SOLAR

УЯЗВИМОСТИ ПОД КОНТРОЛЕМ:

Как автоматизировать анализ кода и обеспечить безопасность разработки

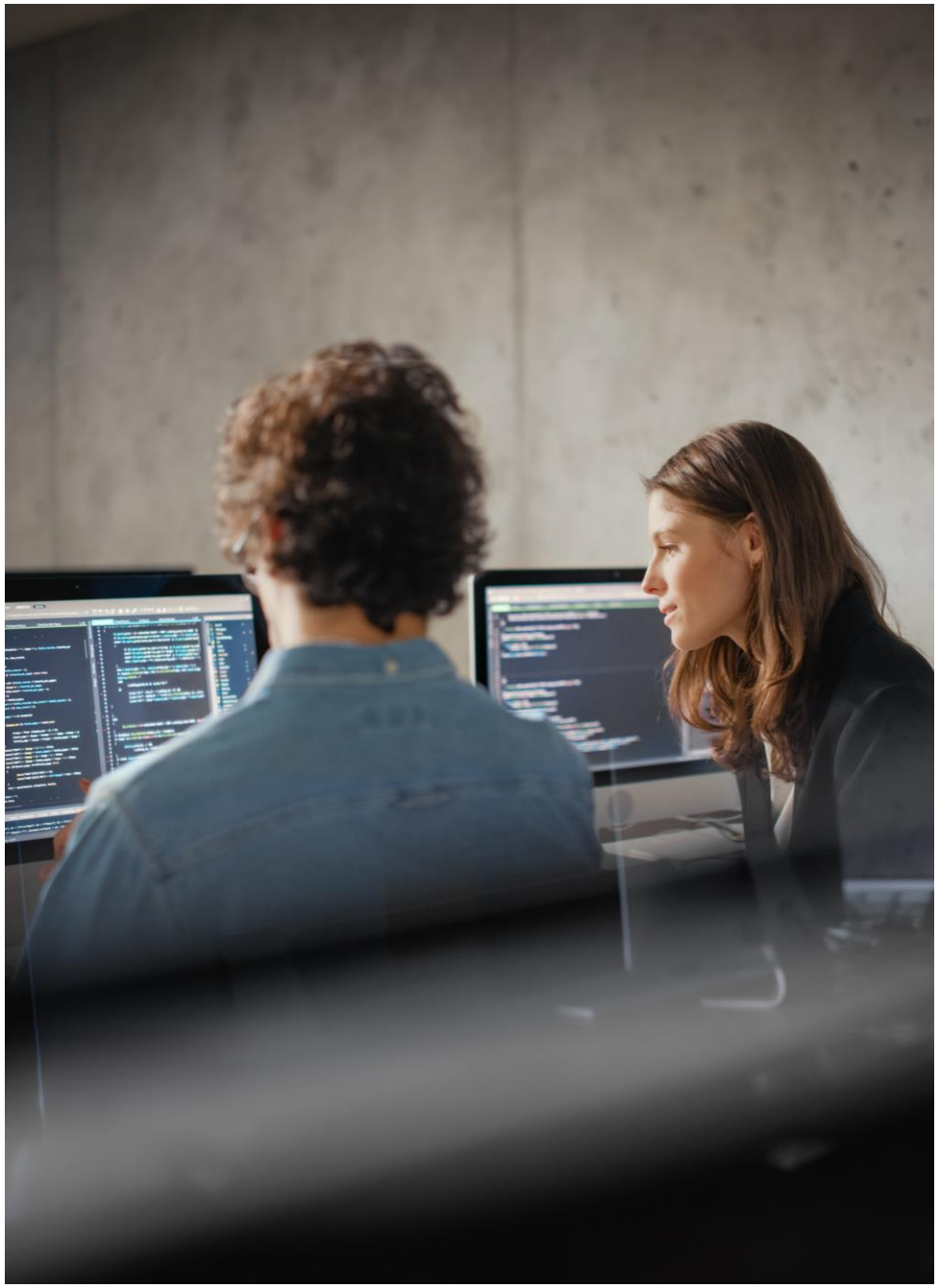
ЛЕВ АРМАНШИН
Эксперт центра разработки решений по контролю безопасности ПО
ГК «Солар»

Зачем проверять безопасность приложений

Обеспечение безопасности должно начинаться с анализа уязвимых точек, которые могут стать началом атаки – например, с анализа безопасности приложений.

