

**Евгений Бондаренко**

Positive Technologies



# Безопасная разработка во фронтенде





# Евгений Бондаренко

---

Тимлид фронтенда PT NAD,  
Positive Technologies



[t.me/fyzlog](https://t.me/fyzlog)

# Обеспечиваем практическую кибербезопасность

**20** лет

опыта исследований  
и разработок

**1,5**<sup>+</sup>  
тыс.

сотрудников: инженеров  
по ИБ, разработчиков,  
аналитиков и других  
специалистов

**250**<sup>+</sup>

экспертов в нашем  
исследовательском  
центре безопасности

**200**<sup>+</sup>

обнаруженных  
уязвимостей нулевого  
дня в год

**250**<sup>+</sup>

аудитов безопасности  
корпоративных систем  
делаем ежегодно

**50**%

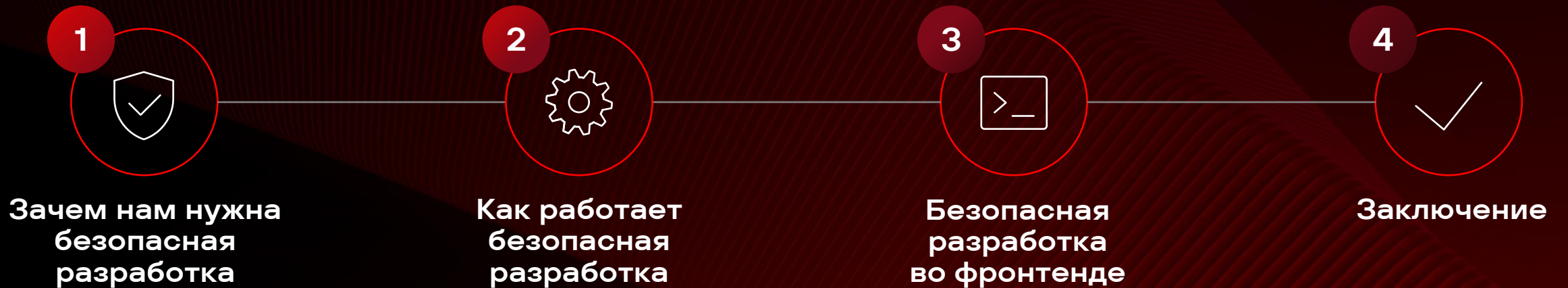
всех уязвимостей  
в промышленности и телекомах  
обнаружили наши эксперты



- Создаем продукты и решения
- Проводим аудиты безопасности
- Расследуем инциденты
- Исследуем угрозы



# Что будет





1

# Зачем нам нужна безопасная разработка

# Анализ защищенности РТ

86%

**Компаний, участвовавших  
в пентестах**

имеют уязвимости в коде  
веб-приложений

63%

**Веб-приложений  
с уязвимостями**

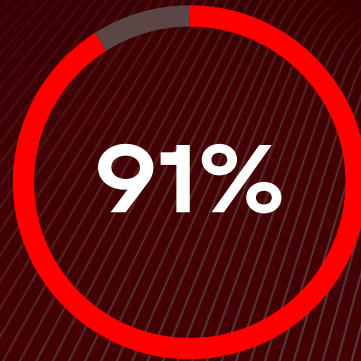
используются для проникновения  
во внутреннюю сеть компании



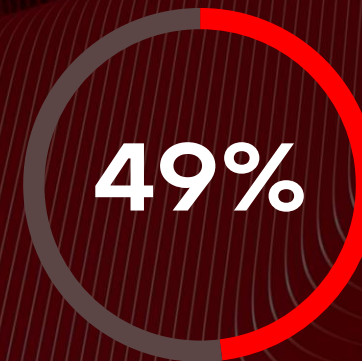
# Уровень безопасности веб-приложений



Веб-приложений  
могут быть  
атакованы

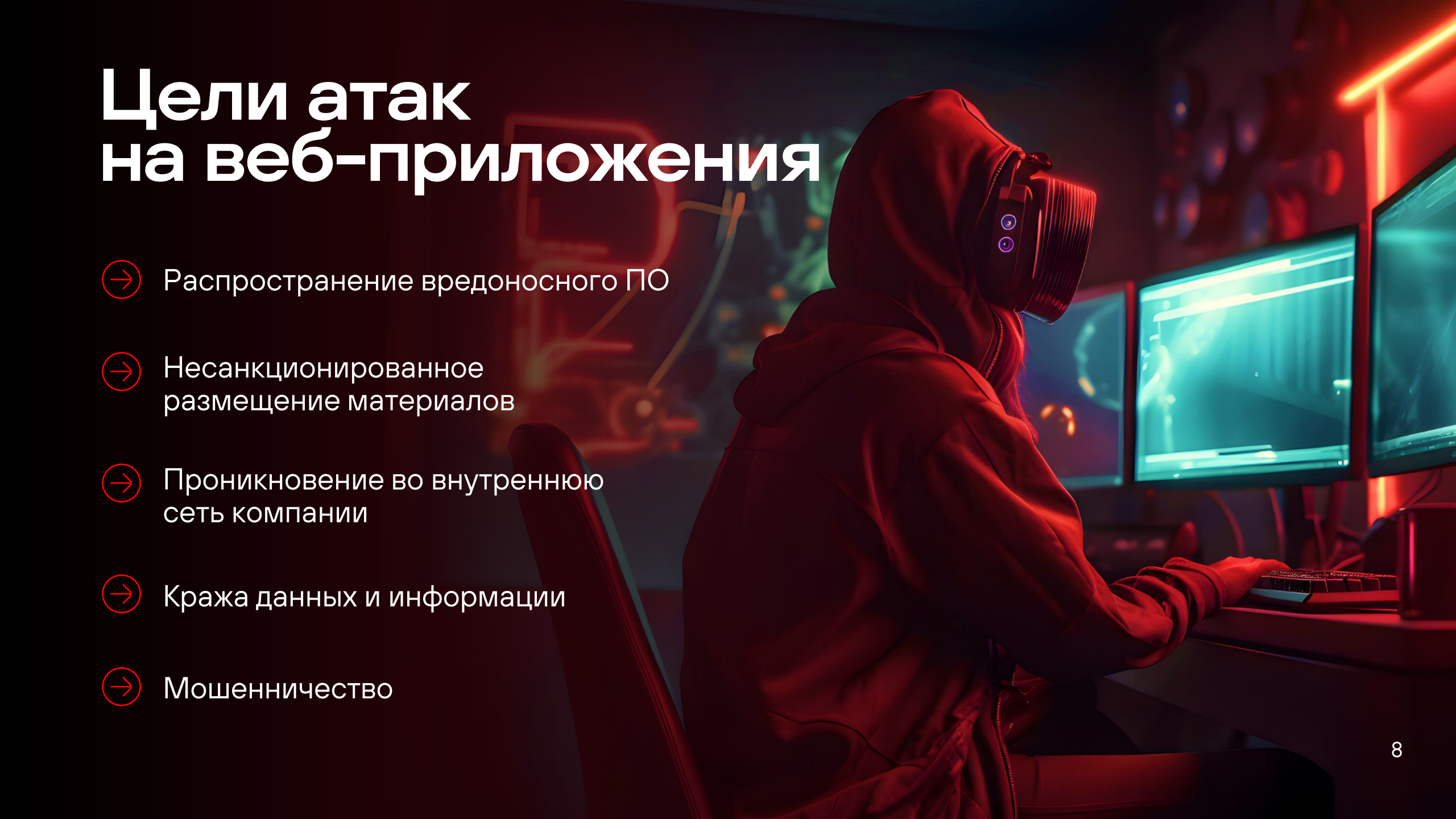


Нарушение  
конфиденциальности  
данных



Низкий  
и экстремально  
низкий уровни  
безопасности

# Цели атак на веб-приложения

A person wearing a dark hoodie and a mask with glowing eyes is sitting at a desk in a dark room. They are looking at several computer monitors that display various data and code. The room is dimly lit with a strong red glow, likely from the monitors or ambient lighting. The person's hands are on a keyboard, and they appear to be focused on their work.

- Распространение вредоносного ПО
- Несанкционированное размещение материалов
- Проникновение во внутреннюю сеть компании
- Кража данных и информации
- Мошенничество



# Зачем безопасная разработка во фронтенде



Уменьшить риски для бизнеса



Снизить затраты на устранение уязвимостей и последствий атак



Повысить устойчивость против атак



Управление инцидентами безопасности и реагированием на них



Внедрение принципов и инструментов повышения надежности во все этапы разработки



в **30-60**  
**раз выше**

стоимость исправления уязвимости в продакшене, чем на этапе проектирования

© NIST

2



# Как работает безопасная разработка





# Безопасная разработка

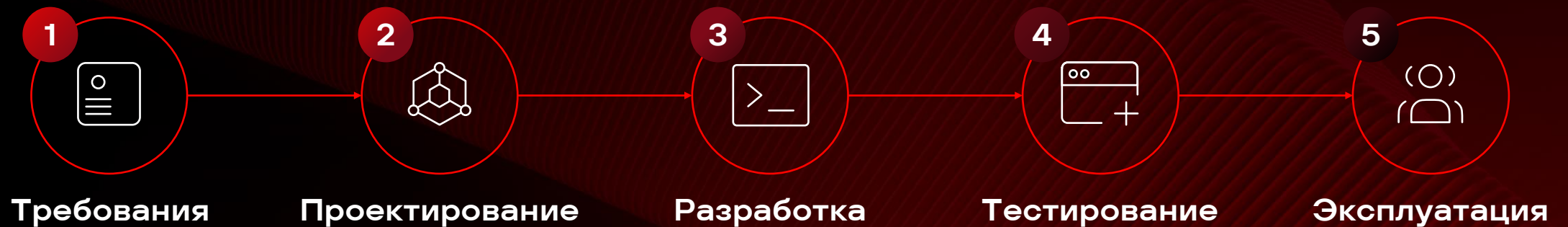
## Методика разработки приложений

- Устойчивость к атакам
- Обеспечение конфиденциальности, целостности и доступности данных
- Уменьшение уязвимостей через устранение багов и логических ошибок в коде

## Принципы

- Отсутствие гарантий безопасности
- Глубинная защита
- Отказобезопасность
- Минимизация полномочий
- Разделение ответственности
- Упрощение механизмов безопасности
- Полное посредничество
- Публичная информация об архитектуре
- Минимизация общих процессов
- Психологическая приемлемость
- Самое слабое место
- Использование существующих компонентов

