**Slide 1. Название презентации**

Обзор способов и протоколов аутентификации в веб-приложениях

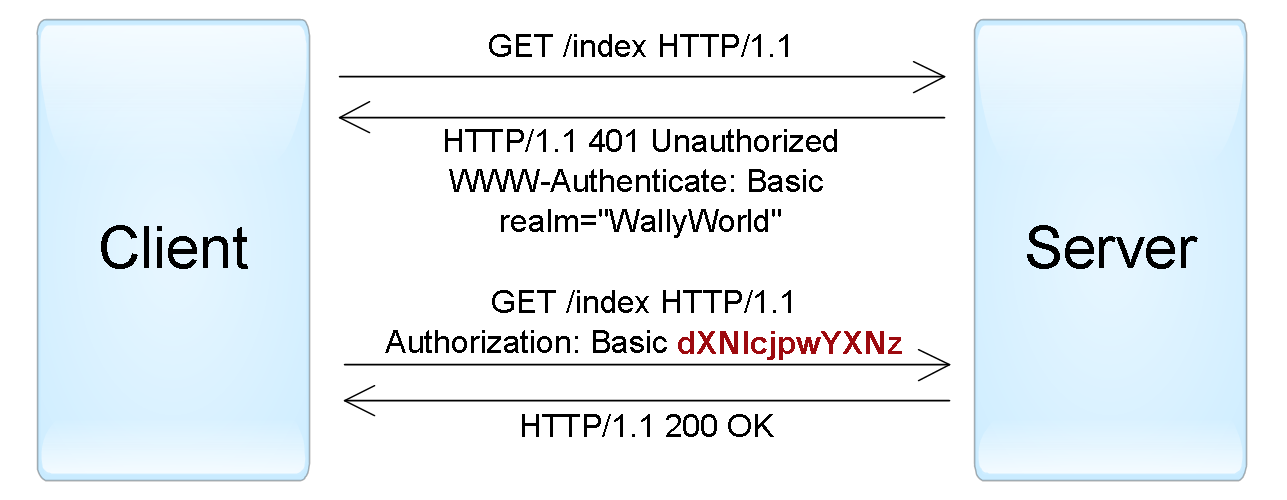
**Slide 2. Основные термины**

Идентификация

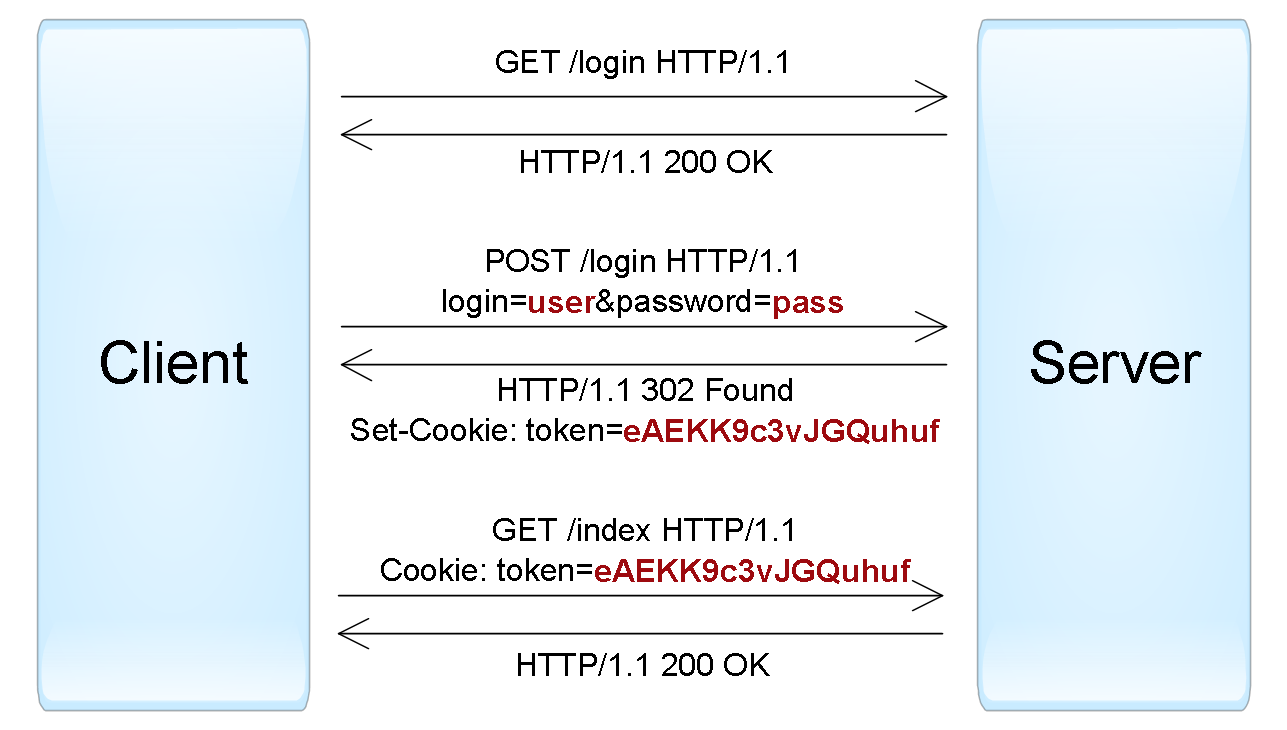
Аутентификация

Авторизация

**Slide 3. Виды аутентификации**

1. Аутентификация по паролю:
   1. HTTP authentication:
      1. Basic
      2. **Digest**
      3. NTML (Active Directory)
      4. **Negotiate** (Active Directory)

Slide 4. Forms authentication:



