

ՅՈՒՆԵՍԿՈ ԵՎ ԵՐԵՎԱՆԻ ՄԱՐԶԻ ԵՐԱՅԵՐԵՎԱՆԻ ՄԱՐԶԻ

ՀԱՅԿԱՍՏԱՆԻ

ՄԱՐԶԻ ԵՎ ՄԱՐԶԻ

Otvoreno računarstvo

Sigurnosni problemi Web aplikacija, problematika sjednice i aplikacije s pamćenjem stanja

- Osnovni koncepti sigurnosti na Internetu i sigurnosti Web aplikacija
- Pamćenje stanja, višekorisničke aplikacije i sjednica
- Rješavanje problema sjednice

Mario Žagar

Osnovni koncepti sigurnosti na Internetu i sigurnosti Web aplikacija

Sigurnost na Internetu



- **Internet je u načelu nesiguran**
 - Niz raznih mogućnosti napada, koje se u praksi kombiniraju
- **Praćenje mrežnog prometa (čitanje) je izuzetno lako**
 - **Prisluškivanje** (*eavesdropping, interception*)
 - Ako podaci nisu kriptirani, kao da stojite na ulici i slušate što drugi govore (*ako znate **koga** treba slušati*)
- **Lažno predstavljanje korisnika**
 - **Utjelovljivanje korisnika** (*impersonation*)
 - Probijanje sigurnosne tehnologije
 - Krađa autentikacijskih uređaja/podataka (kartice, tokena, lozinke, identiteta) – na razne načine
 - Pogađanje lozinki (korisničkih) – napad grubom silom i ostali napadi



Sigurnost na Internetu #2

- **Lažno predstavljanje poslužitelja ili klijenta**
 - Utjelovljivanje usluge ili računala (*impersonation*)
 - Lažno predstavljanje lažiranjem IP (ili MAC) adrese
 - Napad “čovjeka u sredini” (*man-in-the middle attack*, MITM)
 - Presretanje originalnih podataka (paketa) i mijenjanje te slanje dalje kao da je originalan
- **Pogađanje ključeva ili certifikata**
 - Napad grubom silom (*bruteforce attack*)
 - Napad poznatim šifriranim tekstom (*chosen-ciphertext attack*, CCA)
 - Napad poznatim čistim tekstom (*chosen-plaintext attack*, CCA)



Sigurnost na Internetu #3

- **Lažno predstavljanje usluge ili Web sjedišta**
 - Napad lažnim predstavljanjem (*phishing*)
- **Onesposobljavanje usluge preopterećenjem**
 - Napad generiranjem velike količine prometa ili poziva
 - Uskraćivanje usluge (*denial-of-service*)
- **Namjerno odugovlačenje ili ponavljanje poruke ili podataka**
 - Napad reprodukcijom (*replay attack*)
- **Promjena dijela poruke drugim podacima**
 - Napad zamjenom dijela poruke (*substitution attack*)

Sigurnosna zaštita

- Na svaki od ovih sigurnosnih problema se mora **adekvatno** odgovoriti
- **Razine zaštite su onoliko visoke (jake, skupe), koliko je ono što se čuva vrijedno (bitno)**
- Zaštite se obično rade **na nekoliko razina** kako bi se povećala sigurnost
- U načelu se štite:
 - Sustavi
 - Aplikacije
 - Komunikacija



Sigurnost sustava



- **Zaštita sustava**
 - Onemogućavanje **upada u mrežu**
 - Uporaba **vatrozida** (*firewall*)
 - Postavljanje **demilitariziranih zona** (DMZ)
 - Onemogućavanje **stražnjih vrata** (*backdoor*)
 - Zaštita i kontrola **bežičnih mrežnih** (WLAN) **konekcija**
 - Zabrana korištenja **modemskih priključaka**
 - Kontrola uporabe **mrežnih** (LAN) **priključaka**
 - **Praćenje** neuobičajenih i potencijalno štetnih **mrežnih aktivnosti**
 - Uporaba **antivirusnih** alata
 - Uporaba posebnih **sigurnosnih** alata



Sigurnost Web aplikacije



- **Zaštita aplikacije**
 - Onemogućavanje **stražnjih vrata** (*backdoor*)
 - Zaštita javnih servisa
 - Provjera sigurnosti svih dijelova aplikacije
 - Otvaranje sučelja samo prema poznatim klijentima
 - Autentikacija svih klijenata
 - **Praćenje** neuobičajenih i štetnih **aktivnosti**
 - Zapis svih aktivnosti (log)
 - **Sigurnosna testiranja** aplikacije
 - Simulacije namjernih napada

Sigurnost komunikacije



- Autentikacija korisnika
 - Princip ključ-brava – dokazivanje identiteta
 - Korištenje raznih metoda (lozinke, tokeni, certifikati)
- Autorizacija korisnika za skup akcija
 - Provjera da li korisnik ima odgovarajuća prava
- Zaštita poruka od čitanja i mijenjanja
 - Kriptiranje poruka
- Zaštita pristupa i čitanja podataka s komunikacijskog kanala
 - Kriptiranje komunikacije na kanalu



Sigurnosne tehnologije Web aplikacija



- Najčešće Web sigurnosne tehnologije – podskup navedenih tehnologija/tehnika/metoda:
 - **Zaštita komunikacijskog kanala** kriptiranjem
 - HTTPS protokol
 - **Autentikacija korisnika**
 - Lozinka
 - Token
 - Certifikat (npr. na pametnim karticama u sklopu PKI sustava)
 - **Autentikacija klijenta** (preglednika)
 - Certifikat na strani klijenta
 - **Autentikacija poslužitelja**
 - Certifikat na strani poslužitelja



HTTPS



- **https** je URI shema
 - Sintaksa identična **http** protokolu
 - Inicijalna postavka **vrata 443** (umjesto 80)
- Enkripcija/autentikacija između HTTP i TCP sloja temeljena na poznatim kriptografskim protokolima
 - **SSL** (Secure Socket Layer) ili
 - **TLS** (Transport Layer Security)
- Korištenje certifikata
 - Problematika potpisivanja/vjerovanja certifikatu
- Sprečava **niz napada** raznih tipova
- Ali, o svemu ovome više u zadnjem dijelu predavanja...



