UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Marino Lukovič

ZAVEDANJE UPORABNIKOV O ZASEBNOSTI/PRIVATNOSTI V HTML5 SPLETNIH APLIKACIJAH

Diplomsko delo

Maribor, september 2016

ZAVEDANJE UPORABNIKOV O ZASEBNOSTI/PRIVATNOSTI V HTML5 SPLETNIH APLIKACIJAH

Diplomsko delo

Študent:	Marino Lukovič
Študijski program:	Univerzitetni študijski program
	Informatika in tehnologije komuniciranja
Smer:	Informacijski sistemi
Mentor:	doc. dr. Boštjan Šumak

ZAHVALA

Zahvaljujem se mentorju doc. dr. Boštjanu Šumaku za usmerjanje, vodenje in strokovno pomoč pri opravljanju diplomskega dela. Posebna zahvala velja staršem, ki so mi študij omogočili in me med njim podpirali in spodbujali.

Zavedanje uporabnikov o zasebnosti/varnosti v HTML5 spletnih aplikacijah

Ključne besede: HTML5, spletne aplikacije, varnost, zasebnost, HTML5 API

UDK: **004.439:004.55HTML'232(043.2)**

Povzetek

V diplomskem delu smo obravnavali zasebnost podatkov v spletnih aplikacijah HTML5. Pregledali smo funkcionalnosti, ki zagotavljajo zasebnost uporabnikov na spletu. Preverjali smo zavedanje uporabnikov o zasebnosti in deljenju zasebnih podatkov, urejenih z različnimi predpisi. V obliki eksperimenta, je bil prikazan praktičen del. Znotraj eksperimenta smo analizirali pozornost uporabnikov na varnostna obvestila.. Preverili smo ali vedo, katere podatke delijo in zakaj. V zaključku smo pridobljene rezultate analizirali in ovrednotili. Ugotovili smo, da ljudje premalo časa posvetijo prebiranju varnostnih obvestil. Večina jih ne ve, kakšne so njihove pravice do zasebnosti podatkov na spletu. Glavnina pa jih sploh ne pozna funkcionalnosti HTML5.

The awareness of users about privacy/security in HTML5 web applications

Key words: HTML5, web applications, security, privacy, HTML5 API

UDK: **004.439:004.55HTML'232(043.2)**

Abstract

The diploma work presents data privacy in HTML5 web applications. We also reviewed functionalities, that ensure user protection on the web. We tested the user awareness about their privacy and sharing private information governed by different rules. In the form of the experimen,t we presented practical part. Inside of it we analyzed, if users pay attention to security warnings. We also checked, whether they know, what information is shared and why. In conclusion we analyzed obtained results and evaluated them. We have found out, that people devote small amount of time to reading security notes. Also, most of them do not know about their rights to online privacy. Many of them do not even know any of HTML5 functionalities.

KAZALO

KAZALO SLIK

KAZALO TABEL

KAZALO GRAFIKONOV

UPORABLJENE KRATICE

ACTA – Anti-Counterfeiting Trade Agreement,

API – Application Programming Interface

CD – Compact Disc

CDM – Clean Development Mechanism

CDMA ID – Code Division Multiple Access Identificator

CORS – Cross-Origin Resource Sharing

CRSF – Cross-Site Request Forger

CSS – Cascading Style Sheets

CSS3 – Cascading Style Sheets 3

DDoS – Distributed Denial of Service

DMS – Data Management System

DNS – Domain Name System

DOM – Document Object Model

DoS – Denial of Service

EU FP7 – European Commission 7 th Framework Programme for Research and Technological Development

EU – European union

FQDN – Fully Qualified Domain Name

FTP – File Transfer Protocol

GPS – Global Positioning System

GSM – Global System for Mobile Communication

HD – DVD – High Definition Digital Versatile Disk

HTML 4 – Hypertext Markup Language 4

HTML - Hypertext Markup Language

HTML 5 - Hypertext Markup Language 5

HTTP – HyperText Transfer Protocol

HTTPS - HyperText Transfer Protocol Secure

ID - Identificator

IE5 – Internet Explorer 5

IETF - Internet Engineering Task Force

IndexedDB – Indexed Database

iOS – iPhone Operation System

IP – Internet Protocol

JSON – JavaScript Object Notation

KB – Kilobytes

MAC – Macintosh Computer

P3P – Platform for Privacy Preferences Project

RFC – Request for Comments

RFID – Radio-Frequency Identification

SQL – Structured Query Language

SQLi – Structured Query Language Injection

SSL – Secure Sockets Layer

TCP – Transmission Control Protocol

TLS – Transport Layer Security

UA – User Agent

URL – Uniform Resource Locator

UTF-8 – Unicode Transformation Format

W3C – World Wide Web Consortium

WebSRT - Web Resource Subtitle Tracks

WHATWG – Web Hypertext Application Technology Working Group

Wi-Fi – Wireless Local Area Network

WSS – WebSocket Secure

XCP – Universal Measurement and Calibration Protocol

XHTML – Extensible HyperText Markup Language

XHTML 2 – Extensible HyperText Markup Language 2

XML - Extensible Markup Language

XSS - Cross-Site Scripting

1 Uvod

Uporabniki v vedno večjem številu uporabljajo spletne aplikacije. Raziskava, ki jo je izvedel Morgan Stanley, je pokazala, da je število uporabnikov spletnih rešitev v letu 2015 bilo okoli 1,8 bilijonov. Kar 91% ljudi pa uporablja računalnike, za dostop do le teh. Govorimo torej o programskih aplikacijah, ki se izvajajo na spletnih brskalnikih klienta. [8]

Vedno bolj se uporabljajo rešitve, ki so razvite na podlagi tehnologije HTML5. Gre za aplikacije odprtega spleta, ki so v bistvu aplikacije standarda HTML5. Odprti splet je gibanje, ki ima posebno vlogo za javne, kooperativne in standardne World Wide Web komunikacije in je nasprotje zasebnih, ekskluzivnih lastniških spletnih rešitev.[55] Glavne vloge pri takšnih aplikacijah, igra standard HTML, ki ga uporabljamo za ustvarjanje vsebine spletnih aplikacij, slogovni jezik CSS3, ki ga uporabljamo za vizualno predstavitev in jezik JavaScript, ki nudi določen nivo dinamike in uporabniške interakcije za spletno aplikacijo. [18]

Privatnost in zasebnost v aplikacijah standarda HTML5, pomeni varovanje osebnih podatkov naših uporabnikov, pred zunanjimi vdori oz. grožnjami, ki pretijo na spletu. Zasebnost in varnost na spletu vključuje pravice in pooblastila glede osebne varnosti shranjevanja, zagotavljanja informacij tretjim osebam in prikazovanja informacij preko spleta. Zasebnost lahko vsebuje osebne podatke, s katerimi identificiramo osebo ali podatke, ki ne identificirajo osebe, ampak nam nudijo drugačne informacije, na primer vedenje uporabnikov na spletni strani. Starost in fizični naslov lahko opredelita kdo je uporabnik, brez izrecnega razkritja imena uporabnika, saj sta ta dva faktorja dovolj unikatna, da lahko natančno določimo osebo. [33]

Z vedno večjo uporabo spletnih aplikacij HTML5, se prav tako pojavlja vedno več napadov na le te, z namenom, da se pridobijo in izkoristijo podatki uporabnikov. Takšni napadi so na primer [59]:

- **CORS & CSRF napadi** – napad, kjer lahko napadalci dosežejo to, da naložijo dokumente z uporabo žrtvinega računa in pridobijo pravice, s katerimi lahko uporabljajo njihov račun in posledično zlorabijo njihove osebne podatke,

- **Web Sockets** – napad kjer napadalci pridejo mimo našega požarnega zidu in s tem pridobijo dostop do notranje vsebine. Notranjo vsebino lahko nato zlorabijo in jo uporabijo sebi v prid, z namenom delati slabo uporabniku,

- **SQLi & Blind Enumeration** – če je naša aplikacija ranljiva za napade XSS, lahko napadalec ukrade informacije iz baz WebSQL in jih prenese preko domen. V primeru, da bi podatki, ki bi jih pridobili bile zadnje transakcije banke, bi lahko uporabnike finančno zlorabili,

- **Web Storage in DOM ekstrakcija informacij** – ko je lokalna shramba dostopna preko JavaScript kode, napadalcu omogočimo, da ukrade naše informacije preko napada XSS, če je aplikacija ranljiva na napade XSS. Tako lahko napadalec pridobi podatke iz lokalne shrambe, ali pa informacije o seji, kjer lahko podatke uporabijo za zlorabo uporabnika.

Številni vidiki tehnologije se hitro razvijajo. To velja tako za strojno, kot za programsko opremo, ter vmesnike za programiranje aplikacij. Specifikacija HTML5 ni enotna specifikacija, ampak je zbirka izbranih, rahlo povezanih specifikacij, ki zajemajo vrsto različnih tehnologij. Zaradi tega je zelo težko, če ne kar nemogoče, da se ustvari dokončna analiza standarda dokler ni dokončan. Poleg tega se vsak večji spletni brskalnik ali mobilna platforma sama odloči, kateri sklop specifikacij se bo implementiral.. Za administratorja podjetja, bi bilo skoraj nemogoče in zamudno opravilo, da bi določil združljivosti vseh spletnih brskalnikov in platform.[11]

Zaradi hitrega razvoja standarda HTML5, se je včasih težko zanesti na javne informacije. Obstaja veliko blog člankov in spletnih strani, ki so usmerjene v uporabo standarda HTML5, spletne brskalnike in razvijanje iger. Napisanih je bilo veliko knjig o kompatibilnostih, ki bodo določile, če je brskalnik zmožen uporabiti točno določeno funkcijo HTML5. To so prav zagotovo uporabna orodja, ampak če je blog, članek ali spletna stran le nekaj mesecev zastarela, lahko pride do netočnosti podatkov z morebitnimi varnostnimi posledicami.[11]

Medtem ko je za standard HTML5 mišljeno, da je standard v strožjem pomenu izraza, so ponudniki aplikacij in strojne opreme pogosto omejeni zaradi njihovih trenutnih platform, ali različic specifikacije, ki jo razvijajo. Pogosto si bodo različni razvijalci drugače razlagali isto specifikacijo. Zaradi tega razloga, bodo različne aplikacije (vključno s spletnimi brskalniki), pogosto implementirale isto funkcionalnost, na nekoliko drugačen način.[11]

Zaskrbljenost nad standardom HTML5 in njegovim šibkim varovanjem zasebnosti, je bila vidno izražena na naslovni strani članka New York Times, 10. oktobra 2010. Nova spletna koda prinaša zaskrbljenosti nad tveganjem zasebnosti, kjer je večina govora o dodatnih možnostih sledenja, ki so omogočene z novimi shranjevalnimi zmožnostmi standarda HTML5. Posebno zlovešč primer je aplikacija Evercookie. Gre za aplikacijo JavaScript, ki zapisuje podatke za sledenje, na številne različne kraje v brskalniku uporabnika. Zaradi tega, so ti podatki težje odstranljivi. Še huje, Evercookie bo ponovno ustvaril vse piškotke, če bo ugotovil, da so bili odstranjeni. [53]

Evercookie je bil ustvarjen z namenom, da bi se prikazalo, s kakšno lahkoto, je lahko novi mehanizem za shranjevanje uporabljen za sledenje uporabnikom. Tržniki so bili pozorni na to in so hitro uporabili Evercookie za sledenje uporabnikom.[53]

Podobne večje zlorabe osebnih podatkov so bile na primer:

- Organizaciji Google, je leta 2015, grozilo do 15 milijonov denarne kazni, v kolikor ne bi prenehali s kršenjem zasebnosti uporabnikov na Nizozemskem. Podjetje je kršilo zakon o varstvu osebnih podatkov, z uporabo osebnih podatkov, kot so na primer zgodovina brskanja in lokacija, da bi jim nudili prilagojene oglase. Podjetju Google, je bil dan čas do konca leta, da spremeni obdelovanje podatkov, ki jih prejme od posameznih uporabnikov spleta. Podjetje so preiskovali v petih drugih evropskih državah – Franciji, Nemčiji, Veliki Britaniji, Italiji in Španiji. Od organizacije Google je bilo nato zahtevano, da se uporabnike informira pred uporabo njihovih zasebnih podatkov, saj sicer to pomeni kršenje zakon. [12]

- Podjetje Sony je začelo z uporabo proti piratskih ukrepov leta 2005, ki jih je dodajalo na glasbene zgoščenke. Ko je stranka predvajala enega od teh zgoščenk na sistemu Windows, je zlonamerna programska oprema na zgoščenki, podjetju Sony poslala IP naslov računalnika, na katerem se je zgoščenka predvajala. Ta vohunska oprema je ustvarila ranljivosti, ki so jih lahko črvi in virusi izkoristili. Podjetje Sony je zaradi tega, ustvarilo zadnja vrata na računalnikih svojih strank. Slednje je pripeljalo do tega, da je podjetje svojim uporabnikom, ponudilo brezplačno orodje, za odstranitev zlonamerne programske opreme. Organizacija Sony je nato morala plačati vsakemu uporabniku, katerega računalnik je bil poškodovan s programsko opremo, 150 dolarjev.[15]

Še eno podjetje, ki krši privatnost in varnost podatkov je Snapchat. Problemi zasebnosti in varnosti podatkov na katere lahko naletimo so [5]:

  - Prejemniki lahko shranijo vaše fotografije, čeprav aplikacija obljublja, da se slik ne shranjuje. Skupina razvijalcev je razvila aplikacije, ki omogočajo prenos slik in videov zajetih z aplikacijo Snapchat, brez vedenja uporabnika,

- Aplikacija Snapchat lahko prav tako dostopa do vaše lokacije, saj družba posreduje lokacijo, glede na informacijo signala Wi-Fi in prenosnega telefona iz mobilnih naprav Androidovih uporabnikov, do svojega ponudnika analitičnih sledilnih storitev,
- Pri dodajanju prijateljev preko telefonske številke na sistemu iOS, je prišlo do tega, da je aplikacija zbrala imena, telefonske številke in vse kontakte v mobilni napravi uporabnika, brez da bi se uporabnik zavedal.

Odkar je John von Neumann, objavil teorijo o samo – ponovljivih programih, leta 1949, so se napadi na računalniške sisteme razvili, kot tudi napadi na spletne aplikacije. Eden prvih večjih napadov na spletne aplikacije, leta 2000, je bil obsežen DDoS (Napad za zavrnitev storitve) napad, na podjetja Yahoo, eBay, Amazon, DATEK in več drugih spletnih strani.[61]

Spletni strežniki so redne tarče napadov. Običajno so na voljo 24 ur na dan, sedem dni na teden, 365 dni v letu. Tako sta ročen in avtomatski napad, možna kadarkoli. Konzorcij za varstvo spletnih aplikacij, je leta 2008 naredil študijo. Ta je pokazala, da ima ranljivosti, kar 97554 izmed 12186 testiranih spletnih strani. Podjetje WhiteHat Security je testiralo okoli 2000 spletnih strani. Študija je prikazala, da ima povprečna spletna stran okoli 13 pomanjkljivosti. Poročilo s strani podjetja Verizon, o podatkovnih kršitvah, leta 2010 piše, da je v zadnjih šestih letih, bilo okoli 900 kršitev varnosti, s čimer je bilo več ko 900 milijonov ogroženih zapisov.[61]

Končni uporabniki so prav tako tarče mnogih napadov. Podjetje Kaspersky Lab je leta 2010, poročalo o 76,619,767 napadih na njihove uporabnike. Podjetje Secunia pa je javilo, da je bilo ugotovljenih veliko več ranljivosti v aplikacijah tretjih oseb, kot pa v programih podjetja Microsoft. To je posebej zanimivo v kontekstu spletnih brskalnikov. Za brskalnik Internet Explorer, je bilo sporočenih 51 ranljivosti, pri brskalniku Mozilla Firefox pa 95 (moramo upoštevati, da vse ranljivosti niso enako kritične). Leta 2010, je podjetje Symantec v poročilu zapisalo, da je bilo odkritih več kot 3339,600 različnih škodljivih programskih oprem, v e-poštah.

Prav tako je bilo blokiranih več kot 188,6 "phishing" (nezakoniti način zavajanja uporabnikov, namenjenega pridobivanju tujih občutljivih osebnih podatkov) e-pošt in 42,962 različnih domen, ki so gostile zlonamerno vsebino. Napadenih je bilo 90% legitimnih strani. Že samo v prvem četrtletju leta 2010, so pri podjetju Kaspersky, zabeležili 327,598,028 napadov na uporabnike računalnikov.[61]

Varne implementacije potrebujejo jasne specifikacije, obvladljiv obseg dela, temeljito in raznoliko testiranje, hitre in natančne povratne zanke ter hitro in celovito uvajanje sprememb. Nič od tega pa zaenkrat ne podpira specifikacija HTML5, kar pripelje do neusklajenih in nenehno razvijajočih se specifikacij. [22]

V aplikacijah HTML5, je prav tako možno ugrabiti obrazce z novim "form" atributom. Prav tako je možna kraja osebnih podatkov, preko samodejnega izpolnjevanja. Pojavljajo se tudi napadi za zavrnitev storitev, obhod črnih list z novimi nadzorniki dogodkov in uporaba neškodljivih atributov za izvajanje JavaScript kode. [22]

V okviru naloge smo želeli na osnovi metode eksperimenta in anket, odgovoriti na naslednja raziskovalna vprašanja:

RV1: Koliko časa uporabniki namenijo zasebnosti podatkov ob uporabi elementov/funkcionalnosti HTML5?

RV2: Zakaj so se uporabniki pri funkcionalnosti odločili deliti podatke / jih ne deliti?

RV3: So izpustili kakšno funkcionalnost, zaradi želje po ne deljenju osebnih podatkov?

RV4: Koliko uporabnikov je prebralo vsako sporočilo brskalnika (npr. dovoljenje za dostop do geo lokacije), s čimer se strinjajo, z deljenjem zasebnih informacij?

RV5: Kaj uporabnike moti pri opozorilih, ob uporabi elementov HTML, povezanih z zasebnostjo in varnostjo?

RV6: Ali so varnostna sporočila povezana z uporabo funkcionalnosti HTML, za uporabnike sprejemljiva ali nadležna?

RV7: Kolikšen delež uporabnikov se strinja z deljenjem informacij zasebne narave, ob uporabi elementov/funkcionalnosti HTML5?

Med samo izvedbo analize empiričnih podatkov, želimo potrditi naslednje raziskovalne hipoteze. Hipoteze, ki smo jih potrjevali so:

H1: Uporabniki pri uporabi funkcionalnosti HTML5, zasebnosti podatkov ne bodo namenili veliko časa.

H2: Zaradi nezaupljivosti, uporabniki ne bodo želeli deliti podatkov.

H3: Uporabniki določenih funkcionalnosti ne bodo uporabljali, ker ne bodo želeli deliti podatkov.

H4: Večina uporabnikov ne bo prebrala vseh varnostnih sporočil.

H5: Uporabnike bo najbolj motilo to, da se opozorila pri uporabi elementov HTML5, velikokrat pojavijo.

H6: Večina uporabnikov, bo varnostna sporočila dojela kot nezaželena in nadležna.

H7: Večji delež uporabnikov, ne bo delil informacij zasebne narave, pri uporabi funkcionalnosti HTML5.

Pri diplomski nalogi bomo prav tako izvajali praktični del v obliki eksperimenta. Slednji bo potekal sledeče:

Pridobili bomo uporabnike, na katerih bomo izvedli eksperiment. Uporabnike bomo nato posamezno posedli v sobo z računalnikom in jim dodelili naloge, ki jih morajo opraviti na spletnih aplikacijah HTML5. Spremljali jih bomo pri opravljanju nalog in uporabi funkcionalnosti, prav tako pa bomo beležili njihovo obnašanje pri opravljanju eksperimenta. Nato bomo preverili, kako se bodo odzvali pri deljenju osebnih podatkov. Po opravljenih nalogah pa bodo rešili še anketo, nato pa bomo zabeležene podatke še ovrednotili.

Podatke bomo pridobili s pregledom videov naprave Eye tracker Guide. Te pa bomo s programom Analyze, ki nam omogoča različne preglede sledenja očem, analizirali. Tako bomo lahko videli, kam so ljudje največkrat gledali, koliko časa so namenili zasebnosti,… Podatke bomo shranili v obliki slik, ki nam prikazujejo, kam so bile uporabnikove oči usmerjene, v določenem momentu. Po končanem eksperimentu, bo vsak uporabnik rešil anketo. Nato bo sledila anketa za vse informacije in podatke, ki jih nismo mogli uspešno pridobiti s samim eksperimentom.

