



Financira  
Evropska unija  
NextGenerationEU



THE RECOVERY  
AND RESILIENCE  
PLAN

# VARNOST PROGRAMOV



Predavanja #6  
Matevž Pesek

# Teme!

- Vprašanja?
- Popravki abstraktov
- First draft! (27. april)
- Ne pozabite na rok! (15. maj)





# DANAŠNJE TEME

- Avtentikacija, avtorizacija, enkripcija
- OpenID, OAuth

# Od prejšnjič

- Kakšne napade XSS poznamo?
- Kaj so minimalni standardi za preprečitev XSS?
- Kaj je CSRF žeton?





# AVTENTIKACIJA

In posledice

# Avtentikacija

- Proces ugotavljanja identitete
  - Strežnik, klient
- Navadno preko:
  - Uporabniškega imena/gesla
  - Certifikata
  - MFA, passkey, ...
  - Tretje entitete, ki zagotavljajo identiteto

- Česa avtentikacija ne zagotavlja:
  - Pregleda nad operacijami, ki jih avtenticirana oseba lahko/sme izvrševati
  - Avtoritete nad tretjimi entitetami (Sigenca, Rekono, ...)



# Avtorizacija

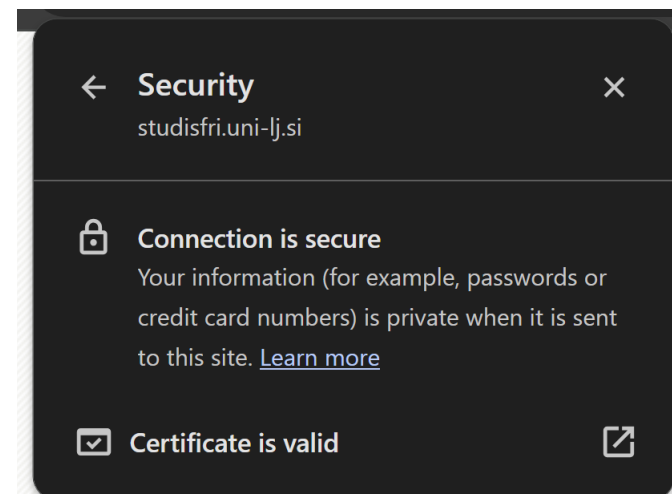
- Proces potrjevanja/omogočanja operacije
  - Navadno po avtentikaciji
  - Morebitna dodatna preverba
- Včasih nepotrebna
- Primeri:
  - Prenos datoteke s spletne strani
  - Potrjevanje plačila v banki

- Česa avtorizacija ne pokriva:
  - Avtentikacije
    - Klienta ali strežnika
  - Načina izvržbe operacije (npr. prenos preko http/s)

# Enkripcija

- Proces transformacije podatkov v obliko, ki ni berljiva/dosegljiva nepooblaščenim osebam
  - Navadno s kombinacijo zasebnega in javnega ključa
- Podatke je mogoče prebrati
  - Če imamo pravi ključ
  - Če je transformacija trivialna ali vsebuje ranljivosti

- Primeri (web)
  - SSH
  - HTTPS





# Primer - koncert

- Enkripcija
  - Kupimo izdelek preko spleta
- Avtentikacija
  - Kupimo kosilo s študentskim bonom (študentska izkaznica)
- Avtorizacija
  - Na koncertu želimo brezplačno pijačo/dostop do golden ringa s svojo karto



# Primer – Študentski informacijski sistem

- Enkripcija
  - Izmenjava podatkov z zaupanja vredno stranjo
- Avtentikacija
  - Identifikacija osebe, ki želi uporabljati spletno stran
- Avtorizacija
  - Dovoljenje, da oseba vidi elektronske indekse študentov

https://studisfri.uni-lj.si/DashboardUcitelj

FRI UNIVERZA V LJUBLJANI  
Fakulteta za računalništvo in informatiko

Predmeti Zaključevanje Študent Prakse Pooblastila Pregled Matevž Pesek

**Pregled ciklov vaj**  
Pregled ciklov vaj za svoje predmete lahko vidite [tukaj](#).

Zaključna tema je bila potrjena.

