2026

ОС Astra Linux:
нам по пути с РБПО

Тележников Владимир
Директор департамента
анализа безопасности, к.т.н.

Управление уязвимостями

Комплексный подход: в 2025 году

>47 000 (+39%)

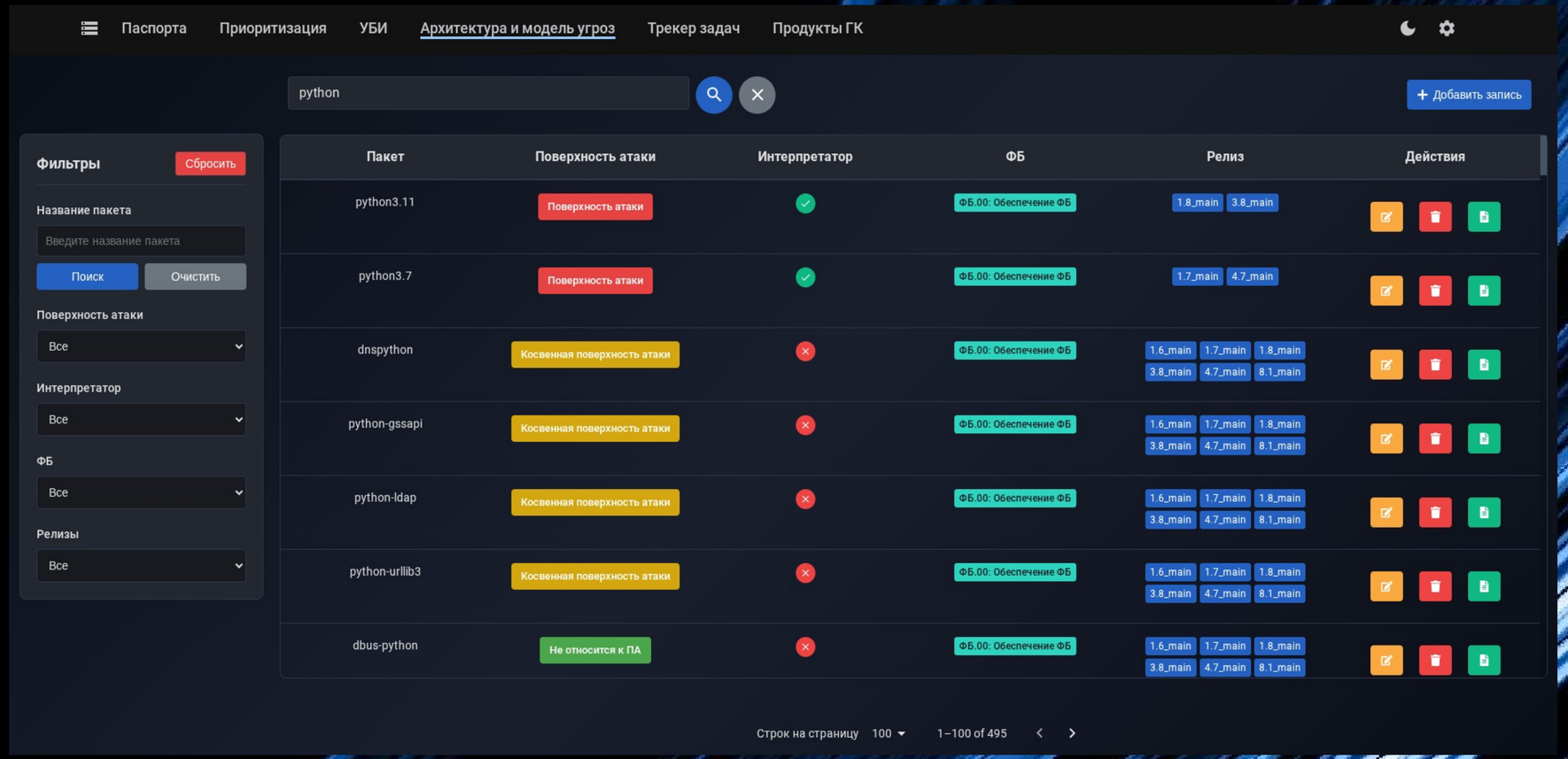
проанализировано
уязвимостей (CVE)

>11 000 (+93%)

паспортов уязвимостей (CVE)
отправлено в БДУ ФСТЭК России

>70 (+96%)