Pamćenje stanja, višekorisničke
aplikacije i sjednica

Pamćenje stanja



- Procesi s **pamćenjem stanja** (*statefull*)
 - Stanje u kojem je obrada se pamti
 - Najčešće za kompleksne, dugotrajne obrade
 - Često koriste transakcijsku okolinu
 - Najčešće postoji oporavak od neuspješnog izvršenja dijela obrade
- Procesi **bez pamćenja stanja** (*stateless*)
 - Stanje u kojem je obrada se ne pamti
 - Najčešće za kratkotrajne obrade
 - Često visokoefikasni, s malim vremenom čekanja
 - Filozofija "obradi i zaboravi"

Pamćenje stanja na poslužitelju



- **Poslužitelji** mogu:
 - **Pamtiti (čuvati) stanje** (*statefull*)
 - Obrada svakog novog zahtjeva (poziva) ovisna o obradi prethodnog zahtjeva (poziva)
 - Pogodno kada između zahtjeva postoji neki odnos
 - Model automata stanja
 - **Ne pamtiti (čuvati) stanje** (*stateless*)
 - Obrada svakog novog zahtjeva (poziva) neovisna o obradi prethodnih zahtjeva
 - Pogodno za pojedinačne nezavisne zahtjeve
 - Jednostavno, nema više "stanja"

Pamćenje korisnika



- HTTP je **protokol bez zadržavanja stanja** (*stateless*)
 - Pitanje – odgovor (*request – response*)
 - Svaki zahtjev je zaseban, ne postoji zavisnost
- Ali Web aplikacije često sadrže **autentikaciju**
 - U aplikaciju se "ulogiravamo" i iz nje "odlogiravamo"
 - Ako smo autenticirani, imamo **prava** za rad - autorizacije
- Web aplikacija treba **pamtiti "tko smo"** tijekom našeg rada – problem sjednice



- **Sjednica (*session*) je trajna veza između korisnika ili klijenta i poslužitelja**
 - Prijenos podataka u oba smjera
 - Pamćenje trenutnog stanja u kojem je proces (obrada)
 - Implementacija korištenjem sjedničkog sloja, mrežnog sloja ili u višim slojevima
 - Najčešće se kod Web aplikacija rješava u aplikacijskom sloju
 - Aktivna tijekom jednog "posjeta" (ili razgovora) ili višestrukih ponovnih "posjeta"
 - Analogija: Razgovor (konverzacija) tijekom jednog susreta ili nastavak razgovora kod sljedećeg susreta



Korisnički sjednički podaci

- Sjednica sadrži **korisničke sjedničke podatke** ili **stanje sjednice** (*session state*)
 - **Podaci o** trenutnom **stanju aplikacije ovisno o** trenutnom **korisniku**
 - Nisu vezani uz konkretni stroj, poslužitelj, virtualnu mašinu, aplikaciju ili klijenta već se po potrebi dijele (*sharing*)
 - Problem odabira mjesta i načina pohrane tih podataka

