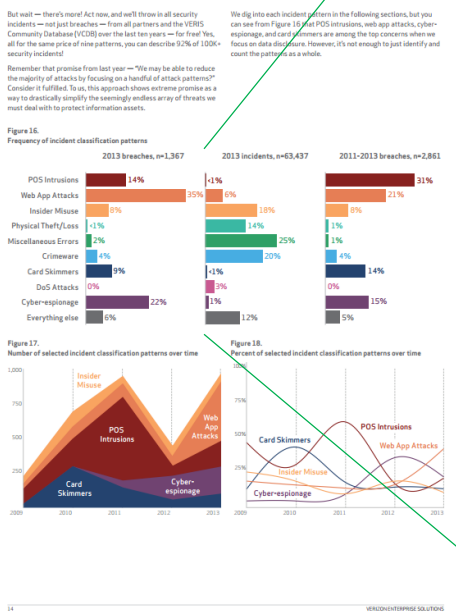
Анализ уязвимостей

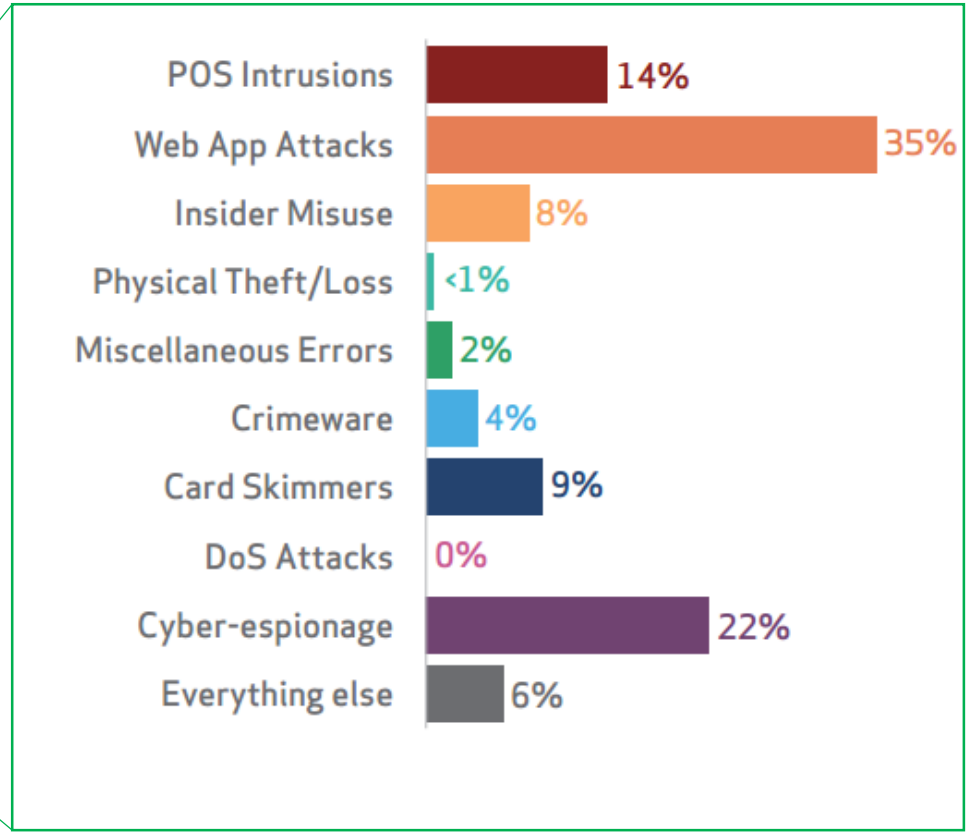
POSITIVE TECHNOLOGIES

ptsecurity.com

Международная статистика (Verizon)

Начиная с 2014 года постоянный рост числа атак на веб-приложения (35%)

