

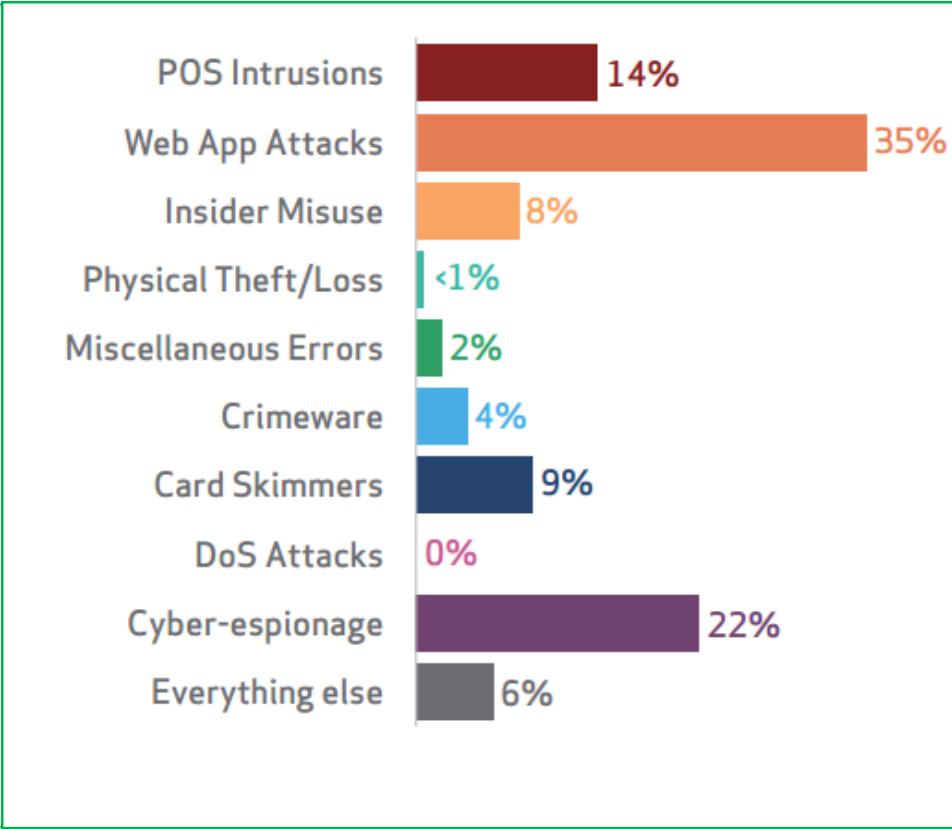
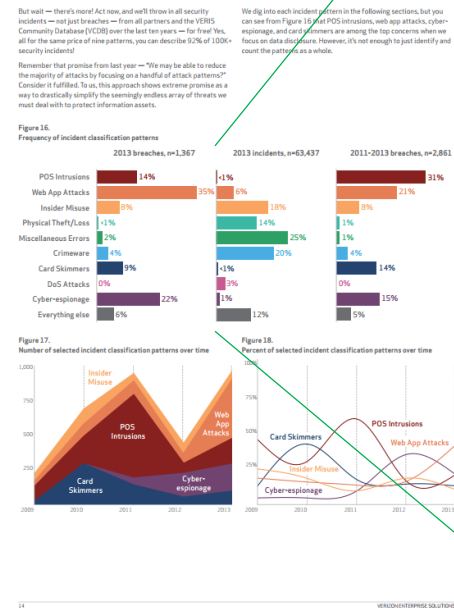
Анализ уязвимостей

