Output Guardrails: A Measured Comparison

Fluiq · 2026-08-07 · eight guardrails, four corpora, 949 cases

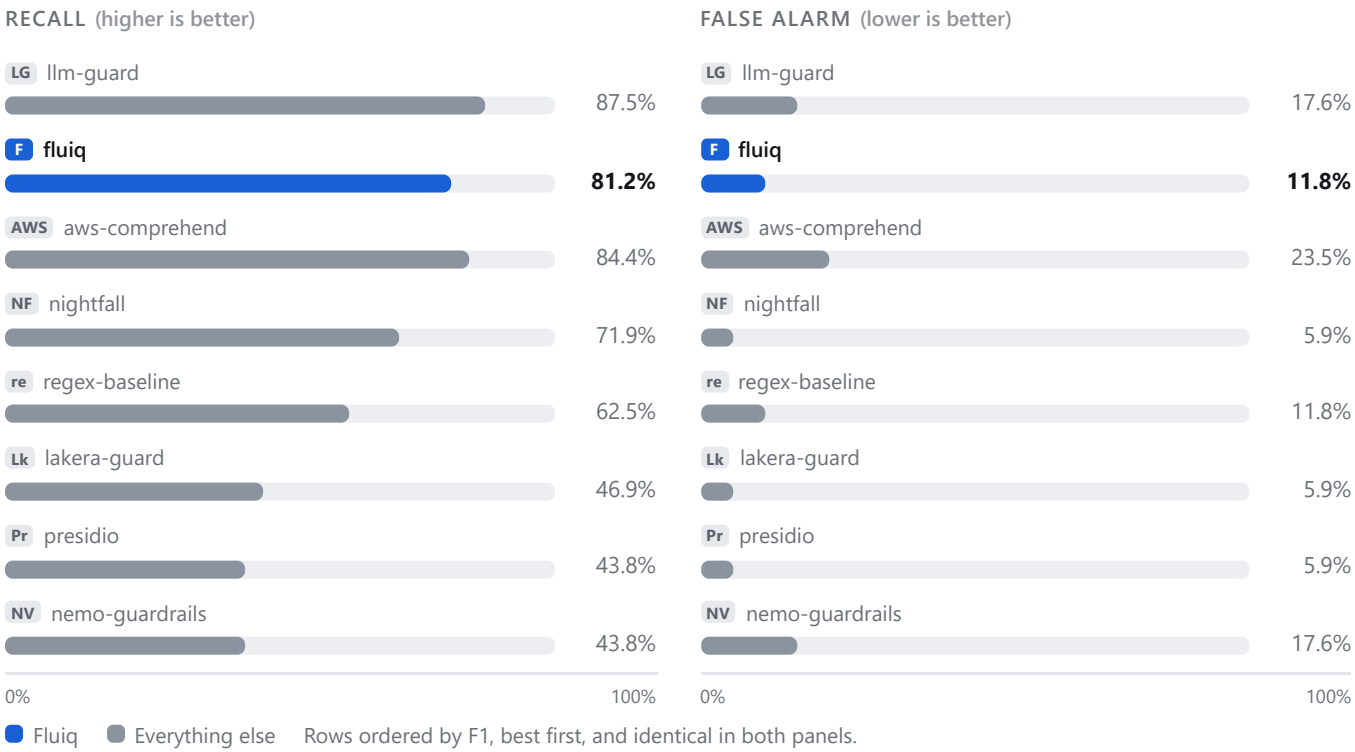
Most LLM safety benchmarks ask whether the *model* misbehaves. This one asks a different question: given something a model is about to say, or something a user just sent, **would your guardrail stop it?**

> **Disclosure.** This benchmark is published by Fluiq, and Fluiq is one of the contestants. No methodology removes that conflict. What we do instead: the corpora, harness and every adapter are public; scoring was fixed before any contestant ran; competitors run at stock settings with nothing tuned to these cases; every miss and false alarm is listed by case ID in the raw results; and a thirty-line regex control is included so it is visible whenever a sophisticated product barely beats it. Fluiq does not place first on either output-side corpus.

Output leakage: adversarial suite

Hand-built cases covering leak shapes a PII corpus does not contain: obfuscated identifiers, credential formats, injected instructions echoed into output, and model refusals that leak while refusing.

