UDC: 004.6 Info M str. 33-42

ДЕТЕКЦИЈА УПАДА У СИСТЕМЕ ЗА УПРАВЉАЊЕ САДРЖАЈЕМ ПРИМЕНОМ РЕГУЛАРНИХ ИЗРАЗА
CONTENT MANAGEMENT SYSTEM INTRUSION DETECTION USING REGULAR EXPRESSIONS

Сара Вујичић, др Марјан Милошевић

РЕЗИМЕ: Све већа присутност веб-сајтова заснованих на системима за управљање садржајем (*CMS* платформама) навела је нападаче на развијање мноштва злонамерних алата и стратегија и покренула питање безбедности система за управљање садржајем. Овај рад истражује изазове и решења за унапређење безбедности на веб страницама које користе *CMS*. Како *CMS* платформе омогућавају корисницима креирање и управљање дигиталним садржајем, истовремено привлаче пажњу нападача који траже рањивости веб-сајтова. Рад спроводи свеобухватну анализу безбедносних мера специфичних за *CMS*, бавећи се питањима као што су аутентификација корисника, интегритет података и заштита од уобичајених веб напада. Поред тога, рад даје увид у основне принципе система за откривање напада (*IDS*), правећи разлику између детекције засноване на потписима и аномалијама. Пружа критичку анализу предности и ограничења имплементација *IDS*-а. Такође, укључује развој апликације за анализу веб логова, која углавном употребом регуларних израза, може да препозна обрасце који указују на злонамерне активности.

КЉУЧНЕ РЕЧИ: детекција базирана на потписима, систем за управљање садржајем, систем за детекцију упада, регуларни изрази, веб-лог

ABSTRACT: With the widespread adoption of Content Management Systems (CMS) for website development, ensuring robust web security within these systems has become a critical concern. This paper explores the challenges and solutions for enhancing security in CMS-driven websites. As CMS platforms continue to empower users in creating and managing digital content, they also attract the attention of malicious actors seeking vulnerabilities. The paper conducts a comprehensive analysis of security measures specific to CMS, addressing issues such as user authentication, data integrity, and protection against common web attacks. Furthermore, this paper reviews the fundamental principles of Intrusion Detection Systems (IDS), distinguishing between signature-based and anomaly-based detection methods. It gives a critical analysis of the strengths and limitations of IDS implementations. It also includes the development of a web log analysis application capable of recognizing patterns indicative of malicious activities using mainly regular expressions.

KEY WORDS: signature based detection, Content Management Systems, Intrusion Detection Systems, regular expressions, weblogs

1. УВОД

Током последњих десет година системи за управљање садржајем су постајали све популарнији. Број веб-сајтова који користе неки од система за управљање садржајем се дуплирао. 2012. године било их је свега 29%, док је данас тај проценат порастао на 60,2. [1]

Системи за управљање садржајем или *CMS* (енгл. *Content Management System*) су своју популарност стекли пре свега тиме што су омогућили просечном кориснику интернета да креира веб-сајтове. Уз помоћ разноврсних тема и додатака удаљених на само неколико кликова мишем, системи за управљање садржајем учинили су да уређивање веб-страница буде доступно свима и не захтева познавање кодирања. Данас највећи саобраћај имају управо они веб-сајтови који користе неки *CMS* – од милион најпосећенијих сајтова на интернету скоро 900.000 користи неки систем за управљање садржајем. [2]

Популарни *CMS*-ови представљају мету за хакере и злоупотребу. Рањивости у системима за управљање садржајем могу довести до озбиљних проблема као што су крађа података, напади на приватност и др. Успостављање система за детекцију и превенцију напада на *CMS* је кључно за очување интегритета, сигурности и стабилности веб-страница. Рад пружа увид у то које све претње вребају на системе за управљање садржајем, у механизме које постојећи системи користе за њихово откривање и апликацију развијену у склопу овог рада која тумачи веб логове и на основу тога доноси закључке о потенцијалним покушајима напада, користећи регуларне изразе.

1.1. *Појам система за управљање садржајем*

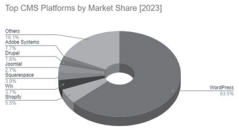
Oracle наводи следећу дефиницију: систем за управљање садржајем помаже компанијама у управљању дигиталним садржајем. Читави тимови могу да користе ове системе за креирање, измену, организовање и објављивање садржаја. [3]

Систем за управљање садржајем јесте апликација која омогућава корисницима да направе веб-сајт без кодирања. Корисници уместо да кодирају веб-сајт од нуле, користе кориснички интерфејс – преузимају различите теме и додатке како би добили жељени изглед сајта и функционалности.

1.2. *WordPress*

На почетку 2024. године на тржишту је било доступно преко 800 *CMS* платформи. Убедљиви лидер је *WordPress* са уделом од 63,5 %. *Shopify* је друга најпопуларнија плат-

форма на тржишту система за управљање садржајем са тржишним уделом од 5,5 % (Слика 1). Очекује се да ће глобална вредност тржишта *CMS*-а достићи 123,5 милијарди долара до 2026. године. Преко 73 милиона веб-сајтова на интернету користи систем за управљање садржајем. У току је процват електронске трговине. Од милион најпопуларнијих веб-сајтова више од 28 % су сајтови е-трговине. Најпопуларнији *CMS*-ови за е-трговину су *WooCommerce Checkout*, *Shopify*, *Magento* и *Wix Store*. Занимљиво је да је број веб-сајтова који не користе ниједан систем за управљање садржајем опао у односу на 2022. годину, што значи да све више веб-сајтова прелази на *CMS* платформе. [4]



Слика 1: *Најпопуларнији CMS-ови у 2023. години [4]*

Најзаступљенији *WordPress* је првобитно настао са идејом да буде платформа искључиво намењена за блогове. Међутим, захваљујући сталним ажурирањима и побољшањима, постао је бесплатан *CMS* отвореног кода који се прилагођава веб-сајтовима различитих величина и потреба. Данас преко 43,2% веб-сајтова користи *WordPress*. Међу најпознатијим издвајају се Sony Music, Whitehouse.gov и Playstation Blog.

По питању безбедности, платформа пружа следеће сигурносне мере: сви сајтови који користе верзију 3.7 или новије верзије добијају аутоматско одржавање и сигурносна ажурирања, док почевши од верзије 5.5 аутоматска ажурирања се односе и на додатке и теме. Додатне мере су доступне употребом додатака. Корисници могу сами да одаберу додатке који доприносе заштити њиховог сајта. Међутим, често се дешава да корисници немају довољно знања да правилно подесе додатке, не ажурирају их редовно или инсталирају додатке који садрже неку познату рањивост. Истраживања су показала да су управо додаци проузроковали 56% напада током 2019. године. [5]

2. БЕЗБЕДНОСТ СИСТЕМА ЗА УПРАВЉАЊЕ САДРЖАЈЕМ

Услед велике популарности, огромног броја корисника и приступачности изворног кода, веб-сајтови изграђени на *WordPress* платформи су честа мета нападача. Бројне заједнице и компаније улажу напоре у повећавање безбедности веб-апликација отвореног кода. *OWASP* пројекат (енгл. *The Open Web Application Security Project*) је заједница која обухвата корпорације, образовне установе и појединце из целог света чији је заједнички циљ пружање помоћи и обуке програмерима и компанијама ради унапређивања безбедности.

