Sigurnost Web aplikacija, Web mjesta i Web poslužitelja

Stojanović, Matej

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:857791

Rights / Prava: Attribution 3.0 Unported/Imenovanje 3.0

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Stojanović

SIGURNOST WEB APLIKACIJA, WEB MJESTA I WEB POSLUŽITELJA

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Stojanović

Matični broj: 45924/17-R

Studij: Informacijski sustavi

SIGURNOST WEB APLIKACIJA, WEB MJESTA I WEB POSLUŽITELJA

ZAVRŠNI RAD

Mentor

Matija Kaniški, mag. inf.

Varaždin, rujan 2020.

Matej Stojanović

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U radu se objašnjava što je Web poslužitelj, Web mjesto i Web aplikacija s naglaskom na sigurnost i zaštitu podataka. Nakon toga opisuju se zakoni o zaštiti osobnih podataka na Internetu koji vrijede u Republici Hrvatskoj. Nadalje, detaljnije se opisuju najznačajniji oblici napada na Web poslužitelj, Web mjesto i Web aplikacije i mjere obrane od istih. Zatim bi se govorilo o ranjivosti koja nije objavljena u bazi otkrivenih ranjivosti i programima nagrađivanja za otkrivene ranjivosti. Spominju se i opisuju osobine sigurnosnih timova koji se bave probijanjem i testiranjem sigurnosti raznih sustava i Web aplikacija. Usporedit će se razni operacijski sustavi i njihove distribucije koji se koriste u praksi za sigurnosno testiranje. Također, u radu se opisuje korištenje virtualnog okruženja s najpopularnijim operacijskim sustavom za provođenje sigurnosnih testiranja. Analizirat će se i testirati alati za provjeru sigurnosti Web poslužitelja, Web mjesta i Web aplikacija. Rad se sastoji i od praktičnog dijela u kojem će se realizirati sigurna Web aplikacija u odabranom programskom jeziku.

Ključne riječi: sigurnost Web aplikacija, sigurnosni napadi, mjere obrane od napada, otkrivenja ranjivost, sigurnosni timovi, alati, provjera sigurnosti, sigurna Web aplikacija

Sadržaj

1. Uvod ... 1
2. Metode i tehnike rada ... 2
3. Web aplikacija, Web mjesto i Web poslužitelj ... 4
   3.1. Zaštita osobnih podataka ... 4
   3.2. Operacijski sustavi za provođenje sigurnosnih testiranja ... 6
      3.2.1. Kali Linux ... 6
      3.2.2. BackBox ... 7
      3.2.3. Parrot Security ... 7
      3.2.4. DEFT Linux ... 7
      3.2.5. Samurai Web Testing Framework ... 8
   3.3. Sigurnosni timovi ... 8
      3.3.1. Crveni tim ... 9
      3.3.2. Plavi tim ... 9
      3.3.3. Ljubičasti tim ... 10
   3.4. Zero Day ranjivost ... 11
   3.5. Baza otkrivenih ranjivosti ... 11
   3.6. ZERODIUM program nagrađivanja ... 13
4. Web aplikacija „Matix“ ... 16
   4.1. Uloga neregistrirani korisnik ... 17
   4.2. Uloga registrirani korisnik ... 18
   4.3. Uloga moderator ... 19
   4.4. Uloga administrator ... 20
5. Zaštita Web aplikacije Matix od OWASP top 10 napada ... 21
   5.1. SQL ubrizgavanje ... 21
      5.1.1. Napad SQL ubrizgavanjem ... 22
      5.1.2. Zaštita od SQL ubrizgavanja ... 25
   5.2. Prekinuta provjera autentičnosti ... 27
      5.2.1. Napad na prekinutu provjeru autentičnosti ... 27
         5.2.1.1. Izvršavanje brute force napada korištenjem alata Burp Suite ... 27
      5.2.2. Zaštita od prekinute provjere autentičnosti ... 31
   5.3. Otkrivanje osjetljivih i ranjivih podataka ... 33
      5.3.1. Napad na otkrivanje osjetljivih i ranjivih podataka ... 33
      5.3.2. Zaštita od otkrivanja osjetljivih i ranjivih podataka ... 34
   5.4. XXE ... 35

5.5. Prekinuta kontrola pristupa 37
5.5.1. Napad prekinutom kontrolom pristupa 37
5.5.2. Zaštita od prekinute kontrole pristupa 38
5.6. Neispravno implementirana sigurnosna konfiguracija 39
5.6.1. Napad neispravno implementiranom sigurnosnom konfiguracijom 39
5.6.2. Zaštita od neispravno implementirane sigurnosne konfiguracije 41
5.7. XSS 42
5.7.1. XSS napad 43
5.7.2. Zaštita od XSS napada 44
5.8. Nesigurna deserijalizacija 46
5.8.1. Napad nesigurnom deserijalizacijom 47
5.8.2. Zaštita od nesigurne deserijalizacije 48
5.9. Korištenje komponenata s već poznatim ranjivostima 48
5.9.1. Napad na otkrivanje komponenata s već poznatim ranjivostima 49
5.9.2. Zaštita od korištenja komponenata s već poznatim ranjivostima 50
5.10. Nedovoljna kvaliteta nadzora i vođenja zapisa o prijavi i aktivnosti korisnika 50
5.10.1. Zaštita od nedovoljne kvalitete nadzora i vođenja zapisa o prijavi i aktivnosti korisnika 50
6. Zaključak 52
Popis literature 54
Popis slika 59
Popis tablica 61

1. Uvod

U današnjem svijetu, Internet broji preko 4.8 milijarde aktivnih korisnika te nešto više od 1.7 milijardi aktivnih web stranica, što predstavlja ogromnu količinu podataka, što javno dostupnih, što privatnih i osjetljivih [1], [2]. Sve prethodno navedene podatke je potrebno kvalitetno zaštiti i onemogućiti neželjenim korisnicima pristup do njih jer mogu biti privatnog i osjetljivog sadržaja, a posjedovanje istih može dovesti do potencijalnih problema. Dakle, vrlo važna stvar je sigurnost podataka jer dolazi do komuniciranja, odnosno razmjene podataka između korisnika i Web poslužitelja, što može narušiti privatnost i sigurnost korisnika ukoliko dođe do zlouporabe te je shodno tome potrebno zaštititi podatke i svakog korisnika na Internetu učiniti sigurnim. U daljnjem radu će se detaljno razjasniti razlika između Web poslužitelja, Web mjesta i Web aplikacije. Zbog enormne količine podataka s kojima Web aplikacije svakodnevno barataju, važno je očuvanje integriteta i sigurnosti tih podataka. Nažalost, postoje ljudi nečasnih namjera koji se pokušavaju domoći raznoraznih podataka te ih koristiti za preprodaju, ucjenu i korištenje van korisnikovog znanja. Stoga je bitno znati izraditi sigurnu Web aplikaciju, spriječiti najčešće oblike napada na Web aplikacije, Web mjesta i Web poslužitelje, odgovorno upravljati osobnim podacima korisnika te profesionalno i savjesno reagirati ukoliko se napad dogodi. Za provođenje sigurnosnih testiranja nad Web aplikacijama, koriste se razni operacijski sustavi, koji će se u radu prikazati i objasniti. Spomenut će se i razni sigurnosni timovi za provođenje sigurnosnih testiranja nad raznim sustavima i Web aplikacijama, kao i neotkrivene ranjivosti koje mogu predstavljati veliki problem sigurnosnim stručnjacima i Web aplikacijama, kao i krajnjim korisnicima. Također će se obraditi i poglavlje baze otkrivenih ranjivosti u kojoj je moguće pronaći veliki broj otkrivenih ranjivosti te na taj način se detaljnije educirati u području sigurnosti Web aplikacija. Kao motivacija za provođenje sigurnosnih testiranja i pronalazaka otkrivenih ranjivosti, postoje razni programi nagrađivanja, od kojih će se prikazati i obraditi trenutno najpoznatiji. Kao praktični dio završnog rada, izradit će se sigurna Web aplikacija – Sustav za prikupljanje sredstva za projekte i upravljanje projektnim timom. Detaljnije će se prikazati i objasniti najčešći napadi na Web aplikacije, Web mjesta i Web poslužitelje, kao i obrana od istih nad Web aplikacijom Matix.

2. Metode i tehnike rada

Metode i tehnike koje su korištene pri izradi ovog završnog rada su analiza, sinteza te evaluacija. Podaci su prikupljeni i proučavani putem prezentacija i video materijali iz kolegija „Web dizajn i programiranje“, znanstvenih članaka i radova, video materijala s YouTubea, literature s Web-a, udžbenika iz Uvod u baze podataka te knjige Web Security, Privacy and Commerce – Second edition.

Za izradu sigurne Web aplikacije korišteni su sljedeći programski jezici i biblioteke:

- HTML5
- CSS3
- JavaScript
- jQuery 3.4.1
- AJAX
- PHP
- SQL

HTML5, CSS3, JavaScript, jQuery 3.4.1. i AJAX su korišteni za izradu i programiranje Web aplikacije na strani korisnika, dok je programski jezik PHP korišten za programiranje na strani poslužitelja. Strukturni upitni jezik SQL je korišten za manipuliranje podataka u relacijskoj bazi podataka.

Alati koji su korišteni pri izradi sigurne Web aplikacije su sljedeći:

- XAMPP
- PicPick
- Paint
- NetBeans IDE 8.2

PicPick i Paint su korišteni za manipulaciju i obrađivanje slika. XAMPP je korišten za konfiguriranje i spajanje na lokalni poslužitelj. NetBeans IDE je razvojno okruženje koje je korišteno prilikom izrade Web aplikacije „Matix“.

Tehnologije koje su se koristile za sigurnosna testiranja:

- Kali Linux
- Burp Suite
- VirtualBox
- Sqlmap
- nmap
- Oracle VM VirtualBox
- OWASP ZAP
- Wireshark

Kali Linux je jedan od najpopularnijih, najomiljenijih i najboljih operacijskih sustava za sigurnosno testiranje, a zasniva se na Debian - Linux distribuciji. Koristio se u završnom radu za provođenje sigurnosnih testiranja. Sam operacijski sustav dolazi s velikim brojem ugrađenih alata za provođenje sigurnosnih testiranja, a sqlmap i nmap su neki od njih koji su se koristili prilikom izrade ovog završnog rada. Kali Linux je vrlo jednostavan i intuitivan za korištenje, a za njegovu instalaciju potrebno je svega desetak minuta vremena.

Oracle VM VirtualBox je alat za virtualizaciju koji nudi mogućnost da na jednom fizičkom računalu bude pokrenuto više operacijskih sustava odjednom. Moguće je na primjer imati kao primarni operacijski sustav Windows 10, a preuzimanjem i uvažanjem gotovih slika (engl. *Images*) operacijskog sustava na VirtualBox-u pokrenuti Kali Linux i/ili Linux te ostale operacijske sustave. Dakle, moguće je pokrenuti „bezbroj“ operacijskih sustava putem VirtualBox-a. Razlog zašto je „bezbroj“ u zagradi je taj da ako korisnik ima velik broj istovremeno pokrenutih operacijskih sustava, može, i često će doći do prekida normalnog rada zbog iskorištavanja čitave radne memorije koju alat za virtualizaciju troši. Samo preuzimanje i uvažanje gotovih slika u VirtualBox je poprilično brzo i jednostavno. Burp Suite je alat korišten za simuliranje brute force napada na Web aplikaciju. OWASP ZAP je alat korišten za analizu i otkrivanje ranjivosti Web aplikacija, dok je Wireshark alat korišten za snimanje mrežnog prometa.

3. Web aplikacija, Web mjesto i Web poslužitelj

Često dolazi do miješanja pojmova Web aplikacija, Web mjesto i Web poslužitelj. U nastavku teksta će se ukratko objasniti razlika između navedenih pojmova.

Web poslužitelj je udaljeno računalo na kojem je instalirana programska podrška koja omogućava primanje HTTP zahtjeva od klijenta i vraćanje istih HTTP odgovora klijentu [3], [4]. Drugim riječima, „poslužuje“ sadržaj klijentu, koji ne mora znati previše o tome što se zbiva u pozadini jer je upravo za to zadužen poslužitelj. Neki od najzastupljenijih i najkorištenijih web poslužitelja u današnjici su: Apache, IIS (Microsoft), LiteSpeed Web Server, nginx, GWS (Google).

Web aplikacija je računalni program koji se izvršava na poslužitelju i generira Web stranice i dokumenta, a na njega se može istovremeno spojiti više korisnika koristeći pritom internetske preglednike (Internet Explorer, Mozilla Firefox, Google Chrome, Opera…) [3], [4]. Glavna svrha web aplikacija je preuzimanje podataka od korisnika i njihovo spremanje (najčešće u bazu podataka) te pretraživanje i prikazivanje korisnicima istih. Neke od najpoznatijih Web aplikacija su Facebook, Twitter, Flickr, YouTube, Netflix.

Web mjesto je logički dio na Web poslužitelju koji je određen svojom jednoznačnom Web adresom, a sastoji se od skupa povezanih Web stranica i dokumenata [3], [4]. Najčešće je realizirano na način da postoji početna stranica, koja je povezana s ostalima putem hiperveza i može im se pristupiti klikom.

Web aplikacije, Web mjesta i Web poslužitelji sadrže velike količine podataka, od kojih su velika većina osjetljivi i osobni podaci, čije bi razotkrivanje i padanje u ruke napadačima i trećim zlonamjernim stranama moglo naštetiti korisnicima.

3.1. Zaštita osobnih podataka

Živimo u svijetu u kojem se svake minute tehnologija sve više i više razvija. Prilikom posjeta raznih društvenih mreža ili Web trgovina, potrebno je unijeti osobne podatke kako bi mogli izvršiti transakciju ili koristiti društvenu mrežu. Osobni podaci mogu biti jako senzitivni i stoga ih je potrebno kvalitetno zaštititi. Svakih 39 sekundi u svijetu se dogodi jedan hakerski napad, a više od 90% napada u baze podataka dogodi se u vladinim sektorima i tehnološkoj industriji [5]. Na slici ispod su prikazani registrirani napadi koji se događaju uživo. Žutim kvadratićem je prikazano izvorište napada, a zelenim krugom odredište napada. Sa slike broj 1 je vidljivo kako je na današnji dan izvršeno 589622 napada diljem svijeta!