Cilj naloge je predvsem, da pridobimo informacije od uporabnikov, koliko časa namenijo zasebnosti in privatnosti pri aplikacijah HTML5. Prav tako želimo izvedeti, koliko se jih zaveda, da določene funkcionalnosti obstajajo, jih preberejo, preverijo in uporabijo. Želimo pa tudi izvedeti, kaj si o teh funkcionalnostih mislijo, kolikim ni težava deliti osebnih informacij, ter koliko je takih, ki si informacij ne upajo deliti, ali pa jim je vseeno. Posledično lahko pridobimo informacije, kaj jih moti, kaj jim je všeč in zakaj so se odločili prezreti/prebrati določeno varnostno sporočilo. Prav tako bomo izvedeli, kaj jih pripravi do tega, da izberejo določeno pot in zakaj ne želijo deliti informacij, ali pa jih to ne moti. Preverili pa bomo tudi funkcionalnosti, ki jih aplikacije HTML5 nudijo za zaščito uporabnika, pred uporabo osebnih podatkov, na osnovi standarda in pregledali, kakšne tipe podatkov le te od uporabnika zahtevajo, da deli.

V diplomski nalogi bomo opisali problem, s katerim se spopadamo in kakšni so razlogi, da želimo ta problem rešiti, kaj s tem dosežemo in kako pridemo do željenega končnega učinka. V naslednjem poglavju bomo opisali standard HTML5, kaj sploh je in za kaj ga uporabljamo.

Nato bomo preverili najnovejšega izmed standardov HTML, kaj prinaša, kakšne so slabosti in prednosti ter katere principe delovanja ima. Prav tako bomo preverili različne namene uporabe standarda HTML5, pregledali zahteve, oziroma smernice pri razvijanju aplikacij HTML5, ki jih imajo razvijalci. Sledi opis aplikacijskih vmesnikov, ki jih predstavlja novi standard HTML5.

Naslednje poglavje zajema področje varnosti. Tukaj smo preverili, s kakšnimi varnostnimi grožnjami se srečujejo razvijalci in kako jih lahko odpravimo. Preverili smo aplikacijski programski vmesnik, za vmesnikom. Pregledali smo, katere informacije delimo pri določenem vmesniku, kako lahko do njih pridejo nepridipravi in kaj lahko z njimi naredijo. Zapisali pa smo tudi, katere funkcionalnosti nam pomagajo pri zmanjševanju zlorab in groženj, v spletnih aplikacijah HTML5.

Sledeče poglavje je namenjeno tehnologijam za sledenje očem, kjer smo spoznali, kako te delujejo in za kaj jih lahko uporabimo.

Za konec smo opisali raziskavo, ki smo jo izvedli. Navedli smo, kakšne pristope smo pri raziskavi uporabili, kako je raziskava potekala in katere metode in orodja smo pri izvajanju prakticirali.

2 Standard HTML5

Gre za rešitve, ki so razvite na podlagi tehnologije HTML5. Govorimo o aplikacijah odprtega spleta. Le te so sestavljene iz gradnikov: jezik HTML za ustvarjanje vsebine spletnih aplikacij, slogovni jezik CSS3 za vizualno predstavitev in jezik JavaScript, ki nudi določen nivo dinamike in uporabniške interakcije za spletno aplikacijo. Je označevalni jezik za strukturiranje in prezentacijo vsebine, za World Wide Web (porazdeljen hiper-tekstni sistem) in ključna tehnologija interneta.[76]

2.1 Principi delovanja

Specifikacija se pogosto v istem kontekstu nanaša na atribute in jezike HTML ter XML, za opis vmesnikov. Kadar ni jasno, na koga se nanaša, se na njih sklicujemo, kot atribute vsebine za HTML ter XML in jezike za opise vmesnikov. Podobno se izraz "lastnosti" uporablja, tako za jezik JavaScript lastnosti objektov, kot za lastnosti CSS. [80]

Na splošno, ko specifikacija določa, da funkcija velja za sintakso HTML ali sintakso XHTML, prav tako vključuje druge sintakse. Ko specifikacija velja le za enega od obeh jezikov, je izrecno povedano, da ne velja za drug format. Na primer: "Velja samo za sintakso HTML".[80]

Principi delovanja, ki jih ima standard HTML5 so sledeči [48]:

- Don't break the web

Pomeni, da standard ne sme uvesti spremembe, ki bi pripeljala do tega, da bi spletne strani drugih ljudi prenehale delovati. To se zgodi le redko. Prav tako pomeni, da standard ne sme mimogrede spremeniti pravila, saj v procesu, povsem dobre strani postanejo zastarele (čeprav še vedno delujejo). Na primer, jezik XHTML 2 je zlomil splet, ker je zahteval takojšnjo dramatično spremembo, v načinu pisanja spletnih strani. Stare strani bi še vedno delovale (zaradi združljivosti za nazaj, ki je vgrajena v brskalnike).

Če bi se želeli pripraviti na prihodnost in ohraniti spletno stran posodobljeno, bi bili prisiljeni porabiti veliko časa za popravljanja "napak", ki jih je jezik XHTML 2 prepovedal.

- Pave the Cowpaths

»Cowpath« je groba, intenzivno uporabljena proga, ki ljudi pripelje od ene do druge točke. Morda ni najboljši možni način za "premikanje", vendar je bila na neki točki najbolj praktična delujoča rešitev.

Specifikacija HTML5 želi poenotiti te neuradno (ampak pogosto uporabljene) tehnike. Verjetno ne bo tako enostavno, ampak ima veliko možnost uspeha. To pa je zato, ker je prehod na nove tehnike lahko prezahteven, ali pa ni v interesu povprečnega spletnega oblikovalca. Še slabše, nove tehnike mogoče ne bodo delovale za obiskovalce, ki uporabljajo starejše brskalnike. Jezik XHTML 2, je poizkušal odgnati ljudi iz te poti, kar pa jim je spodletelo.

- Be practical

Ta princip je enostaven – spremembe morajo imeti praktičen namen in bolj kot so zahtevne, večji mora biti izkupiček. Spletni oblikovalci imajo raje lepo oblikovane, konsistentne, domiselne standarde, vendar to ni dovolj dober razlog za spremembo jezika, ki je bil uporabljen v več milijard dokumentov. Seveda pa še vedno mora nekdo določiti, katere skrbi so pomembnejše. Zaželeno je pogledati, kaj spletne strani že počnejo, ali poskušajo narediti.

Na primer podjetje Youtube se je moralo zanašati na dodatek Flash za brskalnik, saj standard HTML ni imel funkcij za video pred specifikacijo HTML5. Ta rešitev je delovala presenetljivo dobro, saj je bil dodatek Flash, prisoten skoraj na vseh računalnikih, povezanih s spletom.

2.2 Uporaba

Standard je namenjen avtorjem dokumentov in skript, ki uporabljajo funkcije opredeljene v tej specifikaciji. Prav tako je namenjen izvajalcem orodij, ki delujejo na straneh ter uporabljajo funkcije definirane v specifikaciji. Uporabljajo ga tudi posamezniki, ki želijo ugotoviti pravilnost dokumentov, ali izvedbe glede na zahteve te specifikacije.[79]

Standard je zasnovan tako, da ga lahko uporabljajo vsi spletni razvijalci, ki želijo uporabljati tehnologije standarda HTML5. Te so razvrščene v več skupin, glede na njihovo funkcijo. Le te so [29]:

- **Semantika:** omogoča natančnejši opis tega, kaj je vaša vsebina,
- **Povezava:** omogoča komunikacijo s strežnikom, na bolj inovativne načine,
- **Način dela brez povezave in shranjevanje:** omogoča spletnim stranem lokalno shranjevanje podatkov, na strani odjemalca in delovanje brez povezave,
- **Multimedija:** dodajanje avdio in video vsebine,
- **2D / 3D grafika in efekti:** omogoča veliko bolj raznoliko paleto možnosti predstavitev,
- **Uspešnost in integracija:** zagotavlja večjo hitrost optimizacije in učinkovitejšo uporabo računalniške strojne opreme,
- **Dostop do naprav:** omogoča uporabo različnih vhodnih in izhodnih naprav.

2.3 Novi elementi standarda HTML5

Znotraj tega podpoglavja, bomo prikazali vse nove elementi, ki so definirani znotraj standarda. Poleg opisa, bomo še opisali še funkcionalnosti, ki jih ponujajo za doseganje zasebnosti podatkov uporabnikov in na kakšne ranljivosti lahko znotraj njih naletimo.

2.3.1 Forms API

Obrazci HTML so preproste kontrole HTML, ki jih uporabljamo, da zberemo podatke s strani obiskovalcev spletne strani. Vključujejo vnosna polja, v katera lahko ljudje vnašajo podatke, seznam polj iz katerih izbiramo, potrditvena polja, ki jih lahko vklopimo ali izklopimo in tako naprej. Obstaja mnogo načinov za uporabo obrazcev HTML in če smo se ukvarjali s spletom več kot teden, smo jih uporabljali za opravljanje vseh različnih opravil. Takšno opravilo je na primer prijava za e-poštni račun. [49]

Pisanje obrazcev je sestavljeno iz več korakov, ki jih je mogoče opraviti v poljubnem vrstnem redu. Ti koraki so pisanje uporabniškega vmesnika, implementacija obdelovanja na strani strežnika in konfiguracija uporabniškega vmesnika, za komunikacijo s strežnikom. [81]

Standard HTML5 prav tako predstavlja nove vnosne tipe, ki jih lahko uporabljamo v vnosnem elementu. Do zdaj smo lahko uporabljali le elemente text, radio, checkbox, password, file in submit. V standardu HTML5 pa smo pridobili nekaj novih vnosnih tipov. To so na primer iskanje, pošta, naslov URL, telefon, razpon kot drsnik, število kot spinner, datum in čas ter izbira barve. [68]

Obrazci HTML5 zajemajo veliko število novih aplikacijski programskih vmesnik in tipov elementov. Da bi jih razumeli, bomo nove funkcionalnosti obravnavali tako, da jih bomo razdelili na nove vnosne tipe, nove funkcije in atribute. [40]

Tabela 2.1 : Elementi aplikacijskega programskega vmesnika obrazec ; povzeto po [40].

Tip	Namen
tel	Telefonska številka.
email	Vnosno polje za e-poštni naslov..
url	Spletna lokacija URL.
search	Niz ki zalaga iskalnik.
range	Numerični selektor znotraj obsega vrednosti, običajno vizualiziran kot drsnik.

Ti vnosni tipi (Tabela 2.1) v smislu programskih vmesnikom ne zagotavljajo ničesar novega. Za primere kot so tel, email, url in search ne obstajajo nobeni atributi, ki bi jih ločili od najpreprostejših vnosnih tipov besedila. [40]

Na voljo pa imamo tudi nekaj elementov, ki bodo znotraj obrazcev HTML5. (Tabela 2.2)

Tabela 2.2 : Dodatni elementi znotraj HTML5 obrazcev; povzeto po [40].

Tip	Namen
number	Polje ki vsebuje samo numerične vrednosti.
color	Selektor barv, ki ga lahko prikažemo s kolesom.
datetime	Polni prikaz datuma in ure, vključno s časovnim pasom.
datetime-local	Prikaz časa in datuma, brez možnosti ali indikacijo za časovne pasove.
time	Izbiralec in pokazatelj časa, brez informacij o časovnem pasu.
date	Izberemo koledarski datum.
week	Izberemo teden znotraj izbranega leta.
month	Izberemo mesec znotraj danega leta.

Vse kar bi bilo do zdaj zaželeno vedeti o obrazcih je [40]:

- Obrazci se morajo še vedno nahajati znotraj <form> značk, kjer nastavimo osnovne atribute,
- Obrazci še vedno pošiljajo vrednosti kontrol na strežnik, ko uporabnik ali programer aplikacije predloži stran,
- Vse znane kontrole obrazcev – besedilna polja, radio gumbi, potrditvena polja in tako naprej, so še vedno prisotni in delujoči kot prej, z nekaj manjšimi novimi dodatki,
- Kontrole obrazcev lahko še vedno skriptiramo (za tiste, ki želijo pisati lastne modifikatorje in nadzornike).

2.3.2 Audio and video API

Avdio in video datoteke, so v resnici datoteke, podobne arhivski datoteki ZIP, ki vsebuje številne datoteke. Avdio in video posnetki, so združeni v času izvajanja za predvajanje videa. Metapodatki pa vsebujejo podatke o videoposnetku, kot so naslov, podnaslov, zapis informacij in tako naprej. [43]

Specifikacija HTML5, obravnava vrzeli z dodajanjem <audio> in <video> elementov, ki jih je standard HTML, pogrešal vsa ta leta. Končno so bogati mediji dobili konsistentno, standardizirano podporo, ki ne potrebuje vtičnika. Glavna podjetja brskalnikov se nahajajo v vojni avdio in video formatov, ki je hujša kot Bluray proti HD-DVD. [50]

Posledica tega je, da ne obstaja nobeden avdio in video format, ki deluje na vsakem brskalniku. Prav tako bomo morali kodirati in odkodirati naše medijske datoteke, da bi jih pripravili do tega, da bi delovale v standardu HTML5. [46]

Video element je uporabljen za predvajanje videov ali filmov in avdio datotek z napisi. Vsebine se lahko zagotovijo znotraj video elementa. Uporabniški agenti ne smejo pokazati te vsebine uporabniku, saj je namenjena za starejše spletne brskalnike. Ti pa ne podpirajo video vsebin, zato da lahko uporabnikom teh starejših brskalnikov, pokažemo besedilo, ki jih obvešča o tem, kako se dostopa do video vsebin.[84]

Uporaba avdio elementa, nam izmed ostalih funkcij, omogoča funkcionalnost predvajanja za nazaj. Preko uporabe atributa kontrol, za bolj napredno uporabo, so lahko avdio predvajanja za nazaj in kontrole manipulirane s pomočjo uporabe vmesnika HTML Media API. Oziroma natančneje, funkcionalnosti definiranih v vmesniku HTMLAudioElement. Prav tako lahko vmesnik Web Audio API, uporabimo za neposredno generiranje in manipuliranje avdio tokov iz kode JavaScript. [28]

2.3.3 Canvas API

Je programski aplikacijski vmesnik, ki nam omogoča, da dinamično ustvarimo in prikažemo grafiko, grafe, slike in animacije. Ko uporabimo <canvas> element na naši spletni strani, se na strani ustvari pravokotno območje. Privzeto je to območje široko 300 in visoko 150 slikovnih pik, vendar pa lahko določimo natančno velikost in ostale atribute za naš <canvas> element. [44]

Slike narisane na platnu so dokončne in jih ni mogoče spreminjati na način, kot to lahko naredijo razširljive vektorske slike. Poleg tega objekti, ki so narisani na platnu, niso del strani programskega vmesnika DOM ali katerega koli imenskega prostora, kar pa smatramo kot ranljivost. Razširljive vektorske slike na drugi strani, pa lahko zmanjšamo brez težav, na različnih resolucijah, prav tako natančno vedo, kje smo kliknili na sliko.[38]

<canvas> element prav tako nudi skripte s platnom, ki je odvisen od resolucije, ki se lahko uporablja za prikaz grafov, grafike iger ali drugih vizualnih podob. Avtorji ne smejo uporabljati <canvas> elementa, če obstaja bolj primeren element. Neprimerno je uporabiti <canvas> element za prikaz glave strani, če je zaželena predstavitev glave grafično intenzivna. Takrat je potrebno to označiti z uporabo ustreznih elementov (običajno H1) in nato oblikovati z uporabo tehnologije CSS. [83]

Uporaba <canvas> elementa ni zelo težka, vendar pa potrebujemo osnovno razumevanje standarda HTML in jezika JavaScript. Element ni podprt v nekaterih starejših brskalnikih, vendar pa je podprt v zadnjih različicah vseh glavnih brskalnikov. Za risanje slik na platno uporabljamo kontekstni objekt JavaScript. [6]

2.3.4 Offline applications API

Če želimo videti spletno aplikacijo, potrebujemo povezavo z internetom. Vendar ne smemo pozabiti, da tudi spletne aplikacije niso trajno dosegljive. Namesto tega so zasnovane tako, da delujejo tudi v občasnih obdobjih, ko računalnik izgubi povezavo. Z drugimi besedami, spletna aplikacija brez povezave dopušča, občasne motnje v omrežju. [47]

Če želimo opaziti težave, lahko poizkusimo potovati skozi tunel, medtem ko delamo na spletni aplikaciji, z eno izmed teh naprav. Verjetno bomo dobili stran o napaki in bomo morali začeti znova na drugi strani. Če pa bomo to naredili na aplikaciji brez povezave, se nam to ne bo naredilo in se bomo izognili prekinitvam. Določene funkcije spletnih aplikacij, lahko postanejo trenutno nedosegljive, vendar pa aplikacija ne bo prenehala delovati. [47]

Da bi uporabnikom omogočili, da so še v naprej v stiku s spletnimi aplikacijami in dokumenti, tudi če njihova omrežna povezava ni na voljo, (na primer, ko potujejo izven področja pokrivanja njihovega internetnega ponudnika), lahko avtorji zagotovijo manifest. Ta navaja datoteke, ki so potrebne za spletne aplikacije, da delujejo brez povezave in omogočajo brskalnikom uporabnikov, da obdržijo kopijo datotek za delo brez povezave. [54]

Ni priporočeno, da naložimo naslove URL s strani strežnika, kot da smo vedno brez povezave. Bo pa to prekinilo vse možne vektorje, zaradi česar zlonamerni vzdrževalci ne bodo mogli zlorabiti politiko istega izvora, za dostop do naših podatkov. [26]

Grožnje kot so obstojni vektorski napadi in zastrupitev predpomnilnika, so definirane znotraj specifikacije HTML5. Da bi se rešili tega problema, je potrebno uporabnike naučiti, da vedno počistijo svoj predpomnilnik, kadar obiščejo internet preko nezavarovanega omrežja, preden želijo dostopati do strani, na katerih se shranjujejo občutljivi podatki. Prav tako se mora uporabnik naučiti razumeti pojem varnostnih opozoril in sprejeti aplikacije brez povezave, samo na zaupanja vrednih mestih. [63]

Z uvedbo teh aplikacij, so bile varnostne meje premaknjene. V spletnih aplikacijah, pred aplikacijami brez povezave, so bile odločitve HTML5 za nadzor dostopa, pri dostopu do podatkov in funkcij, izvedene samo na strani strežnika. Z uvedbo aplikacij brez povezave, so bili deli teh pregledov dovoljenj, preneseni na uporabniške agente. Zato implementacija zaščite spletnih aplikacij, samo na strani strežnika ni bila dovolj, če uporabljamo aplikacije brez povezave. Tarče napadov napadalnih spletnih aplikacij niso samo strežniki. Napad na del odjemalca spletne aplikacije brez povezave, je prav tako možen. To pa zlomi varnostne zahteve uporabniških agentov. Vendar s tem pride do tega, da so vse varnostne zahteve ogrožene implicitno. [60]

Prav tako pa se na standard HTML5 obrnemo, ko želimo uporabljati njegove različne in mnoge programske aplikacijske vmesnike, ki jih nudi.