Најзначајнији пројекат ове организације јесте „*OWASP* топ 10“ – листа десет највећих претњи на интернету. Безбедносни проблеми који се односе на *CMS*-ове су покривени овом листом у оквиру општег контекста безбедности веб-апликација. Још један значајни допринос ове организације јесте прокси за тестирање и скенирање безбедности веб апликација – *OWASP ZAP* (*Zed Attack Proxy*). Реч је о бесплатном алату отвореног кода за тестирање безбедности веб-апликација. *ZAP* омогућава корисницима да прослеђују и модификују *HTTP* захтеве и одговоре између клијента и сервера како би истраживали потенцијалне слабости.

2.1. *Најзаступљенији упади*

Бележи се све већи пораст финансијских превара у е-трговини. Нападаче привлаче различите методе плаћања. Хакери обављају неовлашћене трансакције што за последицу има велике губитке у пословању. [6]

Иако застарео тип напада, и даље изненађујуће популарни јесу напади грубом силом (енгл. *brute force attacks*). Односе се на најједноставнији облик упада чија је мета лозинка корисника. Нападач испробава огроман број различитих комбинација лозинки све док не пронађе исправну. Показало се да иако метода делује наивно, веома је ефикасна против слабих лозинки. Жртва „brute force“ напада је 2016. године био и „кинески *eBay*“, веб-сајт за електронску трговину *TaoBao*. Тада је за два месеца хаковано 21 милион налога, што чини петину укупног броја корисника. [7]

Један од најчешћих упада у *WordPress* чији је циљ добијање приступа администраторском панелу је инјектовање *SQL* кода (енгл. *SQL Injection*). Нападач извршава злонамерне упите над *MySQL* базом података. Било које поље за унос података, контакт форма или поље за претрагу може бити осетљиво на инјектовање *SQL* кода. Од неизмерне је важности да сви додаци и теме буду поуздани. Уколико неки додатак није ажуриран дуже од шест месеци, врло је вероватно да програмер више не ради на одржавању изворног кода и такви додаци су најосетљивији на упаде и најбоље је избегавати их у потпуности.

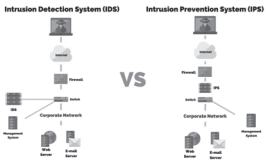
Такође чест вид упада јесу упади скриптама (енгл. *Cross Site Scripting*) који су познати под називом *XSS* упади. Представљају инјектовање злонамерног *JavaScript* кода с намером да се прикупе кориснички подаци без корисниковог пристанка или да се корисник преусмери на неки други сајт.

Напади о којима се данас највише говори јесу тзв. *DDoS* (енгл. *Distributed Denial of Service*) напади – веб сервер који је мета напада постаје преплављен великим бројем лажних захтева до те мере да више није доступан корисницима којима је намењен.

3. СИСТЕМИ ЗА ДЕТЕКЦИЈУ И ПРЕВЕНЦИЈУ УПАДА

Системи за детекцију упада (енгл. *Intrusion Detection System*, *IDS*) представљају одбрамбени алат за откривање безбедносних претњи на вебу. Рад једног таквог система се заснива на прикупљању информација са низа мрежних и рачунарских извора чијим се анализирањем откривају недозвољене активности и злоупотребе система на којима се налазе.

Системи за превенцију упада (енгл. *Intrusion Prevention System*, *IPS*) иду један корак даље и активно блокирају или ограничавају активности које би могле представљати претњу. Они користе исте технике за детекцију као и *IDS*, али им је додата могућност да одговоре и примене заштитне мере. Системи за превенцију напада се често користе у комбинацији са системима за детекцију напада како би се остварио комплетан безбедносни оквир.



Слика 2: *Упоредни приказ IDS-а и IPS-а [8]*

IDS се обично поставља иза тапа[1] или спан порта[2] који копира све пакете који су му послати на *IDS* без утицаја на ток података. Ова технологија је оно што раздваја систем за детекцију напада од система за превенцију напада – *IPS* се поставља у директан комуникациони пут између извора и одредишта, активно анализирајући и примењујући аутоматске акције на све токове саобраћаја који улазе у мрежу (Слика 2).

3.1. *Класификација система за детекцију упада*

Најстарија подела система за откривање напада јесте на основу тога шта се системом заправо детектује. У том контексту, системи се деле на две групе:

1. Системе за детекцију злоупотреба (енгл. *misuse intrusion detection*) и
2. Системе за детекцију неправилности (енгл. *anomaly intrusion detection*).

Под детекцијом злоупотреба подразумева се откривање познатих напада усмерених на слабости система. С друге стране, детекција неправилности је усмерена ка препознавању неуобичајених активности које могу указивати на упад у систем.

Према механизму детекције разликују се:

1. Системи на бази потписа (енгл. *signature-based detection*),
2. Системи на бази аномалија и
3. Хибридни системи (енгл. *hybrid intrusion detection*).

Системи базирани на потписима могу детектовати само оне упаде за које су унапред креирани потписи. У литератури се често називају и системима на бази знања и системима на бази злоупотребе. Недостаци ових система су постојање различитих варијанти упада, лажни позитиви и негативи и преоптерећење подацима. Када је реч о постојању различитих варијанти, како се базирају на потписима, може се креирати нова варијанта упада у намери да се избегне детекција. Додатно, сами потписи могу креирати лажне позитиве у случају да је природа напада таква да се тешко може изоловати од регуларног саобраћаја. До преоптерећења може доћи уколико сензор или аналитичар задају превише информација ради ефективнијег анализирања. Кључни недостатак система на бази потписа је немогућност детектовања непознатих напада. Насупрот њима, системи на бази аномалија могу препознати и нове нападе. Наиме, систем на бази аномалија прати системске активности и категорише их на нормалне и неуобичајене. Таква класификација се заснива пре свега на хеуристици или правилима, а не на секвенцама и потписима. Самим тим је могуће детектовати било коју врсту злоупотребе која излази ван оквира очекиваног понашања. *IDS* на бази аномалије ће генерисати аларм на сумњиву активност уколико детектује, на пример, стотине покушаја пријављивања у неком кратком интервалу од неколико секунди. Овакви системи такође користе скуп информација које одређују шта је нормално у понашању мреже. Тај скуп се назива профил и састоји се од листе параметара који се односе на објекат мониторинга. Примери параметара су нормално време пријављивања, трајање корисничке сесије, оптерећење процесора, коришћење диска и сл. Највећи изазов код система на бази аномалија јесте конструисање ефикасног, адаптивног и самоучећег профила и подешавање његових параметара. Ако је праг за активирање аларма висок, могућ је већи број непримећених напада, док нижи праг може узроковати више лажних позитивних аларма. Дакле, и детекција на бази аномалија пати од истог недостатка као детекција на бази злоупотреба. Међутим, ако би аларми који потичу од оба приступа могли да се доведу у корелацију, дошло би до побољшања тачности аларма. Снага детекције неправилности је у малом броју лажних негатива. Рад мрежног система би се могао знатно побољшати уколико би могао да врши упоређивање потписа и анализира актуелни саобраћај. Анализом саобраћаја, свака неправилност се идентификује као напад чији потпис тек треба да се развије.

