Заштита PHP веб апликација од XSS напада

Препоруке и смернице за заштиту

СРБ-ЦЕРТ

Увод

Веб апликације и сајтови, доступни путем интернета, пружају могућност корисницима да претражују информације, купе жељени производ, уче, слушају музику и мноштво других ствари. Ипак како апликације и сајтови не би угрозили податке које корисници остављају и уређаје који користе неопходно је применити одговарајуће мере заштите. Бројне заједнице и компаније улажу напоре да се повећа безбедност веб апликација и сајтова. Једна од организација која се бори против претњи које постоје на интернету је OWASP (The Open Web Application Security Project) а најзначајнији пројекат ове организације је „OWASP top 10“.

„OWASP top 10“ је списак десет најкритичнијих и најчешћих безбедносних претњи које су усмерене на веб апликације и сајтове. Прва верзија је објављена 2003. године, а последње измене су израђене 2021. године. Према последњој верзији „OWASP top 10“ , треће место по заступљености заузима напад који се извршава уметањем (eng. injection) невалидираних података унетих од стране корисника директно у програмски код веб странице. У ове технике спадају: Cross-site Scripting (XSS), SQL injection, LDAP injection.

XSS напади чешће погађају кориснике апликације, а не саму апликацију. До њих долази кад веб апликације прихватају податке од корисника и динамички их укључују у веб странице без претходне валидације података. То омогућава нападачу да извршава произвољне команде у интернет прегледачу корисника. Потенцијал овог напада лежи у чињеници да се злонамерни код извршава у оквиру сесије корисника, омогућавајући нападачу да заобиђе безбедносна ограничења.

Веб апликације

Web странице се састоје од текста и HTML-a (HyperText Markup Language) који се виде у веб претраживачу корисника. Сервер може да контролише шта ће се приказивати на статичким страницама, али не може у потпуности да контролише приказ на динамичким страницама. Ако нападач дода злонамеран садржај у димамичку страницу, сервер то неће препознати а ни корисник.

За израду динамичког веб садржаја користе се скриптни језици који су дизајнирани тако да се интегришу и комуницирају са другим програмским језицима који су потребни за функционисање веб апликација и повезивање са базама података. Најчешће коришћени скриптни језици су PHP, JavaScript и VBScript.

PHP је скриптни језик намењен за израду динамичног веб садржаја и један је од најчешће коришћених језика у развоју веб апликација заједно са HTML- ом који је „mark up“ језик, односно језик задужен за визуелни изглед веб странице. HTML уз помоћ CSS-a (Cascading Style Sheets) дефинише стилове који одређују изглед HTML елемената као што су фонт, боје и слично.

[1] https://owasp.org

PHP извршава се на серверу (енгл. server-side), а резултати се шаљу клијенту, односно веб претраживачу корисника, док се скриптни програмски језик JavaScript извршава на клијентској радној станици, односно рачунару корисника.

XSS напади

XSS је рањивост која омогућава убацивање злонамерног кода (на пример JavaScript, HTML, PHP, VBScript) у легитимну веб страницу, тако што их нападач уноси са клијентске стране у елементе за унос на форми, као што су поља за текст или поље за адресу странице (address bar), сервер их тумачи као легитимни кориснички унос, осим у случају DOM-XSS где је напад усмерен само на клијентску страну. Тако долази до извршавања нежељених команди или неауторизованог приступа бази података. Ова рањивост се јавља када се не филтрирају уноси корисника пре него што се прикажу на веб страници. На тај начин је PHP код изложен разним методама злоупотребе, па је препорука да се приликом кодирања, на серверској страни примене функције које доприносе да странице буду безбедне. Најбољи ниво заштите се постиже комбинацијом више функција за заштиту од злоупотребе. Такође, уколико за потребе функционисања веб странице није неопходно користити параметре са клијентске стране које контролише корисник, не треба их користити.

XSS напад искоришћава рањивости на динамичким веб страницама, тако што нападач убацује невалидиран злонамеран код са клијентске стране и то постаје видљиво осталим корисницима. Нападач може убацити JavaScript, VBScript, ActiveX, HTML или Flash код који ће се извршити на систему жртве скривајући се иза легитимног захтева.