**Moji predmeti in izpitni roki** [Izpitni roki](#)

Predmet	Sprotne obveznosti	Izpitni roki	
<b>Grafično oblikovanje</b> BUN-LI-RE 2, BVS-RI LJ 2.3., Izmenjave LJ B. seznam študentov (124) <a href="#">delni rezultati anket (94)</a>	točke spr. obv. (124/124)	04. 09. 2024 (sre.) pisni ▲	ni prijavljenih
<b>Interdisciplinarni projekti</b> BMA-MM LJ RE 2. seznam študentov (19)	točke spr. obv. (0/19)	04. 06. 2024 (tor.) pisni ▲ 19. 06. 2024 (sre.) pisni ▲ 20. 08. 2024 (tor.) pisni ▲	ni prijavljenih ni prijavljenih ni prijavljenih
<b>Multimedijske tehnologije</b> BVS-RI LJ 3. seznam študentov (20) <a href="#">delni rezultati anket (18)</a>	točke spr. obv. (19/20)	22. 08. 2024 (čet.) pisni, ob 10:15 ⌚	ni prijavljenih
<b>Varnost programov</b> BVS-RI LJ 2. seznam študentov (21)	točke spr. obv. (0/21)	05. 06. 2024 (sre.) pisni ▲ 17. 06. 2024 (pon.) pisni ▲ 21. 08. 2024 (sre.) pisni ▲	ni prijavljenih ni prijavljenih ni prijavljenih



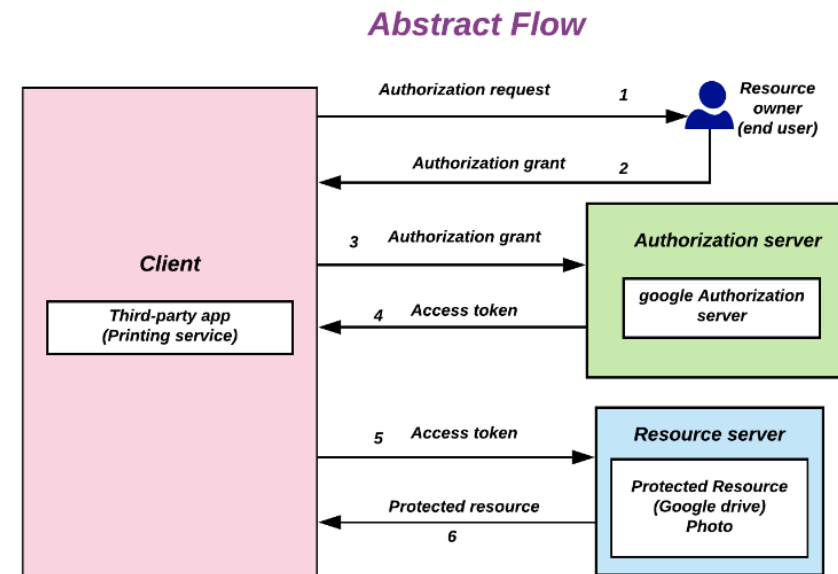
# PRIMERI STORITEV IN STANDARDOV

(PHP) session, OpenID, OAuth

# Storitve avtentikacije in avtorizacije

## Namen

- Minimizacija truda za implementacijo varne avtentikacije
- Poenotenje dostopa
- Omejitev dostopa do (pre)več podatkov o uporabniku
- Primeri
  - PHP session
  - OpenID
  - OAuth, auth0



# PHP Seja (Session)

## Stara šola

- Spremenljivka `$_SESSION`
- Zaledni sistem drži podatke o:
  - Avtentikaciji (uporabnik)
  - Veljavnosti (časovno pogojeno)
- Proces
  - `Session_start()`
  - `Session_unset()`
  - `Session_destroy()`

