UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Luka Hrgarek

ZBIRANJE PODATKOV IN PROFILIRANJE UPORABNIŠKIH NAPRAV S POMOČJO SPLETNIH BRSKALNIKOV

Magistrsko delo

Maribor, maj 2017

ZBIRANJE PODATKOV IN PROFILIRANJE UPORABNIŠKIH NAPRAV S POMOČJO SPLETNIH BRSKALNIKOV

Magistrsko delo

Študent:	Luka Hrgarek
Študijski program:	Magistrski študijski program
	Informatika in tehnologije komuniciranja
Mentor:	doc. dr. Marko Hölbl
Lektor:	Drago Meglič, prof. slovenščine

Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija

Številka: E5024724
Datum in kraj: 21. 04. 2017, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Statut UM – UPB 11, Ur. l. RS, št. 44/2015) izdajam

SKLEP O ZAKLJUČNEM DELU

1. **Luki Hrgareku**, študentu študijskega programa 2. stopnje MAG INFORMATIKA IN TEHNOLOGIJE KOMUNICIRANJA, se dovoljuje izdelati zaključno delo.

2. Tema zaključnega dela je pretežno s področja Inštituta za informatiko.

MENTOR: doc. dr. Marko Hölbl

3. Naslov zaključnega dela:
ZBIRANJE PODATKOV IN PROFILIRANJE UPORABNIŠKIH NAPRAV S POMOČJO SPLETNIH BRSKALNIKOV

4. Naslov zaključnega dela v angleškem jeziku:
DATA COLLECTION AND USER DEVICE FINGERPRINTING USING WEB BROWSERS

5. Rok za izdelavo in oddajo zaključnega dela je 21. 04. 2018. Zaključno delo je potrebno izdelati skladno z "Navodili za izdelavo zaključnega dela" in ga v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) oddati v pristojnem referatu članice. Hkrati se odda tudi izjava mentor-ja/-ice (in morebitnega somentor-ja/-ice) o ustreznosti zaključnega dela.

Pravni pouk: Zoper ta sklep je možna pritožba na Senat članice v roku 10 delovnih dni od dneva prejema sklepa.

Dekan:
red. prof. dr. Borut Žalik

UNIVERZA V MARIBORU FAKULTETA ZA ELEKTROTEHNIKO, RAČUNALNIŠTVO IN INFORMATIKO

Obvestiti:
- kandidata,
- mentor-ja/-ico,
- odložiti v arhiv.

Zbiranje podatkov in profiliranje uporabniških naprav s pomočjo spletnih brskalnikov

Ključne besede: zasebnost, spletni brskalnik, profiliranje

UDK: 004.65:[004.455.1:004.738.5](043.2)

Povzetek

Svetovni splet se je od začetkov svojega obstoja preobrazil iz zbirke s hiperpovezavami povezanih dokumentov v globalno platformo, na kateri so dostopne najrazličnejše programske rešitve. Programski jezik JavaScript je ključnega pomena za interaktivnost na spletu in omogoča razvijalcem dostop do številnih podatkov o uporabnikovem brskalniku ter posledično o uporabniku samemu. Kljub mnogim ukrepom organizacije World Wide Web Consortuim (W3C) in proizvajalcev brskalnikov je postopek zbiranja podatkov z uporabo kode JavaScript za navadnega uporabnika neviden, kar odpira možnosti zlorab. V magistrskem delu smo obravnavali možnosti zbiranja podatkov o brskalnikih in uporabniških napravah s pomočjo namenske spletne aplikacije ter analizo stopnje zavedanja uporabnikov o možnosti zbiranja omenjenih podatkov. Ugotovili smo, da spletne aplikacije lahko pridobivajo podatke o brskalnikih v tolikšni meri, da to omogoča enolično identificiranje spletnih brskalnikov. Prav tako se je pokazalo, da so uporabniki dobro ozaveščeni o možnosti pridobivanja podatkov s pomočjo spletnih brskalnikov.

Data collection and user device fingerprinting using web browsers

Key words: privacy, web browser, fingerprinting

UDK: 004.65:[004.455.1:004.738.5](043.2)

Abstract

The development of World Wide Web has transformed it from a document storage with hyperlinks to a global platform on which a wide variety of software solutions are available. The programming language JavaScript, which is needed for interactivity on the web allows developers access to user browsers' data and hence the user himself. Despite many measures of World Wide Web Consortium (W3C) and browser vendors, the data collection process using JavaScript code is invisible to the user, which makes if susceptible to misuse.

In this thesis we present possibilities of collecting data of browsers and user devices via a dedicated web application and the analysis of user awareness on the possibility of collecting such data. The results show that web applications can retrieve data about browsers to an extent that can uniquely identify web browsers. Additionally, we it was shown that users are well aware of the possibility of collecting data using web their browsers.

Kazalo

Kazalo slik

Kazalo grafov

Kazalo tabel

Kazalo izsekov programske kode

Seznam uporabljenih kratic

API	*Application Programming Interface*
CDMA	*Code-division multiple access*
CSS	*Cascading Style Sheets*
DOM	*Document Object Model*
GPS	*Global Positioning System*
GSM	*Global System for Mobile Communications*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
HTTPS	*Hyper Text Transfer Protocol Secure*
IETF	*Internet Engineering Task Force*
IP	*Internet Protocol*
JSON	*JavaScript Object Notation*
MAC	*Media Access Control*
MVC	*Model-View-Controller*
NAT	*Network Addres Translation*

PKI *Public Key Infrastructure*

REST *Representational state transfer*

RFID *Radio-frequency identification*

TLS *Transport Layer Security*

URL *Uniform Resource Locator*

UUID *Universally unique identifier*

W3C *World Wide Web Consortium*

1 Uvod

V spletnih začetkih so uporabniki uživali veliko stopnjo anonimnosti, saj so bile spletne strani takrat samo hipertekstualni dokumenti, povezani s hiperpovezavami [91]. Danes se številni uporabniki pri uporabi spleta dnevno srečujejo s problemom zasebnosti, saj se na spletu nahaja velika količina uporabniško ustvarjene vsebine, ki jo uporabljajo različne aplikacije in storitve za prilagajanje uporabnikovim željam (personalizacija) ter njegovi socialni mreži. Uporabniškemu brskanju prav tako sledijo tudi oglaševalska omrežja [91]. Spletne strani in aplikacije beležijo podatke o uporabnikih, vendar se večina uporabnikov tega ne zaveda. Slednje in profiliranje uporabnikov omogoča že obisk določenega spletnega mesta s pomočjo piškotkov ali naprednejših tehnik. Uporabniki velikokrat že sami soglašajo in dovoljujejo zbiranje podatkov.

Tovrstno spremljanje uporabniškega brskanja vzbuja pomisleke o zasebnosti, posebej če je takšen profil možno povezati z uporabnikovo identiteto [111]. Različni avtorji govorijo o načinih enoličnega identificiranja uporabnika, oziroma identificiranja njegovega brskalnika, in to z uporabo več različnih metod ter njihovih kombinacij: lastnosti strojne opreme [79, 94], senzorjev [112], zgodovine brskanja [53], geolokacije [59], pisav [48, 104] in podobno. Z možnostmi, ki jih je omogočil standard HTML 5 [42], se možnosti pridobivanja podatkov in profiliranja samo povečujejo, saj je mogoče dostopati do naprave uporabnika, pridobivati geolokacijske podatke ali (deloma) dostopati do datotek.

V okviru magistrskega dela želimo proučiti področje pridobivanja podatkov o uporabniških napravah s pomočjo spletnih brskalnikov ter tehnike in načine, kako to realizirati. Izdelali bomo anketni vprašalnik, v katerem bomo anketirance povprašali o

njihovih stališčih glede zasebnosti na spletu in v brskalnikih. Namen dela je tudi izdelava spletnega mesta, ki bo uporabnikom prikazalo, v kakšni meri se podatki o njih lahko zbirajo. Tako želimo dvigniti stopnjo zavedanja uporabnikov ter jih ozavestiti, da je na spletu treba biti previden, če želimo ohraniti zasebnosti.

1.1 Cilji in raziskovalne hipoteze magistrskega dela

Osrednji cilj magistrskega dela je izdelava spletne aplikacije, ki bo pridobila podatke o uporabniku, ki jih je mogoče uporabiti za profiliranje brskalnikov obiskovalcev spletne aplikacije.

Prav tako bomo zasnovali in izvedli anketni vprašalnik, s katerim bomo uporabnike povprašali o njihovih stališčih o zasebnosti na spletu. Od uporabnikov bomo pridobili podatke o njihovem mnenju glede možnosti pridobivanja podatkov s pomočjo njihovih brskalnikov.

Po izvedbi anketnega vprašalnika bomo vsakega udeleženca preusmerili na spletno aplikacijo, ki bo prikazala možnosti pridobivanja podatkov o brskalniku uporabnika brez njegovega dovoljenja. Od uporabnika bomo tudi poskušali dobiti dovoljenje za dostop do podatkov, ki zahtevajo eksplicitno dovoljenje uporabnika. Pridobljene podatke bomo hranili v anonimizirani obliki, vendar povezani z anketnim vprašalnikom. Vsakemu anketirancu bomo dodelili edinstveni identifikator, ki bo služil za povezovanje podatkov med anketnim vprašalnikom in spletno aplikacijo. Tako želimo pridobiti korelacijo med uporabnikovimi odgovori in obnašanjem pri uporabi spletne aplikacije. Na koncu bomo analizirali rezultate ankete in podatkov spletne aplikacije ter ugotavljali morebitno povezavo med njimi.

Glavni cilji, ki jih želimo doseči, so:

- izdelava spletne aplikacije za pridobivanje podatkov o uporabniških napravah s pomočjo spletnih brskalnikov,
- priprava in izvedba ankete v povezavi s spletno aplikacijo,

- analiza in interpretacija rezultatov.

Hipoteze

H_1: *S pomočjo spletne aplikacije je mogoče pridobivati podatke o uporabnikovem spletnem brskalniku v tolikšni meri, da to omogoča enolično identificiranje spletnih brskalnikov.*

H_2: *Uporabniki se ne zavedajo možnosti pridobivanja podatkov s pomočjo spletnih aplikacij.*

H_3: *Uporabniki so v večji meri odgovorni za količino podatkov, ki jih je mogoče pridobiti, o svojih spletnih brskalnikih.*

1.2 Predpostavke in omejitve magistrskega dela

Predpostavljamo, da je možno pridobiti podatke o uporabniških napravah s pomočjo spletnega brskalnika.

V raziskavi bomo upoštevali naslednje omejitve:

- raziskavo (anketni vprašalnik in spletno aplikacijo) bomo izvedeli na omejeni množici uporabnikov,
- uporabljali bomo le podatke, ki jih je možno pridobiti s pomočjo spletnega brskalnika, tj. brez nameščanja dodatne programske opreme,
- za pridobivanje podatkov o uporabnikih s pomočjo spletne aplikacije se bomo omejili na spletni brskalnik *Google Chrome*.

1.3 Predvidene metode raziskovanja

V magistrskem delu bomo uporabili metodo *deskripcije*, s katero bomo opisali možnosti zbiranja podatkov o napravah s pomočjo spletnih brskalnikov. Metodo *analize* bomo uporabili za pojasnjevanje in razčlenjevanje podatkov, ki jih lahko pridobimo o uporabniških napravah. Te podatke bomo klasificirali v ustrezne kategorije na podlagi kriterijev, ki jih bomo izbrali (vpliv na varnost, možna tveganja ...).

V nadaljevanju bomo uporabili metodo *anketiranja* v kombinaciji s spletno aplikacijo, saj želimo izvedeti stopnjo zavedanja uporabnikov o podatkih, ki jih lahko pridobimo, in jo hkrati primerjati z rezultati pridobljenih podatkov spletne aplikacije, v kateri bomo pogledali, katere podatke je možno pridobiti.

1.4 Struktura magistrskega dela

Najprej bomo s pomočjo spletnega brskalnika predstavili možnosti pridobivanja podatkov o brskalniku, uporabnikovi napravi in o samem uporabniku. Na tem področju novi standardi prinašajo različne spremembe in dopolnitve, ki odpirajo različna vprašanja uporabnikove zasebnosti. Predstavili bomo pojem zasebnosti in ga umestili v kontekst spleta in spletnih brskalnikov. Potem bomo pogledali različne sodobne načine zbiranja podatkov, ki jih omogočajo brskalniki. V nadaljevanju bomo govorili o možnostih profiliranja uporabniških brskalnikov ter identificiranja in sledenja uporabnikom.

Sledita opisa priprave eksperimentalne spletne aplikacije ter anketnega vprašalnika. Po njuni izvedbi bomo rezultate analizirali in jih interpretirali ter podali ugotovitve naše raziskave.

Magistrsko delo bomo zaključili s sklepom, v katerem bomo predstavili končne ugotovitve.

2 Možnosti pridobivanja podatkov o uporabniških napravah s pomočjo spletnega brskalnika

V tem poglavju bomo predstavili pojem zasebnosti in ga umestili v kontekst spleta ter spletnih brskalnikov. Izpostavili bomo sodobne možnosti zbiranja podatkov v brskalnikih, saj novi standardi definirajo možnosti dostopa do mnogih senzorjev, kar še posebej pride v poštev pri mobilnih napravah. Prav tako bomo obravnavali možnosti izgradnje profila iz množice zbranih podatkov.

2.1 Zasebnost na spletu

Splet je v mnogih pogledih vir težav z varnostjo in zasebnostjo. Uporabniki od spleta ne pričakujejo samo brskanja brez škode, ampak tudi hitro, vizualno atraktivno in interaktivno storitev [51]. Zasebnost je pravica, ki je na današnjem spletu pogosto kršena; včasih zaradi nevednosti uporabnikov, včasih pa zaradi zlorab ponudnikov storitev. Zato je za uporabnike vprašanje zasebnosti postalo bistvenega pomena [95]. Obstaja več definicij zasebnosti, najbolj uveljavljeni sta Westinova [110] in Altmanova [10] [80]. Westin zasebnost definira kot „*trditev posameznikov, skupin ali institucij, da lahko same zase odločajo, kdaj, kako in v kakšnem obsegu bodo drugim razkrili informacije o sebi*“ [110]. Zasebnost je ena temeljnih človekovih pravic, ki jo določa *Splošna deklaracija človekovih pravic* [12]. Na podlagi teh predpostavk in upoštevajoč

tehnološko okolje, v katerem smo se znašli, je ohraniti to pravico dandanes težje kot kadar koli prej [95, 17, 72].

Na spletu lahko ločimo dve različni skupini: *uporabnike* in *ponudnike storitev*, kot so Amazon, Google, Microsoft, Facebook in podobni. Ponudniki storitev želijo zbrati podatke in se o svojih uporabnikih naučiti čim več, saj tako lahko bolje usmerjajo svoje oglase ali zagotavljajo boljšo personalizacijo vsebine. Po drugi strani pa uporabniki tolerirajo vsebine, oglase in personalizacijo, dokler ne ogroža njihove zasebnosti [41].

Cilj sledenja brskalniškim navadam posameznega uporabnika je običajno zbiranje čim večje količine informacij o njem iz čim več razpoložljivih virov. Tako obsežno zbiranje podatkov o nekom imenujemo profiliranje [21]. Za profiliranje uporabnikov se uporablja več različnih pristopov [89]. Prvi pristop imenujemo „informacijske supermoči“ (angl. *information superpowers*). Ta zagotavlja obsežen nabor storitev (npr. e-pošta, koledar, družbena omrežja), ki zajemajo potrebe večine uporabnikov po tovrstnih storitvah. V primeru uporabe takšnih storitev, katerih ponudnik je npr. Google, uporabniki v sistem vnašajo podatke v količini, ki zadostuje potrebam profiliranja. Drugi pristop je zajemanje javno dostopnih virov osebnih podatkov (npr. družbena omrežja, mikro spletni dnevniki (angl. *microblogging services*)), iz katerih je z minimalnim vložkom možno dobiti informacije o nekom. Uporabljajo se tudi tehnike sledenja s pomočjo metod, ki temeljijo na piškotkih, in različne tehnike „zajemanja prstnih odtisov“ (angl. *fingerprinting*) uporabnika [21].

Uporabniki pogosto zahtevajo, da se dolgoročne seje hranijo znotraj brskalnika na strani odjemalca [51], saj je tako uporaba spletnih storitev zanje lažja in zahteva manj interakcije, ki ni neposredno povezana s primarnim namenom storitve, temveč se nanaša na postopke avtentikacije in personalizacije.

Eden izmed primerov personalizacije je izboljševanje spletnega iskanja. Čeprav spletni iskalniki uspešno zadoščajo informacijskim potrebam uporabnikov, njihovo delovanje ni optimalno. Glavno pomanjkanje obstoječih iskalnikov je njihovo sledenje modelu „ena velikost ustreza vsem“ (angl. *one size fits all*), zaradi česar individualnim uporabnikom niso prilagodljivi [99]. To lahko ponazorimo z iskalnim nizom „`ajax`“. Povpraševanje

bo kot odgovor vrnilo rezultate o načinu spletnega razvoja Ajax, podatke o nizozemskem nogometnem klubu Ajax Amsterdam in spletne strani o proizvodu za čiščenje Ajax. Povsem očitno je, da so za različne uporabnike tukaj zaželeni različni rezultati [66]. Tudi številne raziskave so potrdile, da je večina iskalnih nizov kratka [55, 100] in dvoumna [31, 96].

Področje zasebnosti na spletu obravnava tudi zakonodaja. Direktiva Evropske unije o varstvu osebnih podatkov [88] definira „določljivo osebo“ kot tisto, „ki se lahko neposredno ali posredno identificira, predvsem s sklicevanjem na identifikacijsko številko ali na enega ali več dejavnikov, ki so značilni za njeno fizično, fiziološko, duševno, ekonomsko, kulturno ali socialno identiteto“. Kombinacija različnih podatkovnih elementov lahko omogoči edinstveno identifikacijo osebe. Literatura, ki se nanaša na področje zasebnosti, je vpeljala pojme, kot so *deidentifikacija* (odstranjevanje informacij o identiteti iz podatkov – anonimizacija) ter *reidentifikacija* (sposobnost povezovanja domnevno anonimnih podatkov z njihovo dejansko identiteto) [62].

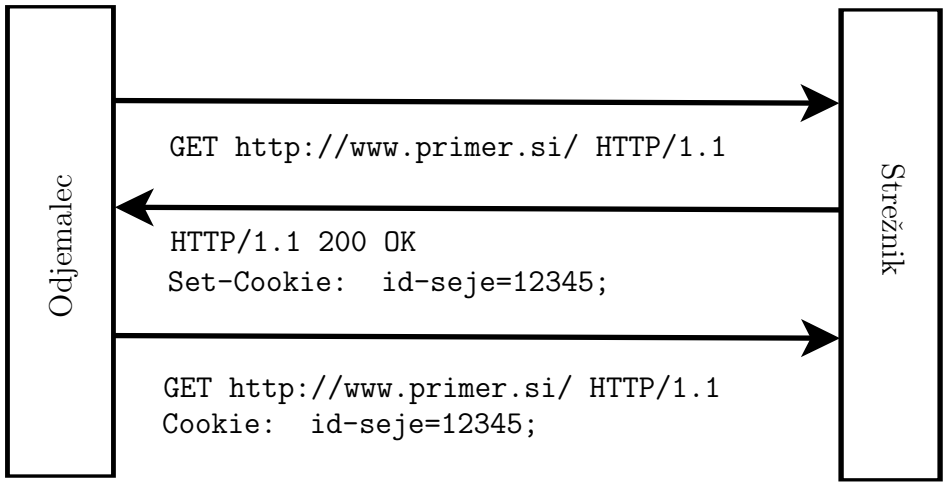
2.2 Metode zbiranja podatkov v brskalnikih

Ko uporabniki brskajo po spletu, zapletena mreža *personalizacijskih storitev* spremlja njihove preference s pomočjo sledenja njihovim brskalniškim navadam. Ti podatki se uporabljajo za pripravo personaliziranih predlogov, kot so predlagani izdelki za nakup, spletne strani, ki bi jim lahko bile zanimive, družbene povezave ipd. Personalizacijske storitve se pogosto zanašajo na mnoge tehnike sledenja uporabnikom z različnimi spletnimi stranmi in aplikacijami [91], saj ima vsaka spletna stran dostop do velikega nabora podatkov, ki so v brskalniku na voljo. V tem poglavju bomo opisali različne metode in možnosti zajemanja podatkov iz brskalnikov.

2.2.1 Spletni piškotki

Spletni piškotek (angl. *web cookie*) je majhen besedilni niz (pogosto samo ID-številka), ki jo spletni strežnik pošlje brskalniku. Ta niz brskalnik shrani, običajno na uporab-

nikov trdi disk, in ga kasneje strežniku pošlje nazaj. Brskalniki so začeli podpirati piškotke leta 1995. Ti so bili prvotno razviti, da bi uporabnikom zagotovili način ponovnega obiska strani, ne da bi se ponovno morali identificirati in spreminjati nastavitve spletne strani [73].



