Zaštita PHP veb aplikacija od XSS napada

Preporuke i smernice za zaštitu

СРБ-ЦЕРТ

Uvod

Veb aplikacije i sajtovi, dostupni putem interneta, pružaju mogućnost korisnicima da pretražuju informacije, kupe željeni proizvod, uče, slušaju muziku i mnoštvo drugih stvari. Ipak da aplikacije i sajtovi ne bi ugrozili podatke koje korisnici ostavljaju i uređaje koji koriste neophodno je primeniti odgovarajuće mere zaštite. Brojne zajednice i kompanije ulažu napore da se poveća bezbednost veb aplikacija i sajtova. Jedna od organizacija koja se bori protiv pretnji koje postoje na internetu je OWASP[1] (The Open Web Application Security Project) a najznačajniji projekat ove organizacije je OWASP top 10.

OWASP top 10 je spisak deset najkritičnijih i najčešćih bezbednosnih pretnji koje su usmerene na veb aplikacije i sajtove. Prva verzija je objavljena u 2003. godini, a poslednje izmene su izrađene u 2021. godini. Prema poslednjoj verziji OWASP top 10 , treće mesto po zastupljenosti zauzima napad koji se izvršava umetanjem (eng. injection) nevalidiranih podataka unetih od strane korisnika direktno u programski kod veb stranice. U ove tehnike spadaju Cross-site Scripting (XSS), SQL injection, LDAP injection.

XSS napadi češće pogađaju korisnike aplikacije, a ne samu aplikaciju. Do njih dolazi kad veb aplikacije prihvataju podatke od korisnika i dinamički ih uključuju u veb stranice bez prethodne validacije podataka. To omogućava napadaču da izvršava proizvoljne komande u internet pregledaču korisnika. Potencijal ovog napada leži u činjenici da se zlonamerni kod izvršava u okviru sesije korisnika, omogućavajući napadaču da zaobiđe bezbednosna ograničenja.

Veb aplikacije

Veb stranice se sastoje od teksta i HTML-a (HyperText Markup Language) koji se vide u veb pretraživaču korisnika. Server može da kontroliše šta će se prikazivati na statičkim stranicama, ali ne može u potpunosti da kontroliše prikaz na dinamičkim stranicama. Ako napadač doda zlonameran sadržaj u dimamičku stranicu, server to neće prepoznati a ni korisnik.

Za izradu dinamičkog veb sadržaja koriste se skriptni jezici koji su dizajnirani tako da se integrišu i komuniciraju sa drugim programskim jezicima koji su potrebni za funkcionisanje veb aplikacija i povezivanje sa bazama podataka. Najčešće korišćeni skriptni jezici su PHP, JavaScript i VBScript.

PHP je skriptni jezik namenjen za izradu dinamičnog veb sadržaja i jedan je od najčešće korišćenih jezika u razvoju veb aplikacija zajedno sa HTML-om koji je „mark up jezik“, odnosno jezik zadužen za vizuelni izgled veb stranice. HTML uz pomoć CSS-a (Cascading Style Sheets) definiše stilove koji određuju izgled HTML elemenata kao što su font, boje i slično.

[1] https://owasp.org

PHP se izvršava na serveru (engl. server-side), a rezultati se šalju klijentu, odnosno veb pretraživaču korisnika, dok se skriptni programski jezik JavaScript izvršava na klijentskoj radnoj stanici, odnosno računaru korisnika.

XSS napadi

XSS je ranjivost koja omogućava ubacivanje zlonamernog koda (na primer JavaScript, HTML, VBScript) u legitimnu veb stranicu, tako što ih napadač unosi sa klijentske strane u elemente za unos na formi, kao što su polja za tekst ili polje za adresu stranice (address bar), a server ih tumači kao legitiman unos korisnika, osim u slučaju DOM-XSS gde je napad usmeren samo na klijentsku stranu. Tako dolazi do izvršavanja neželjenih komandi ili neautorizovanog pristupa bazi podataka. Ova ranjivost se javlja kada se ne filtriraju unosi korisnika pre nego što se prikažu na veb stranici. Na taj način je PHP kod izložen raznim metodama zloupotrebe, pa je preporuka da se prilikom kodiranja, na serverskoj strani primene funkcije koje doprinose da stranice budu bezbedne. Najbolji nivo zaštite se postiže kombinacijom više funkcija za zaštitu od zloupotrebe. Takođe, ukoliko za potrebe funkcionisanja veb stranice nije neophodno koristiti parametre sa klijentske strane koje kontroliše korisnik, ne treba ih koristiti.