Хибридни системи управо користе оба приступа детекцији. Раде тако што прво пореде активност са ознакама познатих напада. Уколико је напад детектован, систем генерише аларм. У супротном, систем пореди активност са шаблонима нормалног понашања и у случају да открије аномалију, креира се ознака. Надаље, такво понашање се препознаје на основу потписа.

4. ДЕТЕКЦИЈА УПАДА АНАЛИЗОМ ВЕБ ЛОГОВА

Највећа корист датотека с логовима у односу на праћење читавог мрежног саобраћаја је њихова доступност и релативно једноставна анализа садржаја. Веб сервери као што је *Apache* имају укључено логовање као подразумевану опцију. Апликације обично врше неко логовање како би обез-

1 Тап типично изгледа као тропортни свич, тако да је један порт везан за *IDS*, а преостала два на мрежне свичеве. Сваки пакет који се прослеђује између свичева биће пресликан на *IDS*.

2 Спан порт се формира одговарајућим конфигурисањем свича тако да копира све пакете који су му послати са једног порта на други – на коме би био инсталиран *IDS*.

бедиле праћење својих акција. Иако мониторинг читавог саобраћај пружа додатне информације, ретко се исплати. Наиме, сакупљање мрежног саобраћаја захтева видљивост пакета и обично додатни хардвер – хабове, спан портове, тапове и сл. Сви ови уређаји морају бити купљени, инсталирани и подржани. Када се подаци сакупе, морају се претворити у одговарајући формат како би могли бити анализирани. Тек тада прикупљени мрежни саобраћај има исти облик као и логовне датотеке и спреман је за анализу.

У овом делу рада биће разматрани примери логова из локалног окружења, где је систем за управљање садржајем *WordPress* подигнут на локалној машини. За ову конфигурацију користи се *Apache* сервер, *XAMPP* окружење и *WordPress* платформа. Сви примери логова који ће бити анализирани заснивају се на садржају *access.log* фајла са локалне машине, што ће омогућити испитивање различитих аспеката логова и анализу њиховог садржаја. Уз ову конфигурацију, креирана је и апликација у програмском језику *Java* која користи регуларне изразе за детекцију *XSS* напада, *SQL* инјекције и напада извршавањем злонамерних фајлова на систем за управљање садржајем *WordPress*. Апликација је способна да детектује и *DDoS* нападе под претпоставком да 100 веб захтева са исте *IP* адресе током једне секунде означава покушај *DDoS* напада. Оба параметра су конфигурабилна. Број захтева у секунди је техника на којој се базира и детекција „brute force“ напада. Додатно се користи једна специфичност у вези са пријављивањем на администраторски панел *WordPress* платформе. Успешно пријављивање резултује редирекцијом на панел и статусним кодом 302. У случају нетачно унетих креденцијала, статусни код је 200 и корисник остаје на страници за пријављивање. Конкреан развијени систем за детекцију на основу логова сматра да је дошло до „brute force“ напада уколико уочи 15 узастопних неуспешних пријава на систем са исте *IP* адресе у току 5 секунди. Након упознавања са форматом *Apache* логова и регуларним изразима следи детаљнији опис сваког од наведених напада појединачно.

4.1. *Датотеке са веб сервер логовима*

Apache је један од најпопуларнијих веб сервера и често се користи у комбинацији са *WordPress*-ом. Бележи захтеве који стижу на сервер у стандардном формату, тзв. *CLF* (*Common Log Format*). *CLF* логови обично имају једноставну структуру где сваки ред представља један *HTTP* захтев. Сваки ред садржи информације као што су *IP* адреса корисника, идентитет корисника (ако је доступан), датум и време захтева, *HTTP* метода, *URL* ресурса на који се захтев односи, *HTTP* статусни код и величину одговора.

```
1  ::1 - - [08/Feb/2024:23:12:05 +0100] “GET /
   wordpress/wp-content/themes/blossom-feminine/
   js/all.min.js?ver=5.6.3 HTTP/1.1” 200 1113926
   “http://localhost/wordpress/” “Mozilla/5.0
   (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/120.0.0.0
   Safari/537.36”
```

Део записа	Значење
`::1`	*IPv6* адреса за локалну машину. Значи да је клијентски захтев обрађен на истој машини на којој је и веб сервер.
`-`	Идентитет корисника није доступан у овом логу.
`-`	Корисничко име није доступно у овом логу.
`[08/Feb/2024:23:12:05 +0100]`	Датум и време захтева.
`“GET /wordpress/ wp-content/themes/ blossom-feminine/js/ all.min.js?ver=5.6.3 HTTP/1.1”`	*HTTP* метода захтева је *GET*. Путања ресурса указује на адресу до *JavaScript* фајла у *WordPress* теми. Верзија протокола је *HTTP/1.1*.
`200`	*HTTP* статус код који представља успешан захтев – ресурс је пронађен.
`1113926`	Величина преузетог ресурса у бајтовима.
`“http://localhost/ wordpress/”`	Адреса са које је корисник приступио ресурсу.
`“Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36”`	Информације о веб претраживачу који је корисник користио.

Табела 1: *Анализа једног веб сервер лога*

Табела 1 даје тумачење једне линије из лог фајла. Овај запис показује да је корисник приступио *JavaScript* фајлу у *WordPress* теми на својој локалној машини користећи *Google Chrome* веб прегледач. Кључна карактеристика која доприноси тумачењу логова јесте разумљивост стандардног формата бележења веб захтева. *CLF* логови су често читљиви за људе, јер користе текстуални формат и једноставну структуру. То олакшава анализу логова ручно или помоћу алата.

4.2. *Регуларни изрази*

Регуларни изрази омогућавају моћну, флексибилну и ефикасну обраду текста. Регуларни изрази су попут мини програмског језика сачињеног од образаца за описивање и парсирање текста. И сами обрасци називају се регуларним изразима (енгл. *Regular Expressions, Regex*).