Pohrana stanja sjednice

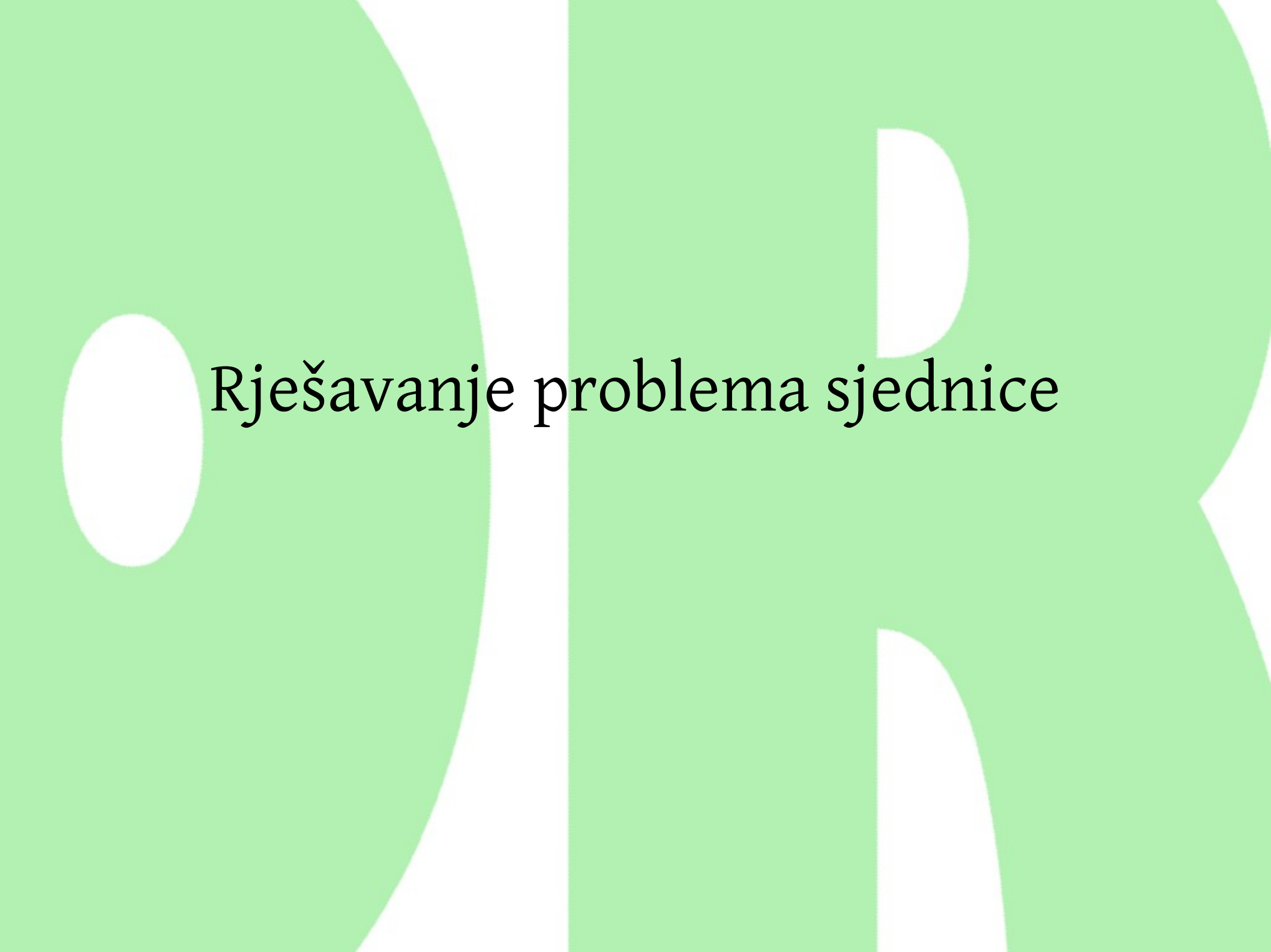


- Tipovi sjednica ovisno o mjestu pohrane stanja sjednice:
 - **Klijentske** (*client side*) sjednice
 - **Poslužiteljske** (*server side*) sjednice
 - **Kombinirane** sjednice
 - Npr. dio na klijentu, a drugi dio na poslužitelju
- Podaci su osjetljivi i moraju biti zaštićeni od neovlaštenog čitanja (**tajnost**), mijenjanja (**integritet**) i stvaranja (**autentičnost**)
 - Sigurnost je često tehnološki teže postići na klijentu

Sjednica i Web aplikacije



- Web aplikacije pohranjuju korisničke podatke
 - Tijekom jednog "posjeta" Web sjedištu
 - Podaci o korisniku i trenutnom "posjetu"
 - Tijekom višestrukih ponovnih "posjeta"
 - Podaci o prethodnim "posjetima"
- Često ipak samo jedan "posjet"
 - Kratkotrajni odnos klijenta (Web preglednika), odnosno korisnika i poslužitelja (Web i/ili aplikacijskog poslužitelja)
 - Životni vijek podataka – samo taj jedan "posjet" Web sjedištu
- Primjer tipične aplikacije – Web trgovina
 - Podaci o korisniku (npr. ime, prezime, adresa)
 - Podaci o trenutnoj "košarici" (pridruženi proizvodi)



Rješavanje problema sjednice

Rješenje problema pohrane sjedničkih podataka



- Pohrana sjedničkih podataka na strani klijenta
 - **Kolačić** (*cookie*)
 - **Skriveno polje** (*hidden field*)
 - **Prepisivanje URL-a** (*URL Rewriting*)
- Pohrana sjedničkih podataka na strani poslužitelja
 - **Memorija poslužitelja**
 - **Baza podataka**
 - **Datoteka**
- Pohrana sjedničkih podataka na strani poslužitelja kod višeposlužiteljskih sustava – dijeljenje podataka
 - **Memory-to-memory replikacija** između poslužitelja
 - **Dijeljena baza podataka**
 - **Peer-to-peer**

Kolačić (*cookie*)



- **Kolačić (*cookie*)**
 - manji podatak koji se razmjenjuje između Web poslužitelja i Web preglednika, a pohranjuje se na klijentu (Web pregledniku)
- Sinonimi: *cookie*, *HTTP cookie*, *web cookie*
 - Nastali iz termina "*magic cookie*" koji imaju smisao tiketa ili tokena
- Služe za pohranu informacija na klijentu određeno vrijeme
 - Pohrana u obliku **naziv – vrijednost**
- Koriste se za:
 - Autentikaciju, praćenje, pohranu podataka i sl.