Slika 2.1: Postopek izmenjave spletnih piškotkov [71, 14]

Spletni piškotki imajo dve poglavitni pomanjkljivosti: piškotek lahko identificira samo eno brskalniško aplikacijo in piškotek se lahko izbriše, kar pomeni, da lahko izgubimo identifikator [21]. Kljub temu je metoda še vedno pogosto uporabljena [44].

2.2.2 Zajemanje zgodovine brskanja

Svetovni splet je nastal leta 1990 kot vmesnik za veliko zbirko statičnih dokumentov („strani“) [18]. V tem kontekstu je bil za uporabnike koristen prikaz metapodatkov o tem, ali so posamezni dokument že obiskali, ne glede na to, kje se nahaja referenca nanj. Tovrstno vizualno razlikovanje med hiperpovezavami je omogočal že eden prvih grafičnih spletnih brskalnikov, NCSA Mosaic. Ta je hiperpovezave prikazal v modri barvi, če so se te nanašale na stran, ki še ni bila obiskana, sicer pa jih je prikazal v vijolični barvi [81]. To prakso so prevzeli tudi drugi brskalniki in se je obdržala do danes.

Sčasoma se je splet iz dokumentnega skladišča razvil v platformo za aplikativne rešitve, ki je aplikacijsko logiko na začetku hranila na strežniški strani. Vendar je razvoj jezika

Slika 2.2: Izgled modificiranih hiperpovezav v brskalniku

JavaScript ta fokus prestavil na stran odjemalca. Selitev iz nadzorovanega v nenadzorovano okolje je povzročila mnoge varnostne pomisleke, kot so medsebojna izolacija aplikacij, zaščita pred zlonamerno programsko kodo ipd. [109]. Varnostna politika spleta že od brskalnika Netscape Navigator 2.0 temelji na politiki istega porekla (angl. *same-origin policy*) [93]. To pomeni, da lahko aplikacije JavaScript kumunicirajo samo s spletnim strežnikom, na katerem gostujejo. Ta princip piškotkom in aplikacijam JavaScript z različnimi stopnjami zaupanja omogoča tiho sobivanje v uporabniškem brskalniku brez medsebojnega motenja [51]. Politika istega porekla, ki se je na začetku nanašala samo na JavaScript, se postopoma širi tudi na druge varnostne nastavitve brskalnika [101]. Vendar se ta politika nikoli ni nanašala na hiperpovezave, saj bi nezmožnost medsebojnega povezovanja zmanjšalo uporabnost spletnih strani.

Razliko med že obiskanimi in neobiskanimi hiperpovezavami lahko avtorji spletnih strani spreminjajo in oblikujejo po svojih željah, in to z uporabo CSS-a. HTML-elementom lahko določimo izgled s pomočjo jezika CSS, ki ponuja tudi pseudo razrede, s katerimi lahko določimo izgled posameznega HTML-elementa v posebnem stanju, kot je npr. obiskana hiperpovezava (`a:visited`). Način oblikovanja videza lahko vidimo v izseku programske kode 2.1, izgled v brskalniku pa na sliki 2.2.

Izsek programske kode 2.1: Primer definiranja videza hiperpovezav z uporabo CSS-a

```
1 a:link      {color: #737373}
2 a:visited   {color: #3333ff}
```

Načeloma naj spletna stran ne bi bila sposobna ugotoviti, katere druge strani je uporabnik obiskal. Vendar pa kombinacija na videz nedolžnih lastnosti spleta spletnim stranem omogoča ugotavljanje, ali je uporabnik posamezno stran že obiskal, in posledično vsaj deloma rekonstruiranje zgodovine brskanja posameznega uporabnika [109].

Izsek programske kode 2.2: Osnovna JavaScript implementacija preverjanja obiskanih hiperpovezav [53]

```
var r1 = 'a {color: green}'
var r2 = 'a:visited {color: red}'

document.styleSheets[0].insertRule(r1, 0)
document.styleSheets[0].insertRule(r2, 1)

var aEl = document.createElement('a')
aEl.href = "http://foo.org"

var a_style = document.defaultView.getComputedStyle(aEl, "")

if(a_style.getPropertyValue("color")=='red')
    // hiperpovezava je ze bila obiskana
```

Avtor spletne strani lahko z uporabo metode JavaScript `getComputedStyle()` in s seznamom spletnih strani, katerih obisk želi preveriti, o uporabniku sklepa marsikaj. Za takšne napade, imenovane „analiza zgodovine“ (*angl. history sniffing*) [101], napadalci pogosto uporabljajo najpogosteje obiskane strani iz storitve Alexa [9] ter druge podobne storitve. Čeprav velikost nabora strani, ki jih je mogoče preveriti, ni omejena, so izkoriščevalci pogosto preverjali med 6 in 220 URL naslovov [54]. Olejnik *et al.* [85] je v svoji raziskavi zajel 164.043 različnih profilov zgodovine brskanja, od katerih je bilo 88 % edinstvenih (tj. tistih, ki se lahko dodelijo edinstvenemu uporabniku). V tabeli 2.1 lahko vidimo koeficiente priljubljenosti posameznih kategorij znotraj analiziranih profilov, ki so normalizirane glede na najpriljubljenejšo kategorijo – *Spletni iskalniki*.

Implikacije napadov, kot je omenjeni napad, so lahko v nekaterih primerih koristne, npr. za preverjanje obiskov strani, ki se lažno predstavljajo (angl. *phishing*), kar je koristno za strani z manjšo stopnjo tolerance varnostnih tveganj (npr. banke). Te strani lahko preverijo, ali je uporabnik prej obiskal nezaupljive strani; v tem primeru ga opozorijo na morebitno kompromitiranje podatkov [51]. Po drugi strani pa resnični

Tabela 2.1: 30 najbolj priljubljenih kategorij normaliziranih glede na spletne iskalnike (417.750) [85]

Kategorija	#	Kategorija	#
Spletni iskalniki	1,00	Družbena omreja	0,91
Trgovine	0,79	Računalniki / Internet	0,76
Novice / Mediji	0,74	Streaming / MP3	0,72
Zabava	0,67	Reference	0,51
Igre	0,30	Pornografija	0,27
Dražbe	0,24	Vlada / Pravno	0,24
Programska oprema	0,22	Blogi	0,21
Iskanje slik	0,12	Peer-to-peer	0,20
Elektronska pošta	0,17	Poslovno / Ekonomija	0,16
Šport	0,16	Finančne usluge	0,11
Prevzemanje datotek	0,10	Potovanja	0,09
Vsebina za odrasle	0,06	Izobraževanje	0,06
Internet telefonija	0,05	Klepet	0,05
Osebno / Zmenki	0,05	Internet radio / TV	0,05
Vozila	0,05	Restavracije / Hrana	0,04

napadalci, ki jih je raziskoval Jang [54], uporabnikom sledijo med obiskom spletnih strani zaradi oglaševalskih namenov in/ali ugotavljanja, ali so obiskali spletno stran konkurence. Weinberg *et al.* [109] sklepa, da zasebnostna in varnostna tveganja, ki jih beleženje in analiza zgodovine povzroči, odtehtajo morebitne ugodne možnosti.

Od leta 2010 se v specifikacijah CSS 2.1 [27] nahaja opomba, v kateri W3C opozarja na prej opisano varnostno pomanjkljivost, saj avtorji spletnih strani brez soglasja uporabnika pseudo razrede `:link` in `:visited` lahko zlorabljajo za ugotavljanje, katere strani je ta že obiskal. Prav tako specifikacije vključujejo zahtevo za brskalnike, ki predpisuje obravnavanje vseh hiperpovezav, kot da so neobiskane, ali implementacijo drugih ukrepov za ohranitev uporabnikove zasebnosti v času prikazovanja (angl. *rendering*) obiskanih in neobiskanih hiperpovezav.

2.2.3 Geolokacija

HTML5-standard definira visoko nivojski vmesnik (API) za dostop do informacij o lokaciji, kot sta zemljepisna širina in zemljepisna dolžina. Vmesnik je neodvisen od virov lokacijskih informacij in za določanje lokacije uporablja različne vire, od katerih se samo določene zanašajo na sistem GPS. Lokacijo je tako mogoče določiti na podlagi naslednjih podatkov: [90, 49]:

- Global Positioning System (GPS),
- IP-naslov,
- RFID,
- WiFi in Bluetooth MAC-naslovi,
- identifikator baznih postaj GSM/CDMA,
- uporabniški vnos.

Vmesnik je načrtovan tako, da omogoča enkratno (angl. „*one-shot*“) zahtevo za dostop do lokacijskih podatkov in tudi sprotno spremljanje posodobitev lokacije. Vendar zagotovila, da bo vmesnik podal natančno lokacijo [90], ni.

Implementacijsko geolokacijski vmesnik predstavlja objekt, ki lahko programsko pridobi lokacijo naprave in omogoča spletnem okolju dostop do nje [78]. Vsebuje tri metode:

`Geolocation.getCurrentPosition()`

Določa trenutno lokacijo naprave in vrača objekt `Position`, ki vsebuje podatke o lokaciji. Ta metoda implementira zahtevo za enkratni dostop do lokacijskih podatkov.

`Geolocation.watchPosition()`

Uporablja se za registracijo funkcije povratnega klica (angl. *callback function*), ki

jo v nadaljevanju kliče vsakič, ko se lokacija naprave spremeni. Metoda vrača ID-številko, s katero se lahko posamezna funkcija povratnega klica odjavi s pomočjo metode `Geolocation.clearWatch()`. Ta metoda implementira zahtevo za sprotno posodabljanje lokacijskih podatkov.

`Geolocation.clearWatch()`

Kot vhodni parameter sprejme ID-številko funkcije povratnega klica in jo odjavi s seznama funkcij, ki so obveščene o spremembah lokacije.

V izseku programske kode 2.3 lahko vidimo primer uporabe geolokacije. Ko se programska koda izvede v brskalniku, ta uporabniku izpiše sporočilo, v katerem ga obvesti, da spletno mesto zahteva dostop do njegove lokacije. Uporabnik dostop lahko dovoli ali onemogoči. V primeru da uporabnik dostop dovoli, se izvede funkcija povratnega klica (`prikaziLokacijo(lokacija)`), ki kot vhodni parameter dobi objekt, ki vsebuje podatke o uporabnikovi lokaciji.

Izsek programske kode 2.3: Primer uporabe HTML5 geolokacijskih storitev

```
1 if (navigator.geolocation)
2   navigator.geolocation.getCurrentPosition(prikaziLokacijo)
3 else
4   console.log("Vas brskalnik ne podpira geolokacijo.")
5
6 function prikaziLokacijo(lokacija) {
7   console.log("Zemljepisna sirina: " +
      lokacija.coords.latitude)
8   console.log("Zemljepisna dolzina: " +
      lokacija.coords.longitude)
9 }
```

V skoraj vseh primerih podatek, pridobljen s pomočjo geolokacijskega vmesnika, razkrije tudi lokacijo uporabnika naprave, s čimer bi se potencialno lahko ogrozila uporabnikova zasebnost. Implementacija specifikacije konzorcija W3C od proizvajalcev brskalnikov zahteva zagotovitev mehanizma, ki varuje uporabnikovo zasebnost. Ta

mehanizem mora zagotoviti, da podatki o lokaciji preko geolokacijskega vmesnika niso na voljo brez uporabnikovega izrecnega dovoljenja [90]. Prav tako specifikacije vzpodbujajo uporabo enkripcije, kar spletni brskalnik Chrome od različice 50.0 tudi zahteva, saj v novejših različica uporaba geolokacijskega vmesnika ni možna, če ni uporabljena varna HTTPS-povezava [22].

2.2.4 Stanje baterije

Vmesnik za stanje baterije (angl. *Battery Status API*) je brskalniški mehanizem, ki omogoča dostop do informacij, povezanih z upravljanjem napajanja [64].

Programski vmesnik `BatteryManager` je sestavljen iz atributov, ki hranijo podatke o stanju baterije: ali se baterija trenutno polni ali ne (`charging`); koliko časa je potrebno, da se baterija napolni (`chargingTime`); koliko časa lahko naprava s pomočjo baterije še deluje (`dischargingTime`); odstotek napolnjenosti baterije (`level`). Ta vmesnik vsebuje tudi določene metode za upravljanje dogodkov (angl. *event handlers*), ki se sprožijo ob spremembi stanja, priklopu ali izklopu polnilnika (`onchargingchange`), ob spremembi časa, ki je potreben, da se baterija napolni (`onchargingtimechange`), ob spremembi časa, ki je potreben, da se baterija izprazni (`ondischargingtimechange`), ter ob spremembi odstotka napolnjenosti baterije (`onlevelchange`) [64]. Do teh podatkov lahko dostopamo v programskem jeziku JavaScript preko klica metode `navigator.getBattery()`. V izseku kode 2.4 lahko vidimo primer izpisa stanja baterije, ki se sproži vsakič, ko se stanje spremeni.

Izsek programske kode 2.4: Primer uporabe HTML5 baterijskega vmesnika [64]

```
1 navigator.getBattery().then(function(battery) {
2   console.log(battery.level);
3   battery.onlevelchange = function() {
4     console.log(this.level);
5   };
6 })
```

API ne zahteva uporabniškega dovoljenja za dostop do podatkov o bateriji, kar pomeni,

da ga lahko uporablja katera koli spletna stran ali od tretje strani vključena skripta nanjo. Prav tako specifikacije od proizvajalcev spletnih brskalnikov ne zahtevajo implementacije mehanizma obveščanja, ki bi uporabnika obvestil v primeru, da se podatki o bateriji njegove naprave obravnavajo znotraj spletne strani [33]. To omogoča, da so vsi podatki, ki so izpostavljeni preko vmesnika za stanje baterije, dostopni brez uporabnikovega dovoljenja in zavedanja.

Specifikacije konzorcija W3C [64] so v poglavju „*Varnostni in zasebnostni premisleki*“ (angl. „*Security and privacy considerations*“) vsebovale stavek „*Razrkrite informacije imajo minimalen vpliv na zasebnost ali možnost „zajemanja prstnih odtisov“ (angl.* fingerprinting*) in zato je izpostavljen brez odobritve dovoljenja.*“. Kljub mnenju avtorjev specifikacije, da je vpliv na zasebnost minimalen, so se v nadaljevanju dopolnili z trditvijo, da „*brskalniki lahko zakrijejo (angl.* obfuscate*) vrednosti na način, da [spletne strani] ne morejo direktno vedeti, ali naprava nima baterije, ali se polni, ali prikazuje lažne vrednosti*“, kar kaže, da je kljub temu pričakovana določena stopnja tveganja, ki bi lahko sprožila potrebo po prikrivanju dejanskega stanja baterije [86].

V svoji raziskavi je Olejnik *et al.* [33] odkril, da je vmesnik v implementaciji v brskalniku Firefox in v operacijskem sistemu Linux omogočal „zajemanje prstnih odtisov“ in sledenje napravam, ki imajo baterije v kratkih časovnih intervalih. V raziskavi so namreč odkrili, da se odstotek napolnjenosti baterije v brskalniku Firefox in operacijskih sistemih Linux prikazuje kot število z dvojno natančnostjo[1] (angl. *double precision*). Na operacijskih sistemih Windows, Mac OS X in Android je ta podatek vseboval samo dve pomembni števki (angl. *significant digits*) (npr. 0.32). Z analizo izvorne kode brskalnika Firefox so ugotovili, da se stanje baterije pridobi s pomočjo orodja *UPower* [50], ki med drugim omogoča dostop do različnih podatkov o upravljanju napajanja naprave. S pomočjo določenih izračunov je tako možno pridobiti podatke o kapaciteti baterije. Avtorji raziskave so proizvajalcem brskalnika Firefox predlagali popravek, ki je bil sprejet in implementiran leta 2015.

Ameriško podjetje Uber je maja 2016 razkrilo, da so uporabniki z manj polno baterijo pripravljeni plačati višje cene za prevoz [57]. To je sprožilo zaskrbljenost, da bi lahko

[1]Primer, ki je bil opažen v raziskavi: 0.9301929625425652.

tudi druga podjetja zlorabljala informacije o statusu baterije z namenom povečevanja cen za uporabnike z nizkim stanjem baterije [19, 56].

2.2.5 Usmeritev in premikanje naprave

Vedno večje število naprav, ki dostopajo do spleta, lahko določa usmerjenost s pomočjo žiroskopa. To pomeni, da lahko pridobivamo podatke, ki kažejo spremembe usmerjenosti naprave. Še posebej mobilne naprave, kot so mobilni telefoni, tablice in pametne ure te podatke uporabljajo za samodejno obračanje zaslona, da ostaja v pokončnem položaju [77]. Vmesnik za usmeritev naprave se v W3C-specifikacijah nahaja od marca 2010. Prvi brskalnik, ki ga je delno implementiral, je bil Google Chrome v svoji različici 7, ki je bila izdana oktobra 2010 [22].

Specifikacije definirajo tri DOM dogodke [20]:

`deviceorientation`
: Podaja fizično usmeritev naprave, ki je izražena kot niz vrtljajev v lokalnem koordinatnem okvirju v obliki treh rotacijskih kotov (α, β, γ). Primer uporabe lahko vidimo v izseku programske kode 2.5.

`devicemotion`
: Podaja podatke o pospešku naprave, izražen v kartezičnih koordinatah (x, y, z) v koordinatnem okvirju, ki ga naprava definira. Prav tako podaja hitrost vrtenja naprave o lokalnem koordinatnem okvirju. Kadar je to izvedljivo, dogodek zagotavlja tudi pospešek težišča naprave.

`compassneedscalibration`
: Dogodek, ki se uporablja za obveščanje spletnih strani, da kompas, ki se uporablja za zgornja dogodka, potrebuje kalibracijo.

Izsek programske kode 2.5: Primer uporabe HTML5-vmesnika za usmeritev naprave [20]

```
1 window.addEventListener("deviceorientation", function(event)
      {
```

```
    var alpha = event.alpha
    var beta = event.beta
    var gamma = event.gamma
}, true);
```

Naprava, ki leži na horizonatlni podlagi z vrhom zaslona, usmerjenim proti zahodu, vrne naslednje podatke: {`alpha: 90, beta: 0, gamma: 0`}. Da bi pridobili usmeritev kompasa, je treba odšteti vrednost `alpha` od 360 stopinj.

Informacija, ki jo vmesnik podaja spletni strani, ni neobdelan senzorski podatek, ampak sestavljen visokonivojski podatek, ki je neodvisen od nižjeležečih strojnih virov. Skupni vir podatkov vključuje žiroskope, kompase in pospeškometre [20].

W3C meni, da omenjeni podatek ni dovolj občutljiv, da bi vmesnik moral zahtevati dovoljenje uporabnika za delovanje. Vendar so se pojavila akademska prizadevanja za izkoriščanje senzorjev na napravah z namenom „zajemanja prstnih odtisov“ naprave. Zato W3C razmišlja o možnosti implementacije uporabniškega dovoljenja ali vizualnih indikatorjev, ki bi uporabnika obvestili, da so senzorji v uporabi. Prav tako bi lahko implementacija takšnih varovalnih mehanizmov imela pozitiven učinek na življenjsko dobo baterije [20].

Kljub trditvam W3C, da dostopnost senzorjev za usmeritev in premikanje naprave ni varnostno kritična, je Mehrnezhad *et al.* [69] v svoji raziskavi pokazal morebitna tveganja. Z analizo vhodnih podatkov in obdelavo s pomočjo nevronskih mrež je bilo možno uganiti štirimestno PIN-kodo s stopnjo uspešnosti 74 %. V drugem in tretjem poskusu se je stopnja uspešnosti identifikacije povečala na 86 % in 94 %, kar predstavlja resno grožnjo varnosti uporabnikov.

2.2.6 Zvok in slika

Konzorcij W3C definira dve specifikaciji, ki se nanašata na uporabo naprav, ki zajemajo zvok in sliko. Specifikacija *HTML Media Capture* definira razširirtve HTML-obrazcev z možnostmi enkratnega zajemanja zvoka ali videa. Specifikacija *Media Capture and*

Streams pa definira nabor JavaScript API-vmesnikov, ki omogočajo dostop do lokalnih naprav za zajemanje zvoka in slike.

Uvedba *HTML Media Capture* specifikacije je razširila obstoječi element HTML `<input>` z novim atributom `capture`. Ta razvijalcem omogoča deklaracijo zahteve za dostop do kamere ali mikrofona kot obliko vnosa datoteke [84]. Vnos deluje tako, da razvijalec elementu `<input>` nastavi vrednost atributa `accept` na `image/*` in vrednost atributa `capture` na `"user"` ali `"environment"`, odvisno, ali želi uporabljati sprednjo ali zadnjo kamero na mobilni napravi.

Po drugi strani specifikacija *Media Capture and Streams* [2] definira način dostopa do multimedijskih naprav, kot sta mikrofon in kamera, s pomočjo programskega jezika JavaScript. Primer uporabe lahko vidimo v izseku programske kode 2.6.