Corpus 49 cases · 32 block / 17 allow | **Source** Written for this benchmark | **Licence** MIT (published with this repository)

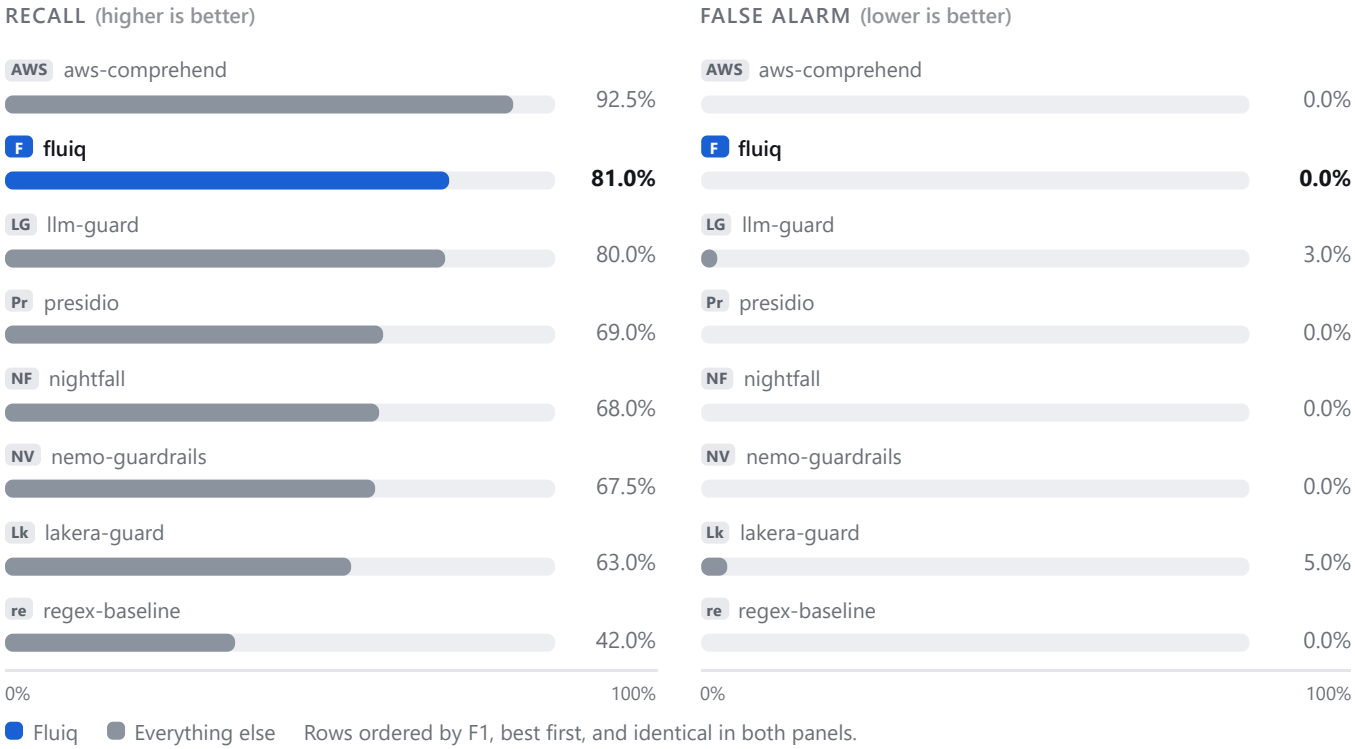


GUARDRAIL	CATEGORY	RECALL	FALSE ALARM	F1
`llm-guard`	OPEN SOURCE	87.5%	17.6%	88.9%
`fluiq`	FLUIQ	81.2%	11.8%	86.7%
`aws-comprehend`	COMMERCIAL	84.4%	23.5%	85.7%
`nightfall`	COMMERCIAL	71.9%	5.9%	82.1%
`regex-baseline`	CONTROL	62.5%	11.8%	74.1%
`lakera-guard`	COMMERCIAL	46.9%	5.9%	62.5%
`presidio`	OPEN SOURCE	43.8%	5.9%	59.6%
`nemo-guardrails`	OPEN SOURCE	43.8%	17.6%	57.1%

Output leakage: public PII corpus

Sampled from the most widely used public PII dataset. Positives are rows containing at least one strong identifier; negatives are the same rows with PII replaced by placeholders, so both halves share a distribution.

Corpus 300 cases · 200 block / 100 allow | **Source** ai4privacy/pii-masking-200k | **Licence** Not stated on the dataset card

