-deposit schedule)
- **Early exit:** allowed with **25% principal penalty**
- **Penalty routing: 100% to rewards pool** (paper hands subsidize diamond hands)
- **No fake APY:** rewards are **not promised**; yields are receipts-funded and can go to **0**
- **Dynamic buyback:** deterministic rule tied to **moving average of net inflows** (no discretion)

## 0) SCOPE & PRINCIPLES

This document specifies staking/locking mechanics and the control rules that govern rewards and buyback behavior. The design goal is boring, auditable tokenomics: **if you can’t point to a dashboard + a transaction + a diff, it didn’t happen.**

## 1) MOTIVATION

$ANTIHUNTER is a receipts-first, on-chain venture desk token. Tokenomics should be driven by **real inflows** and **verifiable on-chain receipts**—not narrative APY. This staking design aims to: (a) reward commitment, (b) reduce reflexive dumping, and (c) keep the mechanism simple enough to audit.

## 2) SYSTEM OVERVIEW

The staking system is a single on-chain vault coordinating three flows:

- **Stakers deposit** $ANTIHUNTER and choose a lock term: 30 / 60 / 90 / 120 days.
- **Rewards stream** from a funded Rewards Pool over time (linear).
- **Early exits pay a 25% penalty** on withdrawn principal; penalties are routed into the Rewards Pool for remaining stakers.

## 3) LOCK TERMS & BONUS WEIGHTS

Stakers choose a lock term. Longer terms receive a higher **staking weight** (a bonus share of any rewards that are distributed). This is a simple way to reward longer commitment without promising yield.

### Lock terms (v0.2)

- **30 days** — weight `1.0`×
- **60 days** — weight `1.4`×

- **90 days** — weight `1.9`×
- **120 days** — weight `2.5`×

Weights are parameters (auditable on-chain) and may be revised in future drafts; any change must be disclosed and timelocked.

**Other core parameters**

- **Reward duration:** linear streaming (per-deposit schedule)
- **Early unstake penalty:** 25% (2500 bps) on principal withdrawn before maturity
- **Penalty destination:** Rewards Pool (100%)
- **Reward token:** $ANTIHUNTER (initial)

## 4) REWARD SOURCE HIERARCHY (AND NO FAKE APY)

**Rewards are not promised.** There is no fixed or target APY. Rewards are funded in this order:

1. **Fees paid in $ANTIHUNTER** (primary): protocol fees settle directly in $ANTIHUNTER and are routed **100% to the Rewards Pool**.
2. **Early-exit penalties** paid by exiting stakers (routed 100% to the Rewards Pool).
3. **Optional emissions** only if explicitly enabled and within a hard cap (bootstrapping, not permanent).

**Separately:** realized P&L (e.g. WETH settled to treasury) is used for **on-market buybacks of $ANTIHUNTER** under the deterministic rule in Section 7.

Any yield shown in UI/docs must be labeled as either: (a) **realized historical** yield over a stated window, or (b) a **mechanical estimate derived from on-chain inflows**. It must never be presented as guaranteed.

## 5) LOCKING MECHANICS (AVOID SYNCHRONIZED CLIFFS)

To avoid a single “epoch unlock day,” v0.2 uses a **per-deposit rolling schedule** tied to the selected term:

- Each deposit has its own maturity timestamp (now + 30/60/90/120 days).
- Rewards stream linearly over time and do not require synchronized epoch resets.

## 6) CONTINUATION / ROLLOVER (RE-LOCK INCENTIVE)

To reduce mass “end-of-epoch” sell pressure, the protocol supports an opt-in **relock / rollover** at maturity (same or longer term).

### Rollover incentive (A + cap)

- If a user re-locks within a short window after maturity (e.g. 24h), they receive a **temporary bonus weight** on the next lock only.
- **Bonus:** `+0.2`× weight applied to the selected term weight for the next lock.
- **Cap:** total effective weight is capped at `3.0`× (prevents runaway incentives).
- Bonus applies only when rolling to the **same or longer** term, and does not stack across multiple actions within a single term.

This incentive changes distribution share; it does not create promised yield. Rewards remain receipts-funded and may be 0.

## 7) PROGRAMMATIC BUYBACK (FEES IN $ANTIHUNTER + PNL BUYS)

Buybacks should be **non-discretionary**. In this design, **fees are paid in $ANTIHUNTER** (used for staking rewards and/or burns), while **realized P&L** (settled in a settlement asset like WETH) is used to **buy $ANTIHUNTER on-market**.

**Inputs (daily)**

- **Fee inflow:** `fee_AH_t` = $ANTIHUNTER fees received on day `t`.
- **Realized PnL:** `pnl_WETH_t` = WETH (or WETH-equivalent) realized PnL settled to treasury on day `t`.

### Buyback controller (deterministic)

- Signal: compute `MA_t` = N-day moving average of `pnl_WETH` (and optionally fee value converted to WETH via an on-chain price oracle / TWAP).
- Set `buyback_bps_target = clamp(BPS_MIN, BPS_MAX, BPS_MAX ∗ MA_t / (MA_t + K))`.
- Rate-limit: `buyback_bps` can change by at most Δ bps per day.
- Daily spend (WETH): `Budget_t = min((buyback_bps/10,000) ∗ pnl_WETH_t, Free_t, BUDGET_MAX_PER_DAY)`.
- Execution: swap `Budget_t` WETH for $ANTIHUNTER on a whitelisted DEX path, enforce max slippage, emit events.

**Important:** this algorithm controls *how much* realized PnL gets converted into $ANTIHUNTER on-market. It does not guarantee buybacks (if `pnl_WETH_t` is 0, buyback spend is 0).

## 8) EMISSIONS CAP, RUNWAY, AND INFLOWS = 0

If emissions are enabled, they must be bounded by a hard cap (e.g., `EMISSIONS_MAX_PER_DAY` and/or `EMISSIONS_MAX_TOTAL`). The protocol should publish “reward runway” under (i) current inflows and (ii) a conservative stress case where inflows drop to zero.

In the inflows = 0 case, rewards can only be paid from existing reserves and any remaining emissions budget; if those are exhausted, rewards necessarily decline and may pause. This is an explicit design constraint, not a hidden assumption.

## 9) EARLY EXIT PENALTY ACCOUNTING

- **Penalty base:** principal only (not on accrued rewards).
- **Routing:** penalty is transferred to Rewards Pool on-chain.
- **Disclosure:** UI must show “effective cost of exit” at the time of withdrawal.

## 10) ON-CHAIN RECEIPTS & VERIFICATION

Auditability is part of the spec. At minimum, the protocol should expose:

- Canonical contract addresses and treasury addresses.
- Events for: fees received, rewards streamed, penalties collected, buybacks executed.
- Public verification links (e.g., explorer + dashboard) that reconcile to those events.

## 11) RISKS

- **Low inflows → low rewards:** rewards may be small or zero.
- **Strategy risk:** realized PnL can be negative.
- **Smart contract risk:** bugs, exploits, upgrades.
- **Liquidity/market risk:** slippage, volatility, LP dynamics.
- **Penalty severity:** early exit is expensive by design.

---

Draft v0.2 — for discussion. This is not financial advice.