Kolačić – primjena i značajke



- Prva primjena je bila za praćenje da li je korisnik "ulogiran" u Web sjedište (aplikaciju) i ostvarenje "košarice za kupnju" kod Web trgovina
- Značajke:
 - Mogu se onemogućiti na klijentu (u Web pregledniku) – potrebna provjera
 - Poništavaju princip bez pamćenja stanja (*stateless*) HTTP protokola – pozivi se povezuju pomoću podatka u kolačiću = ostvarivanje sjednice

Kolačić – princip rada



- Kolačiće postavlja/mijenja/briše:
 - Web poslužitelj
 - Web preglednik pomoću skriptnog jezika
 - Npr. JavaScript
- Princip rada:
 - Nakon zahtjeva (poziva) za stranicom, Web poslužitelj stvara kolačić i šalje ga usporedo s odgovorom Web pregledniku koji ga pohranjuje
 - Web preglednik pri sljedećim zahtjevima (pozivima) prema Web poslužitelju šalje i (nepromijenjeni) kolačić

Princip rada kolačića



WEB PREGLEDNIK

WEB POSLUŽITELJ

Web preglednik šalje zahtjev (*HTTP request*) za stranicom Web poslužitelju
GET /index.html HTTP/1.1
Host: www.fer.hr

Web poslužitelj odgovara na zahtjev (*HTTP response*) Web pregledniku, ali dodatno **stvara kolačić**, te i njega **vraća** zajedno s odgovorom

HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: naziv=vrijednost
... sadržaj stranice ...

Kod svakog sljedećeg zahtjeva Web preglednik **šalje** (nepromijenjeni) **kolačić** Web poslužitelju zajedno sa zahtjevom za stranicom

GET /index2.html HTTP/1.1
Host: www.fer.hr
Cookie: naziv=vrijednost
Accept: */*

Trajnost kolačića



- Postavljanjem postavke "trajnosti" dijelimo ih na:

- **Trajne** (*persistent*) kolačiće



- **Trenutne** kolačiće



- Upravljanje trajnošću

- Postavljanjem datuma brisanja
- Ako se ne postavi datum brisanja, kolačić se briše zatvaranjem preglednika

Kolačići – sigurnost



- Svaki Web preglednik pohranjuje svoj skup kolačića
 - Podatak u određenom Web pregledniku instaliranom na određenom računalu
 - Znači da Firefox i Internet Explorer ne dijele kolačiće
 - Više korisnika koriste isti Web preglednik na istom računalu te dijele kolačiće
 - Problem javnih računala (Web caffe)
- Mnoge države imaju zakonske odredbe vezane uz kolačiće
 - Npr. EU Directive 2002/58/EC on data protection and privacy
 - Donesena 31.7.2002. - rok primjene je bio 15 mjeseci
 - Korisnici moraju imati mogućnost odbijanja pohrane kolačića

