Реагирование на инциденты

Краткие playbook’и для типовых ситуаций. Цель — стабильный порядок действий без импровизации.

A. Нарушение политики (`policy.violated`)

1. Зафиксируйте toast / запись в **Audit** (agent, action, resource, risk).
2. Откройте **Policies** — какая политика и severity.
3. При `required_approval` — обработайте **HITL**.
4. Если в ответе check есть anomaly — откройте **Anomalies**.
5. Решение:
   - исправить контекст агента (consent, lawful basis и т.д.);
   - ужесточить/ослабить политику (**Admin**);
   - временно остановить агента / workflow.
6. Сохраните вывод для тикета (экспорт Audit при необходимости).

B. Очередь HITL / критичное согласование

1. **HITL** → сортировка pending по критичности ресурса.
2. Проверьте action, records_affected, причину запроса.
3. **Approve** или **Reject** + reason.
4. Убедитесь, что workflow вышел из `awaiting_approval`.
5. При перегрузе очереди — эскалация Admin / второй дежурный.

C. Аномалия / подозрение на атаку

1. **Anomalies** → высокий risk → детали.
2. Корреляция в **Audit** по `agent_id` и времени.
3. TP → политика / блок / HITL на действие; FP → feedback `label: 0`.
4. Orphan-агент → **Discovery** Confirm.
5. При деградации детектора — Admin (retrain / rollback), rule fallback.

D. Превышение бюджета

1. **Cost**: баннер EXCEEDED / toast `cost.budget.alert`.
2. Найдите агента с аномальным spend.
3. **Cancel** активные workflow при необходимости.
4. Поднимите `limit_usd` (осознанно) или остановите источник нагрузки.
5. Проверьте, нет ли цикла вызовов / инцидента безопасности.

E. Платформа Degraded

1. Индикатор статуса на Dashboard.
2. Сообщите админу инфраструктуры; проверьте health:

```
curl https://<host>/api/v1/health
curl https://<host>/api/v1/health/ready
```

1. OPA down → workflow fail-closed; не считайте «тишину» успехом.
2. Нет API key → Settings / повторный логин.

F. Shadow AI

1. **Discovery** → evidence кандидата.
2. **Confirm** (зарегистрировать) или **Dismiss**.
3. После Confirm — политики, бюджет, владелец сервиса.