# Software Development Lifecycle (SDLC)





# Secure SDLC

Требования

Проектирование

Разработка

Тестирование

Эксплуатация

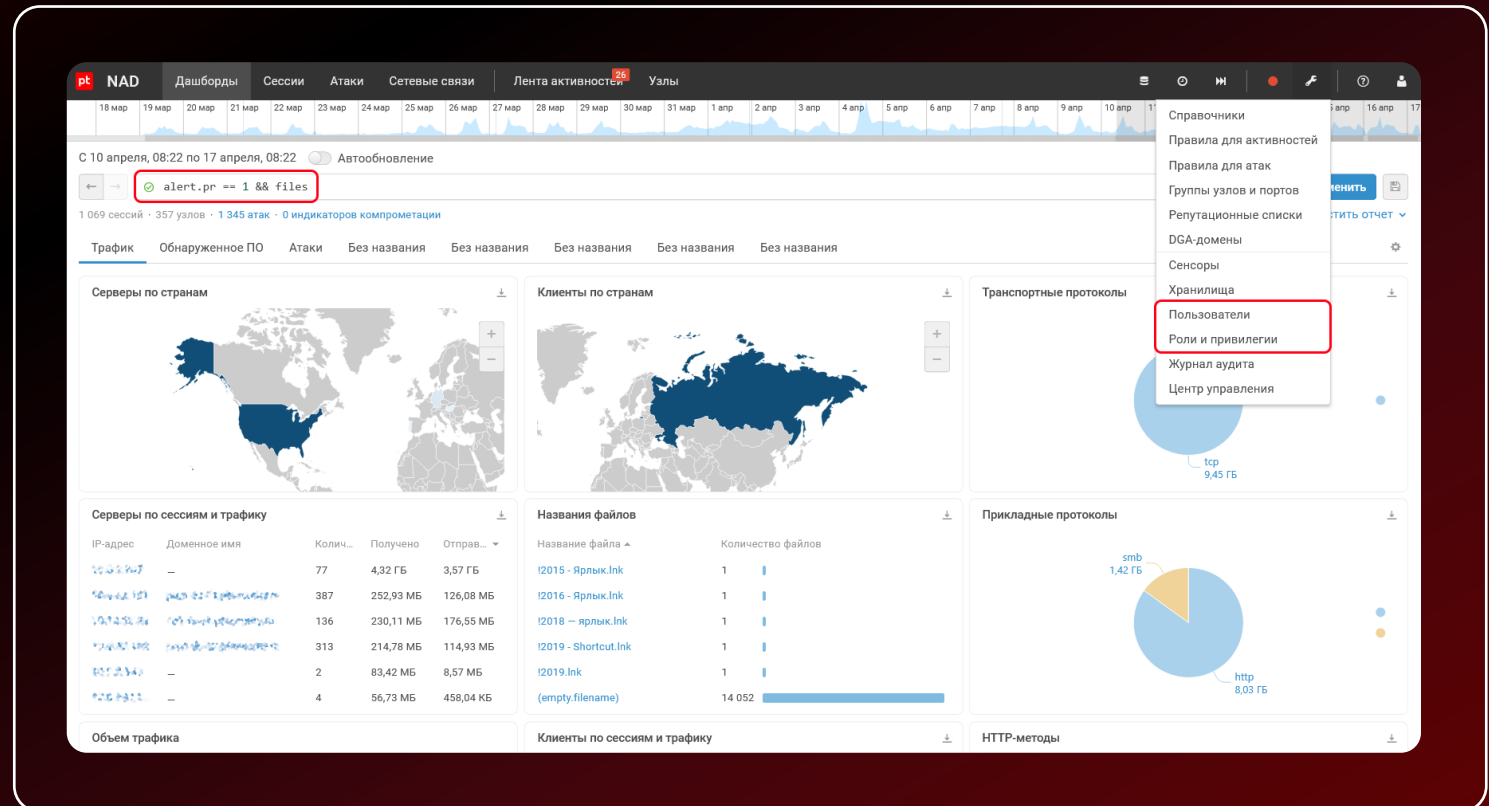
1

## ✓ Требования безопасности

- Какие данные мы храним?
- Как мы передаем данные?
- Есть ли ограничения доступа?

## ! Анализ рисков и угроз

- Небезопасный ввод в фильтр
- Операции с ограниченным доступом



# Secure SDLC

Требования

Проектирование

Разработка

Тестирование

Эксплуатация

2

## ✓ Учет требований безопасности

- Безопасное хранение и передача чувствительных данных
- Разграничение доступа к функциям и данным

## ! Модели рисков и угроз

- Применение фильтра со зловредной нагрузкой
- Получение доступа к недоступным данным и разделам

## 🏗️ Дизайн и ревью архитектуры

The screenshot displays a network monitoring interface with a traffic analysis chart at the top and a role permissions table below. The chart shows traffic volume over time with a search filter 'alert.pr == 1 && files'. The table below is titled 'Роли и привилегии' and lists various permissions for different roles.

Привилегии	Администратор	Оператор	...
<b>Работа с трафиком</b>			
Просмотр общих сведений о трафике	☑	☑	☑
Просмотр подробных сведений о трафике	☑	☑	☑
Просмотр учетных записей в трафике	☑	☑	☑
Экспорт PCAP-файлов	☑	☑	☑
Импорт PCAP-файлов	☑	☑	☑
Перенос захваченного трафика в хранилище	☑	☑	☑
Скачивание файлов, извлеченных из трафика	☑	☑	☑



# Secure SDLC

Требования

Проектирование

Разработка

Тестирование

Эксплуатация

3

## Реализация

- Экранирование и очистка пользовательского ввода
- Проверка прав доступа к разделам, данным и операциям

## Инструменты

- Соглашение о безопасном кодировании
- Ревью кода и решения
- Статический анализ кода
- Анализ зависимостей

The screenshot displays a network security dashboard with the following elements:

- Navigation Bar:** Includes tabs for 'NAD', 'Дашборды', 'Сессии', 'Атаки', 'Сетевые связи', 'Лента активности', and 'Узлы'.
- Time Range:** 'С 11 апреля, 02:43 по 18 апреля, 02:43'.
- Search:** Query: 'alert.pr == 1 && files'.
- Traffic Summary:** 'Общий трафик 8,99 ГБ', 'Отправлено 4,02 ГБ', 'Получено 4,96 ГБ', 'Средняя скорость трафика 14,86 КБ/с'.
- Graphs:** Two line graphs showing traffic volume over time.
- Summary Table:**

Общие сведения	Общие сведения	Атаки
Протоколы	http, tcp	ET INFO Terse Unencrypted Request for Google - Likely Connectivity Check
Приложение	Google	A Network Trojan was Detected
Начало	16 апреля 2024, 20:51:09	
Конец	16 апреля 2024, 20:51:09	
Длительность	445 миллисекунд	
Отправлено	1.55 КБ, 20 пакетов	
Получено	20.64 КБ, 19 пакетов	
Отправитель	H359975	
Получатель	(US) США	
- Files:** Two entries for '(empty.filename).html' (21.57 KB).
- Actions:** 'Отправить в хранилище' and 'Скачать дампы'.

# Secure SDLC

Требования

Проектирование

Разработка

Тестирование

Эксплуатация

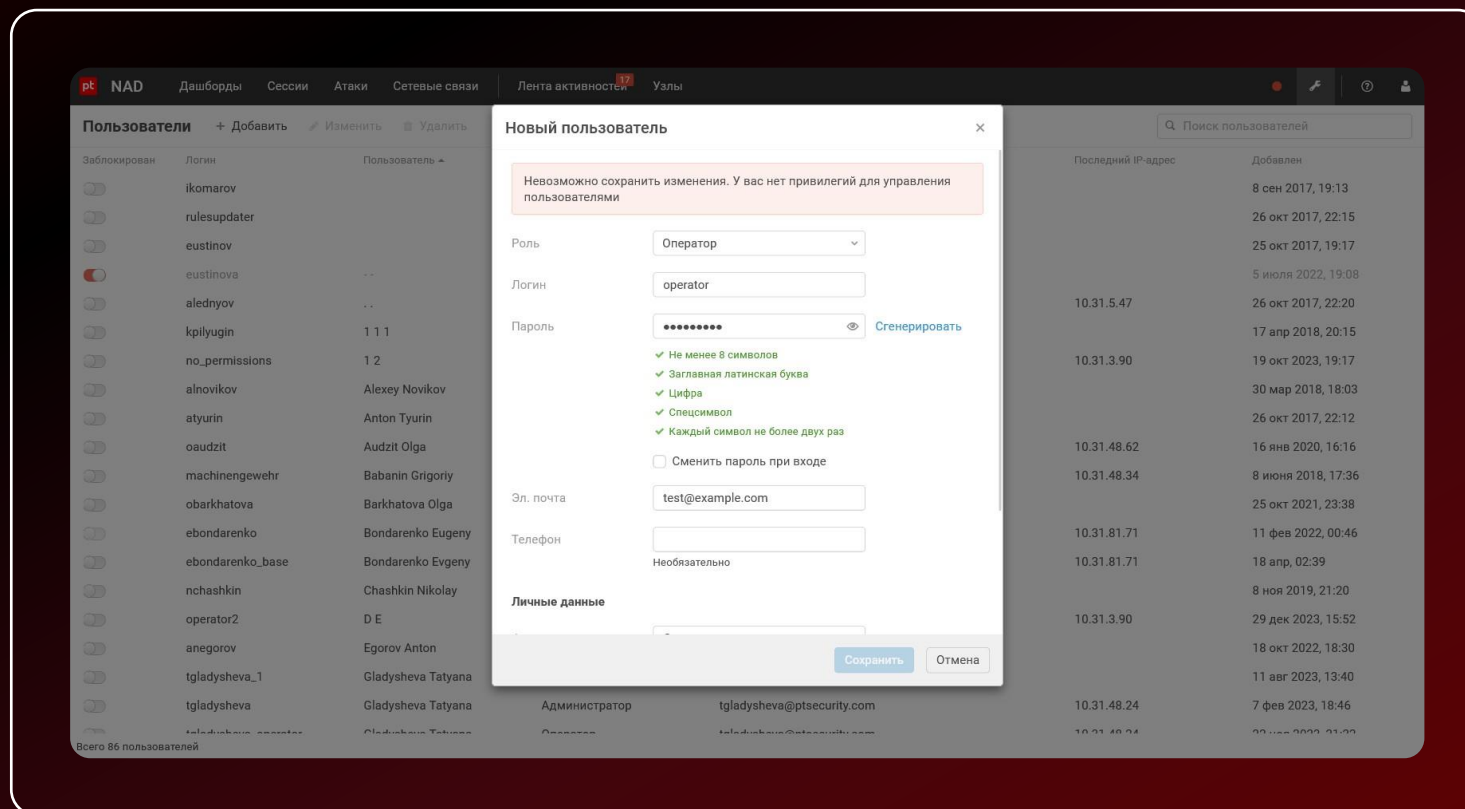
4

## Тестирование угроз

- Ввод зловредной нагрузки в фильтр
- Получение неавторизованного доступа к данным и операциям

## Инструменты

- Динамический анализ кода
- Привлечение пентестеров
- Bug bounty



# Secure SDLC

Требования

Проектирование

Разработка

Тестирование

Эксплуатация

5

- Журналирование, аудит, мониторинг событий безопасности
- Классификация инцидентов
- Реакция на инциденты
- Обработка ошибок и исключений

The screenshot shows a security monitoring interface with a navigation bar at the top containing 'NAD', 'Дашборды', 'Сессии', 'Атаки', 'Сетевые связи', 'Лента активности', and 'Узлы'. Below the navigation bar is a search bar and a filter section. The main content area displays a list of alerts. The first alert is highlighted with a red box and reads: 'Аномальные запросы LDAP' (Anomalous LDAP requests), dated '16 апреля, 21:29 по 17 апреля, 13:29 (16 ч 0 с)'. The description of the alert states: 'Обнаружено сканирование TCP SYN с узла... в рамках которого было открыто 1 376 сессий (50 сессий в секунду). Обнаружены аномальные LDAP запросы с узла...'. Other alerts in the list include 'Новый узел' (New node) and 'ICMP-туннель' (ICMP tunnel).



# Secure SDLC

Требования

Проектирование

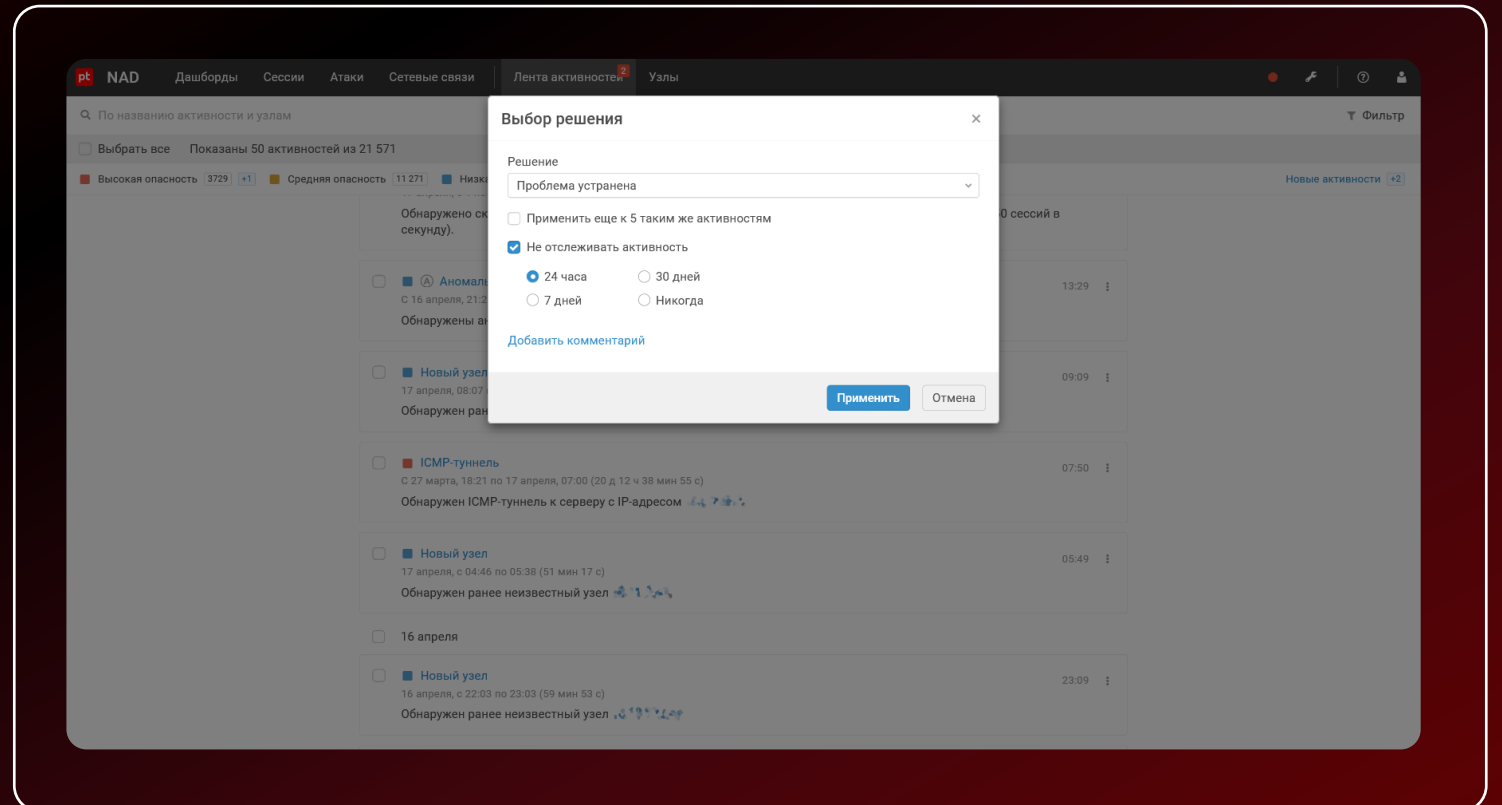
Разработка

Тестирование

Эксплуатация

5

- Журналирование, аудит, мониторинг событий безопасности
- Классификация инцидентов
- Реакция на инциденты
- Обработка ошибок и исключений