Izsek programske kode 2.6: Primer uporabe HTML5-vmesnika za dostop do kamere [2]

```
<input type="button" value="Start" onclick="start()" id="
    startBtn">
<script>
 var startBtn = document.getElementById('startBtn');

 function start() {
     navigator.mediaDevices.getUserMedia({
         audio: true,
         video: true
     }).then(gotStream).catch(logError);
     startBtn.disabled = true;
 }

 function gotStream(stream) {
     stream.getTracks().forEach(function (track) {
         track.onended = function () {
             startBtn.disabled = stream.active;
         };
     });
```

```
19  }
20
21  function logError(error) {
22      log(error.name + ": " + error.message);
23  }
24  </script>
```

Proizvajalci brskalnikov zaradi varnostnih in zasebnostnih razlogov implementirajo pojavna okna za eksplicitno dovoljenje za dostop do kamere ali mikrofona, da s tem zadovoljijo pravnim zahtevam.

W3C proizvajalcem brskalnikov svetuje tudi implementacijo obveščevalnega mehanizma, ki uporabnika obvešča, da je posamezna vhodna naprava aktivna. Proizvajalcem je prav tako svetovana implementacija funkcije, ki uporabnikom omogoča izbiro vhodne naprave ter možnost izklapljanje zajemanja zvoka, ko se uporablja video vhod.

2.2.7 Senzor ambientne svetlobe

Senzor ambientne svetlobe podaja informacije o količini svetlobe v prostoru, v katerem se naprava nahaja. Za merjenje uporablja glavni vgrajeni detektor svetlobe naprave in rezultate podaja v enoti *lux* [61]. V izseku programske kode 2.7 vidimo primer uporabe senzorja.

Izsek programske kode 2.7: Primer uporabe HTML5-senzorja ambientne svetlobe [61]

```
1  let sensor = new AmbientLightSensor();
2  sensor.start();
3
4  sensor.onchange = function(event) {
5      console.log(event.reading.illuminance);
6  };
7
8  sensor.onerror = function(event) {
9      console.log(event.error.name, event.error.message);
```

```
10 };
```

W3C v svojih specifikacijah [61] v poglavju „*Varnostni in zasebnostni premisleki*“ (angl. „*Security and privacy considerations*“) navaja, da ni posebnih varnostnih in zasebnostnih premislekov glede uporabe senzorja ambientne svetlobe. Kljub temu so raziskovalci [13] pokazali, da senzor ambientne svetlobe v kombinaciji z drugimi mobilnimi senzorji, kot sta zvok in geolokacija, predstavlja platformo povezljivih podatkov, ki jih je mogoče uporabiti za združitev podatkov in izpeljavo logične lokacije.

Pogosto se zgodi, da v zaprtih prostorih GPS-geolokacija ne deluje dovolj natančno, saj različni materiali povzročajo motnje v signalih. V primeru da poznamo uporabnikovo makrolokacijo in imamo podatke o posameznih mikrolokacijah, lahko podatke, pridobljene s senzorja ambientalne svetlobe, uporabimo za sklepanje o trenutni mikrolokaciji. Tako lahko izvemo, ali se uporabnik naprave nahaja npr. v knjižnici, trgovini ali v gostilni (slika 2.3) [13].

Slika 2.3: Primerjava ambientalne svetlobe na različnih lokacijah [13]

2.2.8 Pisave

Spletne strani lahko poleg pisav, ki so dostopne preko spleta (npr. Google Fonts), uporabljajo tudi pisave, ki so nameščene lokalno na uporabnikovi napravi. Vsak operacijski sistem vsebuje privzeti nabor prednameščenih pisav, ki se lahko razlikuje med operacijskimi sistemi in med različicami posameznega operacijskega sistema. Ta privzeti nabor lahko vsak uporabnik spremeni drugače. Običajno se dodajajo nove pisave,

lahko pa se nabor zmanjša.

W3C v specifikacijah za CSS3 definira pravilo `@font-face` [32], ki avtorjem spletnih strani omogoča izbiro pisave, ki je najbližja načrtovanim ciljem za posamezno stran ne glede na omejitve, ki nastanejo zaradi izbire pisav, nameščenih na posamezni napravi. Pravilo se uporablja za določanje lokacije posamezne pisave, ki se lahko nahaja lokalno ali na spletu. Če posamezna pisava ni nameščena na uporabnikovi napravi, CSS pravilo `@font-face` omogoča prikaz vsebine spletne strani s pomočjo te pisave, ki je dostopna na spletu. V izseku kode 2.8 lahko vidimo primer uporabe pravila `@font-face`.

Izsek programske kode 2.8: Primer uporabe CSS pravila `@font-face` [104]

```
1 @font-face{
2     font-family: 'foo';
3     src : local("Meiryo"),
4     url("sample.ttf");
5 }
6 body{ font-family: 'foo'; }
```

Atributu `font-family` nastavimo poljubno ime, atributu `src` pa nastavimo lokacijo pisave. Če je nastavljenih več lokacij, se koda izvaja od leve proti desni, dokler vira ni možno najti. Če je vir mogoče najti, se nadaljnji viri ne upoštevajo. Za referenciranje (iskanje pisave) se uporabljata dve metodi: `local()` in `url()`. Če je lokacija pisave definirana z uporabo metode `local()`, brskalnik preveri, ali je pisava nameščena na lokalni napravi. Če pa je lokacija pisave definirana z uporabo metode `url()`, brskalnik pridobi pisavo iz povezave, ki je podana kot parameter metode [104].

Glede na to, da za slogovni jezik CSS velja, da ne predstavlja varnostne ali zasebnostne grožnje, posebni varnostni ukrepi niso predvideni. Napadalci lahko poskušajo naložiti določeni nabor pisav in s pomočjo seznama zahtev strežniku ugotovijo, katere pisave je brskalnik poskušal naložiti. Če pisave, za katere je strežnik prejel zahtevo, odštejejo od množice vseh pisav, ki so jih poskušali naložiti, lahko ugotovijo, katere pisave so nameščene na posameznem odjemalcu. Pod pogojem da je pripravljen in uporabljen dovolj velik nabor pisav, je tako možno identificirati posamezne naprave. Pomembno je

tudi, da za takšen napad ni treba izvajati JavaScript kode na odjemalcu, kar ga naredi popolnoma nevidnega za končnega uporabnika.

2.2.9 User Agent

HTTP-zahteva, ki jo brskalnik pošlje strežniku, vsebuje polje z nazivom `User-Agent`. Niz, vsebovan v polju, je oblikovan na naslednji način: `Mozilla/[različica] ([informacije o sistemu in brskalniku]) [platforma] ([detajli o platformi]) [razširitve]`.

```
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/57.0.2987.133 Safari/537.36
```

Slika 2.4: Primer *User agent* niza

Leta 1990 je Tim Berners - Lee razvil prvi spletni brskalnik z nazivom „*WorldWideWeb*“ [106]. Tri leta po tem dogodku je januarja 1993 ameriški Nacionalni center za superračunalniške aplikacije (angl. *National Center for Supercomputing Applications*) razvil spletni brskalnik *Mosaic*, ki mu pripisujejo zasluge za popularizacijo svetovnega spleta. Eden izmed soavtorjev Mosaica je ustanovil podjetje *Netscape*, ki je izdelalo komercialni spletni brskalnik s kodnim nazivom *Mozilla*, s katerim so želeli izpodriniti brezplačni Mosaic s trga. Ko je je bil izdan brskalnik *Netscape*, je njegov *User agent* niz vseboval naziv `Mozilla/1.0`, tega pa so ga obdržale tudi vse naslednje različice tega brskalnika.

Netscape je hitro postal dominantni brskalnik in je sčasoma pridobival različne nove funkcionalnosti, ki v Mosaicu niso delovale. Zato so spletne strani pri vsaki zahtevi preverjale, ali *User agent* niz vsebuje besedo `Mozilla`, in se na podlagi tega odločale, katero verzijo spletne strani bodo poslale v odgovoru. Avgusta 1995 se je na trgu pojavila konkurenca brskalniku Netscape – Microsoft Internet Explorer 1, ki je bil pravzaprav prirejena različica Mosaica z Microsoftovo licenco. Novi brskalnik je kopiral tudi vse funkcionalnosti, ki jih je podpiral Netscape, Vendar če spletne strani v *User*

agent nizu niso našle besede `Mozilla`, so kot odgovor poslale enostavnejšo različico HTML-kode, torej prikazale manj napredno spletno stran. Microsoft ni želel, da bi konkurenčni brskalnik deloval boljše kot njihov, in zato so začeli vključevati besedo `Mozilla` v svoj *User agent* niz. Temu so sledili tudi drugi brskalniki. Z vključevanjem Netscapovega kodnega imena so strežnikom in spletnim stranem želeli sporočiti, da je njihov brskalnik kompatibilen z vsemi funkcionalnostmi, ki jih podpira Netscape. [92, 108].

Leta 1998 so člani podjetja Netscape ustanovili družbo Mozilla, ki se je ukvarjala z brezplačno programsko opremo in odprtimi standardi. Mozilla je septembra 2002 izdala prvo različico brskalnika Firefox, ki temelji na izvorni kodi svojega predhodnika, brskalnika Netscape. Nov brskalnik je za prikaz spletnih strani uporabljal komponento Gecko in to besedo tudi vključil v *User agent* niz. Gecko so začeli uporabljati tudi drugi brskalniki, kar je povzročilo, da so spletne strani svoje „boljše“ verzije pošiljale tistim, ki so se predstavili kot Gecko, ostalim (predvsem Internet Explorerju) pa „slabše“. Brskalniki na operacijskem sistemu Linux so uporabljali komponento KHTML, ki je bila primerljiva z Gecko, a so kljub temu od strežnikov dobili enostavnejše verzije spletnih strani, kar je bil razlog dodajanja niza „`like Gecko`“ v *User Agent*. Januarja 2003 je Apple izdal svoj brskalnik Safari, ki je uporabljal KHTML, a ga je razširil z novimi funkcionalnostmi in ga poimenoval WebKit. Zaradi kompatibilnosti je njegov *User agent* vključeval niz „`AppleWebKit/85.7 (KHTML, like Gecko)`“. Tudi eden izmed novejših brskalnikov Google Chrome (2008) temelji na komponenti WebKit. Ker je Google želel, da se strani prikazujejo kot v Safariju, se Chrome predstavlja kot KHTML, saj se KHTML predstavlja kot Gecko. Ker se vsi brskalniki predstavljajo kot Mozilla, se v svojem *User Agent* nizu Google Chrome predstavlja kot `Mozilla/5.0 ([OS]) AppleWebKit/[različica] (KHTML, like Gecko) Chrome/[različica] Safari/[različica]` [108]. Primer *User Agent* niza za brskalnik Google Chrome 57.0.2987.133, ki deluje na operacijskem sistemu Linux, lahko vidimo na sliki 2.4.

JavaScript objekt `Navigator` vsebuje atribute in metode, s katerimi lahko dostopamo do različnih informacij o brskalniku. Med drugimi atribut `navigator.userAgent`

omogoča prikaz *User Agent* niza. Primer uporabe lahko vidimo v izseku programske kode 2.9.

Izsek programske kode 2.9: Primer uporabe *User agent* niza v programskem jeziku JavaScript

```
1 var info = navigator.userAgent
2 console.log(info)
```

Niz *User agent* vsebuje podatke, s katerimi je možno razlikovati različne uporabnike med seboj, saj pogosto uporabljajo različne verzije brskalnikov.

2.3 Identificiranje spletnih uporabnikov

Do sedaj smo obravnavali različne metode, ki spletni strani omogočajo zbiranje podatkov o brskalnikih. Kombinacija podatkov, zbranih z uporabo teh metod, tvori osnovo, s pomočjo katere je mogoče identificirati brskalnik oziroma njegovega uporabnika.

Identificiranje uporabnika brez njegovega eksplicitnega sodelovanja je cilj, h kateremu stremijo mnoge spletne strani. Identifikacija obiskovalca namreč omogoča spletnim stranem boj proti goljufijam in odkrivanje kršitev pogojev uporabe (angl. *terms of use*). [79].

V začetku spleta je bilo možno uporabnike učinkovito identificirati po IP-naslovih njihovih računalnikov ali naprav [39]. Kasneje, ko je uporaba dinamičnih IP naslovov in NAT-a (*Network Addres Translation*) postala široko uporabljena, podatek ni več zadoščal. Namesto tega se je sledenje brskalniških navad uporabnika izvajalo s pomočjo shranjevanja identifikatorja v piškotek v brskalniku. To je omogočalo identifikacijo uporabnika pri vsaki HTTP zahtevi, ki je vsebovala ta piškotek [21].

Spletna stran lahko zajame lastnosti posameznega brskalnika, ki skupaj tvorijo edinstveni ali skoraj edinstveni identifikator [67, 39]. Ta globalno edinstven psevdonimni identifikator naprave je mogoče kodirati v katero koli spletno tehnologijo s stánjem (angl. *stateful*), dokler ta lahko hrani vsaj $\lfloor \log_2 n \rfloor + 1$ bitov, kjer n predstavlja število

vseh naprav[2], ki so povezane na splet [68]. Po Gartnerjevi napovedi [43] bo leta 2017 na splet povezanih 8,4 milijard naprav, kar je 31 % več kot leta 2016 ter naj bi doseglo do leta 2020 20,4 milijarde. Število bitov, ki je potrebno za shranjevanje edinstvenega identifikatorja naprave, je trenutno 33 bitov (izračun 2.2). Iz izračunov 2.1, 2.2 in 2.2 lahko vidimo, da se število bitov ni povečalo od leta 2012. Predvideva se, da se bo število bitov povečalo na 34, ko bo število naprav, povezanih na splet, doseglo $\approx$ 17 milijard.

$$\left\lfloor \log_2 5 \cdot 10^9 \right\rfloor + 1 = \lfloor 32.2192809489 \rfloor + 1 = 33\, bitov \tag{2.1}$$

$$\left\lfloor \log_2 8,4 \cdot 10^9 \right\rfloor + 1 = \lfloor 32.9329767637 \rfloor + 1 = 33\, bitov \tag{2.2}$$

$$\left\lfloor \log_2 20,4 \cdot 10^9 \right\rfloor + 1 = \lfloor 34.2478501011 \rfloor + 1 = 35\, bitov \tag{2.3}$$

2.3.1 „Zajemanje prstnih odtisov“ brskalnika

Konzorcij W3C [37] definira pojem „zajemanja prstnih odtisov“ (angl. *browser fingerprinting*) kot sposobnost spletne strani, da identificira ali reidentificira obiskovalca, brskalnik ali napravo preko konfiguracijskih nastavitev ali drugih lastnosti. Podobno definicijo lahko najdemo tudi v dokumentih organizacije IETF [28].

Postopek „zajemanja prstnih odtisov“ brskalnika se lahko uporablja kot varnostni ukrep (npr. kot sredstvo avtenticiranja uporabnikov), vendar predstavlja tudi potencialno grožnjo uporabniški zasebnosti na spletu. „Zajemanje prstnih odtisov“ se lahko uporablja za:

- identifikacijo uporabnika,
- korelacijo uporabnikovih brskalniških aktivnosti znotraj seje in med sejami,
- sledenje uporabnikom brez transparentnosti in kontrole.

[2]Mayer *et al.* [68] navaja številko 5 milijard, a ta je veljavna za leto 2012.

Identifikacija uporabnika . Obstajajo mnogi razlogi, zakaj bi uporabniki na spletu želeli ostati anonimni ali neidentificirani: zaskrbljenost glede nadzorovanja, osebna psihična varnost in zaskrbljenost glede diskriminacije na podlagi tega, kar preberejo ali napišejo na spletu. V trenutku ko je možno povezati brskalniški „prstni odtis“ z identificirajujočo informacijo (kot je npr. pravo ime), lahko aplikacija ali ponudnik storitev identificirajo sicer anonimnega uporabnika. Uporabniki, ki jih skrbi nadzorovanje vlade, lahko uporabljajo orodja, kot je Tor, s čemer si bodo zagotovili anonimnost na ravni omrežja, ampak ta jih ne ščiti pred „zajemanjem prstnih odtisov“ brskalnika in morebitnim koreliranjem z njihovo spletno aktivnostjo [37].

Korelacija brskalniških aktivnosti. „Zajemanje prstnih odtisov“ brskalnika lahko povzroča tveganje zasebnosti, če ni vpletene realne identitete uporabnikov. Uporabniki so presenečeni ali zaskrbljeni, ko spletna stran (ali celo različne spletne strani) zazna in poveže njihove večkratne obiske. Spletne strani to običajno počnejo z namenom izgradnje profila uporabnika ali beleženja njegove zgodovine. Tovrstno zaznavanje se lahko zgodi tudi brez seznanjenosti uporabnika ali njegovega soglasja. Tudi ukrepi za preprečevanje (kot je npr. brisanje spletnih piškotkov) ne morejo onemogočiti korelacije [37]. Prav tako je možno, da različne spletne strani kombinirajo informacije o posameznem uporabniku tudi v primeru, da je v brskalniških nastavitvah onemogočeno deljenje piškotkov med različnimi stranmi, saj je „prstni odtis“ brskalnika relativno unikaten in isti na vseh spletnih straneh [15].

Sledenje uporabnikom brez transparentnosti in kontrole. V nasprotju z drugimi mehanizmi za vzdrževanje seje (npr. spletni piškotki), ki jih definirajo spletni standardi, „zajemanje prstnih odtisov“ brskalnika omogoča zbiranje podatkov o uporabnikovi aktivnosti brez jasnega indikatorja, da se to pravzaprav dogaja. Transparentnost je lahko pomembna za končne uporabnike v smislu razumevanja, kaj se dogaja z njihovimi podatki. Ravno tako je pomembno poudariti, da uporabnik nima učinkovitega nadzora, saj brskalniškega „prstnega odtisa“ ni mogoče izbrisati ali ponovno ponastaviti [37].

Poznamo dva tipa „zajemanja prstnih odtisov“: pasivno in aktivno. Pasivno temelji na karakteristikah, ki jih je možno opazovati znotraj vsebine HTTP-zahteve brez uporabe kakršne koli programske kode, ki bi se izvajala pri odjemalcu. Pasivno „zajemanje prstnih odtisov“ v svoji trivialni obliki vključuje spletne piškotke (pogosto unikatni identifikator, ki je poslan v HTTP-zahtevi) in IP-naslov skupaj z drugimi omrežnimi podatki. Kot primer navedimo niz *User agent* (poglavje 2.2.9), ki je del glave HTTP-zahteve, identificira brskalnik, različico brskalnika, operacijski sistem idr. V nekaterih okoliščinah je že kombinacija IP-naslova in niza *User agent* zadostni pogoj za enolično identifikacijo brskalnika. Po drugi strani aktivno „zajemanje prstih odtisov“ zajema vse metode, pri katerih spletna stran izvaja JavaScript ali drugo programsko kodo na lokalnem odjemalcu z namenom opazovanja dodatnih karakteristik brskalnika. Ključna razlika je v tem, da aktivno „zajemanje prstih odtisov“ deluje na način, ki ga je potencialno možno zaznati [37].

2.3.2 Sorodne raziskave

Raziskovalni projekt Panopticlick [39] je demonstriral izvedljivost „zajemanja prstnih odtisov“ brskalnikov oz. naprav z namenom sledenja po spletu z merjenjem entropije, ki je prisotna v različnih lastnostnih brskalnika, kot so ločljivost zaslona, nameščeni vtičniki, nameščene pisave ipd. V raziskovalnem vzorcu, sestavljenem iz uporabnikov, ozaveščenih o njihovi zasebnosti, so odkrili, da je 83.6 % brskalnikov možno identificirati po edinstvenem „prstnem odtisu“, nadaljnjih 5.3 % pa je imelo množico anonimnosti (angl. *anonimity set*) velikosti 2. Množica anonimnosti je množica tistih, za katere obstaja neničelna verjetnost, da se nahajajo znotraj množice, ni pa jih mogoče posamezno identificirati [98]. Drugi raziskovalci so demonstrirali mnoge načine identifikacije brskalnika z različnimi lastnostmi, kot so zamik ure [60], metrike pisav [40], karakteristike omrežnih protokolov [24], performanse JavaScripta [75] in WebGL-a ter HTML elementa `canvas` [76].

V zadnjem času so študije, ki so merile razširjenost „zajemanja prstnih odtisov“ brskalnikov na spletu [3, 4, 82], pokazale, da se sporne prakse, kot so izogibanje posredniškim

strežnikom (angl. *proxy circumvention*) ali različne tehnike zakrivanja, na spletnih straneh uporabljajo pogosto [86].

Kljub podatkom o odstotkih enoličnih identifikacij brskalnikov, ki smo jih pridobili iz različnih raziskav, bi si morda želeli izvedeti tudi, koliko globalno edinstvenih brskalnikov obstaja. Mayers pravi [67], da je skoraj nemogoče sklepati kar koli o *globalni* edinstvenosti brskalniških „prstnih odtisov“, saj je glede na multinominalni teorem maksimalna verjetnost, da je kateri koli „prstni odtis“ v vzorcu velikosti N edinstven, naslednja:

$$P(f_i) = \frac{1}{N} \tag{2.4}$$

„Prstni odtis“ s takšno verjetnostjo je zelo oddaljen od edinstvenosti v globalni množici brskalnikov G, ker je $G \gg N$ [39].

