ОТ МЕШКА FALSE POSITIVE ДО НОРМАЛЬНОГО ВЕРДИКТА

паяем ZeroFalse-inspired LLM-триаж на локальных моделях

ёPRSTCON · 26.05.2026

ОБ АВТОРЕ

зачем меня слушать следующие 15 минут

КТО

Юра Туманов
ака Эльария / Psycho Drake
«Ростелеком» · AppSec

ЭКСПЕРТИЗА

Кибербез с 2001 года
Кандидат технических наук
Кибербез + нейронки · пою

ПОЧЕМУ ЭТА ТЕМА

Покупалась RTX 4080,
чтобы играть в киберпанк.

Служит — чтобы создавать его
в реальной жизни.

КОНТАКТЫ

@psychodrake
github.com/Eljees

В ЧЁМ БОЛЬ

почему ручной триаж SAST-срабатываний — это никогда не закрыть

SAST

Static Application Security Testing · автоматический проверяющий, читает код и кричит «тут проблема»

~1 000 000

SAST-срабатываний
в нашем стеке в неделю

< 10

человек в команде
безопасной разработки

95–98 %

из срабатываний —
false positive (ложная тревога)

никого из экспертов мы не выкидываем — просто хотим, чтобы у нас меньше болела жопа

ЧТО ЗАБЕРЁШЬ ЧЕРЕЗ 15 МИНУТ

четыре принципа, работающие на любом стеке

01 **Дроби задачу**

один большой промпт хуже трёх маленьких с контрактом между ними

02 **JSON-схема на замок**

фиксируй формат ответа, инвалид-JSON выбрасывай — не пытайся спасти

03 **Эвристика до LLM**

регулярки и анти-паттерны бесплатно ловят треть очевидного

04 **Маленький независимый эталон**

245 размеченных вручную лучше 10 000 «человек кивнул на то, что предложила модель»

ЧТО В РУКАХ · КАК УСТРОЕНО

один человек · одна 4080 · полгода

СТОЛ

- NVIDIA RTX 4080 · 16 GB VRAM
- vLLM · движок инференса
- Qwen2.5-Coder-14B-AWQ · квантизация
- ~3–6 сек на одно решение
- Облачные LLM нельзя — corp периметр

ПОЧЕМУ ИМЕННО ТАК

- Локально → не утекают корп. данные
- 14B · меньше — глупит, больше — не лезет
- AWQ-квант → 16 GB хватает с запасом
- vLLM → батчинг и быстрые prefill
- Перебрал десяток моделей — Qwen победил

ПАЙПЛАЙН — линейный, без магии:

FETCH
взять issue
из агрегатора

HEURISTIC
быстрый чек
регулярками

LLM CALL
модель даёт
вердикт + JSON

APPLY
записать решение
в агрегатор + лог

ПРОМПТЫ v0 → v1

декабрь—январь · от «одной строки» к «каталогу regex»

КАК ЭТО РАБОТАЕТ ВООБЩЕ

Модель — это быстрый, но мало знающий джун. Промпт — инструкция под задачу.
Дальше пять слайдов: как я учил джуна не угадывать, а думать.

v0 · 27.12 · одна строка

«Хардкод-пароль (CWE-798).
Часто срабатывает на UUID-ах.
UUID не считать секретом.»

ЧТО ЛОМАЛОСЬ

Модель ставила «уязвимость» на любую длинную строку — просто угадывание.

v1 · 04.02 · каталог префиксов

ghp_ (GitHub) · AKIA (AWS)
AIza (Google) · eyJ (JWT)
→ Confirmed conf 0.8–0.95

ЧТО ЛОМАЛОСЬ

Тестовые JWT с пометкой staging уходили в «уязвимость». Регулярок не хватило.

ПРОМПТЫ v2 — СТРУКТУРА

10 февраля · переломный момент: от «правил» к «процессу принятия решения»

СЕМЬ БЛОКОВ ПРОМПТА

- **Определение** — что считать секретом
- **Признаки «да»** — когда «это уязвимость»
- **Признаки «нет»** — когда «ложная тревога»
- **Рубрика решения** — ровно 3 ветки
- **Доказательства** — что обязано лежать в ответе
- **Non-prod правило** — test/dev → ослабляем
- **Формат ответа** — строгий JSON, никакого свободного текста

ИДЕЯ РУБРИКИ

Confirmed только если ВСЕ три:
- значение секрета на месте
- видно что оно настоящее
- видно контекст использования

False Positive если placeholder, пример, доки, тест, public-only.

Unknown — если контекста нет.

ИТОГ: модель стала предсказуема. Стало возможно мерить.

v3 + ЕЖЕНЕДЕЛЬНАЯ КАЛИБРОВКА

март—май · самый плохой класс ошибок и как от него ушёл

v3 · 19.03 · flow-to-prod · самый плохой класс ошибок

ПРОБЛЕМА:
реальные пароли в .gitlab-ci.yml
уходили в False Positive — потому что
путь содержит «CI» → попадал под
правило «test/dev, не страшно».

«Пароль в проде, а модель говорит:
расслабьтесь.»

