CARNET

Sigurnosni rizici JavaScript kôda prilikom pregledavanja weba

CERT.hr-PUBDOC-2019-8-385

CERT.hr

Sadržaj

Ovaj dokument izradio je Laboratorij za sustave i signale Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

1 Uvod

JavaScript je već nekoliko godina zaredom najkorišteniji jezik u svijetu programiranja i trenutno ga koristi **više od 95% svih javno dostupnih web stranica** (1) (2).

Današnje web stranice svoje intuitivno i interaktivno sučelje duguju upravo JavaScriptu, stoga ne čudi da se sve češće koriste biblioteke, platforme i razvojni okviri temeljeni upravo na njemu (*JQuery*, *Angular*, *React*…) (2).

I dok s jedne strane JavaScript omogućuje dinamičnost i niz praktičnih funkcionalnosti zbog kojih se korisnicima stranica više sviđa te korisničko iskustvo postaje sve bolje, sa sobom **donosi i brojne rizike za sigurnost i privatnost**.

U ovom dokumentu prikazane su i objašnjene najčešće prijetnje koje se oslanjaju na JavaScript kôd, kao i načini na koje se korisnik može zaštititi. Opisana su i tri dodatka web preglednicima (engl. *browser add-ons*), *uBlock Origin*, *NoScript* i *uMatrix*, kojima je moguće u nekoj mjeri onemogućiti izvršavanje zlonamjernog JavaScript kôda i postići neke dodatne mjere zaštite.

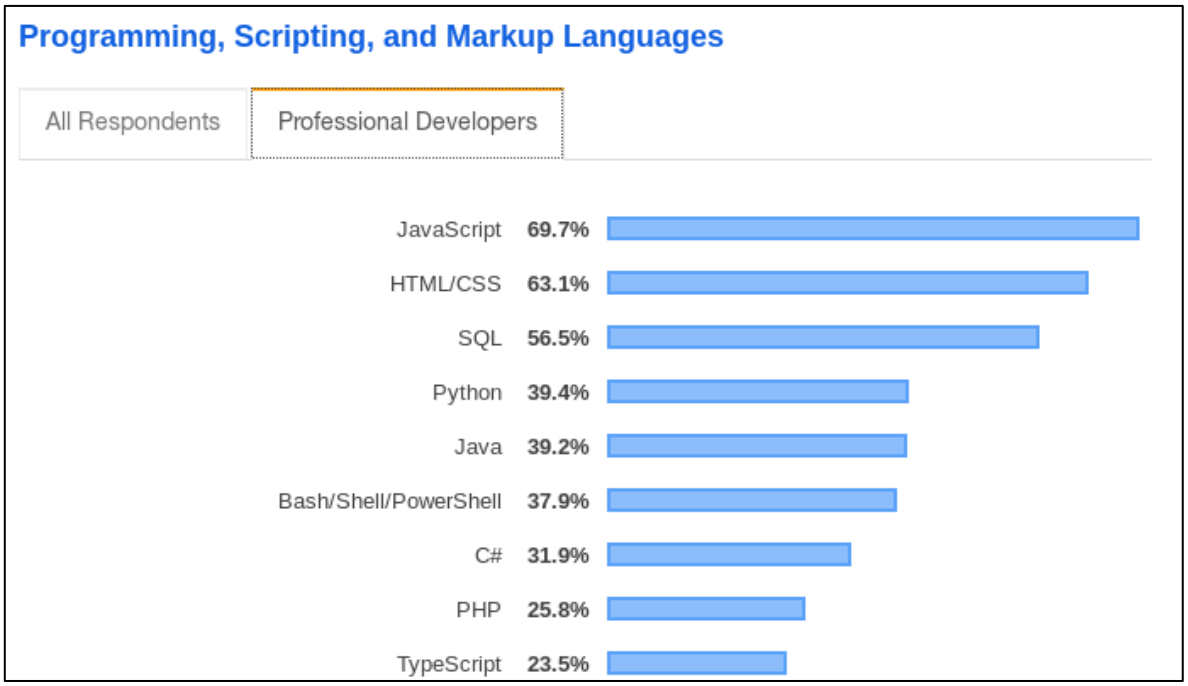
2 Osnovna struktura web stranice

Za razumijevanje rizika JavaScript kôda, potrebno je u osnovama razumjeti kako s tehničke strane funkcioniraju web stranice. Iz perspektive korisnika/klijenta, web stranica se sastoji od:

- **HTML-a** koji definira strukturu i sadržaj stranice,
- **CSS-a** kojim se definira izgled (stil) stranice (npr. boje, margine...),
- te programskog kôda u jeziku **JavaScript** koji dodaje dinamički sadržaj i upravlja ponašanjem stranice ovisno o korisnikovim aktivnostima. Taj program ne izvršava poslužitelj, već web preglednik na korisnikovom računalu i od tud proizlaze opasnosti, jer korisnik sa stranog poslužitelja preuzima nepoznati program i izvršava ga.

Osim navedenoga, bitna komponenta weba su i HTTP, mrežni protokol kojim komuniciraju korisnici i web stranice, te razne tehnologije na strani poslužitelja (jezici poput PHP-a, baze podataka...), no njihovo razumijevanje nije ključno u ovom kontekstu.

Error! Reference source not found. prikazuje rezultate istraživanja tvrtke *StackOverflow* iz kojih je vidljivo da su **JavaScript, HTML i CSS bili najkorišteniji jezici u svijetu programiranja** početkom 2019. godine (2).



***Slika 2.1 JavaScript, HTML i CSS su trenutno najkorišteniji jezici u svijetu programiranja* (2)**

Navedeni jezici izvrsno su dokumentirani i popraćeni primjerima na brojnim službenim i neslužbenim resursima diljem weba. Osim toga, budući da su vrlo često korišteni i postoje više od 20 godina, uređivači teksta su im prilagođeni i imaju ugrađena rješenja koja olakšavaju sam proces pisanja kôda automatskim prijedlozima, zatvaranjem zagrada i oznaka i sličnim korisnim funkcionalnostima koje olakšavaju i ubrzavaju pisanje programa i smanjuju broj formalnih grešaka.

2.1 HTML

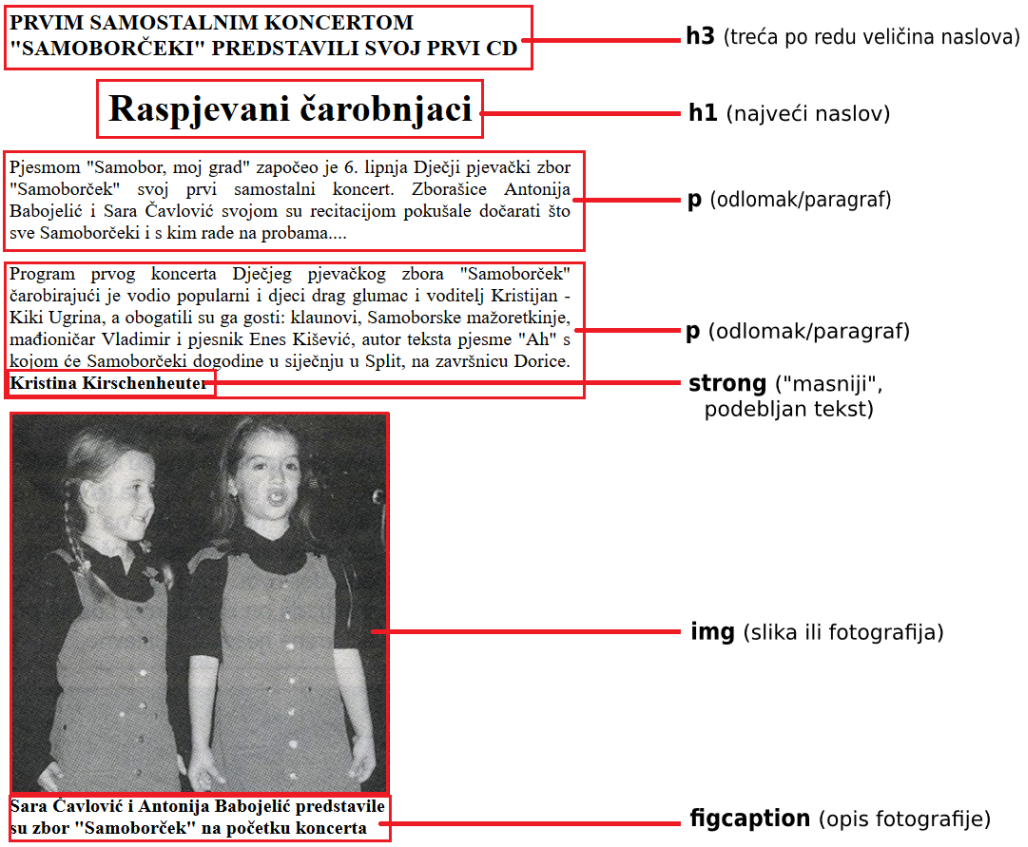
HTML, punim nazivom *HyperText Markup Language*, pripada skupini takozvanih „jezika za označavanje“ (engl. *markup language*) i čini osnovu, tj. **kostur svake web stranice**.

Njim se opisuje struktura stranice, podaci koji će se na stranici prikazivati (tekst, slike, poveznice…) i razdvajaju se različite vrste sadržaja. Za to se koriste **oznake** (engl. *tags*) na način da se sadržaj koji pripada nekoj određenoj vrsti elementa mora nalaziti između odgovarajuće otvarajuće i zatvarajuće oznake, npr. *<body> i </body>*. Oznake su uputa web pregledniku (engl. *browser*) kako interpretirati, odnosno prikazati sadržaj web stranice.

Slika 2.2 prikazuje novinski članak čiji je tekstualni sadržaj logički podijeljen u nadnaslov, naslov i odlomke. Digitalni svijet zamijenio je papirnati, ali pristup prikaza informacija je ostao isti – sadržaj koji se želi prikazati treba podijeliti na logičke dijelove, tj. elemente, kako bi bio pregledniji korisniku koji ga čita. Upravo to radi HTML.

Slika 2.2 Primjer novinskog članka

Kada bi se prethodni članak preslikao u web stranicu, sastojao bi se od elemenata koji su prikazani na Slika 2.3.