# Secure SDLC

Требования

Проектирование

Разработка

Тестирование

Эксплуатация

5

- Журналирование, аудит, мониторинг событий безопасности
- Классификация инцидентов
- Реакция на инциденты
- Обработка ошибок и исключений

The screenshot displays the NAD interface with a navigation bar at the top containing 'Дашборды', 'Сессии', 'Атаки', 'Сетевые связи', 'Лента активности', and 'Узлы'. The main content area shows a list of security events. A red box highlights an event titled 'Аномальные запросы LDAP' (Anomalous LDAP requests) with a green status 'Проблема устранена' (Problem solved). Below this, an 'Аудит' (Audit) section is visible, containing a table of events.

Время события	Пользователь	Действие	Тип объекта	Объект	Результат	Детали
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}
17 апр, 19:08:50	ebondarenko	modify	detection	73828	success	{'identity_key': 'anomaly_ldap_H2091', 'status': 'solved', 'tracking_enabled': false}

Worldwide Security  
**OWASP**  
Open Application Project

## Открытый проект по обеспечению безопасности веб-приложений



Создают инструменты,  
руководства,  
документацию  
по безопасности  
веб-приложений



Направления

- Защита
- Обнаружение
- SSDLC



Разрабатывают  
стандарты



# OWASP Projects



OWASP Top Ten



OWASP Application  
Security Verification  
Standard



OWASP Cheat  
Sheet Series



OWASP Web Security  
Testing Guide



OWASP Juice Shop



# OWASP Top-10

<https://owasp.org/www-project-top-ten/>

- 1 A01:2021-Broken Access Control
- 2 A02:2021-Cryptographic Failures
- 3 A03:2021-Injection
- 4 A04:2021-Insecure Design
- 5 A05:2021-Security Misconfiguration
- 6 A06:2021-Vulnerable and Outdated Components
- 7 A07:2021-Identification and Authentication Failures
- 8 A08:2021-Software and Data Integrity Failures
- 9 A09:2021-Security Logging and Monitoring Failures
- 10 A10:2021-Server-Side Request Forgery



# OWASP Top-10

## Client-Side

<https://owasp.org/www-project-top-10-client-side-security-risks/>

- 1 Broken Client-side Access Control
- 2 DOM-based XSS
- 3 Sensitive Data Leakage
- 4 Vulnerable and Outdated Components
- 5 Lack of Third-party Origin Control
- 6 JavaScript Drift
- 7 Sensitive Data Stored Client-Side
- 8 Client-side Security Logging and Monitoring Failures
- 9 Not Using Standard Browser Security Controls
- 10 Including Proprietary Information on the Client-Side





# OWASP Top-10

## Client-Side

<https://owasp.org/www-project-top-10-client-side-security-risks/>

- 1 Broken Client-side Access Control
- 2 DOM-based XSS
- 3 Sensitive Data Leakage
- 4 Vulnerable and Outdated Components
- 5 Lack of Third-party Origin Control
- 6 JavaScript Drift
- 7 Sensitive Data Stored Client-Side
- 8 Client-side Security Logging and Monitoring Failures
- 9 Not Using Standard Browser Security Controls
- 10 Including Proprietary Information on the Client-Side

3

# Безопасная разработка во фронтенде

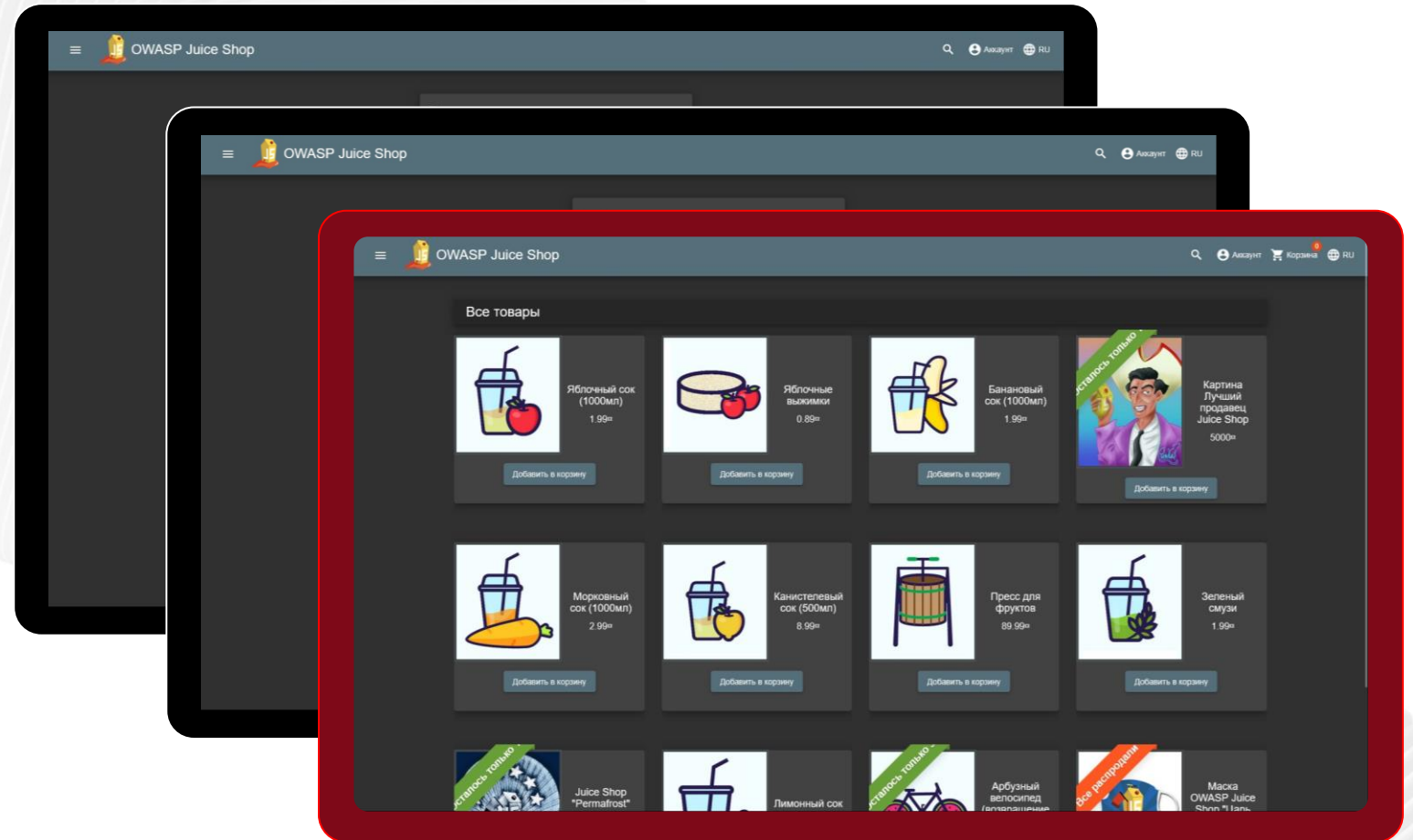
# OWASP Juice Shop

Пример

**небезопасного**  
веб-приложения

→ Angular v15

→ Express 4





# Broken Client-side Access Control



Нарушение  
доступа к данным



Нарушение доступа  
к операциям



Нарушение  
конфиденциаль-  
ности данных



Нарушение  
целостности  
данных

# Broken Client-side Access Control



User



User



User

BID	USER	CART
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7902	FORD	3000
7934	MILLER	1300