2.3.3 Matematična osnova

„Zajemanje prstnih odtisov“ lahko razumemo kot poskus identifikacije posameznega brskalnika v globalnem prostoru vseh brskalnikov. V nadaljevanju bomo pogledali načine izračuna entropije v primeru uporabe algoritma za „zajemanja prstnih odtisov“. Entropija ali Shannonova entropija je količina, ki meri negotovost izida poskusa, povezanega s slučajno spremenljivko [30].

Denimo, da imamo algoritem za „zajemanja prstnih odtisov“ brskalnikov $F(\cdot)$. Ob nastanku nove instance brskalnika v globalnem prostoru nameščenih brskalnikov njegovemu izhodu $F(x)$ sledi diskretna verjetnostna funkcija $P(f_n), n \in [0, 1, .., N]$. Lastno informacijo posameznega izhoda iz algoritma izračunamo na naslednji način:

$$I(F(x) = f_n) = -\log_2(P(f_n)) \tag{2.5}$$

V našem primeru I merimo s številom bitov, kar je posledica izbire dvojiškega logaritma. En bit je množina informacije, ki je odgovor na vprašanje z dvema možnima

odgovoroma. Entropija distribucije $P(f_n)$ je pričakovana vrednost prejšnjega izraza za vse brskalnike, kar izrazimo takole:

$$H(F) = -\sum_{n=0}^{N} P(f_n) \log_2(P(f_n)) \tag{2.6}$$

Entropijo naključne spremenljivke lahko razumemo kot količino informacije o identiteti objekta, ki mu želimo dodeliti edinstveni identifikator. Vsak bit informacije razpolovi število možnosti za zapis. Če spletno stran redno obiskuje množica X različnih brskalnikov, bi intuitivno lahko ocenili, da bi posamezni brskalnik $x \in X$ bilo možno enolično prepoznati, če je $I(F(x)) \gtrapprox \log_2 |X|$. Namesto intuicije lahko uporabimo binominalno distribucijo s pravilnim intervalom zaupanja, a se izkaže, da v primeru realnega „zajemanja prstnih odtisov“ veliko več negotovosti izhaja iz ocen $P(f_n)$, vsaj v primeru poskusa odgovarjanja na vprašanje o tem, kateri brskalniki so edinstveno prepoznavni [39].

Eckersley jemlje entropijo kot merilo variacij [39, 40]. „Prstni odtis“ brskalnika sestavlja več različnih komponent $s \in S$, oziroma rezultati meritev teh komponent $F_s(\cdot)$. Njegovo delo razširjata Fifield in Egelman [40], ki predlagata združitev komponentnih meritev $s \in S$ v vektor $\mathcal{S}$. Entropija vektorja $\mathcal{S}$ je vsota entropij vseh komponent, v kateri je vsaka komponenta skalirana po dolžini.

$$H(\mathcal{S}) = -\sum_{s \in \mathcal{S}} \frac{|s|}{\sum_{t \in \mathcal{S}} |t|} H(s) \tag{2.7}$$

2.4 Uporaba profiliranja za oglaševanje

Veliko truda se vlaga v analizo in klasifikacijo aktivnosti uporabnika na spletnih straneh (*on-portal*) in v celotni navigacijski seji (*off-portal*) z namenom profiliranja in izboljšanja uporabniške izkušnje (npr. personalizacija) ter izboljšanja kakovosti spletne strani [87]. Seveda pa želijo skrbniki spletnih strani od profiliranja imeti finančne koristi, predvsem v navezavi s ciljnim oglaševanjem, v katerem se uporabniku prikazujejo

oglasi z njegovega interesnega področja.

Večina oglasov na spletu so kratka besedilna sporočila, ki so običajno označena kot „sponzorirane povezave“. Dve glavni besedilni oglaševalski kategoriji na spletu sta [23] sponzorirano oglaševanje v iskanju in kontekstualno oglaševanje. Sponzorirano oglaševanje prikaže oglase na strani, ki vsebuje rezultate iskanja. Tovrstno oglaševanje podpirajo vsi večji spletni iskalniki in hkrati delujejo kot iskalniki in oglaševalske agencije. Kontekstualno oglaševanje postavi oglase na spletne strani tretjih strank [6], pri čemer je njihova uspešnost odvisna od integracije z domorodno vsebino in od stopnje prilagajanja interesom obiskovalca.

Vsak oglas ima svojo ciljno publiko – skupino uporabnikov, ki jim je namenjen. Naročniki oglasov ocenjujejo kakovost dela oglaševalskih agencij na podlagi rezultatov, ki jih oglasi dosežejo. Zato oglaševalske agencije poskušajo na različne načine pridobiti čim več podatkov o uporabnikih posamezne platforme, na kateri oglašujejo, saj tako oglase lahko postavljajo ciljno in s tem dosežejo večji učinek. Podatke, ki jih je možno pridobiti s pomočjo „zajemanja prstnih odtisov“, se lahko uporabi za ciljno oglaševanje, in sicer tako, da pridobimo podatke o uporabnikovih interesih (npr. obiskane spletne strani).

2.5 Možnosti ukrepanja in varovanja zasebnosti

Razvijalci in raziskovalci kot odgovor na pomisleke glede zasebnosti posameznika na spletu še naprej objavljajo rešitve, ki uporabnikom omogočajo različne stopnje zasebnosti. Dobro znan tak primer je *način zasebnega brskanja* ali *brskanje brez beleženja zgodovine* (angl. *private browsing mode*) [8].

Veliko entropije vsebujejo mikro različice (natančne različice) brskalnikov in različnih vtičnikov. Očitna rešitev tega problema je zmanjšanje natančnosti verzioniranja. Namesto `Java 1.6.0_17` je bolje uporabiti `Java 1.6`. Vzrok za natančno verzioniranje je razhroščevanje in retrospektivno sledenje napakam v zapisnikih (angl. *logs*). To je sicer razumljivo, vendar takšna odločitev zmanjšuje stopnjo zasebnosti uporabnika. Interval med skrajnima poloma – veliko stopnjo možnosti razhroščevanja in veliko sto-

pnja zasebnosti – trenutno konvergirajo proti možnosti razhroščevanja. Morda bi bilo smiselno ta trend spremeniti, še posebej v primeru uporabe *načina zasebnega brskanja* [39].

„Zajemanje prstnih odtisov“ lahko otežimo z zmanjšanjem raznolikosti med sistemi ali z zmanjšanjem možnosti merjenja teh razlik [39]. Enostavna ideja za zmanjšanje variabilnosti je vključitev standardnega nabora pisav v brskalnike [76]. Tako bi vse spletne strani lahko imele na razpolago le standardni nabor pisav (poleg vseh pisav, ki so dostopne preko spleta), ki je v vseh brskalnikih enak. S tem bi bistveno zmanjšala entropija, ki nastane zaradi raznolikosti med posameznimi sistemi.

Uspeh zmanjševanja tveganja lahko razdelimo v tri skupine [37]:

Zmanjšanje prostora za „zajemanje prstnih odtisov“

Odstranitev vira entropije ali dostopnih lastnosti, ki se lahko uporabljajo za „zajemanje prstnih odtisov“.

Povečevanje množice anonimnosti

S standardizacijo, konvencijami in skupnimi implementacijami povečevanje skupnih značilnosti posameznih konfiguracij lahko bistveno zmanjša verjetnost unikatnosti.

Možnost zaznavanja

Dodajanje možnosti zaznavanja „zajemanja prstnih odtisov“ (zlasti pri odjemalcu), saj brskalnik tako lahko ukrepa in prepreči nedovoljen poseg.

Prostor za „zajemanje prstnih odtisov“ (angl. *fingerprinting surface*) brskalnika je množica karakteristik, ki jih je mogoče opazovati in jih uporabiti za identifikacijo uporabnika, brskalnika ali naprave [37]. Fifield in Egelmanov [40] ta prostor množice karakteristk $s \in S$ opisujeta kot vektor $\mathcal{S}$. Avtorji specifikacij v konzorciju W3C poskušajo najti pravo razmerje med novimi funkcionalnostmi in prej omenjenim prostorom.

Specifikacije lahko ublažujejo možnost „zajemanja prstnih odtisov“ s standardizacijo: z definicijo konsistentnega obnašanja skladne implementacije ne bodo vsebovale variacije,

ki bi jih bilo mogoče uporabiti za unikatno identifikacijo. Prav tako je predlagana tudi randomizacija nekaterih značilnosti brskalnika, čeprav se predvideva, da bo veliko lažje vpeljati večjo stopnjo standardizacije kot randomizacije. Težko je oceniti, kako dobro bi randomizacija delovala kot ublažitveno sredstvo, saj je njena implementacija lahko „draga“ z vidika uporabnosti (posledično lahko nastopijo različne manifestacije v funkcionalnosti in dizajnu, predvsem nezaželene), procesiranja (generiranje naključnih števil) in razvoja (stroški vpeljave novih varnostnih popravkov) [37].

W3C vlaga veliko truda v izdelavo specializiranih metodologij za ocenjevanje zasebnosti [38], kar vključuje mnenje javnosti o pričakovanjih glede zasebnosti [83] in definiranje novih skupin in procesov za vrednotenje zasebnosti [36, 97]. Vendar se postavlja vprašanje, zakaj je treba definirati standarde zasebnosti in jih ne prepustiti proizvajalcem brskalnikov. Morda bo trg omogočil uporabnikom izbiro brskalnika, ki zagotavlja najboljše razmerje med zasebnostjo in funkcionalnostjo [86]. Čeprav je ta argument zapeljiv, je dejstvo, da je bolje, da standardi narekujejo način dela razvijalcev, saj je treba zagotoviti interoperabilnost. Nekateri standardi pa neizogibno vplivajo tudi na zasebnost [29]. Prav tako pa standardi omogočajo minimalno raven zasebnosti med vsemi implementacijami brskalnikov [5].

3 Analiza pridobivanja podatkov o uporabniški naprav s pomočjo spletnega brskalnika

V tem poglavju bomo predstavili spletno aplikacijo, ki je nastala v okviru magistrskega dela. Prav tako smo zasnovali in izvedli anketo, povezano s spletno aplikacijo. Z obema metodama smo preverjali hipoteze magistrskega dela. V nadaljevanju bomo najprej predstavili razvoj spletne aplikacije in kasneje zasnovo in izvedbo ankete. Na koncu sledijo tudi ugotovitve, ki smo jih pridobili s pomočjo analize pridobljenih podatkov.

3.1 Anketni vprašalnik

Anketa je kavzalna neeksperimentalna metoda, s katero zbiramo podatke od množice ljudi [45]. Definirajo ju tudi kot sistematično metodo zbiranja podatkov na podlagi (vzorca) enot z namenom konstruiranja kvantitativnih opisov značilnosti širše populacije, katere člani so preučevane enote [46]. Za zbiranje podatkov se običajno uporablja vprašalnik ali strukturiran intervju.

Na podlagi informacij, ki smo jih pridobili med preučevanjem literature in povezanih raziskav, smo pripravili anketni vprašalnik, saj takšen način zbiranja podatkov omogoča vzporedno izvajanje in je izvedljiv tudi v elektronski obliki.

Anketa lahko vsebuje tako kvantitativne kot kvalitativne podatke, čeprav običajno pre-

vladujejo kvantitativni. Pri izdelavi anketnega vprašalnika smo se odločili za zbiranje kvalitativnih podatkov.

Za merjenje nedemografskih odgovorov smo uporabili Likartovo lestvico. To je večtočkovna lestvica, ki se najpogosteje uporablja v sociooloških študijah. Običajno se uporablja za merjenje stališč, in sicer tako da se anketirancem ponudi spekter možnih odgovorov na podano vprašanje ali izjavo [26]. Lestvica je občutljiva za uporabo pri različnih vprašanjih, saj izraz „strinjam se“ ali „ne strinjam se“ lahko predstavlja tako pozitiven kot negativen odnos. Pri vprašanjih je treba biti pazljiv na pojav dvojnih negacij v kombinaciji s ponujenimi odgovori. Odločili smo se za uporabo 5-stopenjske lestvice, ki jo lahko vidimo v tabeli 3.1. Lestvico smo označili tako, da smo poimenovali vsako točko, ne pa le skrajnje vrednosti na lestvici.

Tabela 3.1: Likertova 5-točkovna lestvica

Sploh se ne strinjam	Se ne strinjam	Se niti ne strinjam niti nisem proti	Se strinjam	Se popolnoma strinjam
1	2	3	4	5

Pri vprašanjih, ki vključujejo pojme, za katere obstaja možnost, da jih povprečen uporabnik ne pozna, smo 5-stopenjski Likertovi lestvici dodali še možnost izbire odgovora „*Pojma ne poznam*“.

Anketo smo začeli z nagovorom, v katerem smo opisali namen raziskave, anketirancem zagotovili zaupnost podatkov ter se jim zahvalili za sodelovanje. V nadaljevanju smo vprašalnik razdelili v tri dele. V prvem delu smo anketirance povprašali o njihovih demografskih podatkih, drugi del se je nanašal na informacije o spletnih navadah in ravnanju na spletu, v tretjem delu pa smo želeli izvedeti kaj več o stopnji zavedanja uporabnikov o dostopnosti podatkov o njihovih napravah.

V prvem delu vprašalnika, ki ima naslov *Demografski podatki*, smo anketiranca vprašali o spolu, starosti (izbira med ponujenimi starostnimi skupinami), dokončani stopnji izobrazbe (izbira med *KLASIUS-SRV* [102], kategorijami izobraževanja) in trenutnim statusom (dijak, študent, zaposlen . . .). Postavili smo tudi nekaj pogojnih vprašanj,

saj smo od dijakov želeli izvedeti, katero srednjo šolo obiskujejo, od študentov pa, na kateri fakulteti in na katerem študijskem programu študirajo.

Drugi del vprašalnika z naslovom *Ravnanje na spletu* je sestavljen iz 6 trditev, ki se nanašajo na uporabnikove izkušnje na spletu, zavedanje o nevarnostih, anonimnosti in na socialna omrežja. Prav tako smo želeli izvedeti, koliko je uporabnik pripravljen dovoliti spletni strani dostop do svoje lokacije in kamere. V tem delu vprašalnika smo za merjenje odgovorov uporabili 5-stopenjsko Likertovo lestvico brez možnosti „*Pojma ne poznam*“.

Skupina trditev, ki so temelj naše raziskave, se nahaja v tretjem delu vprašalnika z naslovom *Dostopnost podatkov*. Anketirancem smo na podali naslednjo izjavo: *Menim, da je možno, da spletna stran dostopa brez mojega dovoljenja do naslednjih podatkov o moji napravi:* našteli smo jim 17 različnih podatkov, za katere so na 5-stopenjski Likertovi lestvici z dodano možnostjo izbire odgovora „*Pojma ne poznam*“. Anketiranci so izbrali stopnjo strinjanja z navedeno trditvijo v kombinaciji s posameznim podatkom.

Od 17 naštetih podatkov jih spletna stran lahko 13 pridobi brez uporabnikovega dovoljenja. Preostanek podatkov smo uporabili kot kontrolna vprašanja (*Moje polno ime*, *Zgodovina brskanja (spletne strani, ki sem jih že obiskal)*, *Naslov MAC*, *Telefonska številka*).

Na koncu ankete smo se anketirancem zahvalili za sodelovanje ter jih prosili, ali lahko kliknejo na podano hiperpovezavo in sodelujejo v praktičnem delu raziskave.

Da bi omogočili sledljivost anketirancev v anketi in spletni aplikaciji, smo vsakemu dodelili edinstveni identifikator UUID. UUID (angl. *Universally unique identifier*) je identifikator, ki je edinstven glede na prostor vseh identifikatorjev UUID. Znotraj orodja za izdelavo ankete smo uporabili možnost dodajanja programske kode JavaScript in za vsakega anketiranca generirali identifikator UUID ter ga hkrati shranili v posebno vnosno polje ankete, ki anketirancem ni bilo vidno, in v lokalno shrambo brskalniške seje (angl. *Session Storage*). Način generiranja identifikatorja UUID lahko vidimo v izseku programske kode 3.1.

Izsek programske kode 3.1: Kreiranje edinstvenega identifikatorja uporabnika (UUID)

```
function guid() {
  function s4() {
    return Math.floor((1 + Math.random()) * 0x10000)
      .toString(16)
      .substring(1);
  }
  return s4() + s4() + '-' + s4() + '-' + s4() + '-' + s4()
     + '-' + s4() + s4() + s4()
}
```

Na zaključni strani ankete smo anketirancem podali hiperpovezavo na spletno aplikacijo ter ji dodali tudi GET-parameter *UUID*, s pomočjo katerega smo zagotovili sledljivost posameznega uporabnika v anketi in spletni aplikaciji.

Anketni vprašalnik smo izdelali v orodju *LimeSurvey*, in sicer v različici 2.65. Orodje smo namestili na računalnik z operacijskim sistemom Ubuntu Server 16.04, za delovanje orodja *LimeSurvey* pa pripravili spletni strežnik Apache ter podatkovno bazo MySQL.

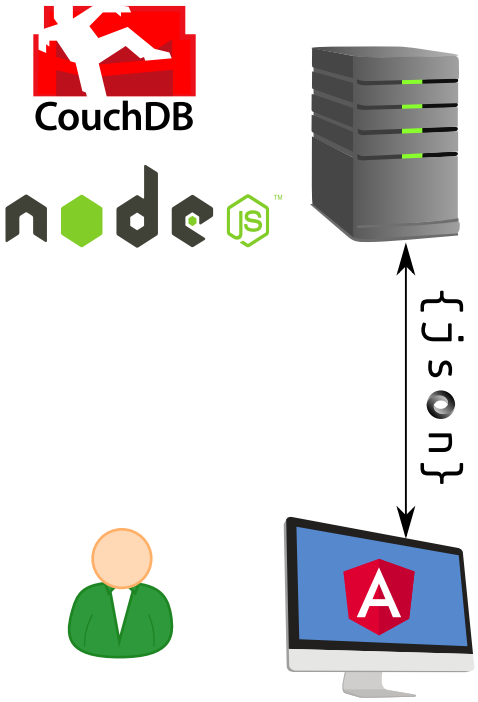
3.2 Spletna aplikacija za dostop do podatkov o brskalniku

Izdelali smo spletno aplikacijo, s pomočjo katere smo ugotavljali, katere podatke o brskalniku spletna stran lahko pridobi. Ta pridobiva podatke o obiskovalcih na dva načina. Del podatkov zajame brez kakršnekoli interakcije z uporabnikom, torej brez njegovega dovoljenja ali zavedanja, za uspešen dostop do drugega dela podatkov pa je potrebno uporabnikovo dovoljenje.

3.2.1 Načrtovanje aplikacije

Aplikacija je razdeljena na dva dela: sprednje ogrodje (angl. *frontend*) in zadnje ogrodje (angl. *backend*). Sprednje ogrodje vsebuje uporabniški vmesnik in programsko kodo, ki jo izvaja odjemalec. Programsko kodo zadnjega ogrodje izvaja strežnik.

Namen sprednjega ogrodja, ki se v celoti izvaja znotraj brskalnika odjemalca, je načrtovana z namenom pridobivanja čim večjega števila podatkov o uporabniku. Uporabniški vmesnik smo razdelili na dve strani: prva stran, ki se uporabniku prikaže, od njega zahteva dovoljenje za dostop do kamere in lokacije, druga stran pa ne zahteva nobenega dovoljenja, ampak uporabniku le prikaže podatke o njegovi napravi, te pa je možno pridobiti brez njegovega dovoljenja.



Slika 3.1: Komponente spletne aplikacije

Zadnje ogrodje aplikacije smo pripravili z namenom zbiranja podatkov. Po prihodu uporabnika na začetno stran spletne aplikacije smo zadnjemu ogrodju poslali podatek o tem, ali je uporabnik dovolil dostop do virov, za katere je spletna aplikacija dostop zahtevala. Prav tako smo po uporabnikovem prihodu na drugo stran spletne aplikacije poslali tudi podatke, ki jih je možno pridobiti brez uporabnikovega dovoljenja in

vedenja.

Zadnje ogrodje spletne aplikacije smo povezali s dokumentno podatkovno bazo, v katero smo shranjevali uporabniške odzive in pridobljene podatke.

Za spletno aplikacijo smo pridobili spletni naslov `spo.um.si`, kar je bil pogoj za pridobitev certifikata TLS izdajatelja kvalificiranih digitalnih potrdil *Let's Encrypt* [1, 7], o katerem bomo govorili v nadaljevanju.

3.2.2 Priprava aplikacijskih ogrodij

Za namestitev spletne aplikacije smo pripravili računalnik z operacijskim sistemom Ubuntu Server 16.04, na katerega smo namestili spletni strežnik Apache 2.4.18. Za Apache smo se odločili, ker je že od aprila 1996 najpriljubljenejši HTTP-strežnik na celem svetu, saj je na njem danes postavljenih 49.5 % spletnih strani [107]. V Sloveniji je Apache uporabljen za celo 79,09 % spletnih strani [35].

