Ростелеком
Солар

Центр контроля безопасности программного обеспечения

На базе анализатора безопасности приложений Solar appScreener

rt-solar.ru

Зачем контролировать безопасность ПО

Программное обеспечение (ПО) используется в деятельности любой организации. С его помощью осуществляются все ключевые бизнес-процессы — прием и обработка заявок клиентов, финансовые транзакции, бухгалтерский учет и т. д. При этом любое ПО содержит уязвимости — неумышленные ошибки, нестыковки и неточности, которые позволяют его взломать, и недекларированные возможности (НДВ) — скрытую функциональность, умышленно внесенную в код.

Большинство используемого ПО имеет веб-доступ или работает на устройствах, подключенных к корпоративной сети — а значит, его безопасность может повлиять на непрерывность бизнес-процессов и финансовую стабильность организации.

№1
веб-атаки — главный инструмент взломов и утечек

72%
вторжений за периметр связаны с веб-уязвимостями

65%
вторжений приводят к полному контролю данных

100%
веб-приложений содержат уязвимости

Данные: Solar JSOC и Positive Technologies, 2018

Необходимость Центра контроля безопасности ПО

Организациям, которые зависят от безопасности используемого ПО, необходимо системно работать над выявлением и устранением возможных уязвимостей и НДВ. В этом поможет корпоративный или ведомственный Центр контроля безопасности ПО, предоставляющий совокупность технических инструментов, практик и процессов анализа ПО. Такой комплекс средств позволяет вносить изменения в ПО или настройки WAF* и обеспечивать надлежащий уровень безопасности.

Центр контроля безопасности ПО необходим, если организация:


Разрабатывает собственное ПО


Использует заказное ПО от внешних подрядчиков


Применяет унаследованное или старое ПО


Использует ПО в ключевых бизнес-процессах


Сталкивается с инцидентами, влияющими на репутацию


Не может обновить критически важное ПО


Должна оперативно блокировать уязвимости

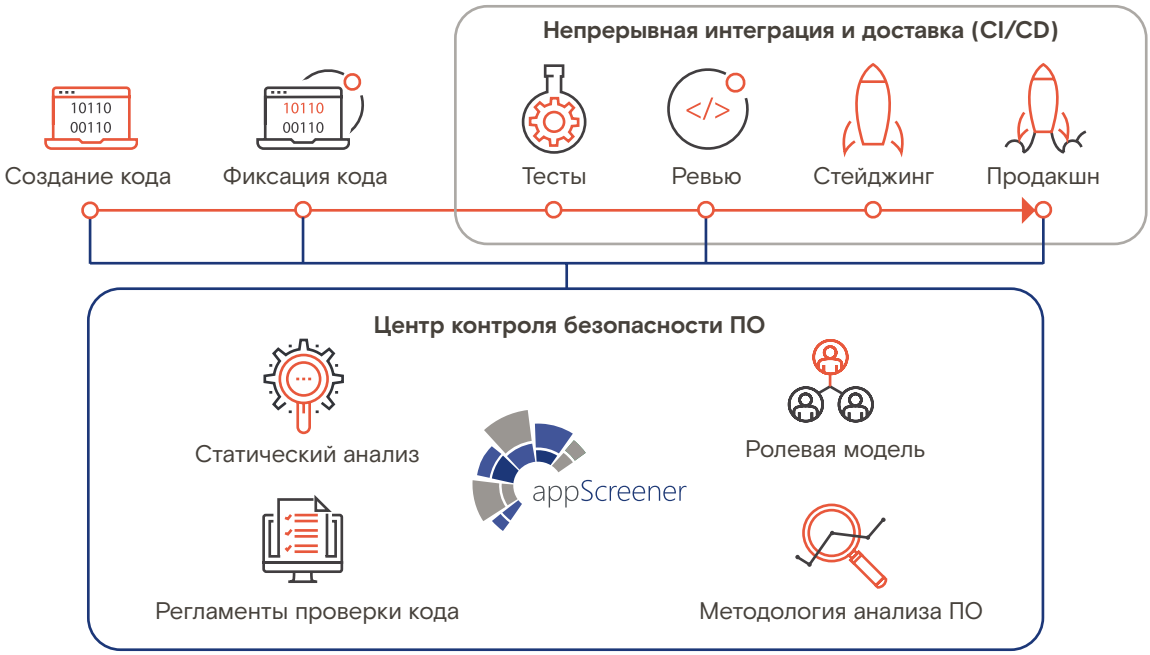

Обязана выполнять требования регуляторов

*Web Application Firewall — межсетевой экран уровня приложений (L7 модели OSI)