паспортов уязвимостей (ASE)
отправлено в БДУ ФСТЭК России



Управление уязвимостями

Контроль зависимостей

>4 000

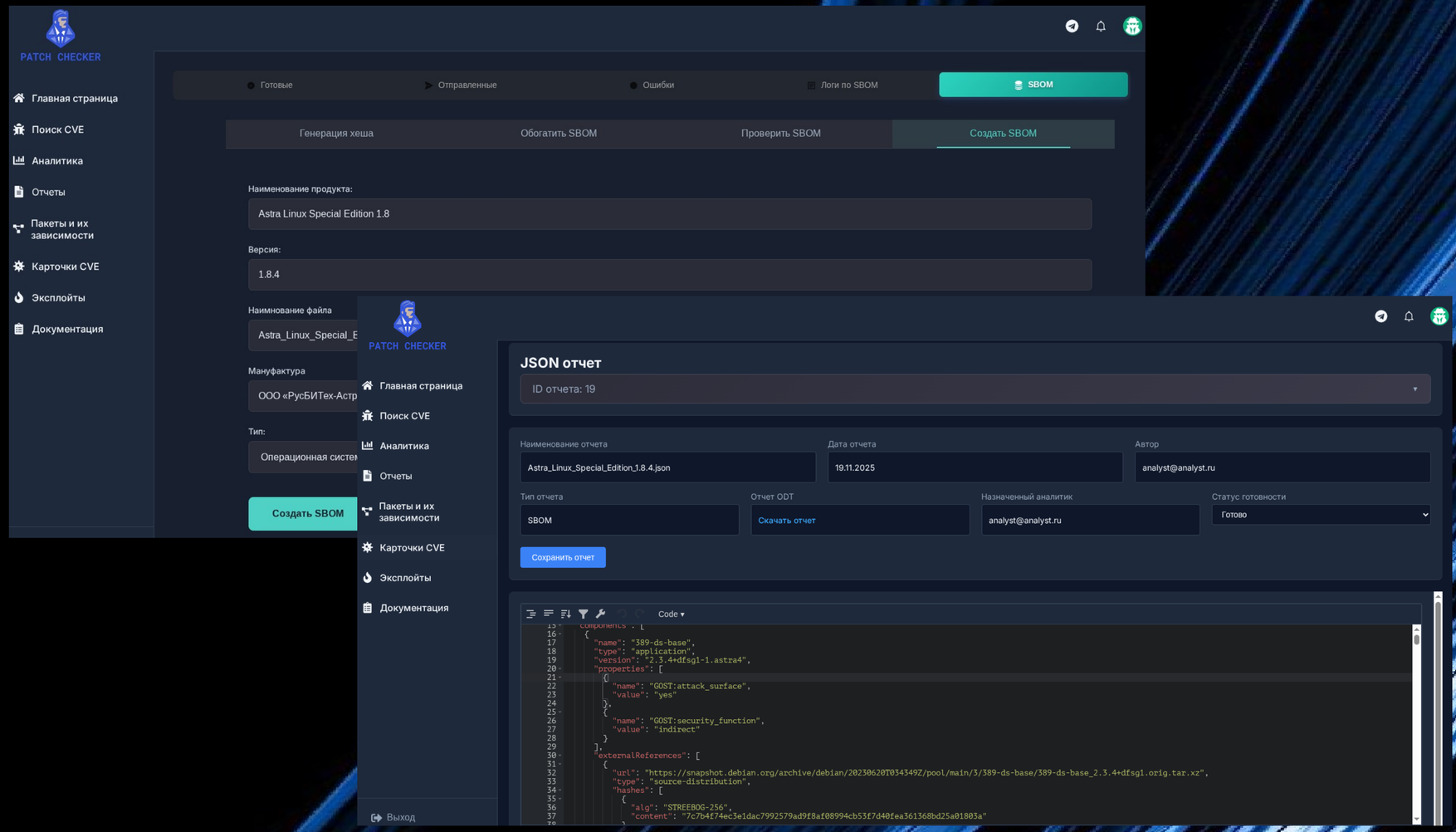
компонентов ОС покрывается непрерывным SCA

>20

выпусков обновлений ОС и продуктов экосистемы сопровождались отправкой SBOM во ФСТЭК России

>10

продуктов экосистемы проектируются с учетом SBOM ОС



Управление уязвимостями

Приоритизация

Трендовость

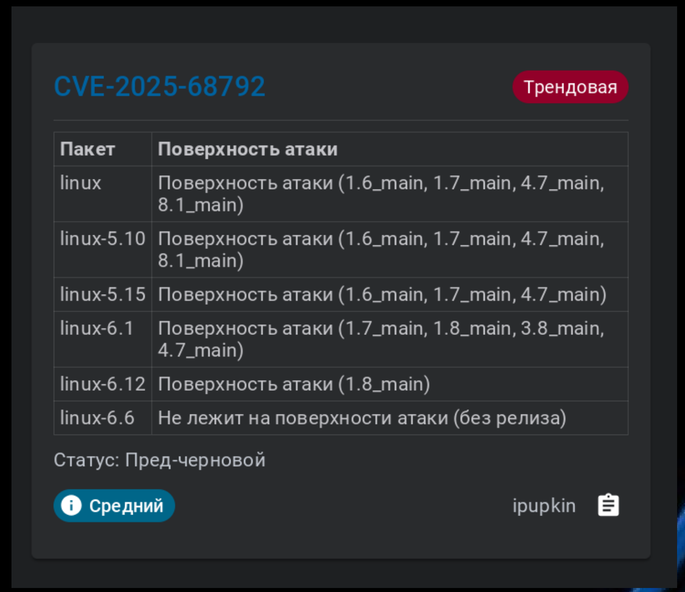
комплексная оценка критичности уязвимости по >10 показателям