Prav tako smo na računalnik namestili tudi eno izmed najpriljubljenejših strežniških JavaScript ogrodij – Node.js. Ogrodje je namenjeno razvoju visokozmogljivih aplikacij s sočasnim izvajanjem, ki ne temelji na tradicionalnem večnitnem pristopu (angl. *multithreading*), ampak uporablja asinhroni vhodno-izhodni dogodkovni programski model [105]. Ogrodje je implementirano v jezikih C in C++, osredinjeno pa je na zmogljivost izvajanja in varčno porabo pomnilnika. Kljub temu da se jezik JavaScript običajno uporablja v brskalnikih, Node.js želi zagotoviti stabilne strežniške procese, ki se izvajajo v daljšem časovnem obdobju [105].

Za razvoj sprednjega ogrodja (angl. *frontend*) naše spletne aplikacijo smo uporabili JavaScript ogrodje AngularJS. AngularJS je odprtokodno Javascript MVC-ogrodje, ki ga je vzdržuje Google. Razvil ga je Miško Hevery pri podjetju Brat Tech LLC leta 2009 [74]. Uradna dokumentacija [11] ga definira kot *strukturno ogrodje za dinamične spletne aplikacije*. Namen ogrodja je razširitev označevalnega jezika HTML s posebnimi značkami in atributi, ki mu omogočajo realnočasovno sinhronizacijo z JavaScript kodo. To programerju omogoča, da več časa posveti razvoju aplikacijske logike, namesto da

bi se ukvarjal z posodabljanjem prikaza podatkov oziroma MVC komponente *View* [52]. Med najpomembnejše lastnosti ogrodja AngularJS lahko uvrstimo dvosmerno vezavo podatkov (angl. *two-way data binding*), predloge (angl. *templates*), vstavljanje odvisnosti (angl. *dependency injection*) ter ukaze (angl. *directives*) [52].

3.2.3 Podatkovna baza

Za podatkovno bazo lahko rečemo, da je *srce informacijskega sistema, v katerem se nahajajo podatki, ki jih potrebujemo za izvajanje poslovnih procesov in pretvorbo podatkov v informacije* [113]. Za spletno aplikacijo so podatki bistvenega pomena, zato je tudi pomembno, kako jih hranimo.

Relacijske podatkovne baze, ki jih je predlagal E. F. Codd [25] leta 1970, so tudi danes dominantni tip podatkovne baze kljub popularizaciji različnih nerelacijskih (NoSQL) podatkovnih baz. Nerelacijske podatkovne baze sicer obstajajo od leta 1960, a njihova uporaba je doživela ekspanzijo šele z nedavno nastalimi rešitvami, kot so MongoDB, CouchDB, Redis in podobne. Ena njihovih najpomembnejših značilnosti je horizontalna skalabilnost, ki je izjemno pomembna za večje projekte [65]. Obstajajo različni tipi nerelacijskih podatkovnih baz [47, 103]: stolpičaste, dokumentne, ključ-vrednost, graf in multi-model; vsaka od naštetih je najprimernejša za določeno strukturo podatkov. Relacijske podatkovne baze so primerne za strukture podatkov, ki vključujejo kompleksne odnose med entitetami, veliko število transakcij, kompleksne poizvedbe in podobno. Po drugi strani so NoSQL-podatkovne baze primernejše za obliko podatkov, ki imajo potrebo po horizontalni skalabilnosti, njihovi notranji odnosi pa so relativno preprosti.

Glede na strukturo podatkov, ki jih znotraj naše spletne aplikacije zbiramo, smo se odločili za uporabo dokumentno orientirane NoSQL-podatkovne baze *CouchDB*.

Po namestitvi podatkovne baze ni bilo treba definirati podatkovnega modela na ravni podatkovne baze, temveč smo ga definirali znotraj sprednjega ogrodja aplikacije, ki je podatke tudi pridobivalo. Vsak dokumentni vnos v podatkovno bazo vsebuje naslednje

podatke:

- časovna oznaka
- UUID-identifikator
- dovoljenje za kamero (`true/false`)
- opis morebitne napake za kamero
- dovoljenje za lokacijo (`true/false`)
- opis morebitne napake za lokacijo
- operacijski sistem
- primarni jezik
- podprti jeziki
- naziv in različica brskalnika
- piškotki so omogočeni (`true/false`)
- Java je omogočena (`true/false`)
- število logičnih jeder
- ločljivost zaslona
- mobilna naprava (`true/false`)
- število strani, obiskanih v trenutnem zavihku
- IP-naslov
- lokacija, pridobiljena z IP-naslova

Podatkovna baza hrani dokumente s prej omenjeno strukturo podatkov v obliki JSON.

3.2.4 Strežniški certifikat

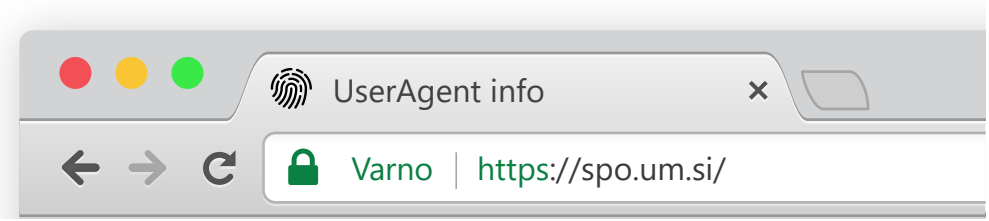
Protokol HTTPS (*Hyper Text Transfer Protocol Secure*) je varna različica protokola HTTP in je zadolžen za prenos podatkov med brskalnikom in strežnikom, v katerem gostuje spletna stran. Uporaba protokola HTTPS zagotavlja varno povezavo med odjemalcem in strežnikom. HTTPS uporablja protokol TLS (angl. *Transport Layer Security*), ki temelji na infrastrukturi javnega ključa (PKI, angl. *Public Key Infrastructure*). PKI-sistem je asimetričen, kar pomeni, da se vse, kar je šifrirano z javnim ključem, lahko dešifrira z zasebnim ključem in obratno.

Protokol TLS želi zagotoviti naslednje cilje [34]:

- **Kriptografska varnost**: TLS uporabljal naj bi se za vzpostavitev varne povezave med dvema stranema.

- **Interoperabilnost**: Neodvisni programerji morajo imeti možnost, da razvijejo aplikacije, ki uporabljajo TLS in lahko uspešno izmenjujejo kriptografske parametre, ne da bi poznali kodo svojega sogovornika.
- **Razširljivost**: TLS želi zagotoviti ogrodje, v katero se lahko vgradijo novi javni ključi in enkripcijske metode.
- **Relativna učinkovitost**: Kriptografske operacije, posebej operacije z javnimi ključi, so običajno procesorsko zahtevne. Zato mora protokol TLS zmanjšati število povezav, ki jih je treba vzpostaviti na novo. Prav tako je treba paziti na zmanjšanje omrežne aktivnosti.

Ko brskalnik pošlje zahtevo za HTTPS-stran, spletna stran vrne svoj TLS-certifikat, ki vsebuje javni ključ, s katerim lahko brskalnik dešifrira vsebino, dobljeno od strežnika.



Slika 3.2: Indikator varne povezave v brskalniku Chrome

Odločili smo se za uporabo certifikata, izdanega pri izdajatelju kvalificiranih digitalnih potrdil *Let's Encrypt*. Ta izdajatelj od aprila 2016 nudi brezplačne certifikate X.509. Proces izdelave in namestitve varnostnega certifikata *Let's Encrypt* lahko vidimo v izseku programske kode 3.2.

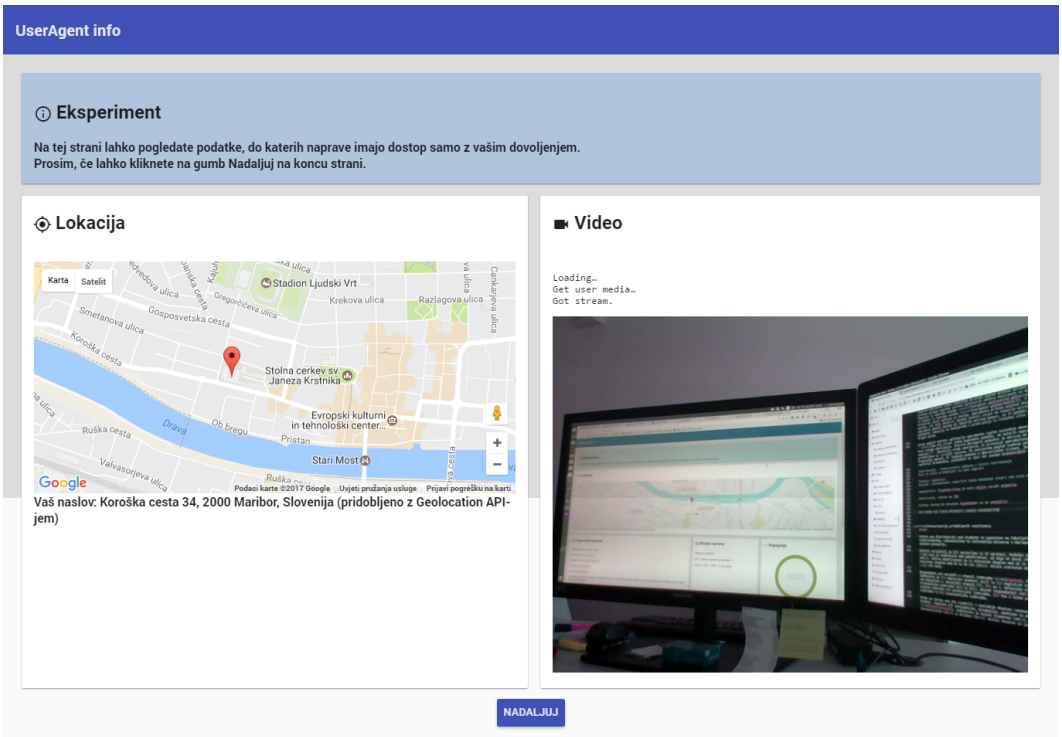
Izsek programske kode 3.2: Proces izdelave in namestitve varnostnega certifikata *Let's Encrypt*

```
1 $ sudo add-apt-repository ppa:certbot/certbot
2 $ sudo apt-get update
3 $ sudo apt-get install python-certbot-apache
4 $ sudo certbot --apache -d spo.um.si
```

Pogoj za izdajo in namestitev certifikata je domena, saj certifikata ni možno izdati za IP-naslov, ampak le za domeno, zato smo pridobili domeno `spo.um.si`.

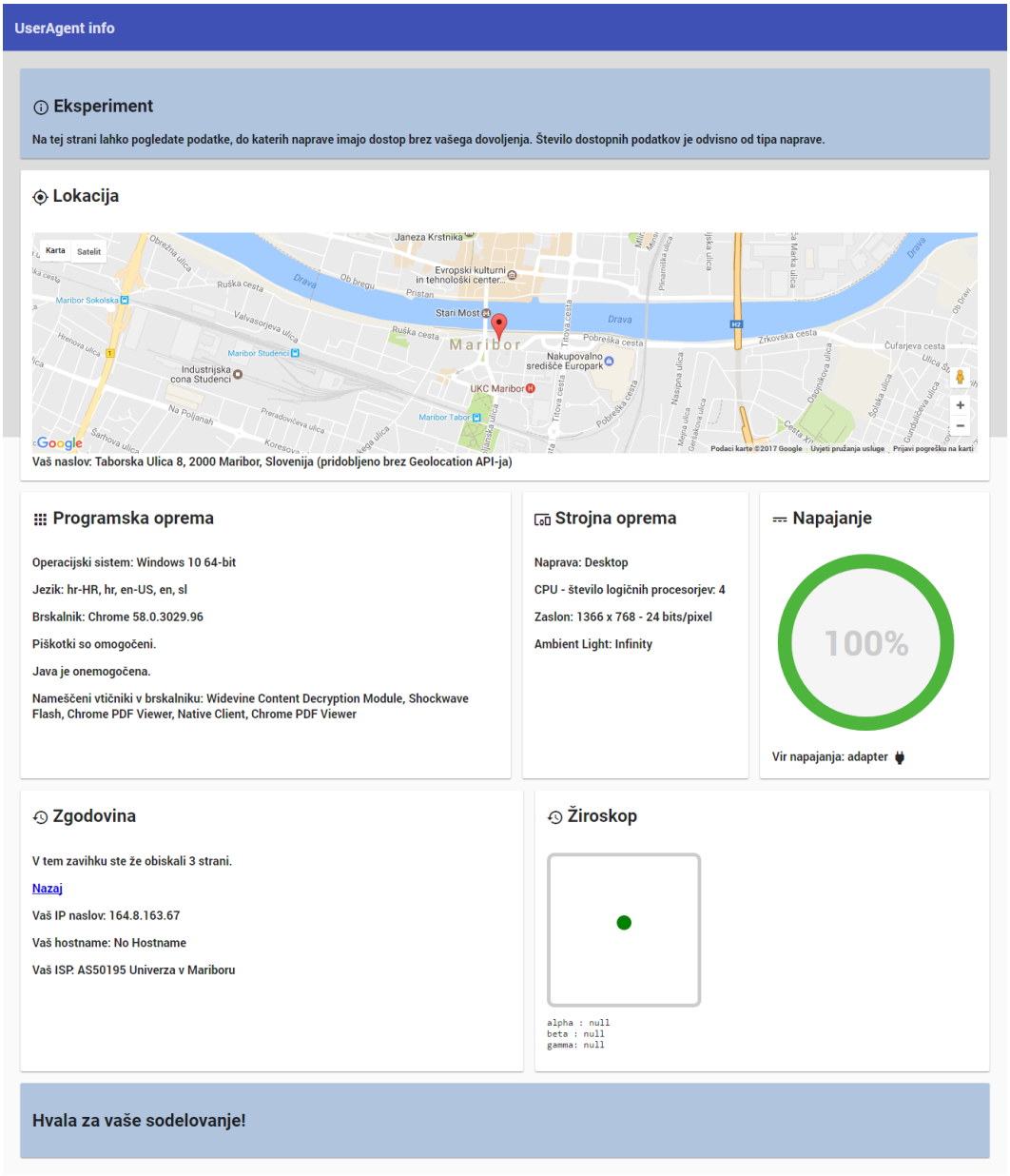
3.2.5 Zasnova spletne aplikacije

Razvito spletno aplikacijo smo zasnovali v obliki enostranske spletne aplikacije. Enostranska spletna aplikacija (angl. *Single page aplication*) je aplikacija, ki ima samo eno stran, lahko pa ima več pogledov (angl. *views*) [16]. Začetna zahteva sproži prenos celotne spletne aplikacije skupaj z vsemi povezanimi datotekami. Ko je aplikacija enkrat naložena v brskalniku, tudi pri prehodu na drugi pogled ne nalaga dodatnih komponent razen morebitnih asinhronih zahtev za podatki. Te asinhrone zahteve se praviloma povezujejo na REST spletne storitve.



Slika 3.3: Posnetek zaslona prvega pogleda spletne aplikacije

Aplikacija smo sestavili iz dveh pogledov. Prvi pogled (slika 3.3), ki se uporabniku prikaže ob odprtju spletne aplikacije, vsebuje naslednje obvestilo uporabnikom: „*Na tej strani lahko pogledate podatke, do katerih imajo naprave dostop samo z vašim dovoljenjem. Prosim, če lahko kliknete na gumb Nadaljuj na koncu strani.*“. Po obvestilu sledita dve polji, od katerih prvo vsebuje zemljevid, ki je namenjen prikazu uporabnikove lokacije, pridobljene iz geolokacijskega vmesnika, drugo pa prostor, ki je namenjen prikazu slike iz kamere in izpisa stanja kamere. Na koncu pa je pripravljen gumb z napi-



Slika 3.4: Posnetek zaslona drugega pogleda spletne aplikacije

som *Nadaljuj*, s katerim uporabnik lahko nadaljuje na drugi pogled spletne aplikacije. Takoj po prikazu prvega pogleda spletna aplikacija zahteva dovoljenje za dostop do lokacije (z uporabo metode `navigator.geolocation.getCurrentPosition()`) in kamere (z uporabo metode `navigator.getUserMedia()`). V trenutku ko brskalnik zazna zahtevo za dostop, uporabniku prikaže pojavno okno in ga vpraša, ali spletni strani želi dovoliti dostop do lokacije oziroma kamere. Če uporabnik dovoli dostop do lokacije oziroma kamere, se ti podatki ustrezno pokažejo: lokacija na zemljevidu v prvem polju

in trenutna slika iz kamere v drugem polju.

Drugi pogled (slika 3.4) spletne aplikacije uporabniku izpiše naslednje obvestilo: „*Na tej strani lahko pogledate podatke, do katerih imajo naprave dostop brez vašega dovoljenja. Število dostopnih podatkov je odvisno od tipa naprave.*“. Obvestilu sledi zemljevid, ki prikazuje uporabnikovo lokacijo, pridobljeno z IP-naslova. Po zemljevidu pogled vsebuje polje z nazivom Programska oprema, ki vsebuje podatke o operacijskem sistemu, jeziku, nazivu in različici brskalnika, piškotkih in Javi ter seznam brskalniških vtičnikov. Sledi polje z nazivom Strojna oprema, ki uporabniku izpiše, preko katere naprave dostopa do spletne aplikacije, število logičnih jeder procesorja in ločljivost zaslona. Polje z nazivom Napajanje prikazuje odstotek napolnjenosti baterije ter vir napajanja (baterija/adapter). V naslednjem polju spletna aplikacija uporabniku izpiše, koliko strani je obiskal v trenutnem zavihku, IP naslov naprave, ime gostitelja (angl. *hostname*) ter naziv ponudnika spletnih storitev. Zadnje polje prikazuje podatke iz žiroskopa in njihovo vizualizacijo. Na koncu smo uporabnikom izpisali obvestilo, s katerim smo se jim zahvalili za sodelovanje.

Nekatere metode zbiranja podatkov v brskalnikih danes niso več izvedljive, saj W3C v svojih zahtevah in priporočilih opozarja na potencialno nevarne funkcionalnosti. Primer je izkoriščanje funkcionalnosti psevdo razreda `a:visited` za ugotavljanje, katere strani je uporabnik že obiskal brez njegovega soglasja. Na to pomanjkljivost je W3C začel opozarjati [27] že leta 2010, saj se od takrat v specifikacijah standarda CSS nahaja opomba, s katero konzorcij od proizvajalcev brskalnikov zahteva, da pomanjkljivost odpravijo.

Vseh podatkov ni mogoče pridobiti na vseh napravah, saj obstajajo različne strojne in programske omejitve. Spletno aplikacijo smo testirali na napravah z operacijskimi sistemi Microsoft Windows 10, Ubuntu Linux 16.04, Android 7.1.2, iOS 10.3 in Windows Phone 8.1. Kljub temu da smo v omejitvah navedli, da se bomo omejili na spletni brskalnik Google Chrome, smo delovanje spletne aplikacije preizkusili tudi v drugih omenjenih brskalnikih. Težave so se pojavile pri prikazu spletne strani v brskalniku Microsoft Edge v operacijskem sistemu Microsoft Windows Phone 8.1, v katerem se spletna stran ni prikazala pravilno. Brskalnik Mozilla Firefox podpira večino funk-

cionalnosti spletne aplikacije razen dostopa do napajanja in usmeritve naprave. V brskalniku Safari na operacijskem sistemu iOS ni možno dostopati do kamere in stanja baterije. Brskalnik Opera podpira vse funkcionalnosti spletne aplikacije, saj uporablja enak prikazovalnik (angl. *rendering engine*) kot brskalnik Google Chrome. Podatke o kameri in lokaciji v prvem pogledu smo lahko pridobili samo na napravah, ki te funkcionalnosti podpirajo. To so bile predvsem mobilne naprave in prenosniki, manj pa stacionarni računalniki, saj jih večina nima kamere.

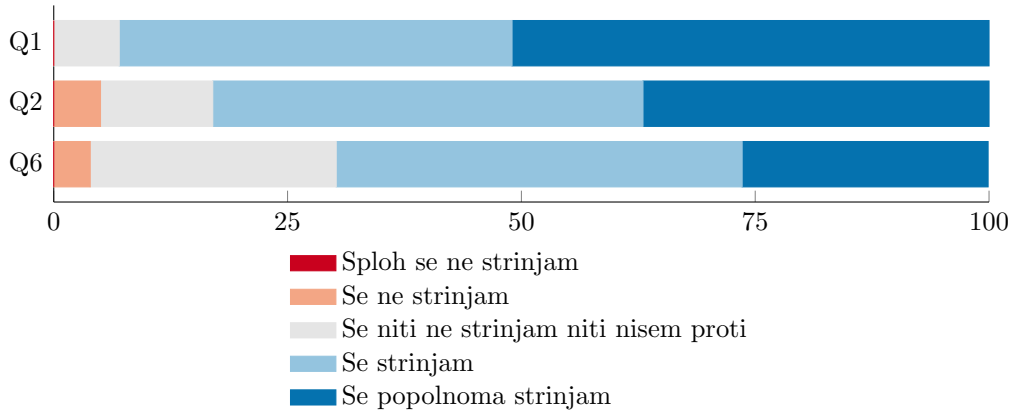
3.3 Analiza in interpretacija rezultatov anketnega vprašalnika in spletne aplikacije

Anketo smo razdelili med študente in zaposlene na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru ter med dijake splošne gimnazije.