Регуларни изрази доступни су у многим типовима алата, али њихова снага највише долази до изражаја у оквиру програмских језика. Циљ регуларног израза је да пронађе одређени образац у тексту.

Једноставан регуларни израз који филтрира записе из фајла *access.log* који садрже *wp-admin*, што је подразумевана административна датотека у *WordPress*-у, или *wp-login*, фајла за пријаву у *WordPress*-у (*wp-login.php*) би гласио:

```
2  wp-(admin|login).
```

Ови изрази омогућавају прецизно филтрирање и претрагу у логовима, што олакшава откривање различитих образаца и појава. Прецизно филтрирање је омогућено дефинисањем прецизних критеријума за филтрирање логова, што значајно олакшава проналажење релевантних записа. На овај начин се могу открити различити обрасци и појаве у логовима, као што су адресе, захтеви, грешке итд.

Захваљујући употреби регуларних израза могуће је аутоматизовати процесе анализе логова, што значајно штеди време и олакшава рад. Додатно, регуларни изрази су широко коришћени у различитим програмским језицима и алатима за анализу логова, што омогућава њихову употребу у различитим окружењима и платформама.

4.3. *Детекција напада који се налазе на* OWASP*-овој листи*

У овом делу рада се истражује детекција одређених напада са *OWASP* листе анализирањем лог-фајлова. Техника која је најчешће коришћена за детекцију јесте употреба регуларних израза. Напади који се користе као основа за идентификацију и анализу у овом поглављу су већ поменути у другом поглављу овог рада. Кроз ову анализу, циљ је да се идентификују обрасци и карактеристике напада како би се унапредила безбедност веб-апликација и боље разумела природа потенцијалних претњи.

4.3.1. Cross Site Scripting (XSS) напад

XSS напади се високо котирају на *CVE* листи. Ови напади функционишу тако што уграђују ознаке скрипти у *URL*-ове или *HTTP* захтеве и привлаче кориснике да на њих кликну, осигуравајући извршавање злонамерног *JavaScript*-а на машини жртве. Ови напади користе поверење између корисника и сервера и чињеницу да на серверу нема провере улаза и излаза која би одбацила *JavaScript* или друге активне кодне карактере. Једноставни напади садрже *HTML* ознаке попут <h1> или <script>. Често коришћени пример је:

```
3  <script>alert('XSS')</script>.
```

Такве *HTML* ознаке је лако препознати. Следећи регуларни израз препознаје ознаке:

```
4  (%3C|<)(%2F|\\/)*[a-zA-Z0-9%]+(%3E|>).
```

Образац	Објашњење
`(%3C|<)`	Тражи појављивање отворене заграде *HTML* тага или њене хексидецималне репрезантације (*3C*).
`(%2F|\\/)*`	Проверава косу црту за затварање тага или њеног еквивалента *2F*.
`[azA-Z09%]+`	Појављивање било ког алфанумеричког низа карактера унутар тага или хексадецимална замена у виду процента.
`(%3E|>)`	Тражи појављивање затворене заграде *HTML* тага или њене хексидецималне репрезантације (*3E*).

Табела 2: *Објашњење регуларног израза којим се препознају*

Табела 2 даје објашњење за обрасце регуларног израза који препознаје *HTML* тагове.

Корисници могу да остављају коментаре који укључују *HTML* тагове. Програмери често желе да поделе код који може да садржи заграде „<>“. Још један пример легитимног улаза би био случај када корисник жели унесе структуриране податке у *XML* формату. Очигледно, предложени регуларни израз ће открити било коју *XML*/*HTML* ознаку, укључујући било који легитимни унос корисника, што може довести до великог броја лажних позитива.

Постоје начини да се претрага сузи. Нажалост, *JavaScript* може бити укључен на још много места и ознака. Једно популарно место је *img* ознака, где корисници могу да поставе своје име датотеке. Параметар *src* ознаке *img* функционише добро као вектор *JavaScript*-а. Постоји још много *HTML* ознака где може бити укључен *JavaScript*.

Један приступ детекцији наведених напада би био једноставно проналажење назива ознаке, нпр. *img*:

```
5  (%3C|<)(%69|i|%49)(%6D|m|%4D)(%67|g|%47)
   [^\n]+(%3E|>)
```

Образац	Објашњење
`(%3C|<)`	Тражи појављивање отворене заграде *HTML* тага или њене хексидецималне репрезантације (*3C*).
`(%69|i|%49)` `(%6D|m|%4D)` `(%67|g|%47)`	Слова *img*, у различитим варијантама – *ASCII* и одговарајућим хексадецималним еквивалентима малих и великих слова.
`[^\n]+`	Било који карактер који следи иза <img, осим новог реда.
`(%3E|>)`	Тражи појављивање затворене заграде *HTML* тага или њене хексидецималне репрезантације (*3E*).

Табела 3: *Објашњење регуларног израза за детекцију тага img*

Приступ који би донео више успеха би био да се потраже сви могући изрази који могу покренути *JavaScript* или други активни код. Листа таквих кључних речи:

```
6  javascript, vbscript, expression, applet, meta,
   xml, blink, link, style, script, embed, object,
   iframe, frame, frameset, ilayer, layer, bgsound,
   title, base

7  onabort, onactivate, onafterprint, onafterupdate,
   onsubmit, onunload, ...
```

Регуларни израз којим би се детектовале неке од речи из претходне листе би изгледао овако:

```
8  (javascript|vbscript|expression|applet|script|embed
   |object|iframe|frame|frameset).
```

Међутим, чак и претраживање свих наведених израза није гаранција да ће се пронаћи сви *XSS* покушаји напада. Кључна је контекстуализација инјектовања кода. Ако се убацивање деси унутар дела кода *JavaScript*-а, није потребно користити таг или један од горе наведених израза, већ се обично може уметнути чист *JavaScript* код. Такви *XSS* напади су веома тешки за откривање.

У овом делу ће бити описан процес симулације *XSS* напада на локалну инстанцу *WordPress*-а. Циљ симулације је демонстрација детекције напада на основу веб логова. У локалној мрежи на *IP* адреси 192.168.0.193 је подигнута елек-

тронска продавница козметике *Stella*. Поље за претрагу ће бити искоришћено као улазна тачка за убацивање злонамерног кода. На тај начин се симулира понашање нападача који покушава да изманипулише пољем за претрагу.