Kolačići – napadi na sigurnost



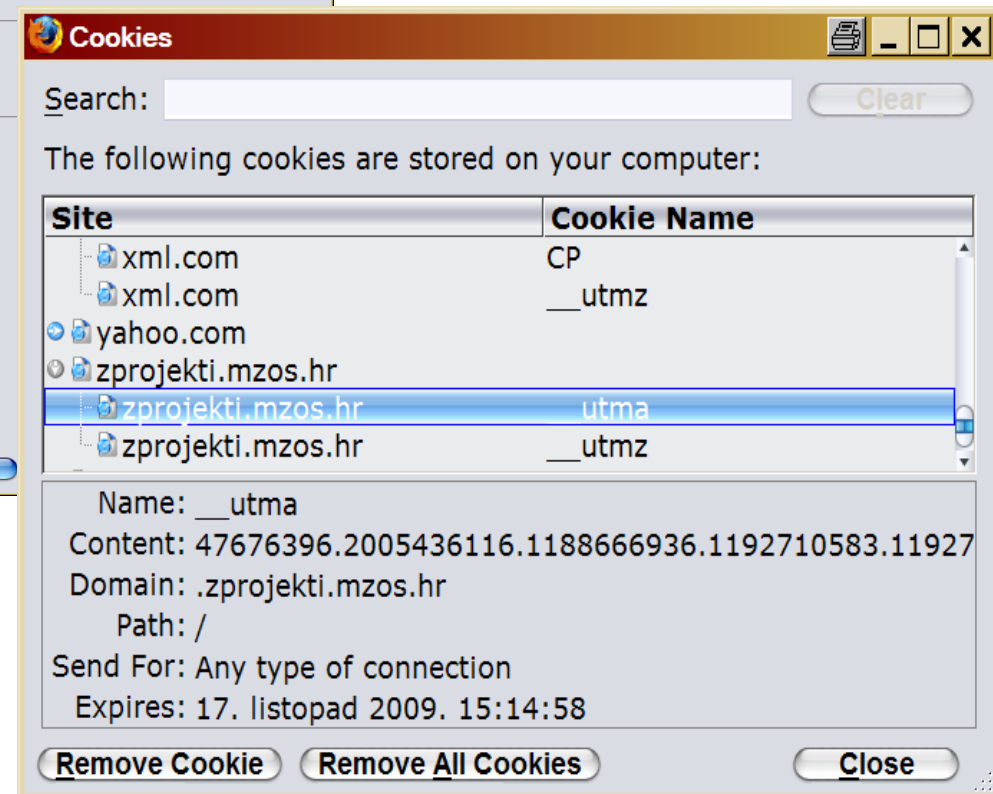
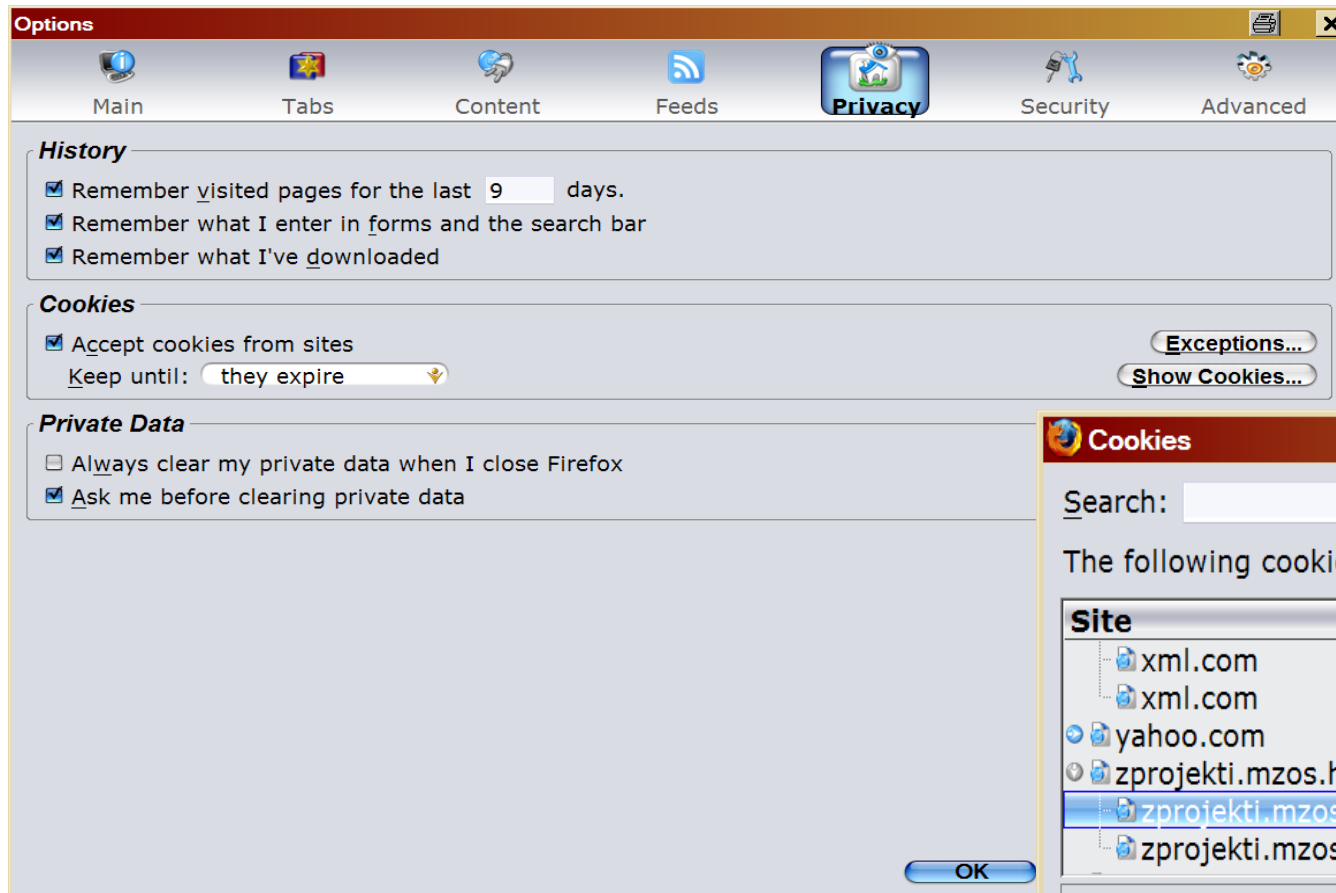
- Najčešći napadi vezani uz kolačiće:
 - Krađa podataka iz kolačića prisluškivanjem komunikacije (*sniffing*)
 - Slanje kolačića iz preglednika nekom drugom poslužitelju putem skriptnog jezika (*cross-site scripting*)
 - Namjerna promjena sadržaja kolačića na pregledniku (*cookie poisoning*)
 - Nekonzistencija preglednika i poslužitelja ("Back" gumb)
 - Kolačići treće strane (*third-party cookies*)
 - Npr. Web stranica sadrži slike i druge sadržaje koji se nalaze na nekom drugom Web poslužitelju, a ne onom kojem pristupamo pozivom te stranice
 - Taj drugi Web poslužitelj može postaviti svoje kolačiće koji se koriste za praćenje korisnika

Kolačići - implementacija



- Problemi kod implementacije:
 - Mogu se onemogućiti na klijentu (u Web pregledniku)
 - Potrebna provjera da li su onemogućeni
 - Ograničen sadržaj – veličina samo 4kB
 - Nesigurni - čitljivi kao podaci pohranjeni na klijentu
 - Ne bi ih trebalo koristiti za pohranu sigurnosnih informacija (npr. broja kreditne kartice)
- Često se pohranjuje samo identifikator sjednice
(*objašnjeno kasnije*)

Kolačići - Primjer



Kolačići – primjer u PHP

- Kako stvoriti kolačić?

- Uvijek postaviti **prije** <HTML> oznake
- Kolačić "korisnik" sa sadržajem "Ivo" i trajanjem 1h

```
<?php setcookie("korisnik", "Ivo", time()+3600); ?>
<html><body>...</body></html>
```

- Kako dohvatiti kolačić?

- Bilo gdje unutar PHP stranice

```
<?php echo $_COOKIE["korisnik"]; ?>
```

- A kako obrisati kolačić?

