Аномалии и Prompt Guard

Аномалии (`/anomalies`)

Доступ: **Security Analyst+**.

Таблица: Timestamp, Agent, Risk, Source (`auto` / `manual`), Status, Details. Клик по строке раскрывает подробности.

Разбор аномалии

1. Отсортируйте по risk (сначала высокие).
2. Откройте детали: что отклонилось от базового поведения.
3. Сверьте с **Audit** того же агента и периода.
4. Вынесите вердикт:
   - **True Positive (TP)** — реальное отклонение / угроза → подтвердите в UI, ужесточите политику, заблокируйте действие через HITL/Admin;
   - **False Positive (FP)** — ложное срабатывание → отметьте FP (feedback), чтобы модель дообучалась.

Feedback через API:

```
curl -X POST http://localhost:8001/api/v1/anomalies/detector/feedback \
  -H "Authorization: Bearer $JWT" \
  -H "Content-Type: application/json" \
  -d '{"anomaly_id": "...", "label": 0}'
```

`label: 0` — FP, `1` — TP. После накопления событий возможен auto-retrain.

Detector Status

На странице — версия модели, backend (`sklearn` / `onnx` / …), samples, contamination, last trained.
Обучение / rollback версий — **Admin+** (`POST /detector/train`, rollback).

Если агент «сирота» (нет в реестре) — смотрите **Discovery**.

Prompt Guard ML (`/guard-ml`)

При `ENABLE_GUARD=true` перед вызовами LLM проверяются injection / jailbreak / утечки.

На **Dashboard** видны:

- Prompts Blocked
- Guard Latency
- Top attack patterns

Оператор:

1. Следит за ростом блокировок и паттернами атак.
2. При всплеске — Audit + Anomalies по затронутым агентам.
3. Сообщает разработчикам, если легитимные промпты массово блокируются (нужна настройка Guard).

Подробнее (EN): Anomalies, Prompt Guard.