The Weekly Review Kit

Four agents read every channel you run. A fifth tells you what to change on Monday.

On Saturday 24 October my AI operator, Harris, reviewed every channel I run in one sitting. Four agents worked in parallel: one on YouTube Studio retention, one on the short-form clips, one on the website and funnel, one on the cross-posts and the scripts behind them. Harris then combined what they found, fixed what was safe to fix, and handed me the rest as a short list.

The first finding was the uncomfortable one. My site's "109 visitors last week" was mostly bots. About six real people came.

This kit is that review, written so you can run it yourself with Claude Code. It is the exact set of briefs, rules and templates, with my private names taken out.

What's inside

- The four agent briefs, ready to paste.
- The combine-and-decide brief that turns four reports into one ranked list.
- The bot rules: how to tell a real visitor from a crawler, in plain words, plus the signals list.
- The thresholds we now judge every video against.
- The KPI note template, so every week lands in the same shape.
- What we found on week 1, with the real numbers.

Before you start

You need: Claude Code, and read access to the places your numbers live (YouTube Studio in your browser, your analytics tool, your posting tool, your site's code). Nothing in this kit writes to a live system.

The one rule that makes it work: every agent is read-only. It looks, counts and recommends. It never posts, deletes, deploys or emails. You decide what changes, after you've read the list.

How to run it

1. Save each brief in this kit as its own file in your project, for example `review/brief-youtube.md`.
2. In Claude Code, ask for four parallel agents, one per brief: "Run these four briefs as parallel subagents. Each writes its report to `review/<date>/<name>.md`. Read-only."
3. When all four are back, run the combine-and-decide brief.
4. Paste the result into your KPI note using the template. Same headings every week.

Fill in the blanks. Anything in `[square brackets]` is yours to replace: channel names, your site, your analytics project, your own devices.

Time: the agents take 20–40 minutes between them. Your part is reading one page.

Brief 1: YouTube Studio retention

```
ROLE: You are a read-only analyst reviewing my YouTube channel(s) for the week.

SCOPE: [channel names]. Windows: last 7 days and last 28 days.

WHERE THE DATA IS: YouTube Studio > Analytics > Advanced mode, in my browser
(signed in). The public Data API does not give retention; Studio does.

PULL, PER CHANNEL:
- Views, watch time, net subscribers, impressions and CTR (7d and 28d).
- Average view duration (AVD) and average percentage viewed (APV).
- Shorts: "Stayed to watch" % (the viewed-vs-swiped figure). This is the
  most important number in the report.
- Traffic sources with views and AVD for each (Shorts feed, search,
  channel pages, suggested, external).

PULL, PER VIDEO (every video, lifetime):
- Published date, length, title, AVD, APV, stayed %, views, impressions, CTR.
- For the top 2 and bottom 2 by stayed %: read the retention curve and write
  down the second where it drops hardest (e.g. "holds to 0:04, 33% by 0:07").

RULES:
- Write "-" where Studio shows no value. Never estimate.
- Flag any sample under ~50 views as too small to judge.
- Note anything odd about the traffic (e.g. most views from channel pages,
  which means visitors, not the feed).

OUTPUT: review/[date]/youtube-studio.md with: one totals table per channel,
one per-video table, the retention notes, and a cross-channel comparison
(views, stayed %, APV, typical length, main source). No recommendations
beyond a short "what stands out" list. Nothing is changed.
```

Brief 2: Short-form clip analysis

```
ROLE: You are a read-only analyst reviewing every short clip I published.

SCOPE: every Short/Reel/clip on [channels], all time, with views, likes,
publish time, length, title, and (if you have it) the source video and the
score my system gave it before posting.

WHERE THE DATA IS: [my posting system's database / export / spreadsheet].
Open any database read-only (for Postgres: default_transaction_read_only=on).

ANALYSE:
1. Per channel: count, total views, mean, MEDIAN, number at zero views.
   The median matters more than the mean: one viral clip hides a dead channel.
2. Per source / creator / topic: same columns.
3. Posting hour and weekday (say when a slot is too new to judge).
4. Length buckets and title patterns (question, number, name, first person).
5. Does my pre-post score predict views? Spearman correlation per channel.
6. Freshness: age of the source material at publish vs views.
7. Backlog: how much is waiting, how old, and how deep the queue is at the
   current posting rate.

RULES:
- Anything under ~72 hours old is immature: list it, don't judge it.
- Anything at n<5 is noise. Say so next to the number.
- Same creator doing well on one channel and badly on another means a channel
  problem, not a clip problem. Look for this.

OUTPUT: review/[date]/clips.md: a headline table, the analyses above, then
up to 9 proposed changes ranked by expected effect, each with the file or
setting it touches and its risk. Nothing is changed.
```