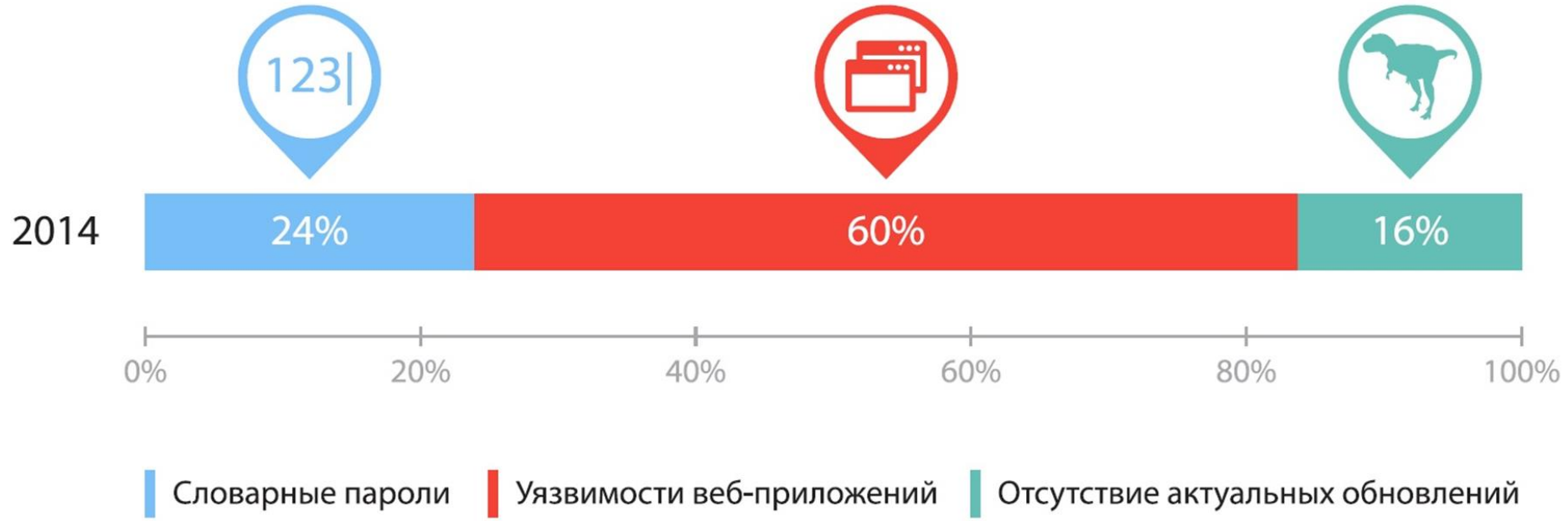




Verizon 2014 Data Breach Investigations Report

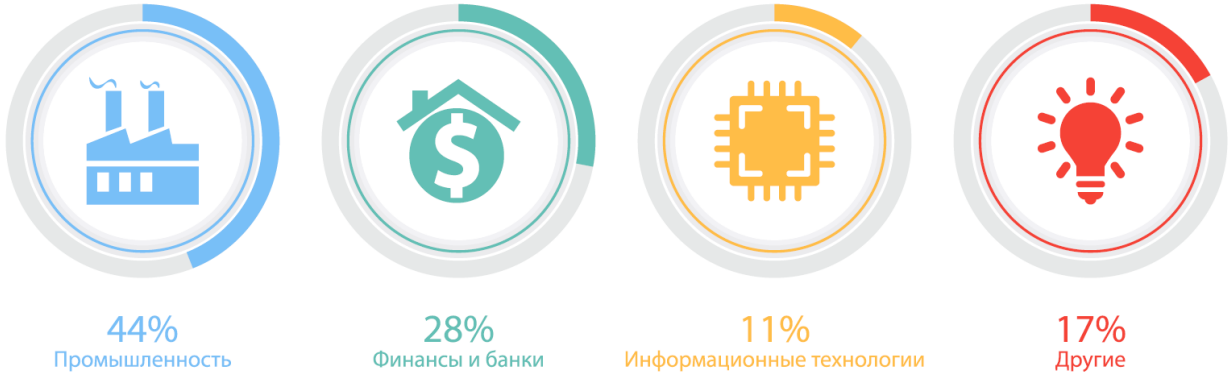
Уязвимости приложений – основная угроза сетевого периметра

Векторы атак для преодоления сетевого периметра

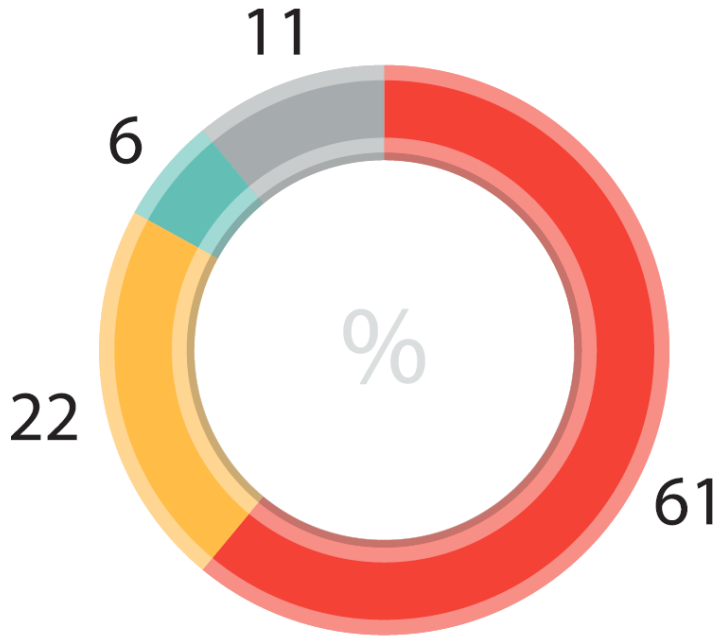


Российская статистика (Positive Technologies)

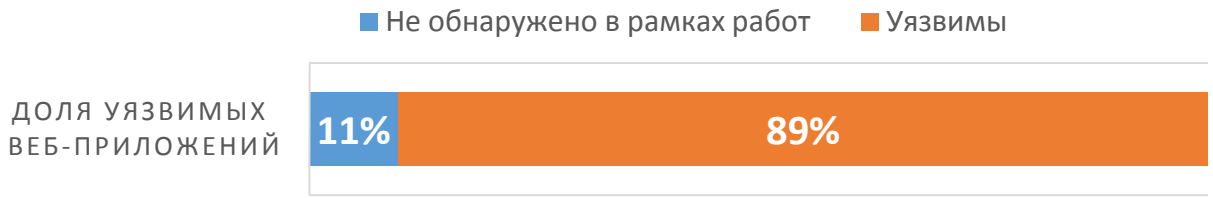
В проведенном исследовании среди различных отраслей