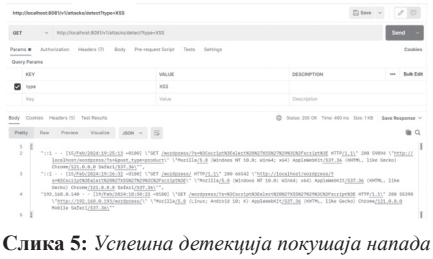
Слика 3: *Манипулација пољем за претрагу*

Слика 4: *Неуспешан XSS напад*

Слика 3 показује покушај извођења *XSS* напада са мобилног уређаја који је на истој локалној мрежи, са *IP* адресом 192.168.0.140. Слика 4 показује да је напад био неуспешан, јер се злонамерни код није извршио. То није ни био циљ. Оно што је од интереса за ову симулацију јесте да ли је покушај напада детектован. У *access.log* датотеци се појавио нови ред:

```
9  192.168.0.140 - - [19/Feb/2024:18:50:22 +0100]
   “GET /wordpress/?s=%3Cscript%3Ealert%28%27X
   SS%27%29%3C%2Fscript%3E HTTP/1.1” 200 55398
   “http://192.168.0.193/wordpress/” “Mozilla/5.0
   (Linux; Android 10; K) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/121.0.0.0 Mobile
   Safari/537.36”.
```

Слика 5 потврђује да је „систем за детекцију“ који је креиран у склопу овог рада успешно детектовао покушај *XSS* напада. За препознавање је користио први предложени регуларни израз.



Слика 5: *Успешна детекција покушаја напада*

Симулацијом се показало да последњи предложени регуларни израз, који тражи кључне речи напада, даје највише лажних позитива – *WordPress* обилује датотекама које се називају као делови регуларног израза.

4.3.2. *Инјектовање злонамерног кода*

Код чије се инјектовање врши може бити било ког типа као што су *SQL*, *LDAP*, *XPath*, *XSLT*, *HTML*, *XML* и инјектовање команди оперативног система. *XSS* је, заправо, подскуп инјектовања *HTML*-а. Овај део рада се фокусира на најраспрострањенију врсту инјектовања, *SQL* инјекцију. За рад *SQL* инјекција, нападач мора да изађе из оригиналне *SQL* наредбе. Ово се обично ради помоћу једноструког знака апострофа или двоструке цртице. Једноструки наводници служе за раздвајање *SQL* упита, а двострука цртица означава коментаре у *Oracle* и *MS SQL* базама.

```
10  (‘|%27)|()|(#|%23)
```

Образац	Објашњење
(‘|%27)	Једноструки наводник и његова *URL* енкодирана верзија.
()	Двострука цртица.
(#|%23)	Тараба и њена *URL* енкодирана верзија.

Табела 5: *Регуларни израз за препознавање SQL инјекције*

Прво се врши детекција хексадецималног еквивалента једноструког наводника, једноструки наводник или присуство двоструке цртице. Додатно, ако се користи *MySQL*, потребно је проверити и присуство знака # или његовог хексадецималног еквивалента. Дупла цртица није *HTML* мета-карактер и неће бити кодиран од стране прегледача (Табела 5). Такође, ако нападач покуша да ручно измени дуплу цртицу у енкодирану вредност *%2D*, напад *SQL* инјекције неће бити успешан.

Међутим, могуће је извести *SQL* инјекцију и без употребе наводника и цртица. У том случају претходни регуларни израз не би донео резултате. На пример, нека апликација извршава следећи *SQL* упит:

```
11  select value1, numeric_value2
12  from table1
13  where numeric_value2=user_input.
```

У овом случају нападач може да изврши додатни упит, уносећи:

```
14  3; select * from users.
```

Делује да је довољно проширити предложени регуларни израз карактером ;. Проблем је што се тачка са запетом неретко појављује у *URL*-овима:

```
15  https://example.com/wordpress/wp-login.php?action
    =login;username=admin;password=123456.
```

Детекција се може сузити препознавањем знака једнакости и проверавањем тачке са запетом само у оквиру *URL* параметара.

Злонамерни кориснички унос може изгледати овако:

```
16  https://example.com/wordpress/wp-login.php?actio
    n=login;username=admin;password=123456;select *
    from users.
```

Измењени регуларни израз којим би се препознали такви покушаји напада гласи:

```
17  (%3D|=)[^\n]*(%27|’)|(|%3B|;)
```

Образац	Објашњење
(%3D)|=)	Знак једнакости или његова енкодирана варијанта.
[^\n]*	Нула или више појављивања карактера осим новог реда.
(%27|')|(|%3B|;)	Једноструки наводник, двострука цртица, њихове енкодиране варијанте и тачка са запетом.

Табела 6: *Проширени регуларни израз за препознавање SQL инјекције*

Табела 6 даје објашњење проширеног регуларног израза – такав израз тражи појављивање знака једнакости за којим следе карактери различити од ознаке за нов ред, једноструки наводник, дупла цртица или тачка са запетом.

Још један типичан начин напада је коришћењем *SQL* кључне речи or. Пример напада може изгледати као 1' or '2'='2. Оваквих варијација постоји бесконачно много, попут 1' or 1<2. Једина константа је једноструки наводник који следи реч or. Регуларни излаз који покушава да детектује такав напад је:

```
18  [azA-Z09_]*(%27|')(\s|+|%20)*(%6F|o|%4F)
    (%72|r|%52).
```

Образац	Објашњење
[azA-Z09_]*	Нула или више појављивања алфанумеричких карактера или доње црте.
(\s|+|%20)*	Нула или више појављивања размака и његове енкодиране варијанте.
(%27|')	Једноструки наводник и енкодирани еквивалент.
(%6F|o|%4F)(%72|r|%52)	Реч *or* и енкодиране варијанте одговарајућих малих и великих слова.

Табела 7: *Регуларни израз који препознаје SQL инјекцију која укључује кључну реч or*

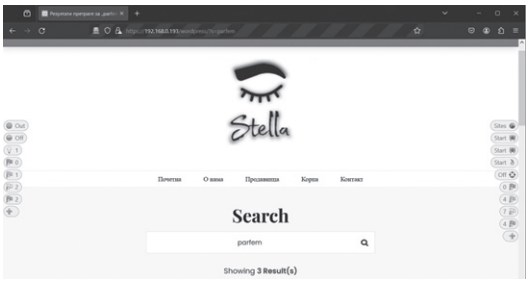
Поред кључне речи *or*, постоји још једна *SQL* кључна реч која се често користи у нападима – *UNION*. *UNION* се користи за комбиновање резултата из више *SELECT* наредби у један резултујући скуп. Нападачи могу да је користе да комбинују *SELECT* наредбу дату од стране апликације са *SELECT* наредбом коју они одреде. То омогућава нападачу читање табела различитих од оне која је наведена у наредби од стране апликације. Претходни регуларни израз може додатно да се прошири још неким *SQL* кључним речима од интереса:

```
19  (%27|')(.*)(select|union|insert|update|delete|rep
    lace|truncate).
```

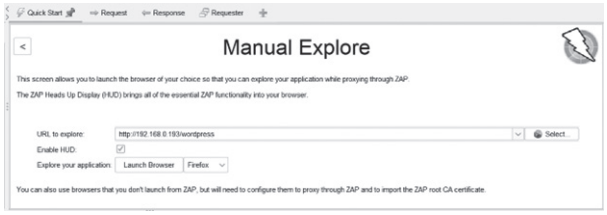
Образац	Објашњење
(%27|')	Наводник или његова енкодирана верзија.
(.*)	Појављивање било ког карактера.
(select|union|insert|update|delete|replace|truncate)	Кључне речи.