Slika 1. Napadi uživo diljem svijeta [6]

Krađa identiteta, ucjenjivanje ili izrada lažnog korisničkog računa s ciljem širenja laži je postala svakodnevnica i može predstavljati ozbiljan sigurnosni problem [7], [8].

Pravila Europske Unije jamče zaštitu osobnih podataka svaki put kada se oni prikupljaju, pohranjuju ili obrađuju [9].

Zaštita osobnih podataka i poštovanje privatnog i obiteljskog života jedne su od temeljnih prava propisane Poveljom Europske unije o temeljnim pravima [10]. Stoga je na razini Europske unije, zbog jačanja istih, s ciljem izjednačenja svih zakona vezanih uz zaštitu osobnih podataka i privatnosti donesena Opća uredba o zaštiti podataka (engl. *General Data Protection Regulation*, GDPR) koja se od 25. svibnja 2018. primjenjuje u svim državama članicama Europske unije.

Čl. 2. GDPR-a propisano je kako bi se načelima i pravilima o zaštiti pojedinaca u vezi s obradom njihovih osobnih podataka trebala poštovati njihova temeljna prava, posebno pravo na zaštitu osobnih podataka, neovisno o nacionalnosti ili boravištu pojedinca. Također, Uredbom se želi doprinijeti uspostava područja slobode, sigurnosti i pravde, te gospodarske unije, gospodarskom i socijalnom napretku, jačanju i približavanju gospodarstava na unutarnjem tržištu te dobrobiti pojedinaca [11].

Uredbom se obuhvaćaju gospodarski subjekti s poslovanjem na teritoriju EU te gospodarski subjekti koji raspolažu podacima europskih građana, bez obzira što su izvan Europske unije, te pojedinci sada imaju više kontrole po pitanju upravljanja i dijeljenja svojih osobnih podataka s trećim stranama.

GDPR-om su predviđene i visoke kazne za kršenje odredaba iste, te se prekršitelj može kazniti do 20 milijuna eura ili s do 4% ukupnog godišnjeg prometa, ovisno o tome što je veće.

Time je Europska unija jasno dala do znanja svim gospodarskim subjektima kolika je važnost zaštite osobnih podataka, te da će učiniti sve kako bi spriječila neprovođenje Uredbe [11].

Pojedinac privolom daje jasnu potvrdu kojom izražava dobrovoljan, informiran i nedvosmislen pristanak na obradu osobnih podataka koji se odnose na njega (npr. označivanje predviđenog polja kvačicom prilikom posjete internetske stranice). Također, svaka obrada podataka mora biti zakonita i poštena, a informacija i komunikaciju u vezi s obradom osobnih podataka mora biti razumljiva i lako dostupna, te se pojedinca mora obavijestiti i o mogućnosti povlačenja privole. Uredbom se između ostaloga propisuje i pravo pojedinca na ispravak netočnih osobnih podataka, pravo pojedinca da se usprotivi donošenju automatiziranih pojedinačnih odluka, pravo na prigovor obrade osobnih podataka, na ograničenje određene obrade te pravo na pristup osobnim podacima [12].

3.2. Operacijski sustavi za provođenje sigurnosnih testiranja

U praksi se koriste razni operacijski sustavi za provođenje sigurnosnih testiranja. Prema listi za 2020. godinu, top pet operacijskih sustava za „etičko hakiranje“ i sigurnosna testiranja su [13]:

1. Kali Linux
2. BackBox
3. Parrot Security
4. DEFT Linux
5. Samurai Web Testing Framework

Gotovo svih top pet operacijskih sustava za provođenje sigurnosnih testiranja se temelje na Ubuntu-u i Linuxu, a nijedan na Windowsu. Razlog tomu je da stručnjaci za provođenje sigurnosnih testiranja preferiraju operacijski sustav Linux i slične njemu jer su optimiziraniji za provođenje testiranja, sadrže razne ugrađene alate te nemaju toliko restrikcija kao Windows. U nastavku će se operacijski sustavi ukratko objasniti.

3.2.1. Kali Linux

Kali Linux je jedan od najpopularnijih, najomiljenijih i najboljih operacijskih sustava za sigurnosno testiranje, a zasniva se na Debian - Linux distribuciji. Osmišljen i dizajniran je za računalnu forenziku i penetracijsko testiranje. Redovito se ažurira i dodaju se najnoviji alati za razna provođenja testiranja i probojnosti.

Može se pokrenuti koristeći razne alate za virtualizaciju (VMWare, Virtual Box…) ili se direktno instalirati na neku od particija na računalu. Kali Linux je vrlo jednostavan i intuitivan za korištenje, a dolazi s velikim brojem ugrađenih alata za testiranje sigurnosti i probojnosti Web aplikacija, bežičnih mreža, sakupljanja podataka, manipuliranja bazama podataka i tako dalje [13], [14].

3.2.2. BackBox

BlackBox je distribucija za Linux koja je utemeljena na Ubuntu-u. Glavna svrha operacijskog sustava BlackBox je asistiranje u etičkom hakiranju i provođenju sigurnosnih testiranja. Ažuriranja samog sustava se provode redovito kako bi distribucija bila stabilna i u koraku s vremenom. Sama distribucija u sebi sadrži preko 70 ugrađenih alata za testiranje sigurnosti Web aplikacija, razne mrežne analize, procjene ranjivosti i provođenje računalne forenzike. Tvorci distribucije su se trudili dizajnirati ju na način da bude što intuitivnija i jednostavnija za korištenje, stvarajući na taj način odlično korisničko iskustvo. Također sama distribucija potiče i omogućuje korisnicima da kreiraju svoje vlastite programske podrške koje mogu podijeliti s drugim korisnicima [13], [15].

3.2.3. Parrot Security

Parrot Security je operacijski sustav koji je temeljen na Debian GNU/Linux, zajedno s Frozenbox operacijskim sustavom i Kali Linux-om. Odličan je za procjenu, analizu i testiranje sigurnosti, a također pruža i anonimnost pri korištenju Web-a. Koristi radno okruženje koje korisnicima pruža jednostavno i intuitivno korištenje grafičkog korisničkog sučelja, kao i za sigurnosno testiranje. Parrot Security je najpoznatiji po svojoj usluzi za podršku, internetskoj zajednici, kao i visokoj prilagodljivosti korisnicima. Operacijski sustav je visoko optimiziran, što znači da radi besprijekorno brzo čak i na zastarjelom sklopovlju (engl. *Hardware*) [13], [16]. Također je besplatan i može ga se preuzeti na službenoj stranici.

3.2.4. DEFT Linux

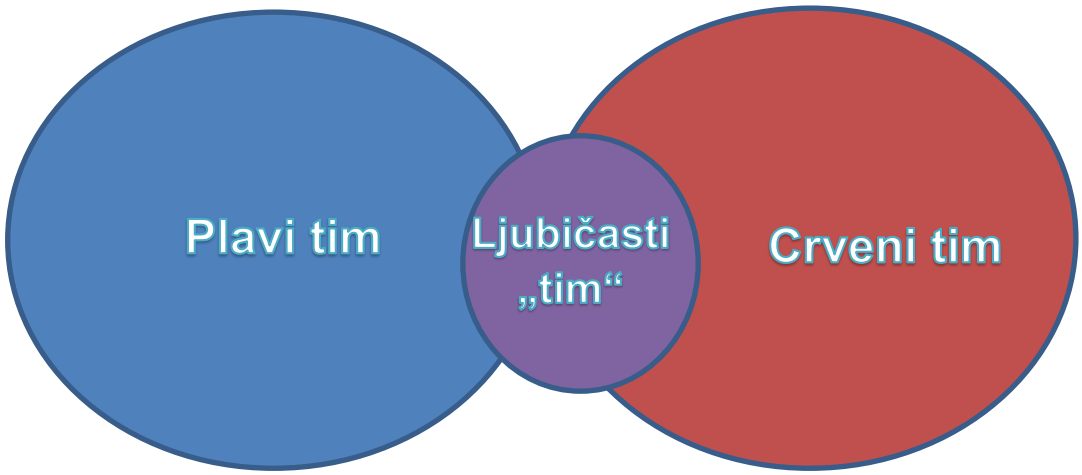
DEFT (engl. *Digital Evidence and Forensics Toolkit*) je operacijski sustav temeljen na GNU Linuxu, a glavna prednost su mu već ugrađeni mnogobrojni alati za forenziku i kvalitetno reagiranje na napad. Sadrži preko 100 kvalitetnih alata za testiranje probojnosti i računalnu forenziku. Također je besplatan, a koristi ga široki spektar korisnika, sve od etičkih hakera, vladinih službenika pa sve do vojske koja ga koristi za provođenje raznih analiza sustava temeljenih na podacima dobivenih računalnom forenzikom. Također, sadrži alate i za analizu i sigurnosno testiranje mobilnih uređaja [13], [17].

3.2.5. Samurai Web Testing Framework

Samurai Web Testing Framework je virtualno okruženje koje se primarno orijentira na sigurnosno testiranje raznih Web aplikacija. Razvili su ga Kevin Johnson, Justin Searle i Frank DiMaggio, poznati stručnjaci za sigurnosna testiranja. Samurai Web Testing Framework se temelji na Ubuntu 9.04, otvorenog koda je te se redovito ažurira zajedno s novim alatima kako bi bio što aktualniji i pristupačniji. Dolazi s ugrađenim popularnim alatima za sigurnosno testiranje kao što su Fierce Domain Scanner (služi za lociranje IP adrese te raznih domena), Maltego (za skupljanje i pronalaženje raznih podataka), WebScarab (za manipuliranje HTTP/HTTPS zahtjevima), Ratproxy (za detektiranje raznih napada na Web aplikacije) i slični [13], [18], [19].

3.3. Sigurnosni timovi

Postoje tri vrste sigurnosnih timova koji se bave probijanjem i testiranjem sigurnosti raznih sustava i Web aplikacija [20]. U nastavku će se navesti i detaljno opisati zadaće pojedinog sigurnosnog tima. Na slici ispod teksta, pod brojem 2, su prikazana tri sigurnosna tima.



Slika 2. Sigurnosni timovi

Na slici pod brojem 2 su prikazani sigurnosni timovi. Crveni tim crvenom bojom, plavi tim plavom bojom i ljubičasti tim u sredini ljubičastom bojom. Ljubičasti tim je u sredini zbog toga što poprima osobine i crvenog i plavog tima. U daljnjem nastavku rada će se detaljno objasniti točna uloga ljubičastog tima. Ukratko, crveni tim napada, plavi tim se brani, a ljubičasti „tim“ nadgleda i balansira snage između crvenog i plavog tima (u daljnjem nastavku će se spomenuti što može predstavljati ljubičasti tim).

3.3.1. Crveni tim

Crveni tim predstavlja „loše momke“, odnosno napadače, koje tvrtka najčešće angažira kako bi im napali, odnosno testirali probojnost raznih sustava i Web aplikacija [20]. Drugim riječima, tvrtka plaća ljudima da ju pokušaju hakirati, odnosno napasti. Možda na prvu zvuči zbunjujuće zašto bi netko to radio, ali to je način na koji se otkriva ranjivost sustava i aplikacija neke tvrtke. Kada se otkriju potencijalne ranjivosti, sljedeći zadatak je spriječiti da se one ne dogode nikada.

Najčešće su crveni timovi sastavljeni od eksperata u napadanju i razbijanju sustava te nisu zaposleni unutar te tvrtke pa nemaju predznanja o samom načinu funkcioniranja sustava što bi potencijalno moglo donijeti prednost pri napadanju. Ono što crveni tim radi je napadanje raznih sustava i aplikacija svom snagom, dakle pokušavaju na sve moguće načine probiti sustave, ući u njih i dohvatiti razne osjetljive podatke i manipulirati njima. Često znaju i testirati same zaposlenike tvrtke na način da se predstave kao neka treća osoba, pokrenu komunikaciju i pokušaju iz samog zaposlenika izvući podatke koji bi im mogli koristiti pri probijanju. Naravno, ne mora sve biti digitalno [21], [22].

Postoji način testiranja koji se naziva „fizički upad“, a čija glavna karakteristika je fizičko testiranje i probijanje postojećih kontrola fizičke sigurnosti te otkriti njihove slabosti. Neki od primjera fizičkog upada su: onesposobljavanje alarma i kamera, provaljivanje raznih sigurnosnih vrata, testiranje protupožarnih i ventilacijskih sustava. Sve ovo navedeno zahtijeva vrlo detaljno analiziranje sustava/aplikacija i razvijanje plana prije samog napadanja. Svaki korak od planiranja pa sve do samog izvođenja napada se detaljno i kvalitetno dokumentira te predaje tvrtki kako bi se otkrivene ranjivosti mogle što efikasnije otkloniti.

Naravno, crveni tim i tvrtka pri angažiranju potpisuju ugovor u kojem piše da crveni tim sve otkrivene ranjivosti, podatke i informacije za vrijeme napada neće nikome otkriti, prodati i na bilo koji način ugroziti tvrtku. Cijena angažiranja crvenog tima za testiranje jako varira o kompleksnosti sustava i aplikacija koje neka tvrtka koristi pa se tako cijena može kretati za male subjekte od par tisuća dolara pa sve do nekoliko stotina tisuća dolara za velike subjekte [21], [22], [23].

3.3.2. Plavi tim

Plavi tim je potpuna suprotnost crvenom timu jer oni predstavljaju „dobre momke“. Najčešće su sudionici plavog tima zaposlenici same tvrtke koji su eksperti za sigurnost sustava i zadatak im je redovno testirati sigurnost sustava, nadograđivati ga i ne dopustiti probijanje istog. Drugim riječima, ne smiju „izgubiti“ od crvenog tima.

Ukoliko dođe do proboja sustava, zadatak im je obraniti se od istoga te ne dopustiti da se taj proboj ponovno dogodi. Plavi tim mora razmišljati „izvan kutije“ te po mogućnosti, predvidjeti potencijalne ranjivosti sustava, spriječiti događanje istih i steći iskustvo kako reagirati u situaciji koja će se možda dogoditi jednoga dana u stvarnom svijetu. U cijeloj ovoj „igri“, ako se može tako nazvati, ključna je konstantna komunikacija između crvenog i plavog tima jer u pravilu igraju za istu tvrtku [21], [22], [23].