54%

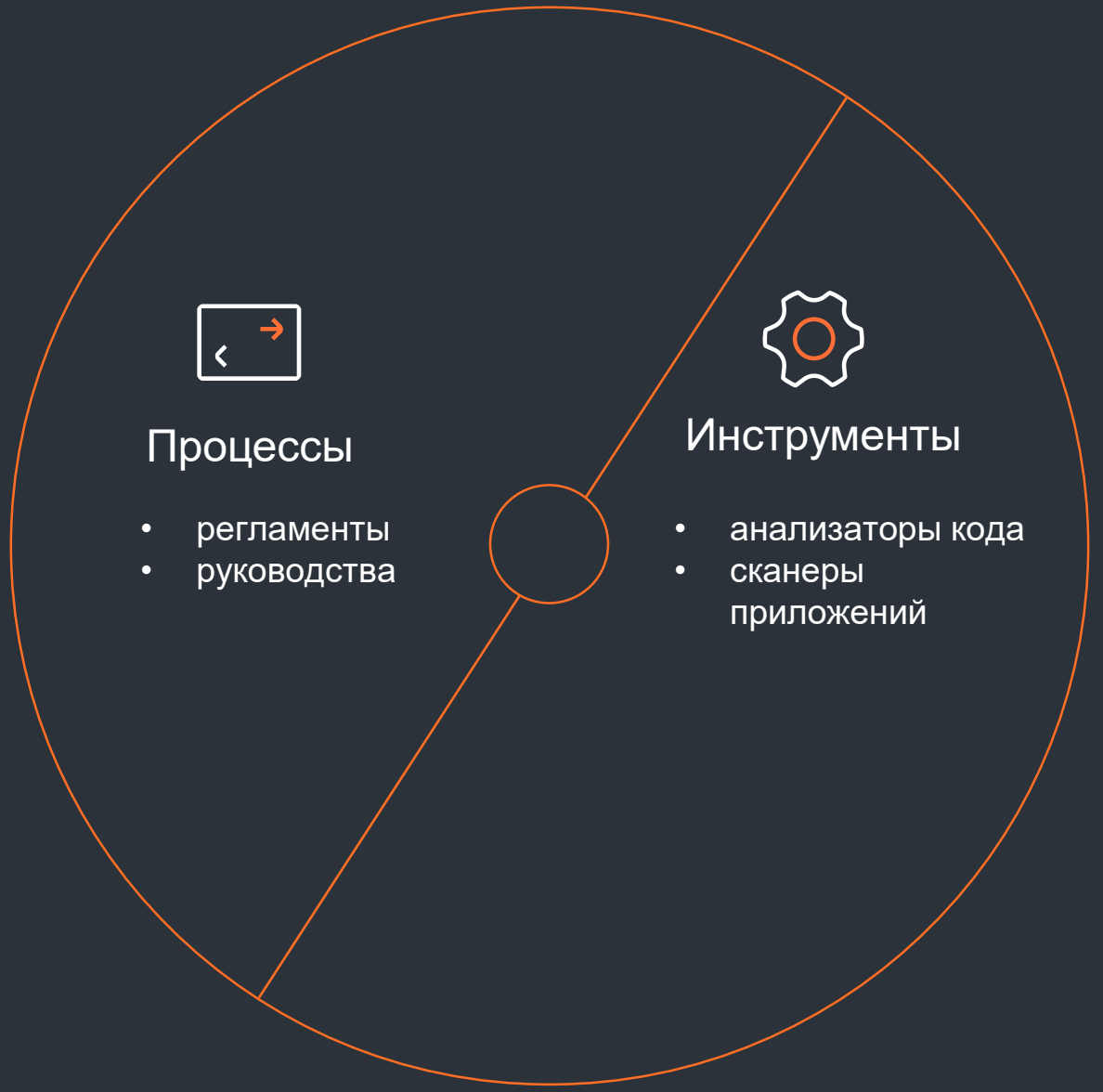
инцидентов происходят из-за уязвимостей в веб-приложениях



Из чего состоит безопасная разработка

SOLAR

- Команда разработчиков должна быть замотивирована писать код безопасно
- Это возможно только если безопасность не тормозит процесс разработки
- Для этого нужны автоматизированные инструменты



Ключевые виды инструментов для анализа кода

SOLAR

СТАТИЧЕСКИЙ АНАЛИЗ КОДА (SAST)

Проверяет исходный или бинарный код приложения на наличие уязвимостей как на ранних, так и на более поздних этапах разработки ПО

ДИНАМИЧЕСКИЙ АНАЛИЗ КОДА (DAST)

Проверяет работающее приложение в конце жизненного цикла разработки ПО или когда оно уже установлено в инфраструктуре

АНАЛИЗ СТОРОННИХ КОМПОНЕНТОВ (OSA)

Выявляет угрозы, связанные с уязвимостями и бэкдорами в open-source-библиотеках, и другие опасности использования сторонних компонентов в коде

Статический анализ кода (SAST)

Статический анализ кода (SAST) позволяет выявить уязвимости и недекларированные возможности в коде приложений.