XSS napad iskorišćava ranjivosti na dinamičkim veb stranicama, tako što napadač ubacuje ne-validiran zlonameran kod sa klijentske strane i to postaje vidljivo ostalim korisnicima. Napadač može ubaciti JavaScript, VBScript, ActiveX, HTML ili Flash kod koji će se izvršiti na sistemu žrtve skrivajući se iza legitimnog zahteva.

Neke od zlonamernih radnji koje se mogu izvršiti uz pomoć XSS napada su:

- Izvršavanje malicioznog programskog koda,
- Redirektovanje korisnika na maliciozni server,
- Iskorišćavanje prava pristupa korisnika,
- Reklame skrivene u pop-up prozorima,
- Manipulacija podacima,
- Krađa podataka,
- Hajdžeking/preuzimanje sesije,
- Otkrivanje šifara brute-force tehnikom,
- Intranet probing,
- Implementiranje Key logging skripte na legitimnu veb stranicu koja prati sve što korisnik otkuca na zaraženoj stranici koju je posetio,
- Distribucija malicioznih datoteka koje korisnik nesvesno preuzima prilaskom na zaraženu stranicu.

XSS napadi

Vrste XSS napada su:

- Reflektivni XSS (engl. reflected XSS)
- Sačuvani XSS (engl. stored XSS) i
- DOM ili lokalni XSS (engl. DOM-Based XSS)

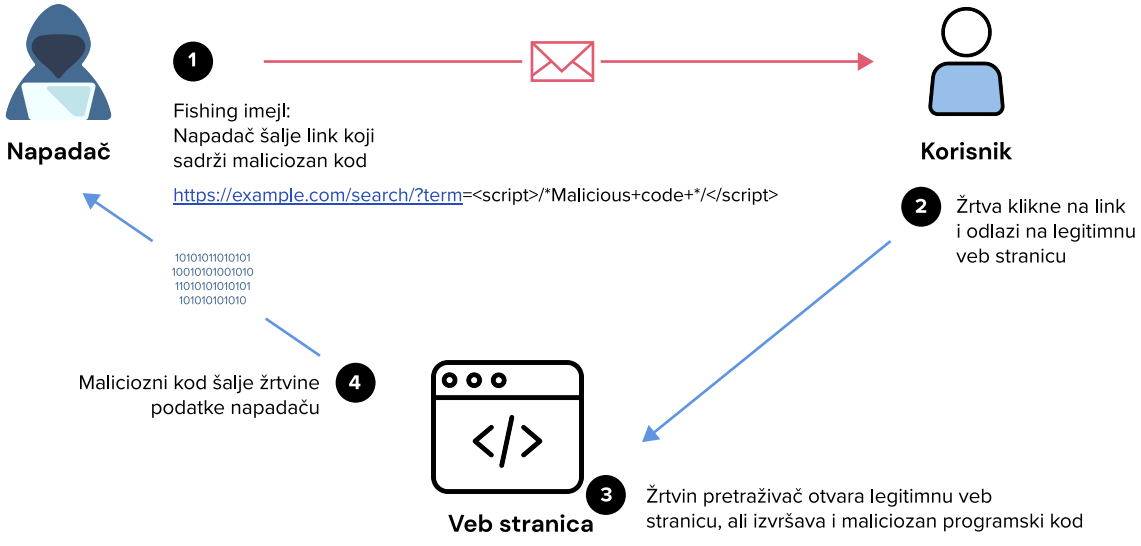
Reflektivni XSS napadi (Tip I)

Reflektivni XSS napad je najjednostavniji i najčešći primer iskorišćavanja ranjivosti veb aplikaci-ja. Izvršava se tako što se vrednost parametra iz URL-a ili sa veb stranice ubacuje u kod HTML stranice koja se dinamički generiše i prikazuje korisniku, a da se pritom vrednost parametra ne proverava. Ova vrsta napada se reflektuje u pretraživaču korisnika, koji klikom na link inicira http zahtev i šalje ga ranjivoj veb stranici. Izmene koje utiču na izgled ili funkcionalnost stranice neće biti trajno sačuvane, već će se prikazivati samo u pretraživaču korisnika koji koristi zaraženi link.

