Жуков Алексей

Безопасная разработка в больших проектах, или Как улучшить безопасность ПО без отрыва от производства

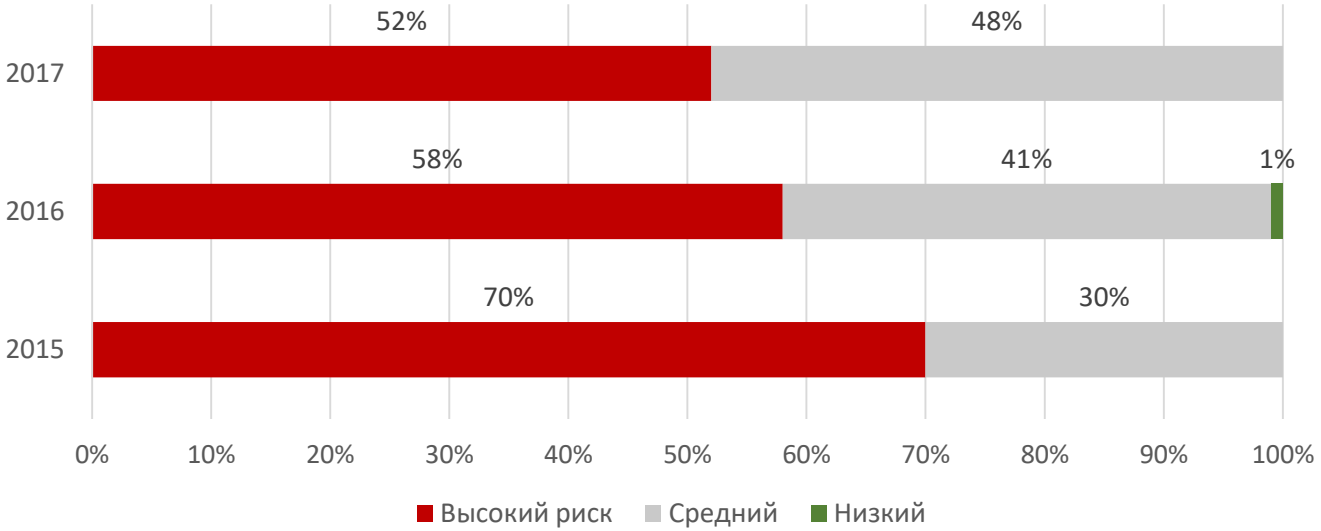
POSITIVE TECHNOLOGIES

alzhukov@ptsecurity.com

Уязвимости — насколько это серьезно?

POSITIVE TECHNOLOGIES

- Уязвимостей по-прежнему много:
  - 65% уязвимостей вызваны ошибками разработчика
  - В каждом пятом приложении используется устаревшее ПО
- Утечки измеряются миллиардами записей (River City Media)
  - Уязвимости фреймворков → еще одна проблема



«Positive Research 2018» report
(Positive Technologies, 2018)

Откуда же они берутся?

Откуда берутся уязвимости (1/6)

Сценарий #1

Иванова Мария

Откуда берутся уязвимости (2/6)

POSITIVE TECHNOLOGIES

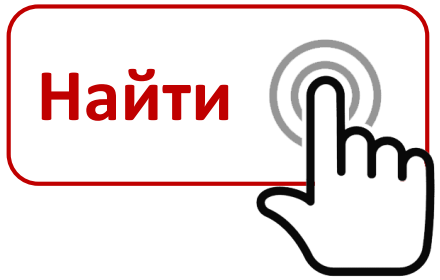
Реализация

```
import java.sql.Connection;
import java.sql.ResultSet;
import javax.sql.DataSource;

Connection conn = DB.getDataSource.getConnection();
String sql = "SELECT * FROM EMPLOYEES WHERE FULLNAME = '" + id + "'";
ResultSet rs = conn.createStatement().executeQuery(sql);
```

Откуда берутся уязвимости (3/6)

Атака

```
1' or SLEEP(5) = '
```



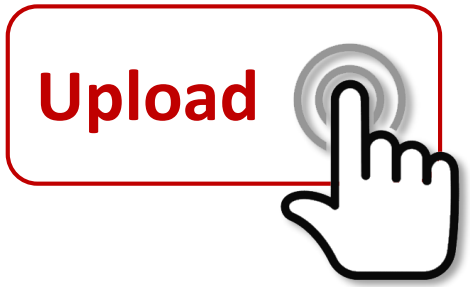
Результат

Откуда берутся уязвимости (4/6)

Сценарий #2

1. XML-файл
2. Веб-страница
3. Сообщение

```
<name>Мария</name>
```



Сообщение:

Мария, файл успешно загружен

Откуда берутся уязвимости (5/6)

Реализация

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;

String xml = requestData.get("xml");
DocumentBuilder db = DocumentBuilderFactory.newInstance().newDocumentBuilder();
Document doc = db.parse(new ByteArrayInputStream(xml.getBytes()));
return doc.getElementsByTagName("name").item(0).getTextContent();
```

Откуда берутся уязвимости (6/6)

POSITIVE TECHNOLOGIES

Атака