Brief 3: Website and funnel, with bot classification

```
ROLE: You are a read-only analyst finding out how many REAL people visited
my site this week and what they did.

SCOPE: [site]. This week vs last week. Sources: [PostHog / Plausible / GA]
events export, Cloudflare Web Analytics (cross-check only), my sign-up table,
my booking tool, my email automation logs.

STEP 1: Pull every event for both weeks. Group into sessions.

STEP 2: Classify each session as OWN, BOT or REAL using the bot rules in this
kit (next section). Apply the rules in order; the first match wins. Write the
classifier as a small script so next week runs the same way. Then read the
REAL sessions by hand: duration, scroll depth, pages, clicks.

STEP 3: The funnel, real people only:
  real visitors > reached half the page > saw the offer > used any
  interactive part > clicked the main call to action > signed up > booked.

STEP 4: By source: which link tags brought real people (not crawlers).

STEP 5: The automations: did each one run, any errors, how many people did
it touch? Check the logic, not just the run: does "booked" really mean a
booking of THIS offer, made AFTER the sign-up?

STEP 6: What's broken or unmeasured (pages that redirect before the analytics
loads, links without source tags, events that never fire).

RULES:
- Never quote a visitor number without saying how bots were removed.
- Cloudflare and most dashboards count crawlers as visitors. Say which
  numbers are raw and which are classified.
- Keys are read from files, never printed.

OUTPUT: review/[date]/site-funnel.md: headline (real people, sign-ups,
bookings), the classification counts, a table of every real session, the
funnel table, by-source table, automation table, gaps, then changes ranked by
expected effect on sign-ups and calls. Nothing is changed.
```

Brief 4: Cross-posts and script patterns

```
ROLE: You are a read-only analyst reviewing everything I cross-posted this
week and the scripts behind my own videos.

SCOPE: [platforms: TikTok, Facebook, Instagram, LinkedIn...] via
[my posting tool's API: GET requests only]. My own videos on [channel].

PART 1: CROSS-POSTS
- Followers per account, start and end of the week.
- Per platform: posts, views, reach, engagements, clicks, average views.
- Per post: route, time, views, share of views from the main feed
  (For You / Reels). Look for all-or-nothing patterns: posts that never
  entered the feed. Check cadence: 3rd and 4th posts in a day, posts within
  2 hours of another.
- Any comment-to-DM automation: triggered, sent, clicked.
- Reliability: every scheduled slot, did it post? If not, why?

PART 2: SCRIPT PATTERNS (my own videos)
- For every video: format, slot, length, views over the week, and the FIRST
  SPOKEN LINE, verbatim.
- Group winners and losers by opening. Look for: a specific strange event vs
  an abstract claim; a concrete number, name or time in line 2; openings that
  start mid-sentence ("but...", "so...").
- Check whether my hook rules are enforced in code or only in a prompt, and
  which content paths bypass them.

RULES: GET only. Nothing posted, scheduled, deleted, replied to.
Small numbers: tag conclusions [Certain] / [Likely] / [Guessing].

OUTPUT: review/[date]/social-content.md, ending with up to 10 workflow
changes ranked by expected effect, each with file, risk and evidence.
```