61% уязвимостей высокой критичности

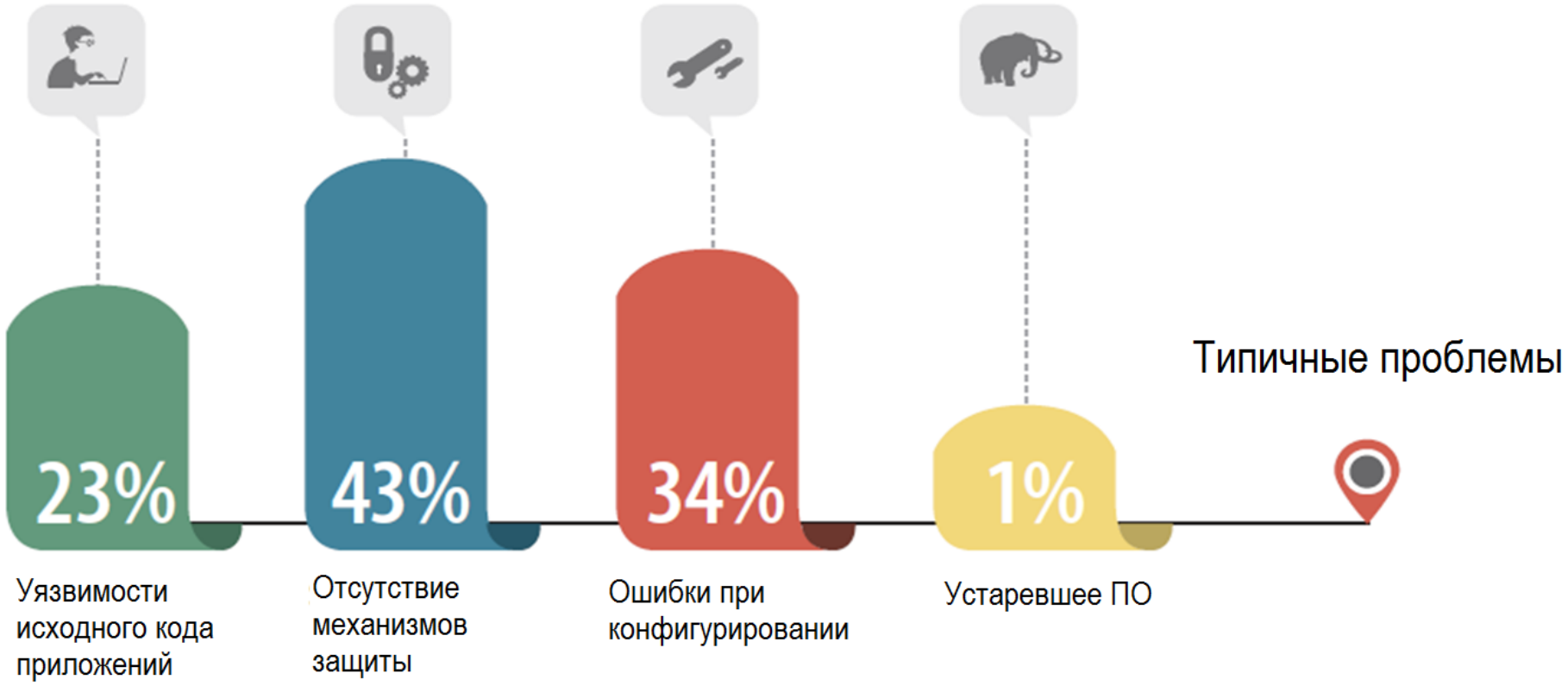


89% ВЕБ-ПРИЛОЖЕНИЙ БЫЛО УЯЗВИМО



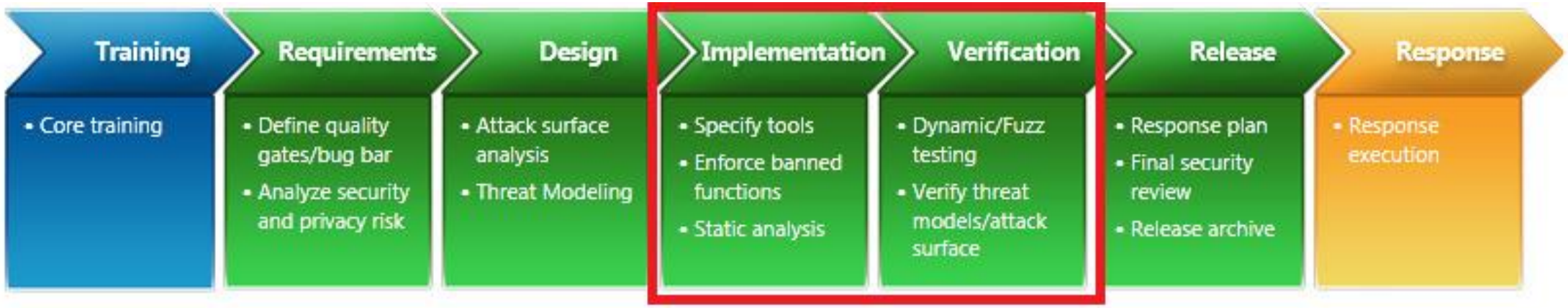
Причины появления уязвимостей в веб-приложениях

Основная проблема - Отсутствие процессов безопасной разработки приложений (SDLC)



Анализ уязвимостей – часть SDL

- Microsoft Security Development Lifecycle



- Cisco Secure Development Lifecycle
- OpenSAMM (Software Assurance Maturity Model)
- BSIMM
- etc…

Анализ уязвимостей – часть SDL

Стандартизация SDL в России

ГОСТ «Защита информации. Разработка безопасного программного обеспечения. Общие требования»

- разработка и принятие стандарта разработки программного обеспечения;
- проведение статического анализа;
- проведение динамического анализа;
- проведение тестирования на проникновение и фаззинг-тестирования;
- анализ уязвимостей – систематический процесс;
- необходимость использования инструментальных средств.