Принцип работы Центра контроля безопасности ПО

Основой Центра контроля безопасности ПО является статический анализатор кода, позволяющий проверять ПО на наличие уязвимостей и НДВ. При этом необходимо применять правильную методологию анализа ПО, использовать подходящую ролевую модель, учитывающую всех участников процесса разработки или приемки ПО, а также соблюдать все регламенты проверки кода.

Компания «Ростелеком-Солар» предлагает услуги по созданию корпоративного или ведомственного Центра контроля безопасности ПО на основе собственного статического анализатора кода Solar appScreener. Его уникальной особенностью является возможность анализа не только исходного, но и бинарного кода (исполняемых файлов), что позволяет анализировать унаследованное ПО и заказные разработки, передаваемые без исходного кода.

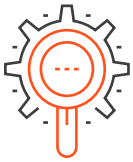


Цели внедрения Центра контроля безопасности ПО

Развертывание Центра контроля безопасности ПО позволяет:

- перейти на цикл безопасной разработки ПО (Secure SDLC) для оперативного устранения уязвимостей и НДВ;
- применять концепции непрерывной интеграции и поставки (CI/CD), сокращая время разработки ПО без ущерба для безопасности;
- регулярно анализировать безопасность кода и исполняемых файлов от сторонних разработчиков;
- максимально быстро закрывать уязвимости и НДВ компенсационными мерами, оперативно перенастраивая WAF;
- обеспечить соответствие требованиям методических рекомендаций по созданию ведомственных и корпоративных центров ГосСОПКА № 149/2/7-200 от 27.12.2017 г., согласно которым необходимо выявлять известные уязвимости ПО;
- обеспечить соответствие требованиям ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования», обязывающего контролировать отсутствие уязвимостей и НДВ.

Возможности Центра контроля безопасности ПО



1. Анализ исходного кода

Поддерживается проверка следующих языков: ABAP, Apex, ASP.NET, COBOL, C#, C/C++, Objective-C, Dart, Delphi, Go, Groovy, HTML5, LotusScript, Java, Java for Android, JavaScript, JSP, Kotlin, Pascal, Perl, PHP, PL/SQL, T/SQL, Python, Ruby, Rust, Scala, Solidity, Swift, TypeScript, VBA, VB.NET, VBScript, Visual Basic 6.0, Vyper, 1C

2. Анализ исполняемых файлов

Технологии декомпиляции и деобфускации кода позволяют анализировать исполняемые файлы следующих форматов: JAR, WAR, EAR, AAR, DLL, EXE, APK, IPA, APP. Для проверки мобильных приложений достаточно скопировать в анализатор ссылку на соответствующую страницу в Google Play или App Store.

3. Выявление уязвимостей

Уязвимости выявляются на основе сложных правил после завершения всех процедур анализа и работы Fuzzy Logic Engine. Применение технологии SCA позволяет выявлять уязвимости не только в собственном коде, но и компонентах на основе свободного ПО.

4. Выявление НДВ

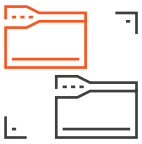
НДВ выявляются по наличию одной из характерных базовых конструкций: хардкодных учетных записей, скрытой сетевой активности, временных бомб и т. д. Наличие базовых конструкций НДВ может свидетельствовать о присутствии более сложной составной закладки.

5. Проверка унаследованногои заказного ПО

Реализованные в Solar appScreener технологии SCA и анализа бинарного кода позволяют проверять унаследованные приложения и заказные разработки, в том числе использующие сторонние компоненты (СПО, готовые библиотеки).