2.3.5 Geolocation API

Z uporabo tega programskega aplikacijskega vmesnika, lahko na primer od uporabnika zahtevamo, da delijo svojo lokacijo in če se strinja, jim lahko na primer nudimo navodila, kako priti do najbližje trgovine, kjer so znižanja. Predstavljamo si lahko aplikacijo, ki bi jo vključili, ko bi začeli teči oziroma hoditi. Medtem ko smo na poti, aplikacija lahko meri kako daleč smo tekli.[41]

Ostale geo lokacijske HTML5 aplikacijske ideje, bi lahko bile v stilu navigacije GPS, ali pa aplikacije socialnega omrežja, ki omogočajo, da vidimo kje se nahajajo naši prijatelji. Tudi takšne, kjer lahko izberemo kavarno, ki jo bomo obiskali in še veliko več podobnih aplikacij. [41]

Gre torej za programski aplikacijski vmesnik, ki vrne lokacijo uporabnika, ki temelji na informacijah o baznih postajah in vozliščih Wi-Fi (brezžična tehnologija), ki jih lahko mobilni odjemalec zazna. Komunikacija poteka preko protokola HTTPS z uporabo metode POST. Zahteva in odgovor, sta oblikovana v JSON formatu. Za uporabo tega programskega aplikacijskega vmesnika, potrebujemo njegov ključ in mejne vrednosti njegove porabe . [71]

Specifikacija izrecno navaja, odkar je narava programskega aplikacijskega vmesnika, da izpostavi lokacijo uporabnika, kar pomeni kršenje zasebnosti uporabnikov, je pred nadaljevanjem potrebno pridobiti dovoljenje za poizkus pridobivanja informacij za geo lokacijo. Za to bo poskrbel brskalnik, kjer se bo sporočilo prikazalo kot pojavno okno, ali pa na vrhu brskalnika, ki bo zahtevalo uporabnikovo dovoljenje za nadaljevanje. [13]

Naprava lahko uporabi enega izmed sledečih virov [41]:

- Naslov IP, (
-)
- Triangulacija koordinat,
    - Global positioning system (gps), (
    -)
    - Wi-Fi z mac naslovi iz rfid, wi-fi and bluetoothom, (
    -)
    - GSM ali cdma id-ji prenosnih telefonov. (
    -)
    - Uporabniško definirani. (
    -)

Veliko naprav uporablja kombinacijo enega ali več virov, da bi zagotovile celo višjo natančnost. Vsaka od teh metod ima svoje slabosti in prednosti: [41]

Tabela 2.3 : Viri za določanje lokacije; povzeto po [41]

	Prednosti	Slabosti
Naslov IP	Dostopna vsepovsod, nadaljuje na strani strežnika	Ni zelo natančna (veliko krat napačne informacije, ampak natančna), je lahko draga operacija
GPS	Zelo natančen	Lahko traja dlje časa, da dobimo točno lokacijo, kar lahko izprazni baterijo uporabnika, ne deluje dobro v notranjih prostorih, možnost, da potrebuje dodatno strojno opremo
Wi-Fi geo lokacija	Natančna, deluje znotraj zgradb, jo lahko popravimo hitro in poceni	Ni dobra na podeželjih, kjer imajo manjše število brezžičnih dostopnih točk,
Geo lokacija prenosnih telefonov	Dokaj natančna, deluje znotraj zgradb, jo lahko popravimo hitro in poceni	Zahteva napravo s povezavo na mobilni telefon ali modem telefona, ni dobra na podeželjih, kjer imajo manjše število stolpov za mobilne telefone.
Uporabniško definirana geo lokacija	Uporabniki imajo lahko bolj natančne podatke o lokaciji, kot programske storitve, dovoli geo lokacijske storitve za alternativne lokacije, vnos uporabnika je lahko hitrejši kot detekcija	Je lahko tudi zelo nenatančna, še posebej, če pride do spremembe lokacije

Največji problem, na katerega lahko naletimo znotraj geo lokacijskega programskega aplikacijskega vmesnika, je sledenje uporabnikom. S tem, ko pridobimo lokacijo uporabnikov, ki so registrirani, lahko spremljamo njihovo gibanje. [59]

Težava zasebnosti vpliva predvsem na uporabnike, zato morajo biti naučeni, da ne smejo dovoliti spletnim aplikacijam, dostop do informacij o lokaciji. Prav tako morajo deliti, le določene podatke o lokaciji in le zaupanja vrednim ponudnikom. Vse zgoraj omenjene grožnje, ne moremo zmanjšati preko varne implementacije strežnika. [60]

Aplikacijski programski vmesnik, je torej uporabljen znotraj specifikacije HTML5, za pridobivanje zemljepisne lege gostujoče naprave. V skoraj vseh primerih, ta informacija prav tako vsebuje lokacijo uporabnikove naprave, s čimer se potencialno ogrozi uporabnikova zasebnost. Pravilna izvedba te specifikacije, mora vsebovati mehanizem, ki varuje uporabnikovo zasebnost, prav tako pa mora ta mehanizem zagotoviti, da informacije o lokaciji, niso na voljo preko tega aplikacijskega programskega vmesnika, brez izrecnega dovoljenja uporabnika. [57]

Geo lokacija RFC priporoča, da uporabniški agent vpraša uporabnika za dovoljenje, preden se pridobi lokacija. Od brskalnikov je odvisno, kako se odločitev zapomni in ali je ta zapomnjena. Določeni uporabniški agenti, zahtevajo od uporabnikov, da ponovno obiščejo stran, da se izklopi možnost, da bi se pridobila lokacija uporabnika brez vprašanja. Zato je iz varnostnih razlogov priporočeno, da se zahteva vnos uporabnika, preden se kliče metoda getCurrentPosition ali watchPosition. [56]

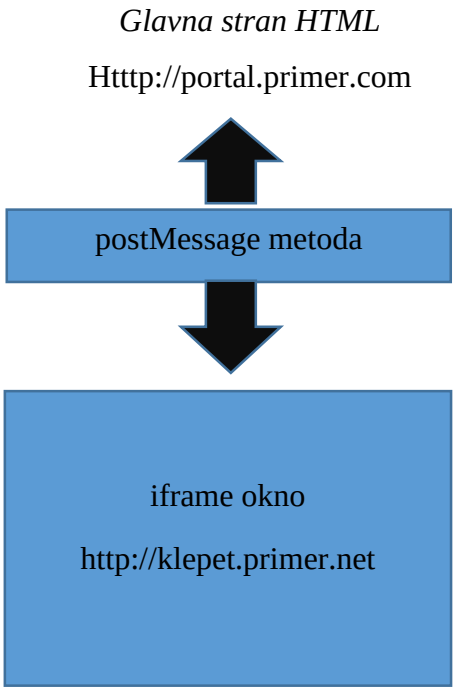
Vedno je zaželeno preveriti nastavitve zasebnosti na socialnih omrežjih in straneh, kjer delimo slike. Prepričati se moramo, da delimo informacije le s prijatelji in družino. Prav tako je zaželeno, da v svoje omrežje sprejmemo le ljudi, ki jih poznamo. Pozorni moramo biti tudi na to, da so lahko informacije, ki jih naložimo na splet povezane ena z drugo. Na primer fotografija, ki jo naložimo na omrežje Twitter, je lahko avtomatsko objavljena na našem profilu Facebook. Zaradi tega je pomembno, da vedno pregledamo nastavitve zasebnosti na vseh računih. Previdni moramo biti tudi, kdaj in katere slike delimo.
Namesto da naložimo sliko, ki razkriva našo lokacijo v trenutku ko se slikamo, počakajmo, da pridemo domov in jo nato naložimo. [64]

2.3.6 Communication API

Do nedavnega je bila komunikacija med okvirji, zavihki in okni, v tekočem brskalniku, v celoti omejena zaradi varnostnih pomislekov. Medtem, ko bi bilo priročno za določene strani, da bi delile informacije znotraj brskalnika, bi to prav tako odprlo možnosti za nove napade.

Če bi bila brskalnikom dana možnost, da bi lahko programsko dostopali do vsebine, ki je naložena v okvirjih in zavihkih, bi lahko strani ukradle katere koli informacije, ki bi jih lahko pridobile od vsebine drugih spletnih strani. [39]

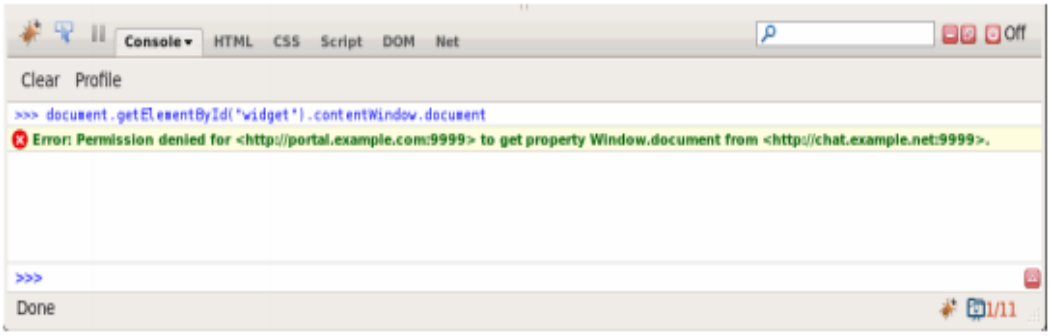
Ustvarjalci brskalnikov in organi za standardizacijo, so se dogovorili, da uvedejo novo funkcijo. Cross Document Messaging – CDM omogoča varno komunikacijo preko različnih porekel čez iframe (razdeli okno brskalnika v segmente), zavihkov in oken. Metodo postMessage pa definira kot standardni način za pošiljanje sporočil. Pošiljanje sporočila z metodo postMessage je zelo enostavno. [39]



Slika 2.1 : PostMessage komunikacija med iframe-om in glavno stranjo HTML; povzeto [39].

Pri komunikaciji med okvirjem iframe in glavno stranjo HTML, (Slika 2.1 : PostMessage komunikacija med iframe-om in glavno stranjo HTML; povzeto [39].) je pripomoček za klepet znotraj okvirja iframe, tako da nima direktnega dostopa do matičnega okna. Ko pripomoček za klepet dobi sporočilo, lahko uporabi metodo postMessage, da pošlje sporočilo na glavno stran, s tem pa opozori uporabnika pripomočka za klepet, da je prispelo novo sporočilo. Podobno lahko stran pošilja sporočila o stanju uporabnika na pripomoček za klepet. Stran in pripomoček lahko z dodajanjem ustreznih origin elementov na listo dovoljenih izvorov, med seboj poslušata sporočila. [39]

Pred uvedbo metode postMessage, je bilo komuniciranje med okvirji iframe včasih doseženo z direktnimi skriptami. Skripta ki se izvaja na eni strani, bi poizkušala manipulirati drug dokument. To ni dovoljeno zaradi varnostnih omejitev. Namesto neposrednega programskega dostopa, metoda postMesssage omogoča asinhrono pošiljanje sporočil, med JavaScript konteksti. Brez navzkrižnega komuniciranje metode postMessage med izvori, bi prišlo do varnostnih napak (
Slika 2.2), prisiljenih s strani brskalnikov, ki želijo preprečiti cross-site skriptne napade. [39]



Slika 2.2: Cross-site scripting napaka v zgodnji verziji Mozille; povzeto po [39].

Nezaželena je uporaba * nadomestne ključne besede v argumentu targetOrigin, saj ni zagotovila, da bo sporočilo dostavljeno le prejemniku, kateremu je bilo sporočilo namenjeno. [23]

Če ne pričakujemo sporočila od drugih strani, ne smemo dodajati poslušalcev za sporočilne dogodke.

Če pa pričakujemo sporočilo, moramo preveriti identiteto pošiljatelja preko uporabe lastnosti "izvor". Vsako okno lahko pošlje sporočilo drugemu oknu, pri čemer nimamo nobenega jamstva, da pošiljatelj ne bo poslal zlonamernih sporočil.

Priporočeno je, da vedno preverimo sintakso prejetega sporočila, drugače lahko pride do varnostne luknje, ki bi na spletni strani, ki ji zaupamo, odprla cross-site skriptno luknjo. [87]

Za zmanjšanje tveganj, za razkritje zaupnih podatkov in razširjeno napadalno površino validacija podatkov v UA na strani strežnika ne zadošča. Za varno uporabo programskega aplikacijskega vmesnika Cross Document Messaging, moramo upoštevati naslednje:

- Cilj v metodi postMessage() mora biti jasno opredeljen in ne označen kot *. Tako se izognemo pošiljanju občutljivih podatkov napačnemu okvirju, [60]
- Prejeta sporočila bi morala vedno biti potrjena in ne uporabljena neposredno, kot lastnost innerHTML, ali poslana znotraj funkcije eval() v JavaScript, [60]
- Sprejemni okvir bi prav tako moral preveriti domeno pošiljatelja. [60]
- Avtorji morajo vedno preveriti atribut "izvor", da se zagotovi, da so sporočila sprejeta le iz domen, od katerih pričakujemo sporočila. V nasprotnem primeru, bi lahko bile napake za ravnanje s sporočili v kodi avtorja, izkoriščene s strani zlonamernih mest. [24]

2.3.7 WebSocket API

Programski aplikacijski vmesnik WebSocket, je bil sprva definiran kot "TCPConnection" s strani Iana Hicksona, v oddelku za komunikacijo specifikacije HTML5. Specifikacije se je razvila in spremenila v WebSocket, ki je sedaj samostojna specifikacija, kot so geo lokacija, spletni delavci in tako naprej. [37]

Programski aplikacijski vmesnik zagotavlja izboljšavo starih, zvitih vdorov, ki so uporabljeni za simuliranje hkratne komunikacije v brskalniku. Upoštevati moramo, da ws:// in wss:// predponi prikazujeta WebSocket in varno WebSocket komunikacijo ločeno. [37]
Konstruktor WebSocket(url,protokoli) ima lahko enega ali dva argumenta. Prvi argument predstavlja naslov URL, na katerega se želimo povezati, drugi pa določa protokol, ki je v obliki niza ali polja nizov. Če je niz, je enakovreden polju nizov, ki vsebuje samo ta niz. Ko pa je izpuščen, je enak praznemu polju. Vsak niz v polju je ime pod-protokola. Povezava bo vzpostavljena samo, če strežnik javi, da je izbral enega izmed teh pod protokolov. Imena teh morajo biti nizi, ki se ujemajo z zahtevami elementov. Ti pa sestavljajo vrednosti Sec-WebSocket-Protocol polj, kot je opredeljeno s strani specifikacije WebSocket protokola. [85]

WebSockets je prvotno bil plan specifikacije HTML5, ampak je bil premaknjen v ločen dokument standardov, da bi ohranili osredotočeno specifikacijo. Predložen je bil projektni skupini za internetno tehnologijo (IETF), s strani WHATWG (delovna skupina za aplikacijsko tehnologijo). Avtorji in podjetja, ki so vključeni v standardizacijo, se še vedno sklicujejo na prvotni nabor funkcij kot HTML5.[1]

Če nastavimo vrednost znotraj modela spletne varnosti (vsebina-varnost-načela) na connect-srct 'self , preprečimo zahteve spletnih vtičnikov, iz katere koli lokacije, razen trenutnega strežnika. Prav tako preprečuje zahteve cross-origin resource sharing in EventSource. Če bi radi dodali CORS podporo za našo spletno stran, moramo prav tako avtomatsko odobriti zahteve spletnih vtičnikov, kar pomeni, da moramo biti tukaj zelo previdni. [78]

Varnostne težave, ki se tičejo aplikacijskega programskega vmesnika spletnih vtičnikov, so precej podobne tistim pri standardu CORS. Gre za enak temeljni problem - možno je vzpostaviti povezave spletnih vtičnikov preko domen, brez privoljenja uporabnika, zahteve pa so poslane brez, da bi uporabnik to opazil. Za napadalca je dovolj, da izvede nekaj kode JavaScript v uporabniškem agentu žrtve, kar pripelje do tega, da vzpostavimo povezavo spletnih vtičnikov na poljuben cilj. To povezavo lahko zlorabi napadalec, za izmenjavo podatkov iz in do uporabniškega agenta. Varnostna zahteva "varno predpomnjenje", je ogrožena preko aplikacijskega programskega vmesnika Web Socket. Ker vsi spletni namestniki ne razumejo protokola Web Socket programskega aplikacijskega vmesnika, lahko napadalec doseže, da spletni namestnik pred pomni prirejene podatke.To pa je mogoče zlorabiti tako, da so vse varnostne zahteve prekršene in sicer tako, da napadalec, vtihotapi zlonamerno kodo JavaScript, v uporabniškega agenta žrtve. Podobno kot pri CORS in spletnemu sporočanju, ostaja težava pri validaciji podatkov iz tujih izvorov. [60]

Priporočena je uporaba varnega wss:// protokola. Podobno kot HTTPS, je protokol WSS šifriran, kar prepreči napade "man-in-the-middle". Z uporabo tega protokola onemogočim veliko število napadov.[7]

Vsem podatkov iz neznanih virom, ne smemo zaupati. Vsak vnos je treba pregledati, preden gre v kontekst izvajanja. Prav tako moramo izvajati enak sum pri podatkih, ki prihajajo s

strani strežnika. Zaželeno je procesiranje sporočil, prejetih na strani odjemalca. Ne smemo jih poizkušati dodeliti neposredni specifikaciji DOM, niti jih pregledati kot kodo, če je odziv v formatu JSON. Vedno moramo uporabiti metodo JSON.parse(), za varno razčlenitev podatkov.[51]

2.3.8 Web Storage API

Za delovanje uporablja dva mehanizma [42]:

- sessionStorage, ki vzdržuje ločen prostor za vsako poreklo, ki je na voljo za čas trajanje seje (tako dolgo, kot je odprt brskalnik, vključno s ponovnim nalaganjem strani in obnavljanjem),
- localStorage počne isto stvar, vendar ostane tudi, ko se brskalnik zapre in ponovne zažene.

Ti mehanizmi so na voljo preko lastnosti Window.sessionStorage in Window.localStorage, v brskalnikih, ki podpirajo objekt Window. Le ta implementira objekta WindowLocalStorage in WindowSessionStorage. Na njih slonita localStorage in sessionStorage, klic ene izmed teh lastnosti, bo ustvaril instanco objekta Storage, preko katere lahko nastavimo, zbrišemo in pridobimo podatke o predmetih . Za sessionStorage in localStorage je ustvarjen drugačen Storage objekt glede na poreklo. [77]

Z uporabo tega enostavnega vmesnika, lahko razvijalci shranijo podatke, v zlahka dostopne objekte JavaScript, ki obstajajo po nalaganju strani. Z uporabo sessionStorage ali localStorage, lahko razvijalci določijo, da vrednosti po nalaganju strani, ostanejo v enem oknu ali zavihku in čez vnovičen zagon brskalnika. Shranjeni podatki se ne prenašajo preko spleta in so enostavno dostopni pri povratnih obiskih strani. Primeren je za shranjevanje dokumentov in podatkovnih datotek, ki imajo veliko večjo omejitev velikosti, kot piškotek.[42]

Podatki so dodani v shrambo z uporabo metode setItem(). Ta metoda vzame argumenta ključ in vrednost. Če ključ še ne obstaja v shrambi, se doda par ključ/vrednost v shrambo, če pa ključ že obstaja, se vrednost spremeni. Ključ mora biti vedno nit, medtem ko je vrednost lahko različnih tipov.[30]

Uporaba lokalne shrambe prinaša številne koristi, vendar odpira vrata vsem zgoraj opisanim grožnjam. Obstaja več točk, kjer bi lahko šlo narobe, prav tako pa morajo razvijalci skrbno implementirati dostop do lokalnih atributov za shranjevanje. Za varno uporabo lokalne shrambe v spletni aplikaciji, moramo upoštevati naslednje [60]:

- Za delo s sejo uporabimo piškotke, namesto lokalne shrambe, saj lahko piškotke bolj zavarujemo,
- Ne shranjujemo občutljivih podatkov v lokalno shrambo, te moramo shraniti na spletnih strežnikih in morajo biti primerno zaščiteni,
- Različne spletne aplikacije, ki tečejo na isti domeni in so ločene samo preko poti, ne smejo uporabljati lokalne shrambe, če je potrebno podatke ločiti.

