SOLAR

SOLAR APPSCREENER

ОХОТА НА УЯЗВИМОСТИ

Open source под микроскопом

W: RT-SOLAR.RU

T: 8 (800) 302-85-23

Риски для сторонних разработчиков

Когда работаешь с крупными корпорациями, безопасность становится абсолютным приоритетом.

Каждая незамеченная уязвимость — это потенциальная катастрофа для бизнеса клиента, и это не гипербола.

Когда с этим сталкиваются крупные организации, масштаб стоящих перед ними задач кратно увеличивается: соблюдение строгих требований и нормативов, защита от внешних угроз.

Все это ставит перед разработчиками не только задачу быть суперпрокачанными в коде, но и понимать, какие риски могут поджидать за каждым углом.

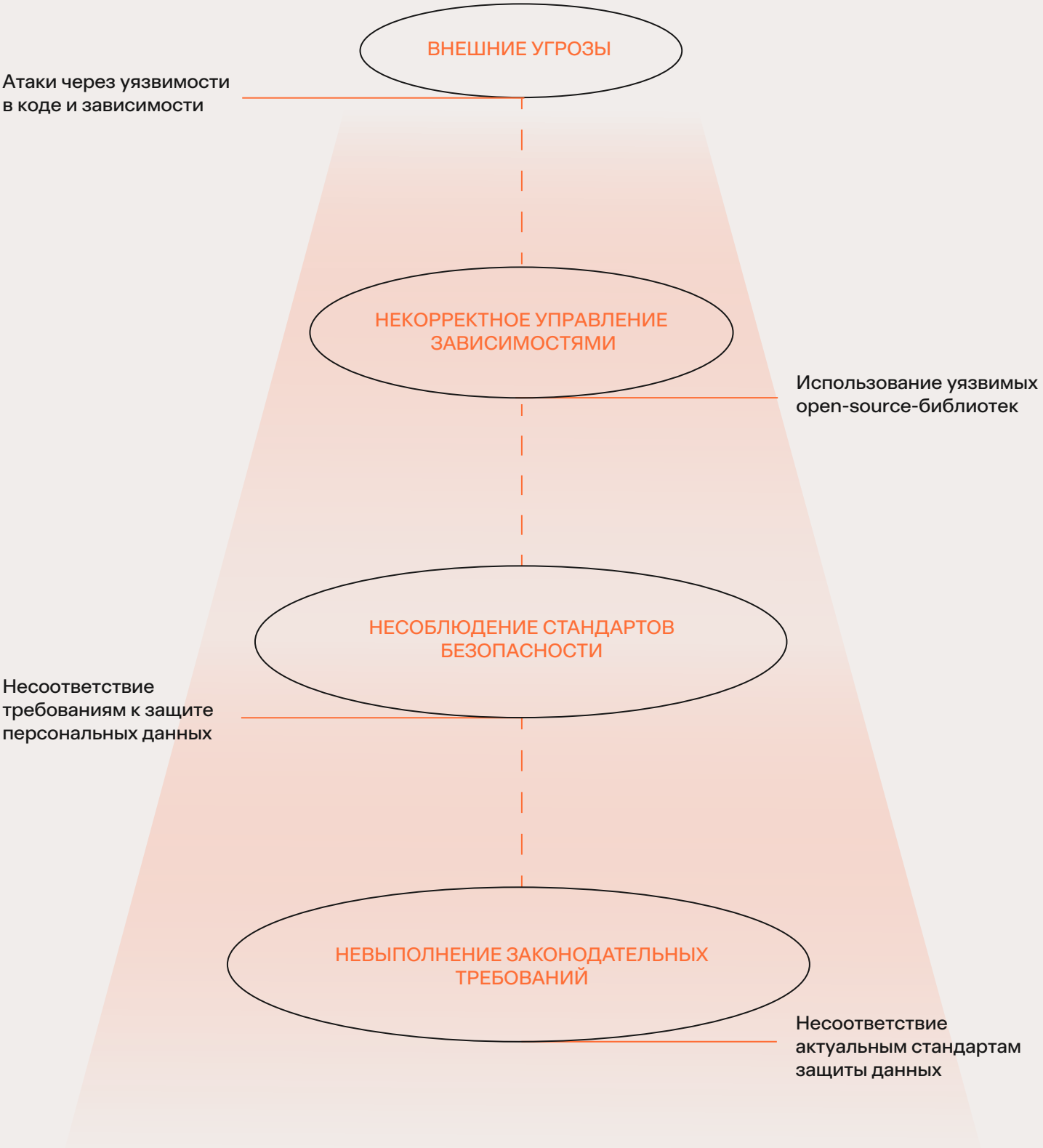
БЕЗОПАСНОСТЬ КАК СТРАТЕГИЧЕСКОЕ МЫШЛЕНИЕ