# Broken Client-side Access Control



User



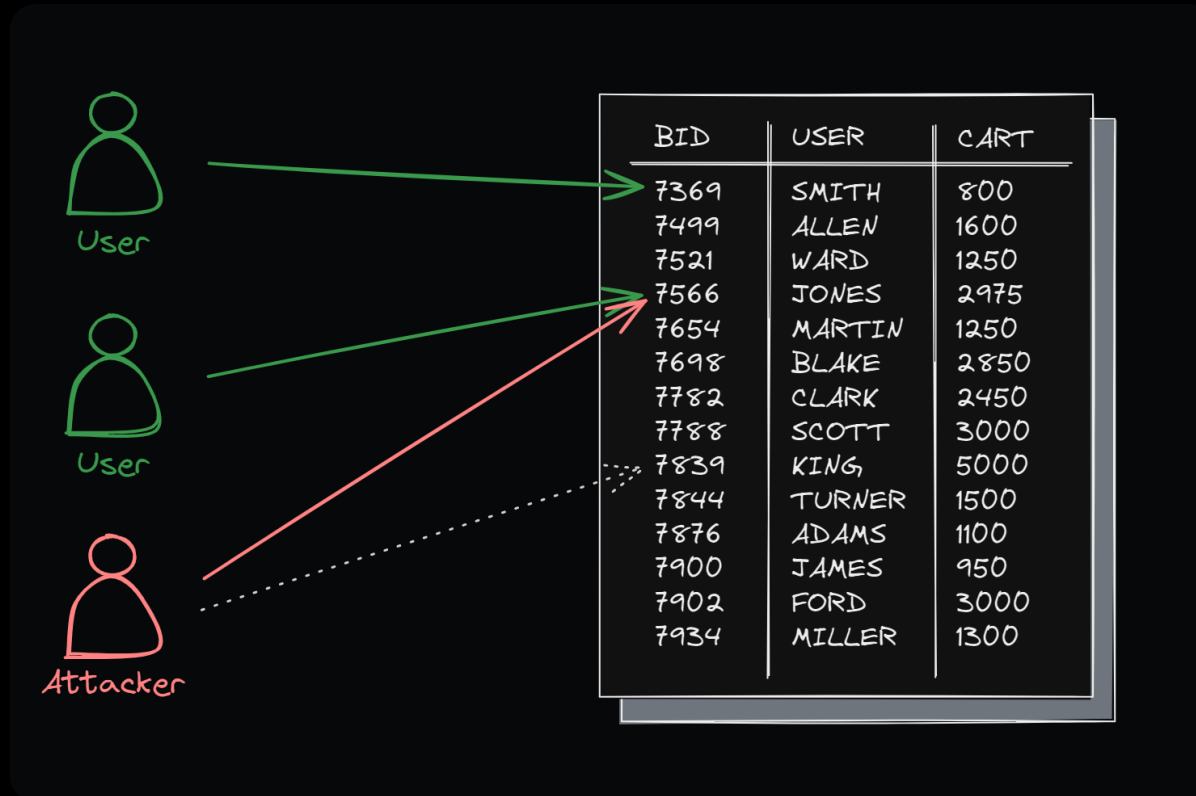
User



User

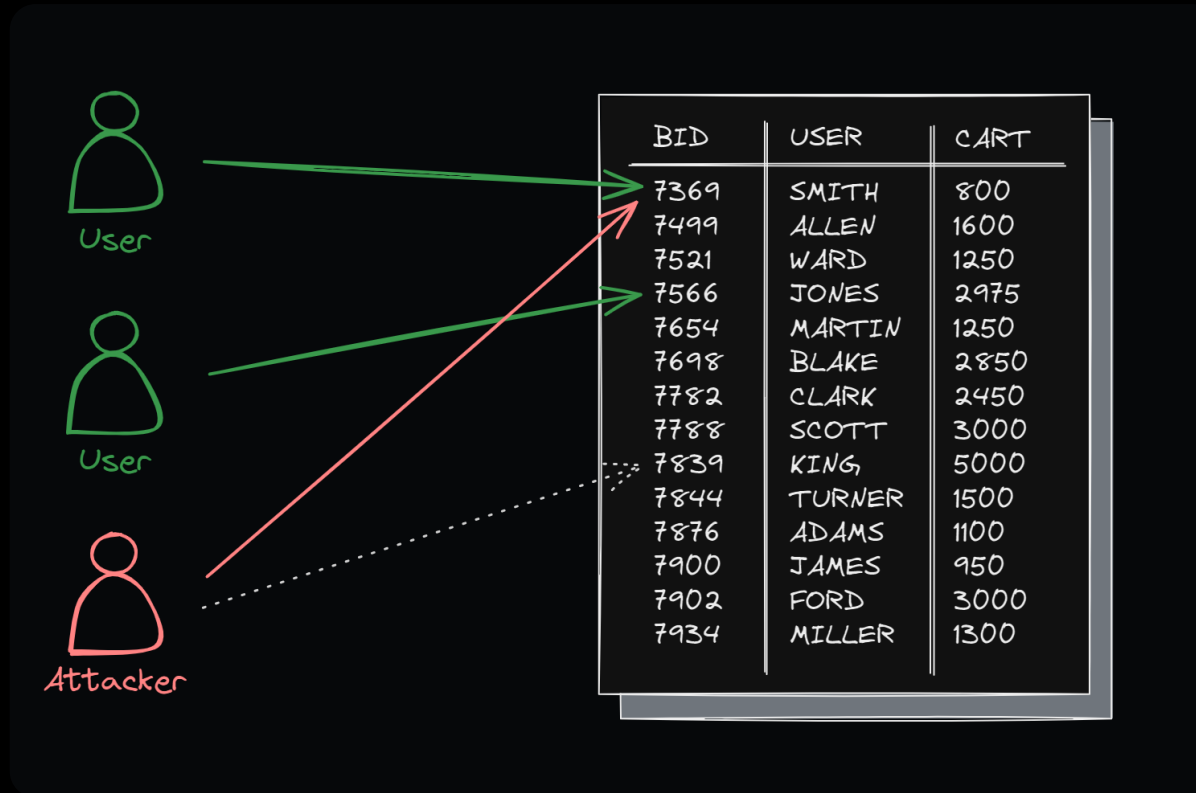
BID	USER	CART
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7902	FORD	3000
7934	MILLER	1300