POSITIVE TECHNOLOGIES

ptsecurity.com

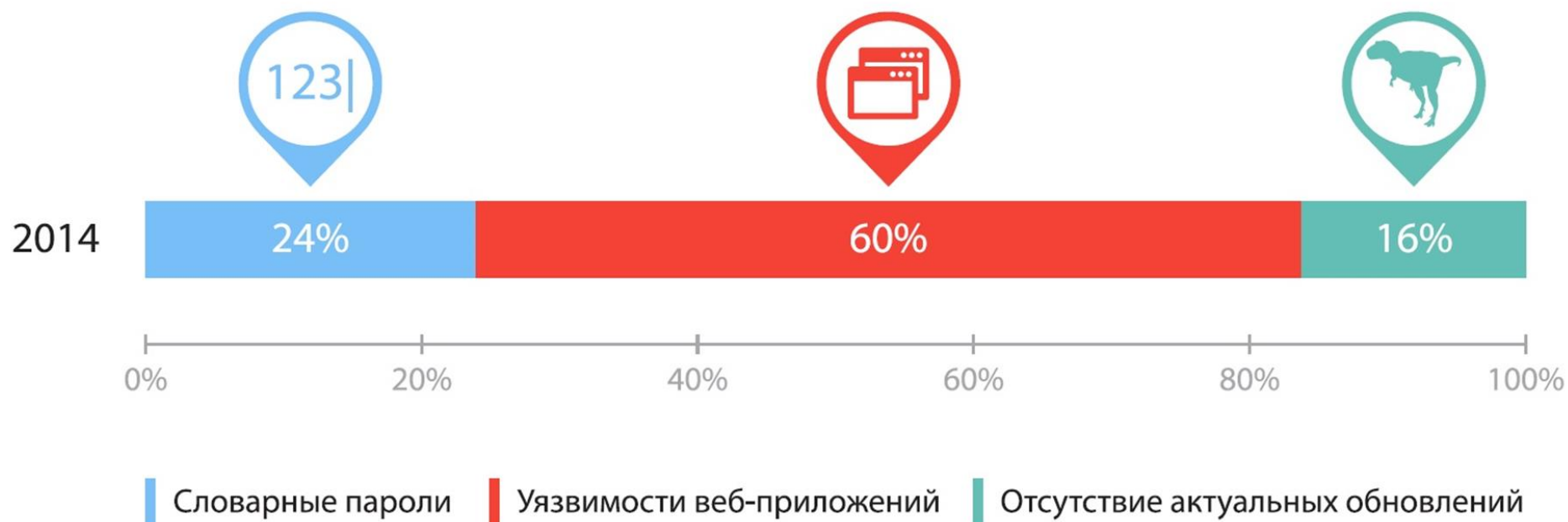
Международная статистика (Verizon)

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

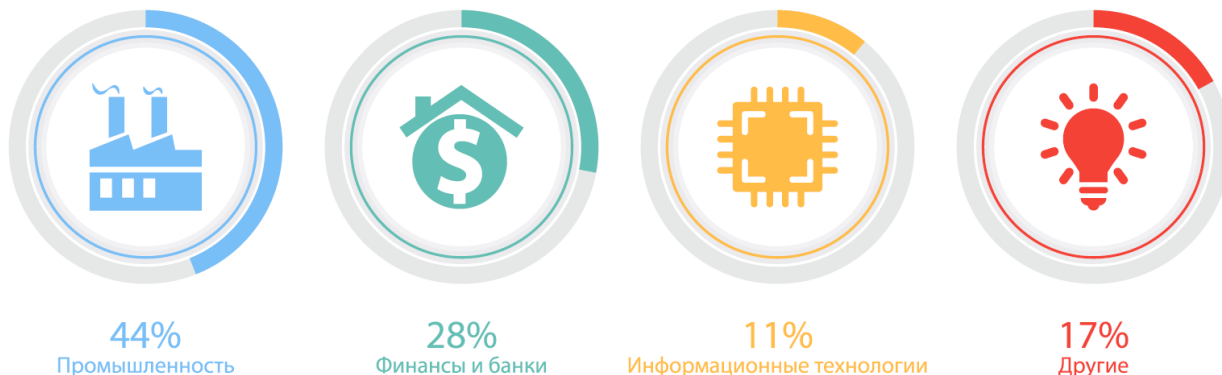


Verizon 2014 Data Breach Investigations Report

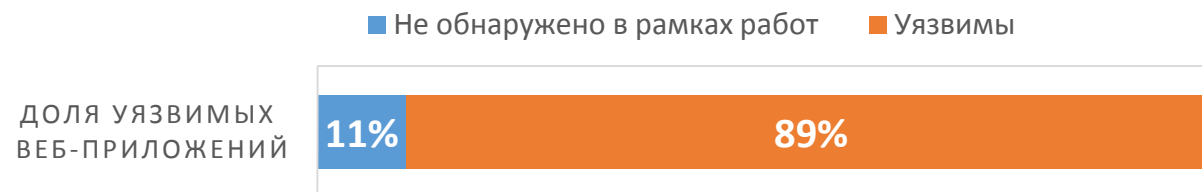
Векторы атак для преодоления сетевого периметра