Безопасность требует умения держать в голове как технические детали, так и стратегические вопросы, потому что от этого зависит не только репутация, но и успешность проекта.

БЕЗОПАСНОСТЬ

> Это не просто пункт в списке дел разработчика, а основополагающий принцип, без соблюдения которого ваше приложение обречено на провал

Что угрожает безопасности корпоративных приложений?



Юридические риски

Небрежность в проверке на безопасность может стать причиной утечек данных, что влечёт за собой правовые последствия для компании.

Несоблюдение стандартов безопасности данных, таких как ФЗ-152 «О персональных данных», может привести к штрафам и судебным искам.

53%

На серверной части мобильных приложений чаще всего уязвимости связаны с утечкой отладочной и конфигурационной информации

40%

Некорректная реализация контроля доступа

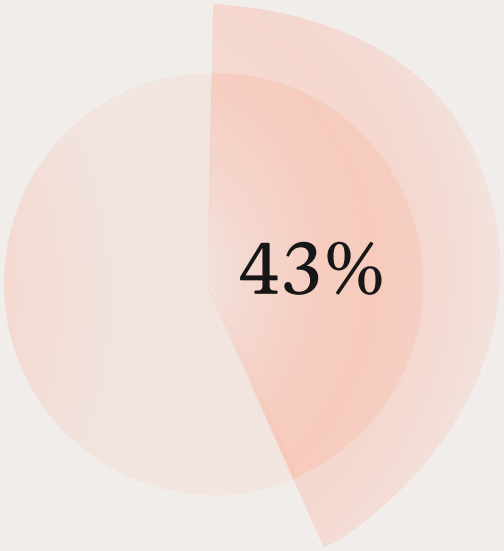
40%

Основная проблема клиентской части в 2024 году — отсутствие обфускации кода мобильного приложения

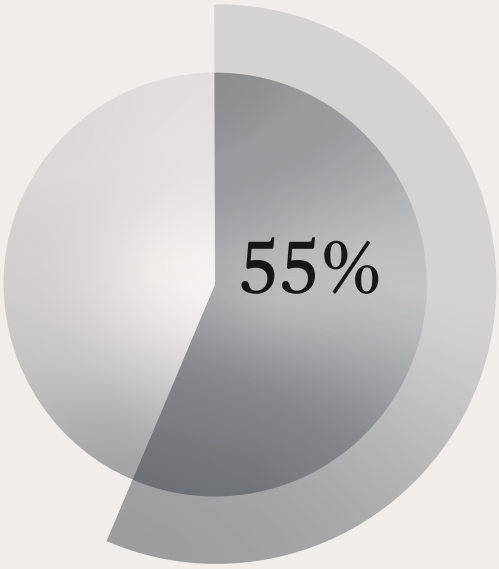
Open-source в России: когда твоя зависимость — это уязвимость

Если вы работаете в стартапе или маленькой команде, то знаете, как это бывает: сроки поджимают, фича горит, а на GitHub уже все есть. Один pip install, и пошло-поехало. Но за этим удобством может скрываться «минное поле» — особенно в России, где киберугрозы растут день ото дня.

В последние годы участились кибератаки, использующие уязвимости в open-source.



Хакерских проникновений произошло через уязвимости в корпоративных приложениях



Атак на малый бизнес в России связаны с недостаточной безопасностью цепочек поставок и зависимостей кода

* Данные Центра исследования киберугроз Solar 4RAYS за первое полугодие 2024 года

Когда библиотека — это бэкдор

Собрали три ярких (и пугающих) кейса, чтобы стало понятно, почему обновления и аудит зависимостей — это не бюрократия, а самосохранение.

В последние годы наблюдается рост числа инцидентов, связанных с использованием уязвимых open-source-библиотек, что приводит к серьезным последствиям для безопасности приложений. На следующей странице приведены пять наиболее заметных случаев.