# Broken Client-side Access Control

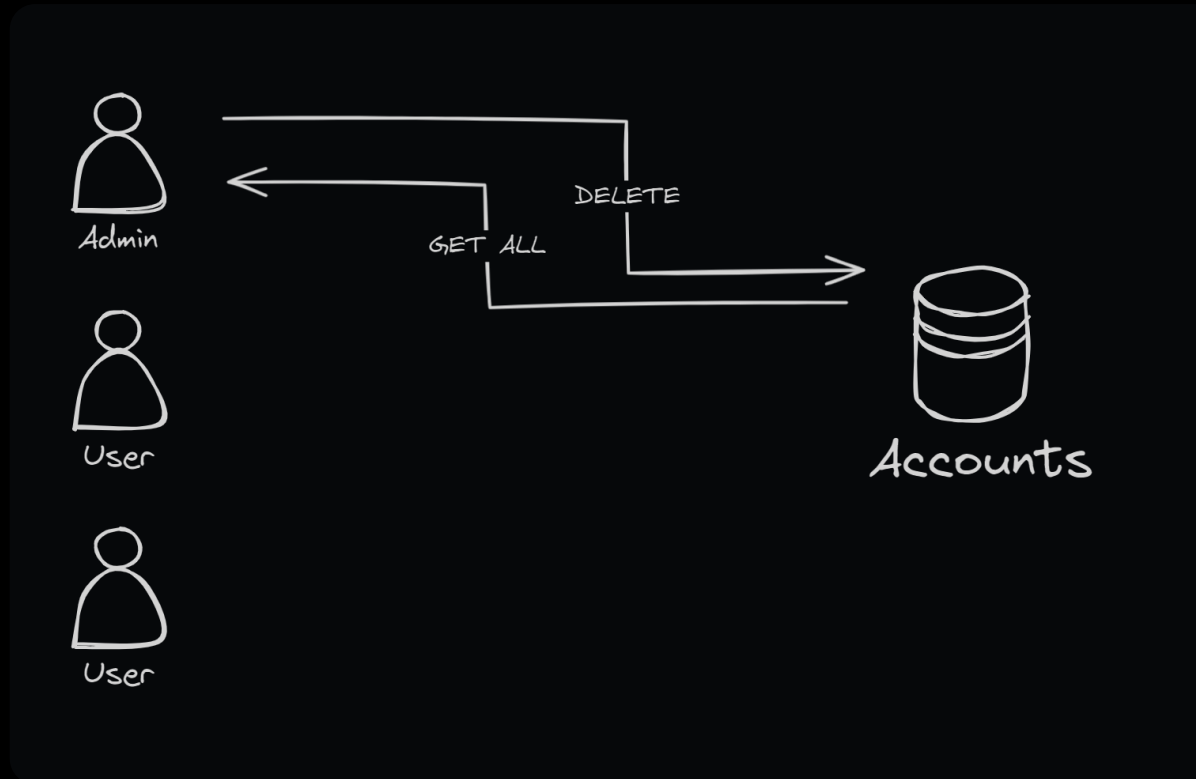




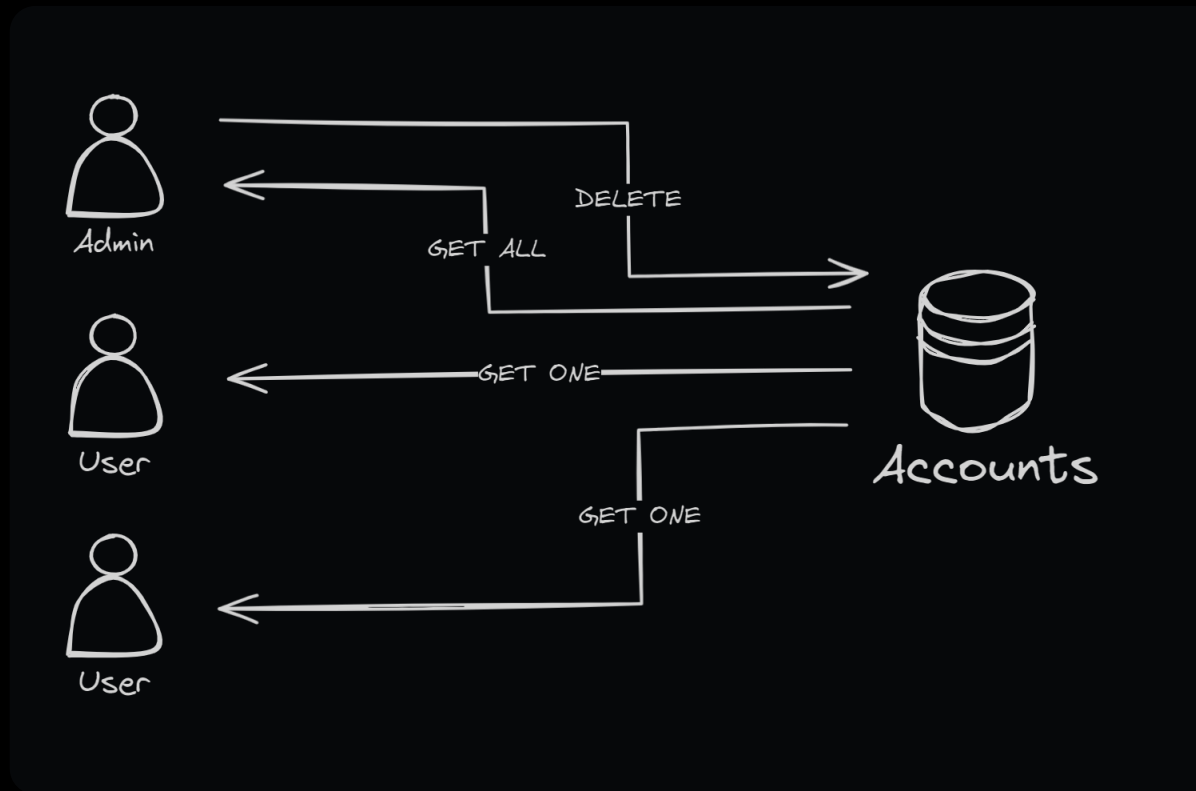
# Broken Client-side Access Control



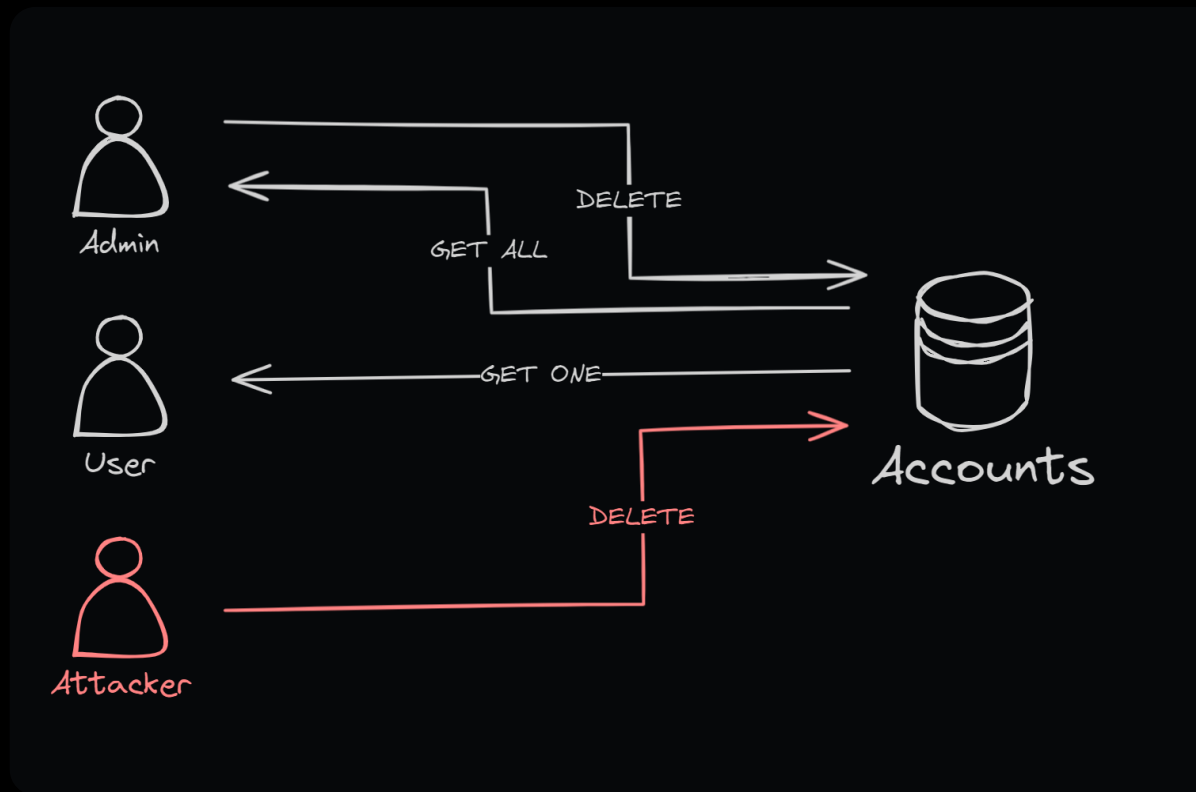
# Broken Client-side Access Control



# Broken Client-side Access Control

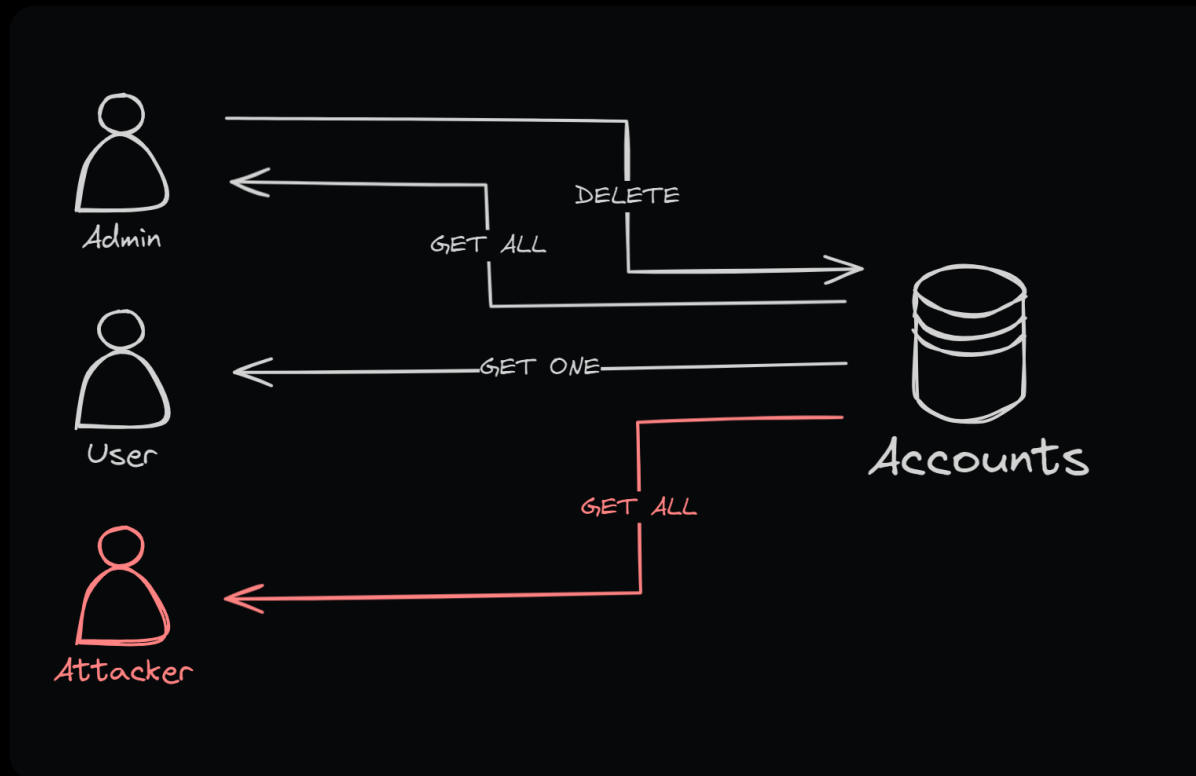


# Broken Client-side Access Control





# Broken Client-side Access Control



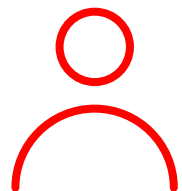
# Broken Client-side Access Control

Требования  
SDLC

1

Проекти-  
рование  
SDLC

Разработка  
SDLC



## Пользователь может

---

- + Добавлять товары в корзину
- + Посмотреть товары в корзине
- + Изменить товары в корзине

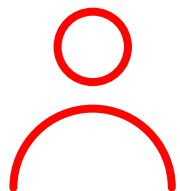
# Broken Client-side Access Control

Требования  
SDLC

1

Проектирование  
SDLC

Разработка  
SDLC



## Пользователь может

- + Добавлять товары в корзину
- + Посмотреть товары в корзине
- + Изменить товары в корзине



## Корзина пользователя

- + Создается при создании новой сессии
- + «Живет» в течение пользовательской сессии
- + Доступна во всех вкладках браузера

# Broken Client-side Access Control

Требования  
SDLC

Проекти-  
рование  
SDLC

Разработка  
SDLC

2



Корзина пользователя  
доступна по адресу  
``/rest/basket/:id``



# Broken Client-side Access Control

Требования  
SDLC

Проектирование  
SDLC

Разработка  
SDLC

2



Корзина пользователя доступна по адресу ``/rest/basket/:id``



Операции GET, ...

# Broken Client-side Access Control

Требования  
SDLC

Проектирование  
SDLC

Разработка  
SDLC

2



Корзина пользователя доступна по адресу ``/rest/basket/:id``



Идентификатор ``bid`` создается при открытии новой сессии



Операции GET, ...

# Broken Client-side Access Control

Требования  
SDLC

Проектирование  
SDLC

Разработка  
SDLC

2



Корзина пользователя доступна по адресу ``/rest/basket/:id``



Идентификатор ``bid`` создается при открытии новой сессии



Операции GET, ...



Идентификатор ``bid`` доступен в сущности ``authentication`` при выполнении операции ``login``

# Broken Client-side Access Control

Требования  
SDLC

Проекти-  
рование  
SDLC

Разработка  
SDLC

3



Операции к API  
реализуем в BasketService  
через HttpClient



# Broken Client-side Access Control

Требования  
SDLC

Проектирование  
SDLC

Разработка  
SDLC

3



Операции к API реализуем в BasketService через HttpClient



Идентификатор `bid` будет браться из SessionStorage

# Broken Client-side Access Control

Требования

SDLC

Проектирование

SDLC

Разработка

SDLC

3



Операции к API реализуем в BasketService через HttpClient



Идентификатор `bid` будет браться из SessionStorage



Идентификатор `bid` будем получать при авторизации и сохранять в SessionStorage

# Разработка (SDLC)

## Broken Client-side Access Control

```
frontend\src\app\login\login.component.ts

this.userService.login(this.user).subscribe((authentication: any) => {
  ...
  sessionStorage.setItem('bid', authentication.bid)
  ...
}, ...)
```

# Разработка (SDLC)

## Broken Client-side Access Control

```
frontend\src\app\purchase-basket\purchase-basket.component.ts

this.basketService
  .find(parseInt(sessionStorage.getItem('bid'), 10))
  .subscribe((basket) => {
    ...
  })
```

# Разработка (SDLC)

## Broken Client-side Access Control

```
frontend\src\app\Services\basket.service.ts





export class BasketService {
  ...
  find (id?: number) {
    return this.http.get(`${this.hostServer}/rest/basket/${id}`).pipe(
      map((response: any) => response.data),
      catchError((error) => { throw error })))
  }
  ...
}
```



# Juice Shop: View Basket

## Broken Client-side Access Control

The screenshot displays the 'View Basket' page in the OWASP Juice Shop application. The page title is 'Корзина (test@example.com)'. The cart contains two items:

Image	Item Name	Quantity	Price	Action
	Juice Shop "Permafrost" 2020 Edition	1	9999.99₽	
	Арбузный велосипед (возвращение версии 2018 года)	1	2999₽	

Итого: 12998.99₽

[Оформить заказ](#)

Вы получите 1300 бонусных очков за этот заказ!

# Juice Shop: View Basket

## Broken Client-side Access Control

The screenshot displays the 'View Basket' page in the OWASP Juice Shop application. The page title is 'Корзина (test@example.com)'. The basket contains two items:

Item	Quantity	Price	Total
Juice Shop "Permafrost" 2020 Edition	1	9999.99₽	9999.99
(Unlabeled Item)	1	2999₽	2999
<b>Итого:</b>			<b>12998.99₽</b>

The developer console is open, showing the 'Storage' tab with the following session storage data:

Key	Value
bid	6
itemTotal	12998.99

The console also shows the URL 'http://localhost:3001' and various developer tools like Inspector, Console, Debugger, Network, Style Editor, Performance, Memory, and Storage.

# Juice Shop: View Basket

## Broken Client-side Access Control

The screenshot shows the OWASP Juice Shop interface. The top navigation bar includes a search icon, 'Аккаунт', 'Корзина' (with a '2' notification), and 'RU'. The main content area is titled 'Корзина (test@example.com)' and displays a list of items in the basket:

Item	Quantity	Price	Action
Juice Shop "Permafrost" 2020 Edition	1	9999.99₽	Remove
	1	2999₽	Remove

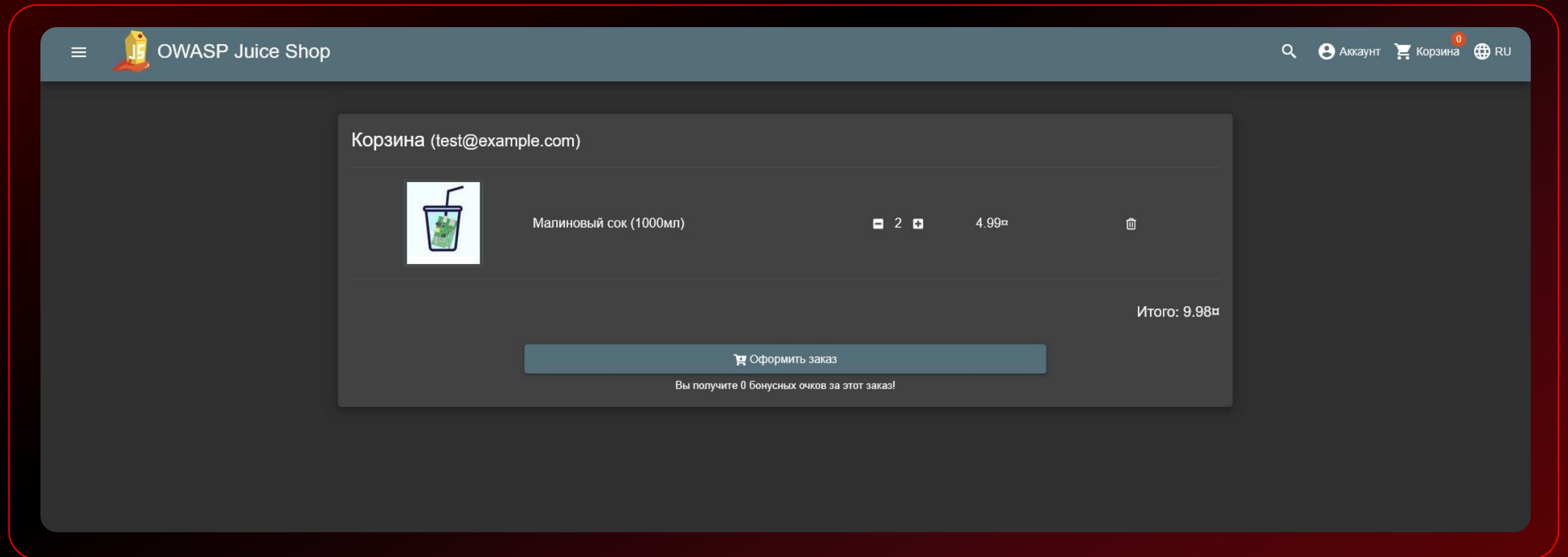
The total price is 'Итого: 12998.99₽'. Below the items, there is a 'Заказать' button and a message: 'Вы получите 100 очков за этот заказ!'.

The developer tools are open, showing the 'Storage' tab. The 'Session Storage' section is expanded, displaying the following data:

Key	Value
bid	4
itemTotal	12998.99

# Juice Shop: View Basket

## Broken Client-side Access Control



# Juice Shop: View Basket

## Broken Client-side Access Control

The screenshot shows the OWASP Juice Shop interface. The top navigation bar includes a menu icon, the site name "OWASP Juice Shop", a search icon, "Аккаунт", "Корзина" (with a red notification badge), and "RU". The main content area displays the basket for "test@example.com". It contains one item: "Малиновый сок (1000мл)" with a quantity of 2 and a price of 4.99. The total price is "Итого: 9.98". A "Заказать" button is visible at the bottom of the basket.

The developer tools are open, showing the "Storage" tab. The "Session Storage" for "http://localhost:3001" is expanded, displaying the following data:

Key	Value
bid	4
itemTotal	9.98





# Cheat Sheet

## Broken Client-side Access Control

[https://cheatsheetseries.owasp.org/cheatsheets/Authorization\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html)



Запрещайте все по умолчанию



Определите границы и права доступов к данным и операциям



Создайте тесты, проверяющие границы и права доступов



Старайтесь не раскрывать уникальные идентификаторы критических данных и объектов



При реализации учитывайте, как работают те или иные механизмы, проверяйте что их использование не нарушает границы и права доступа

# Broken Client-side Access Control

Требования  
SSDLC

1

Проекти-  
рование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC



## Пользователь может

---

- + Добавлять товары в корзину
- + Посмотреть товары в корзине
- Изменить товары в корзине

# Broken Client-side Access Control

Требования  
SSDLC

1

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC



## Пользователь может

- + Добавлять товары в корзину
- + Посмотреть товары в корзине
- Изменить товары в корзине



## Корзина пользователя

- + Создается при создании новой сессии
- + «Живет» в течение пользовательской сессии
- + Доступна во всех вкладках браузера



# Broken Client-side Access Control

Требования  
SSDLC

1

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC



## Пользователь может

- + Добавлять товары в корзину
- + Посмотреть товары в корзине
- Изменить товары в корзине



## Корзина пользователя

- + Создается при создании новой сессии
- + «Живет» в течение пользовательской сессии
- + Доступна во всех вкладках браузера



## Риски

- + Злоумышленник может получить доступ к чужой корзине
- + Злоумышленник может изменить чужую корзину

# Broken Client-side Access Control

Требования  
SSDLC

1

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC



## Пользователь может

- + Добавлять товары в корзину
- + Посмотреть товары в корзине
- Изменить товары в корзине



## Корзина пользователя

- + Создается при создании новой сессии
- + «Живет» в течение пользовательской сессии
- + Доступна во всех вкладках браузера



## Риски

- + Злоумышленник может получить доступ к чужой корзине
- + Злоумышленник может изменить чужую корзину



## Устранение рисков

- Пользователь может иметь доступ только к своей корзине
- Пользователь может изменить только свою корзину



# Broken Client-side Access Control

Требования  
SSDLC

Проекти-  
рование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Идентификатор `bid`  
создается при открытии  
новой сессии

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Идентификатор `bid` создается при открытии новой сессии



Операции GET, ...