3.3.3. Ljubičasti tim

Zamislimo da se igra jedna specifična humanitarna nogometna utakmica. Crveni tim u crvenim dresovima okuplja najbolje svjetske napadače, dok plavi tim u plavim dresovima okuplja najbolje svjetske obrambene igrače. Crveni tim stalno napada plavi tim. Utakmica ne bi bila atraktivna ako crveni tim zabije 15 golova, kao i da ne može sklopiti tri kvalitetna dodavanja i zabiti niti jedan gol. Upravo zbog te moguće neravnomjernosti u igru dolazi ljubičasti tim. Zamislimo da je ljubičasti tim trener obje momčadi koji će koordinirati i upravljati timovima te tako stvoriti jednu kvalitetnu utakmicu. Dakle, kada dođe do loše komunikacije ili do pretjerane dominantnosti crvenog ili plavog tima, potrebno je dovesti u kontrolu i koordinirati rad oba tima kako bi cijelo testiranje bilo što kvalitetnije.

Ukoliko crveni tim dominira i probije sustave bez problema, a plavi tim se iznenadio i ne zna kako se obraniti i spriječiti daljnje potencijalne napade, onda je to loša utakmica kako se spomenulo prije i ne pridonosi razvitku i poboljšanju sigurnosti tvrtke ni na koji način. Upravo zato je važna komunikacija i razmjena znanja, podataka i informacija između crvenog i plavog tima.

Važno je za naglasiti da ljubičasti tim nije „tim“ u istom smislu kao plavi/crveni tim kojeg čine veći broj eksperata te u tome postoje dva slučaja. Prvi slučaj je da stvarno postoji jedan ili više koordinatora koji će predstavljati ljubičasti „tim“, a to se odvija u situacijama u kojima je stvarno enormna razlika između kvalitete crvenog i plavog tima. U ovom slučaju su ti koordinatori zapravo trener na nogometnoj utakmici u primjeru.

Drugi slučaj je metafora u kojoj „ljubičasti tim“ zapravo predstavlja kvalitetnu, dobronamjernu i redovitu suradnju te komunikaciju između crvenog i plavog tima. Zbog toga se i upotrebljava sintagma „ljubičasti tim“ jer miješanjem crvene i plave boje dobije se ljubičasta. U toj suradnji, eksperti se međusobno nadopunjuju u znanju te uče jedan od drugog i u biti tako tvore jedan „tim“ koji funkcionira kako treba [21], [22], [23].

3.4. Zero Day ranjivost

Zero day ranjivost je noćna mora svakog stručnjaka za sigurnost Web aplikacija. Razlog tomu je što je zero day ranjivost naziv za sve neotkrivene ranjivosti i propuste, odnosno one koje nisu poznate programerima i kibernetičkim stručnjacima. Napadač, koji zapravo jedini zna za te ranjivosti, može ih vrlo lako iskoristiti na način da razvije i ubaci zlonamjerni kod u samu Web aplikaciju.

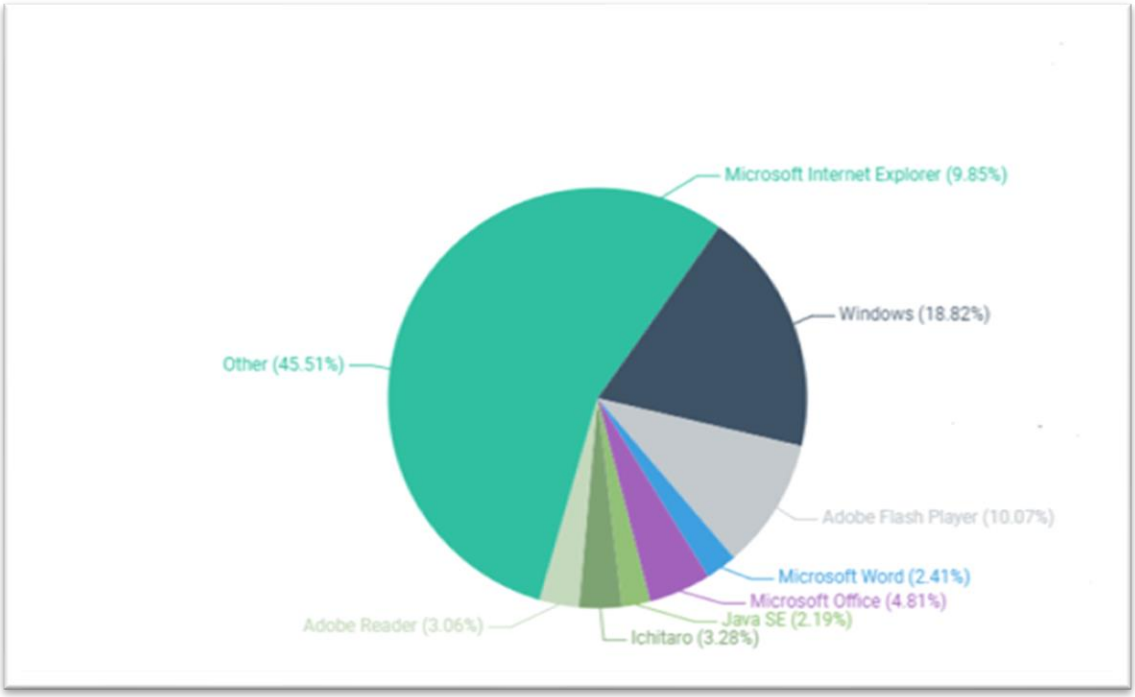
Sam naziv „zero day“ je oznaka vremenske ranjivosti [23]. Dakle, taj napad se javlja isti dan (nulti dan) ili dan prije nego što odgovorni ljudi na Web aplikaciji saznaju za ranjivost i uspiju popraviti nastali sigurnosni propust. Dakle, nula dana se „zna“ za postojanje te ranjivosti. Postoji dobra i loša strana traženja zero day ranjivosti. Dobra strana je traženje ranjivosti kako bi se ona što prije neutralizirala i uklonila, a loša strana je traženje ranjivosti kako bi se ona iskoristila i tako nanijela šteta. Važno je za napomenuti kako ove ranjivosti vrlo opasne jer uopće ne postoji nekakav „protuotrov“ s kojim bi se one mogle otkloniti u što kraćem roku.

Uzmimo u obzir jedan primjer za lakše razumijevanje pojma zero day ranjivost. Recimo da osoba ima sef u kojem drži sve svoje vrijedne materijalne stvari i neke osobne podatke. Taj sef, kada se čekićem udari u jedan vijak pod točnim kutem, otvori se. Za tu ranjivost ne zna ni osoba koja posjeduje sef, a ni proizvođači samog sefa. Za tu ranjivost zna jedino napadač koji je nekako uspio saznati za to [23].

3.5. Baza otkrivenih ranjivosti

Baza otkrivenih ranjivosti (engl. *Exploit Database*) je jedna posebna baza u kojoj se nalazi znatan broj otkrivenih ranjivosti i ranjivih softvera [24], [25]. Ova baza je sjajan izvor za sve koji se bave testiranjem probojnosti ili istraživanjem ranjivosti. Cilj ljudi koji održavaju bazu otkrivenih ranjivosti je skupiti što više otkrivenih ranjivosti na jedno mjesto i učiniti ih dostupnima kako bi se poboljšala svijest o sigurnosti sustava, ali i pružio izvor za sve zainteresirane za etičko testiranje probojnosti, bilo profesionalno, bilo u slobodno vrijeme.

Ispod teksta, na slici pod brojem 3, prikazan je graf vezan za bazu ranjivosti.

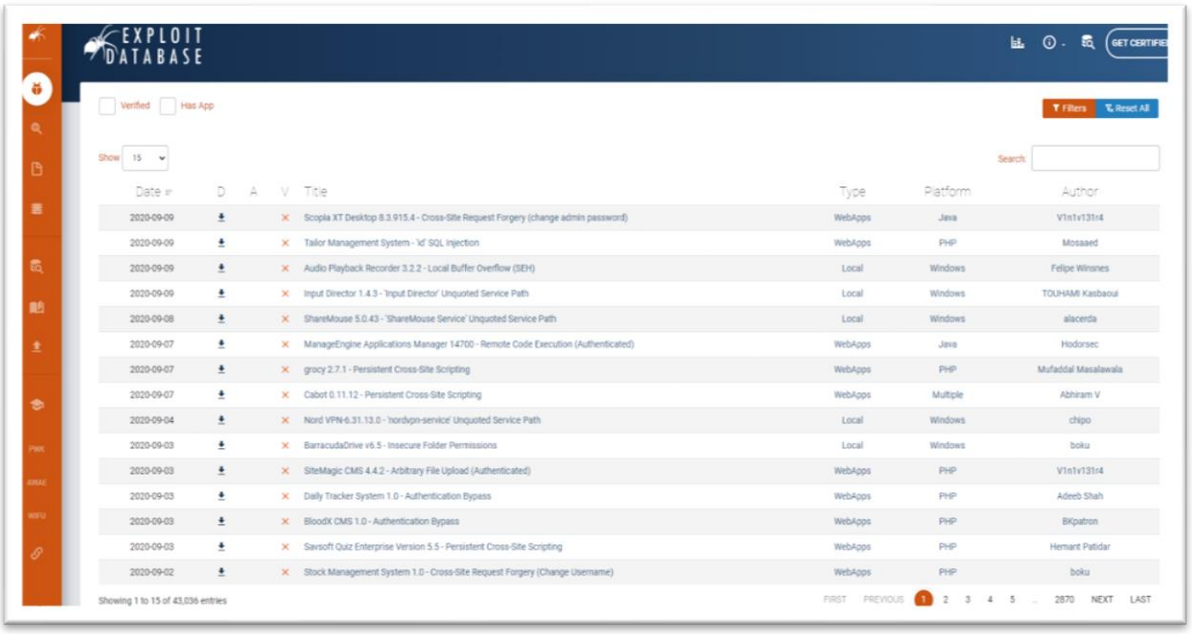


Slika 3. Postotak zastupljenosti softvera u bazi ranjivosti [23]

Na slici pod brojem 3 se nalazi graf na kojem su prikazane zastupljenosti pojedinog softvera u bazi ranjivosti od 2006. do 2020. godine. Vidljivo je kako daleko prednjači operacijski sustav Windows sa svojim propustima i ranjivostima te je upravo zbog toga često na meti (još i dan danas) napadača. Neke od najpoznatijih baza otkrivenih ranjivosti su Exploit DB, CIRCL (*Computer Incident Response Center Luxembourg*), VulDB i Rapid7 [24], [25].

Exploit DB je jedna od najpopularnijih i najdetaljnijih baza otkrivenih ranjivosti. Sadrži preko 7 000 otkrivenih ranjivosti, a svaka od njih se može preuzeti i time dobiti detaljan izvještaj o samoj ranjivosti, kao i primjer koda iste. Najstarija otkrivena ranjivosti datira iz davne 1988 godine. Postoje razni parametri po kojima se baza može sortirati i pretraživati, sve od datuma, naziva ranjivosti, platforme (Windows, Linux, ASPX, iOS, XML, Python,…), tipu ranjivosti te vrsti ranjivosti (CSRF, XSS, XXE, DoS,…). Važno je za napomenuti kako se Exploit DB redovito ažurira i sadrži najaktualnije otkrivene ranjivosti [26].

Jedna od otkrivenih ranjivosti koja se nalazi u Exploit DB bazi je „WordPress Plugin Database Backup < 5.2 Remote Code Execution“ za verzije starije od 5.2. Navedena ranjivost vezana je uz napad *command injection*, koji omogućuje napadaču proširivanje funkcionalnosti Web aplikacije izvršavanjem zlonamjernih naredbi kao što su brisanje tablica u bazi podataka. Na slici pod brojem 4 je prikazana početna stranice baze ExploitDB.



Slika 4. Početna stranica ExploitDB [26]

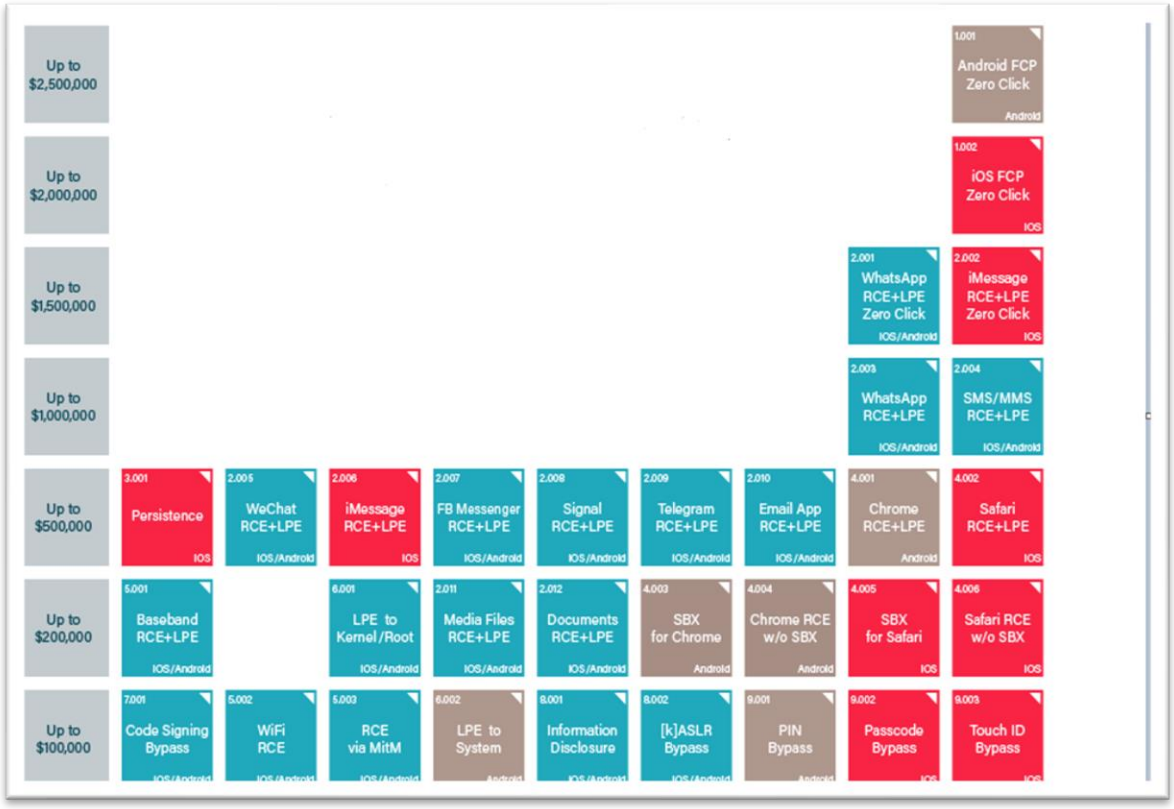
Kao što je vidljivo na slici pod brojem 4, ExploitDB se redovito ažurira pa tako skoro svaki dan objavljuju novootkrivene ranjivosti uz pripadno detaljno objašnjenje istih. Klikom na neku ranjivost, otvara se detaljan izvještaj koji sadrži autora otkrivene ranjivosti, tip ranjivosti, datum otkrivanja, platforma na kojoj se ona otkrila, pripadni ranjivi izvorni kod, kao i objašnjenje same ranjivosti.