[01]

Максимальное количество выявленных уязвимостей за счет анализа исходного кода

[02]

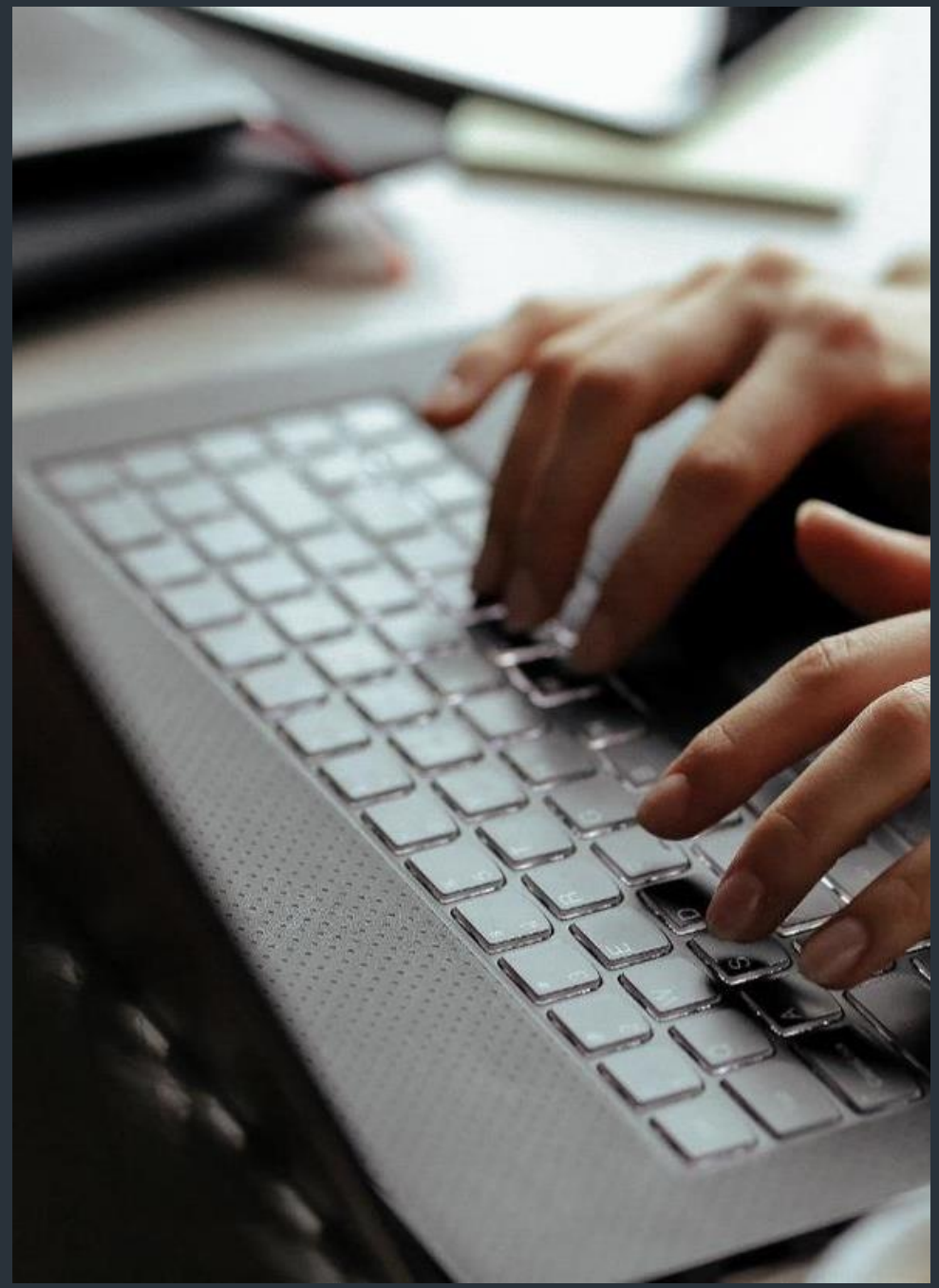
Выявление уязвимостей на ранних этапах разработки

[03]