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Идентификатор `bid` создается при открытии новой сессии



Операции GET, ...



Корзина пользователя доступна по адресу `/rest/basket/:id`



Корзина пользователя доступна по адресу `/rest/basket`

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Идентификатор `bid` создается при открытии новой сессии



Корзина пользователя доступна по адресу `/rest/basket/:id`



Корзина пользователя доступна по адресу `/rest/basket`



Операции GET, ...



Идентификатор `bid` доступен в сущности `authentication` при выполнении операции `login`



Идентификатор `bid` извлекается из серверной пользовательской сессии при каждой операции в корзине

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Идентификатор `bid` создается при открытии новой сессии



Корзина пользователя доступна по адресу `/rest/basket/:id`



Корзина пользователя доступна по адресу `/rest/basket`



Идентификатор `bid` хранится только в серверной пользовательской сессии



Операции GET, ...



Идентификатор `bid` доступен в сущности `authentication` при выполнении операции `login`



Идентификатор `bid` извлекается из серверной пользовательской сессии при каждой операции в корзине



# Broken Client-side Access Control

Требования  
SSDLC

Проекти-  
рование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

3



Операции к API  
реализуем в BasketService  
через HttpClient

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

3



Операции к API реализуем в BasketService через HttpClient



Идентификатор `bid` будет браться из SessionStorage

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

3



Операции к API реализуем в BasketService через HttpClient



Идентификатор `bid` будет браться из SessionStorage



Идентификатор `bid` будем получать при авторизации и сохранять в SessionStorage

# Разработка (SSDLC)

## Broken Client-side Access Control

```
frontend\src\app\login\login.component.ts

this.userService.login(this.user).subscribe((authentication: any) => {
  ...
-  sessionStorage.setItem('bid', authentication.bid)
  ...
}, ...)
```

# Разработка (SSDLC)

## Broken Client-side Access Control

```
frontend\src\app\purchase-basket\purchase-  
basket.component.ts  
  
this.basketService  
-   .find(parseInt(sessionStorage.getItem('bid'), 10))  
+   .find()  
    .subscribe((basket) => {  
      ...  
    })
```



# Разработка (SSDLC)

## Broken Client-side Access Control

```
frontend\src\app\Services\basket.service.ts

export class BasketService {
  ...
-   find (id?: number) {
+   find () {
-     return this.http.get(`${this.hostServer}/rest/basket/${id}`).pipe(...)
+     return this.http.get(`${this.hostServer}/rest/basket`).pipe(...)
  }
  ...
}
```

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

3

## Ревью MR командами разработки и AppSec

>

### Линтинг

- eslint

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

3

## Ревью MR командами разработки и AppSec

>

### Линтинг

- eslint

>

### Статический анализ кода

- SonarQube
- PT Application Inspector

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

3

## Ревью MR командами разработки и AppSec

>

### Линтинг

- eslint

>

### Статический анализ кода

- SonarQube
- PT Application Inspector

>

### Проверка зависимостей

- npm audit

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

4



Тесты для проверки получения корзины другого пользователя через `/rest/basket``



# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

4



Тесты для проверки получения корзины другого пользователя через `/rest/basket``



Тесты для проверки возможности получить корзину другого пользователя другими методами

# Broken Client-side Access Control

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

4



Тесты для проверки получения корзины другого пользователя через `/rest/basket``



Тесты для проверки возможности получить корзину другого пользователя другими методами



Динамический анализ кода:

- OWASP ZAP
- PT BlackBox

# DOM- based XSS

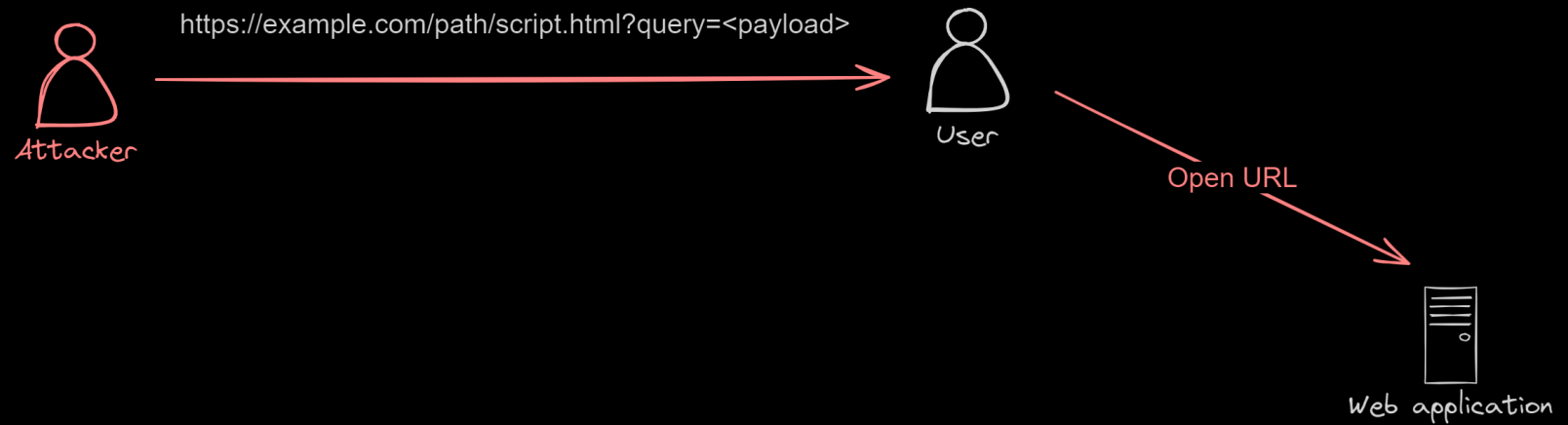
**Type-0 XSS**

- Зловредный код выполняется ТОЛЬКО на клиенте
- Есть доступ к сессии, данным на странице и т. п.
- Зловредная нагрузка не всегда доходит до сервера

# DOM-based XSS

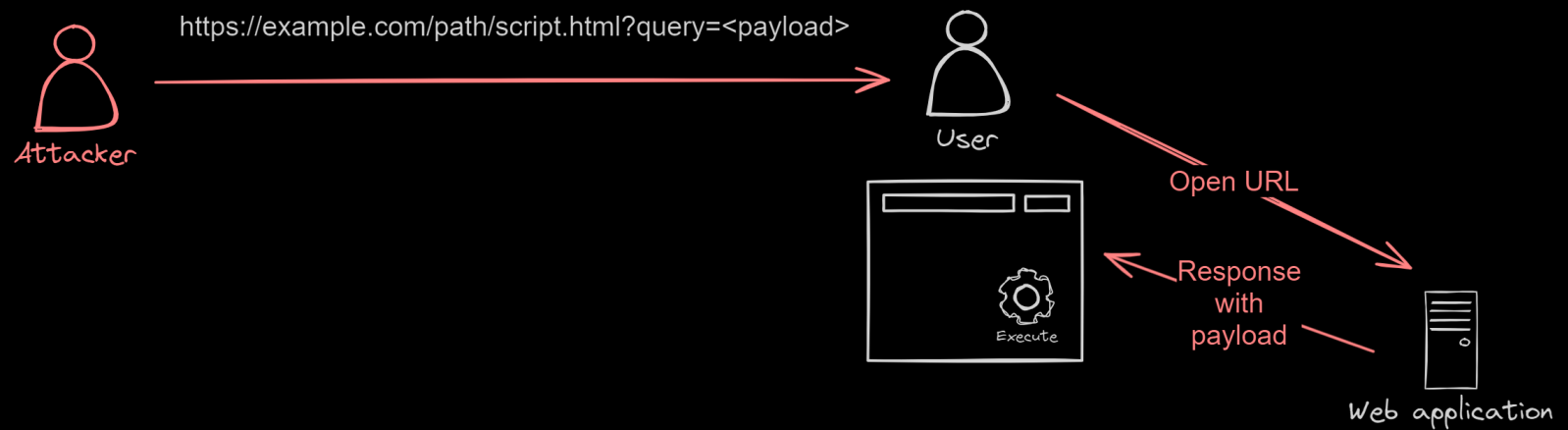


# DOM-based XSS

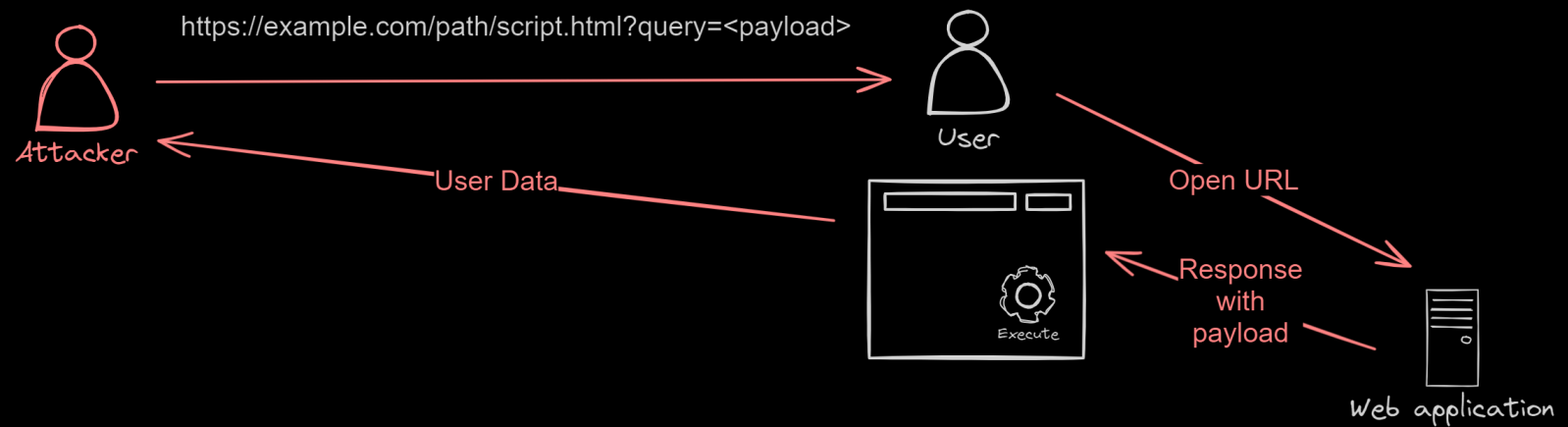




# DOM-based XSS



# DOM-based XSS



# DOM-based XSS

Требования  
SDLC

1

Проектирование  
SDLC

Разработка  
SDLC



Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

# DOM-based XSS

Требования

SDLC

1

Проектирование

SDLC

Разработка

SDLC

>

Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

>

Пользователь должен видеть свой поисковый запрос на странице результатов поиска

# DOM-based XSS

Требования

SDLC

1

Проектирование

SDLC

Разработка

SDLC

>

Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

>

Пользователь должен видеть свой поисковый запрос на странице результатов поиска

>

Поисковый запрос также должен быть добавлен в URL, чтобы пользователь мог сохранить результаты поиска или поделиться ими



# DOM- based XSS

Требования  
SDLC

Проекти-  
рование  
SDLC

Разработка  
SDLC

2



Получаем поисковый запрос  
из поля поиска, применяем  
к загруженным данным из BE  
и отображаем результаты

# DOM-based XSS

Требования  
SDLC

Проектирование  
SDLC

Разработка  
SDLC

2



Получаем поисковый запрос из поля поиска, применяем к загруженным данным из BE и отображаем результаты



На странице результатов отображаем поисковый запрос

# DOM- based XSS

Требования  
SDLC

Проекти-  
рование  
SDLC

Разработка  
SDLC

3



Фильтруем и отображаем  
товары

# DOM-based XSS

Требования  
SDLC

Проектирование  
SDLC

Разработка  
SDLC

3



Фильтруем и отображаем  
товары



Отображаем поисковый  
запрос

# DOM-based XSS

Требования  
SDLC

Проектирование  
SDLC

Разработка  
SDLC

3



Фильтруем и отображаем  
товары



Отображаем поисковый  
запрос



Подставляем его в URL  
в параметр `q`

# Разработка (SDLC)

## DOM-based XSS

```
frontend\src\app\search-result\search-result.component.ts

let queryParam: string = this.route.snapshot.queryParams.q
if (queryParam) {
  queryParam = queryParam.trim()
  this.searchValue = this.sanitizer.bypassSecurityTrustHtml(queryParam)
  ...
}
```



# Разработка (SDLC)

## DOM-based XSS

```
frontend\src\app\search-result\search-result.component.html

<div *ngIf="searchValue">
  <span>{{"TITLE_SEARCH_RESULTS" | translate}} - </span>
  <span id="searchValue" [innerHTML]="searchValue"></span>
</div>
```

# Juice Shop: Search Results

## DOM-based XSS

The screenshot displays the OWASP Juice Shop search results page. The browser's address bar contains the payload `<script>alert("xss")</script>`. The page header shows the site name "OWASP Juice Shop" and navigation links for "Аккаунт", "Корзина", and "RU". The main content area is titled "Все товары" and lists several items. A modal dialog box is open in the foreground, showing the payload `<script>alert("xss")</script>` and a close button. The background items include "Банановый сок (1000мл)" for 1.99, "Картина Лучший продавец Juice Shop" for 5000, and other items with "Добавить в корзину" buttons.