3.6. ZERODIUM program nagrađivanja

ZERODIUM je vodeća svjetska online platforma za nagrađivanje zero-day ranjivosti. Nagrade su često vrlo visoke kako bi stimulirali što više testiranja probojnosti i samim time pridonijeli otkrivanju raznih ranjivosti i propusta. ZERODIUM se fokusira na visokorizičnim ranjivostima te nude najveću nagradu u iznosu od 2 500 000 američkih dolara. Trenutno imaju programe nagrađivanja za operacijske sustave, Web preglednike, Web poslužitelje, mobitele pa čak i za usmjernike (engl. *Router*) [27]. Na slici ispod teksta, pod brojem 5, nalazi se program nagrađivanja za stolna računala i poslužitelje.

Slika 5. ZERODIUM program nagrađivanja za računala/poslužitelje [27]

Na slici pod brojem 5, ljubičastom bojom su prikazane ranjivosti vezane uz Windows operacijski sustav, narančastom bojom ranjivosti vezane uz macOS, plavom bojom ranjivosti vezane uz Linux/BSD, a tamno žutom bojom ranjivosti vezane u sve ostale operacijske sustave. Otkrivene ranjivosti na dnu se smatraju manje „kritičnim“ i „vrijednima“ pa su nagrade za njih nešto niže, počevši od 10 000 američkih dolara. Vidljivo je kako su najveće nagrade vezane uz otkrivanje ranjivosti kod Microsoftovog servera IIS (engl. *Internet Information Services*), Web preglednika Chrome-a i otkrivanje Windows RCE (engl, *Remote Code Execution*) sigurnosnih propusta [27]. Na slici ispod teksta, pod brojem 6, prikazan je program nagrađivanja za mobilne uređaje.



Slika 6. ZERODIUM program nagrađivanja za mobilne uređaje [27]

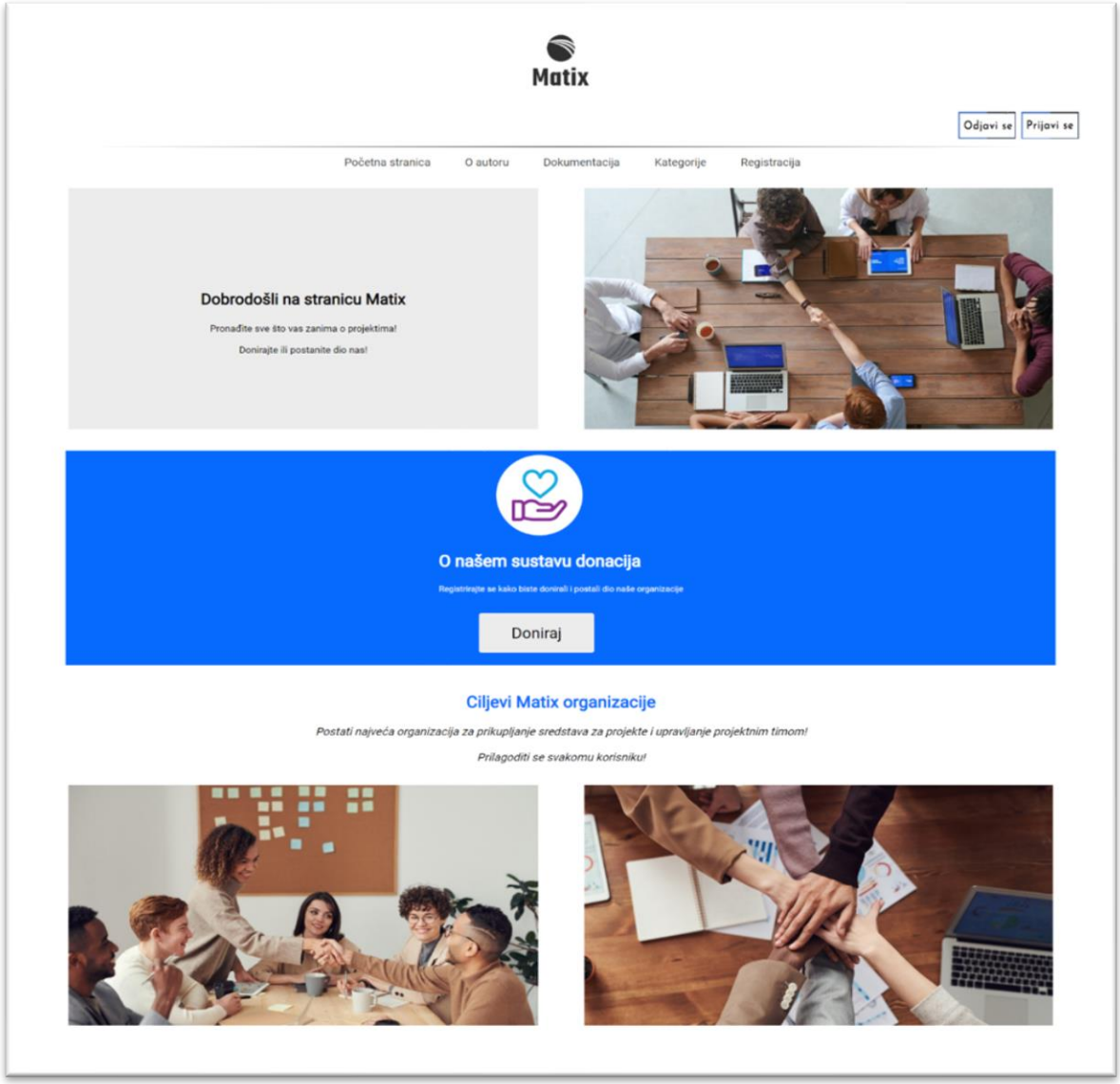
Na slici pod brojem 6, crvenom bojom su vezane ranjivosti uz iOS operacijski sustav, smeđom bojom ranjivosti vezane uz Android mobilni operacijski sustav, a plavom bojom svi ostali mobilni operacijski sustavi. Vidljivo je kako se ovdje nagrade kreću nešto višim iznosima pa je najmanja nagrada do 100 000 američkih dolara za pronalazak ranjivosti vezanih uz probijanje PIN-a (engl. *Personal Identification Number*), lozinki, otisaka prstiju te bežičnog umrežavanja. Najveće nagrade vezane su uz otkrivanje ranjivosti na iOS i Android mobilnim uređajima, a vezani su uz takozvane *zero-click* napade. Zero-click je specifična vrsta napada u kojima nije potrebna interakcija žrtve kako bi se zlonamjerni kod počeo izvršavati, od toga i potječe naziv „zero“, jer nije potrebno kliknuti ništa [27], [28].

Vidljivo je kako su programi nagrađivanja za otkrivanje ranjivosti primamljivi i nude veliku svotu novaca te upravo to može predstavljati jedan od razloga za motivaciju za učenje o sigurnosti Web aplikacija, Web mjesta i Web poslužitelja.

4. Web aplikacija „Matix“

Web aplikacija „Matix“ je napravljena u svrhu ovog završnog rada, a služi za prikupljanje sredstva za projekte i upravljanje projektnim timom. Moguće donirati određeni iznos željenom projektu te mu tako pripomoći da ostvari minimalni iznos za njegovu realizaciju, odnosno početak izvođenja.

Ostale funkcionalnosti Web aplikacije se realiziraju kroz četiri postojeće uloge: neregistrirani korisnik, registrirani korisnik, moderator i administrator. U daljnjem nastavku će funkcionalnosti biti detaljno prikazane za pojedinu ulogu, kao i potencijalne ranjivosti Web aplikacije zajedno s primjerima zaštite od istih.



Slika 7. Početna stranica Web aplikacije „Matix“

Na slici pod brojem 7 je prikazana početna stranica Web aplikacije Matix. Na njoj se nalaze osnovne informacije o aplikaciji, kao i poveznice do obrazaca za prijavu, registraciju i donaciju projektu, kao i popis kategorija i osnovnih podataka o autoru same aplikacije.

4.1. Uloga neregistrirani korisnik

Neregistrirani korisnik je korisnik koji nema korisnički račun na sustavu. Ukoliko želi korisnički račun, potrebno se registrirati i unijeti tražene osobne podatke. Ima na uvid popis kategorija projekata koji nisu skupili minimalan iznos za njegov početak realizacije.

Također vidi galeriju svih videa nekog projekta odabirom na kategoriju u koju taj projekt spada. Posljednja funkcionalnost Web aplikacije kojoj ima pristup je pretraživanje projekata i filtriranje po datumu završetka, kao i sortiranje po prikupljenom/minimalnom potrebnom iznosu donacija.

Na slici pod brojem 8 prikazan je popis aktivnih projekata.

Pretrazi projekte | Traži | Osvježi

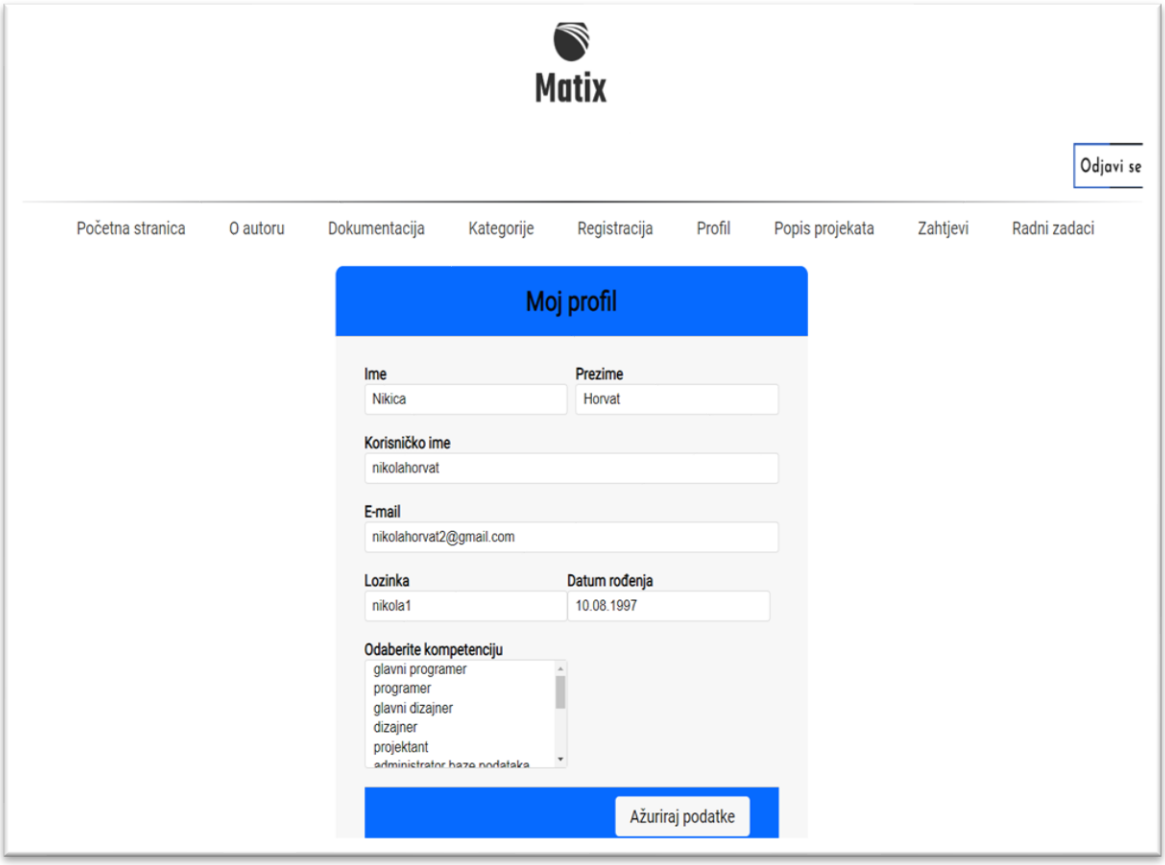
Naziv projekta	Opis	Datum početka	Datum završetka	Prikupljeni iznos	Minimalno potrebni iznos
GTA 6	Izrada novog GTA serijala koji ?e se zvati GTA 6	10/04/2020	20/12/2023	115731	58000000
Last of Us 3	Izgradnja zadnjeg nastavka serijala Last of Us	24/04/2020	27/03/2023	24578500	1400000
Becarac	Izrada web aplikacije koja ce sadrzavati poznate slavonske becarce. Takodoer ce sluziti i za dijeljenje istih medu korisnicima.	13/05/2020	08/01/2021	10965	120000
FestivaliClick	Izrada web stranice koja ce sadrzavati informacije o najpoznatijim svjetskim filmskim festivalima.	22/07/2020	10/04/2021	2500	7500000
GoghWeb	objavljivanje do sada neobjavljenih slika poznatog slikara Van Gogha.	27/04/2020	27/12/2020	1056	20000000
Film Dimnjačar	snimanje filma koji ce se temeljiti na poznatoj predstavi Ljubomira Kerekesa- Dimnjacar	11/05/2021	01/04/2023	66619	1000000
Dokumentarac o Officeu	Snimanje dokumentarca o poznatoj seriji The Office (US naravno)	18/06/2020	10/04/2022	659	30000000
DinamoZG re	izrada novog web-shopa za GNK Dinamo Zagreb	10/04/2020	10/04/2021	2000600	2000000
Lunapark Matix	Gradnja lunaparka Matix u okolici Zagreba.	10/05/2021	09/09/2022	21254600	350000000
Zoološki vrt Matix	Gradnja zoološkog vrta pod nazivom "Matix" u Vara?dinu	07/04/2020	06/04/2025	7846000	740000000
Dizajniranje novog trga u Zagrebu	Tražimo kompetentne strucnjake za dizajniranje novog trga u Zagrebu na lokaciji: Ulica Ivana Rangera 147.	05/10/2021	05/07/2022	900547	90000
CS GO Source Engine 2	Prebacivanje CS GO na Source 2 engine	07/01/2020	05/01/2022	524558	4500000
Pelješki most	Izgradnja Pelješkog mosta	13/06/2016	12/06/2019	60000000	55000000
Dota 3	Nastavak popularne MOBA PC igrice Dota2	01/04/2018	10/03/2020	2000000	3200000
FC Barcelona mobilna aplikacija	Izrada mobilne aplikacije za FC Barcelona	10/06/2019	28/05/2020	34000	35000
Stambena zgrada	Postaviti temelj nove stambene zgrade	22/06/2020	09/02/2021	1654	1500000

Slika 8. Pregled aktivnih projekata i kategorija projekata

U popisu projekata, kao što je prikazano na slici pod brojem 8, moguće je i pretraživati projekte po ključnim riječima kao što su naziv, opis, datum početka i datum završetka projekta.