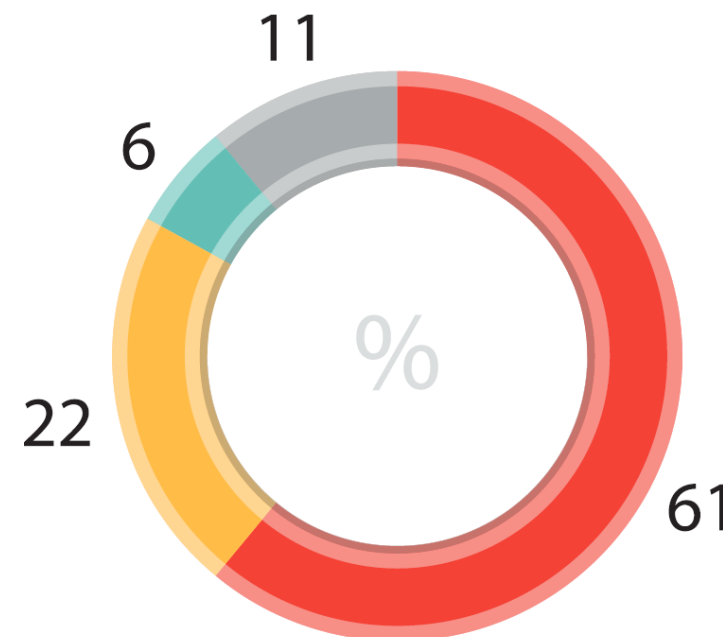
В проведенном исследовании среди различных отраслей