Jedan od primera ovog napada je fišing imejl u kome se nalazi link koji vodi na legitimnu stran-icu, ali ceo URL sadrži i maliciozan programski kod koji pretraživač neće prepoznati kao takav i izvršiće ga ukoliko korisnik klikne. Na zaraženoj stanici se od korisnika može tražiti da unese podatke za logovanje na svoj profil i kada korisnik klikne „Submit“, odnosno klikne na dugme za potvrdu i slanje podataka, napadač će primiti te informacije.

Primer URL putanje koja sadrži maliciozan kod:

https://example.com/search/?term=<script>/*+Maliciozan+kod+*/</script>



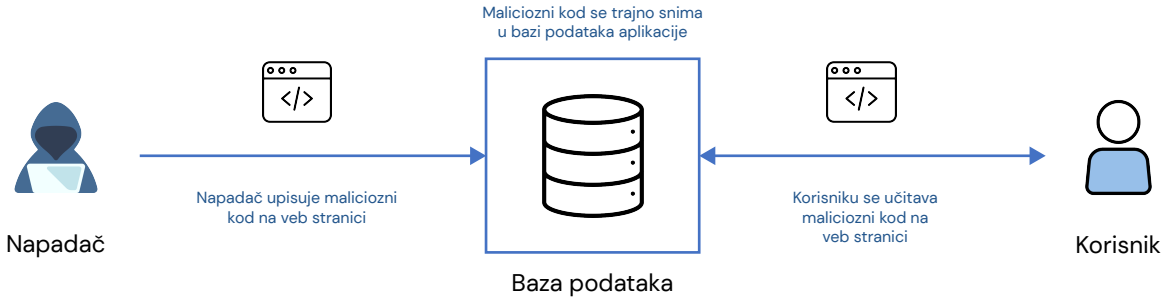
Slika 1. Reflektivni XSS napad

Perzistentni XSS napadi (Tip II)

Perzistentni XSS napad za razliku od reflektovanog XSS-a, zlonameran programski kod snima na veb serveru ranjive aplikacije, u bazama podataka ili datotekama. Ova vrsta XSS napada iskorišćava propuste i ranjivosti na veb stranici na kojoj ne postoji kontrola unosa u polja kao što su korisničko ime, komentari ili polja za pretragu. To su najčešće veb aplikacije za međusobnu interakciju korisnika ili stranice na kojima se ostavljaju poruke koje ostali korisnici mogu da vide (forumi, blogovi, socijalne mreže). Napad se sprovodi tako što napadač ubacuje maliciozni programski kod (na primer JavaScript) u polja za unos, a uneti podaci se trajno čuvaju na strani servera. Kada ostali korisnici pristupe stranici, zlonamerni kod se izvršava u okviru njihovih sesija.

Na slici 2 je prikazano unošenje programskog koda kroz mehanizam dodavanja nove poruke na veb aplikaciji. Programski kod koji je dodat se snima i izvršava svaki put kada se pristupi kompromitovanoj stranici, koja može biti ona sa koje je napadački kod i ubačen ili druga na kojoj se snimljeni zlonamerni kod samo izvršava.

Napadač u ovom slučaju ne mora kreirati linkove koji upućuju na ranjivu veb stranicu, linkove koji u sebi sadrže napadački kod i slati ih korisnicima da bi ih oni pokrenuli. Dovoljno je da jednom maliciozni kod ubaci na veb stranicu i sačeka da korisnik pristupi zaraženoj stranici