4.2. Uloga registrirani korisnik

Registrirani korisnik je korisnik koji ima kreiran i aktiviran korisnički račun. Potrebno se prijaviti u sustav, odnosno unijeti korisničko ime i lozinku. Korisničko ime i lozinka mogu se unijeti više puta. U slučaju uspješne prijave, kreira se korisnička sesija koja traje do odjave korisnika sa sustava. Na slici pod brojem 9 je prikazan profil registriranog korisnika.

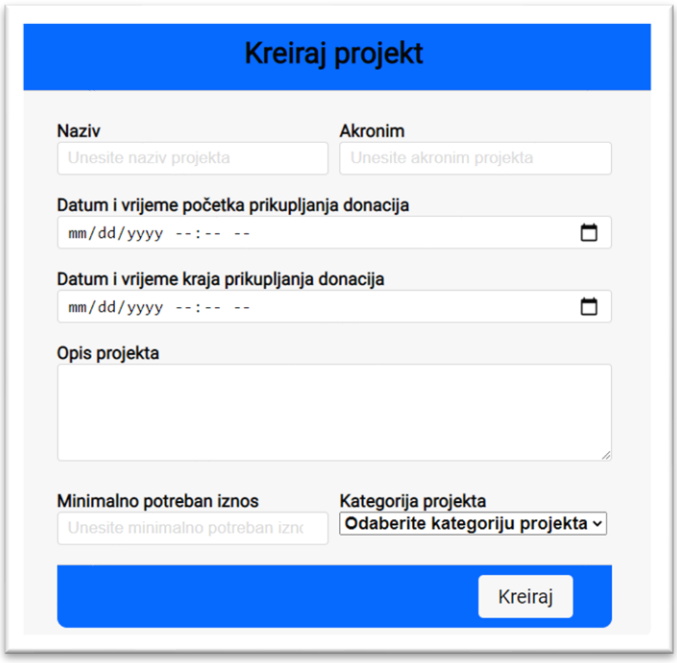


Slika 9. Profil registriranog korisnika

Registrirani korisnik ima sva prava kao i neregistrirani i još kao dodatno može ažurirati svoje korisničke podatke te odabrati kompetencije koje posjeduje. Ima na uvid popis aktivnih projekata i može donirati određeni iznos željenom projektu s popisa. Vidi zahtjeve za sudjelovanje na projektu te pojedine može ili odbiti ili prihvatiti. Ukoliko prihvati, dodjeljuju mu se radni zadaci koje je potrebno izvršiti u zadanom roku. Zadatak je završen kada korisnik postavi video rješenja na sustav te kao dodatna mogućnost, ima na odabir želi li da se video prikazuje javno ili ne.

4.3. Uloga moderator

Moderator ima sva prava kao i registrirani korisnik te još dodatno. Može kreirati projekt (slika 10), vidi popis svih korisnika grupirano po kompetencijama te može odabranom korisniku poslati zahtjev za sudjelovanje na projektu. Također, ima na uvid popis svojih projekata na kojima je on moderator (slika 11), može unositi radne zadatke za projekt ukoliko je prikupljen minimalni iznos i šalje ih željenom korisniku. Ima mogućnost pregleda statistike završenih i nezavršenih radnih zadataka u nekom razdoblju završetka zadatka.



Slika 10. Obrazac za kreiranje projekta

Kategorija	Naziv projekta	Akronim	Unesi zadatak
igrica za PC	GTA 6	GTAVI	
igrica za PS4	Last of Us 3	LoUs3	
glazba	Becarac	BeCaR	
gradevinarstvo	Lunapark Matix	MatixPark	
igrica za PC	CS GO Source Engine 2	CSsrc2	
gradevinarstvo	Pelješki most	Pmost	
igrica za PC	Dota 3	DOTA3	
gradevinarstvo	Stambena zgrada	ZgrD	

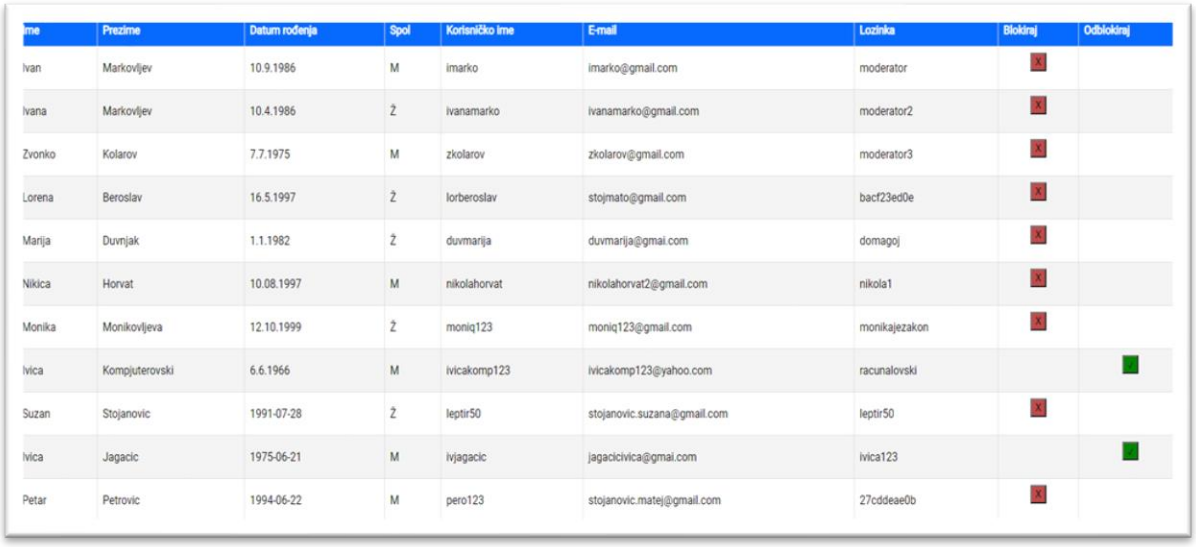
Slika 11. Moderatorovi projekti

Na slikama pod brojevima 10 i 11 su prikazani obrazac za kreiranje projekta te svi projekti kojima moderator ima pristup.

4.4. Uloga administrator

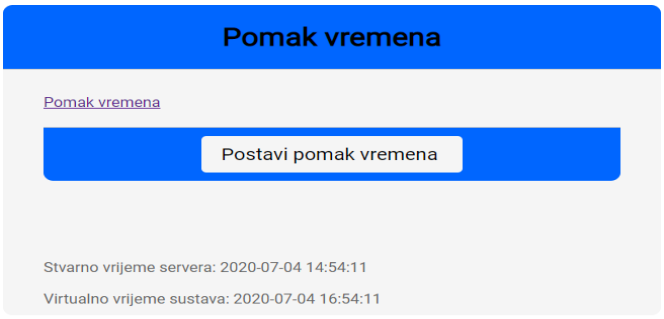
Administrator ima sva prava koja imaju prethodno definirane uloge te dodatne mogućnosti. Ima mogućnost unosa, ažuriranja i pregleda svih podataka u sustavu, uvid u dnevnik rada u kojem se bilježe aktivnosti od korisnika kao što su prijava/odjava, doniranje, kreiranje projekta i slično. Može blokirati ili odblokirati korisnika, konfigurirati trajanje kolačića, broj redaka za straničenje te napraviti pomak vremena. Također, kreira kategorije projekata te dodjeljuje moderatore toj kategoriji i kreira popis kompetencija.

Rad same Web aplikacije se temelji na virtualnom vremenu, odnosno vremenu koje polazi od stvarnog vremena na poslužitelju koje se korigira (pomiče) za određeni pomak vremena. Tako na primjer administrator može pomaknuti vrijeme za + ili - , odnosno pomaknuti vrijeme unaprijed ili unazad. Na slici pod brojem 12 je prikazan popis korisnika.



Ime	Prezime	Datum rođenja	Spol	Korisničko ime	E-mail	Lozinka	Blokiraj	Odblokiraj
Ivan	Markovljev	10.9.1986	M	imarko	imarko@gmail.com	moderator	X	
Ivana	Markovljev	10.4.1986	Ž	ivanamarko	ivanamarko@gmail.com	moderator2	X	
Zvonko	Kolarov	7.7.1975	M	zkolarov	zkolarov@gmail.com	moderator3	X	
Lorena	Beroslav	16.5.1997	Ž	lorberoslav	stojmato@gmail.com	bacf23ed0e	X	
Marija	Duvnjak	1.1.1982	Ž	duvmarija	duvmarija@gmai.com	domagoj	X	
Nikica	Horvat	10.08.1997	M	nikolahorvat	nikolahorvat2@gmail.com	nikola1	X	
Monika	Monikovljeva	12.10.1999	Ž	moniq123	moniq123@gmail.com	monikajezakon	X	
Ivica	Kompjuterovski	6.6.1966	M	ivicakomp123	ivicakomp123@yahoo.com	racunalovski		
Suzan	Stojanovic	1991-07-28	Ž	leptir50	stojanovic.suzana@gmail.com	leptir50	X	
Ivica	Jagacic	1975-06-21	M	ivjagacic	jagacicivica@gmai.com	ivica123		
Petar	Petrovic	1994-06-22	M	pero123	stojanovic.matej@gmail.com	27cddeae0b	X	

Slika 12. Popis korisnika



Slika 13. Pomak i virtualno vrijeme

Na slici pod brojem 13 je prikazan pomak vremena koje administrator može obaviti te na taj način odrediti virtualno vrijeme na kojemu će se temeljiti rad aplikacije.

5. Zaštita Web aplikacije Matix od OWASP top 10 napada

OWASP (engl. *Open Web Application Security Project*) je online svjetska i neprofitna organizacija čiji je glavni cilj analiziranje, unaprjeđenje i informiranje korisnika o sigurnosti Web aplikacija [29]. Sastavljena je od najvećih i najpoznatijih svjetskih eksperata za kibernetičku sigurnost. Često objavljuju razne članke o sigurnosti Web aplikacija gdje navode glavne potencijalne nesigurnosti i moguće proboje, detaljno ih objašnjavaju i navode moguće mjere zaštite od istih. Također, održavaju i konferencije diljem svijeta gdje poučavaju mlade ljude u području IT sektora, ali i usavršavaju znanje onih koji su godinama već u tom području.

Svakih nekoliko godina OWASP radi istraživanja i prikuplja podatke te izbaci TOP 10 metoda napada na Web aplikacije. Zadnja ljestvica je generirana 2017 godine, a za period od 2017 do 2020. godine bi trebala izaći ljestvica krajem studenog/početkom prosinca ove godine.

OWASP top 10 napada prema [29] iz 2017.godine će biti objašnjeni kroz nadolazeća poglavlja. Izvršit će se samo oni napadi koji su izvedivi nad Web aplikacijom Matix, odnosno oni putem kojih je Web aplikacija ranjiva.

5.1. SQL ubrizgavanje

Gotovo svaka Web aplikacija koristi u pozadini bazu podataka u kojoj se nalaze svi potrebni i osjetljivi podaci za kvalitetan rad aplikacije. Upravo zbog toga je potrebno obratiti pažnju na sigurnost tih podataka u bazi. Jedan od napada na bazu podataka, a samim time i Web aplikaciju, je SQL ubrizgavanje (engl. *SQL injection*) čija je karakteristika manipuliranje upitima nad bazom podataka kako bi se promijenilo ponašanje Web aplikacije i time došlo do osjetljivih podataka [30]. Dakle, ubrizgavanje spada pod napade na Web aplikaciju.

Uzmimo u obzir jedan klasičan obrazac za prijavu korisnika. Umjesto unošenja podataka koji se očekuju od korisnika (ime, prezime, korisničko ime, lozinka,…), manipulira se upitom i dodaju se drugi nizovi znakova čije spajanje drastično mijenja rad Web aplikacije. Dakle, napadač će u polja za unos unijeti određeni niz znakova, odnosno maliciozni SQL upit, koji će poslužitelj obraditi i ukoliko nije zaštićen od ubrizgavanja, otkriti podatke napadaču koji onda može manipulirati bazom podataka na način da promijeni, izbriše ili ukrade dobivene podatke [30], [31].

Nezaštićen isječak programskog koda, iz razvojnog okruženja NetBeans, je prikazan na slici broj 14 ispod.

```
"SELECT * FROM korisnik WHERE "
 . "korisnicko_ime= 'username' "
 . "AND lozinka= 'password'";
```

Slika 14. Upit koji nije zaštićen od SQL ubrizgavanja

Zamislimo da je potrebno unijeti korisničko ime i lozinku pri ispunjavanju obrasca za prijavu. Ukoliko korisnik unese točnu kombinaciju, kreira se korisnička sesija i preusmjerava se korisnika na početnu stranicu.