# Juice Shop: Search Results

## DOM-based XSS

The screenshot shows the OWASP Juice Shop search results page. The browser's address bar displays the URL `http://localhost:3001/#/search`. The page content includes the text "Результаты поиска -" and a search results section. A developer tools window is open, showing the DOM tree. The selected element is a `<span id="searchValue" _ngcontent-clv-c47="">` containing the payload `<script wfd-invisible="true">alert("xss")</script>`. The browser's console shows the alert message "xss".

```
<app-welcome _ngcontent-clv-c139="" _ngghost-clv-c133=""></app-welcome>
<router-outlet _ngcontent-clv-c139=""></router-outlet>
<app-search-result class="ng-star-inserted" _ngghost-clv-c47="">
  <div _ngcontent-clv-c47="" fxlayoutalign="center" style="place-content: stretch center; align-items: stretch; flex-direction: row; box-sizing: border-box; display: flex;">
    <div class="table-container custom-slate" _ngcontent-clv-c47="">
      <div class="heading mat-elevation-z6" _ngcontent-clv-c47="">
        <div class="ng-star-inserted" _ngcontent-clv-c47="">
          <span _ngcontent-clv-c47="">Результаты поиска -</span>
          <span id="searchValue" _ngcontent-clv-c47="">
            <script wfd-invisible="true">alert("xss")</script>
          </span>
        </div>
      </div>
    </div>
  </div>
</app-search-result>
```

# Juice Shop: Search Results

## DOM-based XSS



```
<iframe src="javascript:alert(`xss`)">
```

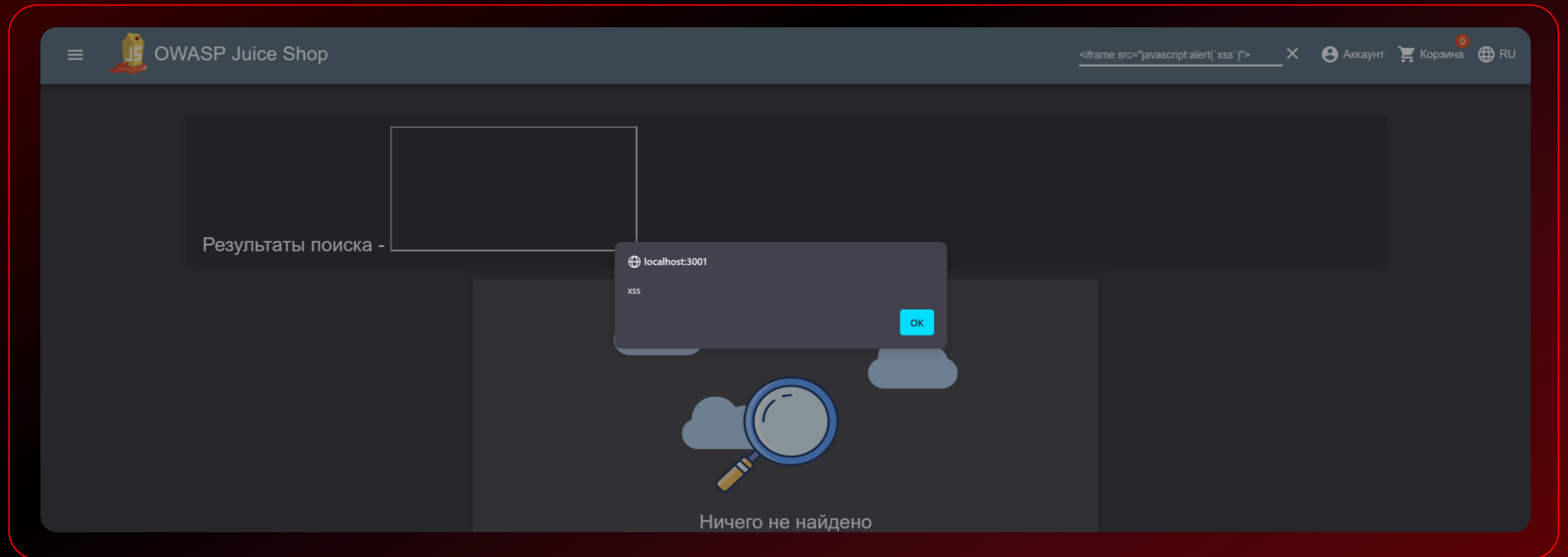
```
http://localhost:3001/#/search?  
q=%3Ciframe%20src%3D%22javascript:alert(%60xss%60)%22%3E
```

```
<img src="" onerror="javascript:alert(`xss`)">
```

```
http://localhost:3001/#/search?  
q=%3Cimg%20src%3D%22%22%20onerror%3D%22javascript:alert(%60xss%60)%22%3E
```

# Juice Shop: Search Results

## DOM-based XSS







# Cheat Sheet

## DOM-based XSS

[https://cheatsheetseries.owasp.org/cheatsheets/DOM\\_based\\_XSS\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html)



Ненадежные  
данные — текст



Очищай и  
экранируй



Не evalируй



Строй DOM сам



Не верь  
небезопасным  
источникам



# DOM-based XSS

Требования

SSDLC

1

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

>

Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

# DOM-based XSS

Требования

SSDLC

1

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

>

Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

>

Риск внедрения зловредной нагрузки через поле поиска и параметр в URL

# DOM-based XSS

Требования

SSDLC

1

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

>

Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

>

Риск внедрения зловредной нагрузки через поле поиска и параметр в URL

>

Пользователь будет видеть свой поисковый запрос на странице результатов поиска

# DOM-based XSS

Требования

SSDLC

1

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

>

Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

>

Риск внедрения зловредной нагрузки через поле поиска и параметр в URL

>

Пользователь будет видеть свой поисковый запрос на странице результатов поиска

>

Необходимо экранировать и очищать небезопасные данные из поля ввода и параметра URL

# DOM-based XSS

Требования

SSDLC

1

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

>

Пользователь может ввести поисковый запрос в строку ввода и посмотреть результаты поиска на отдельной странице

>

Пользователь будет видеть свой поисковый запрос на странице результатов поиска

>

Поисковый запрос также должен быть добавлен в URL, чтобы пользователь мог сохранить результаты поиска или поделиться ими

>

Риск внедрения зловредной нагрузки через поле поиска и параметр в URL

>

Необходимо экранировать и очищать небезопасные данные из поля ввода и параметра URL

# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

2



Получаем поисковый запрос из поля поиска, применяем к загруженным данным из BE и отображаем результаты



# DOM-based XSS

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Получаем поисковый запрос из поля поиска, применяем к загруженным данным из BE и отображаем результаты



Очищать небезопасные данные

# DOM-based XSS

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Получаем поисковый запрос из поля поиска, применяем к загруженным данным из BE и отображаем результаты



На странице результатов отображаем поисковый запрос



Очищать небезопасные данные

# DOM-based XSS

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Получаем поисковый запрос из поля поиска, применяем к загруженным данным из BE и отображаем результаты



На странице результатов отображаем поисковый запрос



Очищать небезопасные данные



Отображать поисковый запрос в виде строки

# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

3

Тестирование

SSDLC



Фильтруем и отображаем  
товары

# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

3

Тестирование

SSDLC



Фильтруем и отображаем  
товары



Очистка данных из поля  
ввода и URL через  
`DomSanitizer.sanitize()`

# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

3

Тестирование

SSDLC



Фильтруем и отображаем товары



Очистка данных из поля ввода и URL через `DomSanitizer.sanitize()`



Отображаем поисковый запрос



# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

3



Фильтруем и отображаем товары



Очистка данных из поля ввода и URL через `DomSanitizer.sanitize()`



Отображаем поисковый запрос



Не использовать `[innerHTML]` и `Element.innerHTML`` для вставки контента

# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

3



Фильтруем и отображаем товары



Очистка данных из поля ввода и URL через `DomSanitizer.sanitize()`



Отображаем поисковый запрос



Не использовать `[innerHTML]` и `Element.innerHTML`` для вставки контента



Подставляем его в URL в параметр ``q``

# Разработка (SSDLC)

## DOM-based XSS

```
frontend\src\app\navbar\navbar.component.ts

search (value: string) {
  if (value) {
-     const queryParams = { queryParams: { q: value } }
+     const queryParams = { queryParams: { q:
this.sanitizer.sanitize(SecurityContext.HTML, value) } }
    this.ngZone.run(async () => await this.router.navigate(['/search'],
queryParams))
  }
}
```

# Разработка (SSDLC)

## DOM-based XSS

```
frontend\src\app\search-result\search-result.component.ts

let queryParams: string = this.route.snapshot.queryParams.q
if (queryParams) {
  queryParams = queryParams.trim()
-  this.searchValue = this.sanitizer.bypassSecurityTrustHtml(queryParams)
+  this.searchValue = this.sanitizer.sanitize(SecurityContext.HTML, queryParams)
  ...
}
```

# Разработка (SSDLC)

## DOM-based XSS

```
frontend\src\app\search-result\search-result.component.html

<div *ngIf="searchValue">
  <span>{{"TITLE_SEARCH_RESULTS" | translate}} - </span>
-   <span id="searchValue" [innerHTML]="searchValue"></span>
+   <span id="searchValue">{{searchValue}}</span>
</div>
```

# Разработка (SSDLC)

## DOM-based XSS



```
-<iframe src="javascript:alert(`xss`)">  
+&lt;/body&gt;
```

```
-<img src="" onerror="javascript:alert(`xss`)">  
+<img src="">
```