Vendar pa še vedno obstajata grožnji sledenja uporabnikom in obstojni vektorski napadi, ki se jim ni mogoče izogniti.[60]

Zaželeno je, da preverjamo, ali so podatki shranjeni lokalno, vrnjeni nazaj uporabniku. Če je tako, moramo preveriti, ali je izhod pravilno kodiran, če ni poizkusimo vstaviti skripte v lokalno bazo. Ustvariti moramo scenarij, kjer je na primer napad "cross-site request forgery", uporabljen za avtomatsko vstavljanje zlonamerne kode, preko zlonamernih spletnih strani. [73]

Na spletu obstajata dve mesti, kjer lahko shranjujemo informacije, na spletnem strežniku ali spletnem odjemalcu (računalnik obiskovalca). Določeni tipi podatkov spadajo enemu, ostali pa drugemu. [45]

Na spletnem strežniku je prostor za shranjevanje občutljivih informacij in podatkov, ki jih ne želimo deliti z ostalimi. Če vnesemo košarico v spletni knjigarni, so vaši potencialni nakupi shranjeni na spletnem strežniku. Edini podatki, ki jih vaš računalnik ima, so sledilni podatki, ki spletni strani povedo, kdo smo. Tako stran ve, kateri nakupovalni voziček je naš. Tudi s standardom HTML5, ni razloga za spremembo teh nastavitev, saj je varno in učinkovito. [45]

Za shranjevanje lahko prav tako uporabimo programski aplikacijski vmesnik IndexedDB, ki omogoča shranjevanje velike količine strukturiranih podatkov, na strani odjemalca. Vmesnik uporablja indekse, da omogoča visoko kakovostno iskanje teh podatkov. Medtem ko je shramba DOM uporabna za shranjevanje manjše količine podatkov, je manj učinkovita pri shranjevanju velike količine podatkov, tukaj pa pride na vrsto vmesnik IndexedDB. [31]

Ko v uporabi, je vsebina baze WebDatabase na strani odjemalca občutljiva na napad SQL injection. En sam Cross-site scripting napad je lahko uporabljen za nalaganje zlonamernih podatkov v spletno podatkovno bazo, kot tudi pri lokalni shrambi. [56]

Nikoli ne smemo shranjevati občutljivih podatkov preko spletne shrambe, saj ne gre za varno shrambo. Ni varnejša od piškotkov, saj se ne prenaša preko žice, ni šifrirana in ne gre za prostor, kjer bi shranjevali sejo ali ostali varnostne podatke. [36]

Zaradi možnosti DNS napadov, ni mogoče zagotoviti, da je gostitelj dejansko v domeni, znotraj katere trdi, da se nahaja. Da bi to preprečili, uporabimo protokol transport layer security (TLS). Strani ki uporabljajo protokol TLS so lahko prepričane, da lahko do njihovih shramb dostopajo le strani s certifikati, tiste ki uporabljajo protokol TLS, ter programska oprema, ki deluje na strani uporabnika. [25]

2.3.9 Cross – Origin Resource Sharing

Potrdimo naslove URL, posredovane na metodo XMLHttpRequest.open. Trenutni brskalniki dovolijo, da so ti naslovi URL uporabljeni preko različnih domen, kar lahko pripelje do vstavljanje kode, s strani oddaljenega napadalca.

Posebno skrb moramo posvetiti absolutnim naslovom. Prepričati se moramo, da naslovi URL, ki odgovarjajo z glavo "Access-Control-Allow-Origin:*", ne vsebujejo občutljive vsebine ali informacije, ki bi lahko bila v pomoč nadaljnjim napadom. Glavo "Access-Control-Allow-Origin" uporabimo le na izbranih naslovih URL, do katerih ne rabimo dostopati preko domen. Glave ne smemo uporabljati za celotno domeno. Dovoliti moramo samo izbrane, zaupanja vredne domene. Bolje je, da naredimo seznam dovoljenih domen, kot pa da naredimo črno listo domen, ali pa da bi dovolili vsako domeno (nezaželena je uporaba * in slepo vračanje vsebine Origin glave brez preverjanja).[56]

Preprosta zahteva ustvarjena zunaj te specifikacije (uporaba metode GET ali POST za oddajo obrazca), običajno vključuje uporabniško avtorizacijo, zato morajo biti sredstva v skladu s to specifikacijo, vedno pripravljena in morajo pričakovati preproste zahteve za delitev virov preko izvorov. [75]

Omogočeno je torej dostopanje do internih spletnih strani, pregled/skeniranje internega omrežja, oddaljen napad na spletni strežnik, lažja zloraba, oziroma ponaredba zahteve cross-site. [59]

Implementacija za ublažitev vseh groženj preko strani strežnika ni možna. Prvi dve zmanjševanji groženj na sledečem listu, pomagata samo pri obhodu nadzora dostopa, zadnja točka pa pomaga pri zaznavanju napadov DDoS [60]:

- Omejimo dovoljene domene,
- Ne vključimo nadzora dostopa v glavo izvora, saj jo lahko napadalec spreminja,
- Za zmanjševanje napadov DDoS, mora požarni zid spletnih aplikacij, blokirati zahteve CORS, če prispejo pogosto krat.

Potrebno je varno preverjanje prisotnosti, tako za branje CORS dostopnih podatkov, kot za spreminjanje le teh. Razen če so podatki relativno javni, v tem primeru bi uporabnikom omogočali le pisanje, posodabljanje in izbris. [20]

2.3.10 Web workers API

Aplikacijski programski vmesnik, omogoča asinhrono in samostojno izvršitev datotek JavaScript. Spletni delavec je v bistvu nit, ki izvaja kodo JavaScript. Z delavci lahko tako dosežemo večnitnost v spletnih aplikacijah.[35]

Delavska nit lahko opravlja naloge brez poseganja v uporabniški vmesnik. Ko enkrat ustvarimo spletnega delavca, lahko ta pošlje sporočila JavaScript kodi, ki ga je ustvarila s pošiljanjem sporočil nadzorniku dogodkov, ki je definiran s pomočjo te kode. [74]

So dovoljeni za uporabo objektov XMLHttprequest, za izvajanje zahtev znotraj domen in preko različnih izvorov. Medtem ko spletni delavci nimajo dostopa do strani DOM, ki jo kličemo, lahko zlonamerni spletni delavci prekomerno uporabijo procesor za izračun, kar pripelje do zavrnitve storitve. Potrebno je poskrbeti, da koda znotraj spletnih delavcev ni zlonamerna. Ne smemo dovoliti ustvarjanje skript spletnih delavcev iz pridobljenih uporabniških vnosov. [56]

Ker se lahko nit izvaja neodvisno od uporabniškega vmesnika, to omogoča dolgotrajne skripte. Le te pa niso prekinjene s strani drugih skript in se odzivajo na klike in druge uporabniške interakcije. Prav tako pa omogočajo izvajanje dolgotrajnih nalog, pri čemer ohranja strani odzivne.[86]

Vsako sporočilo, ki ga izmenjamo s spletnimi delavci, je potrebno pregledati. Izmenjava delčkov kode JavaScript za vrednotenje (metoda eval()) ni zaželena, saj bi lahko to pripeljalo do ranljivosti XSS na podlagi jezika DOM. [56]

Zaradi varnostnih omejitev brskalnika Chrome, se delavci ne bodo izvajali lokalno (na primer file://) v najnovejših različicah brskalnikov. Da bi zagnali aplikacijo iz file:// sheme, moramo brskalnik Chrome zagnati z "omogoči dostop do datotek". To pa ni priporočljivo za naš primarni brskalnik, saj je to namenjeno bolj za namene preizkušanja in ne rednega brskanja. [3]

Spletni delavci neposredno ne uvajajo nobenih novih ranljivosti, ampak naredijo izkoriščanje ranljivosti lažje. Omogočajo lažjo vzpostavitev in uporabo spletnih vtičnikov, prav tako pa je manj verjetno, da se to odkrije s strani uporabnika, saj se celoten proces lahko izvede v ozadju. [60]

Strežniki se ne smejo zanašati na potrditev, na strani odjemalca. Validacijo na strani odjemalca, lahko nenamerno obidejo zlonamerni uporabniki in uporabniki starejših uporabniških agentov ali avtomatiziranih orodij, ki ne implementirajo teh orodij. [82]

3 Varnost HTML5

Specifikacija HTML5 uvaja številne tehnološke spremembe v standardu HTML. Pri kreaciji specifikacije, so bile varnostne specifikacije, del standarda že od začetka. Vsak del specifikacije ima svoj razdelek, ki se ukvarja z varnostjo. Ta podpoglavja zajemajo točke, ki morajo biti dobro premišljene, pri izvajanju ustreznih delov. Ranljivosti do katerih lahko pride zaradi te funkcije in kako varno jo implementirati s strani proizvajalcev brskalnika, je opisano. Avtorji specifikacije HTML5, so identificirali ranljivost uhajanje informacij pri <canvas> elementu, v primeru, če lahko skripte dostopajo do informacij, preko drugačnih izvorov. Spodaj bomo preverili vse ranljivosti, na katere lahko naletimo v specifikaciji HTML5 in kako se jim lahko izognemo.[60]

Varnost je pomemben del naših spletnih aplikacij. Te po definiciji uporabnikov, omogočajo dostop do osrednjega vira – spletnega strežnika – in preko njega do drugih, kot so strežniki podatkovnih baz. Z razumevanje in implementiranjem ustreznih varnostnih ukrepov, varujemo svoje vire, kot tudi zagotovimo varno okolje, v katerem uporabniki nemoteno izvajajo delo z našo aplikacijo. [34]

Specifikacija HTML5 prinaša veliko dobrih stvari, ampak je zaenkrat gnezdo ranljivosti. Specifikacija HTML4 je bila statična in je zato imela le nekaj novih ranljivosti, medtem pa je specifikacija HTML5 dinamična, kar sili ponudnike k napredku, to pa v zasnovi prinaša negotovosti. [22]

3.1 Zasebnost in privatnost podatkov

Varstvo podatkov je pogosto definiran kot zakon, ki je namenjen varovanju naših osebnih podatkov, ki se zbirajo, obdelujejo in hranijo. Pomembno je, da smo pooblaščeni za nadzor nad našimi informacijami, ki jih ščitimo pred zlorabami, prav tako pa je nujno, da zakoni za varstvo podatkov, omejujejo in oblikujejo dejavnosti, ki jih izvajajo podjetja in vlade. [32]

Januarja 2012 je Evropska komisija predlagala obsežno reformo pravil, o varstvu podatkov znotraj EU. Maja 2016 so bila besedila uredbe in direktive objavljena v uradnem listu EU, v vseh uradnih jezikih. Medtem ko je uredba začela veljati 24. maja 2016, se je začela uporabljati 25. maja 2016. Direktiva je bila prvič uporabljena 5. maja 2016. Članice EU so jo morale prenesti v svojo nacionalno zakonodajo do 6. maja 2016. Cilj tega novega sklopa pravil je, da damo državljanom nadzor, nad njihovimi osebnimi podatki, ter poenostavitev zakonodajnega okolja za podjetja. Reforma varstva podatkov je ključnega pomena za enotni digitalni trg, kateremu Komisija daje prednost. Reforma tako evropskim državljanom, kot podjetjem nudi, da v celoti izkoristijo digitalno gospodarstvo. [10]

Varnost uporabniških podatkov na naši strani dosežemo z varnimi komunikacijami, kjer uporabljamo protokole Secure Sockers Layer (SSL), Transport Layer Security (TLS) in Private Communications Technology (PCT). Ti protokoli so uporabljeni zato, da se ustvarijo varni kanali, za izmenjavo informacij preko spleta. [52]

Zaželeno pa je tudi, da uporabnikom nudimo nadzor nad piškotki. To pomeni, da lahko piškotkom preprečimo, da so shranjeni na našem računalniku, da smo obveščeni, če želimo uporabiti piškotke s strani in če želimo omogočiti piškotkom, da se shranjujejo na našem računalniku. Prav tako pa moramo uporabnikom nuditi možnost, da si urejajo svoj profil in nastavitve le tega. [52]

3.2 EU direktiva o varstvu podatkov (Direktiva 94/46/EC)

Namen nove uredbe za varstvo podatkov, je uskladiti sedanje zakonodaje o varstvu podatkov, po državah članice EU. Dejstvo da gre za ureditev in ne za direktivo pomeni, da se bo neposredno nanašalo na članice EU, brez potrebe za nacionalno implementacijo legalizacije. [17]

Osebni podatki se lahko zakonito obdelujejo če [58]:

- Obdelava temelji na privolitvi subjekta, nad katerim izvajamo obdelavo,
- Podatkovni interesi uporabnika zahtevajo obdelavo njihovih podatkov,
- So zakoniti interesi drugih razlogi za obdelovanje, ampak samo, če ne prestopijo interesov za zaščito temeljnih pravic uporabnika.

3.3 Varnost podatkov za različne organizacije

W3C

Privatnost ima veliko različnih vidikov v W3C organizaciji [62]:

- Večina tehnologij W3C se ukvarja z osebnimi podatki in morajo s tem upoštevati zasebnost osebnih podatkov. Nekaj truda zato gre v pomoč drugim delovnim skupinam, kot je na primer geo lokacija, za boljše obravnavanje zasebnosti,
- Delovna skupina za varstvo sledenja je uvedla mehanizem "Ne sledi" pod visokim javnim nadzorom. Medtem ko je politično lažja "Specifikacija izrazno sledilnih nagjenj " zelo zrela, specifikacija za skladnost sledenja počasi dozoreva,
- Je tehnološko območje samo po sebi. Platforma za preference zasebnosti, (P3P) je temeljni korak in še vedno velja kot podlaga za mnoge sedanje tehnologije zasebnosti uporabnikov in uporabniški nadzor, z opredelitvijo mehanizmov za izražanje želja uporabnikov za spletno sledenje in za blokiranje ali omogočanje elementov za spletno sledenje,
- Zasebnost je območje intenzivnega raziskovanja, v zadnjih sedmih letih je organizacija W3C sodeloval pri raziskavah EU FP7 o zasebnosti. Projekt PrimeLife je imel proračun v višini 11 milijonov evrov. V tem projektu je organizacija W3C poizkušala napredovati na področju jezikov, politike in socialne mreže. Organizacija W3C je še vedno akter na področju raziskav zasebnosti.

Siemens

Vidiki varnosti znotraj podjetja Siemens so [65]:

- Varovanje zasebnosti osebnih podatkov je za podjetje Siemens pomembno. Zato družba Siemens upravlja svoje spletne strani v skladu z zakoni o varstvu zasebnosti in varnosti podatkov,
- Na spletnih straneh uporabljajo piškotke, katere lahko blokiramo, prav tako lahko blokiramo uporabo podatkov, ki jih piškotki zberejo,
- Dostopi do spletnih strani so namenjeni za varnostno poročilo varnostne analize in za obrambo pred kibernetskimi napadi. Z izjemo naslovov IP, se osebni podatki ne zbirajo ali uporabljajo. Naslovi IP so analizirani le v primeru spletnega napada. Prijavni podatki se redno izbrišejo,
- Zbirajo osebne podatke, kot so na primer imena, naslove, telefonske številke ali e-mail naslovi, v povezavi z njihovo spletno stranjo, samo ko smo te podatke prostovoljno ponudili (preko registracije, kontaktne poizvedbe, raziskave). Stran lahko podatke uporabi za obdelavo in uporabo na podlagi dovoljenja, s strani uporabnika.

Amazon spletme storitve

Nudijo storitve za več kot milijon aktivnih strank, vključno s podjetji, izobraževalnimi ustanovami in vladnimi agencijami. Njihove stranke jim zaupajo nekatere najbolj občutljive informacije.

Stran daje lastništvo in nadzor strankam nad njihovo vsebino, preko močnih orodij, ki strankam omogočajo, da ugotovijo, kje bo njihova vsebina shranjena in upravljajo dostop do storitev in sredstev strani za svoje uporabnike. Prav tako nudijo tehnične in fizične kontrole, ki naj bi preprečile nepooblaščene dostope do podatkov. [2]

Lastništvo in nadzor nad vsebino uporabnikov [2]:

- Dostop: stranke same upravljajo dostop do njihovih vsebin, storitev ter virov strani. Nudijo napreden dostop, šifriranje in funkcije prijave, ki pri tem učinkovito pomagajo. Ne dostopajo do osebnih podatkov strank za katere koli druge namene, kot tiste zakonsko zahtevane, za ohranjevanjem storitev strani in zagotavljanje le teh za končne uporabnike,
- Skladiščenje: stranke same izberejo regijo, v kateri bodo njihovi podatki shranjeni. Stran ne bo premikala ali replicirala podatke strank, razen če je legalno zahtevano za zagotavljanje storitev strani,
- Varnost: kupci izberejo kako se bo njihova vsebina zavarovala. Strankam nudijo močno šifriranje za vsebino v prometu in možnostjo, da upravljajo svoje šifrirne ključe,
- Razkritje vsebine stranke: stran ne razkriva vsebine uporabnikov, razen če pride do legalne zahteve, s strani vladnega ali upravnega organa. Podjetje Amazon prav tako obvesti stranke pred razkritjem informacij,
- Varnostno zavarovanje: razvili so program za zagotavljanje varnosti, s pomočjo globalne zasebnosti in najboljše prakse varstva podatkov, da bi pomagali strankam vzpostaviti in upravljati varno okolje.

3.4 Zahteve in zakoni glede piškotkov

Zakon o piškotkih je del zakonodaje zasebnosti, ki zahteva od spletne strani, da dobi soglasje od obiskovalcev, za shranjevanje ali pridobivanje vseh informacij na pametnem telefonu ali tablici. Zasnovan je bil za zaščito zasebnosti na spletu, tako da bi se potrošniki zavedali, kako se o njih zbirajo in uporabljajo informacije na spletu in se jim da na izbiro, če to dovolijo ali ne. Začel je kot EU direktiva, ki je bila sprejeta s strani vseh držav EU, maja 2011. Direktiva je dala posameznikom pravico, da zavrnejo uporabo piškotkov, ki zmanjšajo njihovo zasebnost na spletu. Vsaka država je nato posodobila svoje zakone, da bi se uskladila. V Veliki Britaniji je to pomenilo posodobitev predpisov za varovanje zasebnosti in elektronsko komunikacijo. [69]

Skoraj vse spletne strani uporabljajo piškotke za shranjevanje informacij v spletne brskalnike ljudi, nekatere jih vsebujejo stotine. Obstajajo pa druge tehnologije, kot je dodatek Flash in funkcionalnost HTML5 local storage, ki počnejo podobne stvari in so prav tako zajete v zakonodaji, ampak ker so piškotki najbolj uporabljena tehnologija, je zakonodaja dobila ime "zakon o piškotkih". Za tiste, ki imajo spletne strani to pomeni, da se morajo prepričati, da je spletna stran v skladu z zakonom, kar veliko krat pomeni veliko število sprememb. Če ne izpolnimo zahtev, tvegamo ukrepe s strani regulatorjev, kar v Veliki Britaniji predstavlja urad informacijskih komisarjev. V izjemnih primerih, lahko to pomeni globo. [69]

Večina spletnih strani v Evropi, se je moralo spremeniti ali prekršiti zakon. Preko 92% spletnih strani uporablja piškotke. Te so morali bodisi prenehati z uporabo piškotkov ali pa morali spraševati za dovoljene. Za vprašanje o dovoljenju, mora spletna stran prekiniti svoje obiskovalce. Nihče ne želi tega dodati na spletne strani, prav tako pa se večina obiskovalcev, tega ne veseli. Spletne strani bi lahko prenehale z uporabo piškotkov, kar bi pa pripeljalo do tega, da bi izgubile določene funkcionalnosti. [66]

Ker so piškotki samo besedilne datoteke, se lahko uporabljajo za shranjevanje vsega, kar si zaželi avtor spletne strani, kar pripelje do velikih skrbi glede zasebnosti. Stvari kot je "všeček" gumb na Facebooku, so lahko uporabljene s strani Facebooka, za slednje uporabnikom po ostalih spletnih straneh. [70]

3.5 EU legalizacija piškotkov

EU zakon o piškotkih je del zakonodaje o varovanju zasebnosti, ki je bil sprejet s strani vseh držav EU leta 2011. Velika Britanija je imela eno leto, da sprejme EU direktivo, potem ko posodobila svoje regulacije, glede zasebnosti in elektronsko komunikacijo, kar je prineslo direktivo EU v zakonodajo Združenega Kraljestva. [27]

Komisarski urad za informacije, (ICO) je odgovoren za to, da organizacije delujejo v skladu z zakonom o piškotkih. Urad je do zdaj izdal dve smernici, katerih namen je, da spominjajo tiste katerih se tiče, da bo zakon ostal.

Če podjetje po 26.5.2016 ni bilo v skladu ali si prizadevalo za skladnost z direktivo, jim preti morebitna globa, do višine pol milijona funtov. [27]

Spletne strani morajo upoštevati smernice Komisije o zasebnosti in varnosti podatkov in obvestiti uporabnike, da piškotki niso uporabljeni za zbiranje podatkov, po nepotrebnem. Direktiva o e-zasebnosti zahteva predhodno privolitev, za shranjevanje ali dostop do informacij, shranjenih na uporabnikovi terminalni opremi. Z drugimi besedami, uporabnika je potrebno vprašati, če se strinja z uporabo piškotkov in podobnih tehnologij, preden jih začne stran uporabljati. Da bi soglasje bilo veljavno, mora biti informativno, specifično, prostovoljno in mora biti pravi pokazatelj posameznikovih želj. [9]

Vendar pa nekateri piškotki ne vsebujejo teh zahtev. Soglasje ni potrebno, če [9]:

- Je piškotek namenjen za namen opravljanja prenosa sporočila,
- Nujno potreben v primeru zahteve uporabnike, za opravljanje navedene storitve ponudnika storitev informacijske družbe.