## Kako deluje?

- Vsakič preverimo identifikator
  - `session_id`
- Vrednosti hranimo v polju `$_SESSION`
- Na klientu držimo podatek v `PHPSESSION` piškotku



# Problematike tradicionalnih načinov<sup>+</sup> avtentikacije

## Omejitve

- Zgolj en zaledni sistem
- Vsaka storitev ločena avtentikacija
  - Poleg avtorizacije
- Poskusi reševanja z dodatnimi storitvami
  - Redis caching server
  - Vse storitve morajo imeti dostop do tega strežnika
  - Navadno omejeno na produkte ene entitete

## Problematike

- Možne ranljivosti
  - “kraja” piškotka
  - Več sej (slaba implementacija)
- Izogib problematikam
  - JWT žeton



# Razlike med storitvami

## **OAuth**

- Spletni standard za autorizacijo
- Uporablja (večinoma) JWT za kodiranje podatkov
- Podprti primeri:
  - Open authorization – avtorizacija
  - Dostopi, operacije v imenu uporabnika
  - Dodatno: poenotenje dostopa in upravljanje z uporabniki

## **SAML in OpenID**

- Avtentikacijska protokola
- SAML
  - Uporablja XML
- OpenID
  - Uporablja JWT žeton
- Primeri uporabe (že znani):
  - SSO dostop do tretjih entitet (spletnih strani in sistemov)
  - Žetoni za dostop s strani tretjih aplikacij/sistemov

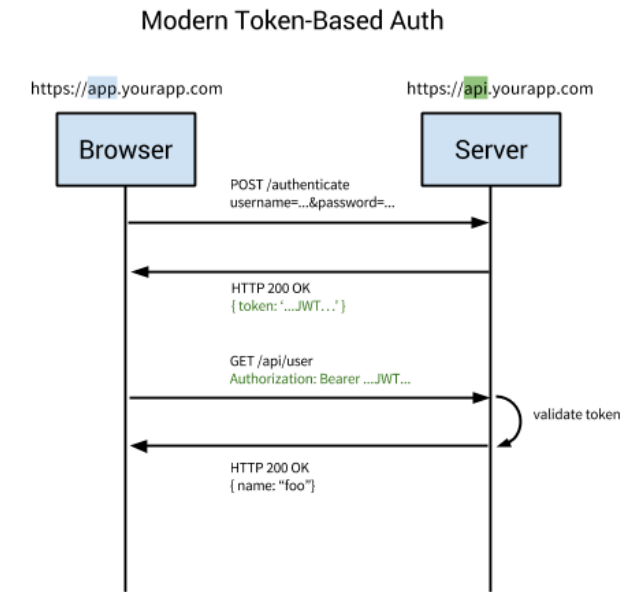
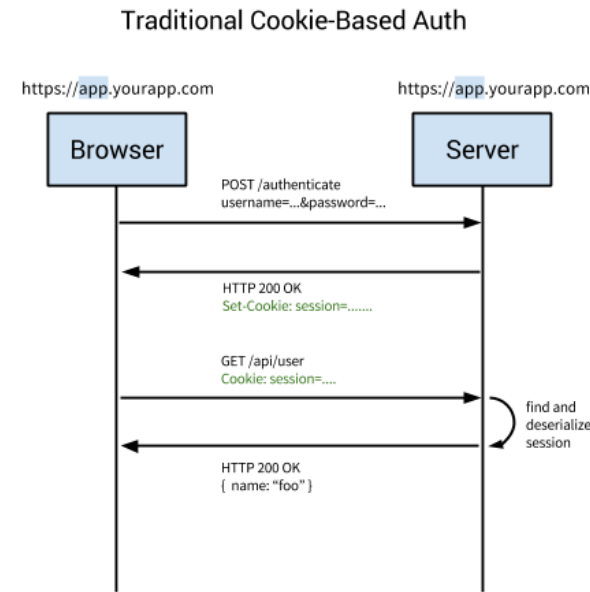