Анализ уязвимостей – часть SDL

Стандартизация SDL в России

ГОСТ Р ИСО/МЭК ТО 20004-2 «Информационная технология. Методы и средства обеспечения безопасности. Детализация анализа уязвимостей программного обеспечения в соответствии с ГОСТ Р ИСО/МЭК 15408 и ГОСТ Р ИСО/МЭК 18045. Часть 2. Тестирование проникновения»

- стандарт для испытательных лабораторий;
- независимый анализ уязвимостей, начиная с ОУД2;
- рекомендации к использованию инструментальных средств.

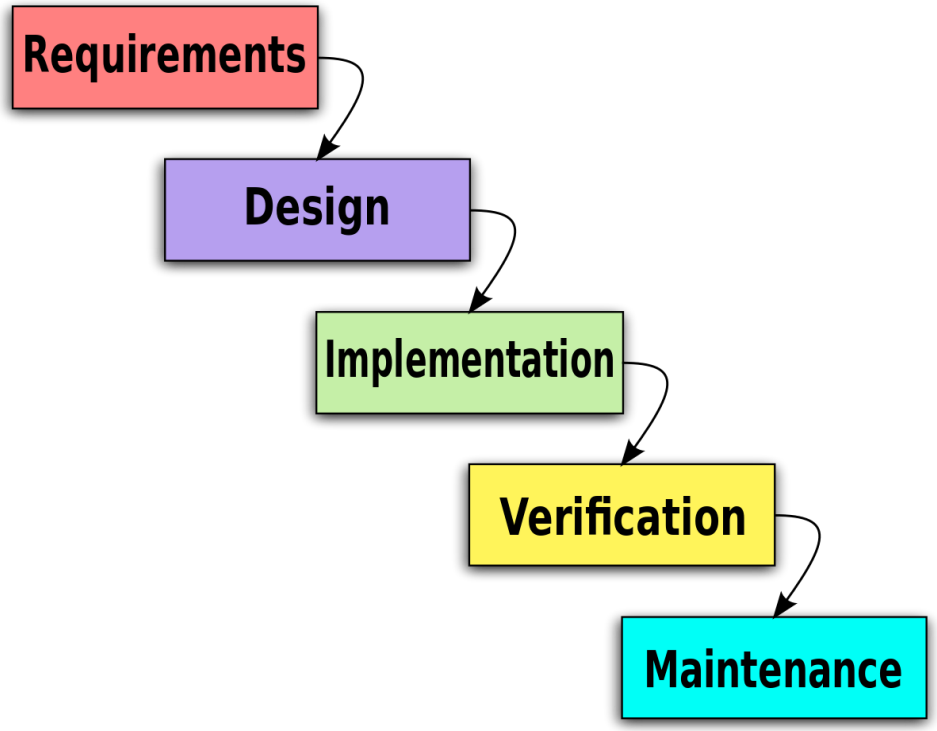
ГОСТ Р 51624 «Защита информации. Автоматизированные системы в защищеном исполнении. Общие требования»

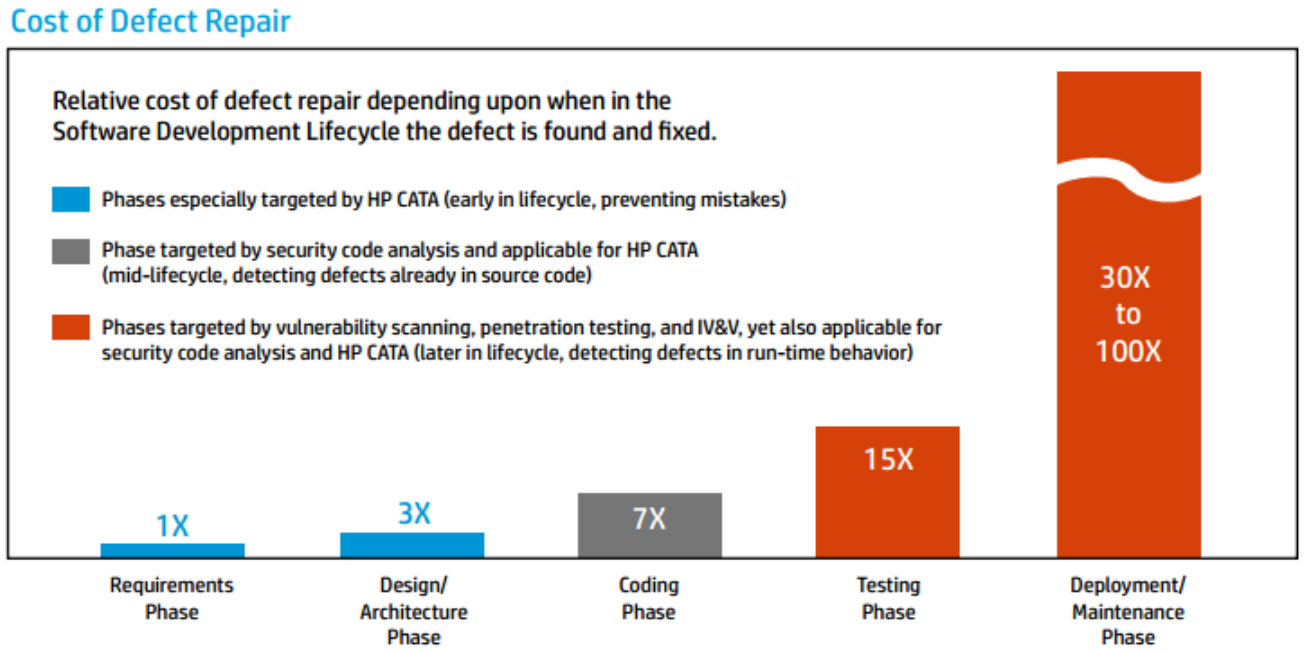
- стандарт для вендоров, чье ПО используется в ГИС.