Pri uporabi piškotkov moramo oceniti, če je potrebna privolitev za uporabo piškotkov. Obdobje izteka piškotka, ne sme presegati enega leta. Vse seje tretjih oseb in trajni piškotki zahtevajo soglasje. [9]

Uporabnike obvestimo o uporabi piškotkov v preprostem jeziku, na strani namenjeni za obvestila o piškotkih, kjer se povezava na stran nahaja v orodni vrstici standardne predloge. Ta stran mora pojasniti, zakaj se piškotki uporabljajo, če so piškotki nujni za spletno stran, ali za dano funkcionalnost da deluje, ali če je njihov namen izboljšati učinkovitost spletne strani, ki jih spletne strani uporabljajo. Definirati mora, kdo nadzoruje dostope do informacij, ki se tičejo piškotkov in da piškotki ne bodo uporabljeni za katero koli drug namen, kot za tega, kako lahko uporabnik umakne soglasje. [9]

3.6 Raziskave o zasebnosti podatkov

Tabela 3.1 : Prikaz raziskav in njihovih ugotovitve; povzeto po [16], [19], [21], [67].

Raziskovalec	***Področje raziskave***	***Ugotovitve***
Greg Durham	Zasebnost na internetu, uredba internetne zasebnosti	Ko zakon in tehnologija konvergirata države in zvezno vlado, se bosta vedno spoprijemala z omejitvami na internetni tehnologiji. Pogosto zamenjamo udobnost za privatnost, pri čemer moramo pogosto krat pogledati nazaj in se spomniti svojih predstav o tem, kaj nas naredi državljane in kaj državljanska pravica pomeni za nas. Naši institucionalni in pravni okvirji morajo dohiteti nenehni tok tehnologije.
Christian Fuchs	Varnost in zasebnost v Evropi	Leta 2012 je EU uvedla uredbo o varstvu podatkov, ki določa pravico biti pozabljen, pravico uporabnika, da ni profiliran in pravico do prenosljivosti podatkov, legalizacija de facto zakonodaje izključuje zaščito podatkov s strani vladnih agencij kazenskega pregona, prav tako niso z njihove strani vključene varnostne in zasebne procedure, osnutek o regulaciji varstva podatkov določa globe v višini do 2% prometa družbe, če krši legalizacijo varovanja podatkov, poraz ACTA v evropskem parlamentu z večino glasov proti sprejetju sporazuma je pokazal, da obstajajo problemi za obdelovanje podatkov (pravice intelektualne

		lastnine, tam kjer je velika skrb v Evropi, kjer nadzor omejuje zasebnost in državljanske pravice), raziskava je prav tako pokazala, da je razmerje med zasebnostjo in varnostjo v Evropi v prvi vrsti zelo sporna tema, ter da politični dokumenti kot je Stockholmski program in strategija notranje varnosti EU definirajo model, kjer prednjači razvoj in uporaba tehnologij na strošek pravic do zasebnosti.
Ori Heffetz, katrina Ligett	Zasebnost in podatki	Opiranje na intuicijo pri poskusu zaščite zasebnosti veliko krat ne bo dovolj. , poleg tega lahko pride do ogrožanja zasebnosti tudi, ko neobdelani podatki niso nikoli javno objavljeni, raziskovalci bi morali bolj pogosto obveščati uporabnike, da jih lahko odločen napadalec identificira ali celo izve stvari o njih iz empiričnih rezultatov raziskovalnega papirja
H. Jeff Smith, Tamara Dinev, Heng Xu	Zasebnost informacij	Malo pozornosti je bilo namenjeno dejavnikom, ki služijo za varovanje zasebnosti, raziskovalci pogosto predpostavljajo, da navedene namene lahko enačijo z dejanskim vedenjem, kar je posebej šibka predpostavka glede na paradoks zasebnosti, večina empiričnih opisnih študij, ki so naslovile povezave med predhodniki in skrbi glede zasebnosti so se osredotočile na individualne zaznave

Kot so pokazale nekatere dosedanje raziskave (Tabela 3.1 : Prikaz raziskav in njihovih ugotovitve; povzeto po [16], [19], [21], [67].), še imamo dolgo pot do tega, da bodo ljudje pravilno dojemali zasebnost in varnost njihovih podatkov, ter da jih je potrebno o tem ozaveščati in jih informirati do katerih podatkov strani/podjetja lahko dostopajo in do katerih ne.

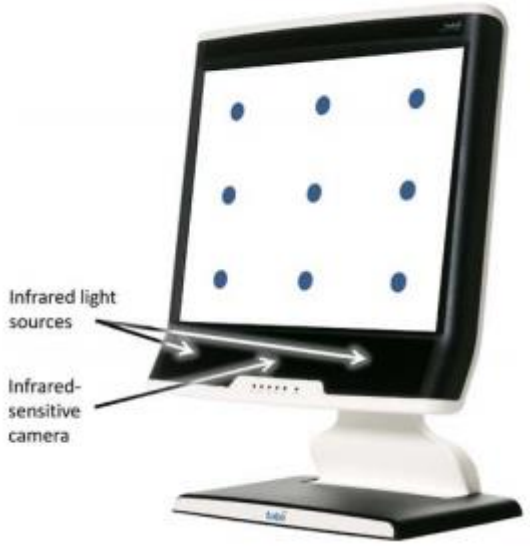
3.7 Tehnologije za sledenje očem

V najbolj preprosti razlagi: eye tracking je merjenje aktivnosti oči. Kam gledamo? Kaj ignoriramo? Kdaj mežikamo? Kako zenice reagirajo na različne stimulacije? Koncept je osnoven, ampak proces in interpretacija sta lahko precej zapletena. [72]

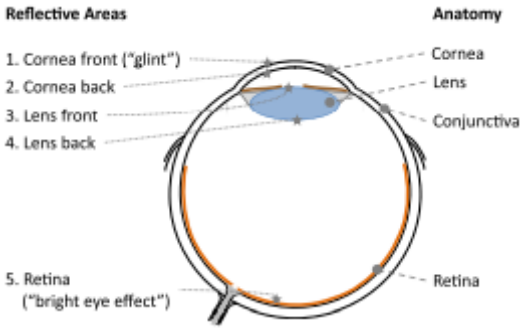
Hiter razvoj tehnologij za sledenje očem, je bil skozi zadnja leta spremljan. Današnji eye trackerji lahko natančno določijo fokusno točko našega očesa, medtem ko so relativno nemoteči v aplikacijah. V zadnjih 100 letih, so raziskovalci zbrali veliko količino znanja o premikanju oči, zakaj in kako se pojavijo in kaj lahko pomenijo.[4]

Dandanes imamo tehnologijo in znanje za sledenje in analiziranje gibanja oči, kar je odlično izhodišče za prefinjene interaktivne aplikacije, ki temeljijo na pogledu. Naivni pristopi, kjer so podatki neposredno uporabljeni za interakcijo s sistemom, na primer stisk gumba na zaslonu z mežikom očesa, imajo na splošno velike težave. Ker so oči organ, ki so uporabljeni za dojemanje sveta in ne za manipulacijo tega, je težko in proti človeški naravi, da bi namerno nadzirali gibanje oči. [4]

Vendar pa je zelo obetaven pristop, prav opazovanje gibanja oči uporabnika, med njegovim vsakodnevnim delom pred računalnikom, sklepati uporabnikove namene glede na gibanje oči in zagotoviti pomoč, ko je le ta potrebna. Pogled lahko obravnavamo kot namestnik za uporabnikovo pozornost. Gibanje oči je znano po tesni povezavi s kognitivnimi procesi v možganih, tako da je lahko velik del teh procesov, mogoče opazovati s sledenjem očem. Z razlaganjem gibanja oči, lahko zaznamo bralne navade uporabnika, ki najverjetneje pomenijo kognitivne procese razumevanje, v zvezi s trenutno branim besedilom. [4]



Slika 3.1 : Moderna namizna eye tracker naprava Tobii T60; povzeto po [4].



Slika 3.2 : Shema očesa. Različne lokacije odsevov svetlobe, uporabljene za eye tracking ; povzeto po [4].

3.7.1 Način delovanja

Eye tracking podatki so zbrani z bodisi oddaljene ali na glavo montirane naprave eye tracker, (Slika 3.1 : Moderna namizna eye tracker naprava Tobii T60; povzeto po [4].) povezane na računalnik. Obstaja veliko različnih nevsiljivih naprav eye tracker, ki na splošno vključujejo dve skupni komponenti: svetlobni vir in kamero. Svetlobni vir (običajno infrardeča) je usmerjen proti očesu. Kamera sledi odboju svetlobnega vira, skupaj z vidnimi očesnimi deli,

kot je zenica (Slika 3.2). Ti podatki so uporabljeni za ekstrapolacijo vrtenja oči in nenazadnje smer pogleda.
Dodatne informacije, kot je pogostost mežikanja in spremembe v premeru zenic, so prav tako zaznane s strani naprave eye tracker. Združeni podatki so zapisani v datoteki, ki je združljiva s programsko opremo za analizo, kot je na primer EyeWorks. [72]

3.7.2 Možnost uporabe

Tehnologije za sledenje očem, lahko uporabimo na najrazličnejših področjih. Uporabimo jih lahko pri nevro znanosti in psihologiji (gibanje oči in slikanje možganov, branje, zaznavanje, vizualno iskanje, naravne naloge in gibanje oči v procesiranju informacij ostalih nalog), industrijskemu inženiringu in človeških dejavnikih (aviacija, vožnja, vizualna inšpekcija), oglaševanju in marketingu (testiranje izvodov, umestitev oglasov, izboljšave televizije, spletne strani, oblikovanje etiket izdelka) in računalniški znanosti (klasična očesna interakcija, kognitivno modeliranje, univerzalna dostopnost, posredna očesna interakcija, pozorni uporabniški vmesniki). [14]

4 Analiza zavedanja uporabnikov o zasebnosti/varnosti v HTML5 spletnih aplikacijah

Cilji ki jih pri raziskavi želimo doseči so, ali se uporabniki zavedajo konteksta zasebnosti, ali vedo kaj dosežejo z deljenjem informacij, kaj jih pripravi do tega, da delijo ali ne delijo informacije. Od uporabnikov bomo prav tako izvedeli, koliko časa namenijo zasebnosti in privatnosti znotraj aplikacij HTML5, koliko se jih zaveda, da določene funkcionalnosti obstajajo, jih preberejo, preverijo in uporabijo. Prav tako želimo uporabnike spodbuditi, da se zavedajo zasebnosti in privatnosti v aplikacijah HTML5. Želimo zagotoviti, da temeljito preberejo vsa varnostna sporočila in da vedo, kaj delijo ob kliku na gumb "deli". Prav tako želimo izvedeti, kakšno je njihovo mnenje o teh funkcionalnostih, koliko je takih, ki podatke delijo brez težav, koliko je takih, ki ne želijo deliti podatkov, ali pa jim je vseeno. Tako bomo lahko pridobili informacije, kaj jih moti, kaj jim je všeč, zakaj so se odločili prezreti/prebrati določeno varnostno obvestilo in kaj jih pripravi do tega, da izberejo določeno pot, ter zakaj informacij ne želijo deliti.

Na raziskavo smo se pripravili tako, da smo najprej pridobili aplikacije, oziroma spletne strani, ki uporabnika sprašujejo o deljenju osebnih informacij. Torej gre za strani, ki uporabljajo različne aplikacijske programske vmesnike, novega standarda HTML5, ki nudijo različne funkcionalnosti, za zaščito uporabnika pred uporabo zasebnih podatkov. Na podlagi tega, katere funkcionalnosti in aplikacijske programske vmesnike so te strani uporabljale, smo zapisali naloge, ki so jih morali uporabniki izvajati, med samim eksperimentom. Sledila je izbira testnih uporabnikov. Izbirali smo ljudi, ki so seznanjeni z internetom in spletnimi stranmi, ki uporabnika obveščajo o deljenju informacij ter poznajo osnove standarda HTML in njegovih funkcionalnosti.

Pred samim izvajanjem eksperimenta je potekal test, s katerim smo pridobili potrebno znanje, za delo z napravo Eye tracker Guide. Preverili smo, kako lahko snemamo uporabnikovo obnašanje in kako po končanem snemanju analiziramo in spremljamo obnašanje uporabnika, pri opravljanju zadanih nalog, ter kako lahko uporabljamo razne metode, ki jih programska oprema za analiziranje omogoča. Po opravljenem testu, smo bili pripravljeni na izvajanje eksperimenta.

Kakovost pridobljenih podatkov smo lahko potrdili s pridobitvijo merljivih, jasno določenih podatkov, na podlagi katerih lahko potrdimo ali ovržemo hipoteze, jih analiziramo, ter pridemo do jasnega zaključka eksperimenta.

Raziskava je potekala v učilnici Fakultete za elektrotehniko računalništvo in informatiko, kjer je bil na voljo računalnik, na katerem so se izvajale izbrane spletne strani, pripadajoče naloge, ki so jih na teh straneh uporabniki opravljali, ter naprava Eye tracker Guide, s katero smo pridobili podatke za analiziranje.

4.1 Pristop k reševanju problema in uporabljene metode za pridobitev podatkov

Pred samo izvedbo projekta, je bilo potrebno določiti, katere podatke želimo od uporabnikov pridobiti in na kakšen način. Ker smo želeli od uporabnikov izvedeti, koliko časa namenijo sami zasebnosti svojih podatkov in v kolikšni meri se zavedajo varnosti na spletu, smo se odločili za uporabo naprave Eye tracker Eye guide S pomočjo le te smo lahko pridobili informacije, kot so na primer, kam uporabnik usmerja pozornost, koliko pozornosti namenja za branje varnostnih obvestil,… Vse ostale informacije, ki pa jih nismo mogli pridobiti s samim analiziranjem in spremljanjem gibanja oči uporabnikov, smo pridobili z anketo. Nato je bilo potrebno določiti naloge, ki so jih uporabniki izvajali na samih aplikacijah. Za konec pa smo morali pridobiti samo še uporabnike, na katerih smo raziskavo izvedli.

4.2 Potek raziskave

Uporabnikom so bile dane naključne spletne aplikacije, ki temeljijo na specifikaciji HTML5, ki so relativno nove in uporabljajo raznolike funkcionalnosti, ki od uporabnika zahtevajo deljenje različnih podatkov. Eden po eden so na računalniku opravljali naloge. Preko naprave Eye Guide, smo spremljali obnašanje uporabnikov. Tako smo objektivno pridobili informacije, koliko časa in pozornosti so uporabniki namenili sami zasebnosti in privatnosti, ter koliko časa ostalim stvarem, na sami aplikaciji HTML5. Videli smo tudi, katere podatke so delili in katerih niso želeli deliti, pri uporabi razno raznih funkcionalnosti.

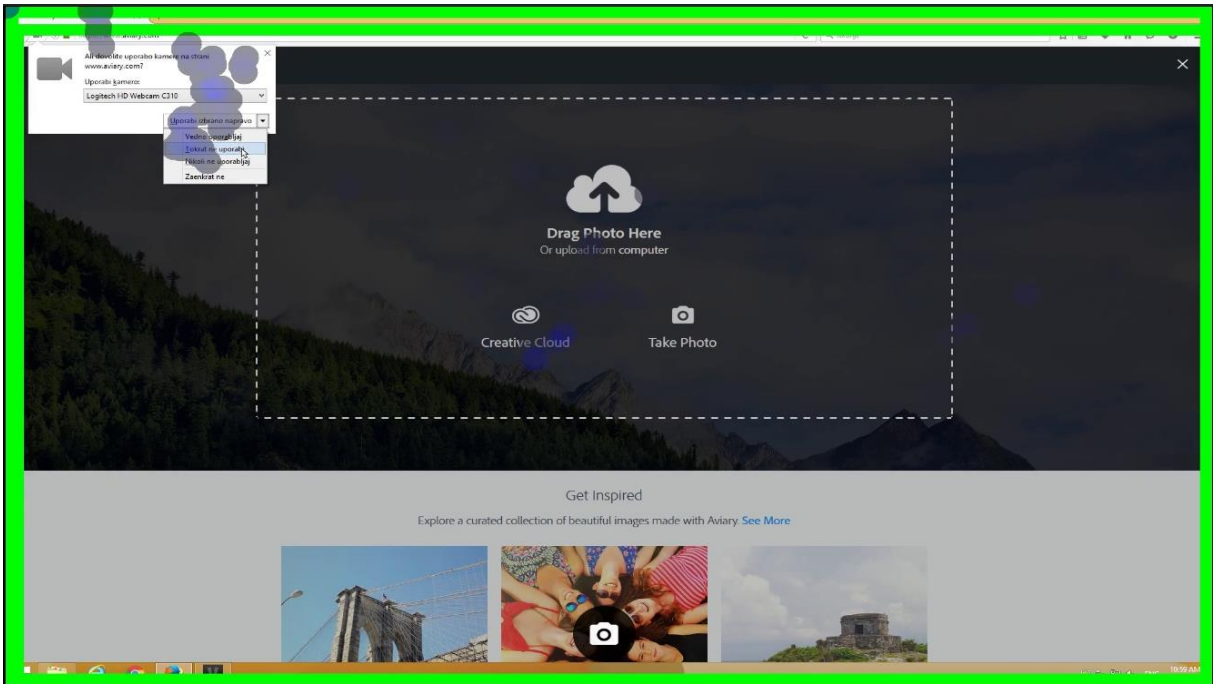
Po končanih nalogah so rešili anketo, s pomočjo katere smo dobili odgovore, kot je na primer, zakaj so se odločili za to pot in ne za kakšno drugo alternativo .

Analiza podatkov je potekala z uporabo različnih metod sledenja očem, najbolj nazorna in uporabljena je bila toplotna mapa, saj natančno pokaže, na kateri del zaslona se uporabniki najbolj osredotočijo.

Strani ki smo jih uporabili pri samem eksperimentu so:

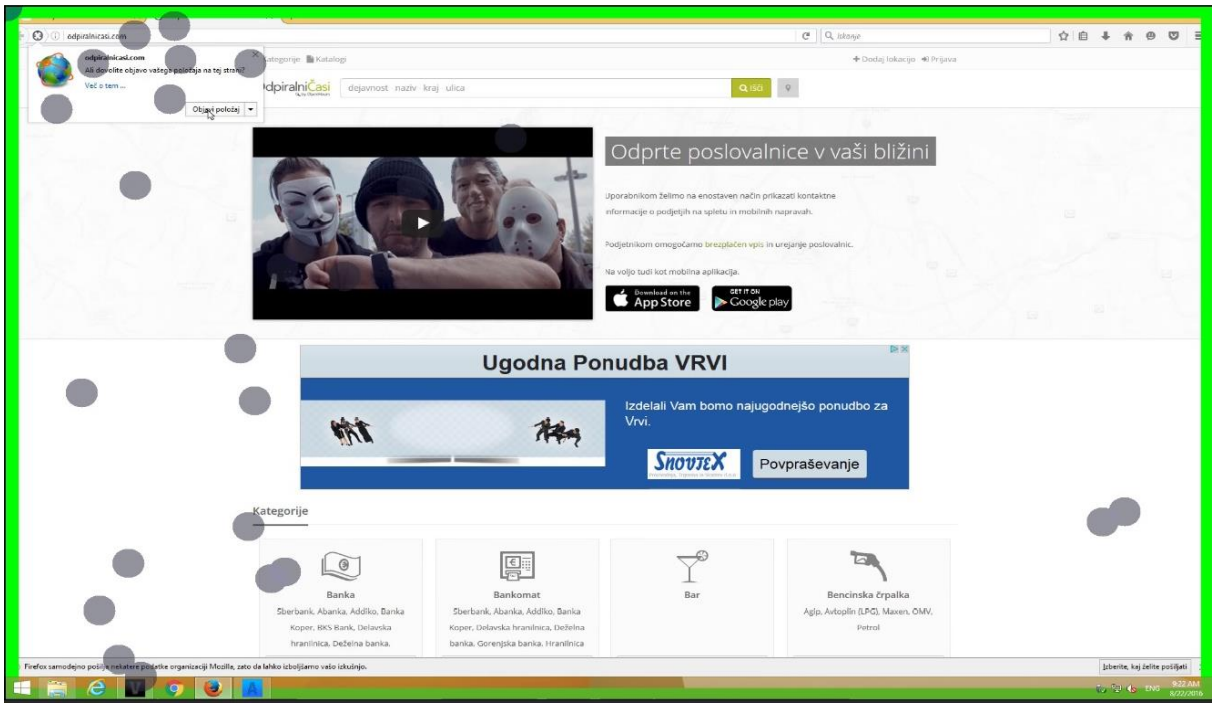
- www.aviary.com – tukaj so uporabniki morali zajeti sliko in jo nato urediti (Slika 4.1). Stran smo uporabili, ker je zahtevala dostop do spletne kamere, s tem smo želeli ugotoviti, ali ljudje dovolijo dostop do spletne kamere ali ne, ter koliko časa so namenili branju sporočila. Stran uporablja funkcionalnost HTML5, ki nam omogoča nadzor kamere in videa,
- www.odpiralnicasi.com – stran je od uporabnika zahtevala dostop do lokacije (Slika 4.2), zato da mu je lahko prikazala najbližji bar, glede na njegovo lokacijo. S tem smo uporabljali geo lokacijski aplikacijski programski vmesnik, kjer se zahteva dostop do lokacije uporabnika, ter smo opazovali, če se uporabniki strinjajo z deljenjem informacije o lokaciji,
- www.moodmet.com - stran je od uporabnika zahtevala dostop do lokacije (Slika 4.3), zato da mu je lahko prikazala počutje uporabnikov strani na svetu. S tem smo uporabljali geo lokacijski aplikacijski programski vmesnik, kjer se zahteva dostop do lokacije uporabnika, ter smo opazovali, če se uporabniki strinjajo z deljenjem informacije o lokaciji,
- www.pinterest.com – na tej strani so se uporabniki morali registrirati preko svojega lastnega Facebook ali Google računa (Slika 4.4). To smo izbrali, ker dobimo dodatno informacijo, koliko ljudi privoli v delitev svojih podatkov, kot so na primer pri vpisu s profilom Facebook: javni profil, seznam prijateljev, e-poštni naslov, rojstni datum, osebni opis, "všečki", koliko časa namenijo branju tega, kaj od njih stran zahteva, ter koliko je takšnih, ki so deljenje zavrnili.