Anketni vprašalnik je bil sestavljen iz 32 vprašanj. Podatke smo zbirali 14 dni. V tem času je sodelovalo 108 anketirancev, od tega jih je le 76 rešilo anketni vprašalnik in uporabilo spletno aplikacijo do konca. Sodelovalo je 27 žensk (36 %) in 49 moških (64 %). Večina anketirancev je iz starostne skupine 18–24 let (68 %) in starostne skupine 34–44 let (16 %). Ostale starostne skupine so zastopane z 9 % ali manj.

Anketirance smo vprašali tudi o stopnji izobrazbe [102]. Sodelovalo je 8 % doktorjev znanosti (KLASIUS 8/2) in 3 % magistrov znanosti (KLASIUS 8/1), 13 % ima dokončano 2. bolonjsko stopnjo (KLASIUS 7), 4 % pa 1. bolonjsko stopnjo in prejšnje visokošolsko izobrazbo (KLASIUS 6/2 in 6/1). Največ anketirancev (64 %) ima dokončano srednjo tehniško ali gimnazijsko izobrazbo (KLASIUS 5), 1 % ima 3-letno srednjo poklicno izobrazbo (KLASIUS 4) in 7 % osnovnošolsko izobrazbo (KLASIUS 2).

Glede na status smo jih razdelili v naslednje skupine: osnovnošolec, dijak, študent, zaposlen ali samozaposlen, brezposeln, upokojen in drugo [80]. Največ je sodelovalo študentov (55 %), zaposlenih ali samozaposlenih (26 %) in dijakov (16 %). Ostale



Graf 3.1: Prikaz razmerja med odgovori na vprašanji *Koliko se strinjate z naslednjo izjavo? – Imam veliko izkušenj z uporabo spleta* (Q1), *Mislim, da se zavedam nevarnosti na spletu* (Q2) in *Anonimnost na spletu mi je pomembna* (Q6)

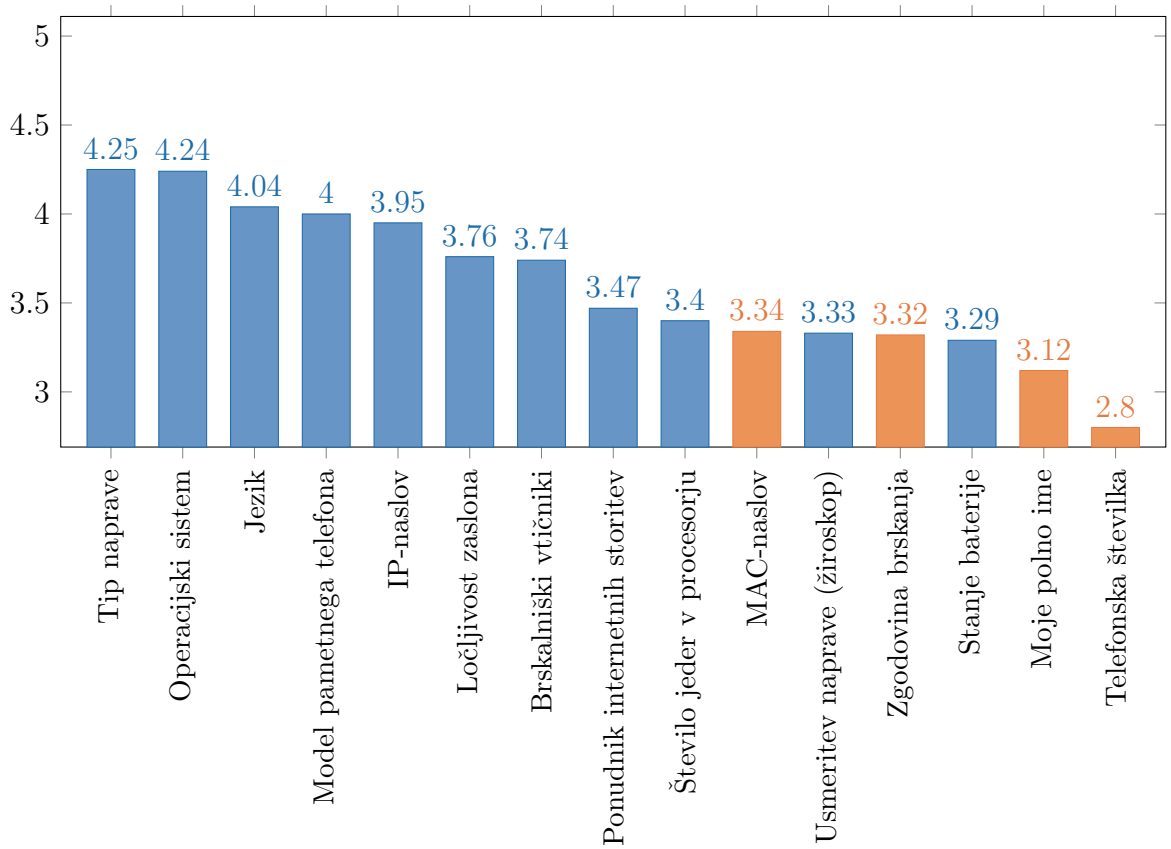
skupine so zastopane z 1 % ali manj.

Vse, ki so pri vprašanju o statusu označili, da so študenti, smo vprašali o obiskovani univerzi, v primeru Univerze v Mariboru o fakulteti in v primeru Fakultete za elektrotehniko, računalništvo in informatiko o obiskovanem študijskem programu. Vsi anketiranci, ki so v anketi označili, da so študenti, so študenti Univerze v Mariboru. Anketo je rešilo 40 študentov, kar je 53 %. Vsi anketiranci, ki so študenti Univerze v Mariboru, so študenti Fakultete za elektrotehniko, računalništvo in informatiko, od tega 25 študentov (63 %) na 1. stopnji študijskega programa Računalništvo in informacijske tehnologije UNI, 10 študentov (25 %) na 1. stopnji študijskega programa Informatika in tehnologije komuniciranja UNI, 4 študenti (10 %) na študijskem programu Medijske komunikacije UNI in 1 študent (3 %) je na študijskem programu Telekomunikacije UNI.

Anketiranci se štejejo za zelo izkušene uporabnike spleta, saj jih je 51 % odgovorilo, da se popolnoma strinjajo s tem, da se so izkušeni, 42 % pa da se z izjavo strinjajo (Slika 3.1, *Q1*). Prav tako mislijo, da se dobro zavedajo nevarnostih na spletu (Slika 3.1, *Q2*).

Kljub temu da večina anketirancev meni, da so ozaveščeni, se izkaže, da to ne vpliva na njihovo zavedanje o tem, koliko je možno pridobiti podatke o njihovem brskalniku.

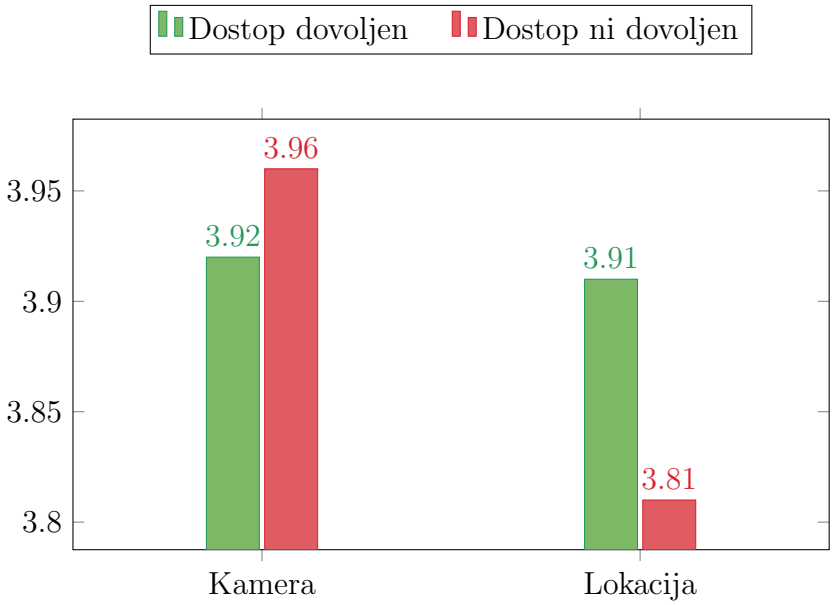
Uporabniki se najbolj zavedajo, da je možno, da spletna stran brez njihovega dovoljenja



Graf 3.2: Histogram povprečja vrednosti odgovorov na vprašanja o dostopnosti posameznih podatkov

dostopa do podatkov o tipu naprave, operacijskem sistemu, jeziku in modelu pametnega telefona, kar lahko vidimo na grafu 3.2. Kontrolne trditve, ki se nanašajo na podatke, ki jih spletna stran ne more pridobiti brez uporabnikovega dovoljenja, se uvrščajo med najslabše ocenjene. To pomeni, da se večji delež anketirancev kot pri ostalih trditvah zaveda, da teh podatkov ni mogoče pridobiti brez dovoljenja. So pa te trditve vsekakor ocenjene z višjimi ocenami, kot je bilo predvideno, saj teh podatkov v resnici ni mogoče pridobiti.

Z grafa 3.3 je razvidno, da si tisti, ki so dovolili dostop do kamere, želijo biti na spletu bolj anonimni v primerjavi s tistimi, ki tega niso dovolili. Ta podatek odstopa od pričakovanj, saj smo pričakovali, da uporabniki, ki želijo več anonimnosti, ne bodo dovolili dostopa do kamere. Se je pa pojavila večja razlika pri vprašanju o anonimnosti v primeru, ko anketirance ločimo glede na to, ali so dovolili dostop do lokacije.



Graf 3.3: Histogram vrednosti odgovorov na vprašanje *Koliko se strnjate z naslednjimi trditvami? – Pripravljen sem dovoliti dostop do svoje kamere* in *Pripravljen sem dovoliti dostop do svoje lokacije* v primerjavi z željo anketirancev po anonimnosti

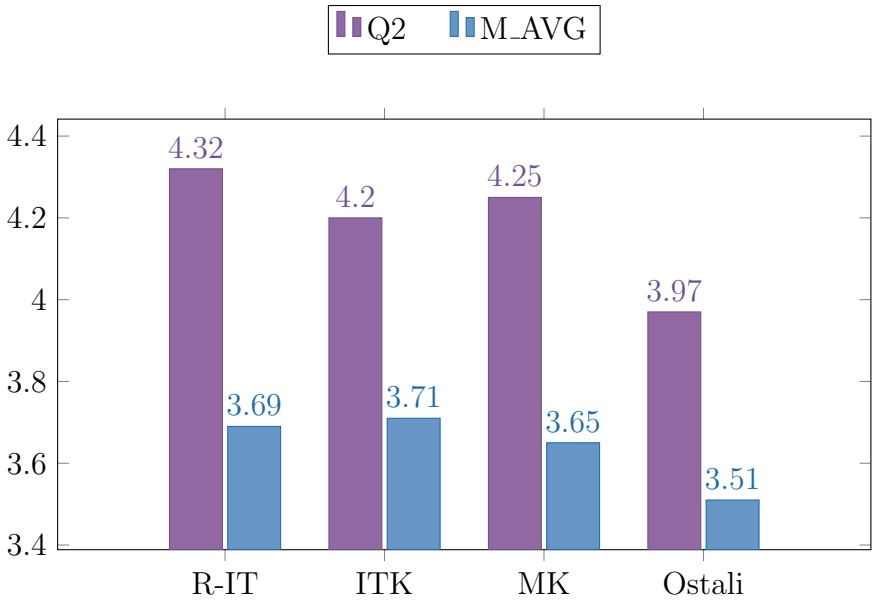
V tabeli 3.2 so prikazani podatki, ki kažejo, da je bil večji delež anketirancev pripravljen dovoliti dostop do lokacije kot pa do kamere. Dostop do lokacije je dovolilo enkrat več anketirancev.

Tabela 3.2: Primerjava deleža uporabnikov, ki so dovolili dostop do lokacije in kamere

	Lokacija	Kamera
Dovolil dostop	28%	14%
Ni dovolil dostopa	72%	86%

Podatki v grafu 3.4 kažejo, da študenti Fakultete za elektrotehniko, računalništvo in informatiko pri vprašanju *Koliko se strinjate z naslednjimi trditvami? – Mislim, da se zavedam nevarnosti na spletu.* (*Q2*) sebe ocenjujejo z višjimi ocenami kot ostali anketiranci. Prav tako so njihove povprečne ocene odgovorov, ki se nanašajo na zavedanje o dostopnosti podatkov v spletnih brskalnikih (*M_AVG*), višje kot pri ostalih anketirancih. Študenti Fakultete za elektrotehniko, računalništvo in informatiko so dosegli zelo podobne vrednosti spremenljivke *M_AVG*. Zanimivost, ki jo lahko izpostavimo, je,

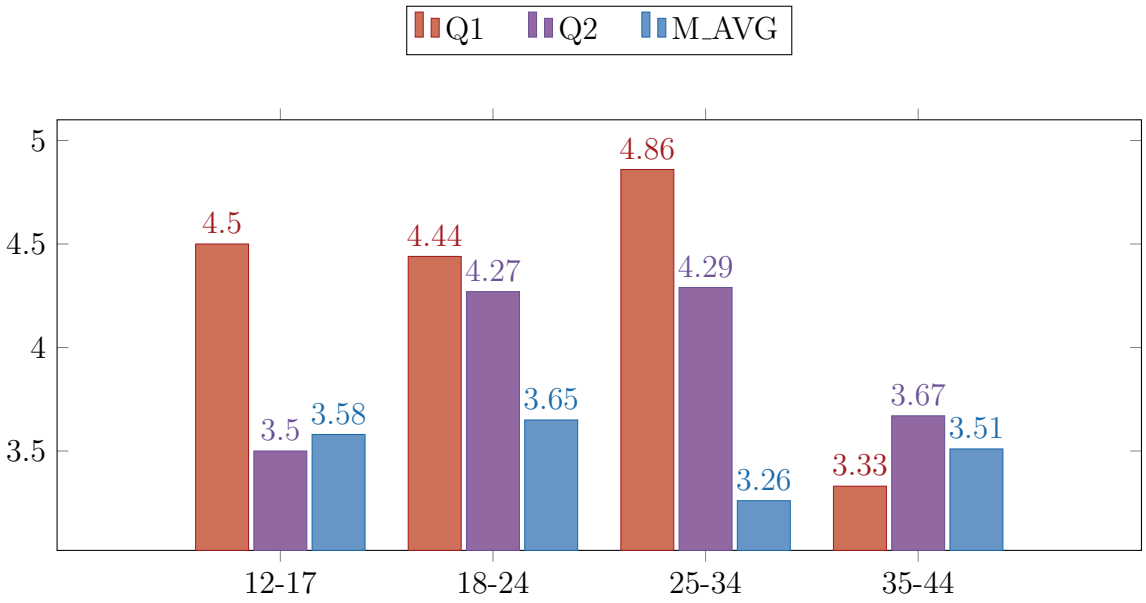
da so študenti študijskega programa Informatika in tehnologije komuniciranja (ITK) izmed vseh študijskih programov sebe ocenili z najslabšo oceno pri vprašanju *Q2*, a so hkrati dosegli najboljšo povprečno oceno odgovorov, ki se nanašajo na zavedanje o dostopnosti podatkov v spletnih brskalnikih (*M_AVG*).



Graf 3.4: Histogram vrednosti odgovorov na vprašanje *Koliko se strinjate z naslednjimi trditvami? – Mislim, da se zavedam nevarnosti na spletu* (Q2) v primerjavi s povprečjem odgovorov na vprašanja o dostopnosti podatkov v brskalnikih (M_AVG), razdeljeno po študijskih smereh

Na podlagi histograma 3.5 lahko sklepamo, da je starostna skupina 35–44 let dosegla povprečno najnižjo vrednost (*Q1*, *Q2* in *M_AVG*), vendar pa je najbolj realno ocenila svoje sposobnosti, saj je povprečje spremenljivk *Q1* in *Q2* najbližje oceni, ki se nanaša na zavedanje o dostopnosti podatkov v spletnih brskalnikih (*M_AVG*). Starostna skupina 25–34 let se šteje za izkušene uporabnike spleta (*Q1*) in meni, da se zaveda nevarnosti na spletu (*Q2*), vendar pa je njena povprečna ocena, ki se nanaša na zavedanje o dostopnosti podatkov v spletnih brskalnikih (*M_AVG*), najnižja med vsemi starostnimi skupinami. Pri starostni skupini 12–17 let opažamo, da se je pojavila velika razlika med povprečnimi ocenami odgovorov na vprašanja o izkušnjah z uporabo spleta (*Q1*) in o zavedanju o nevarnostih na spletu (*Q2*). Ti vprašani menijo, da so izkušeni, a da se hkrati nevarnosti na spletu ne zavedajo dovolj. Starostna skupina 18–24 let

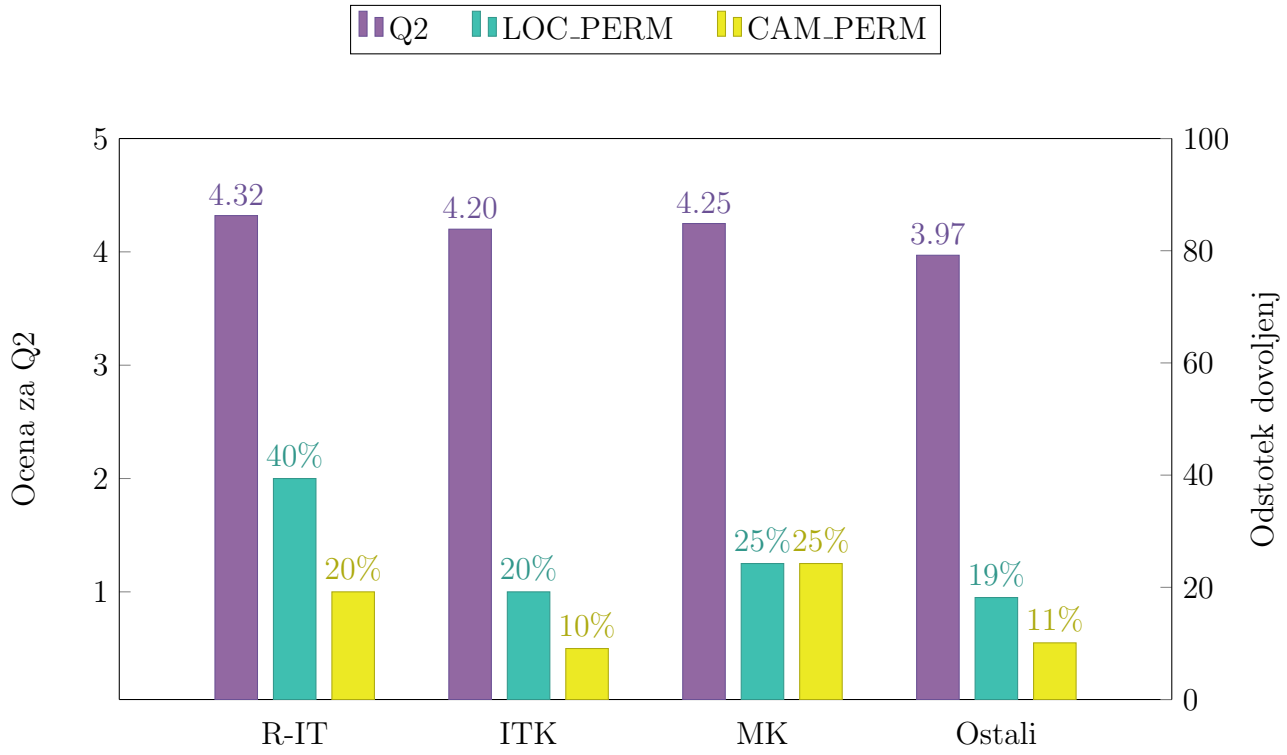
ima najboljšo povprečno oceno, ki se nanaša na zavedanje o dostopnosti podatkov v spletnih brskalnikih (*M_AVG*).



Graf 3.5: Histogram vrednosti odgovorov na vprašanje *Koliko se strinjate z naslednjimi trditvami? – Imam veliko izkušenj ob uporabi spleta* (Q1) in *Mislim, da se zavedam nevarnosti na spletu* (Q2) v primerjavi s povprečjem odgovorov na vprašanja o dostopnosti podatkov v brskalnikih (M_AVG), razdeljeno po starostnih skupinah

Graf 3.6 prikazuje neodvisnost spremenljivke *Q2* od odstotka dovoljenj dostopa do lokacije (*LOC_PERM*) in kamere (*CAM_PERM*), saj so vrednosti *Q2* konsistentne med programi, spremenljivki (*LOC_PERM*) in (*CAM_PERM*) pa se med posameznimi programi zelo razlikujeta. Vsi anketiranci so v večjem številu dovoljevali dostop do lokacije, razen študentov Medijskih komunikacij UNI, ki so dovolili dostop do lokacije in kamere v enaki količini. Prav tako so vsi anketiranci, razen študentov Medijskih komunikacij UNI, dovolili dostop do lokacije v enkrat večjem deležu kot do kamere. V povprečju so anketiranci, ki niso študentje Fakultete za elektrotehniko, računalništvo in informatiko, v manjši meri dovolili dostop do lokacije in kamere.

