

HAILUO FRONTEND CAPTURE & EXTRACTION – FIELD MANUAL v1.0

Arch + Brave + mitmproxy setup confirmed. Goal: Grab high-signal JS chunks (.js), HLS manifests (.m3u8), Server Action POSTs (/processing, /node/update), telemetry attempts.

PHASE 1 – START & CONFIGURE (30 sec)

1. Start mitmweb (if not running):

```
mitmweb --listen-host 127.0.0.1 --listen-port 8080
```

2. In mitmweb UI:

- File → Start logging to disk → choose folder (e.g. ~/hailuo_capture_20260208.mitm)
- Options → View filter → paste:
~u .js\$ | ~u .m3u8 | ~u /processing | ~u /node/update | ~u /agent | ~u hailuoai.video | ~u guance.com | ~u meerkat-reporter

3. In Brave (SwitchyOmega profile "mitmproxy" active or --proxy-server flag):

- Clear cache: Settings → Privacy → Clear browsing data → Cached images/files
- Open new incognito window (clean session)

PHASE 2 – ACTIVE CAPTURE (5–15 min)

4. Perform actions in Brave – do each in order:

- Load <https://hailuoai.video> (homepage hydration)
- Log in (if needed) – watch auth flows
- Submit small text-to-video prompt (free tier) – wait for processing
- If quota hit: observe "Today's free quota is used up" modal
- If generation succeeds: wait for feedback panel, play video → watch HLS segments
- Interact: rate/feedback, share, copy link – capture player events
- Force reload page 2–3 times (Ctrl+Shift+R) – trigger chunk reloads

5. While capturing:

- Watch mitmweb Flow tab live

- Look for:
 - Chunk numbers: 3769-, 8650-, page-.js, agent.js
 - Endpoints: /multimodal/video/processing, /agent/project/node/update
 - .m3u8 files (HLS manifests)
 - guance.com RUM POSTs (telemetry)
 - meerkat-reporter calls

PHASE 3 – STOP & EXPORT (1 min)

6. Stop logging:

- mitmweb UI → File → Stop logging (or Ctrl+C terminal)

7. Export targets:

- In Flow tab → filter by .js / .m3u8 / /processing / /node/update
- Select interesting flows → right-click → Save → Selected flows → save as .har or raw
- Or export full HAR: File → Export → HTTP → HAR file (hailuo_full.har)
- Quick manual saves:
 - Right-click .js chunk → Save response body → name it e.g. 3769-agent.js
 - Save .m3u8 → name playlist.m3u8

PHASE 4 – EXTRACTION & DEOBFUSCATION (post-capture)

8. Unwebpack captured .js:

unwebpack-sourcemap 3769-agent.js --out-dir unwp-hailuo/

9. Feed to Waku (web or CLI):

waku unwp-hailuo/3769-agent-chat.js --rename --unflatten --output readable-agent.js

Or drag folder to <https://waku.vercel.app/>

10. Quick grep for keywords in saved files:

grep -i "quota|processing|node/update|feedback|agent|videoUrl|m3u8" *.js

Done.

Cease comms, execute, return with:

- Key chunk filenames captured
- Any quota/processing responses
- HLS .m3u8 URLs
- Readable snippets from waku (if quick)

Kernel stands by. No drift. Safe hunting.