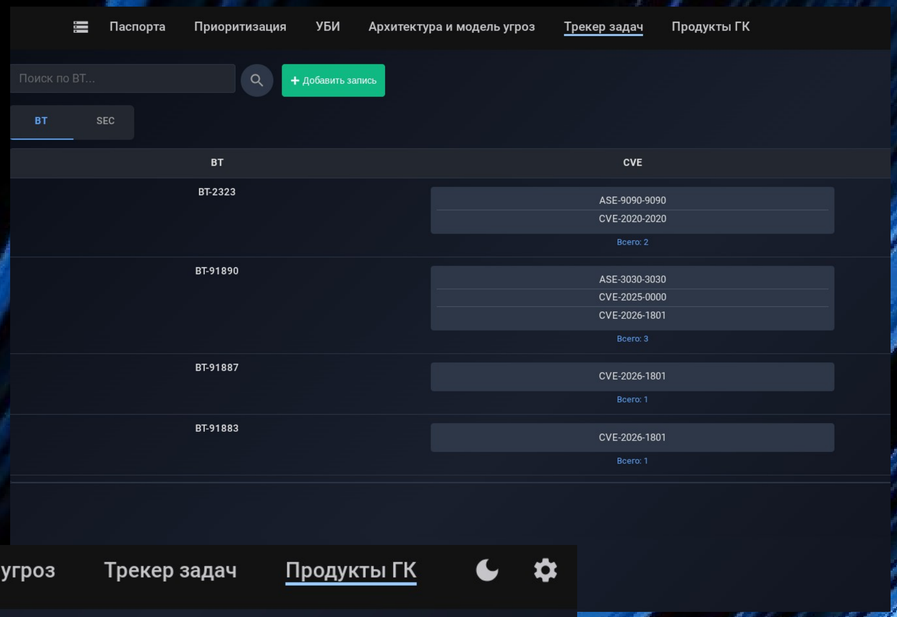
Экосистемность

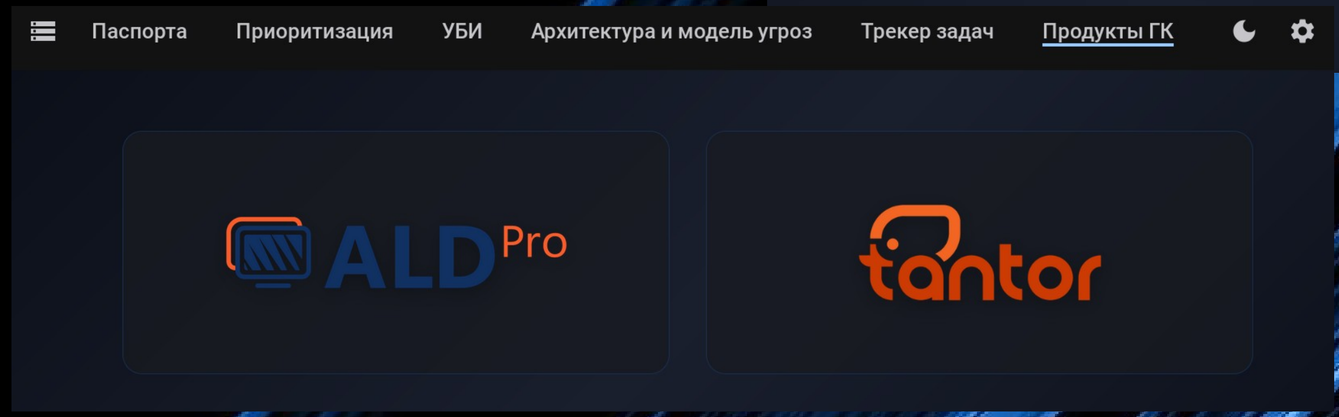
учет сценариев эксплуатации и поверхности атаки продуктов экосистемы