6. Построение отчетов

Отчеты по уязвимостям и НДВ формируются автоматически, а их содержание выбирает пользователь. Результаты могут быть выгружены в соответствии с классификацией уязвимостей по версии БДУ ФСТЭК России, ОУД4, PCI DSS, OWASP Top 10, OWASP Mobile Top 10, HIPAA или CWE/SANS Top 25.



7. Сравнение результатов проверок

В рамках одного проекта можно сравнивать результаты проведенных тестирований для отслеживания динамики устранения или появления уязвимостей. При этом учитываются изменения, характерные для процесса написания кода.

8. Разграничение прав разработчиков

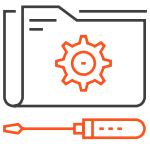
Для повышения уровня информационной безопасности можно разграничить права доступа разработчиков. Поддержка Microsoft Active Directory позволяет упростить управление правами доступа к Solar appScreener при большом числе разработчиков.

9. Подготовка рекомендаций для разработчиков

Разработчики заинтересованы сдавать проекты максимально быстро и с минимальными замечаниями. Поэтому рекомендации для разработчиков включают описания уязвимостей и НДВ, ссылки на содержащие их участки кода, а также конкретные советы по изменению кода.

10. Подготовка рекомендаций для офицеров безопасности

Офицерам безопасности необходима максимально полная информация о найденных уязвимостях и НДВ. Рекомендации для них содержат детальные описания уязвимостей и НДВ, включая способы их реализации, а также советы по настройке WAF от Imperva, ModSecurity или F5.



11. Интеграция в процесс разработки

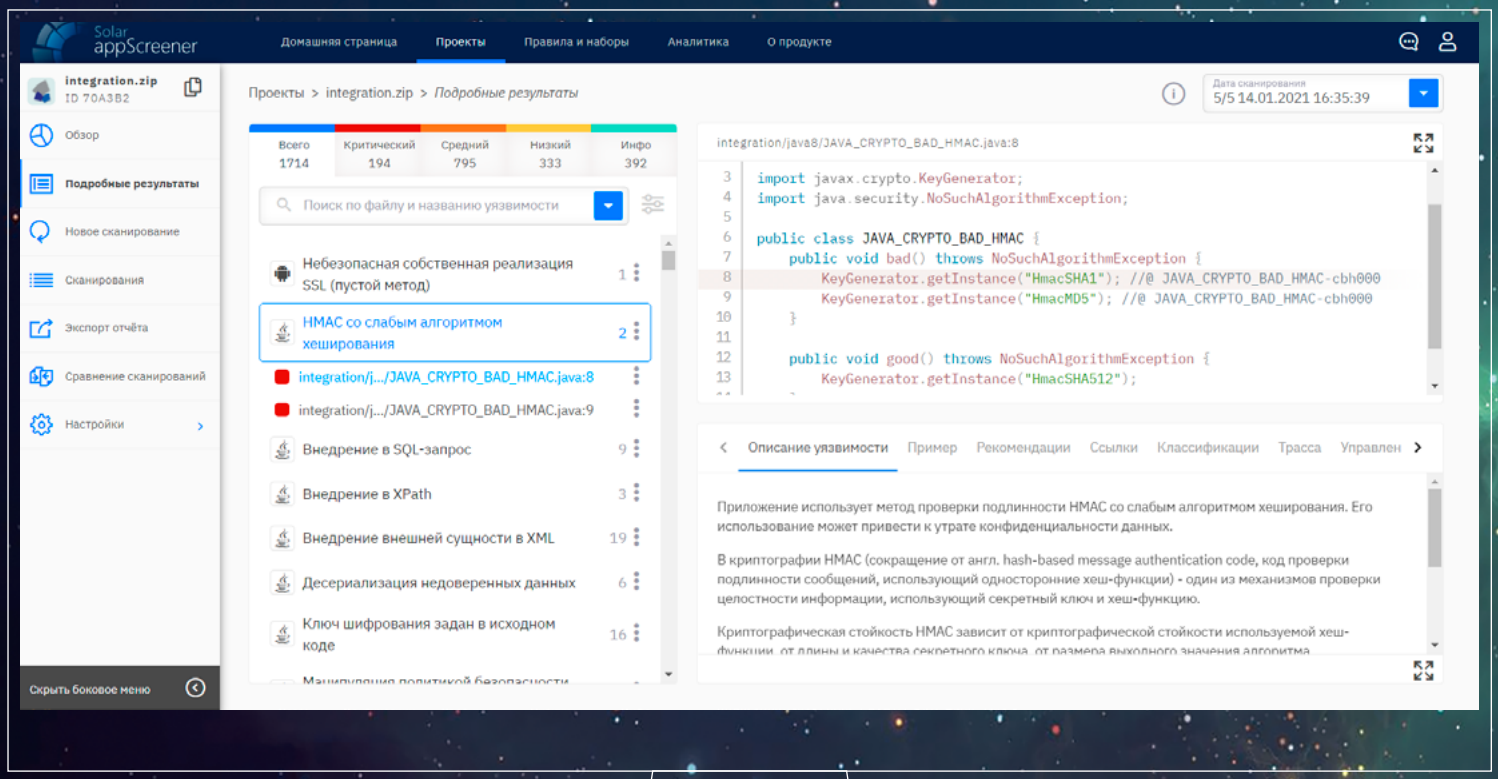
Solar appScreener можно связать с репозиториями Git и Subversion, VCS хостингами GitLab, GitHub, Bitbucket, интегрированными средами разработки Eclipse, Microsoft Visual Studio и IDEA, средствами сборки Xcode, Maven, Gradle, sbt, Visual Studio, CMake, Make, Autotools серверами CI/CD Jenkinks, Azure DevOps Server и TeamCity, а также платформой анализа качества кода SonarQube. Это позволяет встроить Solar appScreener в процесс разработки и реализовать полноценный Secure SDLC. С помощью открытого API доступна интеграция с другими системами и сервисами.

