Open InfoSec Days

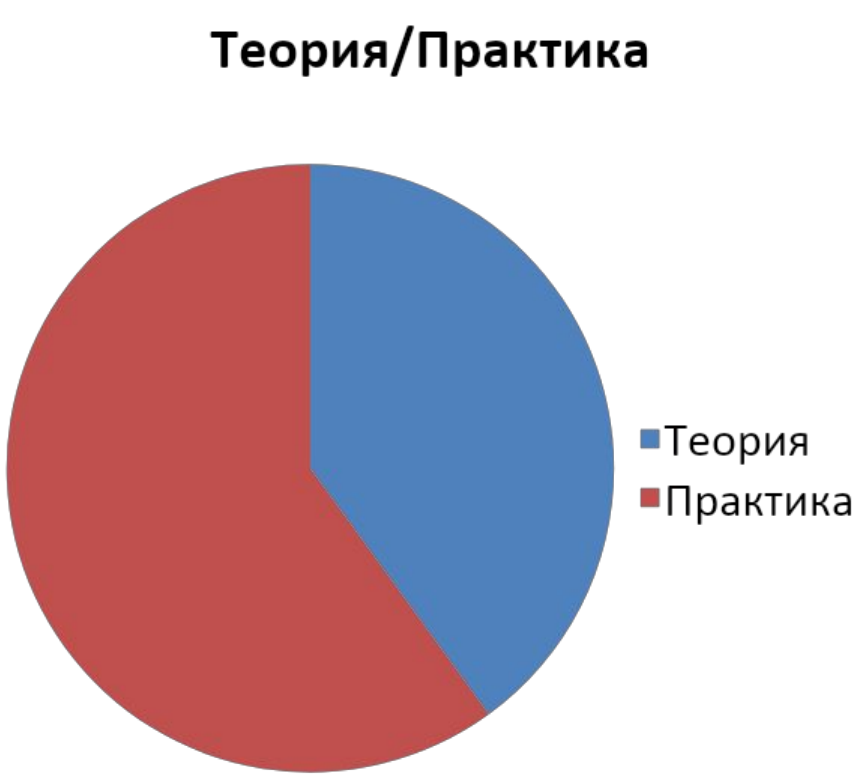
Глава 1. Атаки на веб-приложения и методы защиты

Томск, 2011

Отказ от ответственности

- Информация предоставлена исключительно в ознакомительных целях.
- Всю ответственность за использование и применение полученных знаний каждый участник берет на себя

Содержание курса



Атаки на веб-приложения и методы защиты

1) Cross-Site Scripting
   Что такое XSS, виды XSS, их обнаружение;
   Пример внедрения сниффера и угона cookies;
   Немного о фильтрах.
   key words: post, get, cookies, html, javascript,
   web-dev lang (php)

2) Cross-site request forgery
   Пояснение идеи подделки межсайтовых запросов;
   CSRF через POST & GET, практический пример;
   CSRF + passive XSS -> снова "угоним" cookies;
   Защищаемся с помощью токенов.
   key words: post, get, cookies, html(frame), javascript,
   web-dev lang (php)

3) Remote/Local File Inclusion
   Пример уязвимого кода
   LFI через логи/переменные окружения и др.
   Нулл-байты
   key words: web-dev lang (php)

4) SQL inj
   SQL инъекции как следствие недостаточной фильтрации;
   Последовательный разбор схемы внедрения произвольного запроса;
   Отличие инъекций в разных БД;
   Практика: от инъекции до произвольного выполнения команд на сервере.
   key words: sql, php, unix

5) Denial of Service в веб-приложениях
   Отличия DDoS от DoS;
   DoS сервера БД через инъекцию, практический пример;
   DoS веб-сервера;
   DoS с помощью TCP-flood'инга (?).
   key words: sql, tcp, RFC 2616

6) Обзор софта
   Сканеры брешей в веб-скриптах;
   Утилиты для "раскрутки" sql-инъекций;
   Дебаггеры, локальные прокси;
   key words: web scanner, debug, proxy, sql

Cross Site Scripting (XSS)

- (вики) XSS (англ. Cross Site Scripting — «межсайтовый скриптинг»)
  - тип уязвимости интерактивных информационных систем в вебе. XSS возникает, когда в генерируемые сервером страницы по какой-то причине попадают пользовательские скрипты. Специфика подобных атак заключается в том, что вместо непосредственной атаки сервера они используют уязвимый сервер в качестве средства атаки на клиента.

- Реальные угрозы:
  - Воровство cookie
  - DoS атаки
  - Атаки на браузер пользователя, воровство данных
  - Выполнение произвольных действий на сайте под учетной записью пользователя

Виды XSS

- Пассивные
  - Пассивные XSS подразумевают, что скрипт не хранится на сервере уязвимого сайта, либо он не может автоматически выполниться в браузере жертвы. Для срабатывания пассивной XSS требуется некое дополнительное действие, которое должен выполнить браузер жертвы (например, клик по специально сформированной ссылке). Их также называют первым типом XSS.

- Активные
  - При активных XSS вредоносный скрипт хранится на сервере, и срабатывает в браузере жертвы при открытии какой-либо страницы заражённого сайта. Их также называют вторым типом XSS.

Методы использования

- Более менее-актуальные:
  - Обычный, непосредственная вставка HTML кода
  - Использование DATA (base64)

- Менее актуальные:
  - Flash, Изображения
  - TRACE, UTF-7

Подстановка HTML-кода

Форма:

<input type = “text”>

Введенное значение:

<h1>Hello!</h1>

Должно быть на выходе (с фильтрацией)

<h1>test</h1>

Без фильтрации (xss)

<h1>Hello!</h1>

data:text/html;base64, ...

- Требуется поддержка браузером RFC 2397
- Используется в основном в скриптах перенаправления
- Защита с преобразовыванием символов не спасает от уязвимости (архитектурный баг)

Flash/Изображения

- Flash:
  - Требуеся возможность вставки .swf
  - Использование JS функций

- Изображения
  - Особенность браузеров обрабатывать js код в содержимом картинки (IE)

UTF-7/Trace

- UTF-7
  - Успех зависит от браузера
  - <title> до установки charset
- Trace
  - Зависит от конфигурации сервера