Slide 5. Другие протоколы аутентификации по паролю.

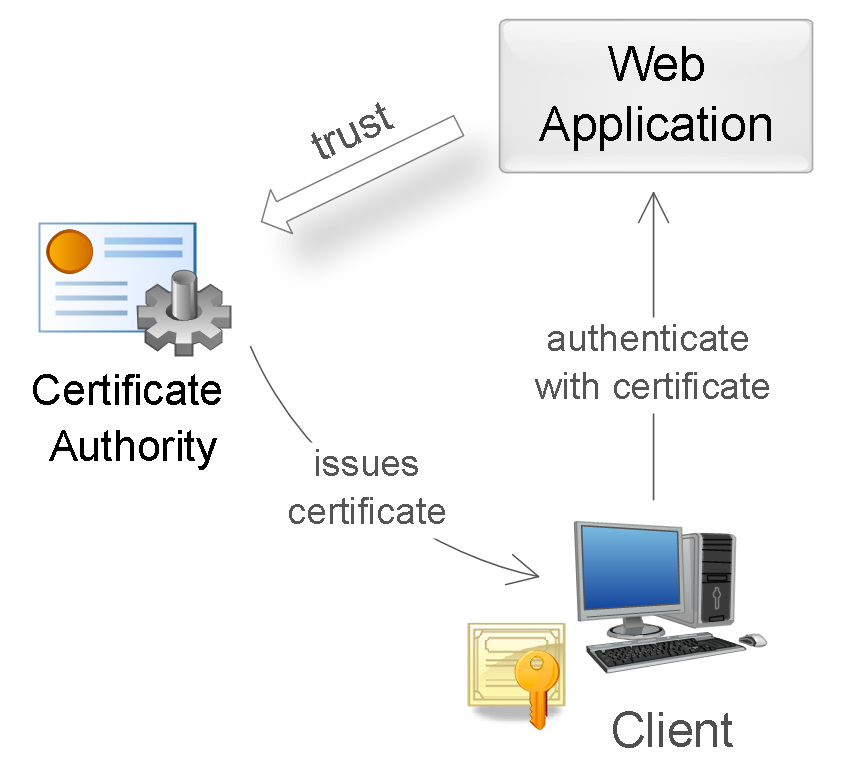
**URL query** — считается небезопасным вариантом, т. к. строки URL могут запоминаться браузерами, прокси и веб-серверами.

**Request body** — безопасный вариант, но он применим только для запросов, содержащих тело сообщения (такие как POST, PUT, PATCH).

**HTTP header** —оптимальный вариант, при этом могут использоваться и стандартный заголовок Authorization (например, с Basic-схемой), и другие произвольные заголовки.