Неке злонамерне радње које могу да се изврше уз помоћ XSS напада су:

- Извршавање малициозног програмског кода,
- Редиректовање корисника на малициозни сервер,
- Искоришћавање права приступа корисника,
- Рекламе скривене у pop-up прозорима,
- Манипулација подацима,
- Крада података,
- Хајџекинг /преузимање сесије,
- Откривање шифара brute-force техником,
- Интерна мрежа probing,
- Имплементирање Key logging скрипте на легитимну веб страницу која прати све што корисник откуца на зараженој страници коју је посетио,
- Дистрибуција малициозних датотека које корисник несвесно преузима приласком на заражену страницу.

XSS напади

Врсте XSS напада су:

1. Рефлективни XSS (енгл. reflected XSS)
2. Сачувани XSS (енгл. stored XSS) i
3. DOM или локални XSS (енгл. DOM-Based XSS)

Рефлективни XSS напади (Тип I)

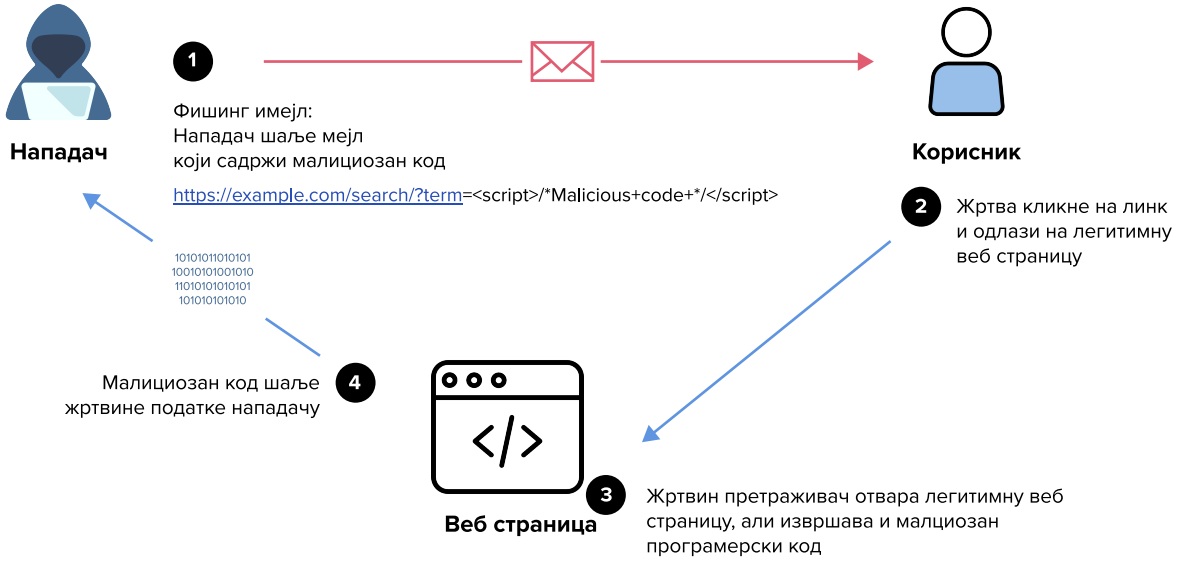
Рефлективни XSS напад је најједноставнији и најчешћи пример искоришћавања рањивости веб апликација. Извршава се тако што се вредност параметра из URL-а или са веб странице убацује у код HTML странице која се динамички генерише и приказује кориснику, а да се притом вредност параметра не проверава.

Ова врста напада се рефлектује у претраживачу корисника, који кликом на линк иницира http захтев и шаље га рањивој веб страници. Измене које утичу на изглед или функционалност странице неће бити трајно сачуване, већ ће се приказивати само у претраживачу корисника који користи заражени линк.

Један од примера овог напада је фишинг имејл у коме се налази линк који води на легитимну страницу, али цео URL садржи и малициозан програмски код који претраживач неће препознати као такав и извршиће га уколико корисник кликне. На инфицираној станици, од корисника се може тражити да унесе податке за пријављивање на свој профил и када корисник кликне „Submit“, односно кликне на дугме за потврду и слање података, нападач ће примити те информације.