Slika 2.3 Novinski članak preslikan u web stranicu

Na Slika 2.4 prikazan je HTML kôd za prethodno prikazanu web stranicu.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5      </head>
6      <body>
7          <h3> PRVIM SAMOSTALNIM KONCERTOM "SAMOBORČEKI" PREDSTAVILI SVOJ PRVI CD </h3>
8
9          <h1> Raspjevani čarobnjaci </h1>
10
11         <p> Pjesmom "Samobor, moj grad" započeo je 6. lipnja Dječji pjevački zbor "Samoborček" svoj prvi
           samostalni koncert. Zborašice Antonija Babojelić i Sara Čavlović svojom su recitacijom pokušale
           dočarati što sve Samoborčeki i s kim rade na probama.... </p>
12
13         <p> Program prvog koncerta Dječjeg pjevačkog zbora "Samoborček" čarobirajući je vodio popularni i
           djeci drag glumac i voditelj Kristijan... <span> Kristina Kirschenheuter </span> </p>
14
15         <img src="slike/slika.jpg" />
16         <figcaption> <strong> Sara Čavlović i Antonija Babojelić predstavile su zbor "Samoborček" na
           početku koncerta </strong> </figcaption>
17
18     </body>
19 </html>
```

uputa web pregledniku da je riječ o HTML 5.0 dokumentu

zaglavlje s metapodacima (podaci o dokumentu)

omogućuje prikaz hrvatskih slova

početak sadržaja stranice, tj. HTML dokumenta

Slika 2.4 Prikaz HTML dokumenta u uređivaču teksta

2.2 CSS

Iako su se nekada u HTML-u izravno koristili ugrađeni atributi koji su definirali izgled poput poravnanja, boje, veličine teksta i sličnih karakteristika, u današnjem HTML 5.0 standardu to više nije podržano jer se za dizajn mora koristiti **CSS**.

CSS, punim nazivom *Cascading Style Sheets*, jezik je za **stiliziranje prikaza sadržaja stranice**. Drugim riječima, HTML opisuje sadržaj koji će se prikazati, a CSS kako će izgledati.

Osim ljepšeg izgleda, CSS je bitan i za **responzivnost**, tj. on omogućava stranicama da budu prilagođene različitim veličinama zaslona. Računalni monitori, zasloni laptopa, zasloni mobitela i tableta dolaze u različitim veličinama i potrebno je korisniku omogućiti da na svakome može pregledavati stranicu bez poteškoća.

Slika 2.5 prikazuje prethodnu stranicu nakon dodavanja jednostavnog CSS stila.


PRVIM SAMOSTALNIM KONCERTOM "SAMOBORČEKI" PREDSTAVILI SVOJ PRVI CD

Raspjevani čarobnjaci

Pjesmom "Samobor, moj grad" započeo je 6. lipnja Dječji pjevački zbor "Samoborček" svoj prvi samostalni koncert. Zborašice Antonija Babojelić i Sara Čavlović svojom su recitacijom pokušale dočarati što sve Samoborčeki i s kim rade na probama....

Program prvog koncerta Dječjeg pjevačkog zbora "Samoborček" čarobirajući je vodio popularni i djeci drag glumac i voditelj Kristijan - Kiki Ugrina, a obogatili su ga gosti: klaunovi, Samoborske mažoretkinje, mađioničar Vladimir i pjesnik Enes Kišević, autor teksta pjesme "Ah" s kojom će Samoborčeki dogodine u siječnju u Split, na završnicu Dorice. ***Kristina Kirschenheuter***

Sara Čavlović i Antonija Babojelić predstavile su zbor "Samoborček" na početku koncerta

Slika 2.5 Stranica nakon dodavanja CSS stila

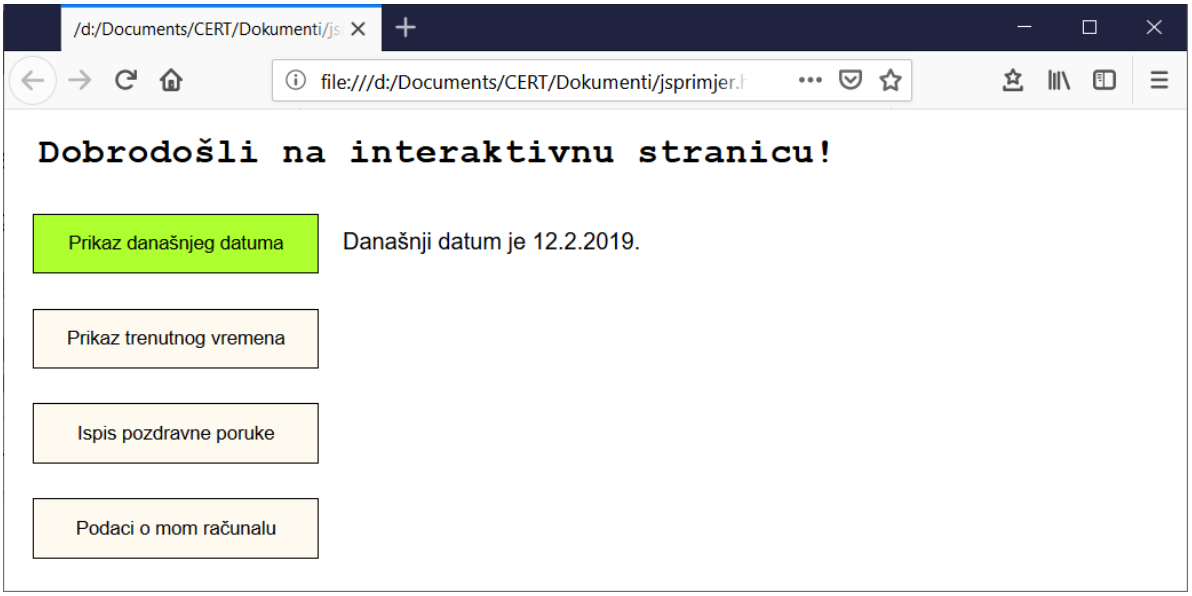
2.3 JavaScript

JavaScript je (skriptni) jezik koji se koristi za dodavanje interaktivnih elemenata web stranicama. Za razliku od HTML-a i CSS-a koji su statični, JavaScript **omogućuje dodavanje dinamičkog sadržaja.** Najčešće se koristi za interakciju s korisnikom, dodavanje novog ili mijenjanje postojećeg sadržaja, obradu podataka koje je korisnik unio u obrazac, promjenu izgleda stranice ili elemenata te reakciju na korisnikovu aktivnost.

Programe napisane JavaScriptom izvršava **web preglednik** (engl. *browser*) koji u sebi ima ugrađeni modul za JavaScript (engl. *JavaScript engine*). Danas je JavaScript podržan u svim široko korištenim web preglednicima i nema puno ograničenja što se sve JavaScriptom može postići unutar stranice u koju je ugrađen. Prema W3Techsu, **više od 95% svih postojećih stranica sadrži JavaScript kôd** (1).

JavaScript manipulira DOM-om (engl. *Document Object Model*), modelom za prikaz i interakciju s objektima u HTML dokumentu. Na taj način može mijenjati, brisati ili dodavati nove HTML elemente i atribute ili CSS stilove, reagirati na određene događaje i na korisnikovo ponašanje na stranici. Nastavak poglavlja kroz jednostavne primjere demonstrira kakve je funkcionalnosti moguće ostvariti JavaScriptom.

Na Slika 2.6, nakon što korisnik pritisne tipku „Prikaz današnjeg datuma“, JavaScript program će bez ponovnog učitavanja stranice, dakle bez razmjene podataka s udaljenim poslužiteljem, dodati novi, dinamički sadržaj s informacijom o datumu. Stranica se sad može smatrati interaktivnom – odgovara na korisnikove aktivnosti na stranici.



Slika 2.6 Prikaz današnjeg datuma JavaScriptom

Na Slika 2.7 prikazan je sadržaj HTML datoteke, a na Slika 2.8 JavaScript funkcija za prikaz datuma. Za prikaz datuma koriste se metode za dohvat trenutne godine, mjeseca i dana. JavaScript manipulacijom DOM-a mijenja strukturu HTML stranice tako da dodaje novi sadržaj na definirano mjesto. U postojeći HTML dokument dodaje se atribut *„onclick“* koji će obraditi događaj pritiska na tipku „Prikaz današnjeg datuma“. Obradit će ga tako što će pozvati funkciju *showDate()* koja zatim dohvaća i prikazuje datum.

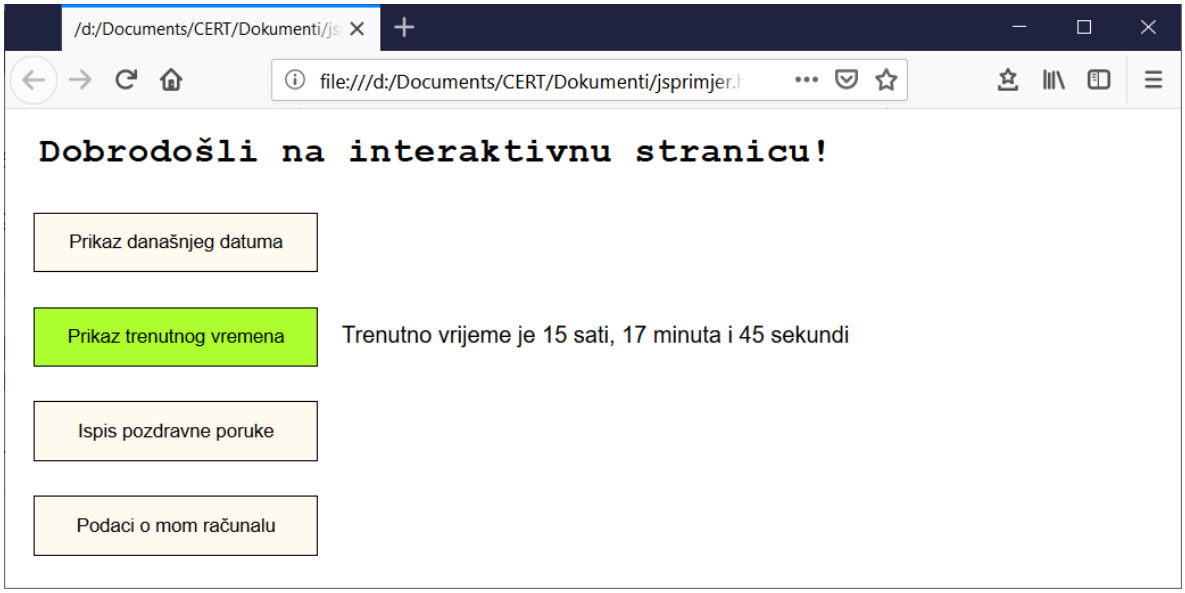
```html
<html>
    <head>
    <meta charset="utf-8">
    <link href="style.css" rel="stylesheet" type="text/css">
    <script src="javascript.js"></script>
    </head>
    <body>
        <h3>Dobrodošli na interaktivnu stranicu!</h3>

        <div>
            <button type="submit" onclick="showDate()">Prikaz današnjeg datuma</button>
            <p id="date"></p>
        </div>