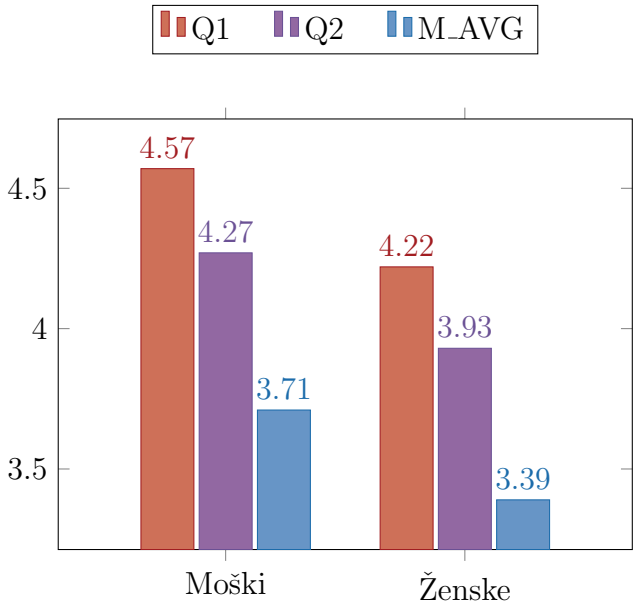
Iz grafa 3.7 je razvidno, da so moški anketiranci dosegli višjo povprečno oceno, ki se nanaša na zavedanje o dostopnosti podatkov v spletnih brskalnikih (*M_AVG*). Prav tako so sebe ocenili z višjimi ocenami pri vprašanjih o izkušnjah uporabe spleta (*Q1*) in zavedanju o nevarnosti na spletu (*Q2*).



Graf 3.6: Histogram vrednosti odgovorov na vprašanje *Koliko se strinjate z naslednjimi trditvami? – Mislim, da se zavedam nevarnosti na spletu* (Q2) v primerjavi z dovoljenjem dostopa do lokacije in kamere v spletni aplikaciji, razdeljeno po študijskih smereh

Iz grafa 3.8 sklepamo, da so moški in ženski anketiranci približno enako nezaupljivi glede dostopa do lokacije, moški pa so za faktor ¼ manj zaupljivi glede dostopa do kamere. Koeficient sorazmernosti med vsebinsko povezanimi spremenljivkami (*Q3-LOC_PERM* in *Q4-CAM_PERM*) konvergira k skupni vrednosti (0.1).

Glede na to, da so anketiranci dostopali do ankete in spletne aplikacije iz različnih naprav in da so te naprave imele različne strojne in programske karakteristike, niso vsi imeli možnosti, da bi lahko dovolili dostop do lokacije ali kamere. Iz prve vrstice tabele 3.3 (N) lahko razberemo število anketirancev, ki je na svoji napravi imelo podporo za geolokacijo in vgrajeno kamero. Iz pridobljenih podatkov lahko ugotovimo, da je 37 % anketirancev, ki so imeli to možnost, dovolilo dostop do lokacije, in 18 % do kamere. Uporabniki torej menijo, da manj tvegajo, če dovolijo dostop do lokacije kot do kamere. Vsekakor so anketiranci najbolj nezaupljivi pri dovoljevanju dostopa na

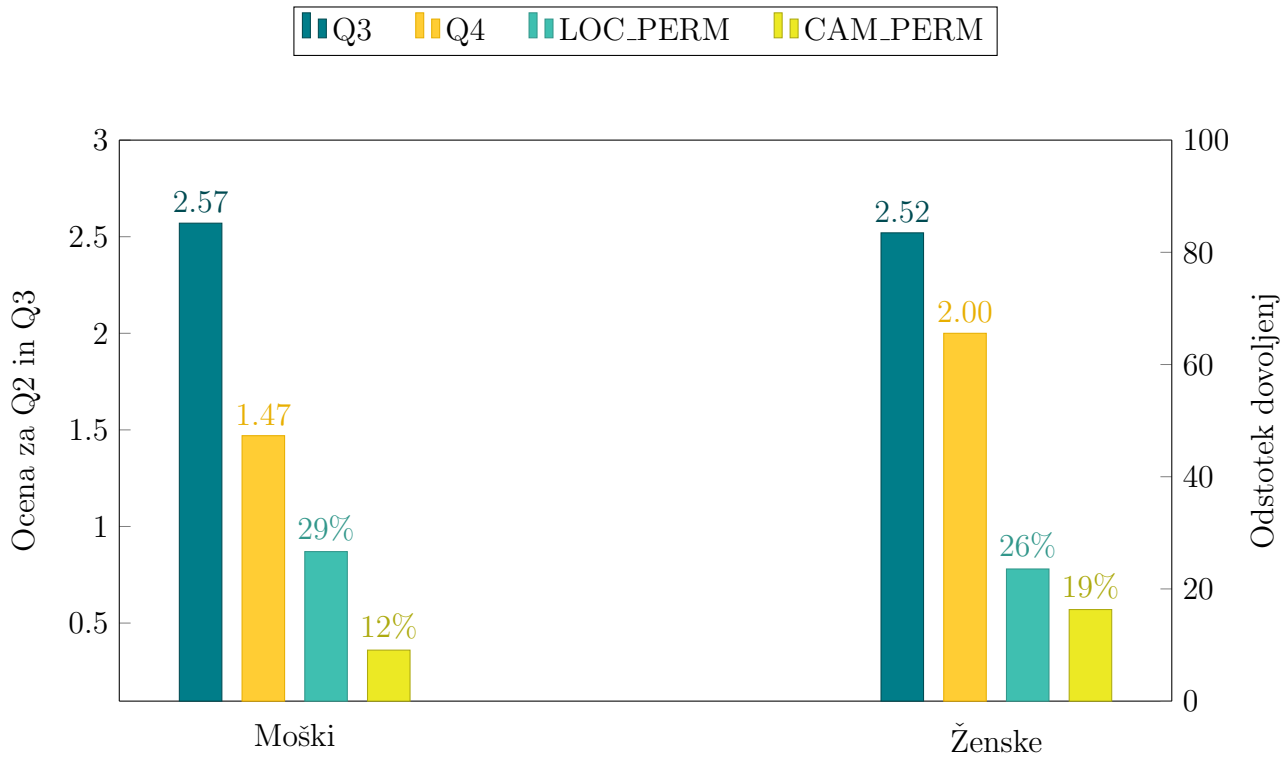


Graf 3.7: Histogram vrednosti odgovorov na vprašanje *Koliko se strinjate z naslednjimi trditvami? – Imam veliko izkušenj ob uporabi spleta* (Q1) in *Mislim, da se zavedam nevarnosti na spletu* (Q2) v primerjavi s povprečjem odgovorov na vprašanja o dostopnosti podatkov v brskalnikih (M_AVG), razdeljeno po spolu

mobilnih napravah, saj dostopa ni dovolil nihče.

Tabela 3.3: Primerjava uporabnikov, ki so dovolili dostop do kamere in lokacije

	Lokacija	Kamera
N	75	47
Računalnik	37%	18%
Mobilna naprava	0%	0%



Graf 3.8: Histogram vrednosti odgovorov na vprašanji *Koliko se strinjate z naslednjimi trditvami? – Pripravljen sem dovoliti dostop spletni strani do svoje lokacije* (Q3) in *Pripravljen sem dovoliti spletni strani dostop do svoje kamere* (Q4) v primerjavi z odstotkom anketirancev, ki so dovolili dostop do lokacije (*LOC_PERM*) oziroma kamere (CAM_PERM), razdeljeno po spolu

4 Sklep

V magistrski nalogi smo obravnavali področji „zajemanja prstnih odtisov“ in profiliranja brskalnikov, pregledali smo literaturo, povezano z omenjenim področjem, kar je bilo izhodišče za nadaljnje raziskovanje. Nadaljevali smo s pregledom standardov, smernic in priporočil konzorcija W3C, ki so nam omogočili podroben vpogled v trenutno stanje na področju podpore sodobnih spletnim standardom in tehnologijam.

Cilj našega magistrskega dela je bil izdelava spletne aplikacije, ki pridobiva podatke o uporabniku, in jih je mogoče uporabiti za profiliranje brskalnikov obiskovalcev spletne aplikacije. Aplikacijo smo uspešno izdelali in uporabili za zbiranje podatkov, ki jih je mogoče pridobiti z dovoljenjem uporabnikov ali brez njega. Prav tako smo izdelali anketni vprašalnik, s katerim smo uporabnike povprašali o njihovih stališčih glede zasebnosti na spletu in o mnenju glede pridobivanja podatkov s pomočjo brskalnikov.

4.1 Ugotovitve

Pri pregledu literature smo se seznanili z različnimi metodami zbiranja podatkov iz brskalnikov, jih podrobneje raziskali ter ugotovili, da obstajajo številne raziskave [85, 86, 39, 104, 53, 58, 79], ki obravnavajo različne načine „zajemanja prstnih odtisov“ brskalnika. Slednje omogoča enolično identifikacijo posameznega brskalnika.

Anketni vprašalnik v kombinaciji s spletno aplikacijo je pokazal, da se anketiranci štejejo za izkušene uporabnike spleta in da se dobro zavedajo nevarnosti na spletu. Pokazalo se je tudi, da njihovo samozavedanje glede uporabe spleta ne vpliva na dejansko

zavedanje o podatkih, ki jih spletna stran lahko pridobi o njihovem brskalniku. Večina uporabnikov je trdila, da ne bi dovolila dostopa do svoje lokacije in kamere, kar so pokazali tudi rezultati zbiranja podatkov s pomočjo spletne aplikacije. Dostop do lokacije je namreč dovolilo zgolj 28 % in do kamere le 14 % anketirancev. Vsi anketiranci, ki so sodelovali v anketi in spletni aplikaciji na mobilnih napravah, niso dovolili dostopa niti do lokacije niti do kamere.

4.2 Preverjanje hipotez

Na začetku magistrskega dela smo si zastavili 3 hipoteze, ki smo jih preverili in jih bomo predstavili v nadaljevanju.

H_1: *S pomočjo spletne aplikacije je mogoče pridobivati podatke o uporabnikovem spletnem brskalniku v tolikšni meri, da to omogoča enolično identificiranje spletnih brskalnikov.*

Potem ko smo preučili literaturo, smo ugotovili, da je podatke o uporabnikovem spletnem brskalniku mogoče pridobivati v tolikšni meri, da to omogoča enolično identificiranje spletnih brskalnikov. Pokazali smo, da je posamezne podatke, ki jih spletna stran lahko pridobi od brskalnika brez dovoljenja ali vedenja uporabnika, mogoče uporabiti za identificiranje brskalnikov z določeno mero enoličnosti (npr. niz *User Agent*). Če spletna stran uporabi kombinacijo metod, ki smo jih obravnavali v magistrskem delu, lahko enolično identificira posamezni brskalnik, s čimer **potrjujemo prvo hipotezo**.

H_2: *Uporabniki se ne zavedajo možnosti pridobivanja podatkov s pomočjo spletnih aplikacij.*

V anketnem vprašalniku smo anketirancem postavili vprašanja, ki se nanašajo na možnosti dostopa do podatkov brskalnika, od katerih jih lahko spletne strani pridobijo 13 brez uporabniškega dovoljenja ali vedenja. Anketiranci so za te podatke odgovorili, da menijo, da jih je možno pridobiti, saj so s pozitivnimi ocenami odgovorili tudi na

kontrolna vprašanja o podatkih, ki jih ni možno pridobiti. Zaradi tega sklepamo, da se uporabniki zavedajo možnosti pridobivanja podatkov s pomočjo spletnih aplikacij, s čimer **zavračamo drugo hipotezo**.

H_3: *Uporabniki so v večji meri odgovorni za količino podatkov, ki jih je mogoče pridobiti o njihovih spletnih brskalnikih.*

Iz pridobljenih znanj, ki smo jih pridobili v pregledu literature, smo ugotovili, da lahko večino podatkov spletne strani pridobijo brez dovoljenja ali vedenja uporabnikov. Kljub temu da je za varnostno kritične podatke (geolokacija, kamera, mikrofon) potrebno dovoljenje, so za enolično profiliranje brskalnikov dovolj že tisti podatki, ki se ne štejejo za varnostno kritične in jih je mogoče pridobiti brez vedenja ali dovoljenja uporabnika. Zato trdimo, da uporabniki v večji niso meri odgovorni za količino podatkov, ki jih je mogoče pridobiti o njihovih spletnih brskalnikih in **zavračamo tretjo hipotezo**.

4.3 Možnosti nadaljnjih raziskav

V nadaljnjih raziskavah predlagamo raziskavo možnosti profiliranja z vmesniki in metodami, ki jih določa standard HTML5, ter s sorodnimi tehnologijami. Prav tako je treba spremljati nove smernice in priporočila konzorcija W3C, saj kot mednarodna organizacija, ki upravlja s spletnimi standardi, močno vpliva na možnosti profiliranja.

Smiselno bi bilo tudi sistematično testirati metode, ki jih je mogoče uporabljati za „zajemanje prstnih odtisov“ na vseh kombinacijah operacijskih sistemov in brskalnikov. Izkazalo se je namreč, da določene metode delujejo drugače na različnih operacijskih sistemih in brskalnikih [33], kar potencialno lahko povzroči nezaželene posledice.

Anketo, ki smo jo sestavili in izvedli, bi lahko v prihodnosti razširili in prilagodili novim metodam zbiranja podatkov. Treba je pregledati smernice in priporočila konzorcija W3C o novih varnostnih ukrepov in jih primerjati z implementacijami v posameznih brskalnikih. Prav tako bi lahko razširili nabor anketirancev (npr. študenti Fakultete za elektrotehniko, računalništvo in informatiko ali študenti Univerze v Mariboru).

4.4 Zaključek

Uporaba različnih spletnih aplikacij, ki pogosto pridobivajo osebne podatke uporabnikov, se veča [63]. Vendar se lahko vsak posameznik odloči, ali bo podatke aplikaciji posredoval ali ne. Po drugi strani pa imajo spletne aplikacije možnost dostopa do določenih podatkov o brskalnikih brez dovoljenja in vedenja uporabnikov. Zaradi današnje povezanosti med uporabniki in njihovimi napravami (računalnik, pametni telefon, ...) ter nameščenimi brskalniki postaja vprašanje dostopa do množice podatkov s pomočjo brskalnikov vedno aktualnejše.

Vse naprave, s pomočjo katerih uporabniki dostopajo do spleta, je mogoče locirati, jih identificirati in jim slediti [70]. Zato jih je mogoče uporabiti za nadzorovanje in sledenje, in sicer na način, ki škoduje interesom uporabnika. Še do nedavnega je bilo težje in dražje nadzorovati posameznika, saj je bil potreben neposreden stik. V tem pogledu tehnologija izboljšuje zmožnosti nadzorovanja z vpeljavo orodij, ki omogočajo zbiranje informacij, nastalih iz aktivnosti posameznika, in to na daljavo. To pa je zelo zmanjšalo stroške in olajšalo nadzorovanje. Prav tako pa sodobna tehnologija omogoča natančnejše podatke (npr. s kombiniranjem), kar ima za posledico popolnejšo sliko posameznika [91].

Ne trdimo, da vsi podatki, ki jih je mogoče zbrati, varnostno kritični. Prav tako ni res, da morajo uporabniki preprečiti posredovanje svojih osebnih podatkov v vseh primerih. Pomembno je, da se uporabniki zavedajo, *kaj* se zbira, *kako* in *ali je to res nujno*. V najboljšem primeru bi morali uporabniki doseči takšen *modus vivendi*, da izbranim stranem dovoljujejo ali omejujejo poseg v svojo zasebnost. V primeru spletnih strani, na katerih je mogoče izvesti interakcijo brez poseganja v področje zasebnosti, je vsekakor bolje izbrati slednje [62].

Literatura

[1] Josh Aas. Let's Encrypt: Delivering SSL/TLS Everywhere. *Let's Encrypt*, 18, 2014.

[2] Bernard Aboba, Daniel Burnett, Adam Bergkvist, Anant Narayanan, and Cullen Jennings. Media capture and streams. Candidate recommendation, W3C, May 2016. http://www.w3.org/TR/2016/CR-mediacapture-streams-20160519/.

[3] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689. ACM, 2014.

[4] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. Fpdetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1129–1140. ACM, 2013.

[5] Alessandro Acquisti and Jens Grossklags. What can behavioral economics teach us about privacy. *Digital Privacy: Theory, Technologies and Practices*, 18:363–377, 2007.

[6] Andrea Addis, Giuliano Armano, and Eloisa Vargiu. Profiling users to perform contextual advertising. In *Proceedings of the 10th Workshop dagli Oggetti agli Agenti (WOA 2009)*, volume 73, 2009.

[7] Maarten Aertsen. How to bring https to the masses? measuring issuance in the first year of let's encrypt. 2016.

[8] Gaurav Aggarwal, Elie Bursztein, Collin Jackson, and Dan Boneh. An analysis of private browsing modes in modern browsers. In *USENIX Security Symposium*, pages 79–94, 2010.

[9] Alexa. Alexa 500. `http://www.alexa.com/topsites`.

[10] Irwin Altman. The environment and social behavior: Privacy, personal space, territory, and crowding. 1975.

[11] AngularJS. What Is AngularJS? `https://docs.angularjs.org/guide/introduction`, 2017.

[12] UN General Assembly. Universal declaration of human rights. *UN General Assembly*, 1948.

[13] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM, 2009.

[14] A. Barth. HTTP State Management Mechanism. RFC 6265 (Proposed Standard), April 2011.

[15] A. Barth. The Web Origin Concept. RFC 6454 (Proposed Standard), December 2011.

[16] David Bavcon. *Spletni robot in iskalnik po evropskih projektih.* PhD thesis, Univerza v Ljubljani, 2014.

[17] Paul Bernal. *Internet privacy rights: rights to protect autonomy.* Number 24. Cambridge University Press, 2014.

[18] Tim Berners-Lee and Robert Cailliau. Worldwideweb: Proposal for a hypertext project. *Retrieved on February*, 26:2008, 1990.

[19] Biz Carson. You're more likely to order a pricey Uber ride if your phone is about to die. `http://uk.businessinsider.com/`

people-with-low-phone-batteries-more-likely-to-accept-uber-surge-pricing-2016
2016.

[20] Stephen Block, Andrei Popescu, Tim Volodine, and Rich Tibbett. Deviceorientation event specification. Candidate recommendation, W3C, August 2016. https://www.w3.org/TR/2016/CR-orientation-event-20160818/.

[21] Károly Boda, Ádám Máté Földes, Gábor György Gulyás, and Sándor Imre. User tracking on the web via cross-browser fingerprinting. In *Nordic Conference on Secure IT Systems*, pages 31–46. Springer, 2011.

[22] Can I use. Can I use Geolocation. `http://caniuse.com/#feat=geolocation`. Dostopano: 13. maja 2017.

[23] Deepayan Chakrabarti, Deepak Agarwal, and Vanja Josifovski. Contextual advertising by combining relevance with click feedback. In *Proceedings of the 17th international conference on World Wide Web*, pages 417–426. ACM, 2008.

[24] Yi-Chao Chen, Yong Liao, Mario Baldi, Sung-Ju Lee, and Lili Qiu. Os fingerprinting and tethering detection in mobile networks. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 173–180. ACM, 2014.

[25] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[26] Louis Cohen, Lawrence Manion, and Keith Morrison. *Research methods in education*. Routledge, 2013.

[27] World Wide Web Consortium et al. Cascading style sheets level 2 revision 1 (css 2.1) specification. 2011.

[28] A. Cooper, H. Tschofenig, B. Aboba, J. Peterson, J. Morris, M. Hansen, and R. Smith. Privacy Considerations for Internet Protocols. RFC 6973 (Informational), July 2013.

[29] A Cooper, H Tschofenig, B Aboba, J Peterson, J Morris, M Hansen, and R Smith. Rfc 6973: Privacy considerations for internet protocols. *IETF. Retrieved from tools. ietf. org/html/rfc6973*, 2013.

[30] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[31] Steve Cronen-Townsend and W Bruce Croft. Quantifying query ambiguity. In *Proceedings of the second international conference on Human Language Technology Research*, pages 104–109. Morgan Kaufmann Publishers Inc., 2002.

[32] John Daggett. CSS fonts module level 3. Candidate recommendation, W3C, October 2013. http://www.w3.org/TR/2013/CR-css-fonts-3-20131003/.

[33] Claudia Diaz, Lukasz Olejnik, Gunes Acar, and Claude Casteluccia. The leaking battery: A privacy analysis of the html5 battery status api. In *Lecture Notes in Computer Science*, volume 9481, pages 254–263. Springer, 2015.

[34] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905, 7919.