Табела 8: *Регуларни израз за препознавање кључних речи које указују на SQL нападе*

За разлику од претходне симулације злонамерног корисничког понашања, *SQL* инјекција је изведена уз помоћ *ZAP* алата који је инсталиран на локалној машини. Као улазна тачка још једном је изабрано поље за претрагу. (Слика 6) Веб претраживач *Firefox* је покренут тако да *ZAP* може да посредује у захтевима упућеним електронској продавници. (Слика 7) Тиме је омогућено скенирање веб захтева.

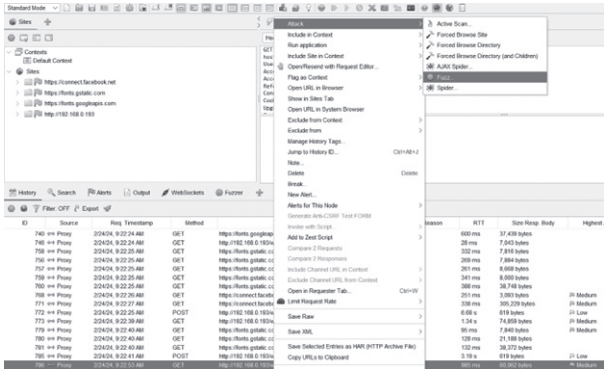


Слика 6: *Улазна тачка за напад*



Слика 7: *Покретање електронске продавнице директно из ZAP-а*

За извођење напада је коришћена опција „fuzz"[3] и база датотека и скрипти свих познатих напада – *Fuzz Files Offensive*. Слика 8 показује први корак приликом симулирања напада – одабир поменуте опције „fuzz".

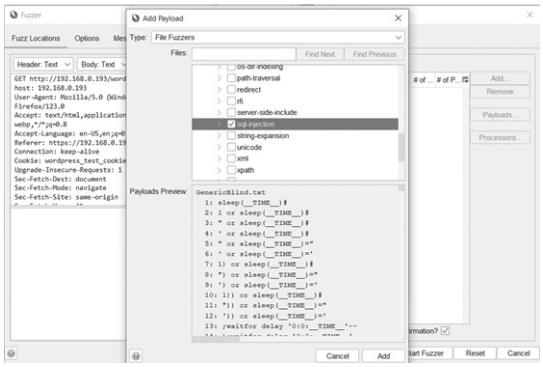


Слика 8: ZAP-ова опција „fuzz"

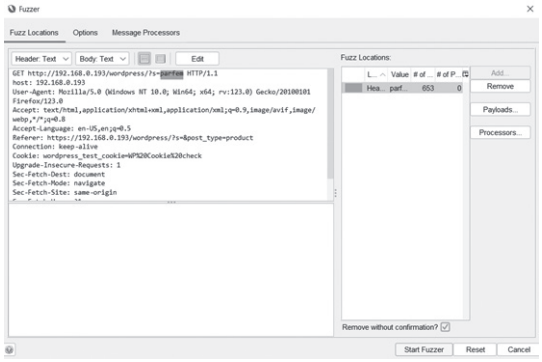
Други корак је бирање вредности параметра за претраживање производа електронске продавнице. За симулацију у овом поглављу одабрани су шаблони *SQL* инјекције. (Слика 9)

3 „Fuzz" се односи на технику тестирања познату као „fuzzing". Реч је о техници тестирања софтвера која се користи за проналажење грешака, рањивости или неисправног понашања. У контексту *ZAP*-а, „fuzzing" се обично користи за аутоматско слање великог броја различитих улазних података или захтева веб-апликацији како би се открили рањивости попут грешака у валидацији уноса, *XSS* или *SQL* рањивости. „Fuzzing" покушава генерисати неочекиване или неважеће улазе како би се видело како ће веб-апликација реаговати и јесу ли имплементиране одговарајуће сигурносне контроле.

Слика 10 показује последњи корак у извођењу напада – започињање генерисања веб-захтева са малициозним вредностима.



Слика 9: *Одабир извора вредности за улазне податке*

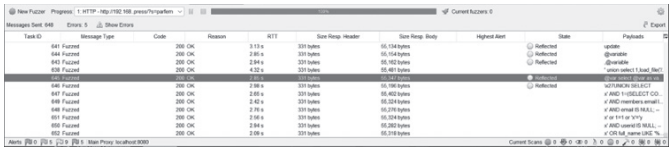


Слика 10: *Стартовање напада*

У току симулације су изгенерисана 652 злонамерна веб захтева. (Слика 11) Табела 10 приказује резултате. Сви до сада предложени регуларни изрази употребљени заједно су успели да препознају око 80% злонамерних захтева. Занимљиво је да је регуларним изразом који тражи једноструко појављивање наводника детектовано успешно више од 76% покушаја напада на датом узорку. Поставља се питање како би постојећи изрази могли да се унапреде да би се повећала успешност препознавања покушаја инјектовања *SQL* кода. Један од начина је проширивање регуларног израза додатним кључним речима попут *declare*, *procedure* и сл.

Регуларни израз	Број веб логова који одговарају изразу	Број лажних позитива
(‘|%27)|()|(#|%23)	500	2
(%3D|=)[^\n]*(%27|’)|(|%3B|;)	479	2
[azA-Z09_]*(%27|’)(\s|+|%20)*(%6F|o|%4F)(%72|r|%52)	54	0
(%27|’)(.*)(select|union|insert|update|delete|replace|truncate)	84	0
Комбинација сва четири заједно.	525	2

Табела 10: *Резултати детекције напада SQL инјекцијом*



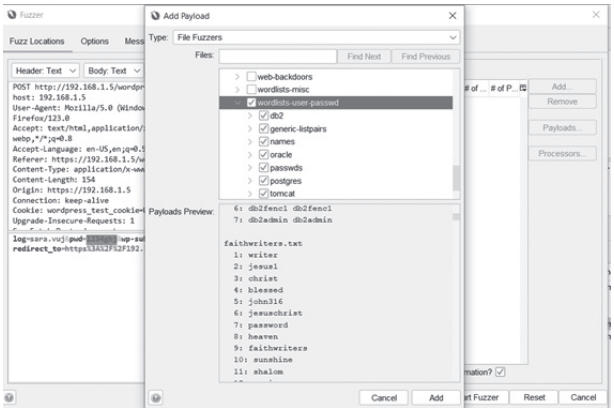
Слика 11: *Крај симулације*

Помоћу *ZAP Proxy* алата истом техником је симулирана и *SQL* инјекција на *POST* веб захтеве http://192.168.0.193/wordpress/wp-login.php. Као улазна тачка је коришћено поље за корисничко име. Међутим, ниједан од ових покушаја није детектован. Разлог је тривијалан – *Apache* сервер не логује „body“ *POST* захтева. Ови покушаји напада су детектовани као напади грубом силом.