- Kolačić "korisnik" bez sadržaja i trajanjem -1h 😊

```
<?php setcookie("korisnik", "", time()-3600); ?>
```

- Primjer ispitivanja postojanja kolačića

```
<?php
if (isset($_COOKIE["korisnik"]))
    echo "Dobrodošao " . $_COOKIE["korisnik"] . "!";
else
    echo "Dobrodošao nepoznati korisniče!";
?>
```


Kolačići – prednosti i mane



- Prednosti:
 - Način uporabe neovisan o klijentu (pregledniku)
 - Jednostavnost (automatska uporaba)
 - Upravljanje trajnošću
 - Dobar način za baratanje kraćim podacima
- Mane:
 - Mogu se isključiti (na klijentu)
 - Nesigurni, prenose se i zapisuju kao tekst (moguće ih je pročitati i mijenjati)
 - Ograničena veličina podatka - 4kB
 - Trenutni sadržaj ovisan o konkretnom Web pregledniku instaliranom na određenom računalu
- Korištenje:
 - Isključivo za male skupove podataka bez potrebe za sigurnošću
 - Potrebne su alternativne metode ako su isključeni na pregledniku

Kolačići – smiješna strana 😊



- Objašnjenje nekih krivih navoda ("legendi") s mreže o kolačićima:
 - Kolačići **nisu** virusi ni crvi
 - Kolačići **nisu** programi
 - Kolačići **ne otvaraju** pop-up prozore
 - Kolačići se **ne koriste** za spam
 - Kolačići **nemaju nužno veze** s reklamiranjem
 - Kolačiće **ne morate** brisati!!! 😊

searchlineinfo.com/InsightExpress_cookie_study

Identifikator sjednice

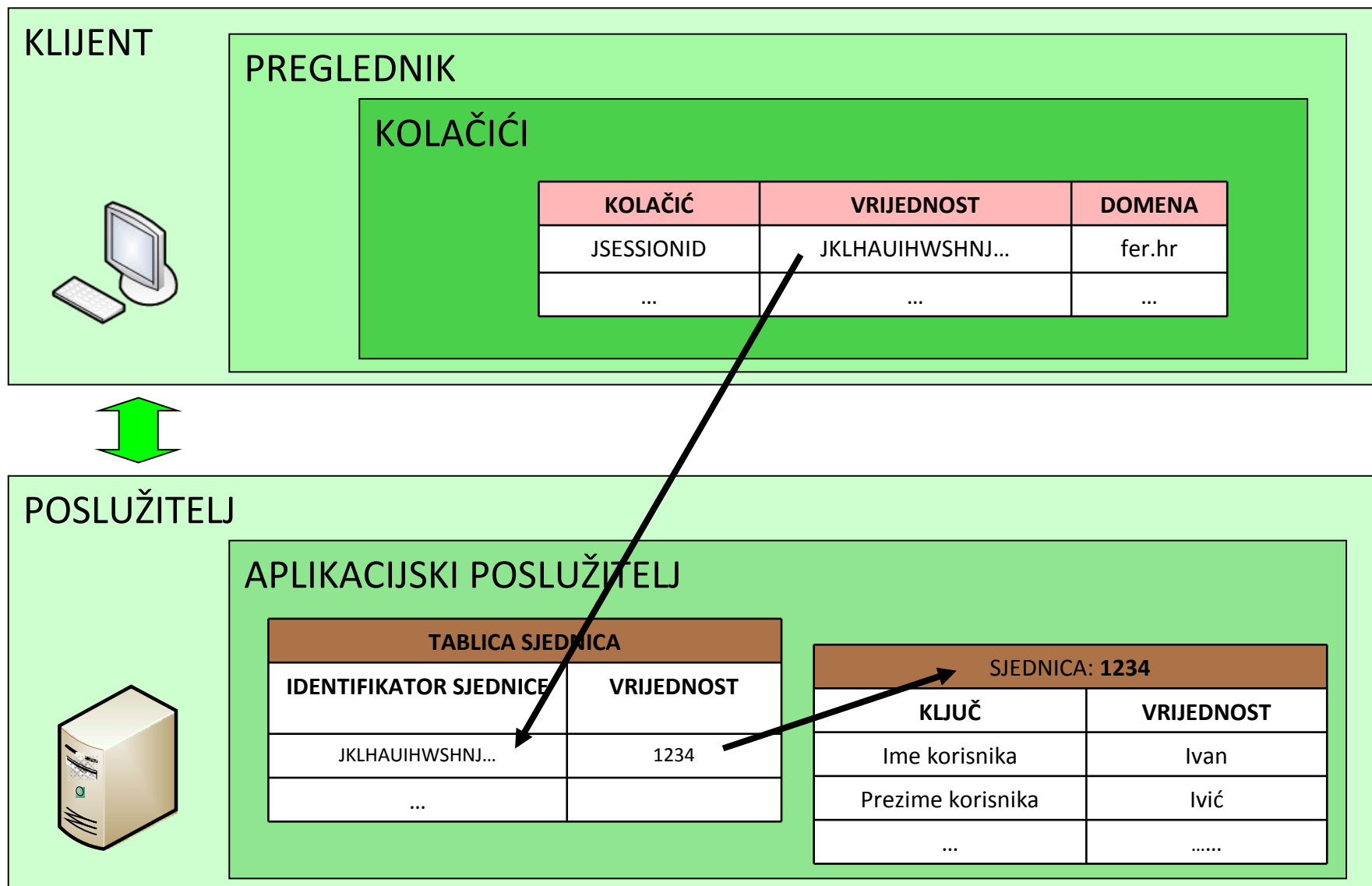


- **Identifikator sjednice** ili **token sjednice** je jedinstveni identifikator koji jednoznačno određuje sjednicu unutar nekog sustava ili aplikacije
- Zapis identifikatora sjednice:
 - Najčešće oblik velikog cijelog broja ili niza znakova
 - Za generiranje često se koristi jednosmjerna matematička funkcija (*hash*)
 - Naziv varijable u kojoj je pohranjen identifikator sjednice ovisi o implementaciji
 - Primjeri naziva u raznim programskim jezicima
 - Java i JSP: **JSESSIONID**
 - PHP: **PHPSESSID**
 - MS ASP: **ASPSESSIONID**

