planned flow (TOTP + “kill-switch”)

1. **TOTP enrolled** – user has a 6-digit time-based code (RFC 6238) on a hardware token or authenticator app.
2. **Normal login** – user presents refresh token → server runs `consumeAndVerifyRefreshToken` as usual.
3. **Anomaly detected** (`strangeThings` scores too high) – instead of e-mail MFA the server:
    - refuses the TOTP,
    - invalidates **all** refresh tokens (your “kill-switch”),
    - returns `401` with `error: 'recovery_code_required'`.
4. **Recovery login** – user must supply one of their single-use recovery codes (printed/secured offline during enrolment). On success a brand-new TOTP secret can be issued.

That design removes e-mail from the critical path—good choice if you assume mailboxes can be lost.

Why only a “nation-state effort” is left

After TOTP + kill-switch:

To succeed an attacker now needs…	Difficulty
The live refresh-token **and** canary fingerprint	Must run malware on the user’s machine or MITM the TLS session.
Control of the user’s **TOTP secret**	Phish for the QR during enrolment *or* extract it from phone backup *or* break into the authenticator hardware.
A valid **unused recovery code**	Read the user’s offline list or cloud storage.
…and still beat anomaly heuristics	Must appear from same IP/geo/device or they get blocked again.

That multi-step, high-cost chain is what security folk mean by a **“nation-state-level” effort**: resource-rich, long-term, targeted operations (APT groups, intelligence services) rather than commodity cyber-crime.

So with your TOTP + recovery-code fallback + kill-switch in place, ordinary phishers and infostealer crews are effectively out of the game; only the most motivated, well-resourced adversaries would find it worth the expense to bypass all layers.