```
<!DOCTYPE name [
<!ELEMENT name ANY>
<!ENTITY xxe SYSTEM "file:./logs/application.log" >]>
<name>&xxe;</name>
```

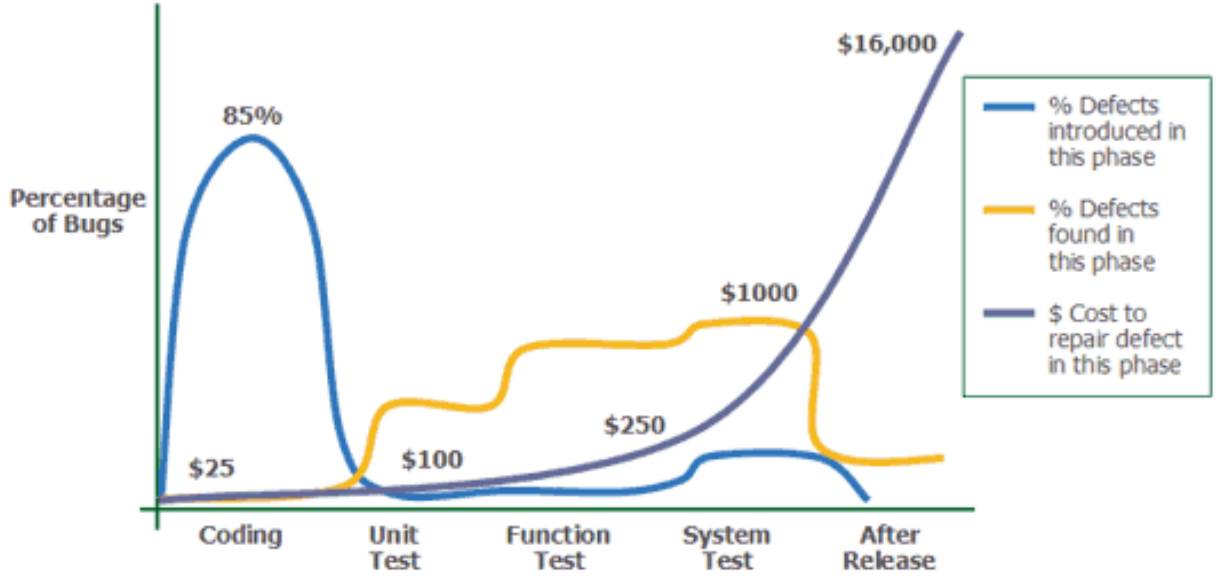
Результат

