

The Ultimate Clipping Playbook

Everything I know about clipping streams. No fluff, no gatekeeping, no "buy my course". Picking a streamer, finding the moment, editing, posting, reading your numbers, and building a system that clips while you sleep. Written by Bruno, streamer and founder of LazyClipper.

I built LazyClipper, so it shows up in here. Every step works with any editor too. Short on time? The [10-minute guide](#) has the essentials.

0. Read this first
1. How clippers work
2. Picking your streamer
3. Rights and rules
4. Your setup
5. Finding moments
6. Anatomy of a good clip
7. Editing, step by step
8. Platform by platform
9. A posting system
10. Numbers and experiments
11. Working with streamers
12. Automating with an API and AI
13. 15 common mistakes
14. Checklists
15. Glossary

00 **Read this first**

Right now, while you're reading this, someone is live on Twitch doing the funniest thing they'll do all year. Their chat is losing it. And tomorrow nobody will remember it, because nobody clipped it.

That's the game. Streamers produce hours of content every single day and almost none of it ever reaches TikTok, Reels or Shorts. Every one of those moments is a video waiting for someone to cut it.

That someone can be you.

Clipping is one of the few things on the internet where you don't need a face, a following, a studio or money to start. You need a laptop, taste, and the discipline to post every day. The people winning at this didn't start with some secret. They started, and they didn't stop.

Most people will read this and do nothing. Some will post three clips and quit. The few who post every day for months are the ones streamers end up looking for. This playbook is for them.

The one rule: one clip posted today beats ten perfect clips "next week".

Results depend on your streamer, your taste and your consistency. Nobody can guarantee views or income, including me. What I can give you is the system.

01 **How clippers work**

A clipper turns long streams into short videos for TikTok, Instagram Reels and YouTube Shorts. Shorts are how most people discover a streamer today, and most streamers don't have time to make them. That gap is the whole job, and it's wide open.

Three setups

SETUP	HOW IT WORKS	GOOD FOR
Fixed deal	You clip for one or a few streamers. Paid per clip, per week or per month, or as a share.	Steady work, direct feedback
Clipping campaigns	A creator or brand posts a campaign with rules. Clippers post clips on their own accounts and get paid by views.	Starting without contacts
Own clip channel	A channel around one streamer or one niche, with permission. Grows an audience you own.	Long-term, needs patience

What you control: which moments you pick, how well you cut them, and whether you show up every day. Nail those three and you're already ahead of most clippers out there.

02 Picking your streamer

This is the most underrated decision in clipping. The streamer matters more than your editing. A great edit of a boring moment still flops, and a rough edit of a legendary moment can blow up.

What to look for

- **Streams often and on a schedule.** No streams, no clips.
- **Reacts big.** Loud laughs, fast comebacks, strong opinions, chaos with friends. Quiet grinders are hard to clip.
- **Works without context.** Can a stranger get the joke in three seconds?
- **Not already clipped to death.** Mid-sized streamers often have no clipper yet and are happy to find one.
- **A language you understand.** You can't pick moments you don't get.

Quick test before you commit

1. Watch two recent VODs at 1.5x speed.
2. Count moments you'd clip. Under three per stream? Keep looking.
3. Check their existing shorts. Which ones did well? That tells you what their audience wants.

03 Rights and rules

A clip is someone else's content. Get this right and everything else gets easier.

- **Ask first.** A short message is enough (template below). Or join a campaign where the streamer asks for clippers.
- **Credit every time.** Tag the streamer and name them in the caption.
- **Add value.** Platforms push down plain reuploads. Captions, a tight cut, a hook and a proper vertical layout count as real changes.
- **No added music you don't have rights to.** Popular songs get clips muted or blocked, especially on commercial accounts.

- **Music in the stream itself** can also get a clip blocked. Skip moments with loud copyrighted music in the background.
- **Play fair with platform rules.** No fake engagement, no bought followers, no spamming the same clip from many throwaway accounts. That gets accounts banned, and your streamer's name dragged with it.

Template: asking a streamer

Hey [name]! I clip streams for TikTok, Reels and Shorts.
I'd love to clip yours and post them on [account], credit and tag always included.
Here are 2 clips I made from your last stream: [links]
If you like them, I'll keep going. If not, no worries, I'll take them down.