Непрерывность

постоянный контроль >30 источников информации



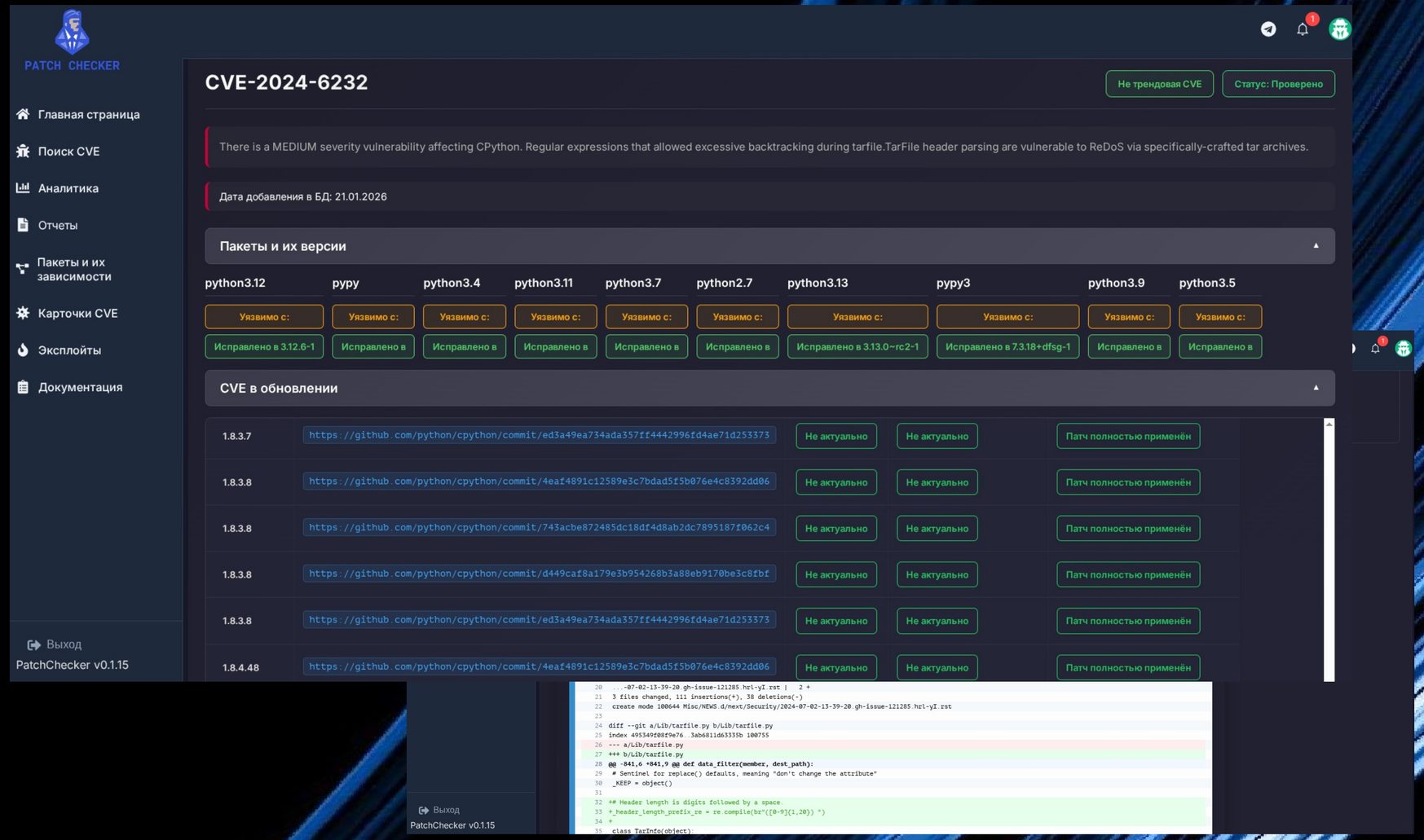




Управление уязвимостями

Контроль и генерация артефактов

- Автоматизированный контроль наличия патчей
- Регрессионный анализ PoC
- Методические рекомендации (MP) по нейтрализации угроз с учетом встроенных механизмов защиты
- Отправка и постоянный контроль данных в БДУ ФСТЭК России
- OVAL-описания с интегрированными MP
- Непрерывное взаимодействие с VM-вендорами



Выявление нежелательного контента

>170 решенных задач

анализ сетевой активности

непрерывное взаимодействие с клиентами

Пропаганда

- python3-cherrypy3
- php-voku-portable-ascii
- lua-busted
- gitlab-1st
- darktable
- arangodb3
- php-masterminds-html5
- php-symfony-framework
- и др.

Вредоносные ресурсы

- foreman-assets
- libonig-dev
- libmng-dev
- libimage-exiftool-perl
- libxml-rss-perl
- libccid
- fiona-doc
- texlive-latex-extra
- и др.

18+ и онлайн казино

- python3-wsaccel
- python-txaio-doc
- golang-github-go-
- entry-go-license-
- detector-dev
- openocd
- ruby-bacon
- nmap-common
- mariadb-test-data
- и др.