Возможность инкрементального сканирования

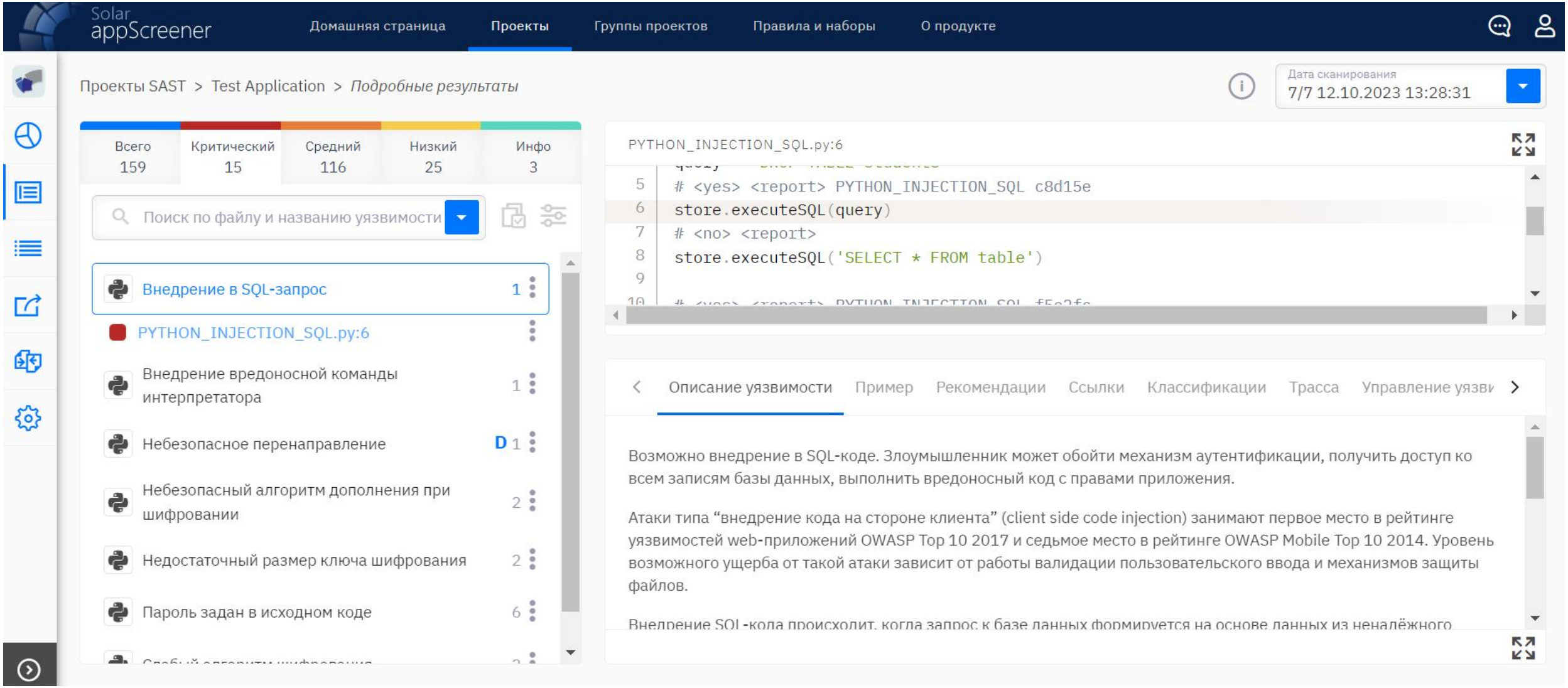
[04]

Большой процент ложных срабатываний



Как работает SAST

SOLAR



Динамический анализ кода (DAST)

SOLAR

Позволяет выявлять уязвимости непосредственно при выполнении программы

DAST имитирует вредоносные внешние атаки, использующие распространенные уязвимости для компрометации приложения

Использование при отсутствии исходного кода

Не привязан к языкам программирования

Минимальное количество ложных срабатываний

Возможно большое количество ненайденных уязвимостей и НДВ

Анализ сторонних компонентов (OSA)

SOLAR

~20%

кода разработчики пишут сами

Open Source Analysis (OSA) - комплексный анализ для защиты от угроз, связанных с open-source-компонентами.

РЕШАЕМЫЕ ЗАДАЧИ

- Выявление всех сторонних компонентов в приложениях
- Своевременное выявление уязвимостей и бэкдоров в open-source-библиотеках
- Предотвращение угроз и снижение рисков ИБ из-за заимствования кода

РЕЛЕВАНТНЫЕ ВИДЫ АНАЛИЗА

АНАЛИЗ СОСТАВА ПО (SCA)

АНАЛИЗ ЦЕПОЧКИ ПОСТАВОК ПО (SCS)