    </body>
</html>
```

Slika 2.7 Sadržaj HTML datoteke

```javascript
function showDate() {
    var d = new Date();
    document.getElementById("date").innerHTML = "Današnji datum je " + d. getDate() + "." +
    d.getMonth() + "." + d.getFullYear() +".";
}
```

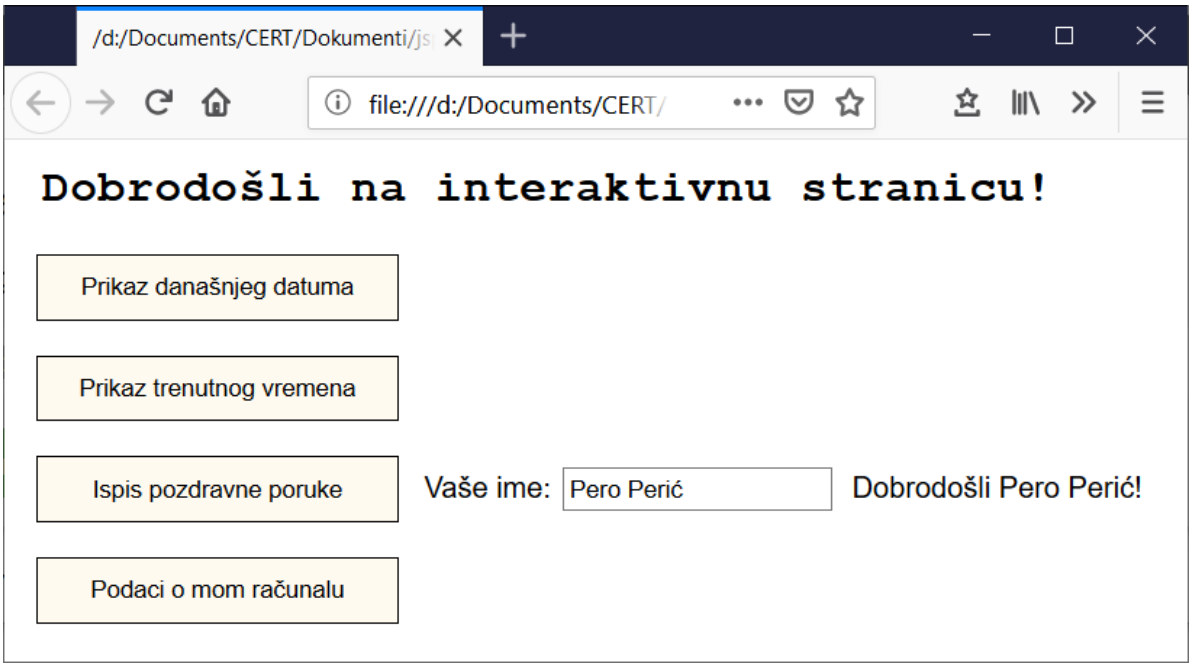
Slika 2.8 JavaScript funkcija za prikaz trenutnog datuma

Sličan primjer prikazan je i na Slika 2.9, ali ovaj se put prikazuje trenutno vrijeme u satima, minutama i sekundama. Korisno je primijetiti kako JavaScript točno zna u kojoj se vremenskoj zoni korisnik nalazi – jer se izvršava u web pregledniku koji ima pristup informaciji o vremenskoj zoni s operacijskog sustava korisnikovog računala.



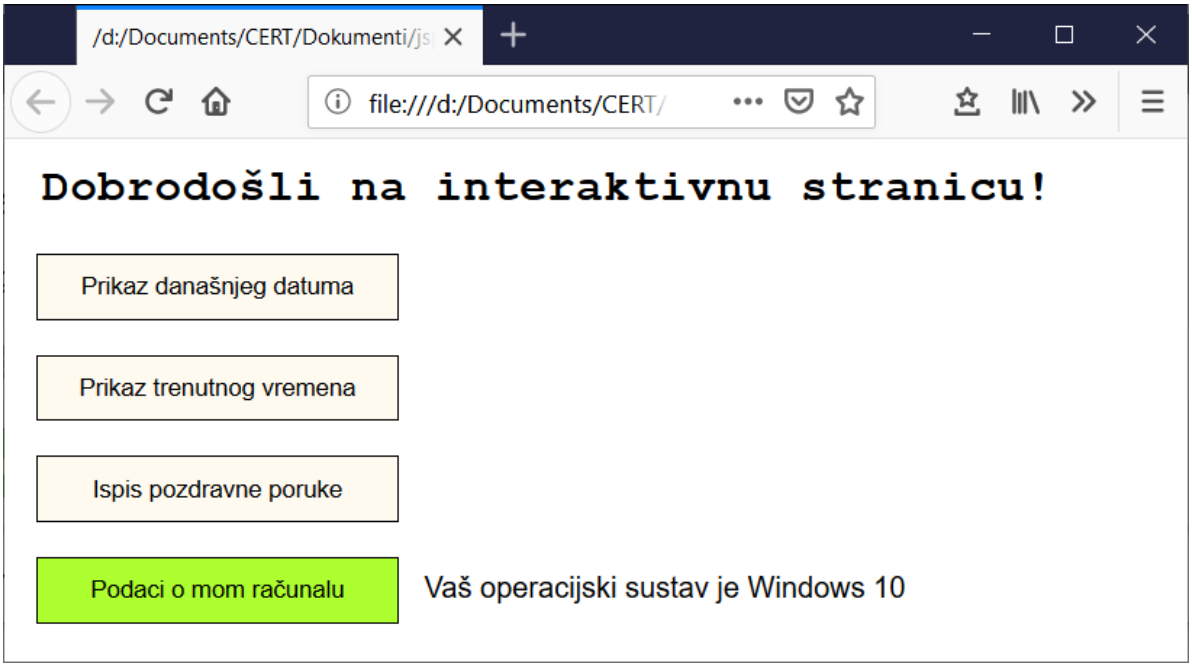
Slika 2.9 Prikaz trenutnog vremena JavaScriptom

Primjer sa Slika 2.10 prikazuje korištenje podataka koje je korisnik unio na web stranicu. Nakon što korisnik unese svoje ime i prezime, JavaScript ih obrađuje, provjerava jesu li uneseni ispravni znakovi i zatim ispisuje pozdravnu poruku ili upozorava korisnika da su uneseni podaci neispravni. Osim što može brzo reagirati na unos podataka, JavaScript može provjeriti (engl. *validate*) podatke iz HTML obrazaca prije no što ih pošalje poslužitelju – na taj način štedi se na vremenu ako se ispostavi da je korisnik unio neispravne podatke ili nije ispunio sva obvezna polja.



Slika 2.10 Korištenje podataka koje je korisnik unio

JavaScript može prikupiti i različite informacije o korisnikovom operacijskom sustavu, web pregledniku, postavkama kolačića i slično. Takvo prikupljanje podataka o korisniku naziva se *fingerprinting*. Primjer kako web stranica JavaScriptom otkriva inačicu operacijskog sustava korisnikovog računala prikazana je na slici 2.11.



Slika 2.11 Otkrivanje operacijskog sustava računala JavaScriptom

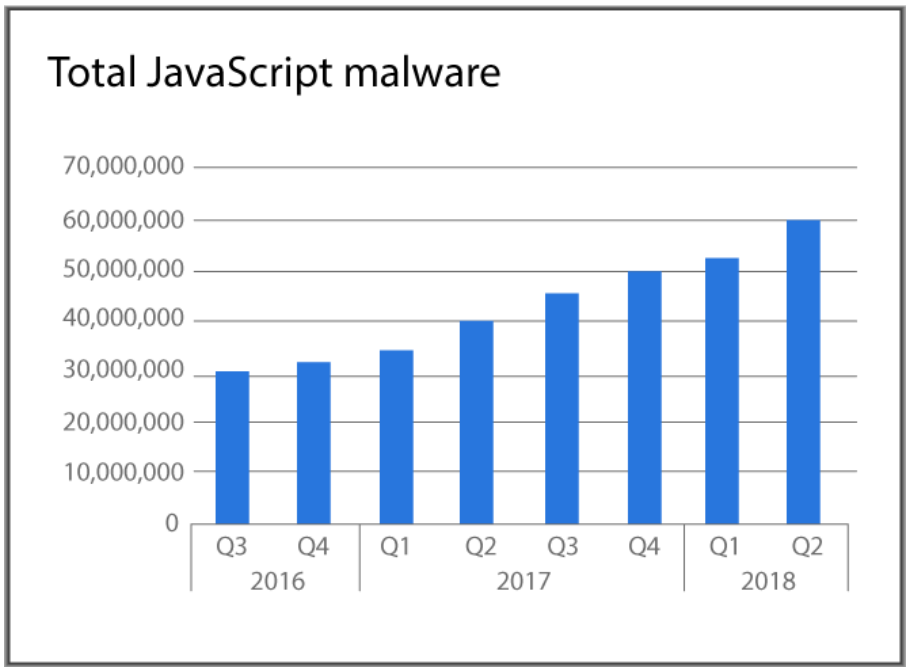
3 Prijetnje na webu koje se oslanjaju na JavaScript kôd

JavaScriptom je moguće ostvariti razne funkcionalnosti **unutar web preglednika** i to **unutar jedne web stranice** (unutar one web stranice na kojoj se nalazi izvršeni kôd). Iz sigurnosnih razloga, nije dopušteno da primjerice JavaScript kôd na nekoj web stranici ugasi korisnikovo računalo ili pristupi korisnikovim podacima na nekoj drugoj web stranici. Upravo kako bi utjecaj JavaScript kôda zaista bio ograničen samo na web stranicu na kojoj se on nalazi, postoje zaštitni mehanizmi koji ograničavaju JavaScript kôd:

- u interakciji s računalom korisnika, npr. kod pristupanja lokalnom datotečnom sustavu ili sklopovlju (kamera, mikrofon...),
- u interakciji s drugim web stranicama, npr. kod čitanja podataka s drugih web stranica.

Neki od tih zaštitnih mehanizma su dio web preglednika, a neki su dio web stranica. No web preglednici i web stranice nisu savršeni, tako da **postoje ranjivosti** („rupe u sigurnosti“) koje se mogu **iskoristiti** kako bi se navedena ograničenja zaobišla i time **ugrozila sigurnost korisnika**.

Na Slika 3.1 prikazano je istraživanje tvrtke *Safety Detective* koje prikazuje stalni porast upotrebe zlonamjernog softvera koji se oslanja upravo na JavaScript (3).



***Slika 3.1 Porast uporabe zlonamjernog softvera koji se oslanja na JavaScript* (3)**

U nastavku ovog poglavlja opisane su najraširenije kategorije prijetnji na webu koje se oslanjaju upravo na zlouporabu JavaScript kôda.

3.1 Iskorištavanje ranjivosti web preglednika i njihovih dodataka

Ova kategorija prijetnji uključuje izravno iskorištavanje ranjivosti u web pregledniku ili u nekom dodatku web preglednika kao što je npr. *Adobe Flash Player*. Scenarij napada može izgledati ovako:

- napadač pripremi web stranicu sa zlonamjernim JavaScript kôdom i popratnim sadržajem,
- žrtva posjeti tu web stranicu,
- web preglednik žrtve (npr. *Internet Explorer*, *Mozilla Firefox* ili *Google Chrome*) otvara i pokreće, odnosno obrađuje tu stranicu: preuzme sav sadržaj, pokrene JavaScript kôd ili *Adobe Flash* animaciju...
- u tom postupku obrade stranice iskorištava se ranjivost u web pregledniku, zbog čega se u konačnici na računalu žrtve (izvan bilo kakvih ograničenja web preglednika) pokreće napadačev zlonamjerni kôd.

U konačnici, rezultat ovog napada (pokretanje zlonamjernog kôda na računalu žrtve) znači da je napadač zarazio računalo žrtve nekim oblikom zlonamjernog softvera, npr. *ransomwareom* ili *spywareom*.

Jedan stvarni primjer pokušaja ovakvog napada odvio se u lipnju 2019. godine – nepoznati je napadač pokušao zaraziti zaposlenike burze kriptovaluta *Coinbase* na sljedeći način (4):

- napadač je zaposlenicima *Coinbasea* poslao *spearphishing* poruku e-pošte kako bi ih naveo da posjete njegovu web stranicu,
- na napadačevoj web stranici se nalazio zlonamjerni JavaScript kôd koji iskorištava ranjivost u web pregledniku *Mozilla Firefox* i u konačnici instalira zlonamjerni softver na računalo žrtve.

Srećom, zaposlenici nisu nasjeli na napadačevu *spearphishing* poruku (objašnjeno u dokumentu Nacionalnog CERT-a: „Phishing“), no da jesu, vrlo je vjerojatno da bi napadač uspio zaraziti njihova računala i pristupiti kriptovalutama u posjedu burze. U ovom je slučaju korištena javno nepoznata ranjivost (tzv. *zero-day vulnerability*) u web pregledniku *Mozilla Firefox*, no općenito su puno češći napadi u kojima se koristi javno poznata ranjivost koju napadač može zloupotrijebiti samo u starijim inačicama web preglednika.

