Коварный мир open-source глазами разработчика и пользователя

Михаил Маркевич

Консультант по безопасности

Москва, 25 августа 2022 года

О чем этот доклад

О том, как выбирать open-source решения, чтобы впоследствии не пожалеть об этом

Для кого этот доклад

- Для dev/sec/ops, которым нужно выбрать технологию или библиотеку с открытым кодом
- Для dev/sec/ops, за которых этот выбор уже сделали
- Для тимлидов, архитекторов, безопасников и просто желающих узнать, что может пойти не так

Обо мне




LEAD AUDITOR
ISO
27001

CISA

CISM

CISSP

irm


DIGITAL
SECURITY

илим

ЛЕНТА

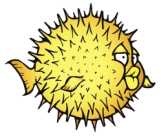
GGA

ЮЛМАРТ

Opera

dhis2





NetBSD



libvirt
powered

openstack



aws




Apache



MariaDB




MAIL

p

SURICATA



php

C++

$_

Терминология

- Разработчик - один из соавторов программы с открытым кодом
- Пользователь - любой из нас, кто использовал, внедрял, разрабатывал что-либо с применением открытого ПО

OFF ONE 2022

Часть 1. Пользователь

Обычный случай из практики

Сервер web-приложений с Tomcat, который очень хочется опубликовать на порту 80 или 443...

modern installation of Tomcat with SSL on port 443

Asked 3 years, 6 months ago Modified 3 years, 5 months ago Viewed 5k times

-1

In the computer course I'm writing I'm using Tomcat for the server. (Students learn how to set up CentOS and everything from scratch. Currently the course has them using Tomcat running on port 8080.) I'm going back to write the section on security. I want students to learn to set up their web server to use SSL/TLS on port 443, with HTTP port 80 redirecting to HTTPS port 443. This should be a very basic, fundamental configuration, no?

Исходный URL

Варианты решения...

... разной степени годности:

- Перевесить Tomcat на порт 443 и запустить as root (!?)
- Перевесить Tomcat на порт 443 и изменить capabilities
- Перенаправить трафик на порт при помощи МСЭ
- Настроить обратный прокси

В итоге...

- Выбор решения зависит от кругозора и опыта инженера
- Стандартное (наименее кастомизированное) решение представляется максимально надежным
- Умение выбирать такие решения в рамках заданных ограничений сродни искусству

Где можно наступить на грабли...

... или удачно их использовать:

- Нетривиальный синтаксис конфигурационных файлов
- Вольное обращение с системными привилегиями
- Нарушение границ доверия (trust boundaries)

Наивный пользователь

- Забывает, что уровень защищенности системы снижается в динамике
- Считает, что функциональность системы не ухудшается при апгрейде (привет, .htaccess)
- Часто изменяет конфигурацию системы таким образом, что при апгрейде появляются новые уязвимости

Параноидальный пользователь

- Старается следовать стандарту и вносить минимальное количество изменений в конфигурацию
- Пытается оценить потенциальные изменения в динамике
- Автоматизирует то, что не успели сделать разработчики и что легко упустить из виду

Признаки плохого ПО

- Отсутствие обновлений или регулярной доработки
- Невнятная документация
- Трудночитаемые файлы настройки и отсутствие инструментов проверки конфигурации
- Значительные изменения функционала ПО при переходе с одной версии на другую

Что делать тимлиду или архитектору?

- Поощрять использование типовых решений
- Расширять кругозор своей команды

Что делать безопаснику?

- Любая нестандартная, сложная, нетривиальная конфигурация повышает вероятность появления уязвимости

- Все трудности, с которыми сталкиваются инженеры - потенциальные уязвимости и готовый чеклист для пентеста или аудита ИБ

Часть 2. Разработчик

Разработчик open-source...

... очень часто работает вслепую:

- Нет обратной связи от пользователей
- Нет инструментов мониторинга ошибок
- План разработки ПО, особенно в небольших командах, отсутствует