Sending two finished clips works better than asking with nothing to show.

04 Your setup

- **An editor.** Any editor with vertical export and captions works. Or a tool that does the cutting for you.
- **Your phone.** Always watch the finished clip on a phone before posting. Captions that look fine on a monitor are often too small or hidden behind buttons.
- **One folder per streamer**, subfolders `to-edit`, `ready`, `posted`. Name files `date_streamer_moment.mp4`.
- **A simple log** (spreadsheet): date, clip, platform, hook, views after 48 hours. You'll need it in chapter 10.

05 Finding moments

Live or VOD

Watching live is fun but slow. Most clippers work from VODs and signals, and only watch the parts that matter.

Signals that point to a moment

- **Chat speed.** The best single signal. When chat suddenly moves fast, something happened. Spam of `KEKW`, `LMAO`, `OMEGALUL` or "clip it" is gold.
- **Viewer clips.** On Twitch, the clips page sorted by "top 24h" shows what viewers already clipped. Viewers are your free scouts.
- **Voice spikes.** Screams, laughing fits, sudden silence.
- **Markers.** Ask the streamer or mods to set a stream marker or type a chat command when something happens.
- **Donations, raids, subs.** Reactions to them are often good, the alerts themselves rarely.

The 10-minute VOD scan

1. Check the viewer clips first. Note the timestamps.
2. Open the VOD and jump to each timestamp, start 30 seconds before.
3. Skim the rest at 2x and stop wherever chat spikes.
4. Write down every candidate: timestamp plus one line ("falls off chair after jump scare").
5. Only then start cutting, best candidate first.

The three-second test: does the moment make sense to someone who has never seen this streamer, within three seconds? If not, it needs a setup line, or it's the wrong moment.

With LazyClipper: the VOD analysis reads the chat curve of a whole stream and suggests the moments. Streamers can also type `!clip` or `!mark` in their Twitch chat, and the clip gets cut after the stream.

06 Anatomy of a good clip

Once you see this, you can't unsee it. Almost every clip that works has the same shape: **hook**, **setup**, **payoff**, and it ends right after the payoff.

Four hook types that work for stream clips

HOOK

WHAT IT LOOKS LIKE

Reaction first	Open on the scream or the laugh, then rewind to how it happened.
----------------	--

Cold open	Start on the line that sets it up: "Chat, I bet I can eat this whole thing."
Question	Top caption asks what the viewer is about to see: "why is he eating out of a vase 🤪"
Stakes	Make clear what's at risk: "1 HP left, 400k on the line."

WEAK START

"Okay so, uh, chat, let me tell you what happened yesterday..."

STRONG START

Straight into the moment, top caption: "he did NOT expect that"

Length

- Most stream clips land between 15 and 45 seconds. A story with a real build-up can go to 60 or longer.
- Cut anything the payoff doesn't need. If you're unsure whether a part is needed, it isn't.

Ending

- End on the punchline or right after the reaction. No outro, no logo screen.
- If the last second flows back into the first, people rewatch without noticing.

07 Editing, step by step

1. Rough cut

- Set in and out points around the moment, then cut dead air: pauses, "uhm", reading chat out loud when it adds nothing.
- Jump cuts are fine. Nobody minds small jumps if the rhythm gets better.

2. Vertical layout

CONTENT	LAYOUT
Gaming with facecam	Facecam on top (about a third), gameplay below.
Just Chatting, IRL, podcast	Full frame on the face. Two or more people: split screen.
Gameplay is the joke	Gameplay big, small facecam in a corner.

Keep important stuff away from the edges. The apps put buttons on the right and captions plus the account name at the bottom. Roughly: keep faces and captions out of the bottom fifth and the right strip.

3. Captions

- Burned into the video, big, bold, two to four words at a time.
- Place them in the middle, between facecam and gameplay, or in the lower middle above the app's text.
- Highlight the one word that carries the joke.
- Proofread. Auto captions mishear names, slang and gamer words. A wrong word in the punchline kills it.