5.1.1. Napad SQL ubrizgavanjem

Zanimljivo je za spomenuti kako postoje razni alati za ispitivanje probojnosti vezano uz SQL ubrizgavanje, a jedan od njih je „sqlmap“. Dakle, sqlmap je alat otvorenog koda koji automatizira proces otkrivanja napada SQL ubrizgavanjem, kao i preuzimanja i manipuliranja osjetljivih podataka iz baze koja se nalazi na poslužitelju. Alat ima vrlo jake mehanizme i moguće je upisati poveznicu do Web aplikacije te će alat automatski, ukoliko je aplikacija ranjiva, prikazati osjetljive podatke poput korisnika i lozinki čime se može manipulirati. [32], [33].

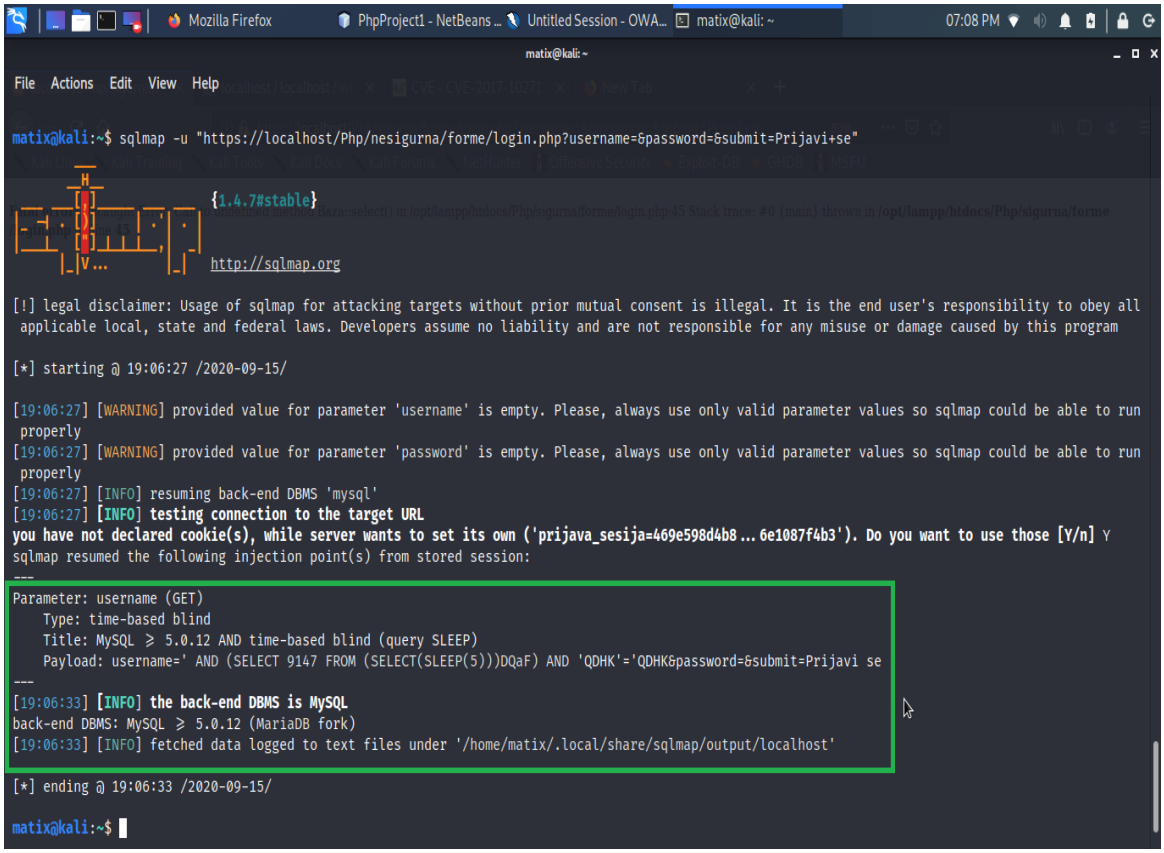
Napad SQL ubrizgavanjem će se izvesti nad obrascem za prijavu. Potrebno je u sqlmap unijeti naredbu: sqlmap -u te poveznicu do obrasca nad kojem se testira ranjivost. U ovom slučaju, ne postoji zaštita od SQL ubrizgavanja, odnosno obrazac za prijavu je ranjiv pa će alat pronaći ranjivosti, kao i pripadajući sustav za upravljanje bazom podataka koji Web aplikacija koristi (u ovom slučaju je to MySQL) te ih ispisati u novostvorenu lokalnu datoteku. Rezultat napada korištenjem alata sqlmap je prikazan na slici ispod teksta, pod brojem 15.

```
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
available databases [6]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
[*] webdip2019x123
```

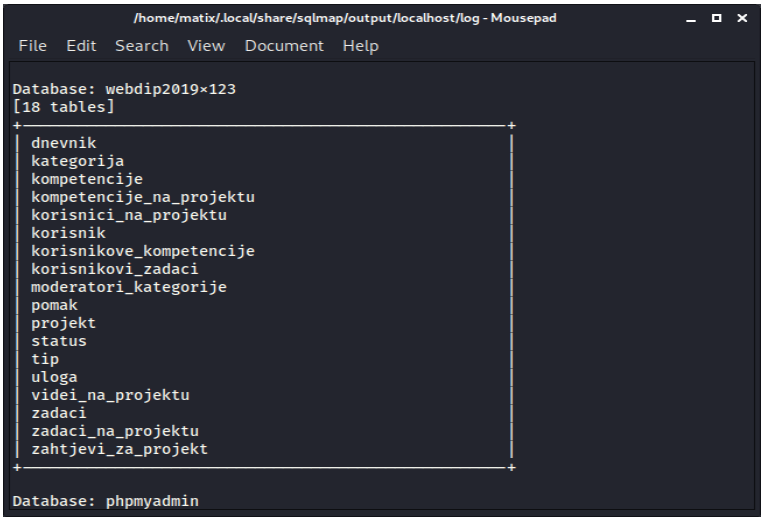
Slika 15. Otkrivene baze podataka na lokalnom poslužitelj

Dakle, već sa slike pod brojem 15 je vidljivo kako je obrazac za prijavu ranjiv, odnosno parametar username (korisničko ime) i to slijepim SQL napadom (engl. *time-based blind SQL injection*). Također, alat je prepoznao da je u pozadini sustav za upravljanje bazom podataka MySQL, što je i točno.

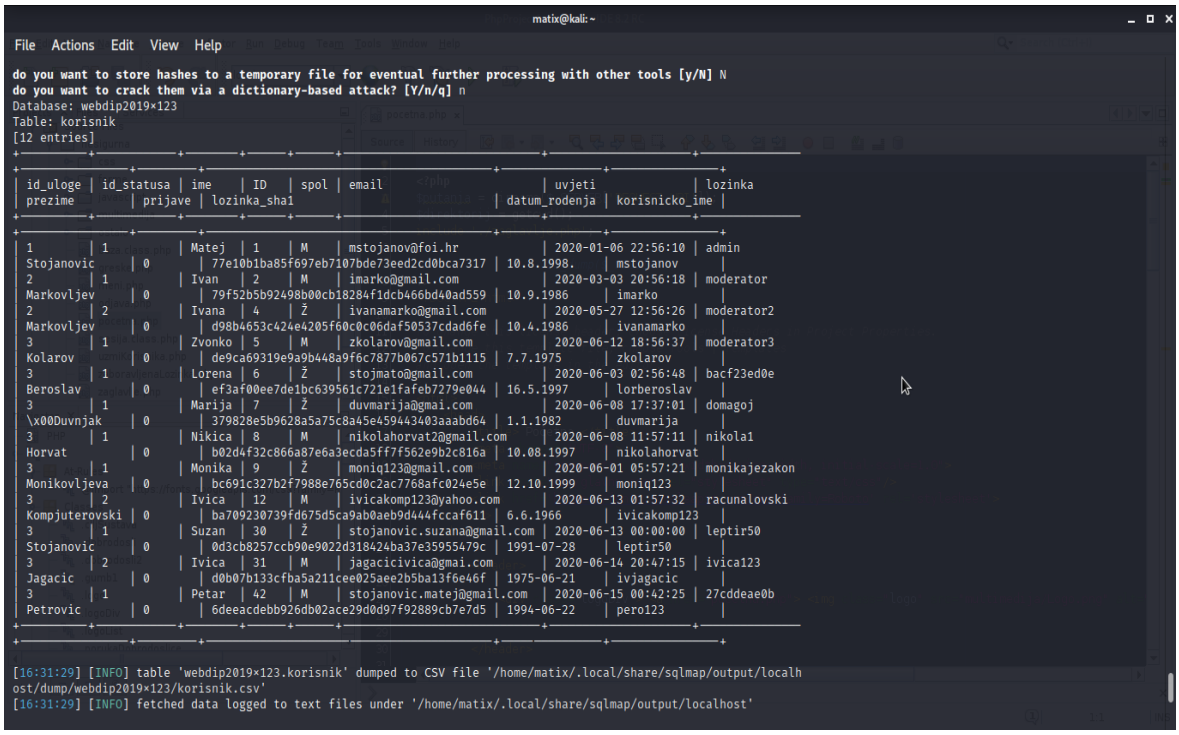
Ostali detalji su zapisani u lokalnoj datoteci, čiji je sadržaj prikazan na slikama pod brojevima 16, 17 i 18.



Slika 16. Napad korištenjem sqlmap alata



Slika 17. Baza podataka koju Web aplikacija koristi



Slika 18. Podaci zapisani u tablici „korisnik“

Dakle, kao što je vidljivo iz slike 18, ukoliko je Web aplikacija ranjiva putem SQL ubrizgavanja, sqlmap alat će to otkriti i moguće je pomoću naredbe „–dump -D webdip2019x123 -T korisnik“ ispisati čitav sadržaj tablice „korisnik“ te na taj način pristupiti i manipulirati osjetljivim podacima.

Pomoću alata sqlmap se zaključilo da je obrazac za prijavu ranjiv putem SQL ubrizgavanja. Sve što napadač treba upisati je zlonamjerni znakovni niz, a jedan od njih je „ 'OR '1'='1„ u polja za unos korisničkog imena i lozinke te će biti prijavljen u sustav kao prvi korisnik koji se nalazi u bazi podataka, a to je u ovom slučaju administrator koji ima najveću razinu ovlasti u sustavu.

Nakon unošenja zlonamjernog koda u polja za unos teksta i pritiska na gumb „Prijavi se“, na slici pod brojem 19 je prikazan rezultat napada, odnosno prijava u sustav s najvećom razinom ovlasti. Važno je napomenuti da je ovo samo jedan od niza znakova koje napadač može unijeti u polja za unos [30], [31].

Slika 19. SQL ubrizgavanje

Slika 20. Rezultat napada SQL ubrizgavanje

Na slici pod brojem 20 je prikazana aplikacija s najvećom razinom ovlasti, odnosno sa svim implementiranim funkcionalnostima.

5.1.2. Zaštita od SQL ubrizgavanja

Kako bi se zaštitili od napada SQL ubrizgavanjem, potrebno je provjeriti podatke koje korisnik unosi prilikom ispunjavanja raznih obrazaca, a to su na primjer: tip podatka, duljina, raspon vrijednosti i slično. Vrlo važno je izbjegavanje mogućnosti da SQL upiti nastaju spajanjem niza znakova, kao što je prije spomenuto. Moguće je, uz kvalitetno organiziran sigurnosti sustav, identificirati SQL napad i spriječiti ga.

U praksi se najčešće koristi ugrađena funkcija mysqli_real_escape_string() koja sprječava manipuliranje SQL naredbe i time onemogućuje spajanje znakova i tako ublažuje napad SQL ubrizgavanjem [34]. Sama funkcija prima dva argumenta od kojih je prvi konekcija na bazu podataka, a druga varijabla na koju se GET/POST zahtjev odnosi. Implementirana zaštita za obrazac za prijavu je prikazana na slici broj 21 ispod teksta.

```
$username=mysqli_real_escape_string($veza->spojiDB(),$_GET['username']);
$password=mysqli_real_escape_string($veza->spojiDB(),$_GET['password']);
$upit= "SELECT * FROM korisnik WHERE "
        . "korisnicko_ime='{$username}' "
        . "AND lozinka= '{$password}';";
$rezultat=$veza->selectDB($upit);
```

Slika 21. Mysqli_real_escape_string funkcija

Na slici pod brojem 21, varijabla veza služi za spajanje na bazu. Koristi se već otprije pripremljena klasa za rad s bazom, što znači da se ovdje poziva gotova funkcija spojiDb(); koja će se spojiti na željenu bazu. Varijable „username“ i „password“ označavaju korisničko ime i lozinku koji se upišu u obrazac prilikom prijave u sustav. Vidljivo je kako je implementirana funkcija mysql_real_excape_string koja će spriječiti manipuliranje znakovnim nizovima i ublažiti napad od SQL ubrizgavanja. Varijabla upit služi za izvršavanje SQL upita koji će dohvatiti korisnika s upisanim korisničkim imenom i lozinkom, ukoliko on naravno postoji. Varijabla rezultat izvršava upit pomoću pripremljene funkcije selectDB iz spomenute klase za rad s bazom.

Kada se zaštita implementira, kao što je prikazano na slici pod brojem 21 i pokuša se alatom sqlmap izvršiti napad SQL ubrizgavanjem, nije moguće manipulirati bazom podataka i doći do osjetljivih podataka kao što je slučaj bio prije zaštite. Rezultat napada SQL ubrizgavanjem putem alata sqlmap, kada je implementirana zaštita, prikazan je na slici pod brojem 22.

```
[19:05:24] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--ris
k' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (
e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
```

Slika 22. Napad sqlmap alatom kada je zaštita implementirana

Kao što je vidljivo na slici pod brojem 22, nakon izvršavanja napada, sqlmap alat je zaključio da niti jedan parametar na obrascu za prijavu nije ranjiv putem SQL ubrizgavanja i nikakve dodatne informacije, poput sustava za upravljanje bazom podataka koji je u pozadini, kao i tablice, nije moguće izvući. Zaključuje se kako je zaštita dobro implementirana.