Općenito, u ovakvim scenarijima napada, ranjivost može biti u JavaScript modulu web preglednika (dijelu preglednika koji izvršava JavaScript kôd) ili može biti i u nekom drugom dijelu (npr. u dijelu zaduženom za CSS ili u dodatku kao što je *Adobe Flash Player*). No neovisno o tome gdje se ranjivost točno nalazi, izvršavanje JavaScript kôda u web pregledniku žrtve obično je neophodan dio napada. Drugim riječima, bez izvršavanja JavaScript kôda u web pregledniku žrtve većina ovakvih napada uopće nije moguća.

3.2 Prikupljanje informacija o korisniku (engl. *fingerprinting*)

Ova kategorija zapravo obuhvaća dvije usko povezane vrste prijetnje:

- prikupljanje informacija o korisniku kako bi se ugrozila njegova privatnost (npr. u svrhe marketinga),
- prikupljanje informacija o softveru korisnika (inačice web preglednika, dodataka...) kako bi se uspješno pripremio napad na njega, npr. iskorištavanje ranjivosti web preglednika.

S tehničke strane, oba navedena slučaja izgledaju slično – JavaScript kôdom (i nekim drugim tehnikama) web stranice prikupljaju podatke o korisniku kao što su: vrsta i inačica operacijskog sustava, inačica web preglednika, instalirani dodaci i njihove inačice, podržane funkcionalnosti (npr. WebGL) itd. Ti se podaci zatim mogu kombinirati kako bi se sastavio jedinstveni „otisak prsta“ (engl. *fingerprint*) korisnika ili kako bi se pripremio napad na ranjive komponente web preglednika.

Više detalja o ovoj prijetnji u kontekstu ugroza privatnosti korisnika dostupno je u prethodnom dokumentu Nacionalnog CERT-a: „Osnove privatnosti na Internetu“.

3.3 *Exploit kits*

Na prethodnoj stranici, u potpoglavlju *„Iskorištavanje ranjivosti web preglednika i njihovih dodataka“*, opisan je scenarij napada u kojem napadač iskorištava točno određenu ranjivost u web pregledniku, odnosno u dodatku web preglednika. No da bi taj napad bio uspješan, žrtva mora koristiti upravo tu ranjivu inačicu web preglednika odnosno dodatka. Primjerice, napadač može pripremiti zlonamjernu web stranicu koja će uspješno napasti žrtve koje koriste web preglednik *Mozilla Firefox* inačicu 66.0.5 ili neku stariju inačicu. No, ako žrtve koriste neki drugi web preglednik (npr. *Google Chrome*) ili noviju inačicu *Mozilla Firefoxa*, onda napad neće uspjeti.

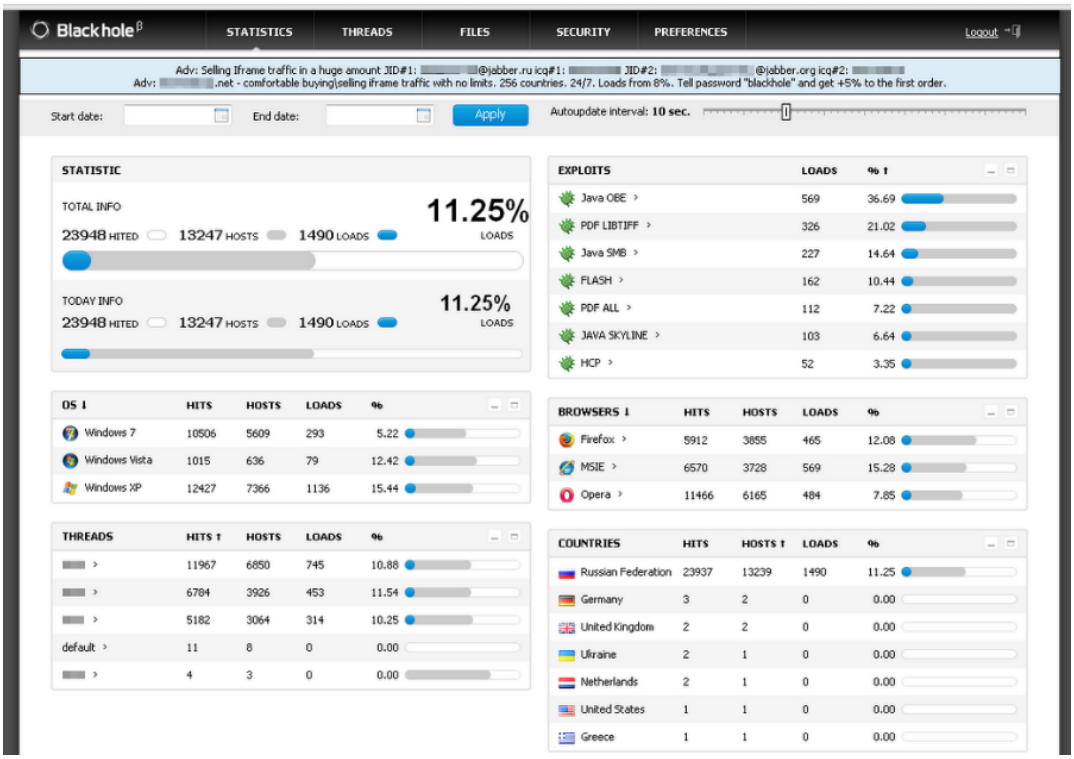
Kako bi ovakva vrsta napada bila učinkovitija, napadači su razvili takozvane *exploit kitove*. *Exploit kit* je sustav koji automatizira dvije prethodno opisane prijetnje: prikupljanje informacija o korisniku i iskorištavanje ranjivosti. Drugim riječima, e*xploit kit* je alat koji:

1) Automatski prepoznaje koji softver žrtva koristi te sadržava li taj softver javno poznate ranjivosti. Primjerice, *exploit kit* može prepoznati da žrtva koristi staru inačicu preglednika *Internet Explorer* i staru inačicu dodatka *Adobe Flash Player*.

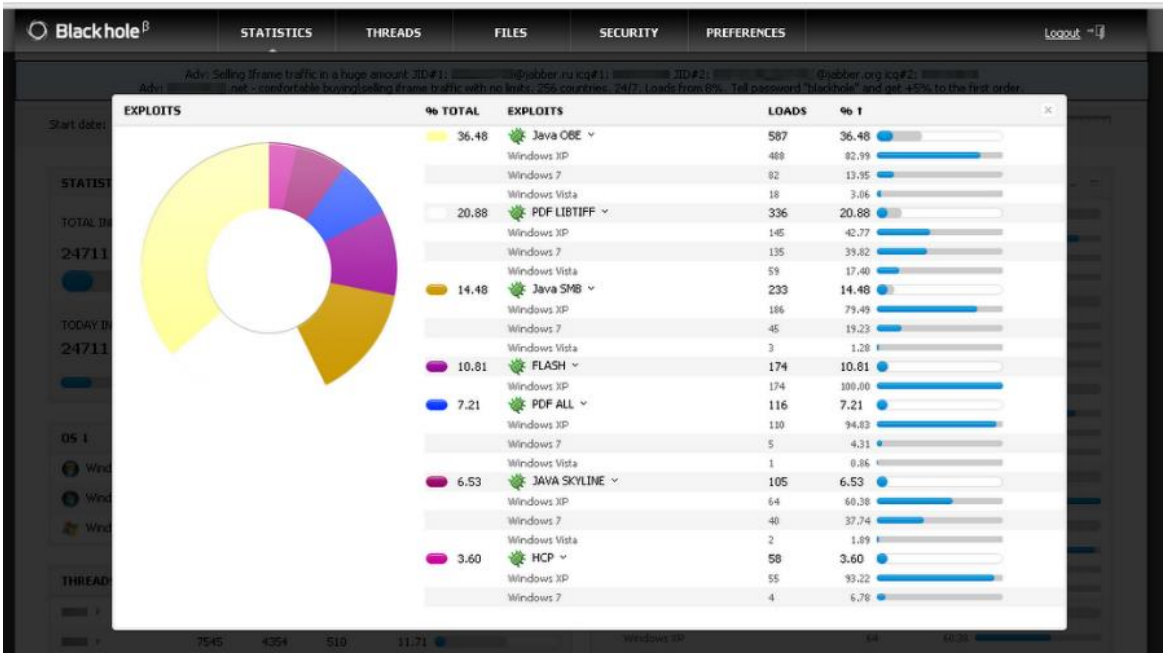
2) Zatim, na temelju prikupljenih informacija, *exploit kit* pokreće napad koji iskorištava upravo one ranjivosti u žrtvinom web pregledniku ili dodatku web preglednika za koje je detektirao da su prisutne.

Oba koraka napada oslanjaju se na korištenje JavaScript kôda. Bez JavaScripta napadačeva web stranica može prikupiti samo osnovne informacije o softveru korisnika, a samo iskorištavanje ranjivosti bez JavaScripta obično uopće nije moguće.

Jedan razlog raširenosti *exploit kitova* je to što za njihovo korištenje nije potrebno nikakvo posebno znanje ni stručnost – kriminalci često jednostavno kupe pristup *exploit kitu* (od drugih kriminalaca), a u paketu s *exploit kitom* dolazi i pregledno korisničko sučelje preko kojega napadač jednostavno pokrene napad i prati uspješnost svoje kampanje, tj. na koliko je korisničkih računala uspio proširiti zlonamjerni softver. Na slikama 3.2 i 3.3 prikazano je korisničko sučelje *exploit kita* Blackhole inačice 1.2.0 koja je bila aktivna 2011. godine.



***Slika 3.2 Korisničko sučelje exploit kita Blackhole – glavni pregled statistika* (5)**



***Slika 3.3 Korisničko sučelje exploit kita Blackhole – pregled statistika korištenih exploita* (5)**

Za napad *exploit kitom* žrtva ipak mora posjetiti neku web stranicu na kojoj se nalazi napadačev zlonamjerni kôd. Uz to, sama činjenica da je žrtva posjetila web stranicu ne znači da će napad uspjeti – napad će uspjeti samo ako žrtva koristi određene ranjive inačice softvera koje *exploit kit* može iskoristiti.

Upravo zato, za uspješne napade *exploit kitom*, napadači žele usmjeriti veliki broj korisnika na stranicu s njihovim zlonamjernim kôdom kako bi u konačnici neki od tih korisnika (oni s ranjivim softverom) bili uspješno napadnuti.

Napadači to obično postižu kompromitacijom popularnih web stranica ili tzv. zlonamjernim oglašavanjem (engl. *malvertising*). Scenarij napada zlonamjernim oglašavanjem i *exploit kitom* izgleda ovako:

1) Žrtva posjeti neku legitimnu web stranicu, primjerice neki portal s vijestima.

2) Ta web stranica sadržava oglase – neki od tih oglasa ostavili su kriminalci i taj oglas sadrži napadačev zlonamjerni kôd što ni vlasnici web stranice ni korisnici stranica ne znaju.