АНАЛИЗ ЛИЦЕНЗИОННЫХ РИСКОВ

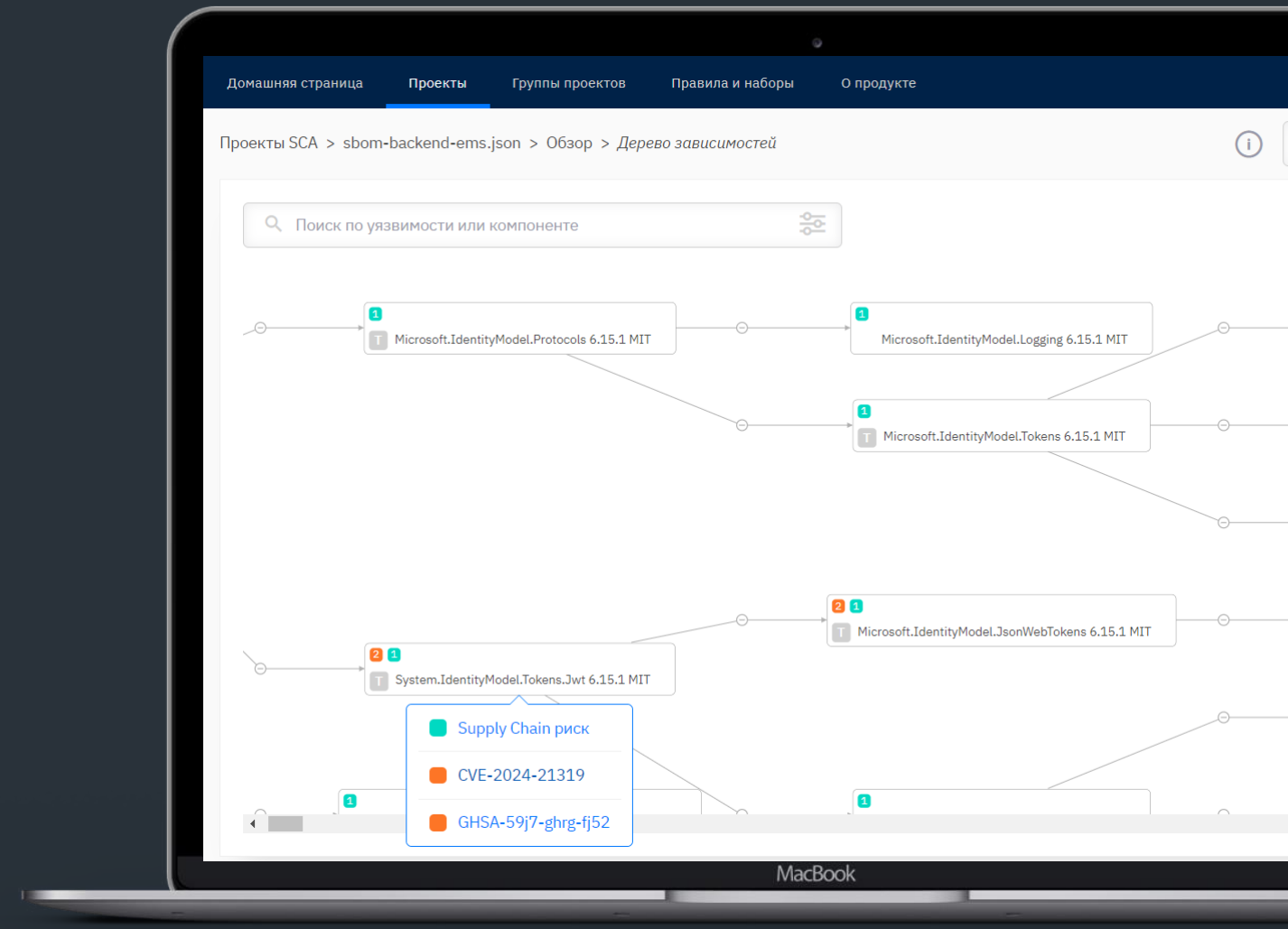
Анализ состава ПО (SCA)

SOLAR

Анализ состава ПО (SCA) позволяет проверять безопасность сторонних библиотек, используемых разработчиками в своем ПО.