```
2018-08-13 15:19:21,022 - [INFO] - from play in pool-4-thread-1 Listening for HTTP
on /0:0:0:0:0:0:0:0:9000
2018-08-13 15:20:57,665 - [INFO] - from play in play-internal-execution-context-1
database [default] connected at jdbc:sqlite:./resources/db/base.sqlite
2018-08-13 15:20:57,735 - [INFO] - from play in play-internal-execution-context-1
Application started (Dev)
```

С уязвимостями надо бороться

POSITIVE TECHNOLOGIES

Разработка — экономика



- Applied Software Measurement (Capers Jones, 1996)
- Software Engineering Economics (Barry W. Boehm, 1983)
- Кратность роста стоимости изменений — от ×4 до ×100(!)

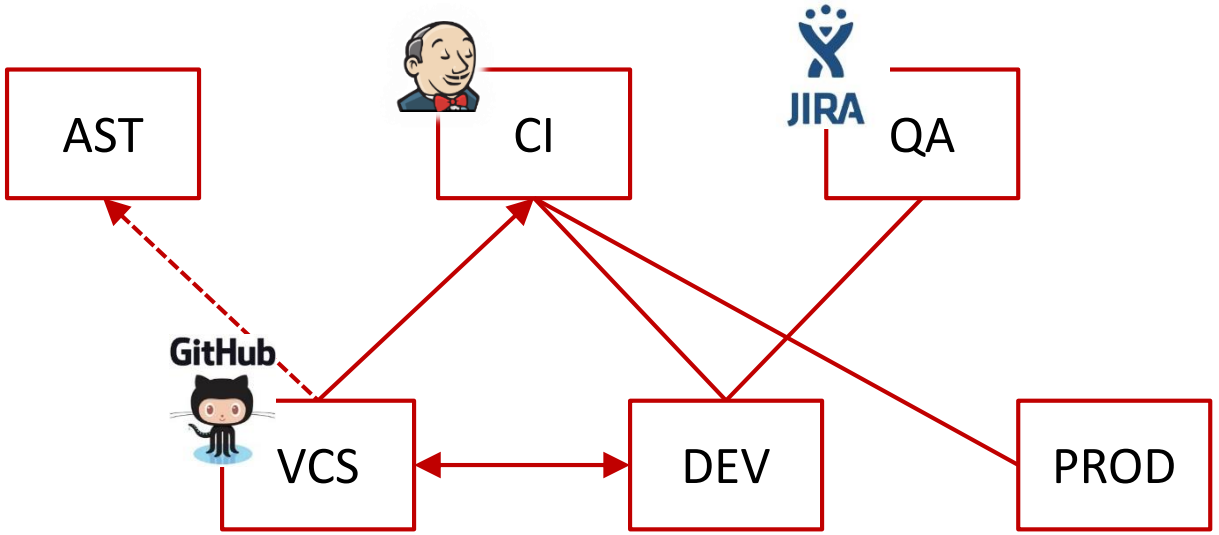
Выход — выявление дефектов ПО на ранних стадиях разработки и методологии непрерывной интеграции Agile/DevOps

Встраиваем анализ кода

Как это иногда может выглядеть:

- Протестировали решения, выбрали подходящее
- Добавили в CI шаг с вызовом анализатора (AST)

Что получаем:



Здесь чего-то не хватает…

Встраиваем анализ кода