89% ВЕБ-ПРИЛОЖЕНИЙ БЫЛО УЯЗВИМО



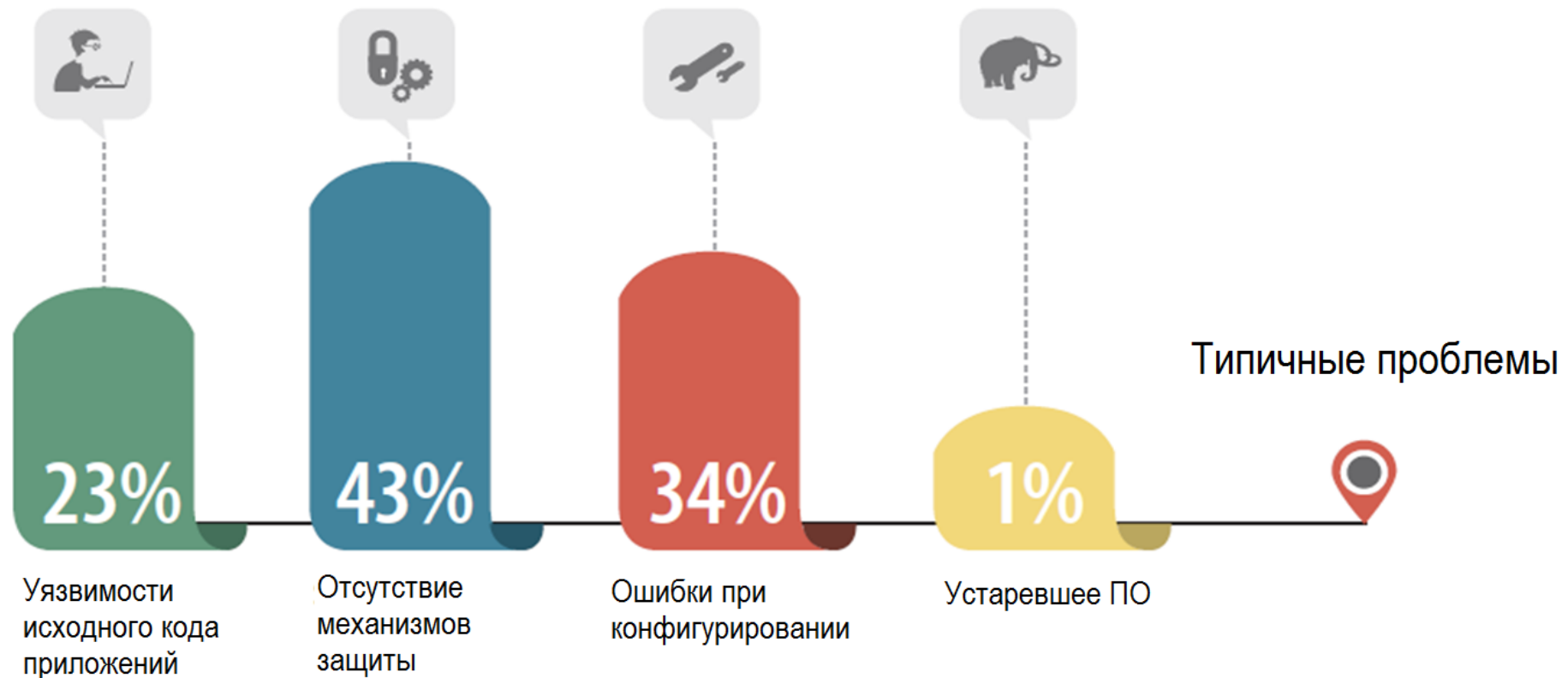
61% уязвимостей высокой критичности



Причины появления уязвимостей в веб-приложениях

4

Основная проблема - Отсутствие процессов безопасной разработки приложений (SDLC)



- Microsoft Security Development Lifecycle



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

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.

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 защищенном исполнении. Общие требования»

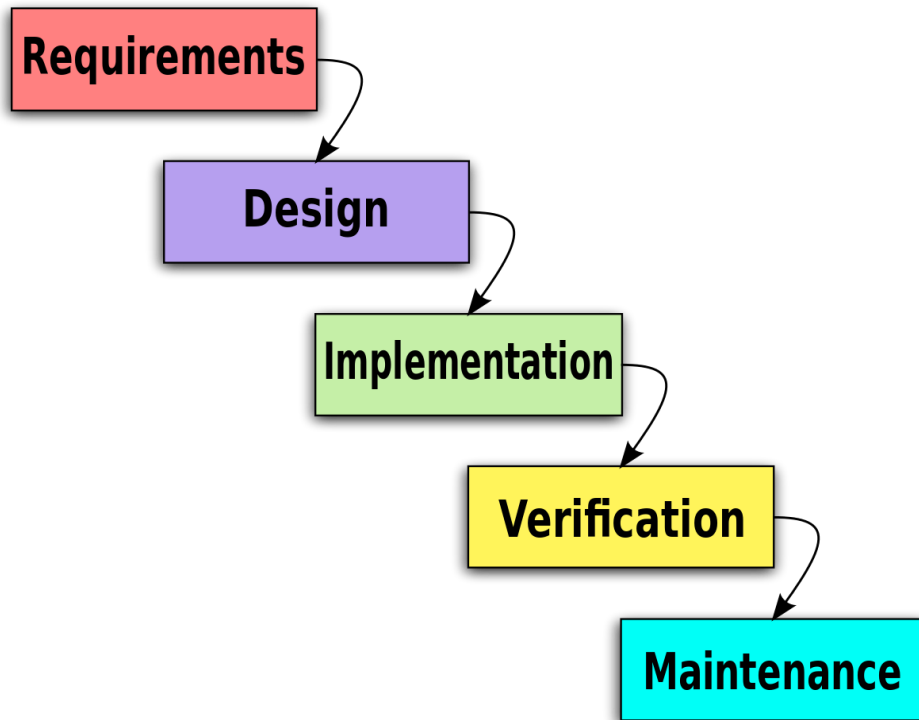
- стандарт для вендоров, чье ПО используется в ГИС.

Что мы хотим?