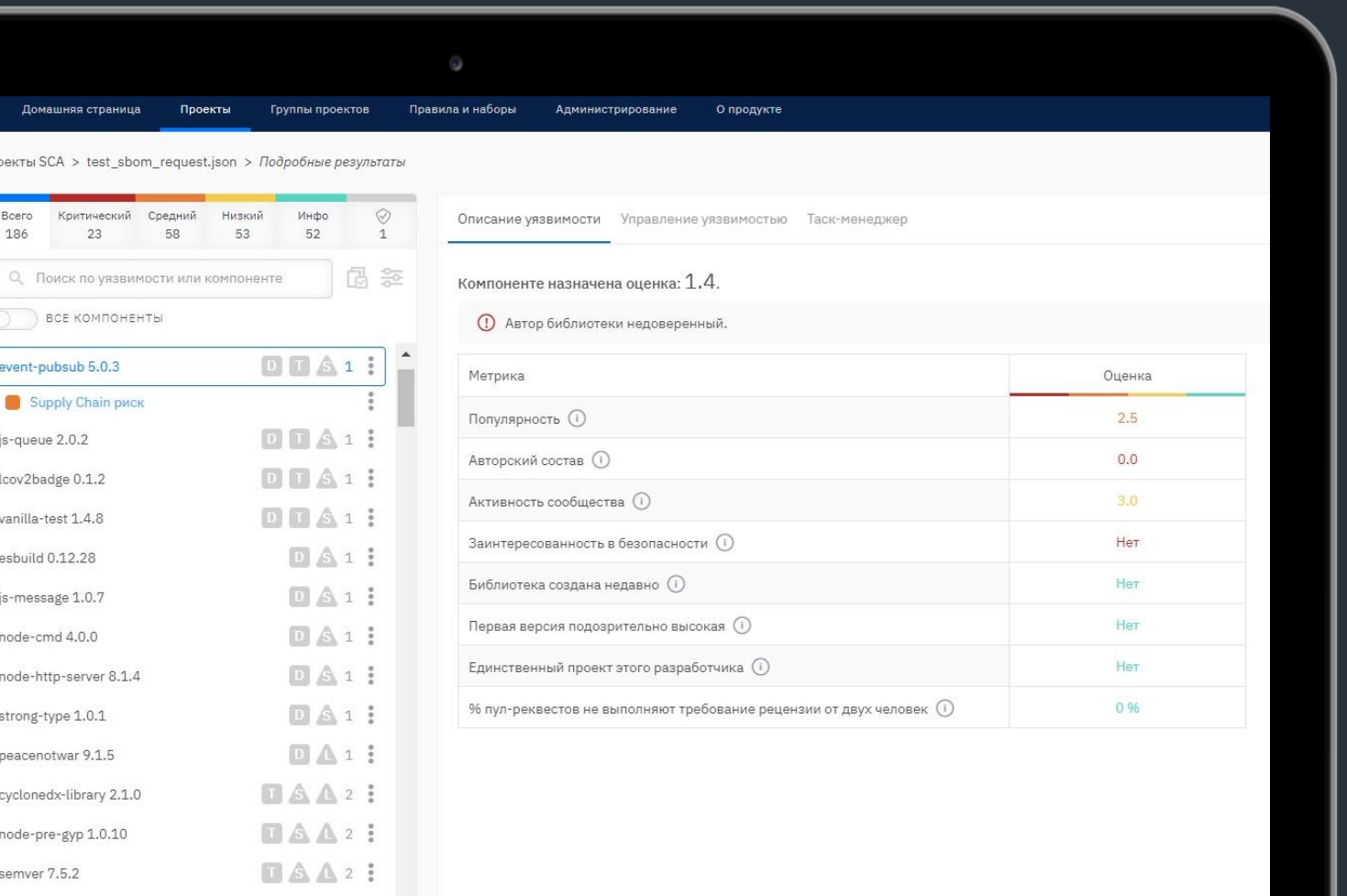
РЕШАЕМЫЕ ЗАДАЧИ:

- Обнаружение и отслеживание всех сторонних компонентов в ПО
- Выявление известных уязвимостей и закладок в сторонних библиотеках
- Предотвращение угроз и снижение рисков ИБ из-за заимствования кода



Анализ цепочки поставок ПО (SCS)

Анализ цепочки поставок (Supply Chain) выявляет потенциально опасные библиотеки, даже если известные уязвимости в них пока не обнаружены.



РЕШАЕМЫЕ ЗАДАЧИ:

- Мониторинг подозрительной активности в используемых сторонних библиотеках
- Снижение рисков ИБ из-за использования заимствованного кода
- Защита от атак supply chain