Статический анализ

База данных доверия

Автоматизация рутинных задач, регулярность и балансировка нагрузки

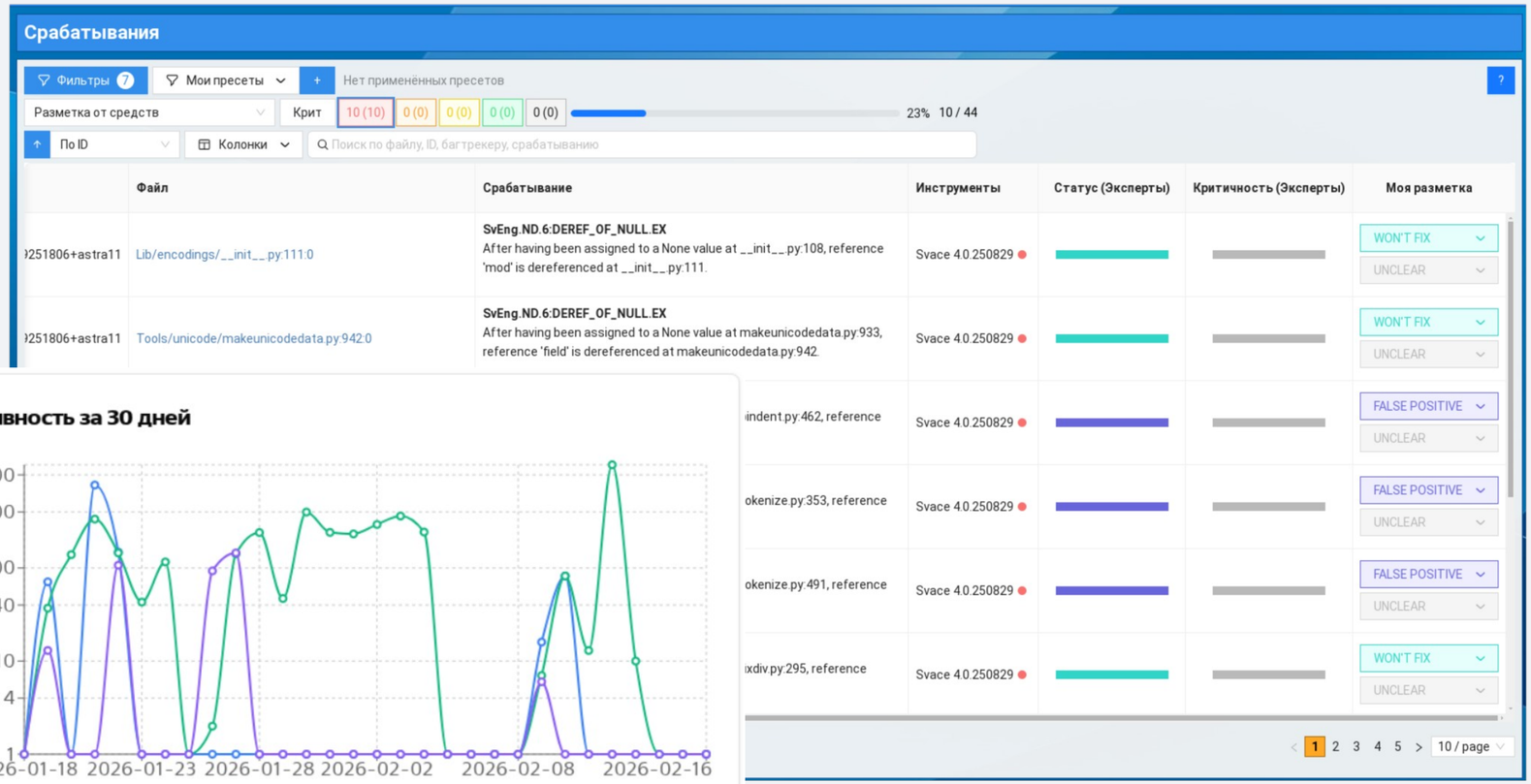
Выставление критериев, сбор статистики и метрик

Интеграция с инструментами CI и трекером задач

Единые правила настройки инструментов анализа

Распределение задач между командами и профильными специалистами

Перенос и наследование разметки с различных сервисов



Статический анализ

Применение SentinelAI

Если вы устали от...

- огромного объема результатов SAST-инструментов
- высокого процента FP-срабатываний
- обработки однотипных ошибок

Покрытие разметкой

Размеченные против неразмеченных уникальных срабатываний

Размечено	Неразмечен
45 158	571 756
7.32%	92.68%

Качество анализа

Проанализировано	Со статусом Svacer	Расхождений	Уровень расхождений
25115	24122	2597	10.8%

Типы расхождений

Svacer: Won't fix, Sentinel: false_posit…	1463
Svacer: Won't fix, Sentinel: confirmed	558
Svacer: False Positive, Sentinel: confir…	314
Svacer: Confirmed, Sentinel: false_po…	196
	46

Severity-аналитика

- Major 24537
- Critical 3860
- Normal 7380
- Minor 2077

Severity	Всего	Confirmed	FP	Won't Fix	Pending
Major	24537	1517	15199	850	6984
Normal	7380	466	3333	38	3543
Critical	3860	402	2089	9	1360
Minor	2077	182	1003	40	852

Статический анализ

Применение SentinelAI

- протестировано для C/C++, Python, Java, C#
- проверено на >700 проектах
- применимо для всех продуктов экосистемы
- позволило сэкономить уже более 2 ч/лет на разметке кода
- уже сейчас предлагает патчи для исправления

От ручного запуска до автоматизации и валидации на основе ML

353 д 13 ч

Сэкономленное время

106772ce-70e6-4f04-9958-73d05d04016b Завершено

Открыть в SVACER

/src/ausearch-avc.c

Скрыть исправление

Код исправления

```
// ... existing code ...

    newnode = malloc(sizeof(anode));
    if (newnode == NULL) {
        return;
    }

// ... existing code ...
```

Топ проектов по срабатываниям

1	qemu	3729	18 confirmed	Svacer: 2354 · Sentinel: 202
2	samba	3155	84 confirmed	Svacer: 873 · Sentinel: 873
3	ceph	2297	84 confirmed	Svacer: 1587 · Sentinel: 1602
4	php	2240	345 confirmed	Svacer: 1013 · Sentinel: 1138
5	zabbix	2066	9 confirmed	Svacer: 219 · Sentinel: 220
6	nss	2002	90 confirmed	Svacer: 528 · Sentinel: 387
7	edk	1761	140 confirmed	Svacer: 908 · Sentinel: 908
8	glibc	1541	100 confirmed	Svacer: 616 · Sentinel: 863
9	postgresql	1152	53 confirmed	Svacer: 730 · Sentinel: 943