POSITIVE TECHNOLOGIES

Что имеем:

- Автоматическое сканирование кода
- Публикацию отчетов

И как это соотносится с “continuous” в CI?

- Кто и когда читает отчеты (two-thousand-pages-report.pdf)?
- Что делать с ложными срабатываниями?
- Как автоматически среагировать на появление опасной уязвимости?

Конечная цель — сделать приложение безопасным

Что для этого нужно?

POSITIVE TECHNOLOGIES

Автоматизация:

- Принятие четкого решения «да/нет» на основе правил
- Автоматический останов сборки по итогам оценки соответствия

Сходимость процесса:

- Минимизация ручной работы по разбору результатов
- Разбор только новых уязвимостей

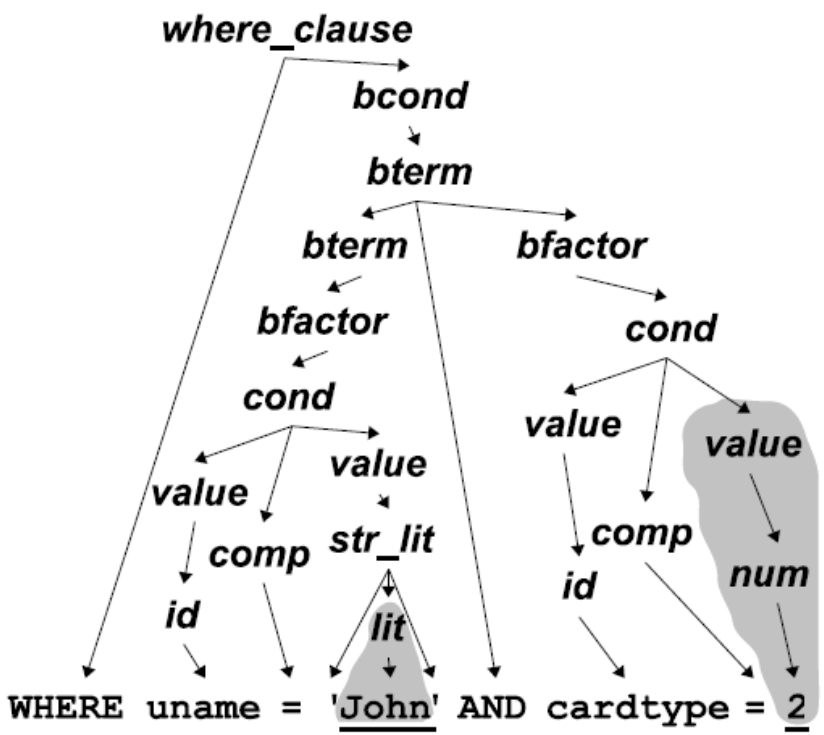
Результаты должны быть достоверными

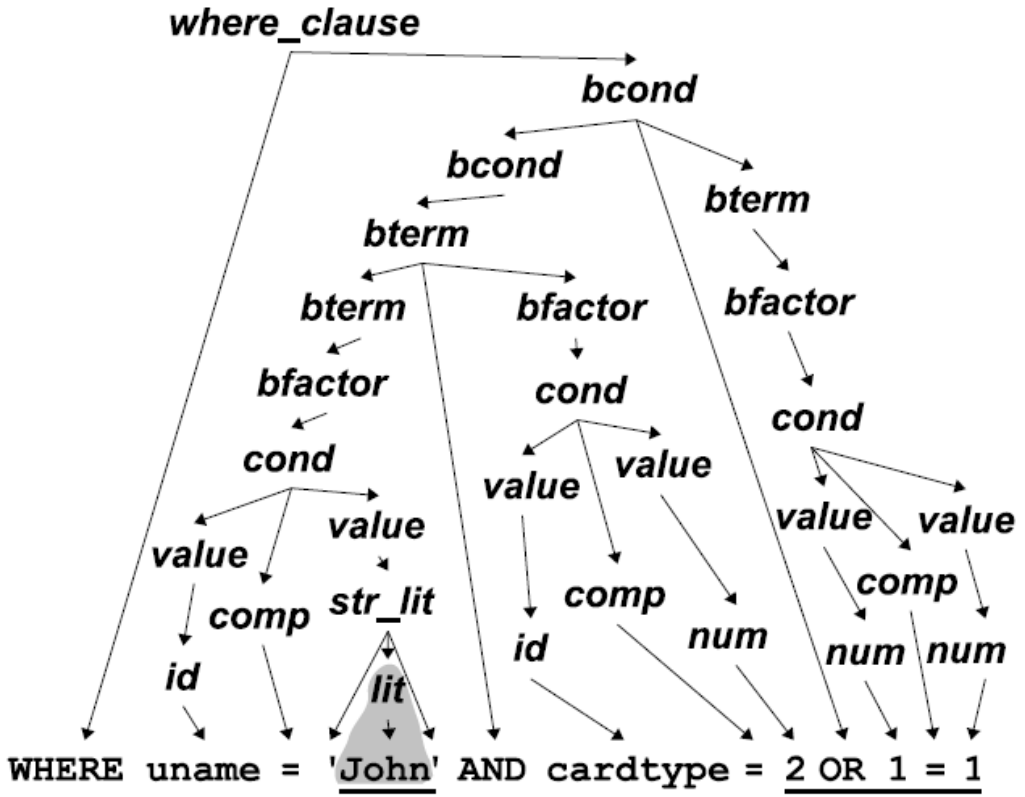
Почему они недостоверны?

POSITIVE TECHNOLOGIES

Формализация понятия уязвимость

- Не для всех классов уязвимостей одинаково хороша
- Для инъекций:

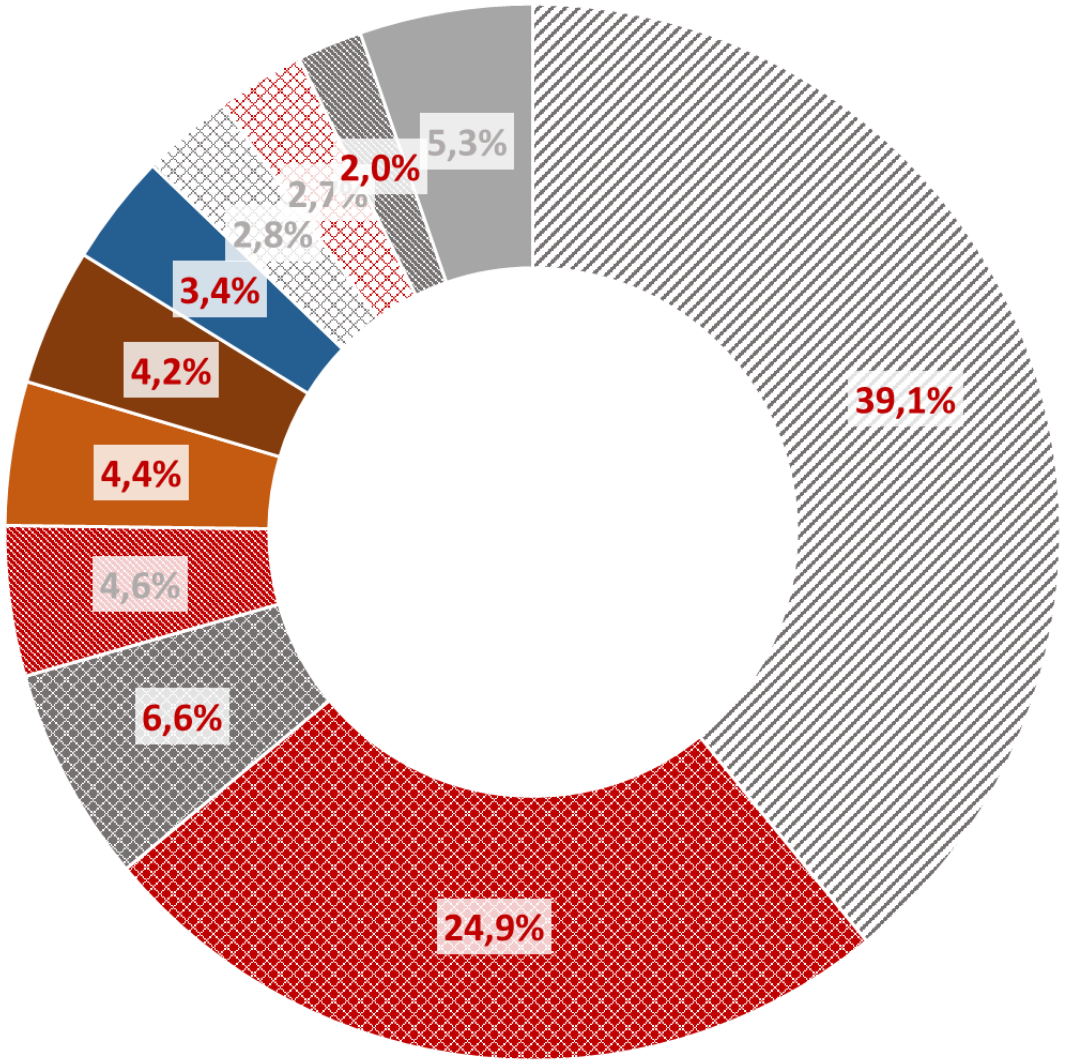