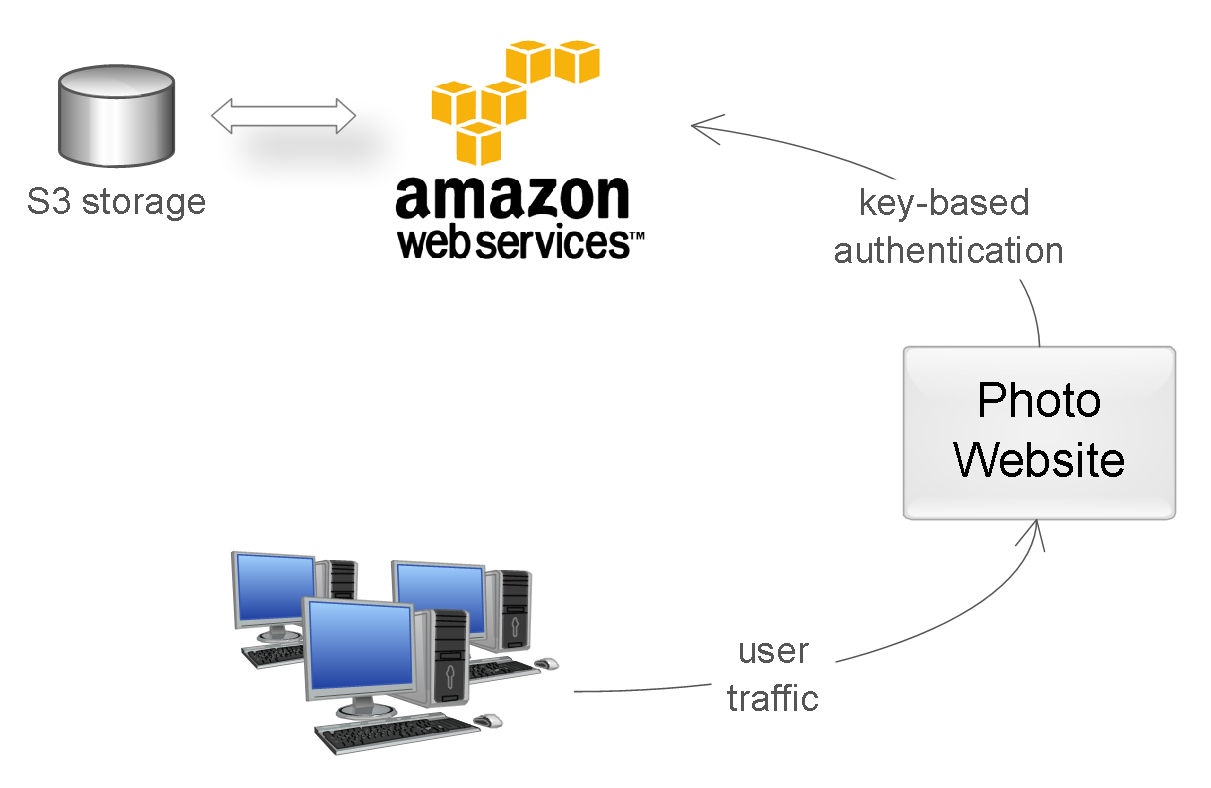
Slide 6. Распространенные уязвимости и ошибки реализации