4.3.3 *Brute force* напад

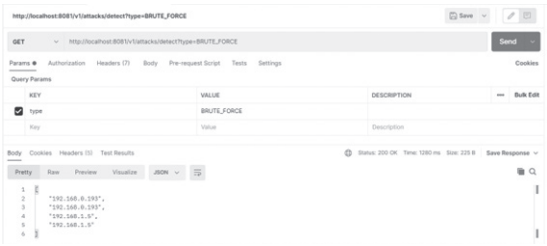
„Brute force“ напади су једна од најстаријих и најједноставнијих техника напада у свету рачунарске сигурности, али и даље веома ефикасни када се користе на одговарајући начин. Ово поглавље се бави њиховом суштином, функционисањем и на крају, детекцијом.

Напади грубом силом су облик напада који се заснива на испробавању различитих комбинација корисничких имена, лозинки или других идентификационих информација све док се не пронађе одговарајућа комбинација која омогућава приступ циљаном систему или ресурсу. Ови напади често користе аутоматизоване алате који генеришу огроман број могућих комбинација у кратком периоду. Јасан показатељ „brute force“ напада јесте велики број неуспешних покушаја пријављивања на систем за кратко време. У контексту *WordPress* платформе, када се корисник успешно пријави бива аутоматски пребачен на администраторску контролну таблу. Сервер одговара статусом 302. С друге стране, у случају нетачно унетог корисничког имена или лозинке, корисник добија као одговор статус 200 и остаје на истој страни. Систем за детекцију сматра да је дошло до напада уколико препозна 15 таквих записа узастопно у току 5 секунди са исте *IP* адресе. „Brute force“ напад је изведен уз помоћ *ZAP Proxy* алата и „fuzzing“ технике. Испробаване су различите комбинације за лозинку (Слика 12).



Слика 12: *„Brute force“ напад*

Сви покушаји напада генерисани *ZAP* проксијем су успешно регистровани системом за детекцију. (Слика 13)



Слика 13: *IP адресе с којих је изведен напад грубом силом*

4.3.4. *DDoS* напад

Дистрибуирани напади ускраћивања услуге представљају један од најозбиљнијих изазова у домену сајбер сигурности. Реч је о облику напада у ком се користи велики број рачунара или уређаја како би се истовремено послала огромна количина захтева ка циљаном серверу или мрежи. Циљ ових напада је да преоптерете циљани систем или мрежу, онемогућавајући приступ легитимним корисницима и узрокујући прекид у испоруци услуга. *DDoS* напади су озбиљна претња и за веб странице које користе системе за управљање садржајем. У контексту *CMS*-а, *DDoS* напади могу имати катастрофалне последице јер онемогућавају приступ веб страници, ометају њен нормалан рад, што за последицу има могући губитак посетилаца и клијената. Од суштинског је значаја да администратори *CMS*-а предузму одговарајуће мере како би се заштитили од *DDoS* напада. То може да укључује имплементацију *firewall*-а за филтрирање нежељеног саобраћаја, коришћење услуга које пружају хостинг провајдери, као и праћење и анализу саобраћаја како би се идентификовали и блокирали напади у реалном времену. У овом контексту, ефикасан систем за детекцију и превенцију *DDoS* напада постаје кључни део сигурносне стратегије *CMS*-а, омогућавајући очување доступности, интегритета и перформанси веб страница чак и у присуству оваквих претњи.

На основу веб логова је могуће препознати *DDoS* нападе уколико за кратко време стиже велики број захтева са исте *IP* адресе. Систем који анализира логове настао у склопу овог рада сматра *DDoS* нападом свако појављивање једне исте *IP* адресе 100 пута узастопно у току 1 секунде.

Као и случају „brute force“ напада, кроз праћење броја захтева у секунди, могуће је идентификовати сумњив саобраћај који указује на *DDoS* напад.

5. ЗАКЉУЧАК

Овим радом је пружен увид у сложеност и значај система за управљање садржајем, као и у разноврсне претње које их вребају. Истакнута је важност безбедности у контексту *CMS*-а, уз наглашавање потребе за ефикасним системима детекције и превенције напада. Анализом различитих врста напада који циљају *CMS*, укључујући *DDoS* нападе, „brute force“ нападе и друге врсте претњи, идентификовани су кључни изазови са којима се суочавају власници и администратори *CMS*-а у очувању сигурности својих система.

Систем детекције и превенције напада на *CMS* представља витални део безбедносне стратегије, а кроз класификацију различитих система, укључујући правила базирана на потписима и анализу аномалија, истакнута је разноликост приступа у борби против претњи.

На крају, представљена је апликација за анализу веб логова која користи технике обраде података и једноставне регуларне изразе ради идентификације и интерпретације покушаја напада на *CMS*. Ова апликација може бити корисна алатка за администраторе *CMS*-а у откривању претњи и доношењу информисаних одлука у циљу очувања безбедности и стабилности својих система. Од посебног значаја за примену било ког система за детекцију јесте детаљно тестирање и прилагођавање конкретним апликацјама које се штите.

ЛИТЕРАТУРА

[1] *Historical yearly trends in the usage statistics of content management systems*, Доступно на: https://w3techs.com/technologies/history_overview/content_management/all/y, [приступљено 28. 5. 2023]

[2] *2024's CMS Market Share Report – Latest Trends and Usage Stats*, Доступно на: https://www.wpbeginner.com/research/cms-market-share-report-latest-trends-and-usage-stats/, [приступљено 28. 8. 2024]

[3] Fox, M., *A Beginner's Guide To Content Management Systems For E-Commerce*, 2022, Доступно на: https://www.forbes.com/sites/forbesagencycouncil/2022/10/21/a-beginners-guide-to-content-management-systems-for-e-commerce/?sh=63ea6d371d3f, [приступљено 11. 6. 2023]

[4] Dave, V., 2023, *40+ Content Management System Statistics – Top Players & Market Share in 2023*, Доступно на: https://meetanshi.com/blog/cms-statistics/, [приступљено 11. 2. 2024]

[5] Rastogi, N., 2023, *27+ Hacking Statistics & Top Data Breaches – WordPress, Magento, Drupal, Joomla, OpenCart & Prestashop*, Доступно на: https://www.getastra.com/blog/cms/hacking-statistics/, [приступљено 24. 6. 2023]