Slika 2 Perzistentni XSS

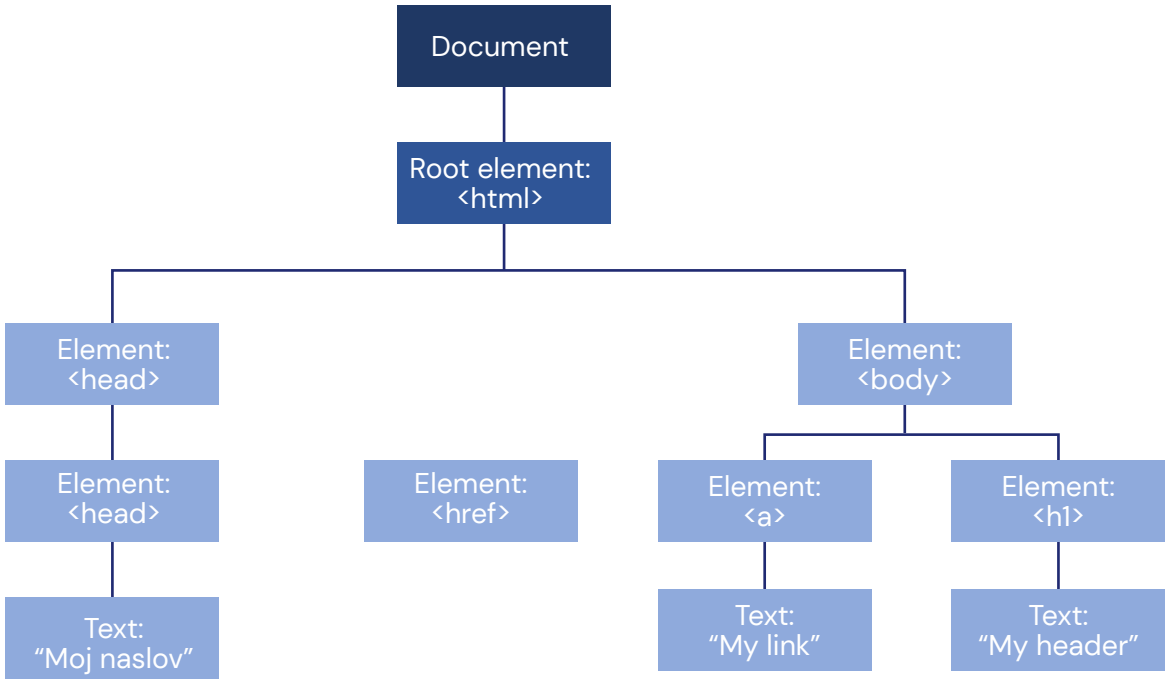
DOM XSS (Tip 0)

DOM (Document Object Model) je standard za način preuzimanja, izmenu, dodavanje i brisanje HTML elemenata, odnosno tagova (oznaka) koji se koriste za definisanje elemenata u dokumentu.

DOM definiše:

- HTML elemente kao objekte,
- Karakteristike HTML elemenata (engl. properties),
- Metode za pristup HTML elementima (engl. methods) i
- Ponašanje HTML elemenata (engl. events).

JavaScript zahvaljujući ovom modelu može da kreira dinamički HTML sadržaj, tako što menja HTML elemente na stranici, HTML atribute, CSS stilove, briše postojeće HTML elemente i atribute, dodaje nove HTML elemente i atribute i utiče na ponašanje elemenata i kreira nova ponašanja (events).



Slika 3. DOM struktura

DOM XSS ili lokalni XSS je vrsta XSS napada koja se izvršava na klijentskoj strani u okviru objekta DOM. Ovaj napad je moguće izvršiti u slučajevima kada se podaci upisuju u DOM objekat bez prethodne provere sadržaja.

Veb stranice za prikaz grešaka, pozdravnih poruka i slično mogu koristiti skripte napisane u JavaScript-u koje se izvršavaju prilikom generisanja stranice u pretraživaču. Te skripte mogu da koriste DOM za preuzimanje podatka iz URL-a ili HTTP zaglavlja i prikazivanje korisniku. Preuzeti podaci se dodaju u HTML kod i ako se sadržaj tih podataka ne proverava, javlja se ranjivost koju napadač može iskoristiti.

Napad se distribuira slanjem fišinga koji sadrži URL sa malicioznim kodom.

Od reflektivnog XSS napada, DOM XSS se razlikuje u tome što se ne iskorišćava ranjivost HTML elemenata, već ranjivost DOM objekta koji se izvršava na klijentskoj strani u okviru pretraživača korisnika. Reflektovani XSS iskorišćava ranjivosti dinamičkih veb stranica, dok DOM XSS iskorišćava ranjivost i statičkih i dinamičkih veb aplikacija.