Пример URL путање која садржи малициозан код:

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



Слика 1. Рефлективни XSS напад

Перзистентни XSS напади (Тип II)

Перзистантни XSS напад за разлику од рефлектованог XSS-а, злонамеран програмски код снима на веб серверу рањиве апликације, у базама података или датотекама. Ова врста XSS напада искоришћава пропусте и рањивости на веб страници на којој не постоји контрола уноса у поља као што су корисничко име, коментари или поља за претрагу. То су најчешће веб апликације за међусобну интеракцију корисници или странице на којима се остављене поруке које други корисници могу да виде (форуми, блогови, социјалне мреже). Напад се спроводи тако што нападач убацује малициозни програмски код (на пример JavaScript) у поља за унос, а унети подаци се трајно чувају на страни сервера. Када остали корисници приступе страници, злонамерни код се извршава у оквиру њихових сесија.

На слици 2 је приказано уношење програмског кода кроз механизам додавања нове поруке на веб апликацији. Програмски код који је додат се снима и извршава сваки пут када се приступи компромитованој страници, која може бити она са које је нападачки код и убачен или друга на којој се снимљени злонамерни код само извршава.

Нападач у овом случају не мора креирати линкове који упућују на рањиву веб страницу и који у себи садрже нападачки код и слати их корисницима да би их они покренули. Довољно је да једном малициозни код убаци на веб страницу и сачека да корисник приступи зараженој страници.

Слика 2. Перзистентни XSS

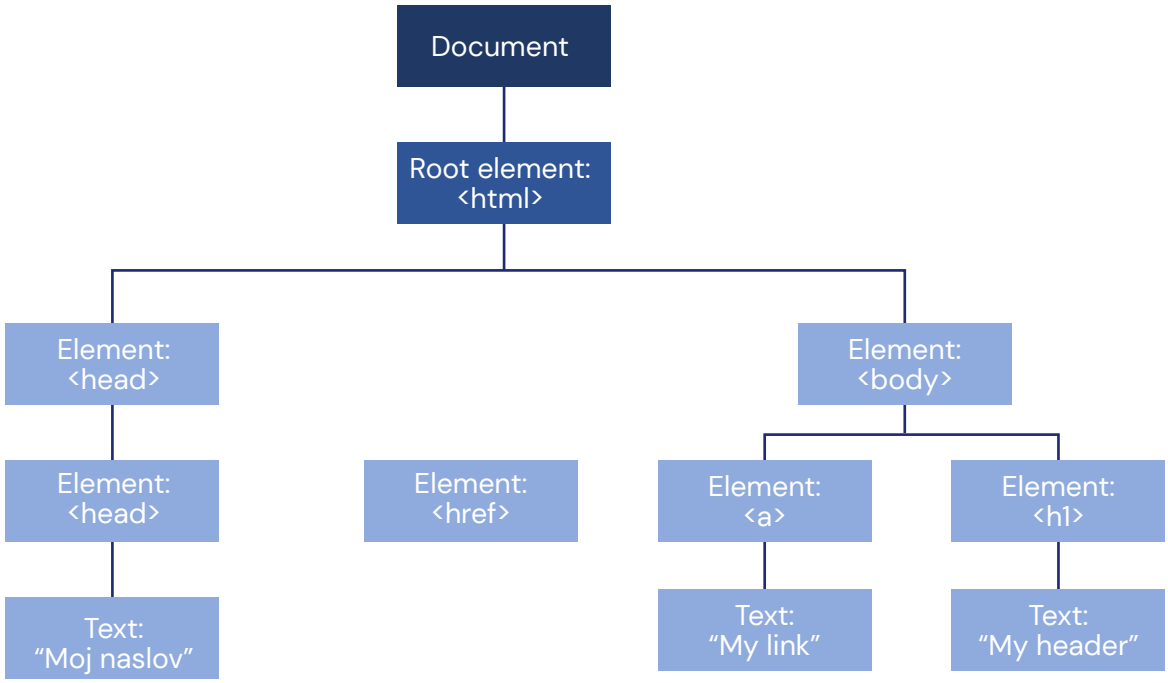
DOM XSS (Тип 0)

DOM (Document Object Model) је стандард за начин преузимања, измену, додавање и брисање HTML елемената, односно тагова (ознака) који се користе за дефинисање елемената у документу.

DOM дефинише:

- HTML елементе као објекте,
- Карактеристике HTML елемената (енгл. properties),
- Методе за приступ HTML елементима (енгл. methods) и
- Понашање HTML елемената (енгл. events).

JavaScript захваљујући овом моделу може да креира динамички HTML садржај, тако што мења HTML елементи на страници, HTML атрибуте, CSS стилове, брише постојеће HTML елементе и атрибуте, додаје нове HTML елементе и атрибуте и утиче на понашање елемената и креира нова понашања (events).