Princip korištenja identifikatora sjednice

- Klijent **pristupa** Web aplikaciji i **autenticira** se
- Poslužitelj prilikom dinamičkog generiranja stranice odgovora, **generira identifikator sjednice** i **šalje ga** klijentu
 - Ovo je trenutak početka sjednice
 - Npr. šalje ga u obliku kolačića ili u kodu generirane stranice
- Kod svakog sljedećeg poziva **klijent koristi** taj isti identifikator sjednice za identifikaciju sjednice
 - Klijent (kao dio zahtjeva) **šalje** identifikator sjednice poslužitelju ili dozvoljava pristup poslužitelju da ga pročita
 - Poslužitelj na temelju tog identifikatora pohranjuje sve druge podatke o sjednici na poslužiteljskoj strani

Pristup stanju sjednice



Identifikator sjednice – sigurnost



- **Pohrana identifikatora sjednice na klijentu**
 - U kolačiću ili u samom kodu stranice
 - **Sigurnost** – klijent pohranjuje samo taj relativno maleni podatak, u načelu besmisleni niz znakova, a svi podaci o sjednici se pohranjuju na poslužitelju
 - **Nema sigurnosnog rizika** čitanja osjetljivih podataka o sjednici na klijentu – oni su pohranjeni na poslužitelju
 - Ostale **podatke o sjednici može čitati/pisati samo aplikacija/sustav** koja ima pristup i poznaje identifikator sjednice
 - **Tajnost** – samo poslužiteljska aplikacija može čitati podatke
 - **Integritet** – samo poslužiteljska aplikacija može mijenjati podatke
 - **Autentičnost** – samo poslužiteljska aplikacija može stvoriti sjednicu i poslati identifikator sjednice

Sučelje za pristup kolačićima

- Sučelje za pristup kolačićima (npr. Java Cookie API)
 - **stvaranje kolačića** – Cookie (String naziv, String vrijednost)
 - **slanje** kolačića – HttpServletResponse.addCookie (Cookie kolacic)
 - **dohvat** kolačića – HttpServletRequest.getCookies ()
 - **dohvat naziva** kolačića – kolacic.getName ()
 - **dohvat vrijednosti** kolačića – kolacic.getValue ()
 - **promjena vrijednosti** kolačića – kolacic.setValue (String nova_vrijednost)
 - **postavljanje trajnosti** – kolacic.setMaxAge (int sekunde)
 - pretpostavljena vrijednost trajanja -1 – dok je aktivan preglednik
 - vrijednost 0 – nalog za brisanje kolačića
 - **ograničenje čitanja** kolačića
 - samo u (pod)domeni – setDomain (String naziv_domene)
 - samo unutar određenog puta – setPath (String naziv_puta)

Skrivena polja

- **Skrivena polja** (*hidden fields*)
- Uključivanje skrivene informacije u HTML obrasce
 - Pomoću INPUT oznake
 - `<INPUT NAME="naziv" TYPE="hidden" VALUE="12">`
- Podaci dohvatljivi kao parametri
 - `request.getParameter(String naziv)` (primjer za Javu)
 - Pohranjeni na klijentu – kao dio same HTML stranice
- Problemi:
 - Ograničen sadržaj
 - Nesigurni – podaci čitljivi u izvornom kodu stranice
 - Ne bi se trebalo koristiti sigurnosne informacije (npr. broj kreditne kartice)

Skrivena polja – prednosti i mane



- **Prednosti:**
 - Način uporabe neovisan o klijentu (pregledniku)
 - Ne mogu se isključiti (na klijentu)
 - Podržavaju veće podatke od kolačića
 - Dovoljno jednostavni za implementaciju
- **Mane:**
 - Nesigurni, prenose se i zapisuju kao tekst (moguće ih je pročitati i mijenjati)
 - Uvijek se moraju prenositi (povećanje mrežnog prometa)
 - Nisu podržani od standardnih struktura podataka u programskim jezicima
 - Rade samo s dinamički generiranim stranicama
 - Rade samo s tekstualnim podacima
- **Korištenje:**
 - Iskoristivi za veće tekstualne podatke

Prepisivanje URL-a

- **Prepisivanje URL-a** (*URL Rewriting*) je dodavanje podataka URL-u kako bi se pri pozivu moglo prepoznati kojoj sjednici poziv pripada
 - Može se koristiti kada klijent ne podržava druge mogućnosti (kolačići isključeni na klijentu)
 - Korištenje
 - Ili kolačići ili prepisivanje ili oboje – ali dosljedno!
 - Koristi se specifično programsko kodiranje URL-a
- Potrebno "pratiti" informacije o sjednici
 - Dodatna pažnja pri programiranju
 - Ograničeno samo na dinamički generirane stranice

Prepisivanje URL-a – prednosti i mane



- **Prednosti:**
 - Potpuna neovisnost o klijentu
 - Koristi metodu GET
 - Ne mogu se isključiti (na klijentu)
 - Dovoljno jednostavni za implementaciju
- **Mane:**
 - Nesigurni, prenose se i zapisuju kao tekst (moguće ih je pročitati i mijenjati)
 - Uvijek se moraju prenositi (povećanje mrežnog prometa)
 - Potrebno dodatno programiranje kod implementacije
 - Pojavljivanje u svakom URL
- **Korištenje:**
 - Koriste se kad nisu moguće druge opcije
 - Najčešće se koriste za identifikator sjednice