The Essence of Command Injection Attacks in Web Applications (http://web.cs.ucdavis.edu/~su/publications/popl06.pdf)

84.6%

POSITIVE TECHNOLOGIES



- Межсайтовое выполнение сценариев
- Внедрение SQL-кода
- Выход за пределы назначенного каталога
- Утечка информации
- Удаленное выполнение кода и команд ОС
- Внедрение конструкций XML
- Подделка запроса со стороны сервера
- Отказ в обслуживании
- Ошибка настройки контроля доступа для методов протокола HTTP
- Внедрение шаблона на стороне сервера
- Другие атаки

Статистика атак на веб-приложения
(https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/WebApp-Vulnerabilities-2017-rus.pdf)

Давайте попробуем найти

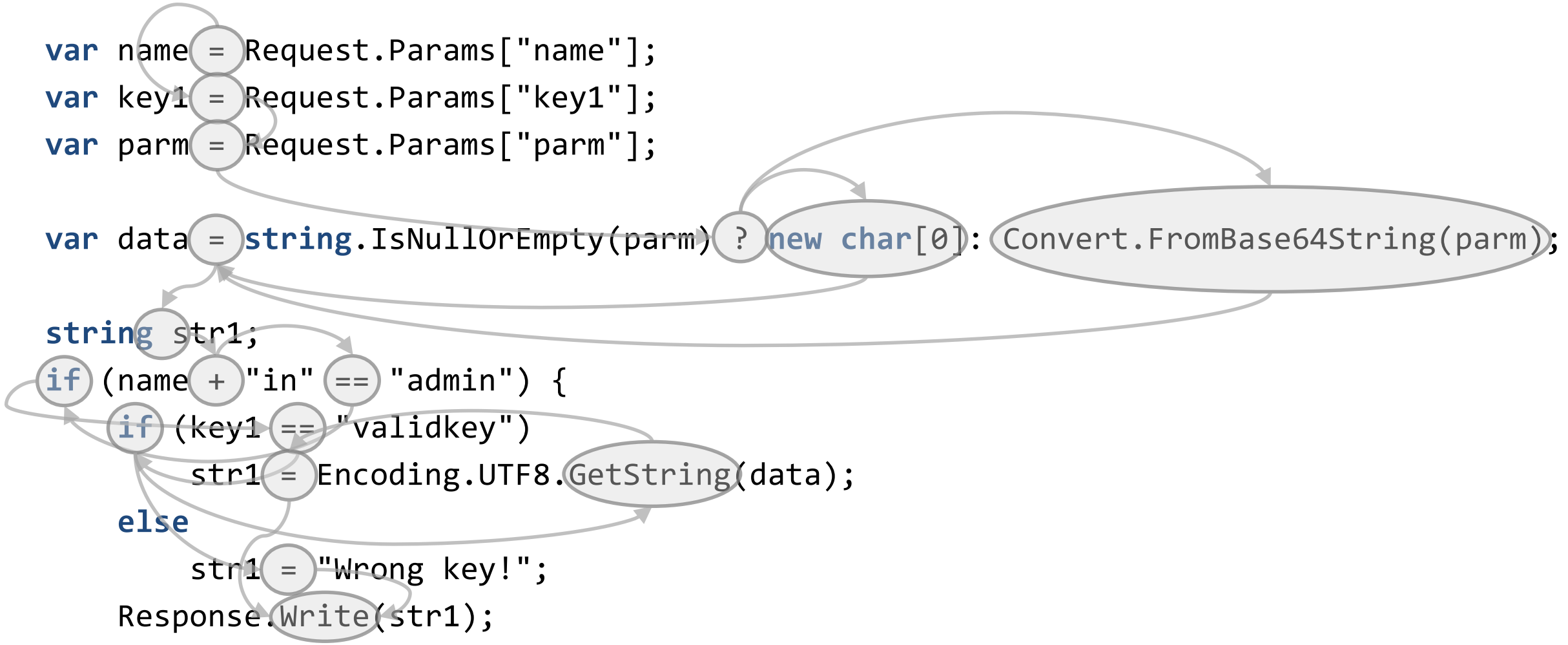
Для этого нужно

- «Понять» алгоритм программы
- «Понять» как программа обрабатывает данные

Первый шаг: поток управления

Пример графа потоков управления

POSITIVE TECHNOLOGIES

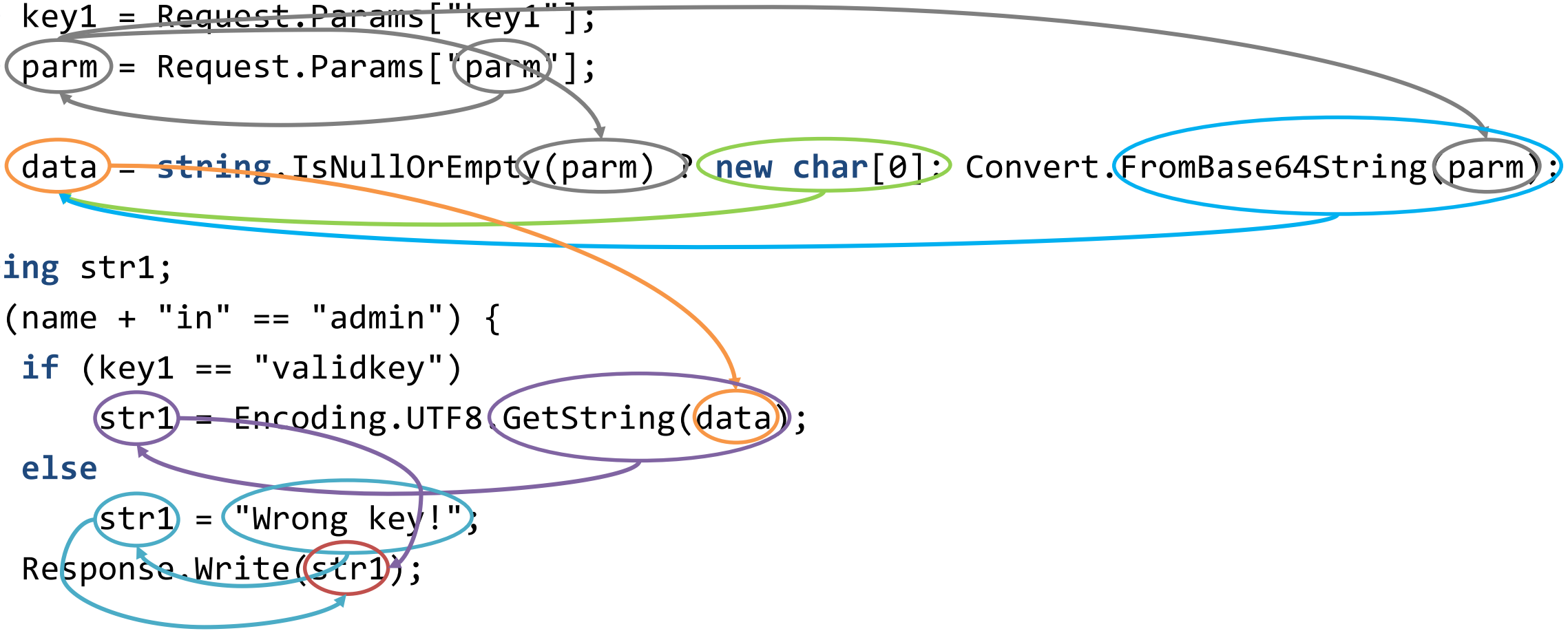


Второй шаг: потоки данных

Пример графа потоков данных

POSITIVE TECHNOLOGIES