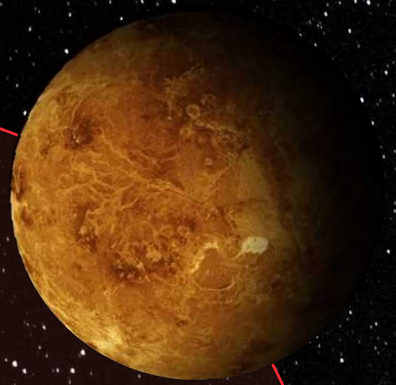
Распределение по Severity

Проект	Critical	Major	Normal	Minor
qemu	941	2294	470	24
samba	223	1318	1299	315
ceph	145	1231	573	348
php	200	1634	297	110
zabbix	194	1047	618	207

Svacer Confirmed · Sentinel Confirmed · Svacer FP · Sentinel FP · Svacer Won't Fix · Sentinel Won't Fix · Pending

Динамический анализ

Платформа AutoFuzz



- автоматизация сборки и запуска оберток
- запуск вручную, по коммитам или по расписанию
- дедупликация крешей
- регрессионное тестирование
- интеграция с БДД

Платформа автоматизации фаззинг-тестирования

>350

проектов интегрировано в платформу фаззинг-тестирования

>10

версий и конфигураций ядер ОС подвергаются фаззинг-тестированию на платформах x86_64, arm64, s390x

>5000

фаззинг-оберток применяются для тестирования затрагиваемых ими изменений кода

>400

обнаруженных и устраненных проблем безопасности

>800

ядер CPU обеспечивают непрерывность процессов фаззинг-тестирования

3&3

три инструмента фаззинга (AFL++, libfuzzer, atheris) и три языка программирования (C/C++, python, go) поддерживаются на платформе

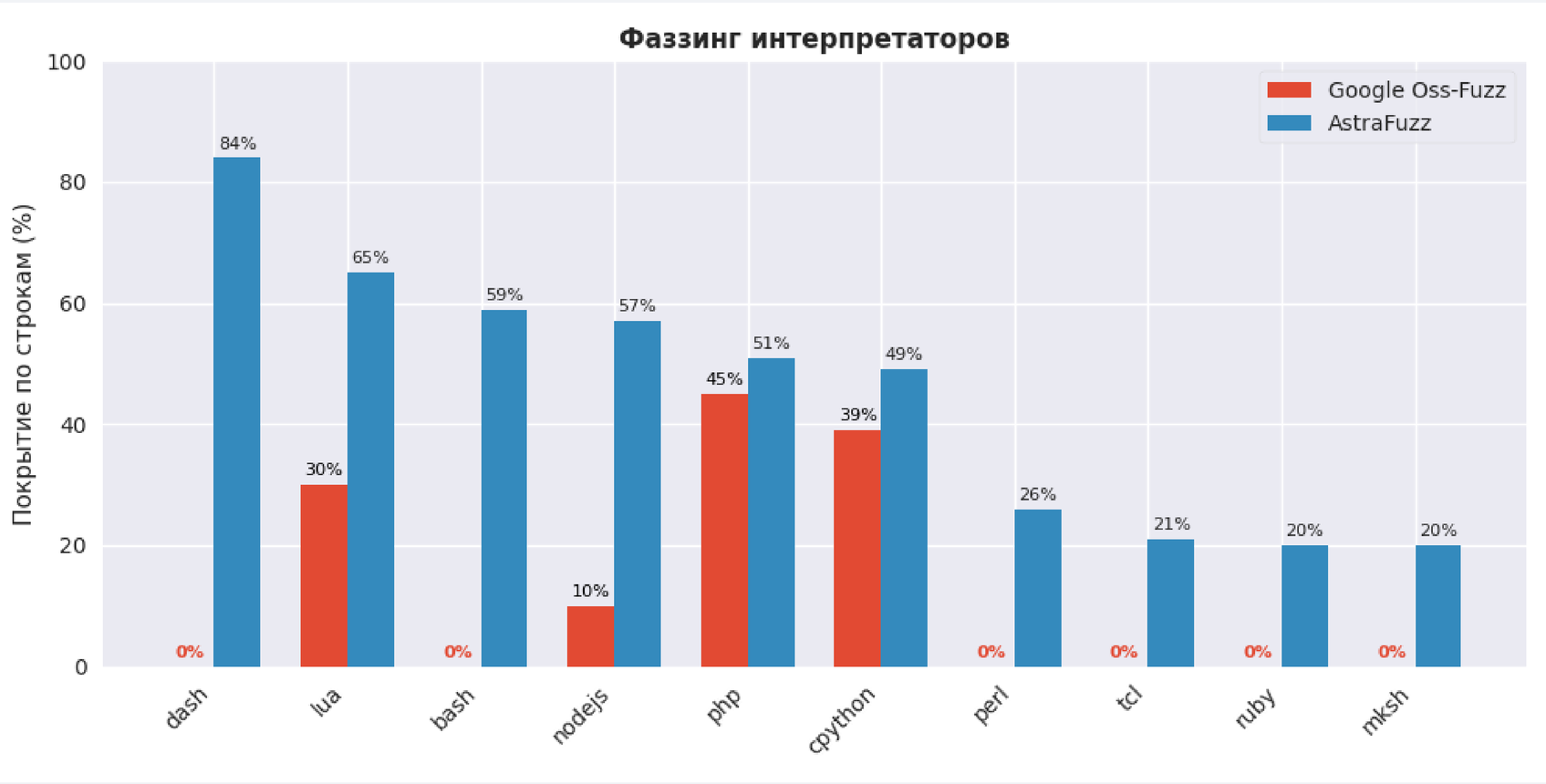
Динамический анализ

Фаззинг-тестирование

>80 выявленных и устраненных проблем безопасности

Непрерывное тестирование новых версий

Академический подход: эксперименты, методики, инструменты, сообщество



Модель угроз и поверхность атаки

- динамически меняется в ходе жизненного цикла ОС и продуктов экосистемы
- учитывается при развитии и разработке дополнительных механизмов безопасности продуктов экосистемы