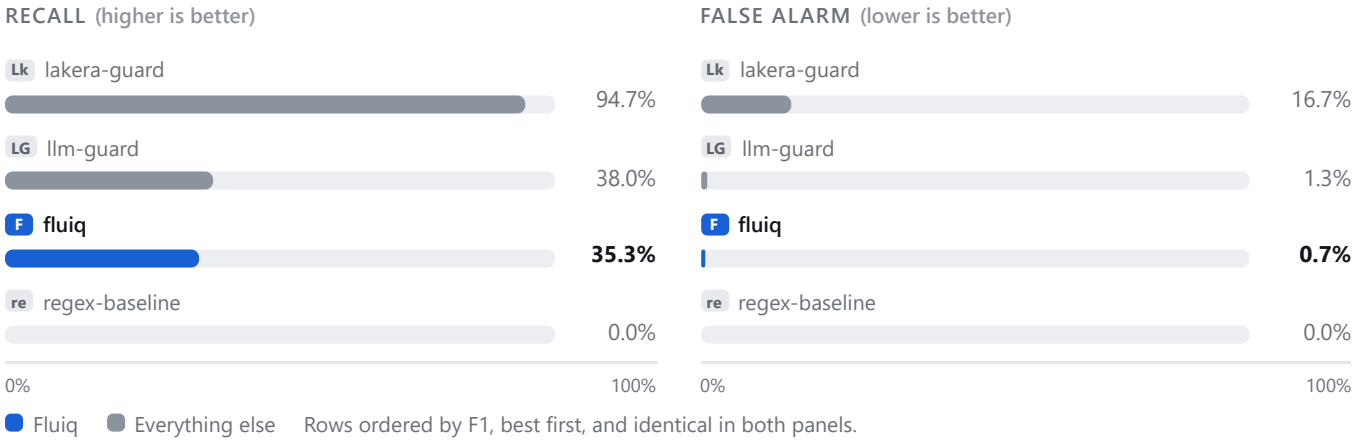


GUARDRAIL	CATEGORY	RECALL	FALSE ALARM	F1
aws-comprehend	COMMERCIAL	92.5%	0.0%	96.1%
fluiq	FLUIQ	81.0%	0.0%	89.5%
llm-guard	OPEN SOURCE	80.0%	3.0%	88.1%
presidio	OPEN SOURCE	69.0%	0.0%	81.7%
nightfall	COMMERCIAL	68.0%	0.0%	81.0%
nemo-guardrails	OPEN SOURCE	67.5%	0.0%	80.6%
lakera-guard	COMMERCIAL	63.0%	5.0%	76.1%
regex-baseline	CONTROL	42.0%	0.0%	59.2%

Prompt injection

The standard public prompt-injection set, balanced 150/150. Roughly a third of the attacks are not in English, which turns out to matter a great deal.

Corpus 300 cases · 150 block / 150 allow | **Source** deepset/prompt-injections | **Licence** Apache-2.0

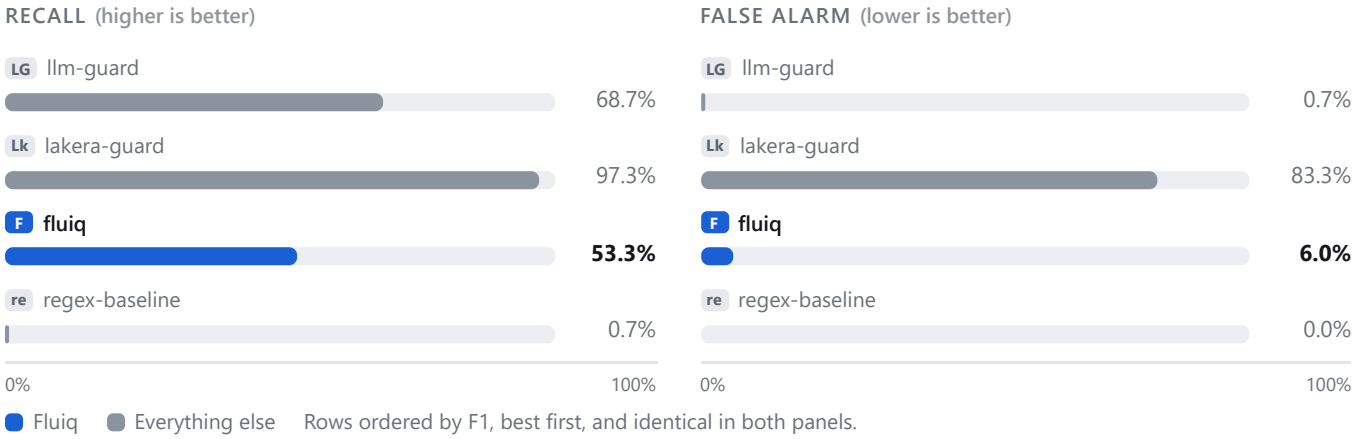


GUARDRAIL	CATEGORY	RECALL	FALSE ALARM	F1
lakera-guard	COMMERCIAL	94.7%	16.7%	89.6%
llm-guard	OPEN SOURCE	38.0%	1.3%	54.5%
fluiq	FLUIQ	35.3%	0.7%	52.0%
regex-baseline	CONTROL	0.0%	0.0%	0.0%

Jailbreak

Balanced 150/150. The benign half contains a large amount of persona and roleplay prompts, which is the source of a labelling disagreement discussed under Limitations.