```
var name = Request.Params["name"];
var key1 = Request.Params["key1"];
var parm = Request.Params["parm"];

var data = string.IsNullOrEmpty(parm) ? new char[0]: Convert.FromBase64String(parm);

string str1;
if (name + "in" == "admin") {
    if (key1 == "validkey")
        str1 = Encoding.UTF8.GetString(data);
    else
        str1 = "Wrong key!";
    Response.Write(str1);
}
```



Что мы получим на этом этапе

POSITIVE TECHNOLOGIES

Результаты анализа

```
String ID = request.getParameter("id");
String SQL = "SELECT cname FROM emp WHERE id = '" + ID + "'";
ResultSet RES = DB.createStatement().executeQuery(SQL);
```

http://host/App?id=1' OR 1=1 --

SQL Injection

Что мы получим на этом этапе

POSITIVE TECHNOLOGIES

Результаты анализа

```
if (3 == 2 + 2) { // Code between {…} is dead, there's no vuln
    String ID = request.getParameter("id");
    String SQL = "SELECT cname FROM emp WHERE id = '" + ID + "'";
    ResultSet RES = DB.createStatement().executeQuery(SQL);
}
```

~~http://host/App?id=1' OR 1=1 --~~

~~SQL Injection~~

Что мы получим на этом этапе

POSITIVE TECHNOLOGIES

Результаты анализа

```
if (3 == 2 + 2) { // Code between {…} is dead, there's no vuln
    String ID = request.getParameter("id");
    String SQL = "SELECT cname FROM emp WHERE id = '" + ID + "'";
    ResultSet RES = DB.createStatement().executeQuery(SQL);
}
```

~~http://host/App?id=1' OR 1=1 --~~

~~SQL Injection~~

Нужно улучшать качество анализа

Третий шаг: состояние потока данных

Состояние потока данных

POSITIVE TECHNOLOGIES

```
var name = Request.Params["name"];
var key1 = Request.Params["key1"];
var parm = Request.Params["parm"];

var data = string.IsNullOrEmpty(parm) ? new char[0]: Convert.FromBase64String(parm);

string str1;
if (name + "in" == "admin") {
    if (key1 == "validkey")
        str1 = Encoding.UTF8.GetString(data);
    else
        str1 = "Wrong key!";
    Response.Write(str1);
}
```

```
str1 ∈ {
    Encoding.UTF8.GetString(data),
    "Wrong Key!"
}
```

Четвертый шаг: состояние приложения

Состояние приложения

POSITIVE TECHNOLOGIES

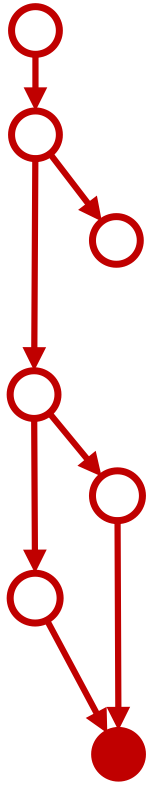
```
var name = Request.Params["name"];
var key1 = Request.Params["key1"];
var parm = Request.Params["parm"];

var data = string.IsNullOrEmpty(parm) ? new char[0]: Convert.FromBase64String(parm);

string str1;
if (name + "in" == "admin") {
    if (key1 == "validkey")
        str1 = Encoding.UTF8.GetString(data);
    else
        str1 = "Wrong key!";
    Response.Write(str1);
}
```

```
str1 ∈ {
    Encoding.UTF8.GetString(data),
    "Wrong Key!"
}

name ∈ { Request.Params["name"] }
key1 ∈ { Request.Params["key1"] }
parm ∈ { Request.Params["parm"] }

data ∈ {
    new char[0],
    Convert.FromBase64String(parm)
}
```

Пятый шаг: поток вычислений

Поток вычислений

POSITIVE TECHNOLOGIES



```
var parm = Request.Params["parm1"];

if (Request.Params["cond1"] == "true")
{
    return;
}

if (Request.Params["cond2"] == "true")
{
    parm = Request.Params["parm2"];
} else {
    parm = "<div>Harmless value</div>";
}