12. Работа с системами отслеживания ошибок

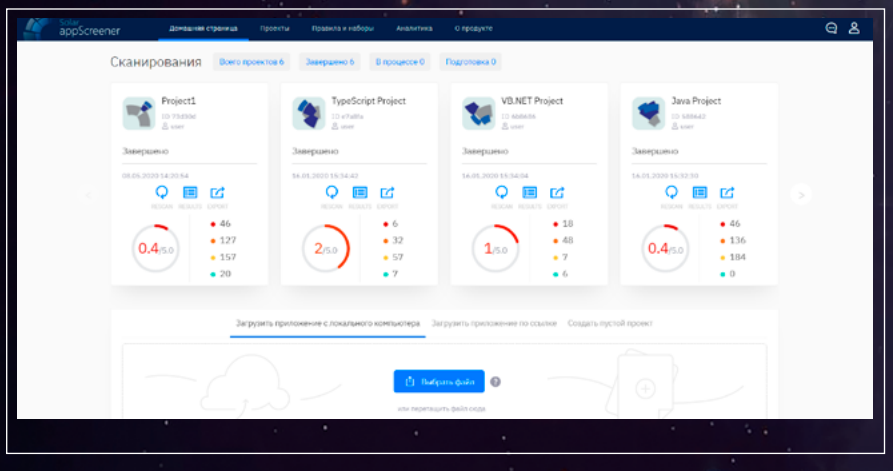
В базовую версию Solar appScreener входит интеграция с Atlassian Jira. Это позволяет заводить в Jira задачи по устранению найденных уязвимостей непосредственно из интерфейса Solar appScreener и отслеживать ход их выполнения. При необходимости можно реализовать поддержку любой другой системы отслеживания ошибок.