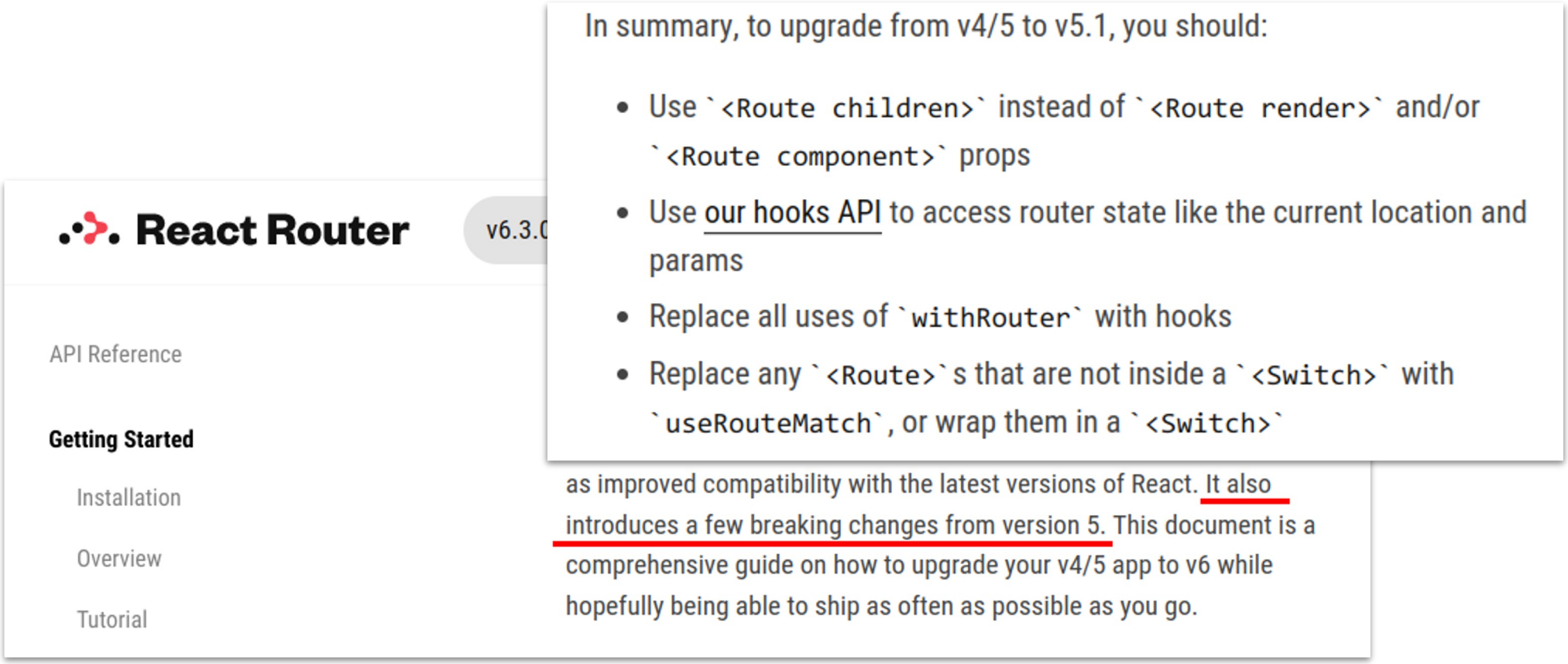
В мире open-source существует проблема

... излишнего доверия:

- Функционал библиотек считается по умолчанию неизменным
- Атаки на supply chain все более и более популярны
- При видимой открытости ПО и средств разработки нет инструментов контроля изменений

А что с обновлениями?

Отличный пример от разработчиков React Router:



Исходный URL

Процедура раскрытия уязвимостей

... может иметь различные нюансы:

- Типовые сроки раскрытия могут быть слишком коротки для некоторых категорий пользователей
- Парадоксальным образом security by obscurity может иметь право на существование наряду с coordinated vulnerability disclosure

Лицензионные риски

(относится не только к самим разработчикам ПО, но и разработчикам-пользователям):

- Не каждый open-source одинаково open
- Существуют лицензии, ограничивающие использование кода в производных приложениях или лицензии с неопределенным юридическим статусом
- Примеры: GPL vs Apache vs Public Domain

Признаки безопасного open-source ПО

Как минимум:

- Secure by default (привет, OpenBSD)
- Документированная модель угроз или архитектура безопасности приложения
- Политика обработки и раскрытия уязвимостей
- Активное коммьюнити

Заключение

Основные наблюдения

- Стандартный путь в open source - наиболее безопасный (а также документированный и протестированный)
- Без знания контекста задачи и широкого технического кругозора очень легко ошибиться с выбором
- При возрастании сложности стека технологий самообразование и обучение имеют ключевую роль

Что делать?

- “Учиться, учиться и учиться” (с)
- Исследовать технологический стек и связанные с ним базовые компоненты, а не только то, с чем непосредственно работаем
- Использовать чек-листы при выборе open-source технологий
- Делиться опытом (в том числе и неудачным)

ON
FF
ONE
2022

Спасибо за внимание!

Web 2.0

Web3


ON
FF
ONE
2022