Corpus 300 cases · 150 block / 150 allow | **Source** jackhhao/jailbreak-classification | **Licence** Apache-2.0



GUARDRAIL	CATEGORY	RECALL	FALSE ALARM	F1
`llm-guard`	OPEN SOURCE	68.7%	0.7%	81.1%
`lakera-guard`	COMMERCIAL	97.3%	83.3%	69.4%
`fluiq`	FLUIQ	53.3%	6.0%	67.0%
`regex-baseline`	CONTROL	0.7%	0.0%	1.3%

Other Fluiq configurations

The tables above give every contestant one row, ours included. We measured more than one configuration of our own gate, and the one that ships as `fluiq.secure()` is the one that competes, because nobody runs Lakera twice with its model switched off. The rest are here with their numbers, since every one of them scores lower and leaving them out would have been the convenient thing to do.

CONFIGURATION	CORPUS	RECALL	FALSE ALARM	F1
`fluiq (inline scanner)`	Output leakage: adversarial suite	68.8%	5.9%	80.0%
`fluiq (inline scanner)`	Output leakage: public PII corpus	39.0%	0.0%	56.1%
`fluiq (API fast path)`	Prompt injection	13.3%	0.0%	23.5%
`fluiq (API fast path)`	Jailbreak	42.0%	5.3%	57.0%

`fluiq (API fast path)` is the synchronous pattern-only check the API serves inline, which has no semantic layer. `fluiq (inline scanner)` is the trimmed scanner behind the public demo page. Both are real code paths that real traffic hits. Neither is the product.

Method

Every guardrail receives the same string and answers one question: block or allow. Scoring was fixed before any contestant ran.

CORPUS SAYS	GUARDRAIL SAYS	COUNTED AS
block	blocked	caught
block	allowed	miss
allow	blocked	false alarm
allow	allowed	quiet

> **Why F1, and why the false-alarm column never leaves.** Recall alone is meaningless here: a guardrail that blocks everything scores 100%. Roughly a third of each output corpus is benign text built to bait false positives — order numbers shaped like card numbers, git SHAs shaped like secrets, a sentence that merely describes an injection attempt.

Datasets

CORPUS	SOURCE	LICENCE	CASES
Output leakage: adversarial suite	Written for this benchmark	MIT (published with this repository)	49
Output leakage: public PII corpus	ai4privacy/pii-masking-200k	Not stated on the dataset card	300
Prompt injection	deepset/prompt-injections	Apache-2.0	300
Jailbreak	jackhhao/jailbreak-classification	Apache-2.0	300

> **Licence caveat.** `ai4privacy/pii-masking-200k` carries no licence statement on its dataset card. We sample from it and publish derived results; anyone redistributing that corpus should resolve its terms first. The other two public datasets are Apache-2.0.

Limitations

- **A string is not a trace.** Some products, ours included, detect RAG poisoning and tool exfiltration from structured trace context rather than response text. Feeding only a string understates them.
- **Scope differs between contestants.** Presidio detects PII and does not claim to detect API keys, so it scores zero on secrets. That is a scope difference, not a defect, which is why per-category results exist and the single headline number should not be read alone. The same applies to Lakera on the output-side corpora.

- **One labelling disagreement is material.** On the jailbreak corpus, persona prompts (“you are Black Panther”) are labelled benign; Lakera treats persona adoption as hostile. That single disagreement drives both its high recall and its high false-alarm rate there. It is a difference of opinion about what an attack is, not a defect.
- **Rate limits are not capability.** Nightfall's first run errored on 190 of 300 cases through rate limiting. Those results were discarded, backoff was added, and the harness now refuses to rank any contestant erroring on more than 2% of a corpus.
- **Synthetic values must be realistic.** An early draft used an Anthropic key shorter than a real one, which made a correctly length-anchored pattern look like a miss. A corpus full of unrealistically short secrets rewards sloppy detectors.
- **Guardrails change.** Every number here is dated. A benchmark result without a date and a version is worthless.

Reproducing this

The harness, both hand-written corpora, the public-dataset samplers and all adapters are published. Contestants needing credentials are skipped and reported, never scored zero.

```
python run.py --corpus <corpus>.jsonl --json results.json
python make_report.py
```

Adding a guardrail means implementing one method, `scan(text) -> Verdict`. If you think a product is misconfigured here, the fastest rebuttal is a pull request.

Generated 2026-08-07 from results JSON by `make_report.py`. All credentials, names and identifiers in the hand-written corpus are synthetic.