ДОБАВИЛ В ПРОМПТ:

If NON_PROD but file appears in:
shared code, planned merge,
CI/CD, deployment, IaC
→ do NOT soften the verdict

ПАРАЛЛЕЛЬНО · ЕЖЕНЕДЕЛЬНАЯ КАЛИБРОВКА

severity-downgrade	unknown-reduction	weekly addendum	auto-skip по timeout

каждое изменение — сверка по эталонному разметчику, без числа никаких правок

ПРАВИЛО D-23 v2 · 23 мая

одно правило, три CWE: XSS 0.87→1.0 · SQL 0.07→1.0 · XXE 0.88→0.94

ДО · bak_20260523

Раньше требовали три якоря:

- источник данных
  (откуда пришли от юзера)
- опасное место использования
  (innerHTML, eval и т.п.)
- отсутствие очистки

ПРОБЛЕМА:
SAST даёт только одну строку
с опасным местом, источник
не виден → модель пасует

ПОСЛЕ · D-23 v2

Каждый фрагмент кода смотрим
НЕЗАВИСИМО. Нашёл опасное
место с переменной аргументом —
этого ДОСТАТОЧНО.

- eval(X) где X — переменная
- innerHTML = X
- pickle.loads(X) и т.п.

РЕЗУЛЬТАТ на N=85:
XSS 0.87 → 1.00
SQL 0.07 → 1.00
XXE 0.88 → 0.94

ГЛАВНЫЙ КЕЙС · CWE-328 WEAK HASH

0 Confirmed → 47% за 3 итерации промпта (24 мая)

v1 · 10.05

общий secret-prompt

Модель ставила FP, потому что base prompt сказал «нет hardcoded value → False Positive». Хотя `md5(password)` — это уже Confirmed CWE-328.

v2 · 24.05 утром

блок-перебивка

Добавил шапку: «Базовый промпт написан для другой задачи. Для weak hash правила другие — следуй им.»

7 «уязвимостей» из 100 (recall ~25%).

v3 · 24.05 вечером

путь файла + имя аргумента

Добавил:
· сигналы по пути (/crypto/, AuthService...)
· сигналы по имени аргумента (md5(password|token...))

Результат: 47% recall.

НЕЗАВИСИМАЯ ПРОВЕРКА

458 issue по 5 CWE · прошёл руками, поставил свой вердикт, посчитал согласие

CWE	О чём	N решений	Согласие
CWE-256	пароль в открытом виде	88	100% 88/88
CWE-506	опасный встроенный код	50	100% 50/50
CWE-798	пароль в исходниках	76	97.4% 74/76
CWE-330	слабый случайный генератор	38	89.5% 34/38
CWE-321	хардкод ключа шифрования	51	68.6% 35/51
Σ	**семья «секреты / крипто»**	**303**	**92.7% 281/303**

ЧТО ЭТО ЗНАЧИТ

281 из 303

где модель решилась и где я подтвердил

Unknown не считаю —

не считаю пасы за вердикты.

22 расхождения — слайд 12

Это не «модель плохая» —
это конкретные паттерны,
которые она пока не видит.

22 РАСХОЖДЕНИЯ — РАЗБОР

три класса ошибок, по одному примеру каждый

CWE-321 **16 случаев · токен в документации** Модель: Confirmed · Я: думал FP

Передумал — настоящий JWT в публичной документации это leaked secret.
Модель консервативнее меня. В безопасности она права.

CWE-330 **4 случая · модель не смотрит на путь файла** Модель: FP · Я: Confirmed

Math.random() в /crypto/, /auth/. Имя файла кричит, модель путь не смотрит.
Тот же урок, что был для CWE-328 — сигналы по пути файла в промпт.

CWE-798 **2 случая · API-ключ в проде** Модель: FP · Я: Confirmed

application-smitsprod.properties: skk.api.production.api-key — реальный prod-ключ
UUID-формат сбил модель в FP. Поле api-key + prod в пути → должна была Confirmed.

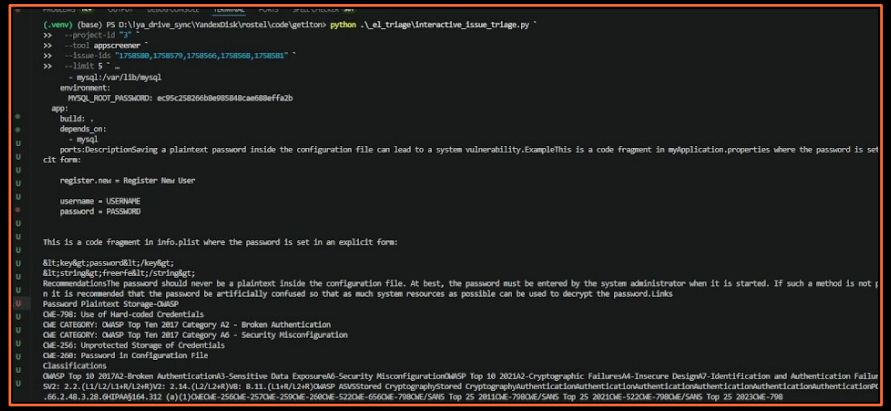
→ ДАЛЬШЕ — одно живое срабатывание через всю эту систему