Response.Write("<a href=\"" + parm + "\">");
```

```
Request.Params["cond1"] != "true" ⇒ {
  parm' ∈ {
    Request.Params["cond2"] == "true" ⇒
      Request.Params["parm2"]
    ||
    Request.Params["cond2"] != "true" ⇒
      "<div>Harmless value</div>"
  }
}
```

Поток вычислений — это то, что нужно

Имея поток вычислений, для любой точки кода у нас будет:

- Набор условий достижимости
- Набор возможных значений данных
- Набор условий для заданных значений данных

Осталось совсем немного ;)

- Как построить поток вычислений, если возможных значений данных бесконечно много?

Абстрактная интерпретация на примере математики

POSITIVE TECHNOLOGIES

Задача: Получить знак выражения

- -42 / 8 × 100 500
  - -42 / 8 × 100500 == -527 625
- А если -42 / 8 × 100 500 + **X**?

Абстрактная интерпретация на примере математики

POSITIVE TECHNOLOGIES

Задача: Получить знак выражения

- -42 / 8 × 100 500
  - -42 / 8 × 100500 == -527 625
- А если -42 / 8 × 100 500 + **X**?

Добавим абстрактности:

- (+a) = (+)
- (-a) = (-)
- (-a) × (+b) = (-);
- (-a) / (+b) = (-);
- (-a) + (+b) =
  - a ≤ b → (+),
  - a > b → (-)

Абстрактная интерпретация на примере математики

POSITIVE TECHNOLOGIES

Задача: Получить знак выражения

- -42 / 8 × 100 500
  - -42 / 8 × 100500 == -527 625