3) Samim otvaranjem web stranice, žrtvin web preglednik učitava zlonamjerni kôd iz oglasa i u pozadini (bez da korisnik primijeti) otvara stranicu *exploit kita*.

4) Ako žrtva ima odgovarajući ranjivi softver (a od velikog broja posjetitelja, neki će imati ranjivi softver), *exploit kit* će uspješno napasti žrtvu, tj. uspješno će instalirati zlonamjerni softver na računalo žrtve.

U cijelom ovom procesu, zabrinjavajuće je to što, osim korištenja ranjivog softvera, korisnik zaista nije napravio ništa krivo – nije posjetio neku sumnjivu stranicu (već samo primjerice uobičajeni portal s vijestima), nije preuzimao i pokretao nikakav softver, nije čak ni kliknuo na oglase, a ipak je zarazio svoje računalo.

3.4 *Cross-site scripting* (XSS)

Cross-site scripting (skraćeno: XSS) je prilično raširena ranjivost u web stranicama koja se svodi na to da napadač može ubaciti svoj (zlonamjerni) JavaScript kôd na tuđu (ranjivu) web stranicu. O ovoj naizgled jednostavnoj ranjivosti mogao bi se napisati cijeli zaseban dokument koji opisuje:

1) kakve sve vrste XSS ranjivosti postoje, tj. gdje je sve moguće pronaći XSS ranjivosti

2) kako se zaštititi od XSS ranjivosti (organizacija OWASP samo na tu temu ima dva velika dokumenta (6) (7))

3) što sve napadač može postići na temelju XSS ranjivosti

XSS neće biti detaljno opisan u sklopu ovog dokumenta. U ovom kontekstu, vezano za XSS najbitnije je znati sljedeće:

1) XSS je ranjivost u web stranici – drugim riječima, XSS ranjivost će postojati zbog greške u kôdu koju su napravili programeri web stranice, a ne zbog greške korisnika

2) iz perspektive programera, zaštititi se od XSS-a, tj. izraditi neku složeniju web stranicu tako da uopće nema XSS ranjivosti, je prilično teško – zato su XSS ranjivosti relativno česte (8)

3) zbog XSS ranjivosti, napadač može izvršiti svoj zlonamjerni JavaScript kôd u pregledniku žrtve **i u kontekstu ranjive web stranice**

U konačnici, dva česta scenarija napada koji se oslanjaju na XSS ranjivost izgledaju ovako (8) (9):

1) Preuzimanje korisničkog računa, primjerice na društvenoj mreže *Facebook*:

   - pretpostavimo da web stranica društvene mreže *Facebook* sadrži XSS ranjivost, te da tu XSS ranjivost napadač iskoristi tako da ubaci svoj JavaScript kôd na web stranicu *Facebooka*

   - jednom kada žrtva (korisnik) otvori *Facebook*, otvorit će se web stranica i pokrenut će se napadačev JavaScript kôd

   - napadačev JavaScript kôd u tom trenutku ima potpuni pristup web stranici iz perspektive žrtve – drugim riječima, preko svog zlonamjernog JavaScript kôda, napadač sada efektivno može raditi na *Facebooku* sve što može raditi i žrtva: čitati poruke žrtve, pristupiti njenim privatnim zapisima, objaviti statuse i fotografije u njeno ime...

2) Napad *exploit kitom*:

   - zbog XSS ranjivosti, napadač ubaci vlastiti JavaScript kôd na legitimnu web stranicu – u ovom slučaju, taj kôd je napravljen tako da (u pozadini) otvara stranicu *exploit kita*

   - napad se dalje odvija kao što je prethodno opisano u potpoglavlju o *exploit kitovima*:

     - korisnik posjeti legitimnu (napadnutu) web stranicu,

     - na njoj se nalazi kôd zbog kojega se u pozadini otvara stranica *exploit kita*

     - te ako korisnik koristi odgovarajući ranjivi preglednik/dodatak, instalira se napadačev zlonamjerni softver na njegovo računalo

U oba navedena scenarija, bez izvršavanja JavaScript kôda u korisnikovom web pregledniku napad neće biti uspješan.

4 Zaštita od zlonamjernog JavaScript kôda

Na temelju prošlog poglavlja, lako je pomisliti kako bi korisnicima bilo bolje kada bi u potpunosti isključili izvršavanje JavaScript kôda unutar svog web preglednika. To je moguće napraviti kroz postavke preglednika. Time bi korisnici bili sigurniji, no to je zaista ekstremno rješenje, jer kao što je i opisano u uvodu dokumenta, velika većina web stranica koristi JavaScript za legitimne svrhe kako bi korisnicima pružila interaktivne funkcionalnosti i intuitivnije sučelje. Korisnici koji bi isključili izvršavanje JavaScripta ne bi uopće mogli koristiti veliki dio današnjih web stranica.

Općenito, korisnici svakako trebaju:

- redovito ažurirati softver,
- ukloniti nepotreban softver koji je česta meta napada (npr. *Adobe Flash Player*),
- koristiti softver za zaštitu (*anti-virus/anti-malware*)
- i općenito biti oprezni prilikom korištenja računala:
    - ne otvarati poveznice iz poruka e-pošte bez razmišljanja,
    - ne kliktati na sumnjive oglase,
    - paziti koji se softver i koji dodaci web pregledniku koriste...

No, fokus ovog poglavlja će biti na specifičnim rješenjima koja korisnicima mogu omogućiti normalno pregledavanje weba, ali ujedno i blokiranje zlonamjernog JavaScript kôda. Konkretnije, ovo poglavlje će opisati tri različita dodatka za web preglednike koji mogu zaštititi korisnika od zlonamjernog JavaScript kôda prilikom pregledavanja weba:

- *uBlock Origin* – široko poznat kao dodatak za blokiranje oglasa (engl. *ad-blocker*), no ujedno do neke mjere štiti korisnike od prijetnji na webu
- *NoScript* – dodatak koji omogućava selektivno blokiranje JavaScript kôda ovisno o izvoru, namijenjen je naprednim korisnicima, ali zato može pružiti snažnu zaštitu od prijetnji na webu