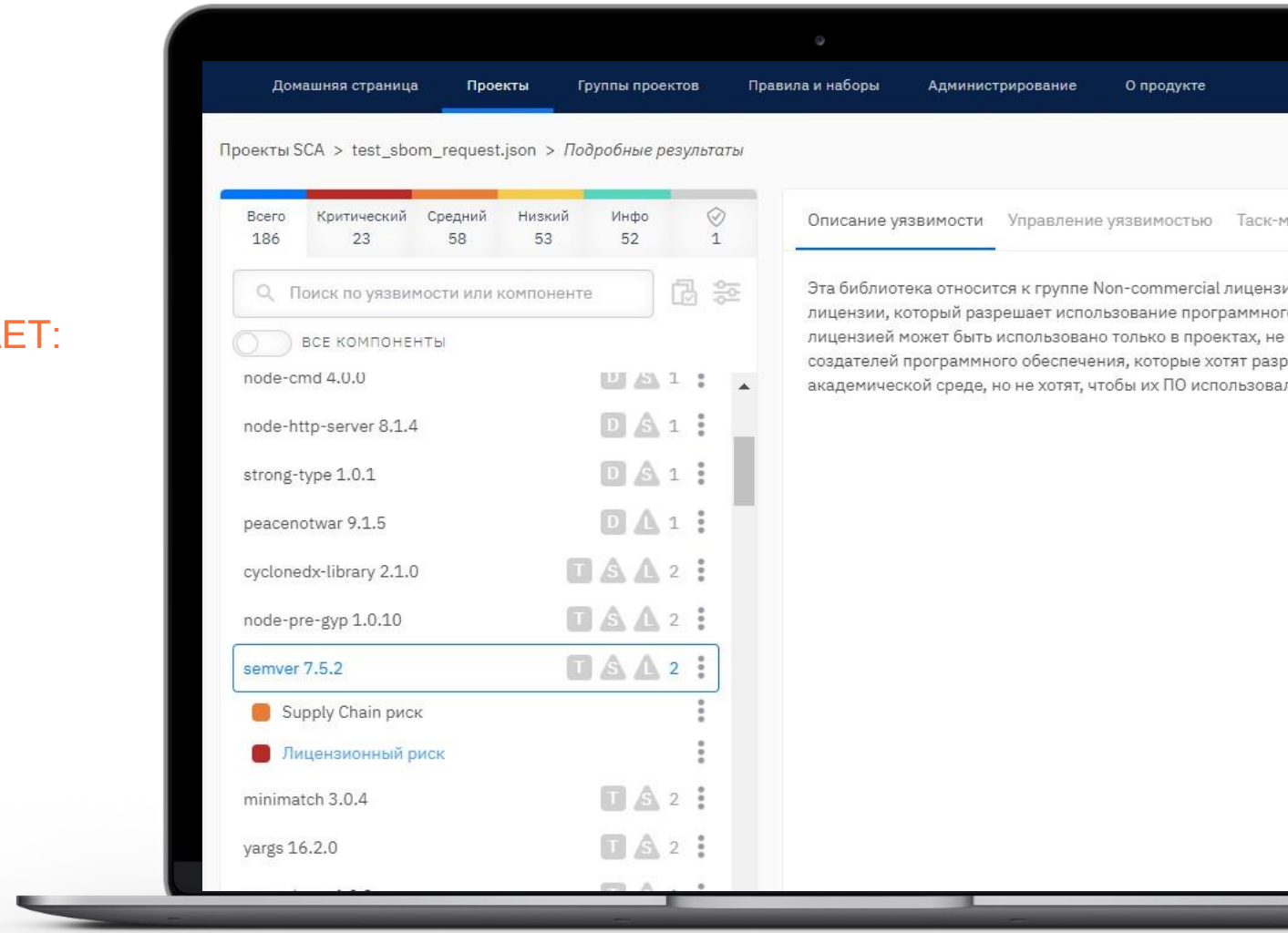
Анализ лицензионных рисков

SOLAR

Не весь open source распространяется свободно!

ОЦЕНКА ЛИЦЕНЗИОННЫХ РИСКОВ ОБЕСПЕЧИВАЕТ:

- Мониторинг лицензионной политики стороннего ПО в вашем приложении
- Снижение юридических рисков, связанных с лицензированием open source: отсутствие лицензии, несовместимость лицензий разных компонентов и т. д.
- Рекомендации по устранению найденных угроз



Корреляция инструментов анализа кода

SOLAR

SOLAR APPSCREENER

СВЯЗЬ МОДУЛЕЙ

Корреляция SAST и DAST:
Результаты статического анализа можно верифицировать с помощью DAST.

Это помогает выделить приоритетные уязвимости и устранить их в первую очередь

Комбинированный анализ OSA и SAST:
SAST проверяет, представляют ли угрозу уязвимости, найденные OSA-анализом.

Это сокращает ложные срабатывания и выявляет только истинные уязвимости.