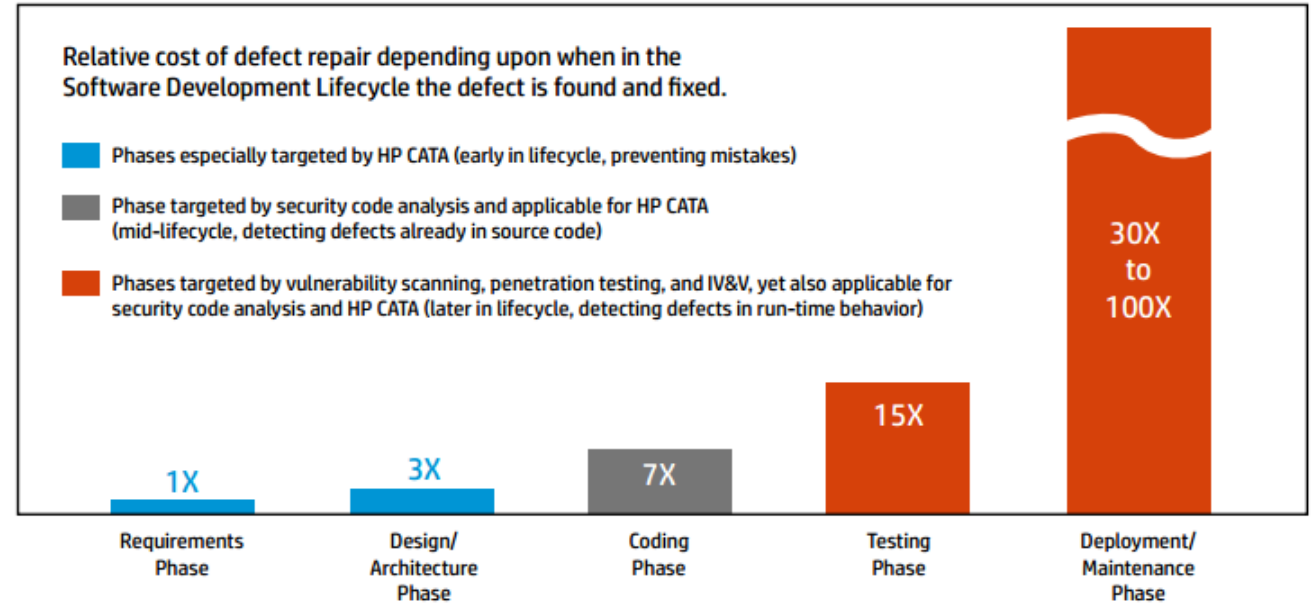
- Знать об уязвимости приложения наверняка
 - И желательно – раньше злоумышленников
- Инструменты предотвращения эксплуатации уязвимостей приложения
 - А не только средства защиты сети или сервера
- Защищать приложение еще до выхода исправления
- Защищать от эксплуатации еще не известных уязвимостей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

Чем раньше – тем лучше

9



Cost of Defect Repair



SAST vs DAST?

10

- 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»



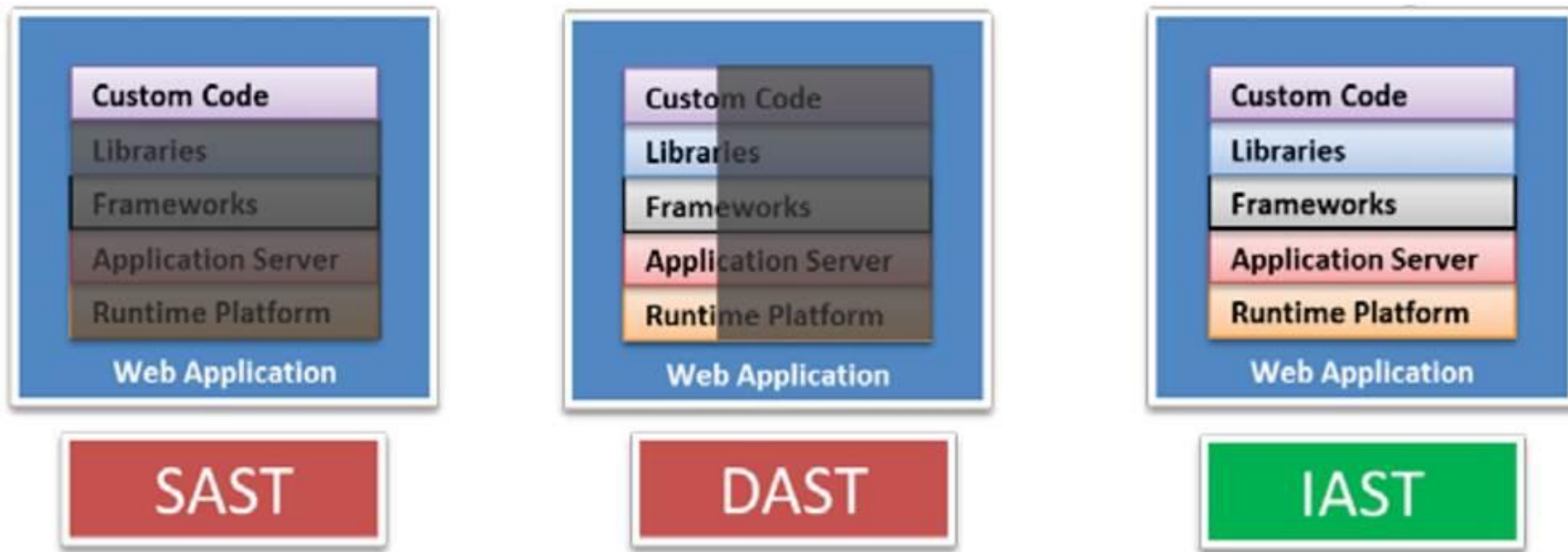
- Плюсы:
 - Легко реализовать;
 - Легко верифицировать результат;
 - Не зависит от языка/фреймворка/платформы.
- Минусы
 - Плохое покрытие;
 - Необходимо развернутое приложение (которое, вероятно, «упадет»);
 - Невозможно перебрать все варианты.



