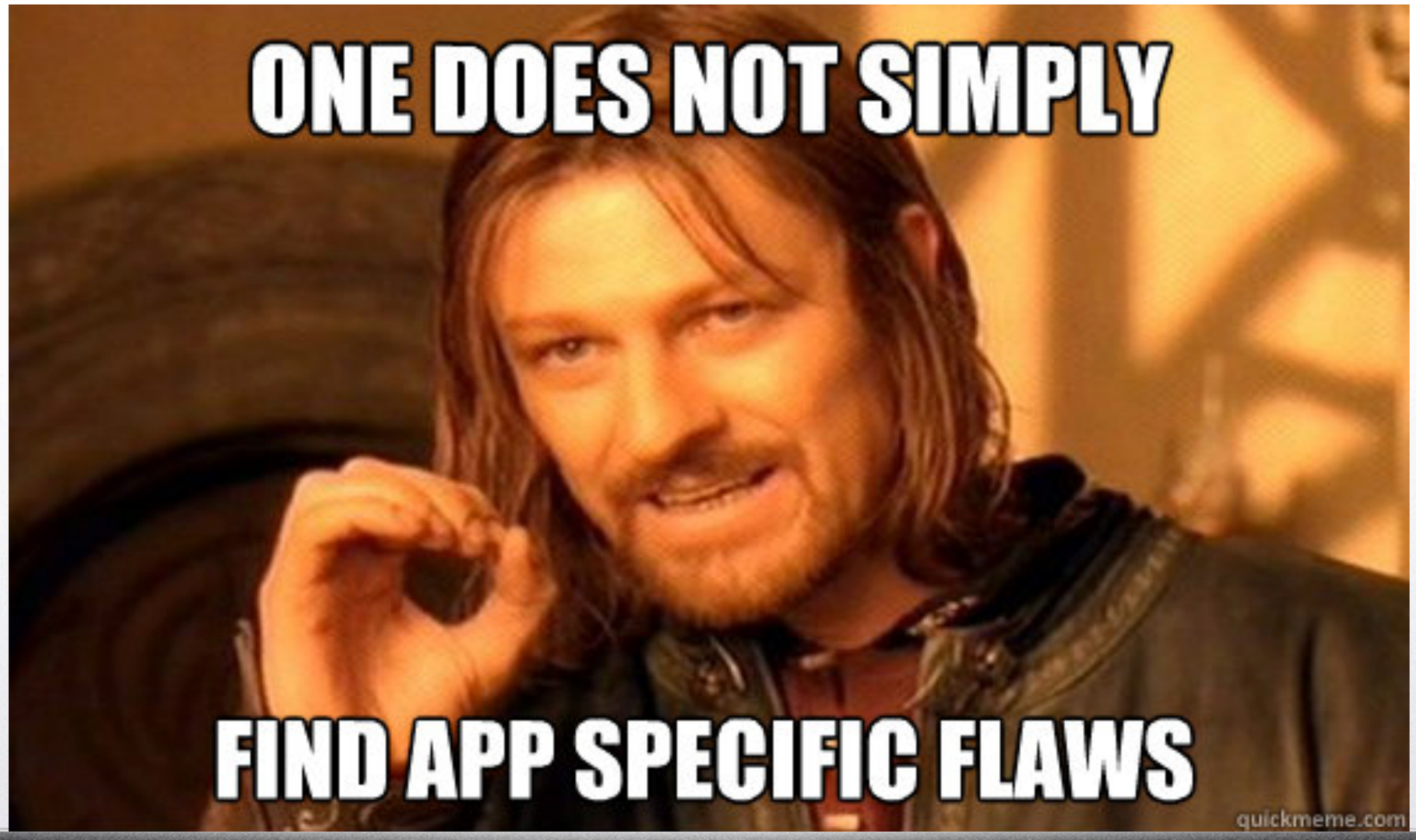
Обнаружение уязвимостей логики приложений методом статического анализа
Где правда, где реклама?