4. Zooms and effects

- A quick punch-in on the face at the reaction works. A zoom on every sentence gets tiring.
- Sound effects: one per moment at most, and quieter than the voice.

5. Sound

- Make the voice loud and even. Around -14 LUFS is a good target for all three apps.
- Turn game audio down under talking, up again for the big moment.

6. Final check on your phone

- Watch it once with sound and once muted. Muted, does it still make sense?
- Is anything hidden behind the app's buttons?

With LazyClipper: facecam detection, split screen for 2 to 4 people, twelve caption styles, sound effects and even voice volume are done automatically. You can still adjust the cut with a preview, fix the facecam, or describe changes by text or voice.

08 Platform by platform

Post every clip on all three. Same video, different text.

TIKTOK

INSTAGRAM REELS

YOUTUBE SHORTS

Text	Short hook in plain words	Short hook, one extra line is fine	Title in capitals, 1 to 2 emojis
Hashtags	Max 5	Max 5	Only #Shorts
Good tag set	streamer name, #twitch or #kick , #stream , one of #fyp / #viral , one topic tag		
Cover	Pick a frame with a clear face	Cover shows in your grid, keep it clean	Auto, the first frame matters

WEAK TEXT

Stream highlights part 12 #gaming #funny
#twitch #streamer #lol #fun #clips #best ...

STRONG TEXT

why is he eating out of a vase 🤔
#streamername #twitch #stream #fyp #food

Describe the moment, not the stream. Thirty hashtags don't help, they just look like spam.

09 A posting system

- **Daily beats perfect.** One to three clips a day per account, around the same times.
- **Batch the work.** One session to find moments, one to cut, one to write texts and schedule. Switching between tasks costs more time than the tasks.
- **Schedule ahead.** Keep two or three days of clips ready, so a bad day doesn't break the streak.
- **Test your times** for two weeks, then keep the ones that work for your audience.
- **Don't delete flops.** Some clips pick up after days.

With LazyClipper: connect TikTok, YouTube and Instagram once, then post or schedule a clip to all three with one click. The posting calendar shows the whole week.

10 Numbers and experiments

NUMBER

WHAT IT TELLS YOU

IF IT'S LOW

Average watch time / completion	Does the clip hold people?	Start is too slow or the clip too long
Swipe-away rate (where shown)	Does the first second work?	Stronger hook, start later
Rewatches	Was the moment good?	Pick better moments, loop the ending
Shares and saves	Would people send it to a friend?	More relatable or more extreme moments
Follows per clip	Do people want more of this streamer?	Stay consistent with one streamer or niche

How to experiment

1. Change one thing at a time: hook type, length, caption style or posting time.
2. Compare after 48 hours, not after 2.
3. Write down what you changed and what happened (your log from chapter 4).
4. After two weeks, keep what worked and test the next thing.

11 Working with streamers

- **Send a weekly update.** Three lines: clips posted, best clip with link, what you'll try next week. Streamers keep clippers who communicate.
- **Ask what they don't want clipped.** Private stuff, sponsor segments, moments they regret. Respect it without discussion.
- **Ask them to help you.** Markers, a chat command, a pinned message telling viewers to clip. It makes your job faster and their clips better.
- **Put everything important in writing,** especially money: what you deliver, how often, how you get paid.

Template: weekly update

Week of [date]: 9 clips posted on TikTok, Reels and Shorts.
Best one: [link], held people best this week.
Next week I'm testing shorter clips (under 25s).
Anything you don't want clipped? Just tell me.

12 Automating with an API and AI

This is where it gets fun. Once you clip for more than one streamer, the repetitive parts eat your day: checking for new clips, downloading, cutting, exporting. That part can run on its own. You stop being the person who edits and become the person who runs the machine. Your job: picking and quality control.

The flow

1. **Find new clips.** Twitch has an official API that lists a channel's clips with view counts. A script checks it every hour for each streamer you work with.
2. **Filter.** Only clips above a view threshold, or clips a mod or the streamer created.
3. **Send them to an editing API.** For example the LazyClipper API: one request per clip link, it returns a finished vertical short with captions.
4. **Collect the results** in your `ready` folder, one per streamer.
5. **You review** and schedule. Never auto-post without looking. One bad clip can cost you a streamer.