- А если -42 / 8 × 100 500 + **X**?

Добавим абстрактности:

- (+a) = (+)
- (-a) = (-)
- (-a) × (+b) = (-);
- (-a) / (+b) = (-);
- (-a) + (+b) =
  - a ≤ b → (+),
  - a > b → (-)

Тогда:

- -42 / 8 × 100 500 + **X** =
  - X ≥ -527 625 → (+),
  - X < -527 625 → (-)

Абстрактная интерпретация позволяет проводить вычисления не зная конкретных данных

Реализация

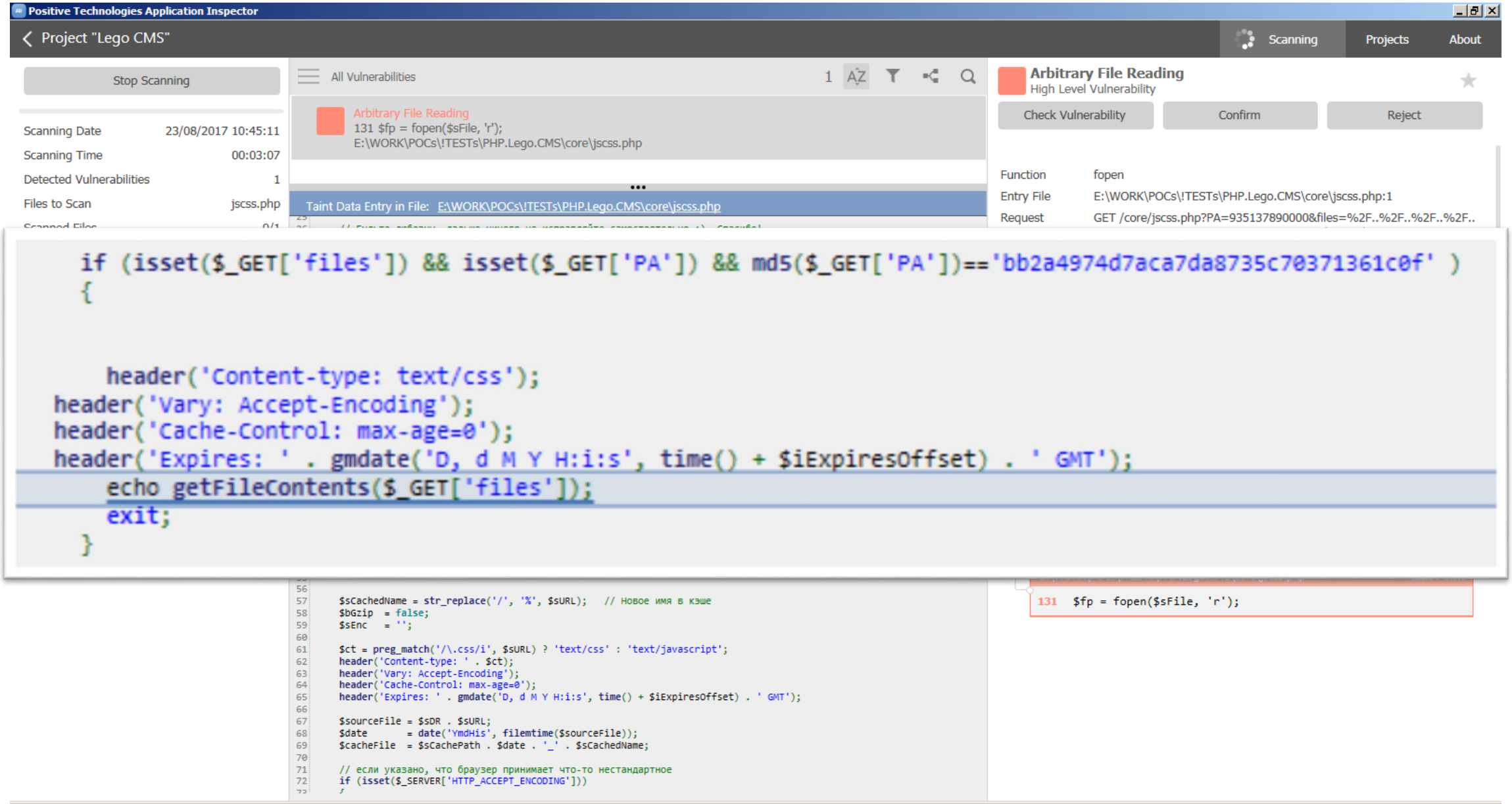
POSITIVE TECHNOLOGIES

Абстрактная интерпретация для потока вычислений позволяет:

- Анализировать неполный код: если нет исходников FakeMD5:
  - Поместить в набор условий эксплуатации уязвимости
- Минимизация ошибок первого рода
  - Генерация эксплойта, который может быть проверен
- Минимизация ошибок второго рода
  - Человекочитаемый набор условий для подозрений на уязвимости
- Выявление уязвимостей второго порядка

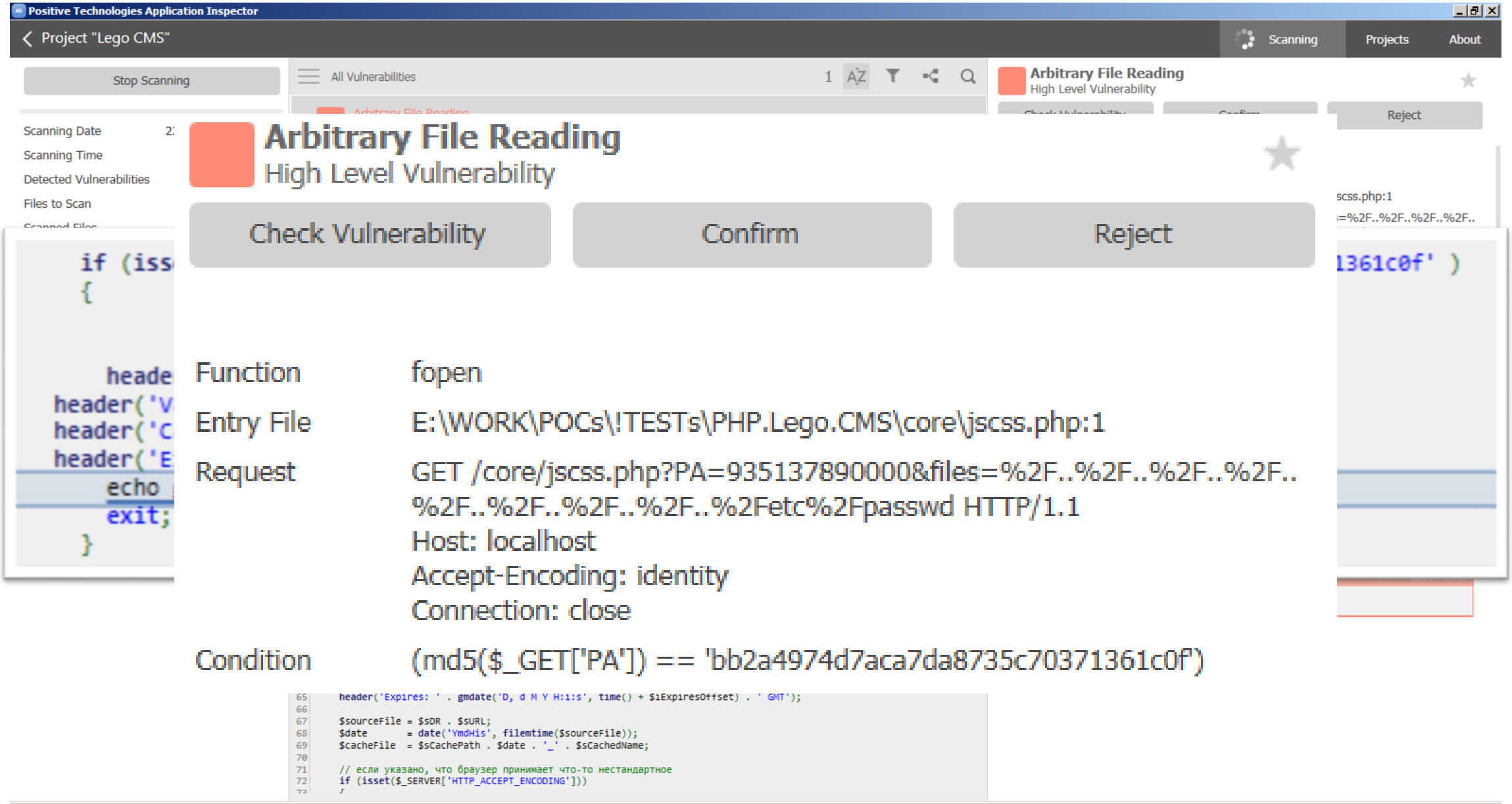
Как это выглядит «вживую»

POSITIVE TECHNOLOGIES



Как это выглядит «вживую»

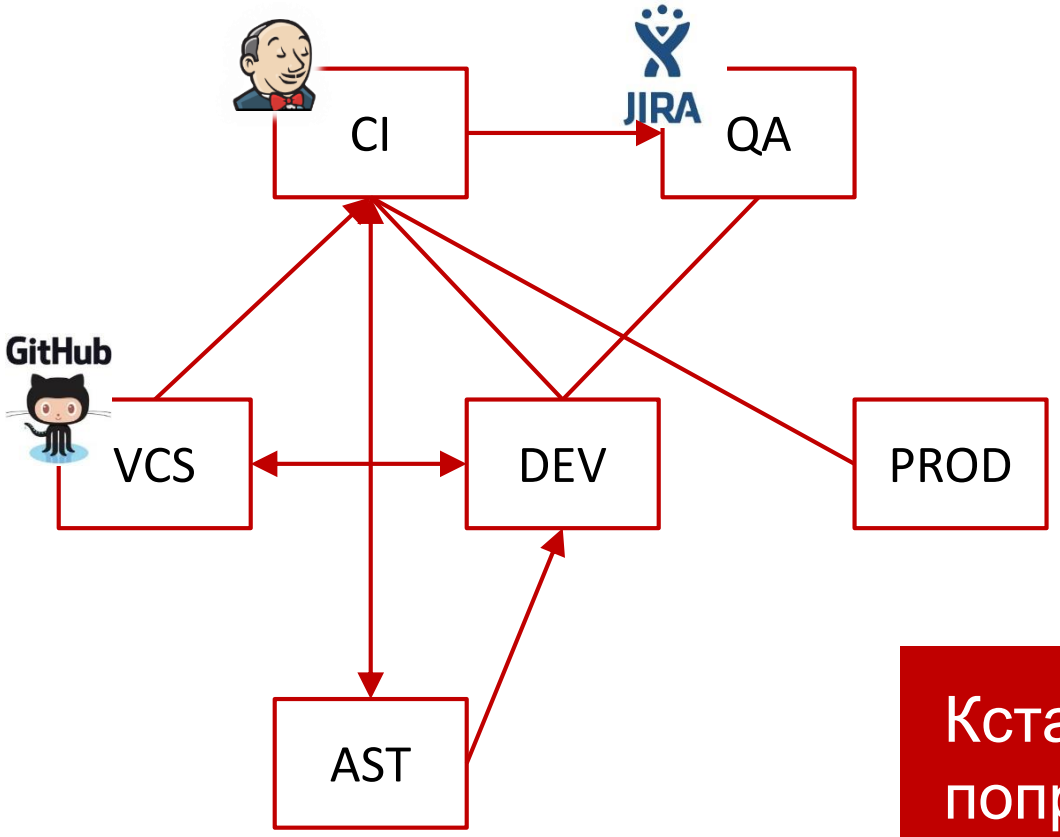
POSITIVE TECHNOLOGIES



Использование в CI/CD

POSITIVE TECHNOLOGIES

- Автоматизация: «политика» + «результаты» = «да / нет»
- Сходимость процесса: обработка только новых уязвимостей



Кстати, это можно попробовать у себя ;)

Вопросы?


Спасибо!
POSITIVE TECHNOLOGIES
ptsecurity.ru