Что мы хотим?

- Знать об уязвимости приложения наверняка
  - И желательно – раньше злоумышленников
- Инструменты предотвращения эксплуатации уязвимостей приложения
  - А не только средства защиты сети или сервера
- Защищать приложение еще до выхода исправления
- Защищать от эксплуатации еще не известных уязвимостей
- Сделать анализ уязвимостей частью процесса разработки

Чем раньше – тем лучше





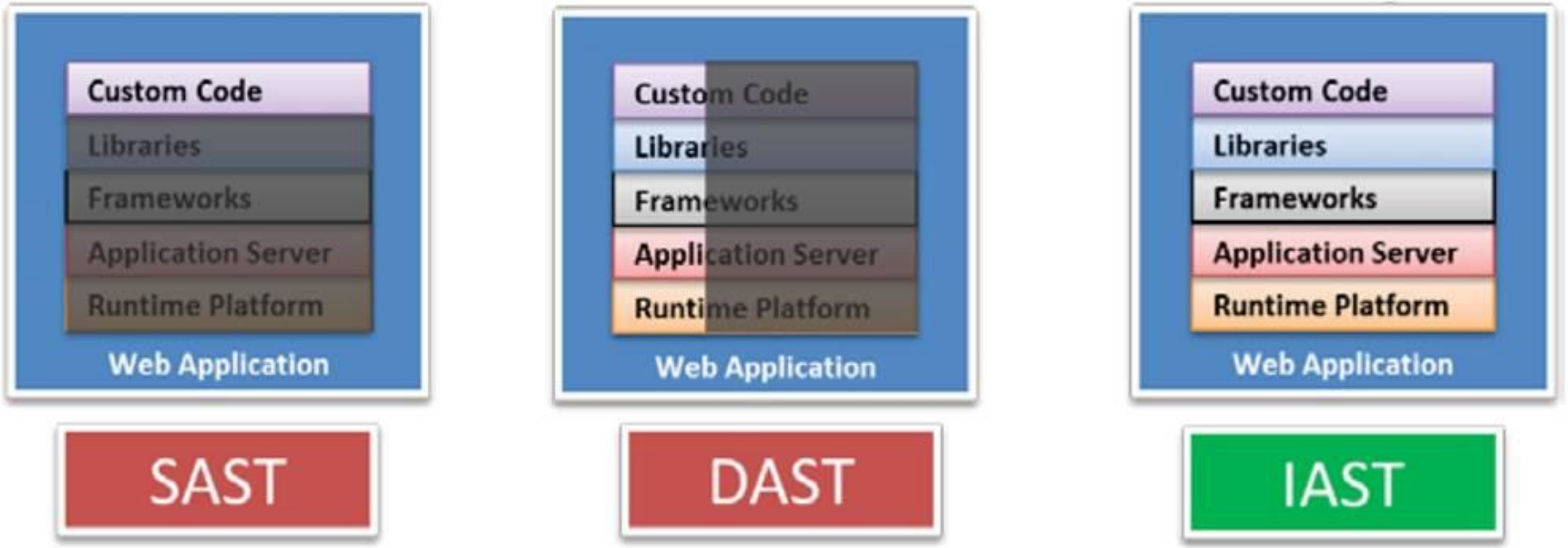
SAST vs DAST?