Интеграция со средствами разработки

SOLAR

РЕПОЗИТОРИИ

git

SUBVERSION

VCS-ХОСТИНГИ

GitHub

GitLab

Bitbucket

СРЕДСТВА РАЗРАБОТКИ

eclipse

Visual Studio

IJ

СРЕДСТВА СБОРКИ

Xcode

GNU (Autotools, Make)

Visual Studio

sbt

Maven

Gradle

CMake

СЕРВЕРЫ CI/CD

Jenkins

TeamCity

Azure DevOps Server

ОТСЛЕЖИВАНИЕ ЗАДАЧ

Atlassian JIRA

ТУРБО

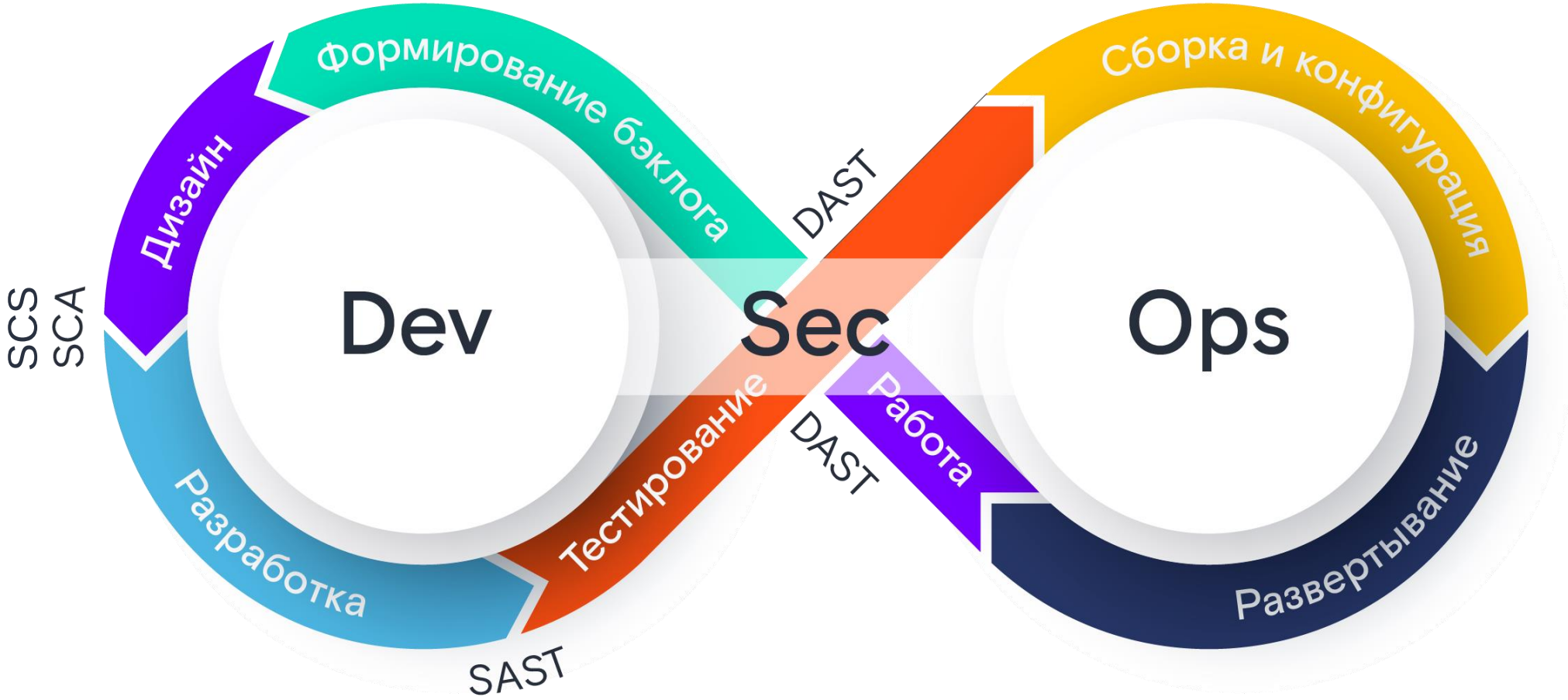
АНАЛИЗ КОДА

sonarqube

Широкие возможности по дополнительной интеграции и автоматизации благодаря открытому API.
Включает в себя JSON API и CLI.

Как встроить инструменты в цикл безопасной разработки

SOLAR



Подход Solar appScreener

Solar appScreener – комплексное решение для контроля безопасности ПО, которое включает технологии статического (SAST), динамического анализа (DAST) и анализа сторонних компонентов (OSA).

Сочетает преимущества всех ключевых методов анализа в одном решении

Позволяет быстро и удобно управлять сканированиями из единого интерфейса

Автоматизирует анализ кода, чтобы освободить ресурсы ваших специалистов и сэкономить время

Удобно встраивается в цикл разработки ПО

SOLAR

Бесплатно проверьте ваш код с Solar appScreener

СПАСИБО ЗА ВНИМАНИЕ!

ЛЕВ АРМАНШИН
Эксперт центра разработки решений по контролю безопасности ПО, ГК «Солар»
l.armanshin@rt-solar.ru