Log4j: логи, которые открывают двери

В декабре 2021 случился инфобез-блокбастер: уязвимость в Log4j (CVE-2021-44228) позволяла удаленно исполнять код просто через специально оформленный HTTP-запрос. Без авторизации, без магии.

Log4j (CVE-2021-44228) — критическая уязвимость в популярной Java-библиотеке логирования, позволявшая удаленное выполнение кода и угрожавшая множеству приложений.

КАК ЗЛОУМЫШЛЕННИКИ МОГЛИ ВОСПОЛЬЗОВАТЬСЯ ЭТОЙ УЯЗВИМОСТЬЮ?

Они могли отправить специальную строку через HTTP-запросы, которая бы инициировала выполнение произвольного кода на сервере. Это могло привести к удаленному исполнению кода и полной компрометации системы. Важно отметить, что для эксплуатации этой уязвимости не требовалось специальных привилегий, а сама уязвимость касалась всех версий Log4j от 2.0 до 2.14.1.

ЧТО ДЕЛАТЬ?

Для защиты от этой угрозы необходимо было сразу же обновить библиотеку до версии 2.15.0 или более поздней. В случае, если обновление невозможно, следовало применить временные меры, такие как отключение функций, связанных с логированием через JNDI, или блокировка внешних DNS-запросов

XZ-utils: бэкдор под видом компрессии

Весной 2022 уязвимость в Xz-utils чуть не стала катастрофой для всей Linux-инфраструктуры. Злоумышленник добавил бэкдор в популярную утилиту, и он попал в тестовые сборки Fedora, Kali и Debian. Через него можно было получить удаленный доступ по SSH.

В 2022 году злоумышленники смогли получить контроль над репозиторием, в который были добавлены две версии утилит (5.6.0 и 5.6.1) со встроенным бэкдором.

Бэкдор находился в коде, который отвечает за компрессию и декомпрессию данных по алгоритму lzma. Этот код используется множеством программ и утилит в составе дистрибутивов Linux, к нему также обращается сервер OpenSSH.

Неавторизованный пользователь через SSH активирует бэкдор специальным ключом и получает полный доступ к системе.

БЭКДОР ПЫТАЛИСЬ ВНЕДРИТЬ «ПОД ПРИКРЫТИЕМ»

Необходимый для его работы код присутствовал только в архивах исходников, которые используют мейнтейнеры дистрибутивов. Атака практически удалась: вредоносный код попал в бета-версию дистрибутивов Fedora 40, Kali Linux, присутствовал в тестовой ветке Debian. По оценке экспертов, в зоне риска оказались до 2 миллионов хостов.

ЧТО ДЕЛАТЬ?

Распространение бэкдора смог остановить сотрудник Microsoft, который проводил тестирование производительности и выявил длительное подвисание процесса liblzma, к которому обращался демон sshd

Heartbleed: старый друг, старая беда

Еще один важный пример касается уязвимости в OpenSSL — широко используемой криптографической библиотеке, которая обеспечивает безопасное подключение в интернете (например, через протоколы HTTPS).

В версии OpenSSL 1.0.1 была обнаружена уязвимость, получившая название Heartbleed. Эта уязвимость позволяла атакующему извлечь из памяти серверов секретные ключи и другие конфиденциальные данные, что могло привести к краже личной информации и данных, передаваемых по защищенному каналу.

НАРУШЕНИЕ БЕЗОПАСНОСТИ

В случае с Heartbleed безопасность была нарушена из-за ошибки в реализации протокола SSL/TLS. Атака происходила на уровне буферов данных, где злоумышленники могли запросить возврат произвольного объема памяти и получить данные, которые могли быть в ней размещены.

ЧТО ДЕЛАТЬ?

Для предотвращения подобных инцидентов необходимо следить за актуальностью версий OpenSSL, регулярно обновлять библиотеки и внимательно относиться к выбору компонентов, которые обеспечивают критическую безопаность

ML-библиотеки — опасность с привкусом машинного обучения

Безопасность — не только про web и backend. Даже библиотеки для ML могут быть уязвимыми. Пример — CVE-2021-41228 в TensorFlow, позволяющая злоумышленникам выполнить произвольный код через утилиту saved_model_cli при обработке данных через параметр --input_examples.