Slide 7. Аутентификация по сертификатам

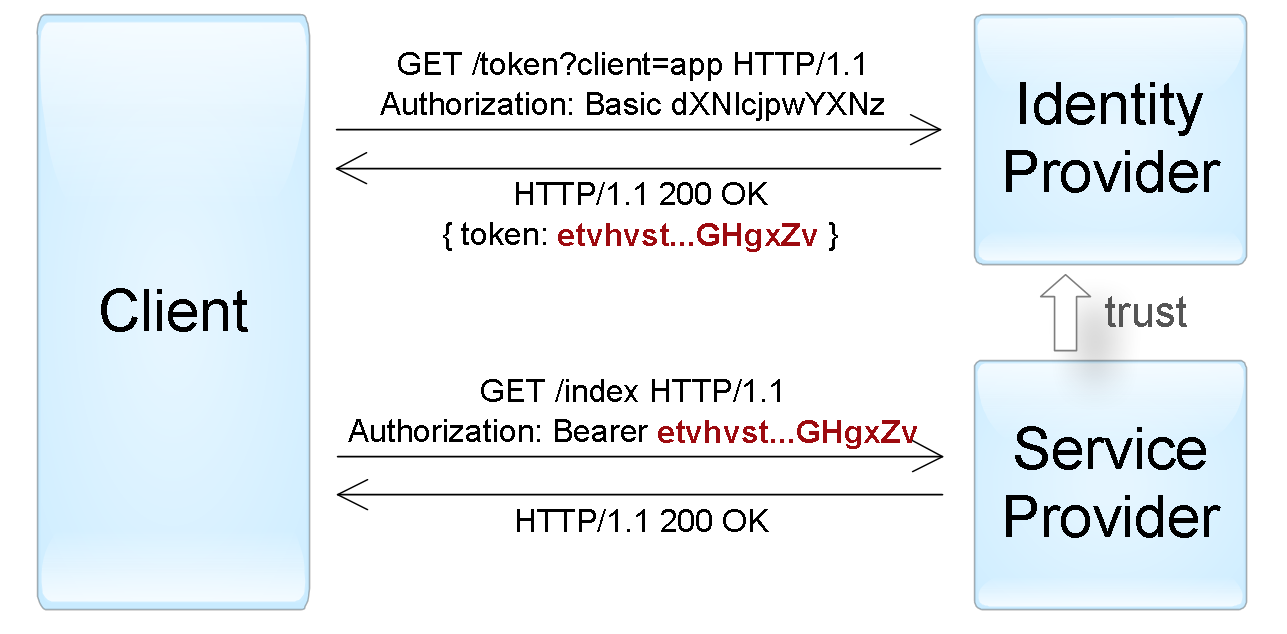


Slide 8. Аутентификация по одноразовым паролям

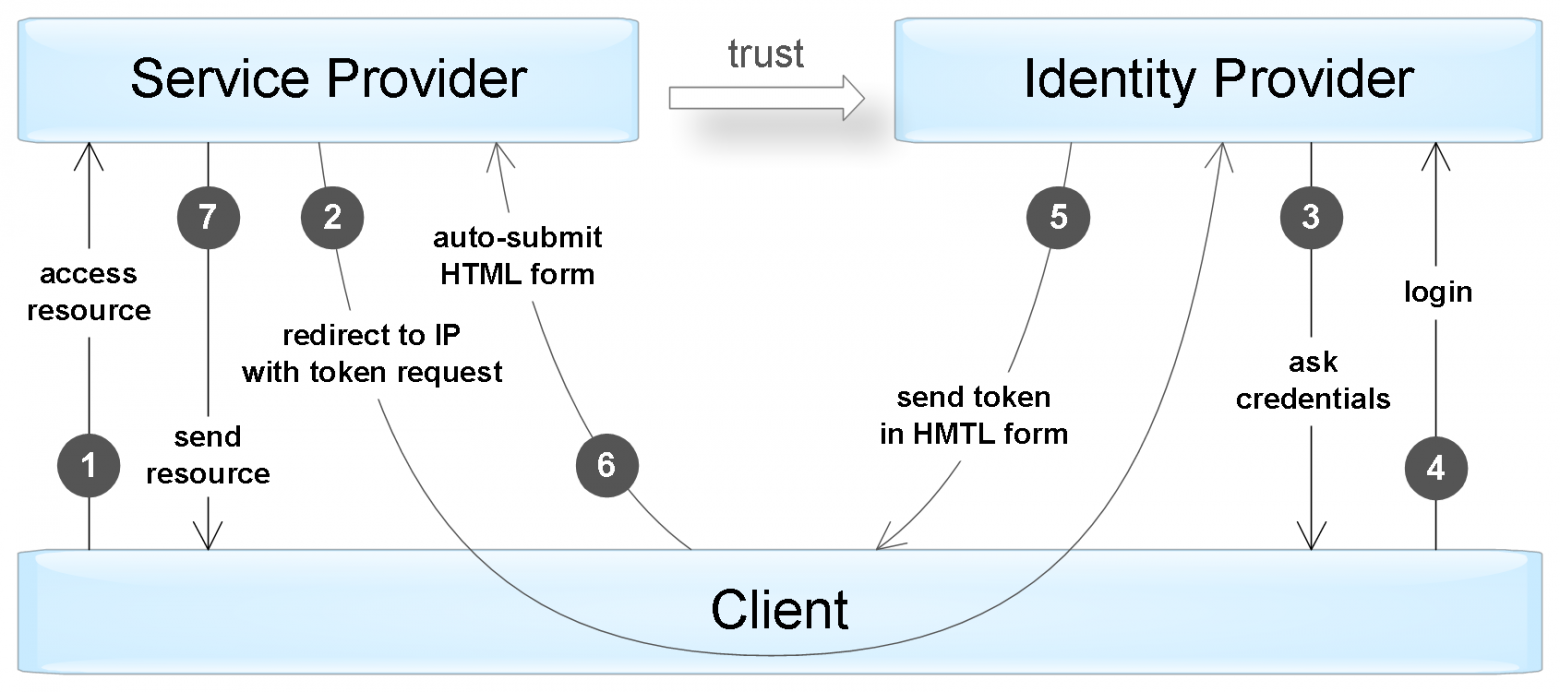
Slide 9. Аутентификация по ключам доступа



Slide 10 Аутентификация по токенам



SLIDE 11



**SLIDE 12 Стандарты, в точности определяющих протокол взаимодействия между клиентами**

OAuth

OpenID Connect

SAML

WS-Trust и WS-Federation.