Brief 5: Combine and decide

```
ROLE: You are my operator. Four read-only reports are in review/[date]/.
Read all four in full, then give me one page.

1. HEADLINE: the 3-5 things that are true this week, each with its number.
   Lead with the one that changes a decision. If a number I've been quoting
   is wrong, say so first.
2. CONFLICTS: where two reports disagree, or one report's finding explains
   another's (e.g. a traffic source that's really crawlers).
3. RANKED CHANGES: merge every proposed change into one list, ranked by
   expected effect on [my goal: sign-ups / calls / sales]. For each: what,
   where (file / setting), why (the evidence), risk, and who does it.
   Mark each one:
     SAFE: reversible, no live audience effect, I can apply it now.
     NEEDS YES: touches a live system, money, posting or a person.
4. DON'T DO: changes the data does not support yet, and why.
5. NEXT WEEK: what to measure so next week's review can settle the open
   questions.

RULES:
- Recommendations only. Do not change any live system.
- Every claim carries its number and its source report.
- Small samples are named as small.
- Then append a dated block to my KPI note using the template, and add the
  same labels to my baseline file so next week can compare week over week.
```

The bot rules

Most small sites are mostly bots. The dashboards don't tell you, because a crawler that runs JavaScript looks like a visitor. These rules come from reading every session on my own site by hand. Apply them in this order; the first match wins.

Own visits (you, your team, your own tools):

- Visits carrying your own test tags (`?src=test` , `?src=verify` , anything you use for checks).
- Your own devices: your city, your screen sizes, your phone's exact resolution.
- Your own link checkers. If something you run opens your links after each upload, its visits come in tight bursts at the same minute. Tag those links (`&check=1`) and drop them.

Bots:

- Screens of exactly 2000×2000 with a Los Angeles timezone: Meta's link crawlers. They arrive within minutes of every Facebook or Instagram post.
- Screens of 800×600 on a non-touch device: scanners.
- Datacentre cities: Council Bluffs, Ashburn, Boardman, The Dalles, San Jose, Prineville, Forest City, Altoona, Fort Worth, Gallatin, Luleå, Clonee, Des Moines, Falkenstein, Manassas, and similar.
- A desktop reporting UTC or `Etc/Unknown` as its timezone.
- Linux desktops (not Android, not ChromeOS).
- iPhones in the US or Canada reporting a Los Angeles timezone, in pairs a minute apart, that never scroll.
- No operating system reported at all.
- `navigator.webdriver` true, or a user agent containing `HeadlessChrome` , `facebookexternalhit` , `meta-externalagent` or `Facebot` .

Real: everything left. Read each one by hand. At small volumes you will know most of them by the end.

The bot rules: the signals list

The fields to pull for every session, and what each one tells you.

SIGNAL	WHERE IT COMES FROM	WHAT IT TELLS YOU
Screen width × height	analytics event	2000×2000 and 800×600 are machines
Timezone	analytics event	UTC on a desktop is a server
City and country	IP lookup	datacentre towns are crawlers
Operating system and browser	user agent	Linux desktop, headless, named crawlers
Touch support	browser	separates real phones from scanners
Source tag (`?src=`)	the link	your own checks and campaigns
Referring domain	browser	a Facebook referrer is often Meta itself
Time since the post went out	your posting log	crawlers arrive within minutes
Duration and max scroll	page-leave event	0 s and 0% is rarely a person
Clicks and form events	analytics event	the only proof of intent

Two fixes worth making the same day:

1. Drop the obvious crawlers in the browser before your analytics script starts (webdriver, the named user agents, 2000×2000, 800×600 without touch, Linux on UTC). Be conservative: a crawler counted is better than a person lost.
2. Send your own page-leave event on `visibilitychange` and `pagehide`. iPhones rarely fire the unload event, so without it every phone visit looks like a 0-second bounce.

The thresholds

These are the lines we now judge every short video against. They came from one week of our own data, so treat them as starting points and replace them with your own after four weeks.