Pohrana sjedničkih podataka na poslužitelju



- **Pohrana sjedničkih podataka na poslužitelju**
 - Efikasan način pohrane sjedničkih podataka
- Najčešće se za pohranu na poslužitelju koristi
 - **Memorija poslužitelja**
 - Radna memorija (RAM)
 - Trajna memorija (tvrdi disk i sl.)
 - **Bazu podataka**
 - Najčešće relacijska baza podataka
 - **Datoteka**
 - Rijetko i zastarjelo

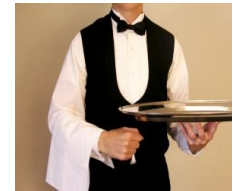
Višestruki poslužitelji

- Kod većih sustava aplikacija je instalirana paralelno na većem broju poslužitelja – višestruki poslužitelji
 - Ista aplikacija na svakom poslužitelju
 - Primjer: Amazon, eBay, ISVU, FER Web, ...
- Najčešće **grozd** (*cluster*) poslužitelja
- Razlozi:
 - **Skalabilnost** – više poslužitelja, pravilno opterećenje sustava za mnogo korisnika
 - **Upravljanje opterećenjem** (*load-balancing*) – ravnomjerno opterećenje svih poslužitelja
 - **Visoka dostupnost** (*high-availability*) – rješava problem nedostupnosti pojedinog poslužitelja



Problem dijeljenja podataka

- Kod višestrukih poslužitelja – problem: kako pristupiti istim podacima
 - Problemi koje aplikacija/sustav mora riješiti:
 - Kojem se poslužitelju prosljeđuje poziv s klijenta?
 - Kako se prebacuju podaci s jednog na drugi poslužitelj?
 - Analogija: jedan gost – više konobara
 - Potreban prijenos informacija ako dođe do promjene
 - Potreba za dijeljenjem podataka kada drugi poslužitelj preuzima posluživanje
 - Analogija: promjena konobara
 - Ako konobaru završi smjena (ili mu pozli) onda drugi konobar mora preuzeti narudžbu – preuzme "blok s narudžbom"
 - Drugi (novi) konobar ne smije "pitati" gosta da ponovi narudžbu



Rješenje problema dijeljenja podataka

- Potrebno omogućiti pristup zajedničkim podacima svim poslužiteljima, odnosno svim dijelovima aplikacije/sustava
 - Poslužitelji moraju dijeliti sjedničke podatke
 - Klijent se može spojiti na bilo koji poslužitelj
 - Jedan poslužitelj preuzima prvi poziv od klijenta, stvara sjednicu i pohranjuje sjedničke podatke na svim poslužiteljima dostupan način/mjesto
- Osnovna postavka – klijent se tijekom sjednice mora moći spojiti na bilo koji drugi poslužitelj koji ima pristup sjedničkim podacima i ne uočiti razliku

Izgradnja dijeljenja podataka

- **Načini dijeljenja podataka kod višeposlužiteljskih sustava**
 - **Memory-to-memory replikacija** između poslužitelja
 - **Dijeljena baza podataka**
 - **Peer-to-peer** dijeljenje podataka
- **Sjednički afinitet**
 - Princip po kojem korisnika nastavlja posluživati poslužitelj koji je i započeo
 - Minimiziranje potrebe za prebacivanjem između poslužitelja osim ako je nužno
 - Analogija: konobar poslužuje gosta od početka do kraja posjeta osim ako mu završi smjena

Pohrana sjedničkih podataka na poslužitelju



- Prednosti:

- Potpuna neovisnost o klijentu
- Moguća pohrana raznih veličina i tipova podataka
- Sigurno (onoliko koliko je sigurna metoda pohrane)
- Poslužiteljska aplikacija/sustav brine o podacima
- Dodatne mogućnosti (*caching, cluster, backup*)

- Mane:

- Ako izgubi identifikator sjednice, klijent ne zna kojim sjedničkim podacima treba pristupiti u sljedećem pozivu

- Korištenje:

- Najčešće kod višeposlužiteljskih sustava
- Najčešće u kombinaciji s drugim metodama

Najbolja praksa



- **Kombinirani pristup**
 - **Pohrana identifikatora sjednice na klijentu**
 - **Pohranu preostalih sjedničkih podataka na poslužitelju**
- **Prednosti kombiniranja**
 - Prijenos identifikatora sjednice i pohrana na klijentu
 - Sigurnost pohrane svih ostalih podataka na poslužitelju
 - Mogućnost korištenja ostalih prednosti velikih sustava (dijeljenje i trajna pohrana podataka sjednice)
- **Najčešće korišteni način kod *enterprise* aplikacija**

Single Sign On



- Single Sign-On (SSO) je metoda kontrole pristupa
- Omogućava pristup nizu aplikacija i sučelja sustava putem jednostruke autentikacije
 - Jedna autentikacija na početku pristupa
 - Pristup nizu aplikacija već autenticiranog korisnika
 - Najčešće se autentikacija događa na samom Web poslužitelju koji prosljeđuje daljnje pozive prema Web aplikacijama na aplikacijskim poslužiteljima
- Preuvjeti:
 - Sustav koristi samo jednu autentikacijsku shemu
 - Sustav je relativno homogen i koristi zajedničku IT infrastrukturu