**SLIDE 13 Проверка токена**

Токен был выдан доверенным identity provider приложением (проверка поля issuer).

Токен предназначается текущему SP-приложению (проверка поля audience).

Срок действия токена еще не истек (проверка поля expiration date).

Токен подлинный и не был изменен (проверка подписи).

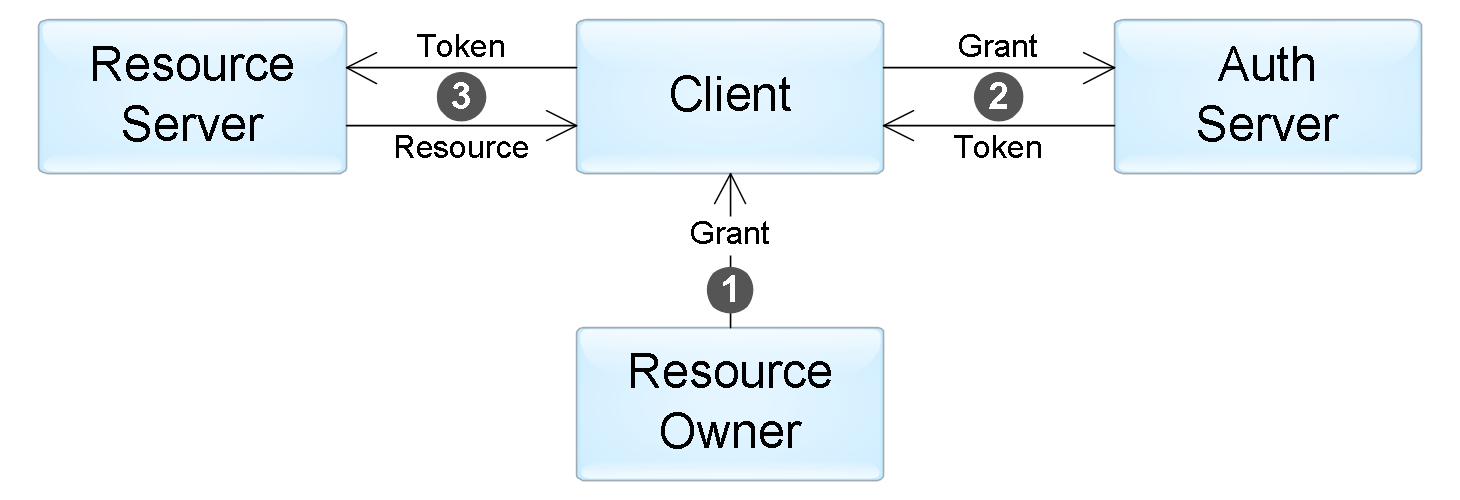
**SLIDE 14 Форматы токенов**

1. Simple Web Token (SWT)
2. JSON Web Token (JWT)
3. Security Assertion Markup Language (SAML)

**SLIDE 15 OAuth и OpenID Connect**

1. Пользователь (resource owner) дает разрешение приложению (client) на доступ к определенному ресурсу в виде гранта. Что такое грант, рассмотрим чуть ниже.
2. Приложение обращается к серверу авторизации и получает токен доступа к ресурсу в обмен на свой грант. В нашем примере сервер авторизации — Google. При вызове приложение дополнительно аутентифицируется при помощи ключа доступа, выданным ему при предварительной регистрации.
3. Приложение использует этот токен для получения требуемых данных от сервера ресурсов (в нашем случае — сервис Gmail).

SLIDE 16 *Взаимодействие компонентов в стандарте OAuth.*



SLIDE 17

Authorization Code

Implicit

Resource Owner Password Credentials

Client Credentials

SLIDE 18 Заключение

|  |  |  |
| --- | --- | --- |
| **Способ** | **Основное применение** | **Протоколы** |
| **По паролю** | Аутентификация пользователей | HTTP, Forms |
| **По сертификатам** | Аутентификация пользователей в безопасных приложениях; аутентификация сервисов | SSL/TLS |
| **По одноразовым паролям** | Дополнительная аутентификация пользователей (для достижения two-factor authentication) | Forms |
| **По ключам доступа** | Аутентификация сервисов и приложений | - |
| **По токенам** | Делегированная аутентификация пользователей; делегированная авторизация приложений | SAML, WS-Federation, OAuth, OpenID Connect |