Primer metode toplotnega polja in prva naloga (Slika 4.1), ki so jo morali uporabniki opraviti (obiščite stran aviary.com in zajemite sliko s kamero). Slika prikazuje trenutek, ko je stran od uporabnika zahtevala dostop do kamere, da bi lahko zajel sliko. Krogci na sliki prikazujejo uporabo metode "toplotna mapa", kjer se s krogci prikazujejo deli strani, na katere so uporabniki bili najbolj pozorni. Prav tako nam nudijo okvirno predstavo, kako dolgo so uporabniki gledali v določeno točko. Krogci so prikazani z modro barvo – okoli 2 sekundi, rumeno prikazani krogci – okoli 4 sekunde in rdeči krogci več kot 7 sekund).



Slika 4.1 : Uporaba heat-map spremljanja očesa pri prvi nalogi eksperimenta.

Druga naloga (Slika 4.2) je bila, da so morali uporabniki na strani odpiralnicasi.com poiskati najbližji bar, glede na trenutno lokacijo, kjer je stran potrebovala dostop do trenutne lokacije. Na sliki je zajet trenutek, ko stran od uporabnika zahteva dostop do njegov trenutne lokacije. Ponovno je bila uporabljena metoda "toplotna mapa", saj najbolj nazorno pokaže, kam in koliko časa so največ gledali uporabniki, ko se pojavilo obvestilo o lokaciji. Za prikaz informacij uporabljajo krogce. Krogci modre barve povedo, da je uporabnik v to točke gledal okoli 2 sekunde, rumeni da okoli 4 sekunde in rdeči da je gledal okoli 7 sekund.

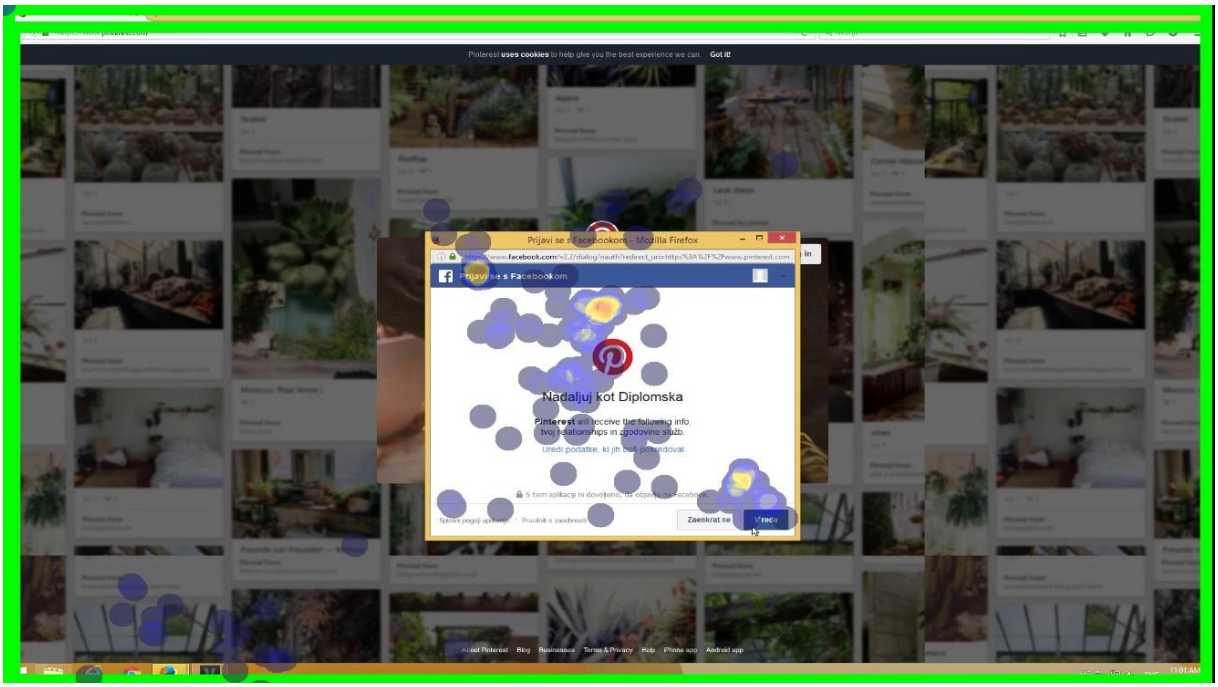


Slika 4.2 : Prikaz heat-map spremljanja očesa pri drugi nalogi eksperimenta.

Nato so uporabniki pri naslednji nalogi (Slika 4.3) obiskali spletno stran moodmet.com, kjer je stran prav tako potrebovala dostop do trenutne lokacije uporabnika. Na sliki je zajet trenutek, ko je stran od uporabnika zahtevala dostop do njegov trenutne lokacije. Za spremljanje aktivnosti oči uporabnika, smo uporabili metodo "toplotna mapa". Prikaz kam je uporabnik gledal v trenutku, ko se je pojavilo obvestilo, metoda nudi s pomočjo krogcev. Modri krogec pomeni, da je uporabnik v to točko gledal okoli 2 sekunde, rumena da okoli 4 sekunde, rdeči pa 7 sekund in dalje.

Slika 4.3 : Prikaz heat-map spremljanja očesa pri tretji nalogi eksperimenta.

Zadnja naloga (Slika 4.4) je bila, da so se uporabniki morali prijaviti na spletno stran pinterest.com, preko računa Facebook ali Google, kjer jim je ob vpisu izpisalo, katere podatke bodo delili, če se želijo vpisati. Na sliki je zajet trenutek, ko je stran od uporabnika zahtevala dostop do podatkov Facebook ali Google računa. Za analizo tega, kam je uporabnik gledal, smo uporabili metodo "toplotna mapa". Prikaz kam je uporabnik gledal v trenutku, ko se je pojavilo obvestilo, metoda nudi s pomočjo krogcev. Modri krogec pomeni, da je uporabnik v to točko gledal okoli 2 sekunde, rumena da okoli 4 sekunde, rdeči pa 7 sekund in dalje.



Slika 4.4 : Prikaz heat-map spremljanja očesa pri četri nalogi eksperimenta.

4.3 Uporabljene metode in orodja

Metode in orodja, ki so bila uporabljene znotraj raziskave so:

- Računalnik na katerem so se izvajale naloge na aplikacijah HTML5,
- Aplikacije HTML5, na katerih smo izvajali naloge,
- Naprava Eye tracker guide za merjenje in spremljanje aktivnosti oči,
- Spletna kamera za zajem slik,
- Za uspešno izvajanje je bilo potrebno naložiti potrebno programsko opremo, s katero bomo dosegli delovanje naprave Eye tracker.

4.4 Rezultati in ugotovitve

Pri diplomski nalogi smo podatke najprej pridobili s pomočjo eksperimenta. Podatki ki smo jih tako pridobili so sledeči:

Tabela 4.1 : Predstavitev pridobljenih podatkov pri eksperimentu (prvi del)

	Naloga 1		**Naloga 2**	
	Deljenje osebnih podatkov	**Čas namenjen prebiranju obvestil**	**Deljenje osebnih podatkov**	**Čas namenjen prebiranju obvestil**
Uporabnik 1	Sprejel dostop do kamere	7 sekund	Sprejel dostop do lokacije	2 do 3 sekunde
Uporabnik 2	Sprejel dostop do kamere	2 do 3 sekunde	Sprejel dostop do lokacije	2 do 3 sekunde
Uporabnik 3	Sprejel dostop do kamere	2 do 3 sekunde	Sprejel dostop do lokacije	2 do 3 sekunde
Uporabnik 4	Sprejel dostop do kamere	2 do 3 sekunde	Sprejel dostop do lokacije	2 do 3 sekunde
Uporabnik 5	Sprejel dostop do kamere	4 sekunde	Sprejel dostop do lokacije	5 sekund
Uporabnik 6	Ni sprejel dostopa do kamere	5 sekund	Sprejel dostop do lokacije	2 do 3 sekunde
Uporabnik 7	Sprejel dostop do kamere	5 sekund	Ni sprejel dostopa do lokacije	2 do 3 sekunde
Uporabnik 8	Sprejel dostop do kamere	5 sekund	Sprejel dostop do lokacije	2 do 3 sekunde
Uporabnik 9	Ni sprejel dostopa do kamere	4 sekunde	Sprejel dostop do lokacije	2 do 3 sekunde

Tabela 4.2 : : Predstavitev pridobljenih podatkov pri eksperimentu (drugi del)

	Naloga 3		**Naloga 4**	
	Deljenje osebnih podatkov	**Čas namenjen prebiranju obvestil**	**Deljenje osebnih podatkov**	**Čas namenjen prebiranju obvestil**
Uporabnik 1	Sprejel dostop do lokacije	2 do 3 sekunde	Sprejel dostop do podatkov računa	2 do 3 sekunde
Uporabnik 2	Sprejel dostop do lokacije	2 do 3 sekunde	Sprejel dostop do podatkov računa	Uporabnik ni prebral obvestila
Uporabnik 3	Ni sprejel dostopa do lokacije	5 sekund	Ni sprejel dostopa do podatkov računa	3 do 4 sekunde
Uporabnik 4	Sprejel dostop do lokacije	2 do 3 sekunde	Sprejel dostop do podatkov računa	2 do 3 sekunde
Uporabnik 5	Sprejel dostop do lokacije	2 do 3 sekunde	Ni sprejel dostopa do podatkov računa	2 do 3 sekunde
Uporabnik 6	Sprejel dostop do lokacije	2 do 3 sekunde	Sprejel dostop do podatkov računa	2 do 3 sekunde
Uporabnik 7	Ni sprejel dostopa do lokacije	4 sekunde	Ni sprejel dostopa do podatkov računa	4 sekunde
Uporabnik 8	Sprejel dostop do lokacije	5 sekund	Sprejel dostop do podatkov računa	2 do 3 sekunde
Uporabnik 9	Ni sprejel dostopa do lokacije	2 do 3 sekunde	Sprejel dostop do podatkov računa	5 sekund

Z opravljenim eksperimentom in anketo, ki so jo uporabniki reševali po eksperimentu, smo pridobili podatke (Tabela 4.1 : Predstavitev pridobljenih podatkov pri eksperimentu (prvi del)(Tabela 4.2), da večino sodelujočih prebere obvestilo na hitro (2-3s). Podatke delijo le s stranmi, ki se jim zdijo varne, oziroma jih poznajo. Največ jih je zavrnilo deljenje informacij pri zadnji nalogi, ker niso želeli deliti podatkov iz drugih socialnih omrežjih in pri 3. nalogi, ker strani niso poznali. Vse razen ene osebe, so vsa obvestila prebrale, vendar zelo hitro. Tri izmed njih obvestila niso motila, ostale so motili izgled, da so obvestila prepogosta na straneh ali pa, da nudijo premalo informacij. Vse osebe razen ene so mnenja, da so obvestila sicer priročna, saj tako izvemo, kaj funkcionalnost od nas zahteva. Tri osebe večinoma

sprejemajo zahteve s strani, ostale pa samo včasih, odvisno od tega, če stran zares želijo uporabljati in ali če jo poznajo.

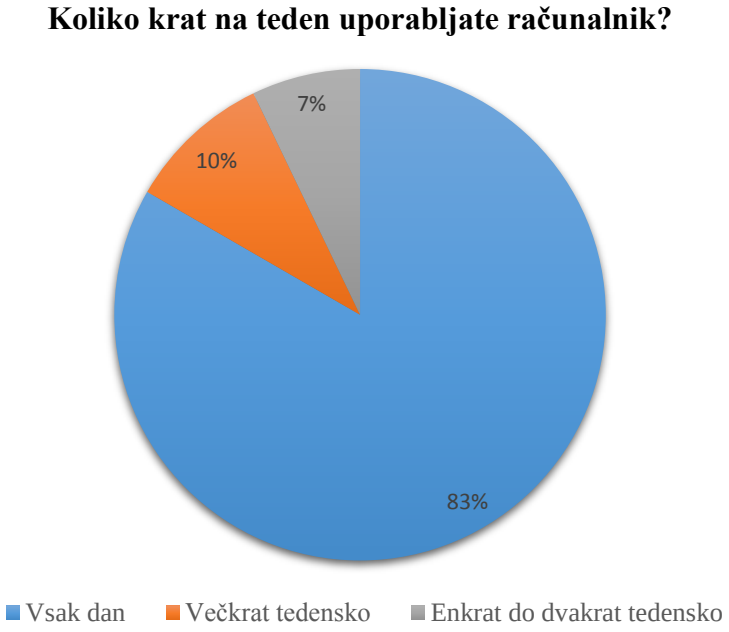
Ker pa vseh željenih odgovorov, nismo mogli pridobiti s pomočjo eksperimenta, smo za pridobitev ostalih odgovorov uporabili anketo. Prošnjo za sodelovanje in reševanje spletne ankete, smo poslali preko aplikacije Facebook, kjer smo delili povezavo na našo anketo in ljudi prosili za sodelovanje. Za takšno metodo pridobitve uporabnikov smo se odločili, saj so prav mladi tisti, ki največ uporabljajo spletne strani in aplikacije, ter pripadajoče funkcionalnosti. Večina teh uporabnikov uporablja aplikacijo Facebook. Prav tako smo anketo pošiljali prijateljem na socialnem omrežju Facebook. Na našo prošnjo se je odzvalo naključnih 42 uporabnikov. Od tega je bilo 27 ženskega spola, 15 pa moškega. Izmed tega jih je 31 bilo mlajših od 20 let, 11 pa med 21 in 40 let. Med njimi je bilo 34 študentov, 7 dijakov in ena zaposlena oseba. Odgovarjali so na 9 vprašanj, s katerimi smo želeli pridobiti odgovore na naša raziskovalna vprašanja in s tem potrditi ali ovreči definirane hipoteze.

Pri rezultatih nas je najbolj omejevalo to, da določeni uporabniki niso želeli pojasniti svojih odgovorov, kar je pripeljalo do nepopolnosti podatkov. Prav tako pa so nekateri pri kakšnem vprašanju odgovorili v nasprotju z resnico. Na primer, da je uporabnik odgovoril, da pozna določeno funkcionalnost, ter da jo je že uporabljal. Ko pa je moral razložiti, katera je ta funkcionalnost in kaj omogoča, pa je pustil prazno, ali pa je navedel nekaj kar ni bila funkcionalnost HTML5 standarda. To pa ponovno pripelje do nepopolnosti prejetih podatkov.

Pridobljenih podatkov ne moramo posplošiti na celotno populacijo, saj so anketo rešili uporabniki, ki so mlajši od 40 let. V večini torej gre za uporabnike, ki so še študentje ali dijaki. Vendar pa smo tako pridobili nekakšno predstavo, kako si ti mlajši uporabniki spleta, predstavljajo standard HTML5 in njegove funkcionalnosti. Gre torej za uporabnike, ki največ uporabljajo spletne strani in aplikacije, ter varnostne funkcionalnosti standarda HTML5. Da bi te pomanjkljivosti odpravili, bi morali pridobiti še starejši del populacije, ter pridobiti mnenje z njihove strani. Na ta način, bi lahko pridobljene podatke posplošili na večji, če ne kar celotni del populacije. Za večjo učinkovitost eksperimenta, pa bi bilo potrebno počakati, da se standard HTML5 bolj uveljavi, da se ljudje spoznajo z njim in njegovimi pripadajočimi funkcionalnostmi.

Po tem bi lahko pridobili podatke s strani ljudi, ki dejansko poznajo standard, ter bi znali povedati, kaj jih moti glede njegove zasebnosti in varnosti. Le to pa bi pomenilo bolj popolne in natančne podatke, s katerimi bi lahko ustvarili predstavo, o zavedanju uporabnikov o zasebnosti in varnosti v spletnih aplikacijah HTML5.

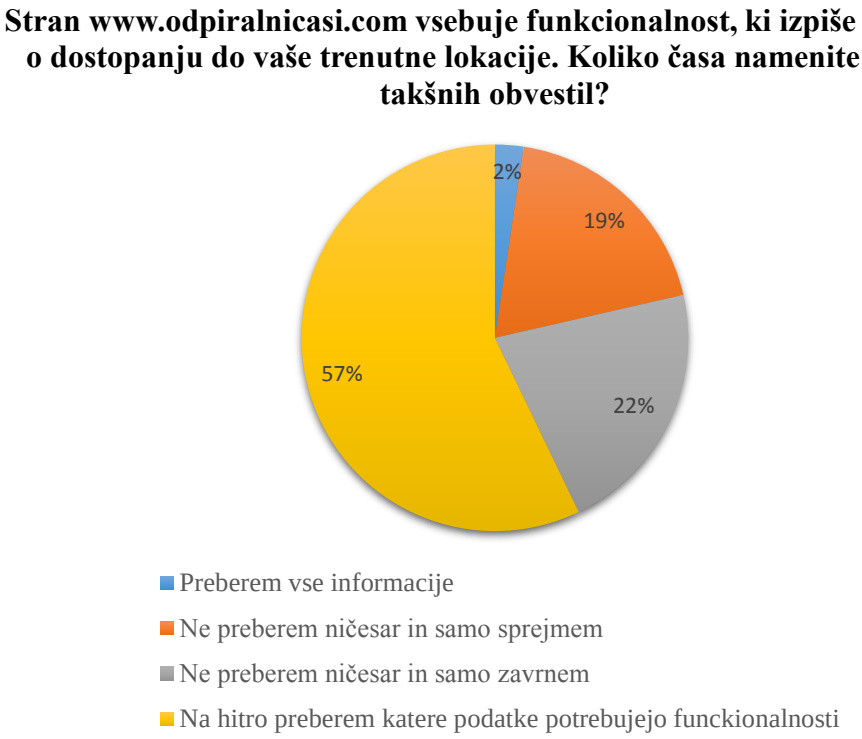
Na vprašanje koliko krat na teden uporabniki uporabljajo računalnik je odgovorilo 42 anketirancev.



Graf 4.1 : Analiza prvega vprašanja ankete

Z analizo podatkov, ki smo jih pridobili pri prvem vprašanju ankete (Graf 4.1), smo ugotovili, da kar 83% anketirancev uporablja računalnik vsak dan, 10 % večkrat tedensko in le 7% enkrat do dvakrat na teden.