ПРОБЛЕМА

Проблема возникала из-за использования небезопасной функции eval для обработки внешних данных, что позволяло атакующему внедрить вредоносный код, который выполнялся на машине пользователя. Уязвимость была исправлена в версиях TensorFlow 2.7.0, 2.6.1, 2.5.2 и 2.4.4, где eval был заменен на более безопасные функции обработки входных данных.

Использование опенсорсных решений для разработки и развертывания ML-моделей, таких как TensorFlow, предоставляет много плюсов, но важно помнить, что безопасность всегда должна быть в приоритете.

ЧТО ДЕЛАТЬ?

Для этого используются SCA-инструменты

Как обезопасить себя и своего заказчика? Подход для небольших команд

НЕ ПАНИКУЙТЕ, НО И НЕ БУДЬТЕ БЕСПЕЧНЫМИ. ЧТОБЫ НЕ СТАТЬ ГЕРОЕМ СЛЕДУЮЩЕГО КЕЙСА

Когда безопасность становится критически важной, но ваши ресурсы ограниченны, важно находить решения, которые не требуют значительных инвестиций или сложной настройки.

Современные инструменты позволяют легко интегрировать защиту кода в процесс разработки без излишних затрат.

ПРИМЕРОМ ТАКОГО ПОДХОДА ЯВЛЯЕТСЯ

Модуль для автоматического анализа сторонних компонентов в коде и лицензионных рисков (SCA и SCS), который позволяет малым и средним компаниям, а также сторонним разработчикам эффективно справляться с рисками безопасности, связанными с использованием сторонних библиотек и зависимостей.

Этот инструмент является простым, но мощным средством обнаружения уязвимостей в коде и библиотеках, что помогает снизить вероятность атак и утечек данных.

Подключите SCA- и SCS-инструменты

Не затягивайте с обновлениями

Проводите аудит библиотеки перед использованием

Обучайте команду основам безопасной разработки

Анализ SCA и SCS для небольших команд разработчиков

ПРОСТОТА В ИСПОЛЬЗОВАНИИ

Облачное решение интегрируется в вашу разработку без сложных настроек, что позволяет вам быстро начать проверку безопасности, не отвлекаясь от основной работы

КОМПЛЕКСНЫЙ АНАЛИЗ БЕЗОПАСНОСТИ

Этот модуль анализирует не только open-source-библиотеки, но и все зависимости вашего приложения, помогая выявить уязвимости и угрозы, которые могут быть использованы хакерами

АНАЛИЗ СОСТАВА ПО (SCA)

Позволяет проверить безопасность сторонних библиотек, используемых разработчиками в своем ПО

ЭКОНОМИЯ НА DEVSECOPS

Снижает затраты на защиту приложений за счет оптимизации процессов и автоматизации анализа, что сокращает время устранения уязвимостей и расходы на безопасность

АНАЛИЗ ЦЕПОЧКИ ПОСТАВОК ПО (SCS)

Отслеживает безопасность ПО на всех этапах, оценивая доверие к внешним компонентам по 8 метрикам (репутация, активность, безопасность и др.).

Как это работает?

/1

ИНТЕГРАЦИЯ В ПРОЦЕСС РАЗРАБОТКИ

Модуль для анализа SCA и SCS легко подключается к вашей CI/CD-системе и запускается на каждом этапе разработки, проверяя код и библиотеки на наличие уязвимостей.

/2

АВТОМАТИЧЕСКИЙ АНАЛИЗ КОДА И ЗАВИСИМОСТЕЙ

Каждое обновление или изменение в коде анализируется, выявляя все потенциальные угрозы, связанные с используемыми библиотеками.

/3

ОТЧЕТЫ С РЕКОМЕНДАЦИЯМИ

Получаете подробные отчеты, которые не только информируют о проблемах безопасности, но и дают четкие рекомендации по их устранению.

ЗАЩИЩЕННЫЙ КОД — ЭТО КОНКУРЕНТНОЕ ПРЕИМУЩЕСТВО!

Узнать подробнее

SOLAR

W: RT-SOLAR.RU
W: RT.RU

T +7 (499) 755-07-70
E solar@rt-solar.ru

Центральный офис, 125009,
Москва, Никитский переулок, 7с1