# JSON WEB TOKEN



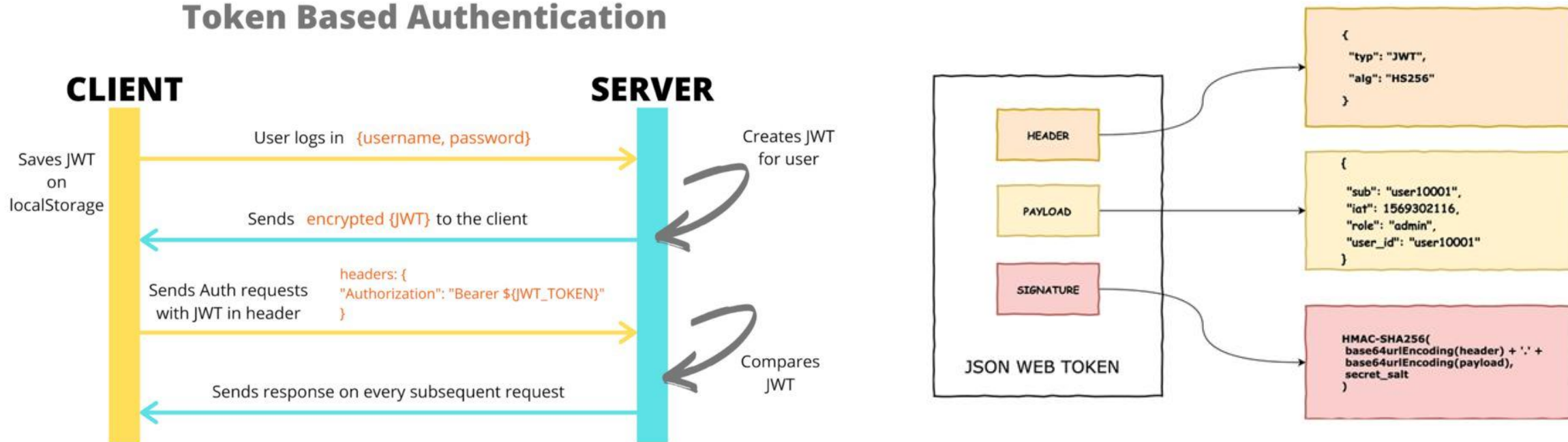
# Problematika prejšnjih metod

- Vsaka spletna aplikacija hrani uporabniška gesla v svoji bazi
- Seje so močno vezane na en zaledni sistem
  - Otežena replikacija storitev mitigacije z implementacijo svojih funkcij ali uporabo caching strežnikov (redis)
- Velika možnost za napake