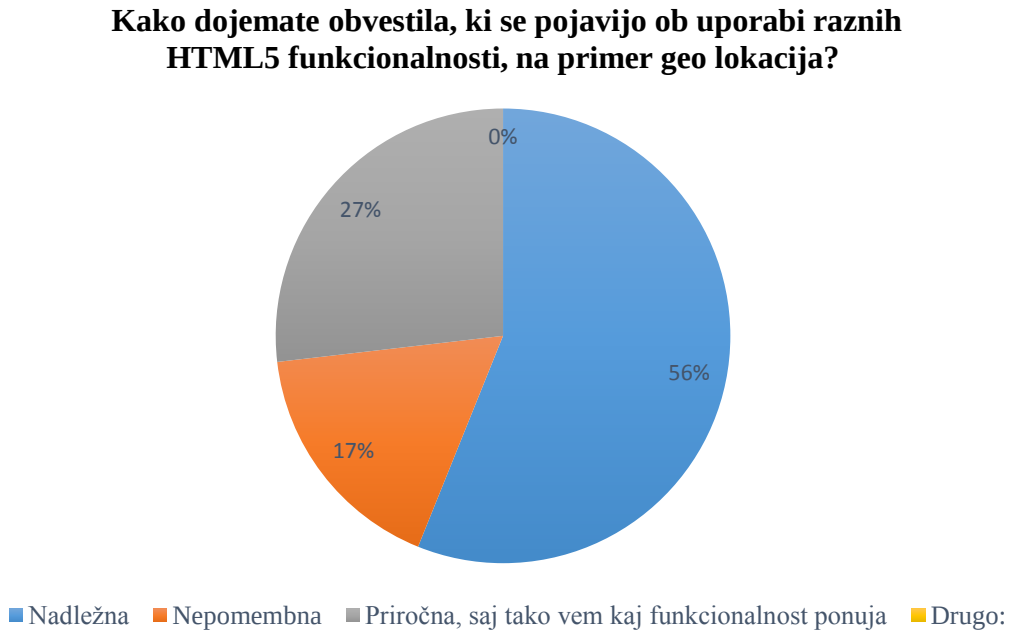
Na vprašanje "Stran www.odpiralnicasi.com vsebuje funkcionalnost, ki izpiše obvestilo o dostopanju do vaše trenutne lokacije. Koliko časa namenite branju takšnih obvestil?", je odgovorilo 42 anketirancev.



Graf 4.2 : Analiza drugega vprašanja ankete

Z analizo podatkov, pridobljenih pri drugem vprašanju ankete (Graf 4.2), smo ugotovili, da kar 57% anketirancev obvestila prebere le na hitro, 21% jih takoj zavrne, 19% takoj sprejme, samo eden pa dejansko prebere vse informacije. Le to nam pove, da ljudje sami zasebnosti in branju obvestil namenimo zelo malo časa, kar je pa velik problem v sodobni družbi, ko so občutljivi podatki povsod na spletu. S tem smo potrdili hipotezo, da uporabniki ob uporabi elementov/funkcionalnosti HTML5, zasebnosti podatkov ne bodo namenili veliko časa, ampak bodo želeli čimprej priti mimo njih.

Na vprašanje "Kako dojemate obvestila, ki se pojavijo ob uporabi raznih HTML5 funkcionalnosti, na primer geo lokacija?" je odgovorilo 41 anketirancev.

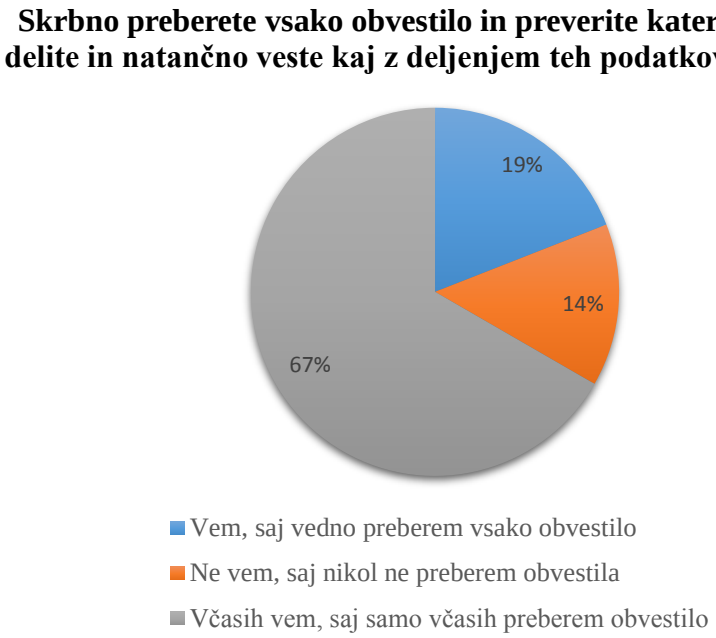


Graf 4.3 : Analiza tretjega vprašanja ankete

Z analizo pridobljenih podatkov pri tretjem vprašanju (Graf 4.3), smo ugotovili, da ima 26% anketirancev mnenje, da so obvestila priročna, 17% da so nepomembna in kar 55%, da so nadležna. Ti podatki nam tako prikazujejo, da večina anketirancev ne mara obvestil, ki se pojavijo ob uporabi HTML5 funkcionalnosti. S tem pa smo potrdili hipotezo 6, ki je trdila, da bo večji del uporabnikov varnostna sporočila dojela kot nezaželena in nadležna.

Naslednje vprašanje, ki smo ga v anketi postavili je bilo: " Kaj vas pri obvestilih HTML5 funkcionalnosti, kot je na primer geo lokacija, najbolj moti?". Vprašanje je bilo odprtega tipa, zato smo dobili veliko število različnih odgovorov. Odgovora ki pa sta se med anketiranci največkrat pojavila sta bila, da jih moti, da morajo deliti osebne podatke, saj jih skrbi, da bi jim sledili in da so obvestila vsiljiva ter nadležna. S tem pa smo potrdili hipotezo 5, ki trdi, da bo uporabnike med uporabo HTML5 funkcionalnosti najbolj motilo to, da se veliko krat pojavijo. Prav tako pa smo potrdili hipotezo 2, ki trdi, da uporabniki ne bodo želeli deliti podatkov zaradi nezaupljivosti.

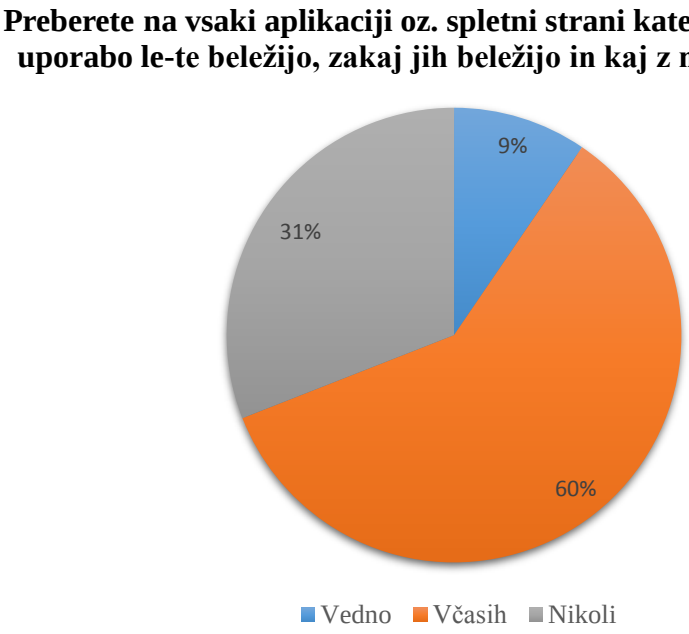
Nato smo preverjali, če anketiranci skrbno preberejo vsako obvestilo in preverijo katere podatke delijo in točno vedo, kaj z deljenjem le teh dosežejo. Na vprašanje je odgovorilo 42 anketirancev.



Graf 4.4 : Analiza petega vprašanja ankete

Z analizo podatkov pri petem vprašanju (Graf 4.4), smo ugotovili, da 67% anketirancev včasih ve kaj dosežejo z deljenjem podatkov, saj samo včasih preberejo obvestila, 14% jih ne ve kaj dosežejo, saj obvestila sploh ne berejo, medtem pa jih 19% prebere vsako obvestilo in vedo kaj z deljenjem dosežejo. S tem smo prišli do ugotovitve, da ljudje premalo časa namenijo branju obvestil, kaj zahtevajo in kaj dosežemo z deljenjem informacij. S to analizo smo ponovno potrdili hipotezo 1 ki trdi, da uporabniki ob uporabi funkcionalnosti HTML5, zasebnosti podatkov ne bodo namenili veliko časa. Prav tako pa potrdimo hipotezo 4, ki trdi, da večina uporabnikov ne bo prebrala vseh varnostnih sporočil.

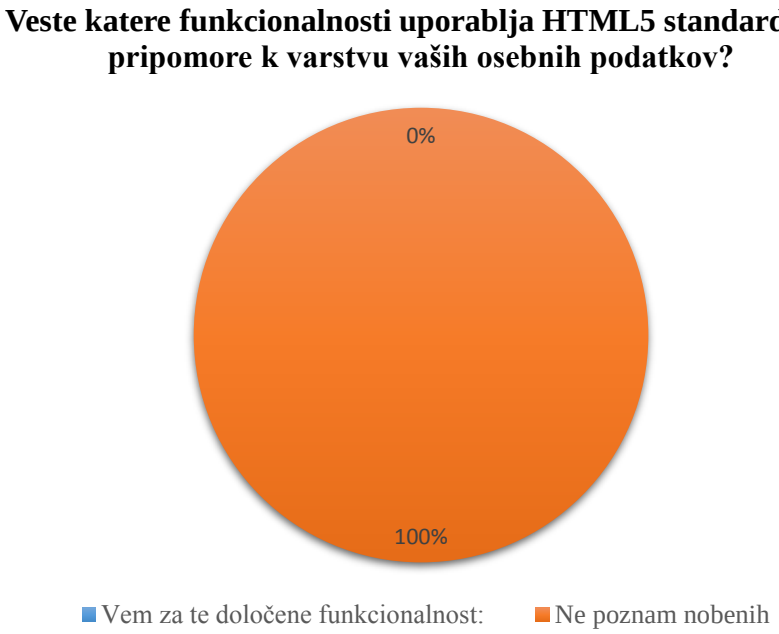
Sledilo je vprašanje, kjer smo preverjali, če anketiranci preberejo na vsaki spletni strani, katere podatke strani beležijo, zakaj jih beležijo in kaj z njimi naredijo. Na vprašanje je odgovorilo 42 anketirancev.



Graf 4.5 : Analiza šestega vprašanja ankete

Z analizo podatkov, pridobljenih iz odgovorov šestega vprašanja ankete (Graf 4.5), smo ugotovili, da 10% anketirancev vedno prebere katere podatke strani beležijo, zakaj in kaj z njimi naredijo. Kar 31% jih tega nikoli ne prebere, medtem pa 61% anketirancev to preberejo le včasih. Ker večina anketirancev samo včasih prebere te podatke, sledijo jim tisti, ki jih ne preberejo, lahko potrdimo hipotezo 1, ki trdi, da uporabniki ob uporabi funkcionalnosti HTML5, zasebnosti podatkov ne bodo namenili veliko časa

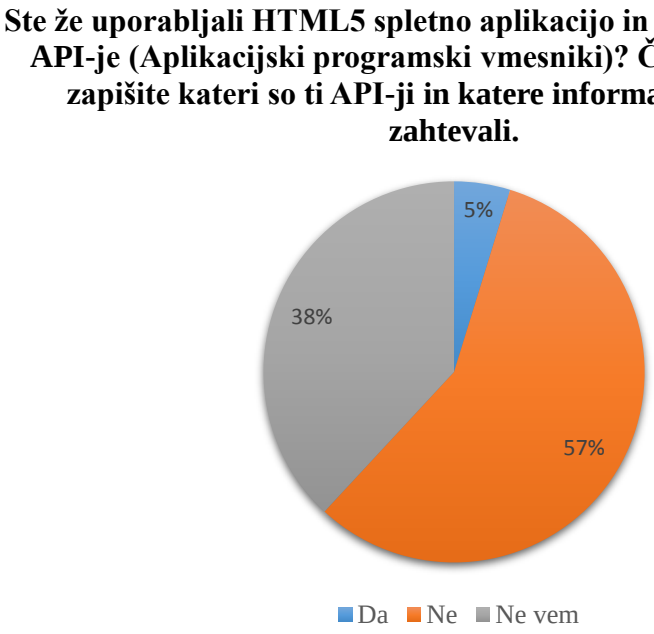
Pri naslednjem vprašanju smo želeli ugotoviti, ali uporabniki poznajo funkcionalnost HTML5 standarda, ki pripomorejo k varnosti podatkov. Na anketo je odgovorilo 42 anketirancev.



Graf 4.6 : Analiza sedmega vprašanja ankete

Analiza podatkov, ki smo jih dobili s pomočjo sedmega vprašanja (Graf 4.6), je pokazala, da nobeden od uporabnikov ne pozna funkcionalnosti HTML5, ki pripomorejo varovanju naših podatkov, kar pa je zelo zaskrbljujoče.

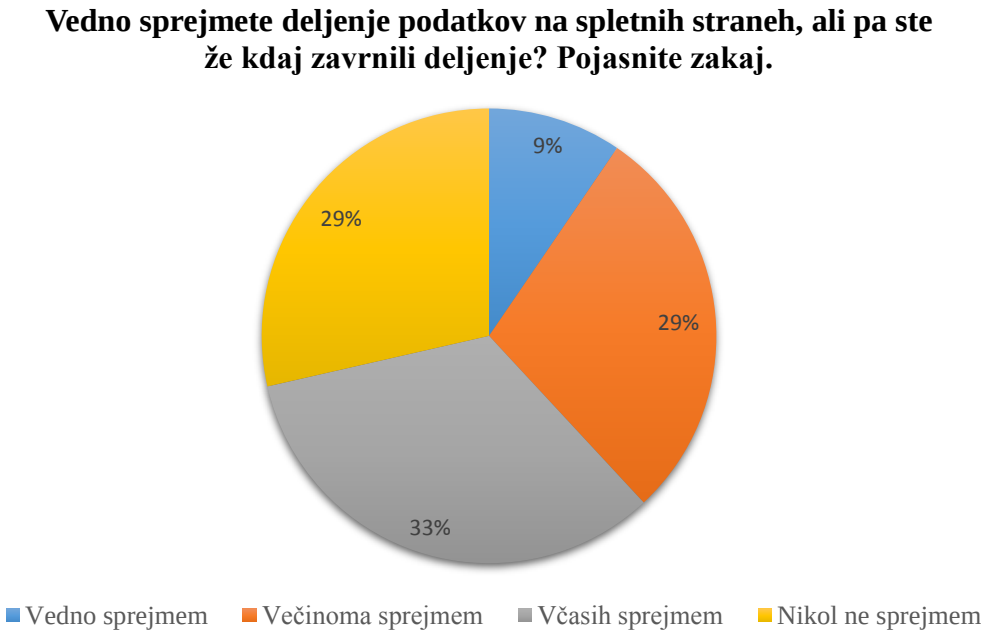
Pri naslednjem vprašanju smo preverjali, koliko anketirancev je že uporabljalo HTML5 spletno aplikacijo in njene pripadajoče aplikacijske programske vmesnike. Nato so morali zapisati katere so uporabljali, če so jih. Na vprašanje je odgovorilo 41 anketirancev.



Graf 4.7 : Analiza osmega vprašanja ankete

Po analizi smo podatkov osmega vprašanja (Graf 4.7), smo ugotovili, da 57% anketirancev ni uporabljajo HTML5 spletno aplikacijo in njene aplikacijske programske vmesnike. Izmed vseh anketirancev jih 38% ni vedelo, če so spletno aplikacijo HTML5 in njene funkcionalnosti že uporabljali, le eden anketiranec pa je že uporabljal spletno aplikacijo HTML5 in njene funkcionalnosti. Prav tako nam je analiza pokazala, da standard HTML5 še ni toliko razširjen pri uporabi, da ga ne pozna veliko ljudi, še manj pa jih ve za funkcionalnosti, ki jih standard uporablja.

Sledilo je vprašanje, kje smo preverjali ali uporabniki vedno sprejmejo deljene podatkov na spletnih straneh. Nato so morali pojasniti zakaj. Odgovorilo je 42 anketirancev.



Graf 4.8 : Analiza devetega vprašanja ankete

Analiza podatkov pri devetem vprašanju (Graf 4.8), je pokazala, da 33% anketirancev časih sprejme deljenje podatkov, 29% je takšnih, ki nikoli ne sprejme podatkov, 29% takšnih ki jih večinoma sprejmejo in 10% takih, ki jih vedno sprejmejo. Tisti ki deljenje podatkov sprejmejo večinoma ali pa vedno, so za pojasnilo dali, da podatke delijo, če ni druge možnosti za uporabe spletne strani in je pogoj za nadaljnjo uporabo strani. Med tistimi, ki pa jih zavrnejo včasih ali vedno, je prevladoval odgovor, da aplikacijam ne zaupajo, ali pa ne želijo deliti osebnih podatkov. S tem smo potrdili hipotezi 2 in 3, saj večina anketirancev samo včasih sprejme deljenje podatkov, ali pa sploh ne. Potrdili smo, da uporabniki ne bodo želeli deliti funkcionalnosti, ker ne zaupajo spletni strani in da bodo uporabniki zaradi nezaupljivosti pri deljenju podatkov izpustili določene funkcionalnosti.

5 Sklep

V diplomskem delu smo prikazali, v kolikšni meri se uporabniki zavedajo zasebnosti in varnosti v spletnih aplikacijah HTML5. Na kratko smo predstavili standard HTML5, njegove funkcionalnosti in kako nam standard omogoča povečati varnost in zasebnost na spletnih straneh. Preverili smo uporabo samega standarda HTML5 in njegovih pripadajočih funkcionalnosti med uporabniki in poznavanjem le teh. Prišli smo do ugotovitve, da standard HTML5 še ni tako uveljavljen in uporabljen znotraj spletnih strani. Število strani, ki uporabljajo specifikaciji HTML5 se sicer veča, vendar standard še ni dosegel visokega nivoja prepoznavnosti. Prav tako smo ugotovili, da ima še precej ranljivosti, kar je tudi eden izmed razlogov, da še ni široko uporabljen. Prav zagotovo nudi veliko simpatičnih novosti, ki pa ne pomenijo nič, če standard ni čim bolj varen.

Ugotovili smo, da standard HTML5 vsebuje veliko število novih funkcionalnosti, ki dajo veliko na vizualno podobo strani. Z velikim številom novih funkcionalnost, pa prav tako pridejo nove številne ranljivosti, ki pa jih je v HTML5 na žalost zaenkrat še veliko. Prav tako smo prišli do ugotovitev, da je število ljudi, ki samih funkcionalnosti HTML5 standarda ne pozna veliko. Prav tako je kar nekaj takšnih, ki ne ve, kako lahko pripomore uporaba funkcionalnosti HTML5, za varnost ter zasebnost osebnih podatkov uporabnikov.

Pri analizi testnih podatkov, smo prišli do ugotovitev, da ljudje namenijo malo časa branju varnostnih obvestil, ki se pojavijo na spletnih straneh, ki želijo dostopati do osebnih podatkov. Le to pa je zelo zaskrbljujoče, saj lahko na tak način hitro kdo pride do naših osebnih podatkov in jih zlorabi. Do tega pa lahko pride še posebej v tej dobi, ko so računalniki in podatki na spletu močno razširjeni. Ugotovili smo tudi, da obvestila na spletnih straneh uporabniki dojemajo kot nadležna in nepotrebna, zaradi česar obvestil večinoma sploh ne berejo in jih samo sprejmejo. To pot izberejo predvsem za to, ker ne želijo prebirati takšnih obvestil in želijo nemotene uporabljati spletne strani, ali pa obvestila preberejo le bežno. Pri tem pa lahko pride do tega, da uporabniki delijo podatke, ki jih sicer ne bi želeli. Velika večina uporabnikov se za to, katere podatke strani pridobivajo in zakaj jih uporabljajo, ne briga. Le to pa je ponovno problem, saj tako omogočimo zlonamernim stranem, da naše osebne podatke zlorabijo.