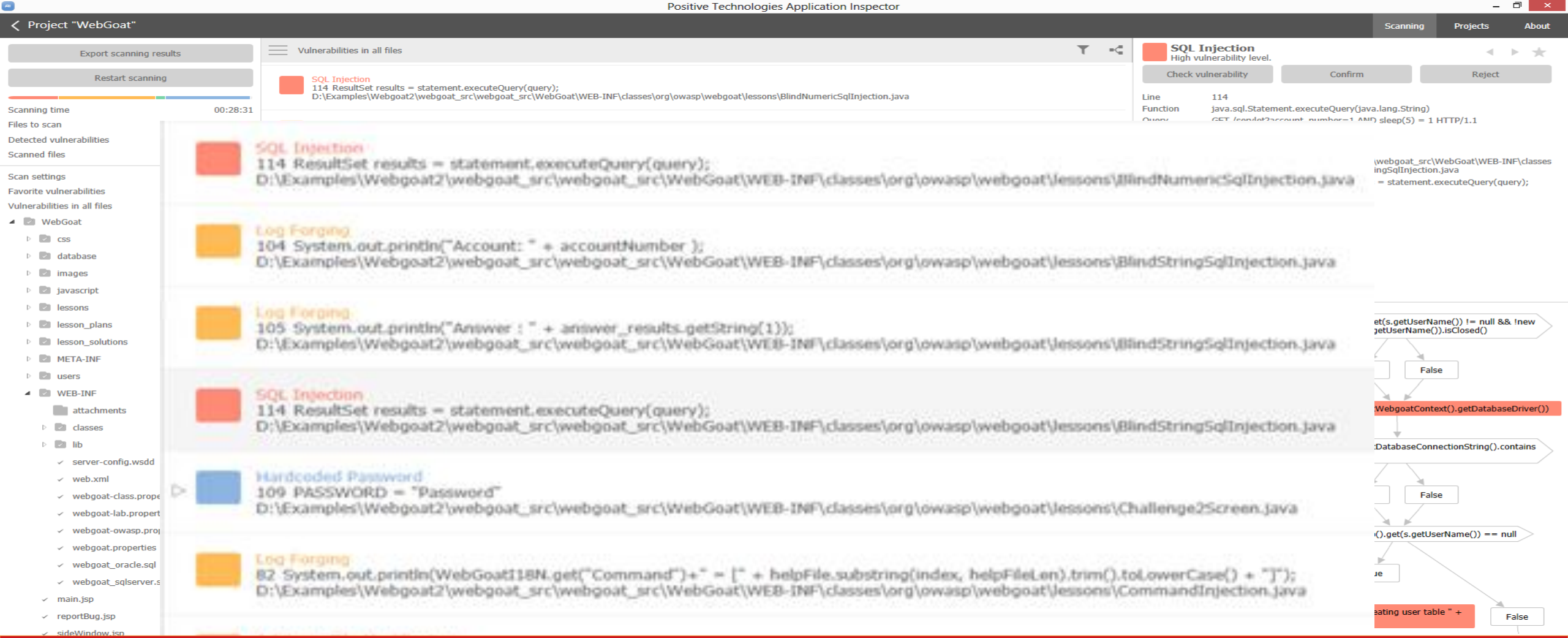
- Плюсы:
  - Хорошее покрытие (в теории)
  - Хорошая производительность (в теории)
  - Не требуется развернутое приложение
- Минусы
  - Сложно реализовать
  - Сложно верифицировать результаты
  - Зависит от языка разработки
  - «Проблема останова»

- Плюсы:
  - Легко реализовать;
  - Легко верифицировать результат;
  - Не зависит от языка/фреймворка/платформы.
- Минусы
  - Плохое покрытие;
  - Необходимо развернутое приложение (которое, вероятно, «упадет»);
  - Невозможно перебрать все варианты.

SAST + DAST!