- *uMatrix* – „veliki brat“ dodatka *uBlock Origin*, sličan je *NoScriptu*, no ima drugačije sučelje i širu podršku za web preglednike

4.1 *uBlock Origin*

Mnogi korisnici računala već koriste neki od dodataka web pregledniku namijenjen blokiranju oglasa (engl. *ad-blocker*). Takvi se dodaci najčešće koriste upravo zato što korisnici žele ukloniti oglase koji ih odvraćaju ili ih čak gotovo i sprječavaju u pregledavanju weba. No manje je poznato da neki od takvih dodataka mogu u nekoj mjeri i zaštititi korisnika prilikom pregledavanja weba.

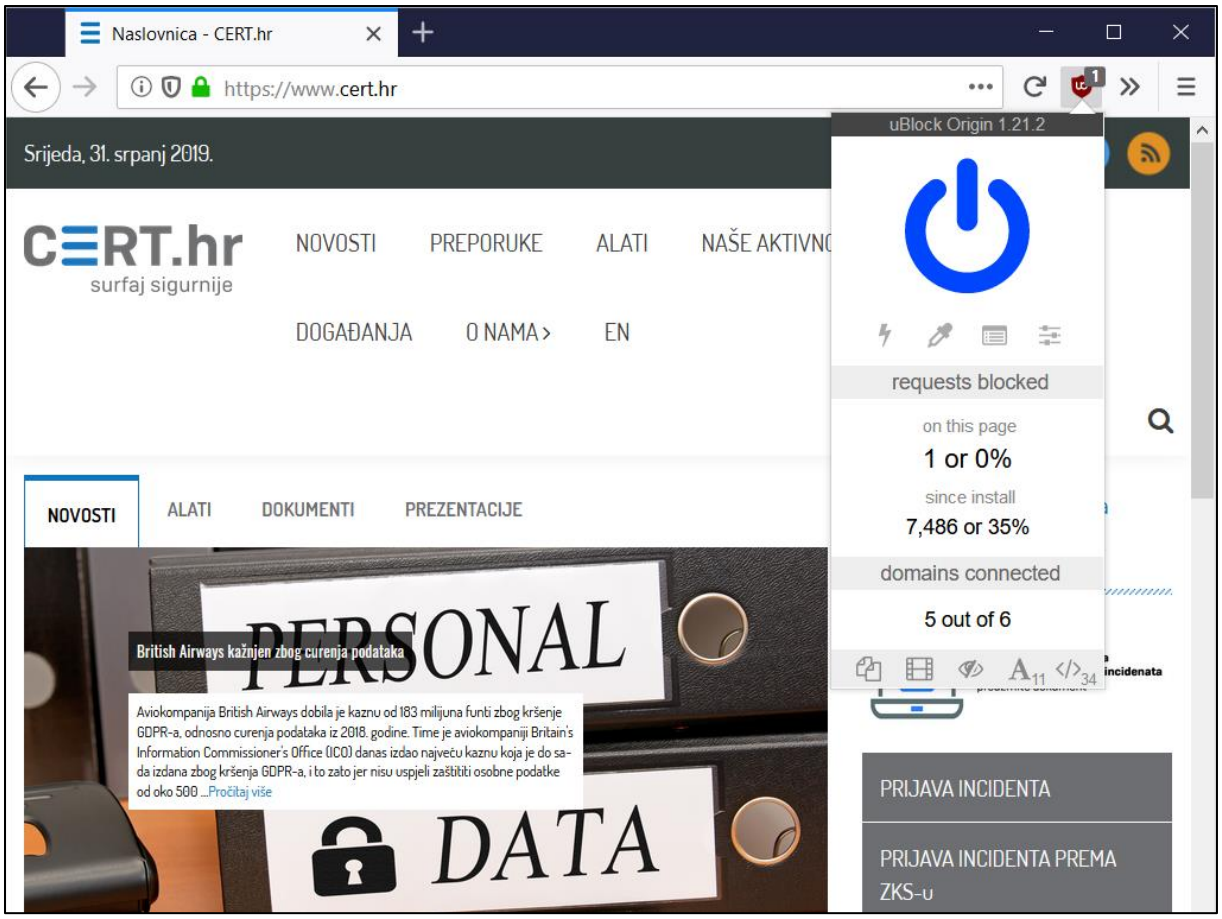
Jedan od takvih dodataka je i *uBlock Origin*, slobodan (engl. *free and open source*) dodatak dostupan za sve široko korištene web preglednike: *Google Chrome* (i ostali preglednici temeljeni na *Chromiumu*), *Mozilla Firefox* (za računala i za pametne telefone), *Microsoft Edge* i *Opera*. Iako se *uBlock Origin* na prvi pogled čini samo kao dodatak za blokiranje oglasa, on nakon standardne instalacije blokira i razne domene za koje se zna da prate korisnike i domene vezane za zlonamjerni softver.

uBlock Origin i slični dodaci web preglednicima čine korisnika sigurnijim na dva načina:

1) kako *uBlock Origin* blokira oglase na webu, tako blokira i napade zlonamjernim oglašavanjem (engl. *malvertising*)

2) osim blokiranja oglasa, *uBlock Origin* blokira pristup domenama na kojima se nalazi zlonamjerni kôd što će općenito spriječiti neke od napada putem web stranica (ne samo napade zlonamjernim oglašavanjem)

Instalacija i korištenje *uBlock Origina* je izrazito jednostavna – dovoljno ga je instalirati sa službene web stranice za dodatke odgovarajućeg preglednika (Mozilla Firefox, Google Chrome, Opera) i od tada će u pozadini automatski blokirati pristup neželjenim resursima. Na slici 4.1 prikazano je sučelje *uBlock Origina* – kroz njega je moguće isključiti blokiranje na određenoj web stranici ili pregledati statistiku blokiranja, no u većini slučajeva to ipak nije potrebno, već je dovoljno samo instalirati *uBlock Origin* i pustiti ga da radi svoj posao u pozadini.



Slika 4.1 – otvoreno sučelje dodatka uBlock Origin

uBlock Origin je jednostavan za korištenje, troši zanemarivu količinu resursa na računalu (obično čak i ubrzava pregledavanje weba zbog toga što blokira dohvaćanje neželjenih resursa) i čini korisnika sigurnijim. Korištenje *uBlock Origina* se preporuča svakom korisniku koji želi na jednostavan način podići razinu sigurnosti prilikom pregledavanja weba. No uzimajući u obzir sve prijetnje s kojima se korisnici mogu susreti na webu, za mnoge *uBlock Origin* ipak nije dovoljno snažna mjera zaštite – zato postoje dodaci kao što su *NoScript* i *uMatrix*.

4.2 *NoScript*

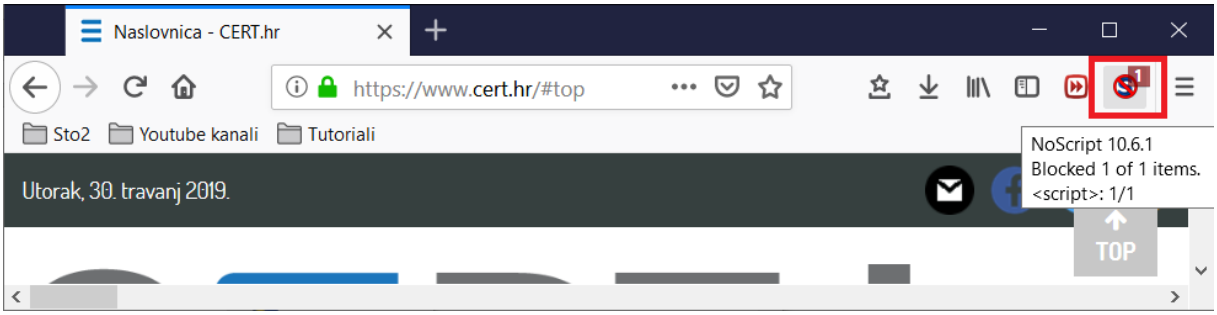
NoScript je jedan od najšire korištenih dodatak web pregledniku za naprednu zaštitu od prijetnji na webu. Vrijednost *NoScripta* su prepoznali brojni sigurnosni stručnjaci (10), te je korištenje *NoScripta* jedan od čestih savjeta naprednim korisnicima računala koji traže dodatne mjere zaštite prilikom pregledavanja weba. *NoScript* je slobodan softver (engl. *free and open source software*), a razvio ga je Giorgio Maone koji je između ostaloga i član organizacija *Mozilla Security Group* te *W3C Web Application Security Working Group*.

NoScript je dostupan za *Mozilla Firefox* te od nedavno i za *Google Chrome* kroz službene web stranice za dodatke web preglednicima:

- Firefox Add-ons
- Chrome Web Store

Nakon instalacije *NoScripta*, JavaScript, Java, Flash, Silverlight i slične funkcionalnosti bit će **automatski blokirane**. Prilikom pregledavanja weba, zbog korištenja *NoScripta*, korisnik **mora izričito dopustiti izvršavanje skripti** (npr. JavaScript kôda) **iz svakog pojedinog izvora**. Dozvola za izvršavanje skripti koju korisnik daje može biti privremena ili trajna (kako korisnik ne bi svaki put ispočetka morao odobravati određene izvore). Osim blokiranja JavaScript kôda i drugih funkcionalnosti na temelju izvorne domene, *NoScript* pruža i zaštitu protiv XSS i *Clickjacking* napada koja funkcionira čak i kada je izvršavanje JavaScript kôda dozvoljeno.

Kao što je prikazano na slici 4.2, *NoScript* se u pregledniku prikazuje kao ikona na alatnoj traci i na svakoj web stranici pokazuje je li pokretanje skripti blokirano, dopušteno ili djelomično dopušteno.

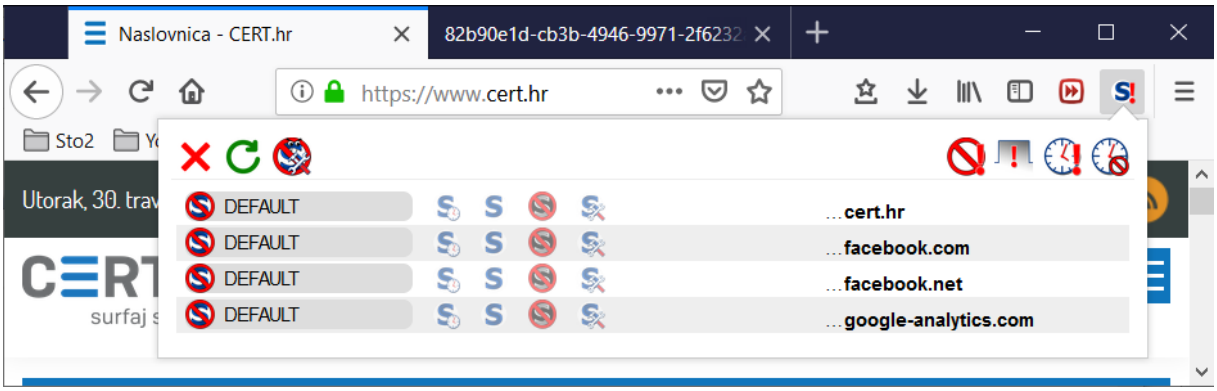


Slika 4.2 NoScript ikona u web pregledniku

Prema zadanim (engl. *default*) postavkama, izvršavanje skripti na web stranicama je blokirano. Blokiranje skripti će spriječiti potencijalne napade, ali će isto tako često spriječiti korisnika u korištenju nekih legitimnih funkcionalnosti. Ako korisnik vjeruje stranici i ipak želi omogućiti pokretanje nekih skripti, to može učiniti na dva načina:

1) desni klik bilo gdje na stranici i zatim lijevi klik na *NoScript*
2) lijevi klik na *NoScript* ikonu na alatnoj traci

Zatim će se otvoriti *NoScript* sučelje koji izgleda kao što je prikazano na slici 4.3.



Slika 4.3 NoScript sučelje

U prvom se redu nalaze sljedeće ikone:

1. – zatvori izbornik
2. – ponovno učitaj stranicu
3. – dodatne postavke
4. – omogući blokiranje skripti na svim stranicama
5. – ukini sve zabrane za stranicu u ovom *tabu*