# DOM-based XSS

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

4



Проверять поля ввода  
на возможность передать  
зловредную нагрузку

# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

4



Проверять поля ввода на возможность передать зловредную нагрузку



Проверять параметр URL на возможность передачи зловредной нагрузки

# DOM-based XSS

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

4



Проверять поля ввода на возможность передать зловредную нагрузку



Проверять параметр URL на возможность передачи зловредной нагрузки



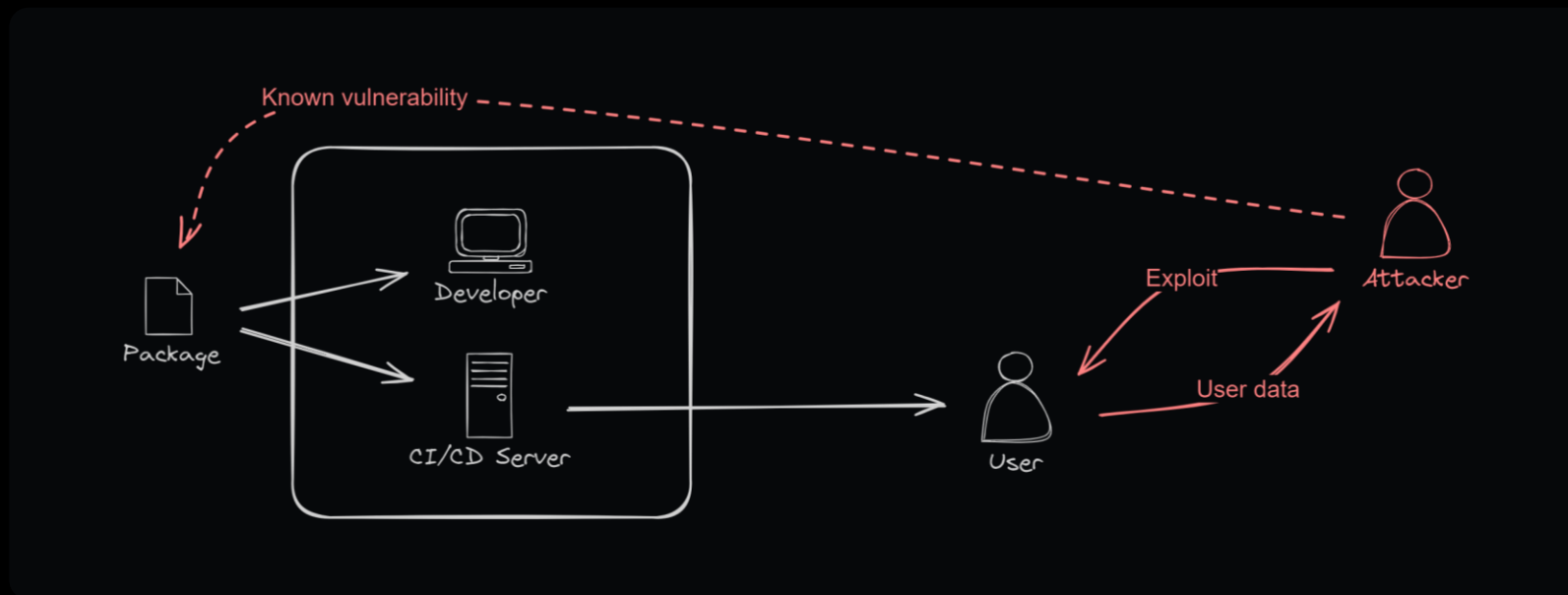
s0md3v/XSSStrike

# Vulnerable and Outdated Components

→ Повышение рисков  
и угрозы атак

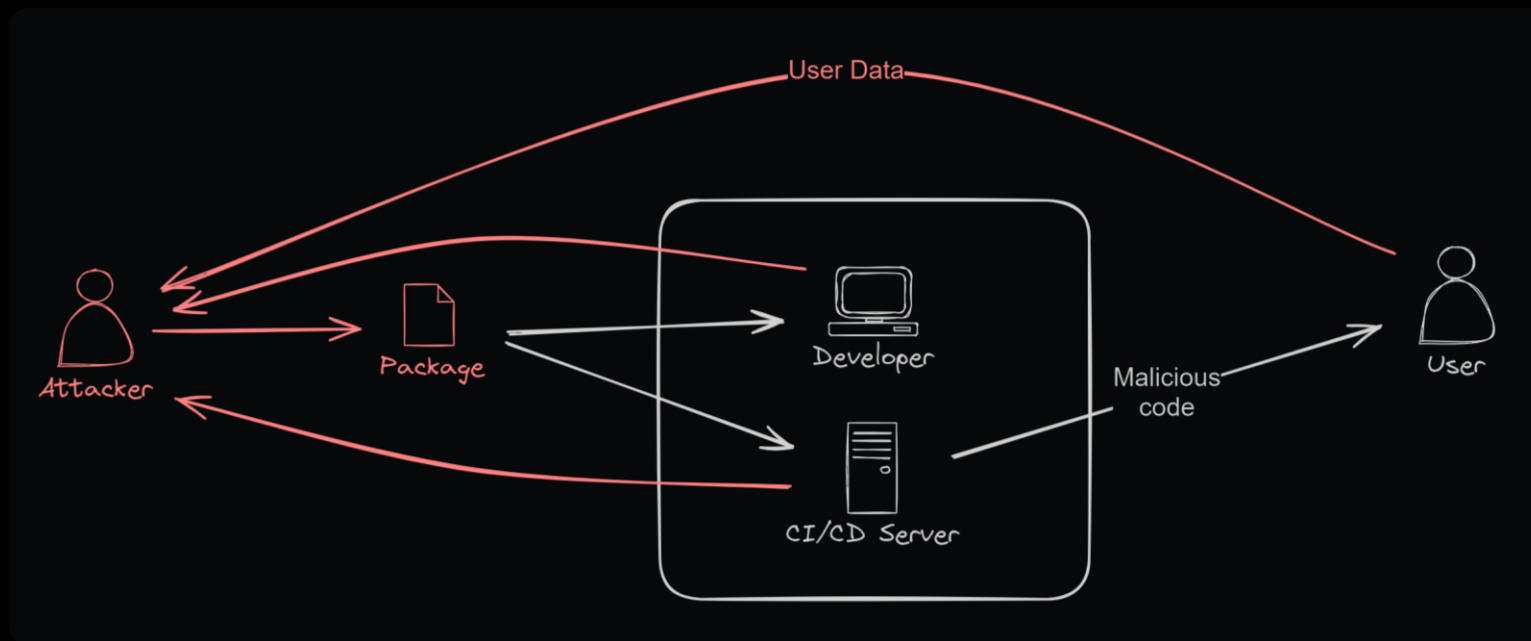
→ Угроза процессам  
сборки и поставки

# Vulnerable and Outdated Components





# Vulnerable and Outdated Components







# Cheat Sheet

Vulnerable and Outdated Components

[https://cheatsheetseries.owasp.org/cheatsheets/NPM\\_Security\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/NPM_Security_Cheat_Sheet.html)



Удаляйте неиспользуемые зависимости



Проверяйте зависимости на уязвимости



Используйте доверенный источник зависимостей



Не храните секреты в репозитории



Не запускайте скрипты хуков пакетного менеджера



Проверяйте зависимости на наличие обновлений

# Vulnerable and Outdated Components

Требования

SSDLC

1

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC



Периодичность проверки уязвимых компонентов

# Vulnerable and Outdated Components

Требования

SSDLC

1

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC



Периодичность проверки уязвимых компонентов



Периодичность обновления устаревших компонентов

# Vulnerable and Outdated Components

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

1



Периодичность проверки уязвимых компонентов



Периодичность обновления устаревших компонентов



Использование достоверного источника компонентов



# Vulnerable and Outdated Components

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

1



Периодичность проверки уязвимых компонентов



Периодичность обновления устаревших компонентов



Использование достоверного источника компонентов



План обновления мажорных версий компонентов

# Vulnerable and Outdated Components

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

1



Периодичность проверки уязвимых компонентов



Периодичность обновления устаревших компонентов



Использование достоверного источника компонентов



План обновления мажорных версий компонентов



Риск эксплуатации известных уязвимостей



# Vulnerable and Outdated Components

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

2



Уязвимые компоненты —  
в начале спринта.  
Обновление с уязвимостями  
medium и выше —  
в течение спринта

# Vulnerable and Outdated Components

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Уязвимые компоненты — в начале спринта.  
Обновление с уязвимостями medium и выше — в течение спринта



Неиспользуемые компоненты — в начале спринта.  
Удаление — в течение спринта

# Vulnerable and Outdated Components

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

2



Уязвимые компоненты — в начале спринта.  
Обновление с уязвимостями medium и выше — в течение спринта



Мажорные версии — раз в полгода



Неиспользуемые компоненты — в начале спринта.  
Удаление — в течение спринта

# Vulnerable and Outdated Components

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

2



Уязвимые компоненты — в начале спринта.  
Обновление с уязвимостями medium и выше — в течение спринта



Мажорные версии — раз в полгода



Неиспользуемые компоненты — в начале спринта.  
Удаление — в течение спринта



Минорные версии — раз в квартал

# Vulnerable and Outdated Components

Требования  
SSDLC

Проектирование  
SSDLC

Разработка  
SSDLC

Тестирование  
SSDLC

3



Перейти  
на внутренний  
артефакторий  
компонентов

# Vulnerable and Outdated Components

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

3

Тестирование

SSDLC



Перейти  
на внутренний  
артефакторий  
компонентов



Закрепить  
зависимости



# Vulnerable and Outdated Components

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

3

Тестирование

SSDLC



Перейти  
на внутренний  
артефакторий  
компонентов



Закрепить  
зависимости



Автоматическая  
проверка в CI/CD  
(npm audit, sonarqube)

# Vulnerable and Outdated Components

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

4



Проверка **уязвимых**  
компонентов

# Vulnerable and Outdated Components

Требования

SSDLC

Проектирование

SSDLC

Разработка

SSDLC

Тестирование

SSDLC

4



Проверка **уязвимых**  
компонентов



Проверка **устаревших**  
компонентов

4



# Заключение





Разработка с учетом  
рисков и угроз повышает  
защищенность



Изучение атак и техник  
повышает эффективность  
методов защиты



The background features a series of parallel, wavy red lines that create a sense of depth and movement. These lines are set against a dark gradient that transitions from black on the left to a deep red on the right. The overall effect is modern and dynamic.

**Спасибо!**



# Q&A



[t.me/fyzlog](https://t.me/fyzlog)



[t.me/Positive\\_Frontend](https://t.me/Positive_Frontend)

# Ссылки на ресурсы



<https://www.ptsecurity.com/ru-ru/research/analytics/results-of-pentests-2021-2022/>



[https://www.ptsecurity.com/ru-ru/research/analytics/kiberbezopasnost-v-2023-2024-gg-trendy-i-prognozy-chast-tretya/?sphrase\\_id=301187](https://www.ptsecurity.com/ru-ru/research/analytics/kiberbezopasnost-v-2023-2024-gg-trendy-i-prognozy-chast-tretya/?sphrase_id=301187)



<https://www.ptsecurity.com/ru-ru/research/analytics/web-vulnerabilities-2020-2021/>




<https://www.ptsecurity.com/ru-ru/research/analytics/>

# Ссылки на ресурсы


 <https://owasp.org/>

 <https://owasp.org/www-project-top-ten/>

 <https://owasp.org/www-project-top-10-client-side-security-risks/>

 <https://owasp.org/www-project-application-security-verification-standard/>

# Ссылки на ресурсы

 <https://owasp.org/www-project-cheat-sheets/>

 <https://owasp.org/www-project-web-security-testing-guide/>

 <https://owasp.org/www-project-juice-shop/>

 [https://owasp.org/www-community/Broken\\_Access\\_Control](https://owasp.org/www-community/Broken_Access_Control)

# Ссылки на ресурсы



[https://cheatsheetseries.owasp.org/cheatsheets/Authorization\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html)



<https://github.com/OWASP/www-project-proactive-controls/blob/master/v3/en/c7-enforce-access-controls.md>



<https://github.com/OWASP/ASVS/blob/master/4.0/en/0x12-V4-Access-Control.md>





[https://owasp.org/www-project-web-security-testing-guide/v42/4-Web\\_Application\\_Security\\_Testing/05-Authorization\\_Testing/](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/05-Authorization_Testing/)



# Ссылки на ресурсы

 [https://owasp.org/www-community/attacks/DOM\\_Based\\_XSS](https://owasp.org/www-community/attacks/DOM_Based_XSS)


 [https://cheatsheetseries.owasp.org/cheatsheets/DOM\\_based\\_XSS\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html)


 [https://owasp.org/www-project-web-security-testing-guide/v42/4-Web\\_Application\\_Security\\_Testing/11-Client-side\\_Testing/01-Testing\\_for\\_DOM-based\\_Cross\\_Site\\_Scripting](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/11-Client-side_Testing/01-Testing_for_DOM-based_Cross_Site_Scripting)

 <https://github.com/wisec/domxsswiki/wiki/>

# Ссылки на ресурсы

 <https://github.com/payloadbox/xss-payload-list>


 [https://cheatsheetseries.owasp.org/cheatsheets/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html)


 <https://github.com/Edr4/XSS-Bypass-Filters>

 <https://gist.github.com/rvrsh3ll/09a8b933291f9f98e8ec>

# Ссылки на ресурсы

 <https://xss-game.appspot.com/>

 <http://www.xssgame.com/>

 [https://owasp.org/Top10/A06\\_2021-Vulnerable\\_and\\_Outdated\\_Components/](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)

 [https://owasp.org/www-project-top-10-insider-threats/docs/2023/INT01\\_2023-Outdated\\_Software](https://owasp.org/www-project-top-10-insider-threats/docs/2023/INT01_2023-Outdated_Software)