SAST + DAST = IAST

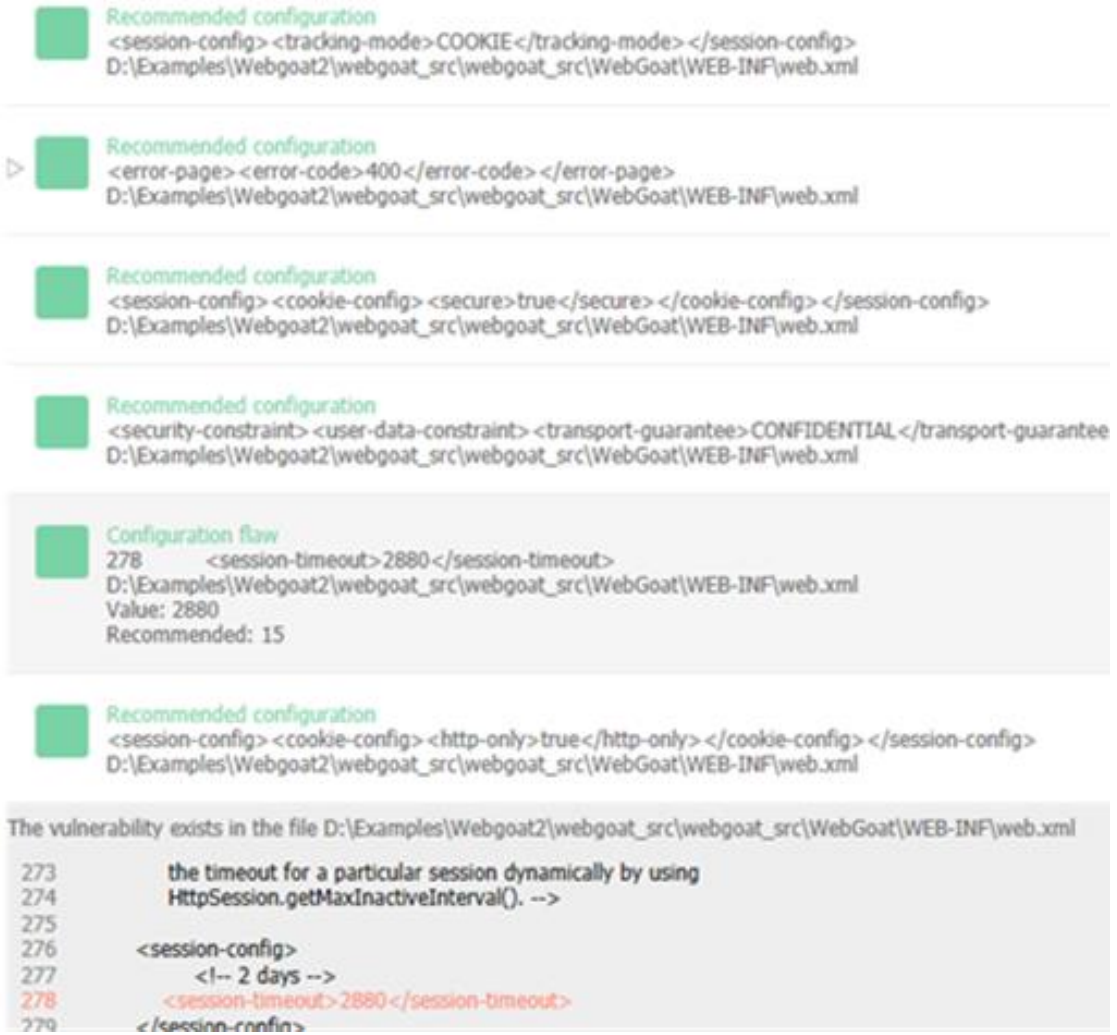




PT Application Inspector

Анализ конфигурации

- Анализ конфигурации
  - Уровня сервера (httpd.conf, server.xml …)
  - Уровня приложения (.htapasswd, web.xml, web.config…)
- Использование security best practice