5.2. Prekinuta provjera autentičnosti

Prekinuta provjera autentičnosti (engl. *Broken authentication*) je napad u kojem napadač nastoji probiti korisnički račun, a oslanja se slabo implementiran autentikacijski i sesijski dio aplikacije. Provjera autentičnosti se smatra „prekinutom“ u trenutku kada napadač dođe do osjetljivih podataka od korisnika, poput lozinka, korisničkih sesija i ostalih privatnih podataka. Jedan od mogućh načina prekidanja provjere autentičnosti je izvršavanje napada pod nazivom „brute force“ [29].

5.2.1. Napad na prekinutu provjeru autentičnosti

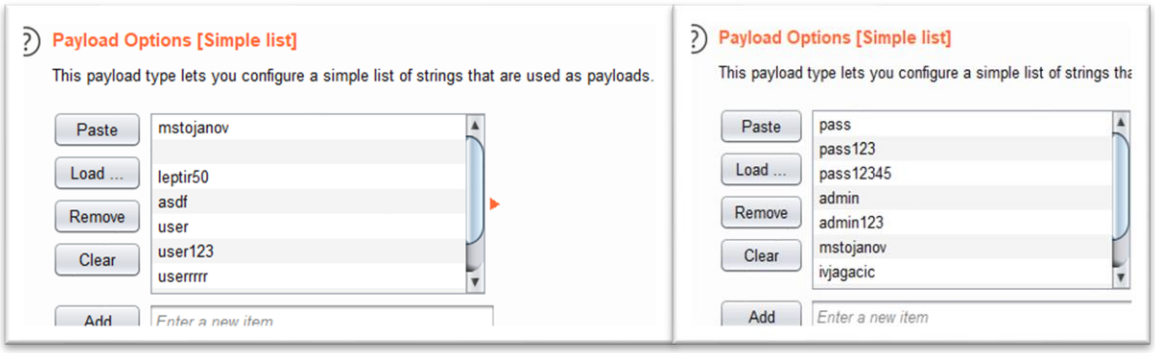
Jedan od mogućih napada na prekinutu provjeru autentičnosti je brute force napad, odnosno napad uzastopnim pokušavanjem (poznato još i kao „napad grubom silom“) je specifična vrsta napada u kojoj napadač uzastopno pokušava pronaći rješenje, odnosno lozinku, na način da prolazi kroz sve moguće kombinacije koje bi mogle činiti lozinku [35]. Za izvedbu brute force napada nije potrebno preveliko znanje kriptiranja jer se sastoji od četiri osnovna koraka koja se ponavljaju sve dok se rješenje, odnosno lozinka, ne pronađe.

Postoje razni alati za izvršavanje brute force napada, a u nastavku će se koristiti i objasniti poznati alat Burp Suite koji je namijenjen testiranju sigurnosti raznih Web aplikacija. Pomoću njega je moguće presresti sve zahtjeve koji se kreću u smjeru Web preglednika i Web aplikacije. Presretanjem tih zahtjeva moguće je izvršavati različite vrste napada na Web aplikaciju te otkrivanjem istih, iskoristiti otkrivene ranjivosti i dodatno zaštititi Web aplikaciju. Sama instalacija alata je vrlo jednostavna, a za njegovo korištenje i konfiguraciju potrebno je ponešto vremena.

5.2.1.1. Izvršavanje brute force napada korištenjem alata Burp Suite

Prije izvršavanja bilo kojeg napada i testiranja sigurnosti, u postavkama samog alata, potrebno je konfigurirati „žrtvu“, odnosno u ovom slučaju Web aplikaciju koja se izvršava na lokalnom poslužitelju kako bi alat presreo sve zahtjeve između Web preglednika i Web aplikacije. Svrha ovog napada je otkrivanje korisničkog imena i pripadne lozinke za to korisničko ime. Prije izvršavanja napada, kreirale su se dvije tekstualne datoteke. Jedna sadrži dva prava korisnička imena koja se nalaze u bazi, dok su ostala lažna kojih nema u bazi podataka. Druga datoteka sadrži prave i lažne lozinke. Nakon toga se u opcijama učitaju obje datoteke te će sav sadržaj koji se nalazi u njima Burp Suite kombinirati. Drugim riječima, isprobat će različite kombinacije za svako korisničko ime i lozinku [36]. Na slici 23 prikazani su *payload sets*, odnosno poslane podatke kojima će sam alat analizirati i kombinirati.

Važno je napomenuti kako postoje razni rječnici (engl. *Dictionaries*) koji u sebi sadrže veliku količinu lozinki koje se najčešće upotrebljavaju pa je moguće i učitati takve rječnike te s njima izvesti *brute force* napad. Neki od njih sadrže i preko milijun lozinki/korisničkih imena te izvršavanje napada pomoću njih zahtijeva nešto veću količinu uloženog vremena,truda, ali i jače računalo za izvršavanje istog.



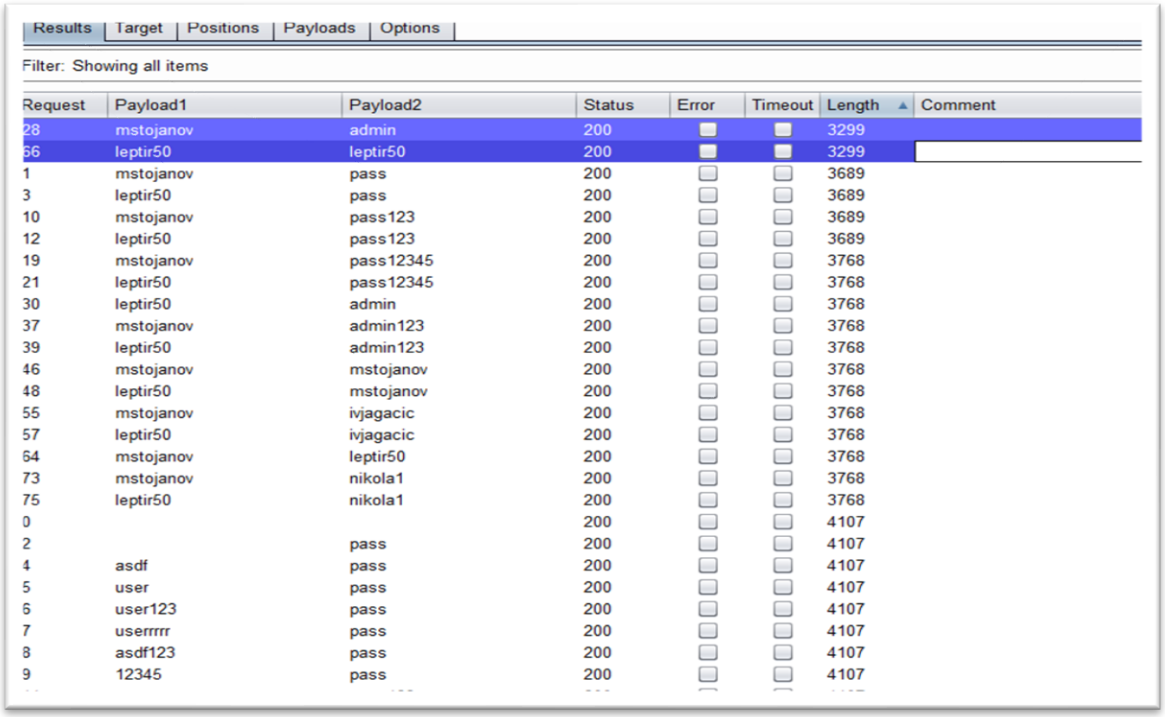
Slika 23. Korisnička imena i lozinke

Ispravne kombinacije korisničkog imena i lozinke, koji su zapisani u bazi podataka, su prikazani u tablici pod brojem 1.

Tablica 1. Kombinacija korisničkog imena i ispravne lozinke

Korisničko ime	Lozinka
mstojanov	admin
leptir50	leptir50

Sljedeći korak je pritisak na gumb „Start attack“ čime započinje napad. Napad je trajao otprilike minutu, minutu i pol. Rezultati napada su prikazani na slici pod brojem 24.



Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
28	mstojanov	admin	200			3299	
66	leptir50	leptir50	200			3299	
1	mstojanov	pass	200			3689	
3	leptir50	pass	200			3689	
10	mstojanov	pass123	200			3689	
12	leptir50	pass123	200			3689	
19	mstojanov	pass12345	200			3768	
21	leptir50	pass12345	200			3768	
30	leptir50	admin	200			3768	
37	mstojanov	admin123	200			3768	
39	leptir50	admin123	200			3768	
46	mstojanov	mstojanov	200			3768	
48	leptir50	mstojanov	200			3768	
55	mstojanov	ivjagacic	200			3768	
57	leptir50	ivjagacic	200			3768	
64	mstojanov	leptir50	200			3768	
73	mstojanov	nikola1	200			3768	
75	leptir50	nikola1	200			3768	
0			200			4107	
2		pass	200			4107	
4	asdf	pass	200			4107	
5	user	pass	200			4107	
6	user123	pass	200			4107	
7	userrrrr	pass	200			4107	
8	asdf123	pass	200			4107	
9	12345	pass	200			4107	

Slika 24. Rezultati napada

Na slici pod brojem 24, plavom bojom su istaknute ispravne kombinacije koje je Burp Suite pronašao nakon izvršavanja napada.

Ispravne kombinacije se mogu prepoznati po tome što im je duljina znatno manja od svih ostalih zahtjeva. Od ukupno 81 zahtjeva koji su bili poslani, 79 ih je bilo ili duljine 4107, 3689 ili 3768 Jedino su dvije ispravne kombinacije korisničkog imena i lozinke bili duljine 3299. Na slici pod brojem 25 potvrđuje se ispravnost unesene kombinacije jer se odmah kreira korisnička sesija, što označava ispravnu prijavu u Web aplikaciju.

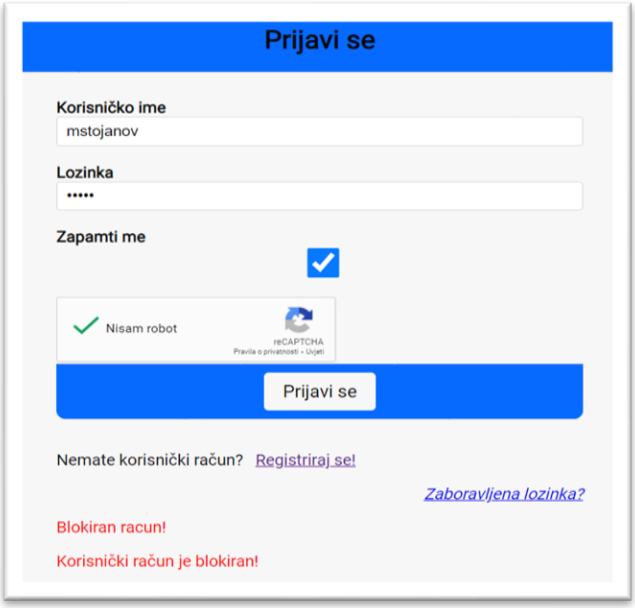
```
1  GET /PhpProject1/forme/login.php?username=mstojanov&password=admin&submit=Prijavi+se HTTP/1.1
2  Host: localhost
3  Connection: close
4  Upgrade-Insecure-Requests: 1
5  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.125
   Safari/537.36
6  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;
   v=b3;q=0.9
7  Sec-Fetch-Site: same-origin
8  Sec-Fetch-Mode: navigate
9  Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 Referer: https://localhost/PhpProject1/forme/login.php?username=mstojanov&password=admin&submit=Prijavi+se
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: prijava_sesija=okkrlitvoupda8oovgub8jqnrk
```

Slika 25. GET zahtjev nakon uspješne prijave u sustav

Zelenim pravokutnikom su označeni uneseno korisničko ime i lozinka, kao i kolačić koji se kreirao nakon uspješne prijave.

5.2.2. Zaštita od prekinute provjere autentičnosti

Zaštita je realizirana na način da nakon tri uzastopne neuspjele prijave zaredom, korisniku se blokira korisnički račun i više ne može pristupiti sustavu. Kako bi se račun odblokirao, potrebno je kontaktirati administratora koji jedini ima ovlasti za otključavanje istog, kao i ostaviti e-mail adresu s kojom je korisnički račun registriran [29]. Nakon implementacije zaštite, napadač koji je uspio pronaći korisničko ime i lozinku se ne može prijaviti s njima jer je sustav zaključao taj korisnički račun. Razlog zašto se račun zaključao je taj da je Burp Suite izvođenjem napada pokušao sve moguće kombinacije korisničkog imena i lozinke, a svaki neuspjeli uzastopni pokušaj prijave se zapisuje u bazu podataka i kada dosegne broj 3, korisnički račun se blokira. Na slici pod brojem 26 je prikazan blokiran korisnički račun dohvaćen *brute force* napadom.



Slika 26. Blokiran korisnički račun

Na slikama pod brojevima 27 i 28 je prikazan programski kod realiziran u NetBeans razvojnom okruženju za implementaciju zaštite od *brute force* napada, pa tako i prekinute provjere autentičnosti.

```
$username=mysqli_real_escape_string($veza->spojiDB(),$_GET['username']);
$upit3="UPDATE korisnik set prijave=prijave+1  where korisnicko_ime= '{$username}'";
$veza->updateDB($upit3);
```

Slika 27. Zapisivanje neuspjele prijave u bazu podataka

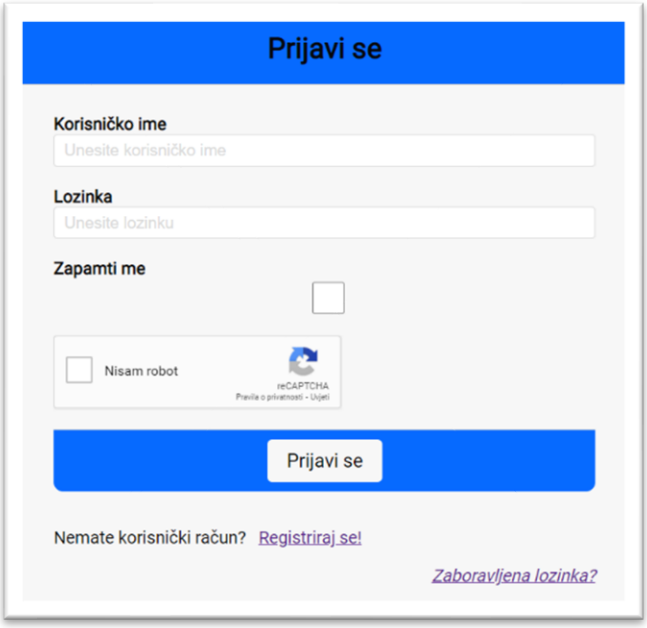
```
if($brojPrijava >= 3 || $status==2){
$greske[]="Korisnički račun je blokiran!";
$username=mysqli_real_escape_string($veza->spojiDB(),$_GET['username']);
$sql="UPDATE KORISNIK set id_statusa=2 where korisnicko_ime= '{$username}' ";
$veza->updateDB($sql);
```

Slika 28. Blokiranje korisničkog računa

Na slici 27 varijabla „$upit“ ažurira bazu podataka način da inkrementira atribut „prijave“ za uneseno korisničko ime u obrascu za prijavu. Varijabla „$veza“ primjenjuje upit i ažurira bazu podataka pomoću funkcije updateDB.