Solar appScreener

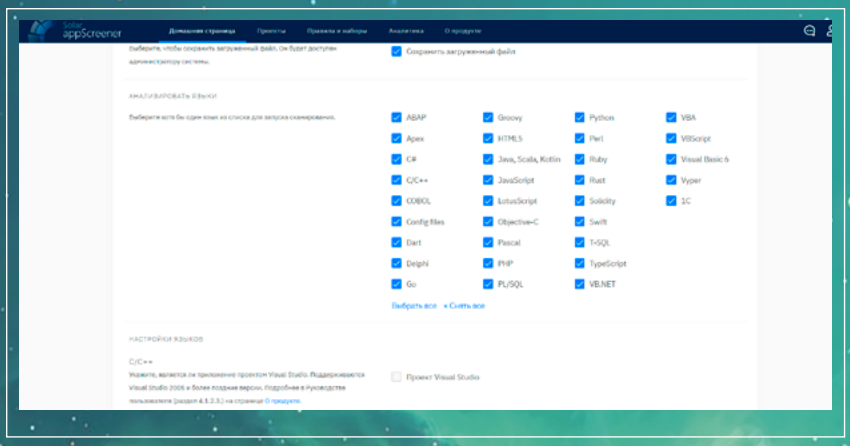
Solar appScreener в первую очередь рассчитан на службу ИБ, а не на разработчиков. В его основе — облегченная логика взаимодействия с пользователем, не требующая глубоких технических знаний для интерпретации результатов анализа. По этой причине интерфейс Solar appScreener отличается простотой и удобством, а сам процесс анализа максимально автоматизирован, что позволяет проверять код приложения в два клика.