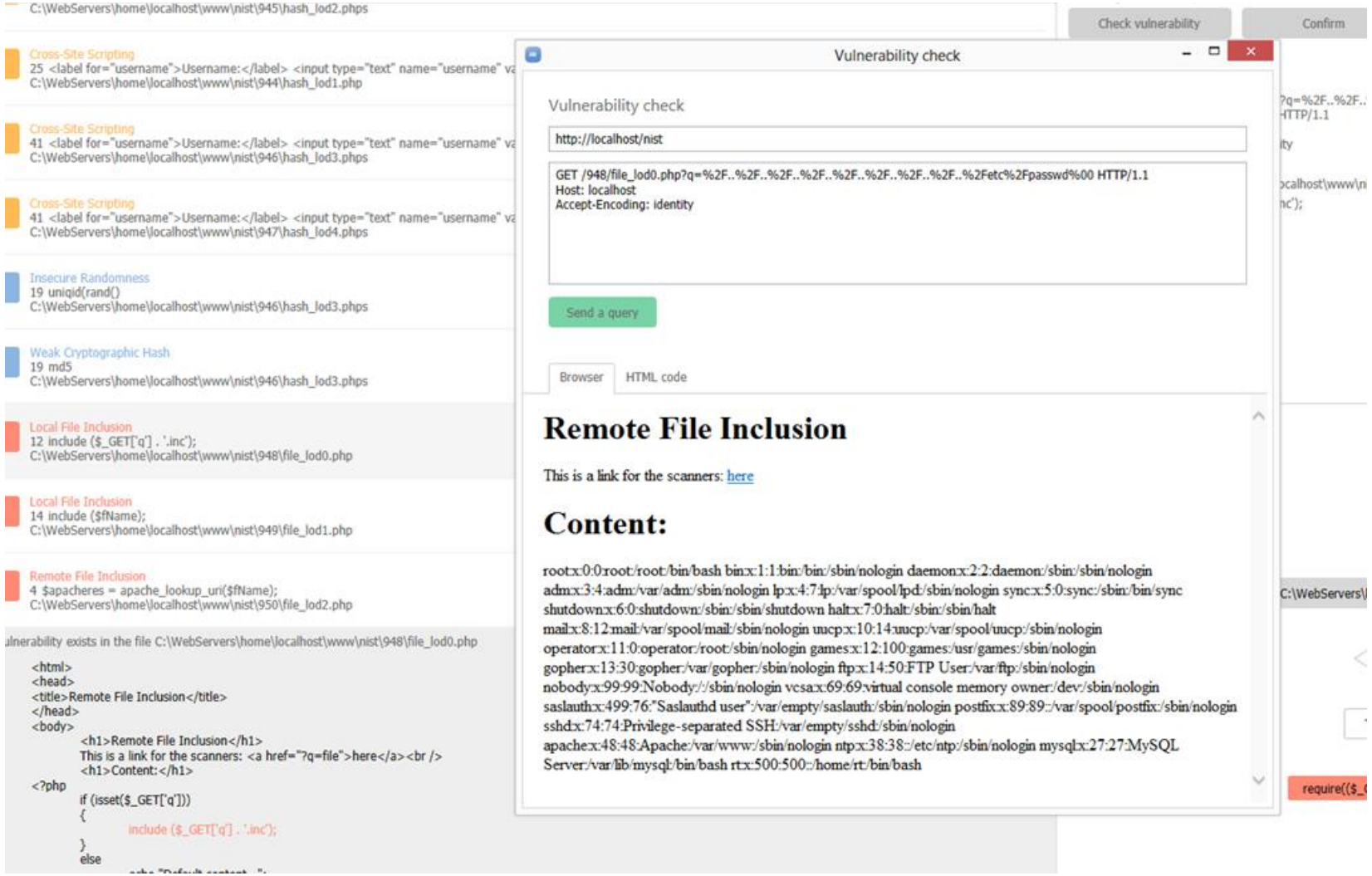


Анализ сторонних компонентов

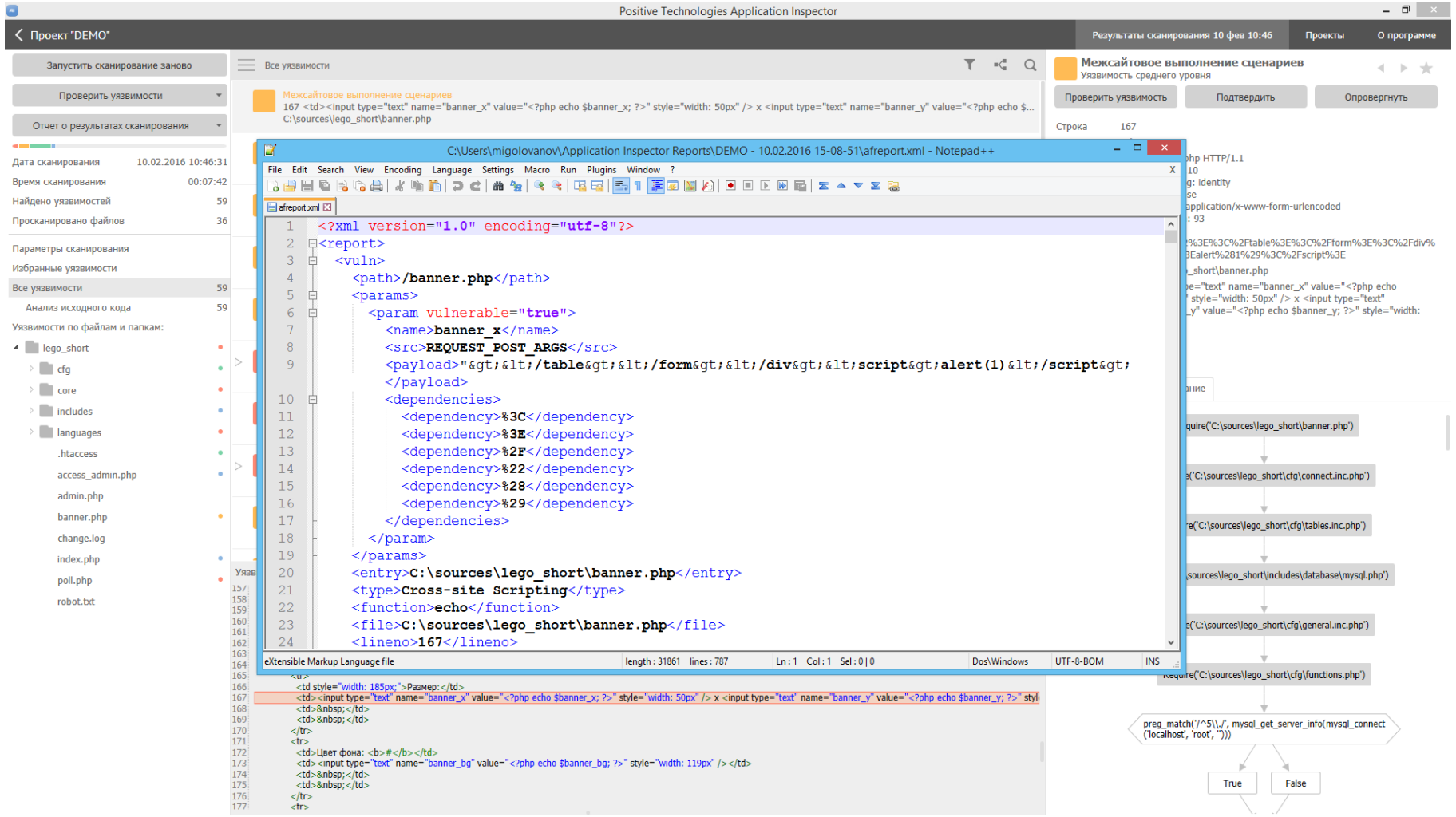
Fingerprinting

- Поиск уязвимостей в сторонних компонентах
  - Фреймворках и библиотеках
  - Open Source модулях (jQuery, YUI, …)
- Бэкдоры и инструменты удаленного администрирования
- Способы обнаружения
  - Информация о файлах (размер, имя…)
  - Хэши файлов
  - Участки кода
- Совместимая с CVE база знаний

Автоматическая генерация эксплойтов



Защита от эксплуатации уязвимостей



PT Application Inspector

- Акцент на обнаружении уязвимостей, а не на проблемах с оформлением
- Режим “Большой Красной Кнопки”
- Решение как для команды разработки, так и для команды ИБ
- Наличие рабочего приложения не обязательно
- Минимальное количество ложных срабатываний
- Автоматическая генерация эксплойтов
- Автоматическая генерация правил фильтрации

Анализ уязвимостей

Спасибо!

Михаил Федоров
mfedorov@ptsecurity.com

POSITIVE TECHNOLOGIES

ptsecurity.com