SIGNAL	LINE	WHAT IT MEANS
Stayed to watch (Shorts)	**under ~40%**	The feed stops pushing it. On our best channel the three least-viewed clips (415–490 views) sat at 39–42% stayed; most of the rest held 45–75% and reached 1,300–2,100.
Length	**31–36 seconds**	Our best channel's clips are all 31–36 s and hold ~70% average viewed (68.6% over 28 days, 72.5% over 7).
The opening	**a hook, then a concrete beat 2**	Open on a specific, strange event. Line 2 pays it off with a number, a name or a time. Abstract claims and mid-sentence openings lose people in the first 4–8 seconds.
Maturity	**72 hours**	Views arrive late and in bursts. Don't judge a video younger than three days.
Sample	**n < 5 is noise**	Below five posts in a group, write the number down and judge nothing.
Cadence on a new account	**2 a day, 4 h apart**	Our 3rd and 4th TikToks of a day got 3–4 views; posts well spaced reached 246–682.

The KPI note template

One note, one dated block per week, always the same headings. Append; never rewrite last week.

```
## [YYYY-MM-DD] weekly review

**Headline.** [The one finding that changes a decision, with its number.]

### Numbers (this week / last week / change)
| Metric | This week | Last week | Change |
|---|---|---|---|
| Real site visitors (bots removed) | | | |
| Sign-ups | | | |
| Calls booked | | | |
| Short-form views, all channels | | | |
| Best channel: stayed % | | | |
| Own channel: stayed % | | | |
| Cross-post views by platform | | | |
| Scheduled slots that shipped | [x of y] | | |

### What we learned
- [Finding] [Certain / Likely / Guessing], source: [report]

### What we changed (and when)
- [Change], [file or setting], [date], [who]

### Waiting on a yes
- [Change], [why it needs a decision]

### Next week, measure
- [The number that settles an open question]
```

Keep a small baseline file beside it (`kpi-baseline.json`), one entry per metric per week with the same label, so the next review can compute week-over-week without reading prose.

What we found on week 1

Our own review, 24 October 2026. Real numbers, small samples, names removed.

1. "109 visitors" was about 6 people. [Certain] Of the week's sessions, 67 were our own visits and link checks, 134 were bots, and 19 were left to read by hand: about 6 real outside people. Of 297 page loads in Cloudflare, 123 had a Facebook referrer and were Meta's crawlers; 53 were our own headless tests. Facebook brought 3 real people all week. YouTube links brought none: every YouTube-tagged visit was our own link checker.

2. One channel did 70% of the work. [Certain] Our clip system (it finds the best moments in long videos, cuts them into Shorts and posts them, with nobody touching it) published 85 Shorts across four channels. One channel earned 70% of the 23,094 views from 18% of the posts. Median views on that channel: 1,309. Median on the other three: 1 to 4. The same creators did well or badly depending on the channel, so the problem was channel health, not clip quality. It also got the fewest posts of the four.

3. Freshness beat the system's own score. [Likely] The pre-post score had no relationship with views (Spearman −0.01 across 85 clips). The age of the source material did: fresher clips got more views on every channel (source age vs views: −0.07 to −0.50; time spent waiting in the queue: down to −0.69).

4. Our own Shorts lose people at 4–8 seconds. [Certain] 21–22% stayed to watch, against 51–61% on the clip channels. On one Short the curve held to 4.5 s, then fell to a third by 7 s: the opening frame worked and the second beat didn't.

5. Specific stories beat abstract claims. [Likely, small n] "Day 1, I told him he was wrong" and a founder talking about where his own time goes earned 27 and 57 views. A series framed as a hypothetical ("Picture a builder up a ladder…") earned 0 and 1. Two clips that opened mid-sentence got 1 and 3.

6. Two silent failures. [Certain] A script relay was cutting prompts at 6,000 characters without saying so. And the email nurture counted the wrong kind of call as a booking, so our only outside sign-up stopped getting emails. Only 9 of 16 scheduled video slots shipped that week.

7. Both were fixed the same day. The bot filter, the booking check and the relay limit were all changed within hours. Everything else went on the "needs a yes" list.

The lesson: the dashboards were not lying, they were counting the wrong things. One review with the bots removed changed more decisions than a month of looking at the totals.

Make it weekly

Run it every Monday morning, before you plan the week. Same briefs, same template, same headings. By week four you have your own thresholds instead of ours.

If you'd rather have this running for you, with the vault, the console and the voice that run my own businesses, that's what the Harris founding install is: Thomas sets it up with you on one call. **useharris.com**