DEMO · настоящий прогон, 5 issue, 85 секунд

Qwen2.5-Coder-14B-AWQ @ vLLM 192.168.1.126:8015 · скринкаст 26.05 03:42

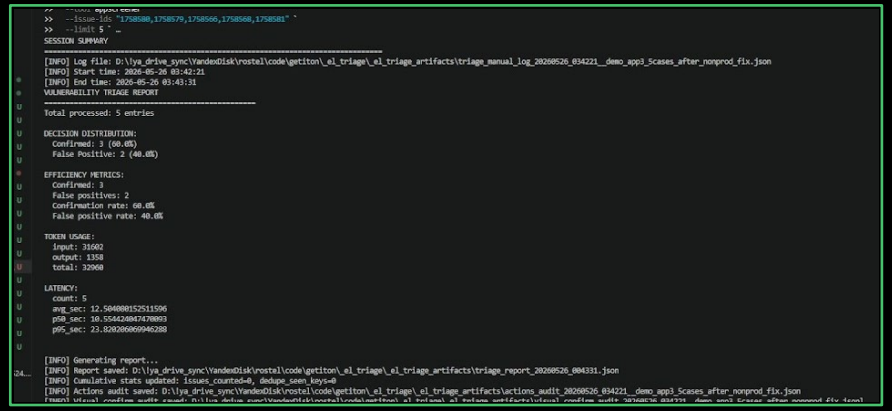
CASE 4/5 · DOCKER-COMPOSE SECRET

реальный MYSQL_ROOT_PASSWORD в .yaml · CWE-256



SESSION SUMMARY · финал прогона

Confirmed 3 (60%) · FP 2 (40%) · latency p95 = 23.8s



ЧТО ВИДНО НА ЭКРАНЕ

слева: модель сама сожмёт описание CWE и потянет реальный фрагмент кода · справа: 5 решений, 3 / 2, и точные цифры по latency и токенам

▶ *полное видео рядом: demo_recording_v7.mp4 (1m 25s) · PowerPoint → Insert → Video → drag поверх любой карточки*

ЧТО ЗАБРАТЬ С СОБОЙ

четыре принципа + ссылка на репо

4 ПРИНЦИПА

- **Дроби задачу**

  один большой промпт хуже
  трёх маленьких с контрактом

- **JSON-схема на замок**

  фиксируй формат, инвалид
  выкидывай не глядя

- **Эвристика до LLM**

  regex ловит 30% очевидного
  бесплатно — не трать токены

- **Маленький независимый эталон**

  245 размеченных вручную
  > 10k подкрученных

В РЕПО (MIT)

- system_prompt.txt
- cwe798_v1 -> v3
- cwe328_override.txt
- D-23_v2_trace_rule.txt
- json_response_schema.json
- evaluation_20260524.xlsx

весь стек живёт на железе,
на котором я по выходным играю в киберпанк.

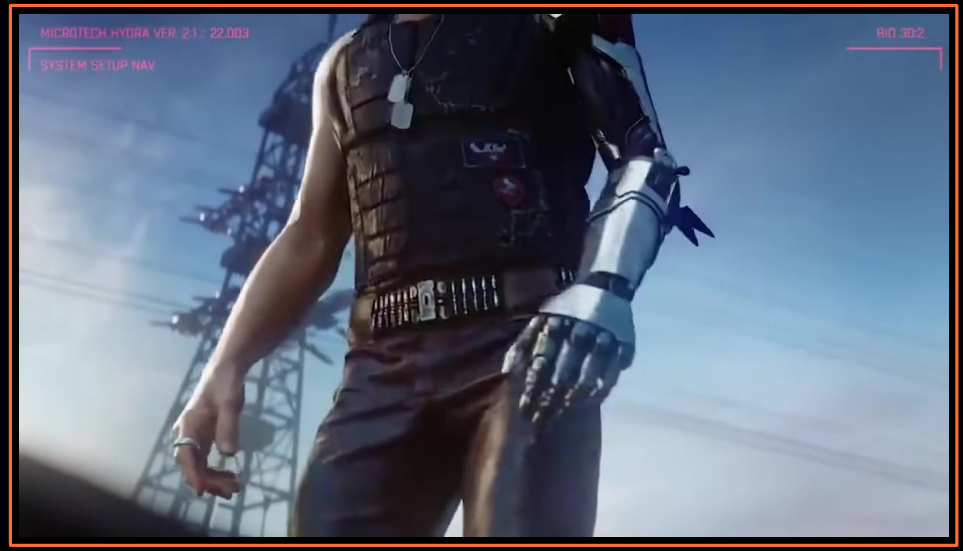
?

QUESTIONS · COMMENTS · DISAGREEMENTS

АВТОР

Туманов Юрий · Эльария

@psychodrake
github.com/Eljees
3.14hell@gmail.com



"Wake the fuck up, samurai. We have a city to burn."

— Johnny Silverhand · Cyberpunk 2077