SAST + DAST!

11

SAST + DAST = IAST



Project "WebGoat"

Export scanning results

Restart scanning

Scanning time00:28:31

Files to scan

Detected vulnerabilities

Scanned files

Scan settings

Favorite vulnerabilities

Vulnerabilities in all files

WebGoat

css

database

images

javascript

lessons

lesson_plans

lesson_solutions

META-INF

users

WEB-INF

attachments

classes

lib

server-config.wsdd

web.xml

webgoat-class.prope

webgoat-lab.propert

webgoat-owasp.proj

webgoat.properties

webgoat_oracle.sql

webgoat_sqlserver.s

main.jsp

reportBug.jsp

sideWindow.jsp

Vulnerabilities in all files

SQL Injection

114 ResultSet results = statement.executeQuery(query);
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\classes\org\owasp\webgoat\lessons\BlindNumericSqlInjection.java

SQL Injection

High vulnerability level.

Check vulnerability

Confirm

Reject

Line114

Functionjava.sql.Statement.executeQuery(java.lang.String)

QueryGET /update?account_number=1 AND sleep(5) = 1 HTTP/1.1

webgoat_src\WebGoat\WEB-INF\classes
ingSqlInjection.java

= statement.executeQuery(query);

SQL Injection

114 ResultSet results = statement.executeQuery(query);
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\classes\org\owasp\webgoat\lessons\BlindNumericSqlInjection.java

Log Forging

104 System.out.println("Account: " + accountNumber);
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\classes\org\owasp\webgoat\lessons\BlindStringSqlInjection.java

Log Forging

105 System.out.println("Answer : " + answer_results.getString(1));
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\classes\org\owasp\webgoat\lessons\BlindStringSqlInjection.java

SQL Injection

114 ResultSet results = statement.executeQuery(query);
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\classes\org\owasp\webgoat\lessons\BlindStringSqlInjection.java

Hardcoded Password

109 PASSWORD = "Password"
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\classes\org\owasp\webgoat\lessons\Challenge2Screen.java

Log Forging

82 System.out.println(WebGoat118N.get("Command")+ " = [" + helpFile.substring(index, helpFileLen).trim().toLowerCase() + "]);
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\classes\org\owasp\webgoat\lessons\CommandInjection.java

et(s.getUserName()) != null && !new
getUserName()).isClosed()

False

WebgoatContext().getDatabaseDriver()

DatabaseConnectionString().contains

False

(().get(s.getUserName()) == null

je

ating user table " +

False

PT Application Inspector

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

Recommended configuration
<session-config><tracking-mode>COOKIE</tracking-mode></session-config>
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\web.xml

Recommended configuration
<error-page><error-code>400</error-code></error-page>
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\web.xml

Recommended configuration
<session-config><cookie-config><secure>true</secure></cookie-config></session-config>
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\web.xml

Recommended configuration
<security-constraint><user-data-constraint><transport-guarantee>CONFIDENTIAL</transport-guarantee>
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\web.xml