Preporuke za sprečavanje XSS napada

U nastavku su navedene odbrambene tehnike koje se najčešće koriste za zaštitu od XSS napada. Za uspešnu odbranu je potrebno napraviti kombinaciju odgovarajućih tehnika.

Primena enkodiranja

Svaka varijabla koja ne prođe kroz proces validacije je potencijalna ranjivost. Da bi se ova opasnost otklonila, potrebno je primeniti metode koje će obezbediti proveru sadržaja parametara koji se šalju veb stranici i na taj način onemogućiti napadaču da ubaci i izvrši maliciozan sadržaj na veb stranici.

Validacija varijabli htmlspecialchars()

Funkcija htmlspecialchars() je najpoznatija metoda za zaštitu od XSS napada[2]. Specijalne karaktere konveruje u HTML entitete i na taj način sprečava zloupotrebu.

Na primer:

Entitet	Enkodirani entitet
'	&#x27
"	&quot;
&	&amp
<	&lt;
>	&gt;

Pored HTML eniteta enkodiranje se može primeniti i na HTML atribute, URL, JavaStript funkcije i CSS primenom odgovarajućih pravila.

[2] https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
https://www.cloudways.com

htmlentities()

Ova funkcija ima slične rezultate kao htmlspecialchars() ali obuhvata više entiteta. Preterana upotreba ove funkcije može dovesti do prekomernog kodiranja i pogrešnog prikaza objekata, pa treba biti obazriv.

Connect-src

Restrikcija URL-ova koji mogu biti učitani sa interfejsa, u cilju prevencije od XSS skriptinga.

strip_tags()

Izdvaja i briše sadržaj koji se nalazi između HTML i PHP tagova i vraća prazan string, odnosno string sa nula bajtova. Ukoliko zagrade za zatvaranje tagova nisu pravilno uparene, ova funkcija neće biti primenjena.

addslashes()

Dodaje znak kose crte da bi se sprečio pokušaj napadača da doda izvršni fajl na kraju komande.

Content Security Policy (CSP)

CSP je poslednja i najrestriktivnija opcija za zaštitu od XSS napada, jer pretraživači izvršavaju JavaScript koji stiže sa servera, bez obzira na to da li je skript maliciozan ili ne.

CSP se može postaviti u HTTP header gde bi se odredila whitelist-a i spisak odobrenih izvora koje pretraživač može koristiti.

Sledeći primer pokazuje da pretraživač može izvršiti samo URL koji pripada trenutnom domenu na kome se sajt nalazi, dok će sve ostale izvore odbaciti.

```
X-Content-Security-Policy: script-src 'self'
```

Limitiranje izvora se može uraditi i za sledeće entitete:

- connect-src: limitira izvore na koje se korisnik može priključiti korišćenjem: XML-HttpRequest, WebSocket...
- font-src: limitira izvore za fontove,
- frame-src: limitira URL izvore koji se mogu dodati na veb stranicu kao “frame”,
- img-src: limitira izvor za slike,
- media-src: limitira izvor za video i audio,
- object-src: limitira izvor za fleš i ostale plaginove,
- script-src: limitira izvor za skripte,
- style-src: limitira izvore za CSS fajlove.

PHP Biblioteke za prevenciju XSS napada

HTML Purifier – http://htmlpurifier.org/
PHP Anti-XSS – https://code.google.com/p/php-antixss/
htmLawed – http://www.bioinformatics.org/phplabware/internal_utilities/htmLawed/

Postavke na Apache serveru

Otvoriti Apache konfiguracionu datoteku koja se može nalaziti na nekoj od sledećih adresa:

```
/etc/apache2/httpd.conf
/etc/apache2/apache2.conf
/etc/httpd/httpd.conf
/etc/httpd/conf/httpd.conf
```

Aktivirati zaštitu od XSS napada

Proveriti da li je na Apache serveru instaliran mod_headers, pa ako jeste dodati u konfiguracionu datoteku sledeće zaglavlje

```
# Add Security Headers
<IfModule mod_headers.c>
   # Protect against XSS attacks
   Header set X-XSS-Protection "1; mode=block"
</IfModule>
```

Zahvaljući izmeni vrednosti na 1 i „mode“ na „block“, pretraživač neće renderovati (prikazati) stranicu ukoliko XSS bude detektovan.

Restartovati Apache server

```
$ sudo service apache2 restart
```