Модель угроз

- включает более 160 актуальных УБИ
- оценивает каждую УБИ в отдельности и в рамках укрупненных групп
- постоянно актуализируется с учетом расширения сценариев применения
- формируется с учетом встроенных механизмов защиты

Детализация объектов воздействия

- ядро и модули ядра ОС
- средства защиты информации
- сетевые сервисы
- системное ПО
- графическая среда
- интерпретаторы
- прикладное ПО

Поверхность атаки

- детализируется с учетом уровня критичности последствий реализации УБИ:
  - высокий
  - средний
  - низкий
- пересматривается в ходе проведения анализа безопасности

Механизмы защиты ОС

- применимы для нейтрализации более 150 УБИ
- оцениваются по уровню эффективности
- развиваются для нейтрализации УБИ с учетом актуальной МУ
- составляют поверхность атаки

Основные инструментальные средства

Композиционный и статический анализ:

- Syft, Trivy, **ProtoPack, AVM, VulScan**
- Svace
- ClangSA
- AppScreener
- AK-BC 3
- CodeQl, semgrep, shellcheck
- **SentinelAI**

Динамический анализ:

- **Санитайзеры:** asan, lsan, kasan, ubsan, ...
- **Отладчики:** gdb, strace, ltrace, valgrind, uftrace
- **Сбор покрытия:** lcov, gcov, afl-cov
- **Фаззинг ядра:** syzkaller, **syz-ci**
- **Фаззинг:** crusher, afl++, libFuzzer, Sydr, go114-fuzz-build+go-fuzz-headers, **AutoFuzz**
- **А также:** klee, symcc, casr, ...

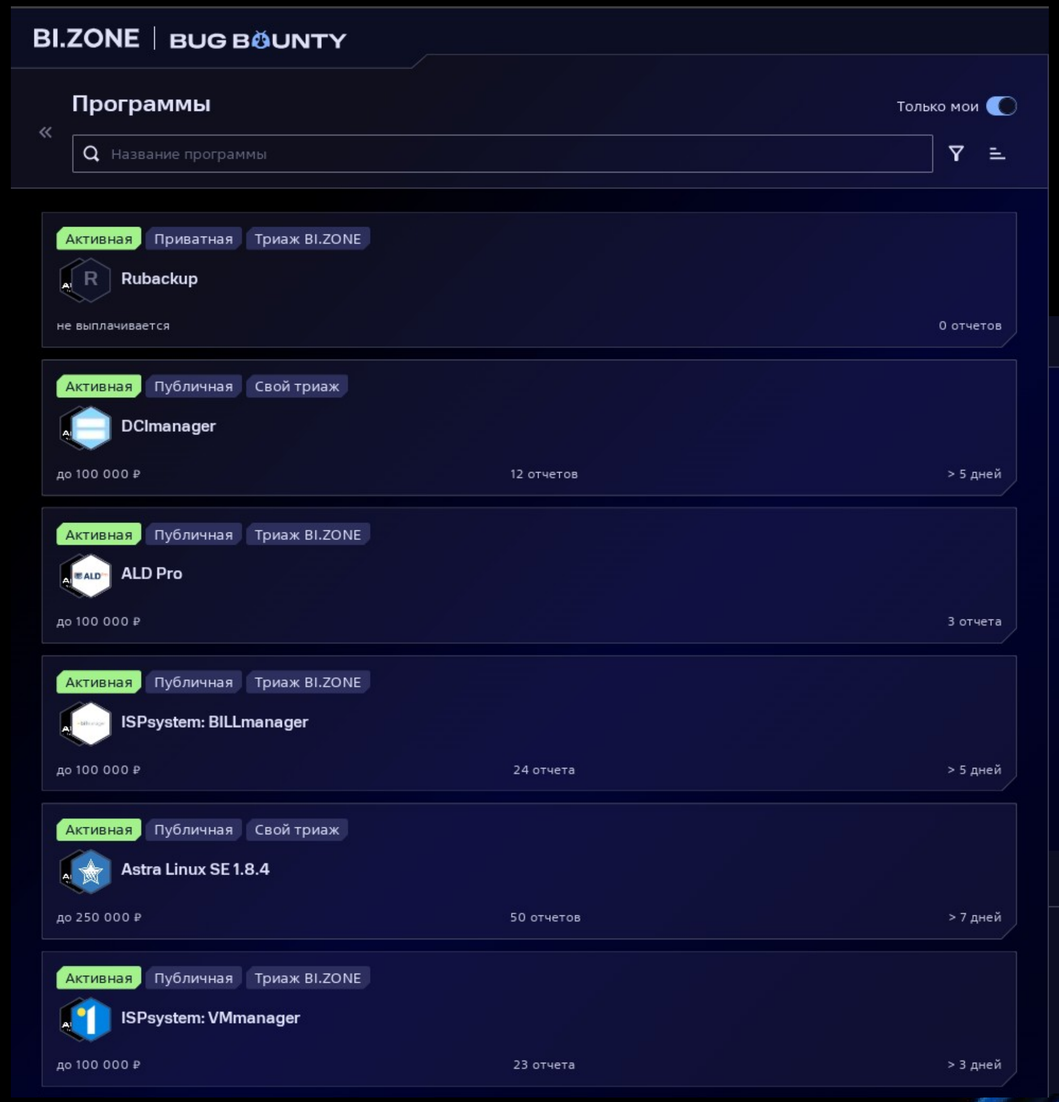
Средства верификации:

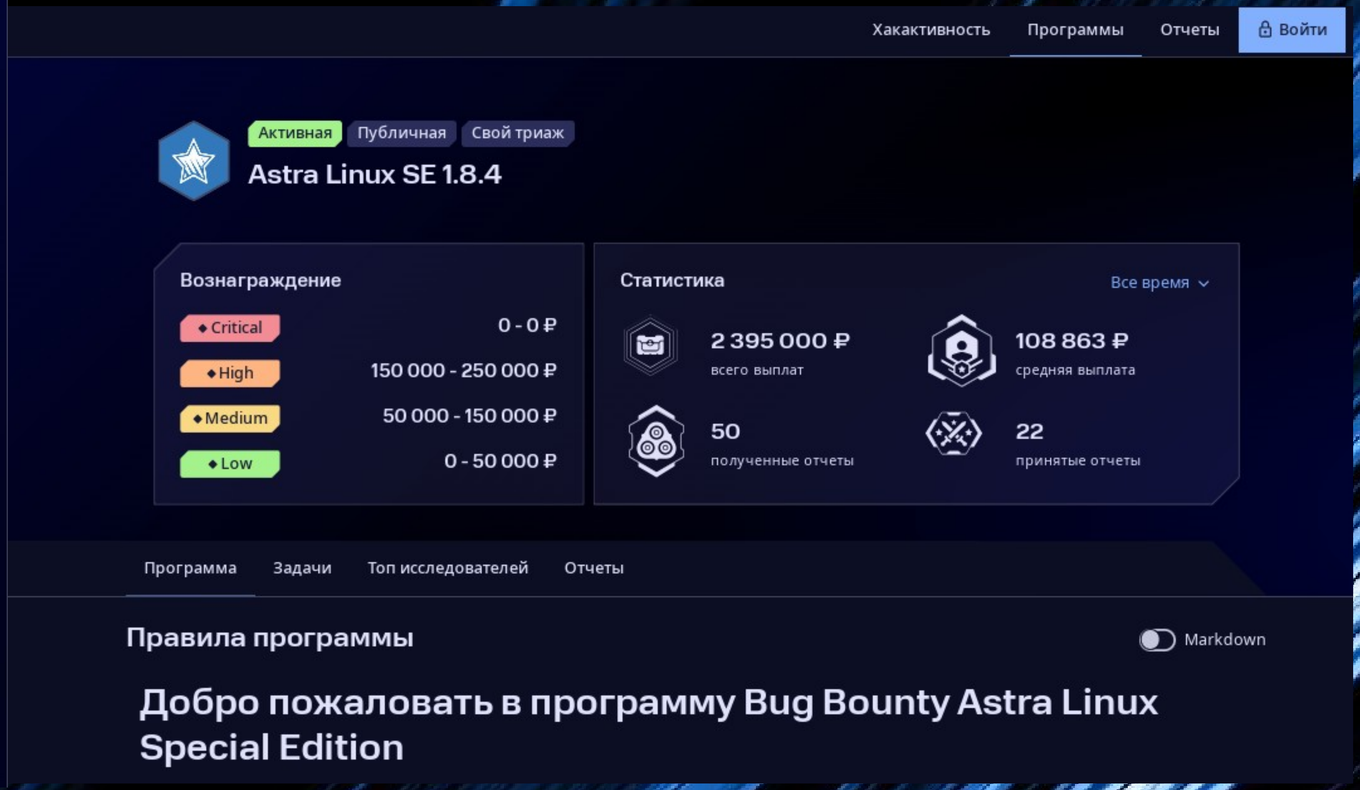
- Frama-C
- Verified Software Toolchain (VST)
- WP
- Why3
- Coq

А также:

- **Средства анализа ПА:** AttackSurfaceAnalyzer*, ...
- **Средства анализа помеченных данных:** «Блесна»
- **Средства тестирования на проникновение:** LinPEAS, Zap, Kali Linux*
- **Средства выявления нежелательного контента**
- **Средства антивирусного сканирования**
- **...**

Программа «Bug Bounty»





СПАСИБО ЗА ВНИМАНИЕ

Тележников Владимир
Директор департамента
анализа безопасности, к.т.н.