6. – označi sve blokirane izvore na stranici privremeno pouzdanima
7. – prestani smatrati pouzdanima izvore označene kao „privremeno pouzdane“

Osim ikona, sučelje prikazuje i popis izvora (domena) s kojih trenutna web stranica pokušava učitati resurse. Ako se pored pouzdane domene nalazi i simbol lokota, to označava da veza koristi HTTPS.

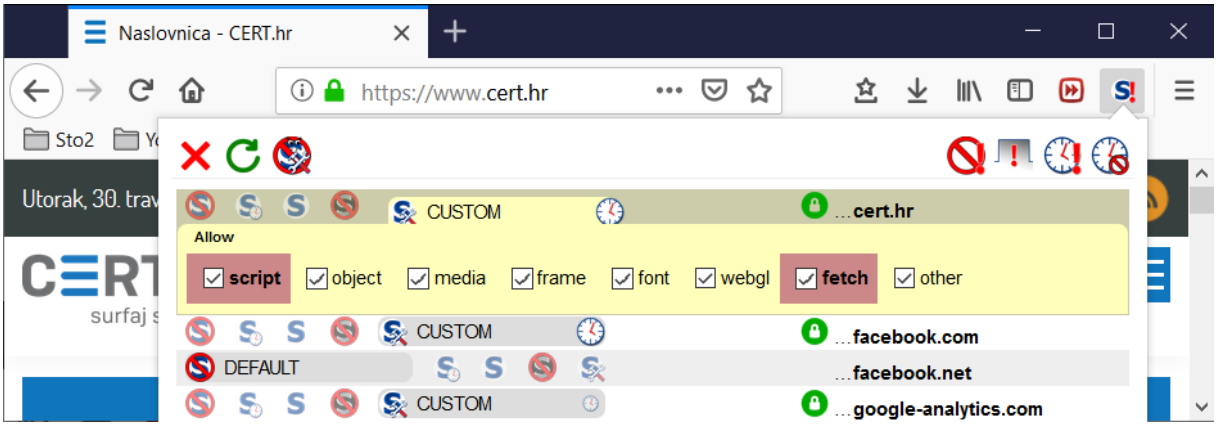
Pored svake domene prikazana je i razina povjerenja koja može biti:

- podrazumijevana (engl. *default*), što znači da će skripte biti blokirane (osim ako nije drugačije odabrano u postavkama)
- privremeno pouzdana (engl. *temporarily trusted*)
- pouzdana (engl. *trusted*)
- nepouzdana (engl *untrused*)
- prilagođena (engl. *custom*)

***NoScript* neće dozvoliti izvršavanje nijedne skripte koja dolazi s nepouzdanog izvora (domene).** Ako je izvor privremeno pouzdan, skripte se smiju izvršavati **samo za vrijeme trajanja sjednice** (engl. *session*) i kad korisnik idući put posjeti stranicu, ona će biti označena kao nepouzdana. Kad je neki izvor skripti, tj. domena, označen kao pouzdan, *NoScript* će i na drugim stranicama dopustiti izvršavanje skripti ako dolaze iz tog izvora. Isto vrijedi i za nepouzdane izvore – skripta iz izvora označenog kao nepouzdan neće se izvršavati ni na jednoj stranici.

Treba biti svjestan da skripte uobičajeno znaju pozivati druge skripte iz trećih izvora. U tom slučaju, takvi treći izvori se neće prikazati u listi izvora sve dok se ne dopusti izvršavanje skripte koja bi ih pozvala.

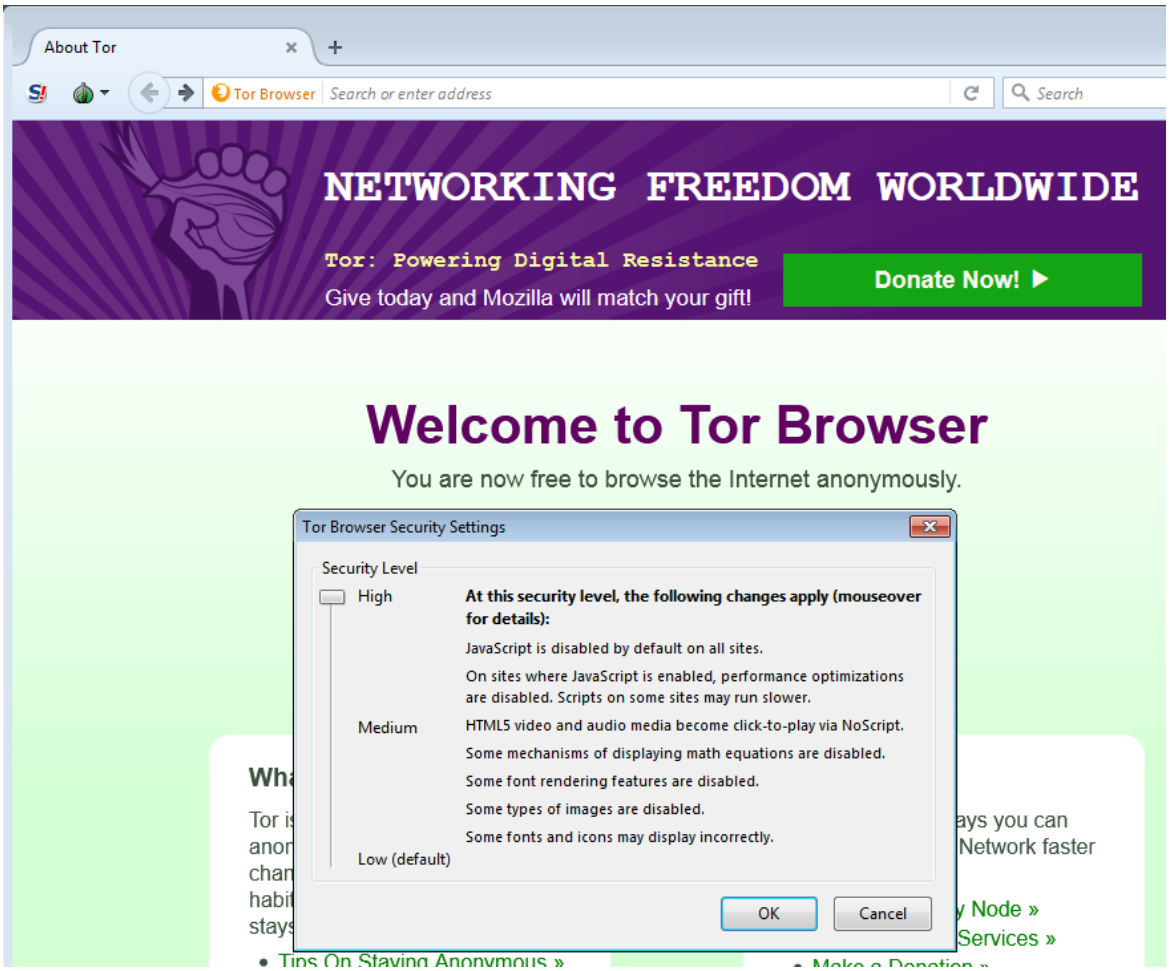
Prilagođena (engl. *custom*) razina povjerenja prikazana je na slici 4.4. Pomoću nje je moguće preciznije odrediti koji se resursi smiju učitati s određenog izvora.



Slika 4.4 Prilagođena razina povjerenja u NoScriptu

Glavni nedostatak *NoScripta* je to što on korisniku zaista otežava pregledavanje weba – dok će običan korisnik koji ne koristi ovakav dodatak jednostavno otvoriti neku web stranicu i odmah pregledati sadržaj koji ga zanima, za korisnika *NoScripta* je uobičajeno da kod otvaranja svake nove web stranice prvo mora selektivno odobriti dohvaćanje resursa i izvršavanje skripta s domena kojima vjeruje. No unatoč tome, brojni napredni korisnici su ipak spremni uložiti trud kako bi zaista znatno povisili razinu svoje sigurnosti i privatnosti na ovaj način prilikom pregledavanja weba.

Zanimljivo je znati i da je *NoScript* integriran u *Tor Browser* – web preglednik s fokusom na privatnost i anonimnost koji usmjerava sav promet preko mreže anonimnosti *Tor*. Na slici 4.5 je prikazano sučelje za podešavanje razine zaštitnih mjera u Tor *Browseru*. Interno, podizanje razine sigurnosnih mjera aktivirat će neke funkcionalnosti ugrađenog *NoScripta*. Više informacija o *Tor Browseru* i *Toru* dostupno je u prethodnim dokumentima Nacionalnog CERT-a: „Tor Browser“, „Tor mreža – tehnička pozadina i napredno korištenje“.



Slika 4.5 Integrirane funkcionalnosti dodatka NoScript u Tor Browseru

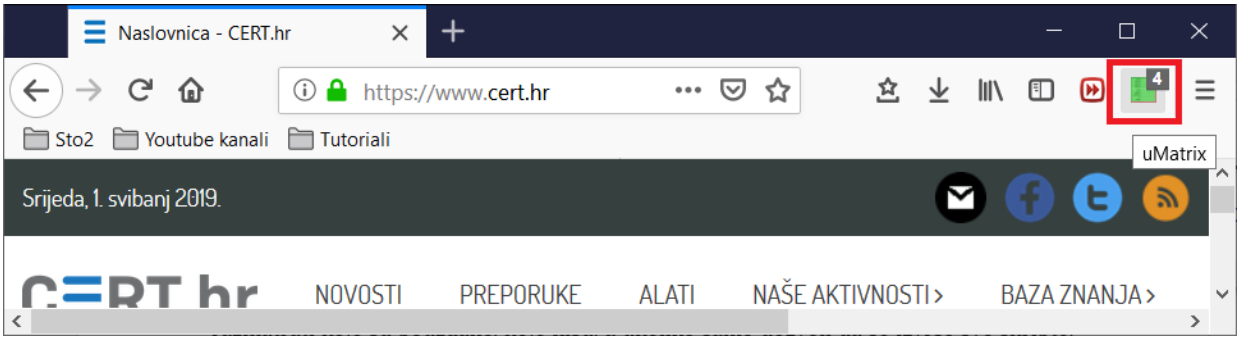
4.3 *uMatrix*

uMatrix je slobodan (engl. *free and open source*) dodatak web pregledniku koji korisniku pruža kontrolu nad resursima koji se učitavaju prilikom posjeta web stranicama. Ti resursi uključuju i JavaScript skripte, zbog čega je *uMatrixom* moguće i blokirati izvršavanje JavaScript kôda. *uMatrix* je izradio isti autor kao i dodatak *uBlock Origin*, no sličniji je *NoScriptu* – namijenjen je naprednijim korisnicima koji su spremni uložiti određenu količinu truda kako bi povisili razinu svoje sigurnosti i privatnosti prilikom pregledavanja weba.

uMatrix je dostupan za preglednike *Mozilla Firefox*, *Google Chrome* i *Opera* preko službenih stranica za dodatke:

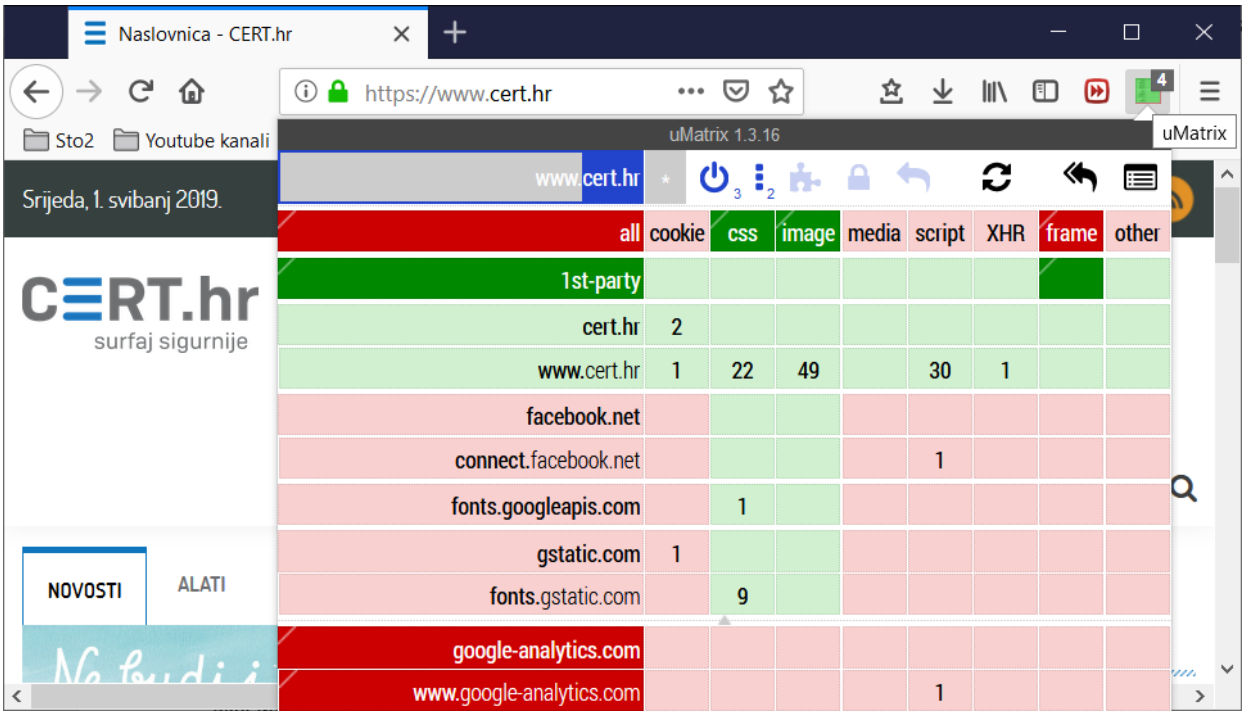
- Firefox Add-ons
- Chrome Web Store
- Opera add-ons

Kao što je prikazano na slici 4.6, nakon instalacije *uMatrix* se u pregledniku prikazuje kao ikona na alatnoj traci.



Slika 4.6 uMatrix ikona u web pregledniku

Klikom na ikonu otvara se sučelje prikazano na slici 4.7.



Slika 4.7 Sučelje uMatrixa

Gledajući sučelje, jasno je kako je *uMatrix* dobio ime – većinu sučelja zauzima matrica u čijim su recima popisani svi izvori (domene) s kojih otvorena stranica pokušava učitati resurse, a u stupcima su vrste elemenata koji se pokušavaju učitati. U ćelijama tablice se dopušta ili zabranjuje učitavanje određene vrste resursa iz nekog izvora.

Sadržaj kojemu je dozvoljeno učitavanje označen je **zelenom** bojom, a sadržaj koji je blokiran **crvenom** bojom. Kako bi se neki sadržaj dopustio, treba kliknuti na gornju polovicu ćelije, a kako bi se zabranio na donju.

Razlikuju se tamne i svijetle nijanse zelene i crvene boje, pri čemu svijetle nijanse znače da zabrana, odnosno dozvola postoji jer je nadređena domena označena kao pouzdana, odnosno nepouzdana. Tamne boje označavaju da je domena izravno u popisu pouzdanih ili nepouzdanih stranica, ili jer ju je korisnik dodao, ili zbog zadanog popisa koji dolazi s *uMatrixom*.