# Kaj je JWT

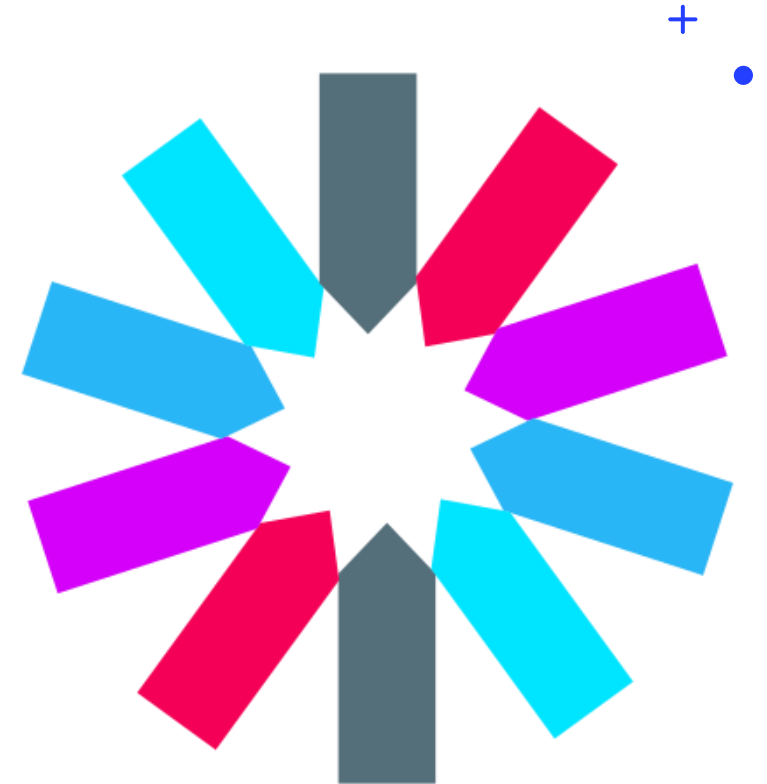
## Token Based Authentication



<https://dev.to/kcdchennai/how-jwt-json-web-token-authentication-works-21e7>

# Zakaj JWT?

- Striktno ločevanje med ponudnikom identitete in storitvijo, ki jo uporablja
  - Preverjanje podpisa na strani storitve brez ponudnika identitete
  - Vpis preko drugih spletnih strani
- Centraliziran sistem za upravljanje z uporabniki
- V primeru napada na storitev ni na sistemu podatkov o uporabniku
  - Lahko imitiramo druge uporabnike, če sistem uporablja simetrično enkripcijo!



# Kaj je JWT

- Razdeljen na 3 segmente
  - Base64 zakodirani
- Glava
  - Vsebuje podatke o algoritmu za preverjanje podpisa
  - HMAC, RSA, ECDSA, RSASSA
- Telo
  - Sestavljen iz atributov
  - Možnost dodajanja svojih atributov po potrebi

+

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "1234567890",  "name": "John Doe",  "iat": 1516239022}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
) ☐ secret base64 encoded
```

A cluster of three white geometric shapes on the left: a plus sign, a solid dot, and an open circle.A cluster of three white geometric shapes on the right: a plus sign, a solid dot, and an open circle.

# PRIMERI NAPADOV NA JWT

Don't try this at home ;)

# None algoritem

## Napad

- Glava JWT-ja vsebuje algoritem za podpis
- Standard vsebuje obvezno implementiran algoritem "None"
  - Namenjen že preverjenim žetonom v zalednih sistemih
- Pogosto vklopimo vse možne algoritme in nato uporabljamo samo enega

## Zaščita

- Eksplicitno določimo uporabljene algoritme
- Knjižnice ne uporabljajo None algoritma, če ga uporabnik ne nastavi

```
109
110 + if (!options.algorithms) {
111 +   options.algorithms = ~secretOrPublicKey.toString().indexOf('BEGIN CERTIFICATE') ?
112 +     [ 'RS256', 'RS384', 'RS512', 'ES256', 'ES384', 'ES512' ] :
113 +     [ 'HS256', 'HS384', 'HS512' ];
114 + }
115 +
116 + var valid;
```



# Zamenjava algoritmov

## Napad

- Asimetrična enkripcija omogoča transport javnih ključev
- JWKS standard omogoča automatizacijo za prenos in preverjanje podpisov
- S spremembo algoritma iz asimetričnega na simetričnega spremenimo logiko preverjanja podpisov

## Zaščita

- Eksplicitno določimo uporabljene algoritme
- Knjižnice probajo uporabljati samo simetrične ali asimetrične algoritme (odvisno od ključa)

```
▼ keys:
  ▼ 0:
    kty: "RSA"
    kid: "vev6SgpUU9B1DGETiVxyJqcRzzxct5UpAP94529ak4Y"
    e: "AQAB"
    ▼ n: "08CwQxnNJeZHWune7Nc7MncaJVtJab666zGUbyQ984BMpLeqnGX39vef
      o_i1XvGwWenK4Db__1aA7fE7npQtZrq5cnxJ0hSDJpU7n2cdBLZN0m8Q"
    use: "sig"
    alg: "RS256"
```

# Prihodnji teden

- Konzultacije v prihodnjih tednih:
  - Konzultacije za seminar (sva na voljo v učilnici/labu).
  - Pridite v čim večjem številu z osnutki vaših člankov!



+



o



•



# HVALA

Vaje

- Napad na JWT