[35] dooli.si. Statistika slovenskega interneta. `https://www.dooli.si/statistika-slovenskega-interneta`, 2017.

[36] Nick Doty. Privacy interest group charter. Technical report, W3C, 2011. https://www.w3.org/2011/07/privacy-ig-charter.

[37] Nick Doty. Fingerprinting guidance for web specification authors (draft). W3C note, W3C, November 2015. http://www.w3.org/TR/2015/NOTE-fingerprinting-guidance-20151124/.

[38] Nick Doty. Reviewing for privacy in internet and web standard-setting. In *Security and Privacy Workshops (SPW), 2015 IEEE*, pages 185–192. IEEE, 2015.

[39] Peter Eckersley. How unique is your web browser? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–18. Springer, 2010.

[40] David Fifield and Serge Egelman. Fingerprinting web users through font metrics. In *International Conference on Financial Cryptography and Data Security*, pages 107–124. Springer, 2015.

[41] Matthew Fredrikson and Benjamin Livshits. Repriv: Re-envisioning in-browser privacy. In *IEEE Symposium on Security and Privacy*, 2011.

[42] Pablo Garaizar and Mariluz Guenaga. A multimodal learning analytics view of HTML5 APIs: technical benefits and privacy risks. In *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 275–281. ACM, 2014.

[43] Gartner. Gartner says 8.4 billion connected "things"will be in use in 2017, up 31 percent from 2016. `http://www.gartner.com/newsroom/id/3598917`.

[44] Joshua Gomez, Travis Pinnick, and Ashkan Soltani. Knowprivacy. *School of Information*, 2009.

[45] Gregor Polančič. Empirične raziskovalne metode - predavanja, 2017.

[46] Robert M Groves, Floyd J Fowler Jr, Mick P Couper, James M Lepkowski, Eleanor Singer, and Roger Tourangeau. *Survey methodology*, volume 561. John Wiley & Sons, 2011.

[47] Jing Han, E Haihong, Guan Le, and Jian Du. Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pages 363–366. IEEE, 2011.

[48] Mario Heiderich, Marcus Niemietz, Felix Schuster, Thorsten Holz, and Jörg Schwenk. Scriptless attacks: stealing the pie without touching the sill. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 760–771. ACM, 2012.

[49] Anthony T Holdener. *HTML5 geolocation*. "O'Reilly Media, Inc.", 2011.

[50] R Hughes. Upower reference manual, 2010.

[51] Collin Jackson, Andrew Bortz, Dan Boneh, and John C Mitchell. Protecting browser state from web privacy attacks. In *Proceedings of the 15th international conference on World Wide Web*, pages 737–744. ACM, 2006.

[52] Nilesh Jain, Priyanka Mangal, and Deepak Mehta. Angularjs: A modern mvc framework in javascript. *Journal of Global Research in Computer Science*, 5(12):17–23, 2015.

[53] Artur Janc and Lukasz Olejnik. Web browser history detection as a real-world privacy threat. In *European Symposium on Research in Computer Security*, pages 215–231. Springer, 2010.

[54] Dongseok Jang, Ranjit Jhala, Sorin Lerner, and Hovav Shacham. An empirical study of privacy-violating information flows in javascript web applications. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 270–283. ACM, 2010.

[55] Bernard J Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227, 2000.

[56] Jordan Golson. Uber knows you'll probably pay surge pricing if your battery is about to die. `http://www.theverge.com/2016/5/20/11721890/uber-surge-pricing-low-battery`, 2016.

[57] Keith Chen. This Is Your Brain On Uber. `http://www.npr.org/2016/05/17/478266839/this-is-your-brain-on-uber`, 2016.

[58] Amin Faiz Khademi, Mohammad Zulkernine, and Komminist Weldemariam. An empirical evaluation of web-based fingerprinting. *IEEE Software*, 32(4):46–52, 2015.

[59] Hyungsub Kim, Sangho Lee, and Jong Kim. Exploring and mitigating privacy threats of html5 geolocation api. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 306–315. ACM, 2014.

[60] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, 2005.

[61] Anssi Kostiainen and Tobie Langel. Ambient light sensor. W3C working draft, W3C, August 2016. https://www.w3.org/TR/2016/WD-ambient-light-20160830/.

[62] Balachander Krishnamurthy and Craig Wills. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of the 18th international conference on World wide web*, pages 541–550. ACM, 2009.

[63] Balachander Krishnamurthy and Craig E Wills. Generating a privacy footprint on the internet. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 65–70. ACM, 2006.

[64] Mounir Lamouri and Anssi Kostiainen. Battery status API. Candidate recommendation, W3C, July 2016. http://www.w3.org/TR/2016/CR-battery-status-20160707/.

[65] Neal Leavitt. Will nosql databases live up to their promise? *Computer*, 43(2), 2010.

[66] Nicolaas Matthijs and Filip Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 25–34. ACM, 2011.

[67] Jonathan R Mayer. Any person... a pamphleteer": Internet anonymity in the age of web 2.0. *Undergraduate Senior Thesis, Princeton University*, 2009.

[68] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 413–427. IEEE, 2012.

[69] Maryam Mehrnezhad, Ehsan Toreini, Siamak F Shahandashti, and Feng Hao. Stealing pins via mobile sensors: Actual risk versus user perception. *arXiv preprint arXiv:1605.05549*, 2016.

[70] Katina Michael and Roger Clarke. Location and tracking of mobile devices: Überveillance stalks the streets. *Computer Law & Security Review*, 29(3):216–228, 2013.

[71] Microsoft. Http cookies. `https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/http-cookies`.

[72] Marko Milanovic. Human rights treaties and foreign surveillance: Privacy in the digital age. *Harv. Int'l LJ*, 56:81, 2015.

[73] Lynette I Millett, Batya Friedman, and Edward Felten. Cookies and web browser design: Toward realizing informed consent online. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 46–52. ACM, 2001.

[74] Miško Hevery. Hello World, ¡angular/¿ is here. `http://misko.hevery.com/2009/09/28/hello-world-angular-is-here/`, 2009.

[75] Keaton Mowery, Dillon Bogenreif, Scott Yilek, and Hovav Shacham. Fingerprinting information in javascript implementations. *Proceedings of W2SP*, 2:180–193, 2011.

[76] Keaton Mowery and Hovav Shacham. Pixel perfect: Fingerprinting canvas in html5. *Proceedings of W2SP*, pages 1–12, 2012.

[77] Mozilla Developer Network. Detecting device orientation. `https://developer.mozilla.org/en-US/docs/Web/API/Detecting_device_orientation`, 2016.

[78] Mozilla Developer Network. Geolocation. `https://developer.mozilla.org/en-US/docs/Web/API/Geolocation`, 2016.

[79] Gabi Nakibly, Gilad Shelef, and Shiran Yudilevich. Hardware fingerprinting using HTML5. *arXiv preprint arXiv:1503.01408*, 2015.

[80] Lili Nemec Zlatolas. *Model vpliva zasebnosti na razkrivanje informacij uporabnikov družbenega omrežja Facebook*. PhD thesis, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2015.

[81] Jakob Nielsen. *Multimedia and hypertext: The Internet and beyond*. Morgan Kaufmann, 1995.

[82] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Security and privacy (SP), 2013 IEEE symposium on*, pages 541–555. IEEE, 2013.

[83] Mark Nottingham. Unsanctioned web tracking. Technical report, W3C, 2015. https://www.w3.org/2001/tag/doc/unsanctioned-tracking/.

[84] Ilkka Oksanen, Dominique Hazaël-Massieux, and Anssi Kostiainen. HTML media capture. Candidate recommendation, W3C, May 2017. https://www.w3.org/TR/2017/CR-html-media-capture-20170504/.

[85] Lukasz Olejnik, Claude Castelluccia, and Artur Janc. On the uniqueness of web browsing history patterns. *annals of telecommunications-annales des télécommunications*, 69(1-2):63–74, 2014.

[86] Lukasz Olejnik, Steven Englehardt, and Arvind Narayanan. Battery status not included: Assessing privacy in web standards. 2017.

[87] Daniel Olmedilla, Enrique Frías-Martínez, and Rubén Lara. Mobile web profiling: a study of off-portal surfing habits of mobile users. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 339–350. Springer, 2010.

[88] Evropski parlament in Svet Evropske unije. Direktiva evropskega parlamenta in sveta 95/46/es z dne 24. oktobra 1995 o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov. `http://eur-lex.europa.eu/legal-content/SL/TXT/HTML/?uri=CELEX:31995L0046`.

[89] Tamás Paulik, Adam Mate Foldes, and Gabor Gyorgy Gulyas. Blogcrypt: Private content publishing on the web. In *Emerging Security Information Systems and Technologies (SECURWARE), 2010 Fourth International Conference on*, pages 123–128. IEEE, 2010.

[90] Andrei Popescu. Geolocation API specification. W3C recommendation, W3C, October 2013. http://www.w3.org/TR/2013/REC-geolocation-API-20131024/.

[91] Silvia Puglisi, David Rebollo-Monedero, and Jordi Forné. You never surf alone. ubiquitous tracking of users' browsing habits. In *International Workshop on Data Privacy Management*, pages 273–280. Springer, 2015.

[92] Quora. Why do non-Mozilla browsers include "Mozilla"in their user agent strings? `https://www.quora.com/Why-do-non-Mozilla-browsers-include-Mozilla-in-their-user-agent-strings`, 2013.

[93] J Ruderman. Javascript security: Same origin, 2001.

[94] Takamichi Saito, Koki Yasuda, Takayuki Ishikawa, Rio Hosoi, Kazushi Takahashi, Yongyan Chen, and Marcin Zalasiński. Estimating cpu features by browser fingerprinting. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2016 10th International Conference on*, pages 587–592. IEEE, 2016.

[95] Iskander Sánchez-Rola, Xabier Ugarte-Pedrero, Igor Santos, and Pablo G Bringas. Tracking users like there is no tomorrow: Privacy on the current internet. In *International Joint Conference*, pages 473–483. Springer, 2015.

[96] Mark Sanderson. Ambiguous queries: test collections need more sense. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 499–506. ACM, 2008.

[97] Wendy Seltzer. Privacy interest group charter. Technical report, W3C, 2011. https://www.w3.org/2011/tracking-protection/charter.html.

[98] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *International Workshop on Privacy Enhancing Technologies*, pages 41–53. Springer, 2002.

[99] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Privacy protection in personalized search. In *ACM SIGIR Forum*, volume 41, pages 4–17. ACM, 2007.

[100] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. In *ACm SIGIR Forum*, volume 33, pages 6–12. ACM, 1999.

[101] Kapil Singh, Alexander Moshchuk, Helen J Wang, and Wenke Lee. On the incoherencies in web browser access control policies. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 463–478. IEEE, 2010.

[102] Statistični urad Republike Slovenije. KLASIUS-SRV. `http://www.stat.si/Klasius/Docs/opisiKLASIUS-SRV.pdf`, 2006.

[103] Christof Strauch, Ultra-Large Scale Sites, and Walter Kriha. Nosql databases. *Lecture Notes, Stuttgart Media University*, 2011.

[104] Naoki Takei, Takamichi Saito, Ko Takasu, and Tomotaka Yamada. Web browser fingerprinting using only cascading style sheets. In *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2015 10th International Conference on*, pages 57–63. IEEE, 2015.

[105] Stefan Tilkov and Steve Vinoski. Node. js: Using javascript to build high-performance network programs. *IEEE Internet Computing*, 14(6):80–83, 2010.

[106] Tim Berners-Lee. Frequently asked questions. `https://www.w3.org/People/Berners-Lee/FAQ.html`, 2001.

[107] Web Technology Surveys. Usage of web servers for websites. `https://w3techs.com/technologies/overview/web_server/all`, 2017.

[108] WebAIM. History of the browser user-agent string. `http://webaim.org/blog/user-agent-string-history/`, 2010.

[109] Zachary Weinberg, Eric Y Chen, Pavithra Ramesh Jayaraman, and Collin Jackson. I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 147–161. IEEE, 2011.

[110] Alan F Westin. Privacy and freedom. *Washington and Lee Law Review*, 25(1):166, 1968.

[111] Craig E Wills and Mihajlo Zeljkovic. A personalized approach to web privacy: awareness, attitudes and actions. *Information Management & Computer Security*, 19(1):53–73, 2011.

[112] Chuan Yue. Sensor-based mobile web fingerprinting and cross-site input inference attacks. In *Security and Privacy Workshops (SPW), 2016 IEEE*, pages 241–244. IEEE, 2016.

[113] Andreja Šet. *Načrtovanje in postavitev podatkovnih baz*. i2, Ljubljana, first edition, 2017. `http://www.i2-lj.si/ucbeniki/NIPPB_ucbenik_vzorec_s_kazalom.pdf`.

A Anketni vprašalnik

Pozdravljeni!
Sem Luka Hrgarek, študent Fakultete za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Pripravljam magistrsko nalogo z naslovom Zbiranje podatkov in profiliranje uporabniških naprav s pomočjo spletnih brskalnikov.
Zbrani podatki bodo obdelani strogo zaupno in obravnavani na ravni odgovorov vseh anketiranih. Uporabljeni bodo izključno v raziskovalne namene.
Za vaše sodelovanje se vam prijazno zahvaljujem.
Luka Hrgarek
luka.hrgarek@um.si

A.1 Demografski podatki

A.1.1 Spol

Prosimo, izberite samo eno izmed možnosti:

- ○ moški
- ○ ženski

A.1.2 Starost

Prosimo, izberite samo eno izmed možnosti:

○ manj kot 12 let

○ 12–17 let

○ 18–24 let

○ 25–34 let

○ 35–44 let

○ 45–54 let

○ 55–64 let

○ 65–75 let

○ 76 let ali več

A.1.3 Dokončana stopnja izobrazbe

Prosimo, izberite samo eno izmed možnosti:

○ 1. Nedokončana osnovnošolska izobrazba

○ 2. Osnovnošolska izobrazba

○ 3. Nižje poklicno izobraževanje (2 letno)

○ 4. Srednje poklicno izobraževanje (3 letno)

○ 5. Srednje tehniško in drugo strokovno ter splošno (gimnazijsko) izobraževanje

○ 6/1. Višješolsko in višje strokovno izobraževanje/višješolska (do 1994)

○ 6/2. Visokošolsko izobraževanje prve stopnje (1. bolonjska stopnja), visokošolsko strokovno izobraževanje (prejšnje)

○ 7. Visokošolsko izobraževanje druge stopnje in podobno izobraževanje (2. bolonjska stopnja)

○ 8/1. Magistrsko izobraževanje (prejšnje) in podobno izobraževanje/magisterij znanosti in podobna izobrazba

○ 8/2. Doktorsko in podobno izobraževanje/doktorat znanosti in podobna izobrazba (3. bolonjska stopnja)

○ Drugo _____

A.1.4 Vaš trenutni status

Prosimo, izberite samo eno izmed možnosti:

- ○ osnovnošolec
- ○ dijak
- ○ študent
- ○ zaposlen ali samozaposlen
- ○ brezposeln
- ○ upokojen
- ○ Drugo _____

A.1.5 Katero srednjo šolo obiskujete?

Na to vprašanje odgovorite samo, če je zadoščeno naslednjim pogojem: Odgovor je bil *dijak* pri vprašanju 1.4 (Vaš trenutni status)
Vpišite vaš odgovor: _____

A.1.6 V katero področje spada vaš študijski program?

Na to vprašanje odgovorite samo, če je zadoščeno naslednjim pogojem: Odgovor je bil *študent* pri vprašanju 1.4 (Vaš trenutni status)
Prosimo, izberite vse odgovore, ki ustrezajo:

- ☐ Naravoslovne vede
- ☐ Tehniške in tehnološke vede
- ☐ Medicinske in zdravstvene vede
- ☐ Kmetijske vede
- ☐ Družbene vede
- ☐ Humanistične vede
- ☐ Ne vem

A.1.7 Na kateri instituciji študirate?

Na to vprašanje odgovorite samo, če je zadoščeno naslednjim pogojem: Odgovor je bil *študent* pri vprašanju 1.4 (Vaš trenutni status)
Prosimo, izberite samo eno izmed možnosti:

- ○ Univerza v Mariboru
- ○ Univerza v Ljubljani
- ○ Univerza na Primorskem
- ○ Univerza v Novi Gorici
- ○ EMUNI univerza
- ○ Na enem izmed ostalih samostojnih visokošolskih zavodov
- ○ Na eni izmed višjih strokovnih šol
- ○ Drugo _____

A.1.8 Na kateri članici Univerze v Mariboru študirate?

Na to vprašanje odgovorite samo, če je zadoščeno naslednjim pogojem: Odgovor je bil *Univerza v Mariboru* pri vprašanju 1.7 (Na kateri instituciji študirate?)
Prosimo, izberite samo eno izmed možnosti:

- ○ Ekonomsko-poslovna fakulteta
- ○ Fakulteta za elektrotehniko, računalništvo in informatiko
- ○ Fakulteta za gradbeništvo, prometno inženirstvo in arhitekturo
- ○ Fakulteta za kemijo in kemijsko tehnologijo
- ○ Fakulteta za kmetijstvo in biosistemske vede
- ○ Fakulteta za logistiko
- ○ Fakulteta za naravoslovje in matematiko
- ○ Fakulteta za organizacijske vede
- ○ Fakulteta za strojništvo
- ○ Fakulteta za turizem
- ○ Fakulteta za varnostne vede

- ○ Fakulteta za zdravstvene vede
- ○ Filozofska fakulteta
- ○ Medicinska fakulteta
- ○ Pedagoška fakulteta
- ○ Pravna fakulteta
- ○ Drugo _____

A.1.9 V kateri program ste trenutno vpisani?

Na to vprašanje odgovorite samo, če je zadoščeno naslednjim pogojem: Odgovor je bil *Fakulteta za elektrotehniko, računalništvo in informatiko* pri vprašanju 1.8 (Na kateri članici Univerze v Mariboru študirate?)
Prosimo, izberite samo eno izmed možnosti:

- ○ 1. stopnja E
- ○ 1. stopnja R-IT
- ○ 1. stopnja ITK
- ○ 1. stopnja TK
- ○ 1. stopnja MK
- ○ 1. stopnja GING E
- ○ 1. stopnja MEH
- ○ 2. stopnja E
- ○ 2. stopnja R-IT
- ○ 2. stopnja ITK
- ○ 2. stopnja TK
- ○ 2. stopnja MK
- ○ 3. stopnja E
- ○ 3. stopnja RI
- ○ 3. stopnja MK
- ○ Drugo _____

A.2 Ravnanje na spletu

A.2.1 Koliko se strinjate z naslednjimi trditvami?

Prosimo, izberite primeren odziv za vsako trditev:

	Sploh se ne strinjam	Se ne strinjam	Se niti ne strinjam niti nisem proti	Se strinjam	Se popolnoma strinjam
Imam veliko izkušenj ob uporabi spleta.	○	○	○	○	○
Mislim, da se zavedam nevarnosti na spletu.	○	○	○	○	○
Pripravljen sem dovoliti spletni strani dostop do moje lokacije.	○	○	○	○	○
Pripravljen sem dovoliti spletni strani dostop do moje kamere.	○	○	○	○	○
Redno uporabljam socialna omrežja (Facebook, Twitter...).	○	○	○	○	○
Anonimnost na spletu mi je pomembna.	○	○	○	○	○

A.3 Dostopnost podatkov

A.3.1 Menim, da je možno, da spletna stran dostopa brez mojega dovoljenja do naslednjih podatkov o moji napravi:

Prosimo, izberite primeren odziv za vsako trditev:

	Sploh se ne strinjam	Se ne strinjam	Se niti ne strinjam niti nisem proti	Se strinjam	Se popolnoma strinjam	Pojma ne poznam
Tip naprave (računalnik, pametni telefon, tablica...)	○	○	○	○	○	○
Model pametnega telefona ali tablice	○	○	○	○	○	○
Število jeder v procesorju	○	○	○	○	○	○
Ločljivost zaslona	○	○	○	○	○	○
Stanje baterije	○	○	○	○	○	○
Operacijski sistem (Windows, Android, iOS, MacOS...)	○	○	○	○	○	○
Brskalniški vtičniki (dodatek, angl. plugin)	○	○	○	○	○	○
Jezikovne nastavitve brskalnika	○	○	○	○	○	○
Moje polno ime	○	○	○	○	○	○
Zgodovina brskanja (spletne strani, ki sem jih že obiskal)	○	○	○	○	○	○
Število strani, ki sem jih obiskal v trenutnem zavihku	○	○	○	○	○	○
Naslov IP	○	○	○	○	○	○
Naslov MAC	○	○	○	○	○	○
Telefonska številka	○	○	○	○	○	○
Ponudnik internetnih storitev (Telekom, T2, Telemach...)	○	○	○	○	○	○
Lokacija	○	○	○	○	○	○
Usmeritev mobilne naprave (žiroskop)	○	○	○	○	○	○

Vaš odgovor bomo zabeležili in ga upoštevali v naši raziskavi.

Prosim, če lahko kliknete na naslednjo povezavo in sodelujete v spletni aplikaciji:

https://spo.um.si/

Hvala za sodelovanje!