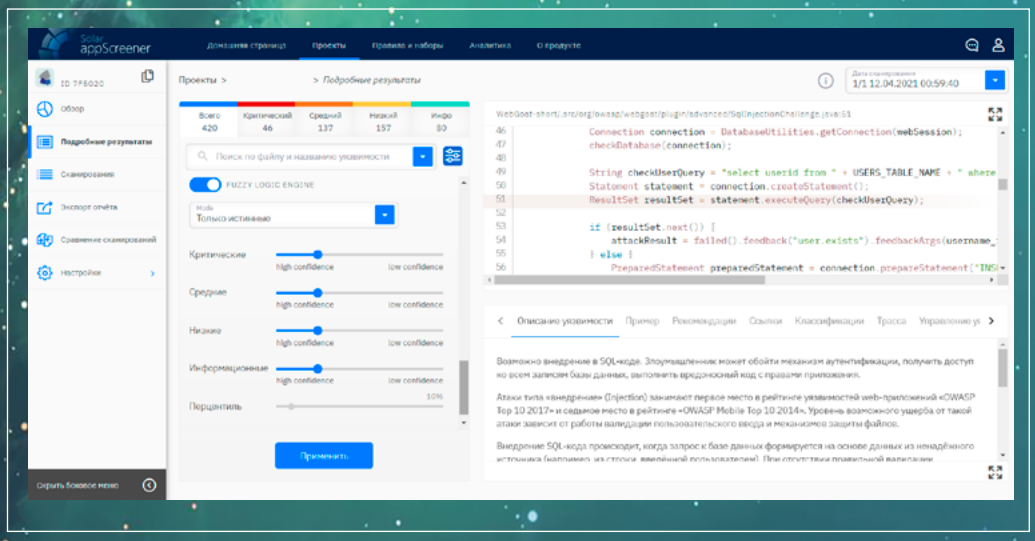
Описание найденных уязвимостей



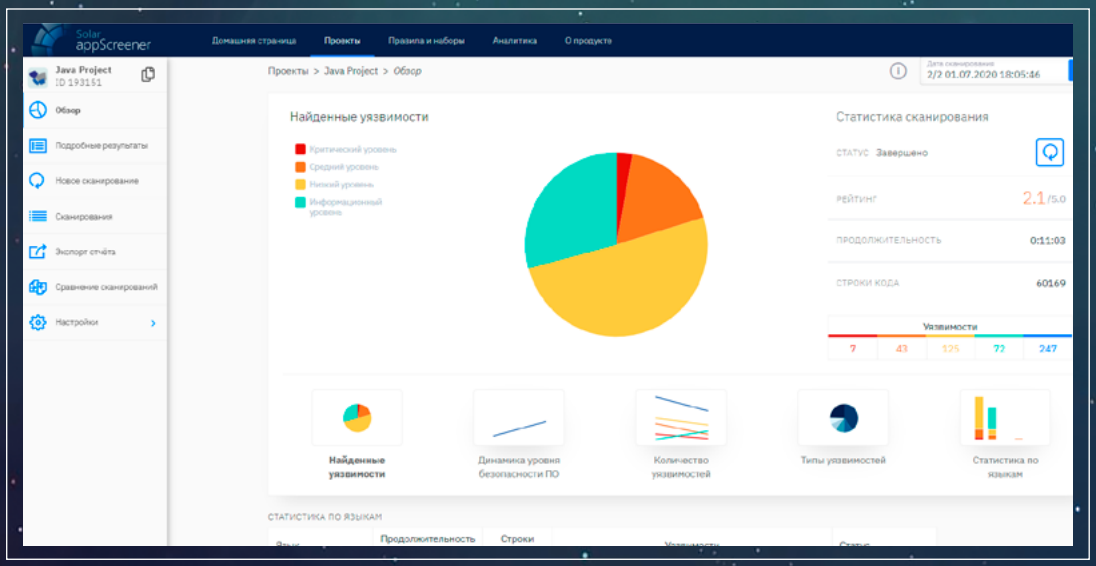
Главное окно интерфейса



Выбор параметров анализа



Работа с Fuzzy Logic Engine



Статистика проверки приложения

Этапы построения Центра контроля безопасности ПО

1 Разработка ролевой модели функционирования Центра с учетом организационно-штатной структуры организации

1. Сбор данных о процессах:
   - взаимодействия между подразделениями, отвечающими за разработку ПО и ИБ;
   - контроля и взаимодействия с внешними подрядчиками, отвечающими за разработку ПО;
   - выбора, внедрения и эксплуатации стороннего ПО, закупаемого организацией.
2. Формирование ролевой модели, включающей всех участников процесса разработки.
3. Определение зон ответственности участников.
4. Согласование модели и выдача рекомендаций по фиксации ответственности.

2 Развертывание программно-аппаратной инфраструктуры на базе Solar appScreener в формате внутрикорпоративного облака

1. Создание рабочей группы на стороне заказчика.
2. Подготовка информации и документации:
   - информация о кодовых базах ПО: объем, используемые средства сборки, библиотеки, технологии, конфигурации сборки;
   - информация о среде CI/CD и автоматизации разработки: описание процесса и технологий.
3. Проведение интервью и периодические встречи рабочей группы.
4. Подготовка вычислительных мощностей для Solar appScreener в соответствии с требованиями.
5. Подготовка доступа к инфраструктуре для специалистов «Ростелеком-Солар».
6. Подготовка среды с установкой Solar appScreener.
7. Согласование документов.
8. Обучение работе с Solar appScreener.

3 Адаптация методологии контроля ПО под текущие процессы и ИТ-ландшафт организации

1. Сбор и уточнение требований к безопасности используемого ПО.
2. Формализация, документирование и согласование требований к проверке исходного кода и исполняемых файлов.
3. Разработка и согласование регламента проверки исходного кода на уязвимости.
4. Формализация требований к автоматизации проверки исходного кода на уязвимости и НДВ в рамках процессов CI/CD заказчика.
5. Интеграция Solar appScreener, в том числе доработка скриптов автоматической сборки и интеграция с системами отслеживания ошибок, системой контроля версий и серверами CI/CD.

4 Создание компетенций Центра на базе сотрудников организации с привлечением экспертов «Ростелеком-Солар»

1. Формирование и согласование программы обучения и состава участников, включающей:
   - теоретическую и практическую части общепринятых практик и подходов к написанию безопасного ПО;
   - разбор самых опасных уязвимостей, которые встречаются в ПО;
   - обзор способов контроля и выявления уязвимостей и закладок.
2. Обучение участников процесса проверки исходного кода на уязвимости в соответствии с методологией, согласованной на предыдущем этапе.
3. Обучение работе с системой специалистов технической поддержки и администрирования.

5 Внедрение и запуск процессов непрерывного анализа ПО на уязвимости и закладки

1. Первичное сканирование разрабатываемого в организации ПО на уязвимости и НДВ в соответствии с согласованной ролевой моделью Центра и методологией.
2. Прохождение этапов сканирования, анализа, согласования, коррекции, повторного сканирования и утверждения результатов совместно со специалистами организации.
3. Практическое обучение сотрудников Центра разбору и контролю результатов в соответствии с принятой методологией и утвержденными критериями приемки.
4. Аутсорсинговая поддержка Центра: его независимый аудит и разбор отчетов.
5. Техническая поддержка и консультации по работе Solar appScreener.