Петухов Андрей

Internal Security –
the foundation for your IT services

РусКрипто 2013

Too Long, Didn't Read



Преамбула

Welcome to the Real World, Neo

- Глобальные задачи в мире ПО и стат. анализ
  - безопасная разработка ПО
  - проверка соответствия ПО заданным требованиям
- Задачи могут решаться владельцем ПО либо внешним подрядчиком (in-house vs outsource)
- Итого три варианта
  - разработка у себя, проверка вне/разработка вне, проверка у себя/разработка вне, проверка вне
- Методически, применение стат. анализа в SDLC - решенная заздача
- Проверка у себя vs проверка спец. конторой - разница в толерантности к ложным срабатываниям

Стат. анализ и типичные недостатки

- Обычно связаны с некорректной обратботкой входных данных (англ. Input validation)
  - ➡ возможность внедрения операторов SQL, возможность переполнения буфера и т.п.
- Есть модели для описания таких недостатков
  - ➡ Non-interference, которая позже получила реализацию в виде т.н. подхода taint-analysis
  - ➡ общий шаблон недостатка: использование данных, контролируемых пользователем, в критичных вызовах без предварительной проверки на корректность
- Понятно как искать, все умеют, борьба идет за снижение кол-ва ложных срабатываний и повышение полноты (нестандартные фреймворки)

Стат. анализ и специфичные недостатки

- Для описания недостатка надо использовать термины из предметной области
  - ➡ возможность манипуляции курсами обмена валют для получения выгоды
  - ➡ возможность просмотра чужой истории платежей в системе ДБО
- Классы недостатков (по-английски)
  - ➡ insufficient access control, insufficient process validation, insufficient anti-automation
- Для того, чтобы что-то искать, надо сформулировать, что искать; как сделать детектирование app specific недостатков “коробочно” (методически, технически)?
- Еще можно еще вспомнить задачу проверки корректности протоколов и их свойств, но это out of scope

Амбула

Доступные исходные данные

- Исходный код (текст)
- Лексемы
- Дерево синтаксического разбора (AST)
- Граф потоков управления (CFG)
- Графы зависимостей (DFG/PDG/SDG)
- Множества состояний программы (см. абстрактная интерпретация)
- Множества возможных значений переменных в точках программы

➡ возможные типы переменных, возможные значения строк (string analysis)

Задача

- Дано ПО в любом из перечисленных представлений
- Спецификации ПО нет
- Формальных требований к ПО нет
- Требуется найти app specific недостатки
- Речь идет только об уязвимостях реализации, а не конфигурации или эксплуатации

Что можно сделать с представлением?

- Можно поискать наличие в представлении фрагмента, соответствующего заданному шаблону
  - ➡ “сигнатурный” метод
  - ➡ варианты: grep, регулярные выражения над потоком лексем, регулярные выражения над AST, поиск подграфа в графах зависимостей (taint-анализ - тоже сигнатурный!!)
  - ➡ так делают все известные мне статические анализаторы
  - ➡ откуда взять сигнатуру специфичного недостатка? получается, о его наличии надо знать заранее, чтобы написать такую сигнатуру? как насчет переносимости на другие приложения?

Что можно сделать с представлением?

- Построить гипотезу спецификаций, неявно предполагаемых разработчиками, и проверить соответствие ПО им
  - ➡ англ. extraction of intended behavior или specification inference
  - ➡ реально ***научная задача***, если вы понимаете, о чем я :)
  - ➡ “Toward automated detection of logic vulnerabilities in web applications”
  - ➡ “Static detection of logic flaws in service-oriented applications”
  - ➡ “Static detection of access control vulnerabilities in web applications”
  - ➡ так не делает ни один из известных мне статических анализаторов

P.S.

Как делаем мы?

- Ручной поиск экземпляров, формирование сигнатур, сигнатурный анализ для обеспечения полноты
- Используем grep, Appercut и свои собственные инструменты
- См. задачу проверки соответствия ПО требованиям, которую решает внешний подрядчик (4-ый слайд)

In-house checks?

- Как искать app specific недостатки in-house, учитывая требования к низкому числу ложных срабатываний и отсутствия своей экспертизы для написания сигнатур?
- Увы.

Вопросы

- Контакты
  - Twitter: @p3tand
  - Mob: +7 916 360-52-49
  - Email: andrew.petukhov@solidlab.ru
  - Blog: http://andrepetukhov.wordpress.com/