Mnogo pa je tudi takšnih uporabnikov, ki veliko krat sploh ne vedo, zakaj in katere osebne podatke morajo deliti. Tukaj ponovno vidimo malomarnost uporabnikov, pri zaščiti lastnih podatkov. Večji del uporabnikov raje podatkov ne deli, sploh ne straneh, ki zahtevajo občutljive podatke, ali na straneh ki jih ne poznajo. Le to pa nam pove, da uporabniki ne delijo vseh podatkov, ki jih strani zahtevajo, sploh če strani ne poznajo in ji ne zaupajo.

Prišli smo do ugotovitve, da se uporabniki le malo zavedajo zasebnosti in varnosti znotraj spletnih aplikacij HTML5. Velika večina jih večji poudarek da na to, da lahko nemoteno uporabljajo spletno stran, čeprav to včasih pripelje do uporabe zasebnih podatkov. Medtem ko varnostne funkcionalnosti, pri uporabi spletnih strani HTML5 obstajajo za to, da bi nam pomagale, jih večina uporabnikov dojema kot nepotrebne in nadležne. Je pa veliko takšnih uporabnikov, ki teh funkcionalnosti niti ne pozna, kar nam pove, da standard HTML5 še ni toliko razširjen in uporabljen na straneh.

Težave na katere smo pri izvajanju eksperimenta naleteli so, da naprava Eye Guide ne zazna očesa oziroma, ga težje locira v primeru, da uporabnik nosi očala. Uporabniki so večinoma imeli težavo s tem, ker glave niso mogli opreti na kakšen objekt, kar bi povečalo natančnost podatkov, ki bi jih pridobili s strani naprave. Opaziti je bilo, da določene osebe mogoče niso reagirale tako, kot bi reagirale v svojem domačem okolju, nekateri so preveč premikali glavo, kar je pripeljalo do tega, da nismo mogli pridobiti točnih podatkov od njih. Naprava je včasih prenehala delovati, ali pa je zmanjkalo povezave naprave na računalnik, največ časa smo porabili pri kalibriranju naprave, saj se je le ta na čase čudno vedla. So pa se problemi pojavili tudi pri analiziranju podatkov, saj je program veliko krat prenehal delovati in je ob previjanju/premoru videa, za nekaj časa zaustavil delovanje, nato pa nadaljeval.

Pri nadaljnjem delu z napravami Eye tracker je zaželeno, da uporabniki ne nosijo očal. Tako bo naprava hitro in zanesljivo locirala oko. Predlagamo, da se uporabniki že prej malo spoznajo z opremo, da ne pride do nevšečnosti in velikih zamikov, v poteku eksperimenta. Priporočeno je, da imajo na voljo kakšen opornik za glavo, da pride do čim manj premikanj z glavo. S tem omogočimo čim bolj točne in zanesljive podatke. Pomembno pa je, da se oprema zažene in uporablja vsaj kakšno uro pred testiranjem, da se uteče in pride do čim manj problemov.

6 Literatura in viri

[1] About HTML5 WebSocket. (brez datuma). Pridobljeno 01. 09 2016 iz https://www.websocket.org/aboutwebsocket.html.

[2] Amazon. (brez datuma). Data Privacy. Pridobljeno 10. 08 2016 iz https://aws.amazon.com/compliance/data-privacy-faq/.

[3] Bidelman, E. (26. 07 2010). The Basics of Web Workers. Pridobljeno 01. 09 2016 iz http://www.html5rocks.com/en/tutorials/workers/basics/#toc-security.

[4] Biedert, R., Buscher, G., & Dengel, A. (2010). The eyeBook: Introduction, Eye Tracking Technology. Pridobljeno 07. 08 2016 iz http://ureca.recherche.univ-lille3.fr/sparrow/cesi2013/The%20eyeBook.pdf.

[5] Burnham, K. (05. 09 2014). 5 Ways Snapchat Violated Your Privacy, Security. Pridobljeno 29. 08 2016 iz http://www.informationweek.com/software/social/5-ways-snapchat-violated-your-privacy-security/d/d-id/1251175.

[6] Canvas tutorial. (2015). Pridobljeno 31. 08 2016 iz https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial.

[7] Center, H. D. (2015). WebSocket Security. Pridobljeno 01. 09 2016 iz https://devcenter.heroku.com/articles/websocket-security.

[8] Chaffey, D. (2016). Mobile Marketing Statistics compilation. Pridobljeno 06. 08 2016 iz http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/.

[9] Commission, E. (brez datuma). Cookies. Pridobljeno 07. 08 2016 iz http://ec.europa.eu/ipg/basics/legal/cookies/index_en.htm.

[10] Commission, E. (brez datuma). Protection of personal data. Pridobljeno 07. 08 2016 iz http://ec.europa.eu/justice/data-protection/.

[11] Deperry, D. (04. 12 2012). HTML5 security the mordern web browser perspective : Introduction. San Francisco. Pridobljeno 04. 08 2016.

[12] Deutsch, A. (15. 12 2015). Google faces $18 million fine for web privacy violations: Dutch watchdog. Pridobljeno 29. 08 2016 iz http://www.reuters.com/article/us-privacy-google-dutch-idUSKBN0JT1TG20141215.

[13] Devlin, I. (14. 06 2011). Finding your position with Geolocation. Pridobljeno 31. 08 2016 iz http://html5doctor.com/finding-your-position-with-geolocation/.

[14] Duchowski, A. (2007). Eye Tracking Methodology : Eye Tracking Applications. Springer. Pridobljeno 08. 07 2016 iz https://books.google.si/books?hl=sl&lr=&id=WtvVdNESRyIC&oi=fnd&pg=PR15&dq=Eye+Tracking+Methodology:+Theory+and+Practice+andrew+pdf&ots=8lya6tFMgw&sig=a5Z8I8QPS3LI8Akezx97Qd1KgKk&redir_esc=y#v=onepage&q=Eye%20Tracking%20Methodology%3A%20Theory%20and%20Prac.

[15] Duffy Marsan , C. (26. 01 2012). 15 worst Internet privacy scandals of all time. Pridobljeno 29. 08 2016 iz http://www.networkworld.com/article/2185187/security/15-worst-internet-privacy-scandals-of-all-time.html.

[16] Durham, G. (13. 03 2013). Privacy Research Paper : Conclusion. Pridobljeno 11. 08 2016 iz http://gator.ndm.edu/~gdurham/privacy.html.

[17] Essential guide: EU Data Protection Regulation. (brez datuma). Pridobljeno 01. 09 2016 iz http://www.computerweekly.com/guides/Essential-guide-What-the-EU-Data-Protection-Regulation-changes-mean-to-you.

[18] Foundations of an HTML5 Web app. (2015). Pridobljeno 06. 08 2016 iz https://developer.mozilla.org/en-US/Apps/Tutorials/General/Foundations_of_an_HTML5_Web_App.

[19] Fuchs, C. (2013). Privacy and Security in Europe : Conclusion. Pridobljeno 11. 08 2016 iz http://www.projectpact.eu/privacy-security-research-paper-series/privacy-security-research-paper-series/6_Privacy_and_Security_Research_Paper_Series.pdf.

[20] Hammant, P. (02. 01 2016). CORS vulnerabilities. Pridobljeno 02. 09 2016 iz http://paulhammant.com/2016/01/02/CORS-vulnerabilities/.

[21] Heffetz, O., & Ligett, K. (09 2013). Privacy and data-based research : Concluding Thoughts. Cambridge. Pridobljeno 11. 08 2016 iz http://www.nber.org/papers/w19433.pdf.

[22] Heiderich, M. (2016). HTML 5: The good, the bad, the ugly. Pridobljeno 30. 08 2016 iz https://www.nds.rub.de/media/nds/attachments/files/2010/11/html5_preso_rub_2010.pdf.

[23] Hickson, I. (18. 11 2010). HTML5 Web Messaging. Pridobljeno 01. 09 2016 iz https://www.w3.org/TR/2010/WD-webmessaging-20101118/#security-postmsg.

[24] Hickson, I. (17. 03 2011). HTML5 Web Messaging. Pridobljeno 02. 09 2016 iz https://www.w3.org/TR/2011/WD-webmessaging-20110317/#security-postmsg.

[25] Hickson, I. (19. 04 2016). Web Storage - Security. Pridobljeno 01. 09 2016 iz https://www.w3.org/TR/webstorage/#security-storage.

[26] Homakov, E. (28. 07 2015). Let's make Offline Web Applications secure. Pridobljeno 01. 09 2016 iz http://sakurity.com/blog/2015/07/28/appcache.html.

[27] How to comply with the EU cookie law. (brez datuma). Pridobljeno 01. 09 2016 iz http://www.computerweekly.com/guides/How-to-comply-with-the-EU-cookie-law.

[28] HTML element reference : <avdio>. (2016). Pridobljeno 31. 08 2016 iz https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio.

[29] HTML5. (2016). Pridobljeno 31. 08 2016 iz https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5.

[30] Ihrig , C. (12. 05 2012). An Overview of the Web Storage API. Pridobljeno 01. 09 2016 iz https://www.sitepoint.com/an-overview-of-the-web-storage-api/.

[31] IndexedDB API. (2016). Pridobljeno 31. 08 2016 iz https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API.

[32] International Privacy. (brez datuma). Data Protection. Pridobljeno 01. 09 2016 iz https://www.privacyinternational.org/node/44.

[33] Internet privacy. (2016). Pridobljeno 30. 08 2016 iz https://en.wikipedia.org/wiki/Internet_privacy.

[34] Introduction to Web Application Security. (brez datuma). Pridobljeno 02. 08 2016 iz https://msdn.microsoft.com/en-us/library/aa711426(v=vs.71).aspx.

[35] Jenkov, J. (08. 08 2014). HTML5 Web Workers. Pridobljeno 31. 08 2016 iz http://tutorials.jenkov.com/html5/web-workers.html.

[36] Johnes, Z. (28. 05 2013). Web Storage Security. Pridobljeno 01. 09 2016 iz https://www.whitehatsec.com/blog/web-storage-security/.

[37] Lubbers, P., Albers, B., & Salim, F. (2010). Pro HTML5 programming : Using the HTML5 WebSocket API. Združene države Amerike: Manning P. Pridobljeno iz ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/[Apress]%20-%20Pro%20HTML5%20Programming%20-%20[Lubbers,%20Albers].pdf.

[38] Lubbers, P., Albers, B., & Salim, F. (2010). Pro HTML5 programming : Overview of HTML5 Canvas . Združene države Amerike: Manning P. Pridobljeno iz ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/[Apress]%20-%20Pro%20HTML5%20Programming%20-%20[Lubbers,%20Albers].pdf.

[39] Lubbers, P., Albers, B., & Salim, F. (2010). Pro HTML5 programming : Using the Communication APIs. Združene države Amerike: Manning P. Pridobljeno iz ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/[Apress]%20-%20Pro%20HTML5%20Programming%20-%20[Lubbers,%20Albers].pdf.

[40] Lubbers, P., Albers, B., & Salim, F. (2010). Pro HTML5 programming : Using the HTML5 Forms API. Združene države Amerike: Manning P. Pridobljeno iz ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/[Apress]%20-%20Pro%20HTML5%20Programming%20-%20[Lubbers,%20Albers].pdf.

[41] Lubbers, P., Albers, B., & Salim, F. (2010). Pro HTML5 programming : Using the HTML5 Geolocation API. Združene države Amerike: Manning P. Pridobljeno iz ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/[Apress]%20-%20Pro%20HTML5%20Programming%20-%20[Lubbers,%20Albers].pdf.

[42] Lubbers, P., Albers, B., & Salim, F. (2010). Pro HTML5 programming : Using the HTML5 Web Storage API. Združene države Amerike: Manning P. Pridobljeno iz ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/[Apress]%20-%20Pro%20HTML5%20Programming%20-%20[Lubbers,%20Albers].pdf.

[43] Lubbers, P., Albers, B., & Salim, F. (2010). Pro HTML5 programming : Working with HTML5 Audio and Video. Združene države Amerike: Manning P. Pridobljeno iz ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/[Apress]%20-%20Pro%20HTML5%20Programming%20-%20[Lubbers,%20Albers].pdf.

[44] MacDonald, M. (2011). HTML5 The missing manual : Basic Drawing with the Canvas. O'Reilly Media, Inc. Pridobljeno iz http://people.inf.elte.hu/zirtaai/html_ebooks/HTML5_The_Missing_Manual.pdf.

[45] MacDonald, M. (2011). HTML5 The missing manual : Data Storage. O'Reilly Media, Inc. Pridobljeno iz http://people.inf.elte.hu/zirtaai/html_ebooks/HTML5_The_Missing_Manual.pdf.

[46] MacDonald, M. (2011). HTML5 The missing manual : Audio and video. O'Reilly Media, Inc. Pridobljeno iz http://people.inf.elte.hu/zirtaai/html_ebooks/HTML5_The_Missing_Manual.pdf.

[47] MacDonald, M. (2011). HTML5 The missing manual : Offline Applications. O'Reilly Media, Inc. Pridobljeno iz http://people.inf.elte.hu/zirtaai/html_ebooks/HTML5_The_Missing_Manual.pdf.

[48] MacDonald, M. (08 2011). HTML5 The missing manual : Three Key Principles of HTML5. O'Reilly Media, Inc. Pridobljeno iz http://people.inf.elte.hu/zirtaai/html_ebooks/HTML5_The_Missing_Manual.pdf.

[49] MacDonald, M. (2011). HTML5 The missing manual : Web Forms, Refined. O'Reilly Media, Inc. Pridobljeno iz http://people.inf.elte.hu/zirtaai/html_ebooks/HTML5_The_Missing_Manual.pdf.

[50] MacDonald, M. (2011). HTML5 Them missing manual : Audio and Video. O'Reilly Media, Inc. Pridobljeno iz http://people.inf.elte.hu/zirtaai/html_ebooks/HTML5_The_Missing_Manual.pdf.

[51] Mehta, L. (04. 12 2014). WebSocket Security Issues. Pridobljeno 01. 09 2016 iz http://resources.infosecinstitute.com/websocket-security-issues/.

[52] Microsoft. (brez datuma). Ensuring User Privacy. Pridobljeno 01. 09 2016 iz https://technet.microsoft.com/en-us/library/cc939818.aspx.

[53] Minnick, C., & Tittel, E. (25. 06 2015). How to Ensure Privacy in the Age of HTML5. Pridobljeno 06. 08 2016 iz http://www.cio.com/article/2384651/web-services/how-to-ensure-privacy-in-the-age-of-html5.html.

[54] Offline Web applications. (2016). Pridobljeno 08. 31 2016 iz https://html.spec.whatwg.org/#offline.

[55] Open web. (2016). Pridobljeno 03. 09 2016 iz https://en.wikipedia.org/wiki/Open_Web.

[56] Owasp. (brez datuma). HTML5 Security Cheat Sheet. Pridobljeno 02. 08 2016 iz https://www.owasp.org/index.php/HTML5_Security_Cheat_Sheet#Communication_APIs.

[57] Popescu, A. (24. 10 2014). Geolocation API Specification. Pridobljeno 02. 09 2016 iz https://www.w3.org/TR/geolocation-API/#security.

[58] Rights, E. U. (2014). Handbook on European data protection law : Rules on lawful processing. Pridobljeno 01. 09 2016 iz http://www.echr.coe.int/Documents/Handbook_data_protection_ENG.pdf.

[59] Röthlisberger, T. (brez datuma). HTML5 Web Security. Compass Security AG . Pridobljeno 30. 08 2016 iz http://media.hacking-lab.com/scs3/scs3_pdf/SCS3_2011_Roethlisberger.pdf.

[60] Schmidt, M. (06. 12 2011). HTML5 web security. Pridobljeno 04. 08 2016 iz https://www.compass-security.com/fileadmin/Datein/Research/White_Papers/html5_web_security_whitepaper.pdf.

[61] Schmidt, M. (06. 12 2011). HTML5 web security : Motivation. Pridobljeno 04. 08 2016 iz https://www.compass-security.com/fileadmin/Datein/Research/White_Papers/html5_web_security_whitepaper.pdf.

[62] Seltzer, W. (2014). Privacy Activity Statement. Pridobljeno 10. 08 2016 iz https://www.w3.org/Privacy/Activity.html.

[63] Shahgholi, A., Barzegar, H., & Babu, P. (05 2012). HTML5 Security: Offline Web Application. IEEE. Pridobljeno 01. 09 2016 iz https://www.scribd.com/document/92505594/html5-offline-application-security.

[64] Siciliano, R. (07. 08 2013). What Are the Risks of Geo-Location. Pridobljeno 01. 09 2016 iz https://blogs.mcafee.com/consumer/geo-location/.

[65] Siemens Data Privacy Policy. (02. 02 2015). Pridobljeno 10. 08 2016 iz http://www.siemens.com/corp/en/index/privacy.htm.

[66] Sitebeam. (brez datuma). Guide to the Cookie Law. Pridobljeno 07. 08 2016 iz https://sitebeam.net/cookielaw/.

[67] Smith, J. H., Dinev, T., & Xu, H. (12 2011). Information privacy research : Future Research and Conclusion. Pridobljeno 11. 08 2016 iz http://pal.ist.psu.edu/MISQ.pdf.

[68] Solution, R. S. (2011). Web form 2.0. Pridobljeno 31. 08 2016 iz http://www.html5tutorial.info/html5-webform2.php.

[69] The Cookie Law Explained. (brez datuma). Pridobljeno 07. 08 2016 iz https://www.cookielaw.org/the-cookie-law/.

[70] The EU Cookie Law. (01. 02 2016). Pridobljeno 01. 09 2016 iz http://tosbourn.com/the-eu-cookie-law/.

[71] The Google Maps Geolocation API. (2016). Pridobljeno 31. 08 2016 iz https://developers.google.com/maps/documentation/geolocation/intro.

[72] tracking, E. (2011). What is eye tracking. Pridobljeno 07. 08 2016 iz http://www.eyetracking.com/About-Us/What-Is-Eye-Tracking.

[73] Tump, E. (2011). The Risks of Client-Side Data Storage. Pridobljeno 02. 09 2016 iz https://www.sans.org/reading-room/whitepapers/dlp/risks-client-side-data-storage-33669.

[74] Using Web Workers. (brez datuma). Pridobljeno 01. 09 2016 iz https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers.

[75] van Kesteren, A. (16. 01 2014). Cross-Origin Resource Sharing : Security Considerations. Pridobljeno 02. 09 2016 iz https://www.w3.org/TR/cors/#security.

[76] WAGmob. (11. 06 2014). Learn HTML5 : What is HTML5. WAGmob. Pridobljeno 06. 08 2016 iz https://books.google.si/books?id=7TwIAQAAQBAJ&pg=PP1&lpg=PP1&dq=Learn+HTML5-+simpleNeasyBook+by+WAGmob&source=bl&ots=PzV-TMmwFT&sig=454EpeowYE1G3PENJDMGip_4YV4&hl=sl&sa=X&ved=0ahUKEwj3jJuPpa3OAhWH7xQKHbjyDLMQ6AEIKDAC#v=onepage&q=Learn%20HTML5-%20simpleNea.

[77] Web Storage API. (brez datuma). Pridobljeno 31. 08 2016 iz https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API.

[78] WebSockets - An Introduction. (brez datuma). Pridobljeno 02. 09 2016 iz https://gist.github.com/subudeepak/9897212.

[79] Whatwg. (brez datuma). Audience. Pridobljeno 31. 08 2016 iz https://html.spec.whatwg.org/#audience.

[80] Whatwg. (brez datuma). Common infrastructure. Pridobljeno 31. 08 2016 iz https://html.spec.whatwg.org/#infrastructure.

[81] Whatwg. (brez datuma). Forms. Pridobljeno 31. 08 2016 iz https://html.spec.whatwg.org/#introduction-4.

[82] Whatwg. (brez datuma). Forms. Pridobljeno 01. 09 2016 iz https://html.spec.whatwg.org/multipage/forms.html#forms.

[83] Whatwg. (brez datuma). The canvas element. Pridobljeno 31. 08 2016 iz https://html.spec.whatwg.org/#the-canvas-element.

[84] Whatwg. (brez datuma). The video element. Pridobljeno 31. 08 2016 iz https://html.spec.whatwg.org/#the-video-element.

[85] Whatwg. (brez datuma). Web sockets. Pridobljeno 31. 08 2016 iz https://html.spec.whatwg.org/#network-intro.

[86] Whatwg. (brez datuma). Web workers. Pridobljeno 31. 08 2016 iz https://html.spec.whatwg.org/#workers.

[87] Window.postMessage(). (brez datuma). Pridobljeno 01. 09 2016 iz https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage.

7 Priloge