Организации, использующие Solar appScreener

МТС Банк · S7 AIRLINES · МСП Банк · CARTREK innovative cars sharing platform · EXANTE

ВСЕРОССИЙСКИЙ БАНК РАЗВИТИЯ РЕГИОНОВ · Яндекс Деньги · ББР БАНК · ATABASCA · PARASOFT

nexagate · MSOFT · Ак Барс Банк · Compliance Control · Sense of Security

РЖД Российские железные дороги · GNP SEGUROS · ISBC RFID · KISA KOREA INTERNET & SECURITY AGENCY

Преимущества Центра контроля безопасности ПО

Статический анализ бинарного кода

Уникальные технологии декомпиляции и деобфускации кода исполняемых файлов позволяют проверять приложения даже при отсутствии исходных кодов, например унаследованные приложения или заказные разработки, в том числе для Google Android и Apple iOS.

30+ языков программирования

Большое число поддерживаемых языков позволяет анализировать почти все приложения, в том числе созданные на 1С, ABAP (для SAP), COBOL или Solidity (для смарт-контрактов). Язык приложения определяется автоматически. Возможен анализ приложений, написанных на нескольких языках.

10+ методов анализа кода

Для анализа приложений в Solar appScreener совместно используется более 10 методов анализа, в том числе синтаксический и taint-анализ, что позволяет максимизировать выявление уязвимостей и НДВ в коде приложения.

Не требует опыта разработки

Solar appScreener в первую очередь рассчитан на службу ИБ, а не на разработчиков — именно поэтому интерфейс отличается простотой и удобством, а сам анализ максимально автоматизирован. В результате с анализатором может работать офицер безопасности, не имеющий опыта разработки ПО.

Широкий охват и высокая скорость

Статический анализ кода приложений охватывает наибольшее число возможных уязвимостей и НДВ, а также отличается высокой скоростью работы. Для завершения анализа не нужно ждать часы и дни — на обычное приложение достаточно 30 минут.

Низкий процент ложных срабатываний

Для минимизации количества ложных срабатываний и пропущенных уязвимостей и НДВ в коде в Solar appScreener используется технология Fuzzy Logic Engine, которая задействует математический аппарат нечеткой логики и является технологическим ноу-хау компании «Ростелеком-Солар».

Сложные правила поиска

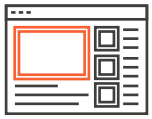
В подготовке правил поиска уязвимостей и НДВ для Solar appScreener участвуют высококвалифицированные специалисты «Ростелеком-Солар», что гарантирует их высокое качество и актуальность. Базы уязвимостей и НДВ можно обновлять как вручную, так и в автоматическом режиме.

Подробные рекомендации

Результаты анализа кода приложения предоставляются разработчикам и службе ИБ в формате конкретных рекомендаций по устранению уязвимостей и НДВ. Так, рекомендации по настройке WAF позволяют блокировать уязвимости и НДВ на время исправления кода приложения.

Быстрый запуск сканирования

Проверка запускается в несколько кликов и не требует долгой предварительной настройки. Для анализа приложений для Android и Apple iOS достаточно указать ссылку на них в магазинах приложений Google Play и App Store.



Современный графический интерфейс

Удобный и современный интерфейс Solar appScreener обеспечивает быструю работу и наглядный результат анализа уязвимостей и НДВ. Сравнение результатов тестирования и работа с интегрированными системами доступны непосредственно из интерфейса.

Отечественное ПО

Solar appScreener разработан в России и внесен в Единый реестр отечественного ПО (№ 6119), что позволяет использовать его для импортозамещения зарубежных аналогов. Над документацией и интерфейсом продукта работали русскоязычные специалисты. Цены номинированы в рублях и не зависят от курсов валют.

Сертификат ФСТЭК России

Solar appScreener сертифицирован ФСТЭК России как программное средство контроля защищенности и соответствует требованиям руководящего документа «Защита от несанкционированного доступа к информации. Часть 1» (Гостехкомиссия России, 1999) по 4-му уровню контроля отсутствия НДВ (сертификат соответствия № 4007).

Ростелеком
Солар

presale@rt-solar.ru rt.ru rt-solar.ru

+7 (499) 755-07-70