Слика 3. DOM структура

DOM или локални XSS је врста XSS напада која се извршава на клијентској страни у оквиру објекта DOM. Овај напад је могуће извршити у случајевима када се подаци уписују у DOM објекат без претходне провере садржаја.

Веб странице за приказ грешака, поздравних порука и слично могу користити скрипте написане у JavaScript-у које се извршавају приликом генерисања странице у претраживачу. Те скрипте могу да користе DOM за преузимање података из URL-а или HTTP заглавља и приказивање кориснику. Преузети подаци се додају у HTML kod и ако се садржај тих података не проверава, јавља се рањивост коју нападач може искористити.

Напад се дистрибуира слањем фишинга који садржи URL са малициозним кодом.

Од рефлективног XSS напада, DOM XSS се разликује у томе што се не искоришћава рањивост HTML елемената, већ рањивост DOM објекта који се извршава на клијентској страни у оквиру претраживача корисника. Рефлектовани XSS искоришћава рањивости динамичких веб страница, док DOM XSS искоришћава рањивост и статичких и динамичких веб апликација.

Препоруке за спречавање XSS напада

У наставку су наведене одбрамбене технике које се најчешће користе за заштиту од XSS напада. За успешну одбрану је потребно направити комбинацију одговарајућих техника.

Примена енкодирања

Свака варијабла која не прође кроз процес валидације је потенцијална рањивост. Да би се ова опасност отклонила, потребно је применити методе које ће обезбедити проверу садржаја параметара који се шаљу веб страници и на тај начин онемогућити нападачу да убаци и изврши малициозан садржај на веб страници.

Валидација варијабли htmlspecialchars()

Функција htmlspecialchars() је најпознатија метода за заштиту од XSS напада. Специјалне карактере конверује у HTML ентитете и на тај начин спречава злоупотребу.

На пример:

Ентитет	Енкодирани ентитет
'	&#x27
"	&quot;
&	&amp
<	&lt;
>	&gt;

Поред HTML енитета енкодирање се може применити и на HTML атрибуте, URL, JavaStript функције и CSS применом одговарајућих правила.

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

htmlentities()

Ова функција има сличне резултате као htmlspecialchars() али обухвата више ентитета. Претерана употреба ове функције може довести до прекомерног кодирања и погрешног приказа објеката, па треба бити обазрив.

Connect-src

Рестрикција URL- ова који могу бити учитани са интерфејса, у циљу превенције од XSS скриптинга.

strip_tags()

Издваја и брише садржај који се налази између HTML и PHP тагова и враћа празан стринг, односно стринг са нула бајтова. Уколико заграде за затварање тагова нису правилно упарене, ова функција неће имати примену.

addslashes()

Додаје знак косе црте да би се спречио покушај нападача да дода извршни фајл на крају команде.

Content Security Policy (CSP)

CSP је последња и најрестриктивнија опција за заштиту од XSS напада, јер претраживачи извршавају JavaScript који стиже са сервера, без обзира на то да ли је скрипт малициозан или не. CSP се може поставити у HTTP header где би се одредила whitelist-а и списак одобрених извора које претраживач може користити. Следећи пример показује да претраживач може извршити само URL који припада тренутном домену на коме се сајт налази, док ће све остале изворе одбацити.

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

Лимитирање извора се може урадити и за следеће ентитете:

- connect-src: лимитира изворе на које се корисник може прикључити коришћењем: XMLHttpRequest, WebSocket...
- font-src: лимитира изворе за фонтове,
- frame-src: лимитира URL изворе који се могу додати на веб страницу као “frame”,
- img-src: лимитира извор за слике,
- media-src: лимитира извор за видео и аудио,
- object-src: лимитира извор за fleš и остале plaginove,
- script-src: лимитира извор за скрипте,
- style-src: лимитира изворе за CSS фајлове.

PHP Библиотеке за превенцију XSS напада

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

Поставке на Apache серверу

Отворити Apache конфигурациону датотеку која се може налазити на некој од следећих адреса:

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

Активирати заштиту од XSS напада

Проверити да ли је на Apache серверу инсталиран mod_headers, па ако јесте додати у конфигурациону датотеку следеће заглавље

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

Захваљући измени вредности на 1 i „mode“ na „block“, претраживач неће приказати (прикажи) страницу ако XSS буде детектован.

Рестартовати Apache сервер

```
$ sudo service apache2 restart
```