Configuration flaw
278 <session-timeout>2880</session-timeout>
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\web.xml
Value: 2880
Recommended: 15

Recommended configuration
<session-config><cookie-config><http-only>true</http-only></cookie-config></session-config>
D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\web.xml

The vulnerability exists in the file D:\Examples\Webgoat2\webgoat_src\webgoat_src\WebGoat\WEB-INF\web.xml

```
273         the timeout for a particular session dynamically by using
274         HttpSession.getMaxInactiveInterval(). -->
275
276         <session-config>
277             <!-- 2 days -->
278             <session-timeout>2880</session-timeout>
279         </session-config>
```

Анализ сторонних компонентов

14

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

**Vulnerable component**
jQuery
D:\CMS\croogo-1.3.5\app\webroot\js\jquery\jquery.min.js

**Vulnerable component**
High vulnerability level.

ConfirmReject

Component	jQuery
File	D:\CMS\croogo-1.3.5\app\webroot\js\jquery\jquery.min.js

All jQuery version up to 1.6.3 are vulnerable to Cross-Site Scripting. The vulnerability occurs in location.hash function targeted to choose elements, that allows an attacker to conduct a remote attack via JavaScript or HTML code injection into a created tag.

Resources with vulnerabilities

- [CVE-2011-4969](#)

Recommendations
Update the module up to the latest version [from the vendor's site](#).

15

```
<html>
<head>
<title>Remote File Inclusion</title>
</head>
<body>
  <h1>Remote File Inclusion</h1>
  This is a link for the scanners: <a href=?q=file>here</a> <br />
  <h1>Content:</h1>
</php
if (isset($_GET['q']))
{
    include ($_GET['q'] . '.inc');
}
else
```

```
?q=%2F..%2F..
+HTTP/1.1

ity

localhost/www/n
nc');
```


Защита от эксплуатации уязвимостей

16

Positive Technologies Application Inspector

Проект "DEMO"

Результаты сканирования 10 фев 10:46

Межсайтовое выполнение сценариев

Уязвимость среднего уровня

Проверить уязвимость Подтвердить Опровергнуть

Строка 167

Дата сканирования 10.02.2016 10:46:31

Время сканирования 00:07:42

Найдено уязвимостей 59

Просканировано файлов 36

Параметры сканирования

Избранные уязвимости

Все уязвимости 59

Анализ исходного кода 59

Уязвимости по файлам и папкам:

- lego_short
 - cfg
 - core
 - includes
 - languages
 - .htaccess
 - access_admin.php
 - admin.php
 - banner.php
 - change.log
 - index.php
 - poll.php
 - robot.txt

Уязв.

15/

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

```
<?xml version="1.0" encoding="utf-8"?>
<report>
  <vuln>
    <path>/banner.php</path>
    <params>
      <param vulnerable="true">
        <name>banner_x</name>
        <src>REQUEST_POST_ARGS</src>
        <payload>"&gt;&lt;/table&gt;&lt;/form&gt;&lt;/div&gt;&lt;/script&gt;alert(1)&lt;/script&gt;
        </payload>
        <dependencies>
          <dependency>%3C</dependency>
          <dependency>%3E</dependency>
          <dependency>%2F</dependency>
          <dependency>%22</dependency>
          <dependency>%28</dependency>
          <dependency>%29</dependency>
        </dependencies>
      </param>
    </params>
    <entry>C:\sources\lego_short\banner.php</entry>
    <type>Cross-site Scripting</type>
    <function>echo</function>
    <file>C:\sources\lego_short\banner.php</file>
    <lineno>167</lineno>
  </vuln>
</report>
```

length: 31861 lines: 787 Ln: 1 Col: 1 Sel: 0 | 0 Dos/Windows UTF-8-BOM INS

require('C:\sources\lego_short\banner.php')

require('C:\sources\lego_short\cfg\connect.inc.php')

require('C:\sources\lego_short\cfg\tables.inc.php')

require('C:\sources\lego_short\includes\database\mysql.php')

require('C:\sources\lego_short\cfg\general.inc.php')

require('C:\sources\lego_short\cfg\functions.php')

preg_match("/^5\./", mysql_get_server_info(mysql_connect('localhost', 'root', '')))

True False

- 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