Let an AI coding assistant write it

You don't need to be a developer. Tools like Claude can write this script for you if you describe it clearly. A prompt that works:

Write a small Node.js script that:

1. Reads a list of Twitch channel names from `channels.json`.
 2. Every hour, fetches each channel's clips from the last 24 hours using the official Twitch Helix API (client id and secret from environment variables, never hardcoded).
 3. Skips clips it already processed (keep their ids in `done.json`) and clips with fewer than 50 views.
 4. For each new clip, sends the clip URL to the LazyClipper API (POST `https://lazyclipper.com/api/v1/clips`, Bearer key from the environment variable `LAZYCLIPPER_KEY`, JSON body `{"url": ..., "targetDuration": "30"}`, header `Idempotency-Key` = the clip id).
 5. Polls GET `/api/v1/clips/{id}` at most every 5 seconds until the status is "done" or "failed", then downloads `/api/v1/clips/{id}/file` into `ready/<channel>/<clip id>.mp4`.
 6. Respects HTTP 429 by waiting for the `Retry-After` header.
- Explain how to run it step by step, for someone who has never used Node.

Rules for automation

- **Keys stay secret.** API keys go in environment variables, never into code you share, screenshots or chats.
- **Permission still applies.** Automate only for streamers who said yes.
- **Quality over quantity.** Ten hand-picked clips beat a hundred automatic ones. Platforms punish spammy accounts.
- **Watch your costs.** Every processed clip costs something, whatever tool you use. Set a view threshold.

With LazyClipper: the API is included in the Studio plan and works with token packages.
Documentation at lazyclipper.com/api.html.

13 15 common mistakes

1. Starting with the warm-up instead of the moment.
2. Clips that need you to know the streamer.
3. No captions, or captions with wrong words.
4. Captions or faces hidden behind the app's buttons.
5. Too long. The payoff comes, but nobody's left.
6. An outro or logo screen at the end.

7. Quiet voice, loud game.
8. Added popular music that gets the clip muted.
9. Thirty hashtags.
0. Titles that describe the stream, not the moment.
1. Posting in bursts, then nothing for a week.
2. Judging a clip after two hours.
3. Changing five things at once, so you learn nothing.
4. Clipping without permission, or without credit.
5. Automating before you know what a good clip looks like.

14 Checklists

Every clip

- ☐ I'm allowed to post it, the streamer is tagged and named.
- ☐ Passes the three-second test.
- ☐ Starts on the moment or the setup line, hook caption on top.
- ☐ Hook, setup, payoff. Ends right after the payoff.
- ☐ Layout fits the content, faces big, nothing behind the buttons.
- ☐ Captions burned in, proofread, keyword highlighted.
- ☐ Voice loud and even, no added copyrighted music.
- ☐ Watched on my phone, with and without sound.
- ☐ Text describes the moment, max 5 hashtags, `#Shorts` on YouTube.

Every week

- ☐ Posted every day on all three apps.
- ☐ Log updated with 48-hour numbers.
- ☐ One experiment finished, next one picked.
- ☐ Weekly update sent to each streamer.
- ☐ Two to three days of clips scheduled ahead.

15 Glossary

VOD	The recording of a past stream.
Hook	The first second or two that decides whether people keep watching.
Payoff	The moment the clip is about: the punchline, the fail, the reaction.
Completion rate	How many viewers watch to the end.
Marker	A timestamp set during the stream, so the moment is easy to find later.
Campaign	An offer from a creator or brand that pays clippers by views, with its own rules.
LUFS	A loudness unit. -14 LUFS is a common target for social video.
API	A way for your own scripts to use a service without clicking through its website.

Now go clip something

Seriously. Close this, open a VOD, find one moment, post it today. Then do it again tomorrow.

And if you want the lazy way: LazyClipper does chapters 5 to 9 for you: paste a Twitch, Kick or YouTube clip link, get a ready-to-post short with facecam layout, captions and sound, then post it everywhere. Streamers get a chat bot for `!clip` and `!mark`. Right now it's in a closed beta with selected streamers.

Want in? Send me **BETA** in a DM on Instagram or write to contact@lazyclipper.com.