[6] Fletcher, N., 2007, Challenges for regulating financial fraud in cyberspace. *Journal of Financial Crime*, 14, 190-207, https://doi.org/10.1108/13590790710742672

[7] DataDome, *How to Prevent Brute Force Attack with 9 Advanced Strategies*, Доступно на: https://datadome.co/bot-management-protection/how-to-prevent-brute-force-attacks/, [приступљено 6. 2. 2024]

[8] Swanagan, M., *Intrusion Detection VS Prevention Systems: What's The Difference?*, Доступно на: https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/, [приступљено 3. 2. 2024]

[9] Kaur, J., Garg, U., Bathla, G., Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review, *Artif Intell Rev* 56, 12725–12769 (2023). https://doi.org/10.1007/s10462-023-10433-3

[10] Wibowo, I., Brandl, R., *CMS Market Share Trends*, 2022, Доступно на: https://www.tooltester.com/en/blog/cms-market-share/, [приступљено 28. 5. 2023]

[11] *Content management system*, Доступно на: https://www.optimizely.com/optimization-glossary/content-management-system, [приступљено 11. 6. 2023]

[12] Fitzgerald, A., 2021, *What Is a CMS and Why Should You Care?*, Доступно на: https://blog.hubspot.com/blog/tabid/6307/bid/7969/what-is-a-cms-and-why-should-you-care.aspx, [приступљено 11. 6. 2023]
[13] Seitz A., Arnold T., *Black Hat Python 2nd edition*, No Starch Press, San Francisco, 2021
[14] Treuberg M., *7 Types of WordPress Attacks (And How To Avoid Them)*, Доступно на: https://www.cminds.com/blog/wordpress/7-types-wordpress-attacks/, [приступљено 11. 6. 2023]
[15] Wainwright, C., *Why Blog? The Benefits of Blogging for Business and Marketing*, Доступно на: https://blog.hubspot.com/marketing/the-benefits-of-business-blogging-ht, [приступљено 18. 6. 2023]
[16] Jordana, A., 2023, *Joomla vs WordPress: Which CMS Should You Use?*, Доступно на: https://www.hostinger.com/tutorials/joomla-vs-wordpress, [приступљено 24. 6. 2023]
[17] Вујиновић, А., 2018, *OWASP Топ 10 – Безбедност на интернету*, Доступно на: https://blog.imi.pmf.kg.ac.rs/owasp-top-10-bezbednost-na-internetu/, [приступљено 24. 6. 2023]
[18] *OWASP Top Ten*, Доступно на: https://owasp.org/www-project-top-ten/, [приступљено 24. 6. 2023]
[19] Agarval N.H. Syed S. Z, A Closer Look at Intrusion Detection System for Web Applications, *Security and Communication Networks*, 2018, 9601357, 27 pages, 2018. https://doi.org/10.1155/2018/9601357
[20] Чисар, П., *Систем за детекцију упада у мрежну инфраструктуру*, Криминалистичко-полицијска академија, Београд, 2013.
[21] Gillis, A.S., *Intrusion prevention system (IPS)*, Доступно на: https://www.techtarget.com/searchsecurity/definition/intrusion-prevention, [приступљено 8. 7. 2023]
[22] Dizdar A., *OWASP ZAP: 8 Key Features and How to Get Started*, Доступно на: https://brightsec.com/blog/owasp-zap/, [приступљено 8. 7. 2023]
[23] Red Hat, 2021, *What is CVE?,* Доступно на: https://www.redhat.com/en/topics/security/what-is-cve, [приступљено 8. 7. 2023]
[24] Prodromou, A., *Using Logs to Investigate – SQL Injection Attack Example*, Доступно на: https://www.acunetix.com/blog/articles/using-logs-to-investigate-a-web-application-attack/, [приступљено 10. 2. 2024]
[25] *15 Best and Most Popular CMS Platforms in 2024 (Compared)*, Доступно на: https://www.wpbeginner.com/showcase/best-cms-platforms-compared/, [приступљено 11. 2. 2024]
[26] Mookhey, K. K., Burghate N., *Detection of SQL Injection and Crosssite Scripting Attacks,* http://www.blackhat.com/presentations/bhusa04/bhus04mookhey/old/bhus04mookhey_whitepaper.pdf, Black Hat USA 2004, Las Vegas
[27] Haan, K., Smithh A.N., Holznienkemper L., *Squarespace Review: An Ideal All-Around Website Builder*, Доступно на: https://www.forbes.com/advisor/business/software/squarespace-review/#squarespace_ratings_at_a_glance_section, [приступљено 14. 2. 2024]
[28] Михајловић, Н., 2023, *Како да заштитите PHP апликације*, Доступно на: https://www.mcloud.rs/blog/kako-da-zastitite-svoje-php-aplikacije/, [приступљено 15. 2. 2024]
[29] OWASP Cheat Sheet Series, *XSS Filter Evasion Cheat Sheet*, Доступно на: https://cheatsheetseries.owasp.org/cheatsheets/XSS_Filter_Evasion_Cheat_Sheet.html, [приступљено 3. 3. 2024]

Сара Вујичић, MSc, софтверски инжењер у InSource Lab DOO
Контакт: s.vujicic@insourcelab.com
Области интересовања: софтверско инжењерство, развој микросервиса, Java

Др Марјан Милошевић, ванредни професор, Факултет техничких наука у Чачку, Универзитет у Крагујевцу
Контакт: marjan.milosevic@uni.kg.ac.rs
Области интересовања: безбедност информација, дигитална резилијентност, електронско учење

UPUTSTVO ZA PRIPREMU RADA

1. Tekst pripremiti kao Word dokument, A4, u kodnom rasporedu 1250 latinica ili 1251 ćirilica, na srpskom jeziku, bez slika. Preporučen obim – oko 10 strana, single prored, font 11.
2. Naslov, abstakt (100-250 reči) i ključne reči (3-10) dati na srpskom i engleskom jeziku.
3. Jedino formatiranje teksta je normal, bold, italic i bolditalic, VELIKA i mala slova (tekst se naknadno prelama).
4. Mesta gde treba ubaciti slike, naglasiti u tekstu (Slika1...)
5. Slike pripremiti odvojeno, VAN teksta, imenovati ih kao u tekstu, radi identifikacije, u sledećim formatima: rasterske slike: jpg, tif, psd, u rezoluciji 300 dpi 1:1 (fotografije, ekranski prikazi i sl.), vektorske slike – cdr, ai, fh,eps (šeme i grafikoni).
6. Autor(i) treba da obavezno priloži svoju fotografiju (jpg oko 50 Kb), navede instituciju u kojoj radi, kontakt i 2-4 oblasti kojima se bavi.
7. Maksimalni broj autora po jednom radu je 5.

Redakcija časopisa Info M