uMatrix je u srži prilično sličan *NoScriptu* – do nedavno, korisnici preglednika *Google Chrome* morali su koristiti *uMatrix* jer je *NoScript* bio dostupan samo za preglednik *Mozilla Firefox*, no sada ti korisnici imaju izbor između oba alata. U načelu su i *uMatrix* i *NoScript* prilično slični, samo što imaju donekle drugačije sučelje, tako da izbor konkretnog dodatka ovisi o korisnikovim osobnim preferencijama. Po pitanju funkcionalnosti, za razliku od *NoScripta*, *uMatrix* ipak ne nudi rješenje za XSS i *Clickjacking* napade koji dolaze s izvora koje je korisnik dopustio.

5 Zaključak

Korisnički dio većine web stranica izgrađen je HTML-om, CSS-om i JavaScriptom. JavaScript je omogućio stvaranje interaktivnih stranica s bogatim korisničkim sučeljima, no ipak, zbog brojnih dostupnih funkcionalnosti, JavaScript je moguće i zloupotrijebiti.

U ovom su dokumentu opisane najčešće prijetnje na webu koje se oslanjaju na JavaScript kôd, no uz njih, postoji i niz drugih prijetnji koje se isto u nekoj mjeri oslanjaju na JavaScript ili ga koriste kako bi unaprijedili napad: *Clickjacking*, *Pastejacking*, *Tabnabbing*, *Reverse tabnabbing*, *Cross-site request forgery* (CSRF)...

Uzimajući u obzir to da se veliki dio prijetnja na koje nailazimo na webu oslanja na izvršavanje JavaScript kôda, logično je potražiti rješenje koje će blokirati takav zlonamjerni JavaScript kôd, dok će nam u isto vrijeme ipak omogućiti normalno pregledavanje weba. Za tu svrhu među najboljim rješenjima su dodaci (engl. *add-ons*) *NoScript* i *uMatrix*, no njih nije komotno koristiti, tako da ih u konačnici obično koriste napredni korisnici koji su spremni uložiti dodatan trud za višu razinu sigurnosti. Za ostale korisnike, solidno rješenje je korištenje dodatka kao što je *uBlock Origin* koji će u pozadini, bez dodatnog angažmana korisnika, blokirati dio prijetnji.

Prvenstveno krajnji korisnik mora biti odgovoran za svoju sigurnost. No ipak, veliki dio odgovornosti za sigurniji web leži i na programerima koji izrađuju te administratorima koji održavaju web stranice, i na programerima koji izrađuju web preglednike. XSS *Clickjacking*, zlonamjerno oglašavanje (engl. *malvertising*), iskorištavanje ranjivosti web preglednika i slične prijetnje će uvijek do neke mjere biti prisutne, no ipak, programeri, administratori i ostali zaduženi za razvoj i održavanje web stranica i web preglednika su oni koji mogu u izvoru spriječiti veliki dio tih prijetnji. Integracijom sigurnosti u cijeli proces razvoja, održavanja i testiranja moguće je već u početku ukloniti veliki dio ranjivosti na koje se ovakvi napadi oslanjaju.

6 Literatura

1. **W3Techs.** Usage statistics of JavaScript as client-side programming language on websites. [Mrežno] [Citirano: 29. srpnja 2019.] https://w3techs.com/technologies/details/cp-javascript/all/all.
2. **StackOverflow.** Developer Survey Results 2019. [Mrežno] 2019. [Citirano: 31. srpnja 2019.] https://insights.stackoverflow.com/survey/2019.
3. **Sanders, Andrew.** Malware Statistics, Trends and Facts in 2019. *SafetyDetective.* [Mrežno] 15. srpnja 2019. [Citirano: 29. srpnja 2019.] https://www.safetydetective.com/blog/malware-statistics/.
4. **Cimpanu, Catalin.** Firefox zero-day was used in attack against Coinbase employees, not its users. *ZDNet.* [Mrežno] 20. lipnja 2019. [Citirano: 30. srpnja 2019.] https://www.zdnet.com/article/firefox-zero-day-was-used-in-attack-against-coinbase-employees-not-its-users/.
5. **XyliBox.** Blackhole exploit kit v1.2.0. [Mrežno] 11. rujna 2011. [Citirano: 30. srpnja 2019.] https://www.xylibox.com/2011/09/blackhole-exploit-kit-v120.html.
6. **OWASP.** Cross Site Scripting Prevention · OWASP Cheat Sheet Series. [Mrežno] [Citirano: 30. srpnja 2019.] https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.
7. —. DOM based XSS Prevention · OWASP Cheat Sheet Series. [Mrežno] [Citirano: 30. srpnja 2019.] https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html.
8. —. OWASP Top 10 - 2017. [Mrežno] 2017. [Citirano: 30. srpnja 2019.] https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf.
9. **Paun, Marius.** The Real Impact of Cross-Site Scripting. *Dionach.* [Mrežno] 29. srpnja 2016. [Citirano: 30. srpnja 2019.] https://www.dionach.com/blog/the-real-impact-of-cross-site-scripting.
10. **Maone, Giorgio.** NoScript - JavaScript/Java/Flash blocker for a safer Firefox experience! - what is it? - InformAction. [Mrežno] 2019. [Citirano: 30. srpnja 2019.] https://noscript.net/.
11. **Zaharia, Andra.** The Ultimate Guide to Angler Exploit Kit for Non-Technical People. *Heimdal Security.* [Mrežno] 18. svibnja 2016. [Citirano: 29. srpnja 2019.] https://heimdalsecurity.com/blog/ultimate-guide-angler-exploit-kit-non-technical-people/.
12. **Segura, Jérôme.** Exploit Kits: A Fast Growing Threat. *Malwarebytes Labs.* [Mrežno] 21. siječnja 2015. [Citirano: 30. srpnja 2019.] https://blog.malwarebytes.com/101/2015/01/exploit-kits-a-fast-growing-threat/.
13. **Malwarebytes Labs.** What is malvertising? [Mrežno] 24. veljače 2015. [Citirano: 30. srpnja 2019.] https://blog.malwarebytes.com/101/2015/02/what-is-malvertising/.
14. **gorhill.** GitHub - gorhill/uMatrix: uMatrix: Point and click matrix to filter net requests according to source, destination and type. *GitHub.* [Mrežno] 2019. [Citirano: 30. srpnja 2019.] https://github.com/gorhill/uMatrix.
15. —. GitHub - gorhill/uBlock: uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean. *GitHub.* [Mrežno] 2019. [Citirano: 30. srpnja 2019.] https://github.com/gorhill/uBlock.