Na slici 28 u prvoj liniji koda se postavlja selekcija koja provjerava je li varijabla „$brojPrijava“ veća ili jednaka od 3 ili je $status jednak 2. Varijabla „status“ može poprimiti vrijednosti 1 ili 2, a označava je li korisnički račun aktivan ili blokiran (broj 2 označava da je korisnički račun blokiran). Ukoliko selekcija popuni istinitu vrijednost, varijabla „greske[ ]“ koja predstavlja polje će ispisati poruku na ekran da je korisnički račun blokiran. Nakon toga, u trećoj liniji koda, ukoliko je broj prijava veći od 3, blokirat će se korisnički račun, odnosno postaviti mu atribut „status“ u bazi podataka na vrijednost 2. Nakon toga varijabla „veza“ izvršava, odnosno ažurira bazu podataka pomoću funkcije „updateDB“, koja kao argument prima varijablu $sql, odnosno SQL upit.

Također, u obrascu za registraciju je implementirana Google reCaptcha kao dodatna zaštita od botova i brute force napada. Kao što je vidljivo sa slike 29 potrebno je kliknuti unutar kvadratića „Nisam robot“ kako bi se potvrdilo i mogao podnijeti obrazac za prijavu.



Slika 29. Google reCAPTCHA

Za samu implementaciju reCAPTCHA-e, potrebno je registrirati na Googleovom obrascu za korištenje reCAPTCHA-e. Nakon toga, korisnik dobije generirani ključ, a sve što treba je kopirati i zalijepiti sljedeće dvije linije koda što je vidljivo iz slike 30. Koristi se vlastiti generirani ključ [37].

```
<script src="https://www.google.com/recaptcha/api.js" async defer></script>
<div class="g-recaptcha" data-sitekey="6LfG5MQZAAAAAHNwuPkJ3hEx6eW1yN7HHXjU2P0x"></div>
```

Slika 30. Potreban kod za implementaciju reCAPTCHA-e

Još jedan način na koji se može najefikasnije zaštititi od brute force napada pa tako i napadom na prekid autentičnosti je korištenje složenih i dugih lozinke pomoću kojih će se onemogućiti probijanje i pronalaženje lozinke. Ukoliko dođe do kompromitiranja baze podataka, a lozinke nisu kvalitetno osigurane i kriptirane, napadaču se doslovno serviraju na pladnju sve lozinke od korisnika. Trenutno najsigurniji algoritmi za kreiranje sažetaka i kriptiranje su Triple Des, RSA, Blowfish, Twofish i AES [38], [39].

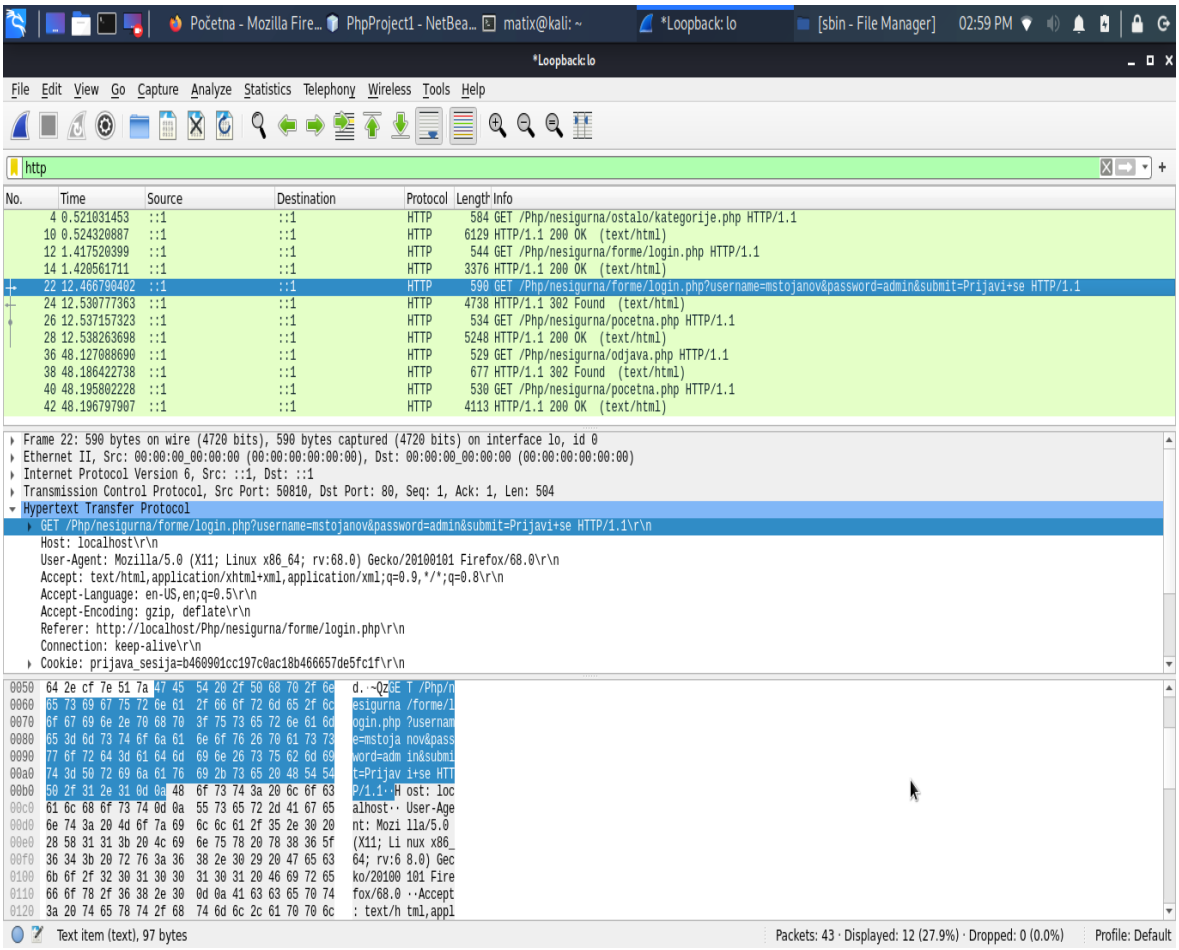
5.3. Otkrivanje osjetljivih i ranjivih podataka

Otkrivanje osjetljivih i ranjivih podataka (engl. *Sensitive data exposure*), kao što je spomenuto i u samom uvodu ovog rada, predstavlja glavni problem svih Web aplikacija i baza podataka koji su u pozadini. Bilo da je otkrivanje počinjeno dobrovoljno ili silom, tisuće i tisuće korisnika su kompromitirani i onaj tko posjeduje te podatke ih može ucjenjivati, manipulirati ili jednostavno te podatke nekome trećem prodati [29].

5.3.1. Napad na otkrivanje osjetljivih i ranjivih podataka

Napad, odnosno demonstriranje otkrivanja osjetljivih i ranjivih podataka će se izvesti koristeći alat Wireshark za snimanje mrežnog prometa. Alat je vrlo jednostavan za korištenje, potrebno je samo odabrati uređaj s kojim je računalo spojeno na mrežu te odabrati opciju „Start capturing packets“ za početak snimanja mrežnog prometa.

Kao ranjiva podloga koristit će se obrazac za prijavu koji ide preko HTTP protokola. Dakle, započet će se snimanje mrežnog prometa i nakon toga upisati korisničko ime i pripadajuću lozinku u obrazac za prijavu. Nakon toga može se vidjeti iz slike 31 kako se u Wiresharku filtrira pretraga po „http“ protokolu.

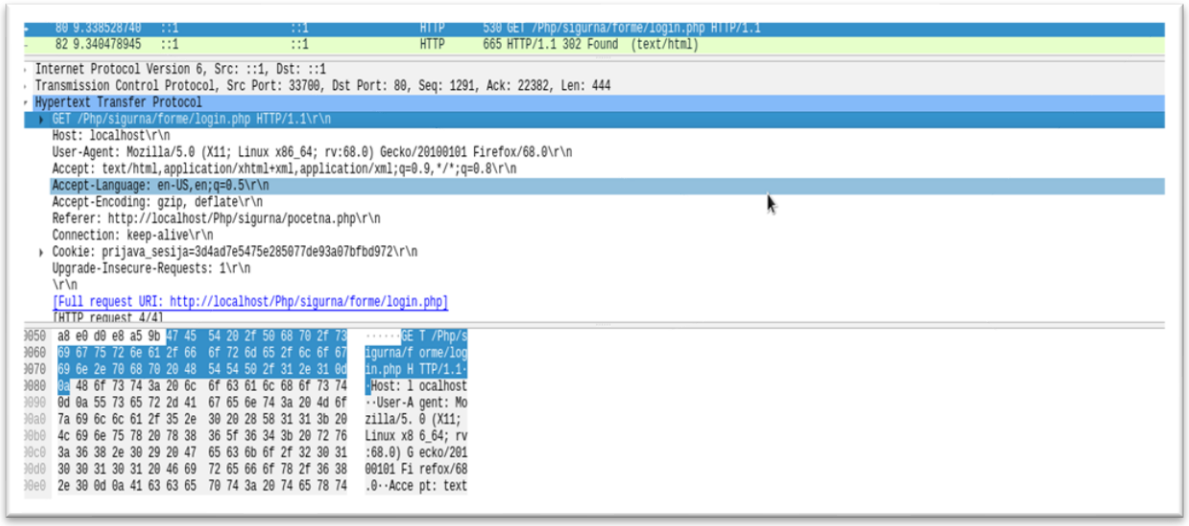


Slika 31. Snimanje mrežnog prometa pomoću Wireshark alata

Kao što je vidljivo i označeno plavom bojom na slici pod brojem 31, obrazac za prijavu koji se podnosi putem GET metode, prikazano je uneseno korisničko ime i lozinka: mstojanov -> admin. Zaključuje se kako korištenje HTTP protokola za prijavu u sustav otkriva osjetljive i ranjive podatke do kojih napadač može vrlo lako doći.

5.3.2.Zaštita od otkrivanja osjetljivih i ranjivih podataka

Ova ranjivost je veoma široki pojam i u principu pokriva sve vrste napada s pomoću kojih je moguće pristupiti i manipulirati osjetljivim podacima. Kao jedna od mogućnosti zaštite, u Web aplikaciji Matix obrazac za prijavu se izvršava preko HTTPS protokola te na taj način kriptira unesene podatke i napadač ne može doći do njih vrlo lako, odnosno potrebni su mu ključevi za enkripciju [40]. Dolazak do tih ključeva za enkripciju je vrlo teško, bilo bi potrebno dobiti kontrolu nad žrtvinim računalom te rekonfigurirati Web preglednik da sprema enkripcijske ključeve na čvrsti disk. Na slici pod brojem 32 je prikazan snimljeni mrežni promet nakon korištenja HTTPS protokola prilikom prijave u sustav.



Slika 32. Snimanje mrežnog prometa sa HTTPS protokolom prilikom prijave

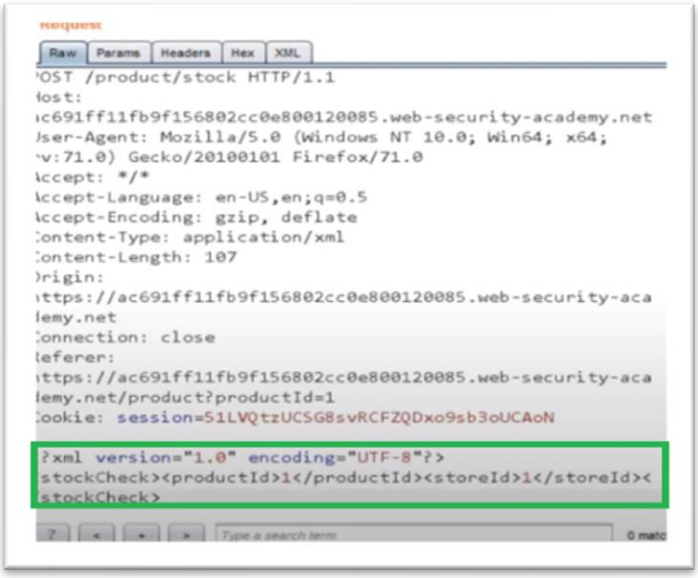
Kao što je vidljivo na slici pod brojem 32, uneseno korisničko ime i lozinka se ne prikazuju prilikom snimanja mrežnog prometa.

Također, kao zaštita od otkrivanja osjetljivih i ranjivih podataka, preporuča se klasificiranje podataka koji se obrađuju i koriste u Web aplikaciji, kao i kontrola pristupa do istih. Potrebno je, prema raznim zakonima o privatnosti i zaštiti podataka, odrediti koji su podaci osjetljivi te njih dodatno zaštititi i osigurati. Ne preporuča se pohranjivanje osjetljivih podataka ukoliko to nije nužno. Potrebno je kriptirati lozinke raznim algoritmima za kriptiranje i stvaranje sažetaka te ih takve spremiti u bazu podataka [29].

5.4. XXE

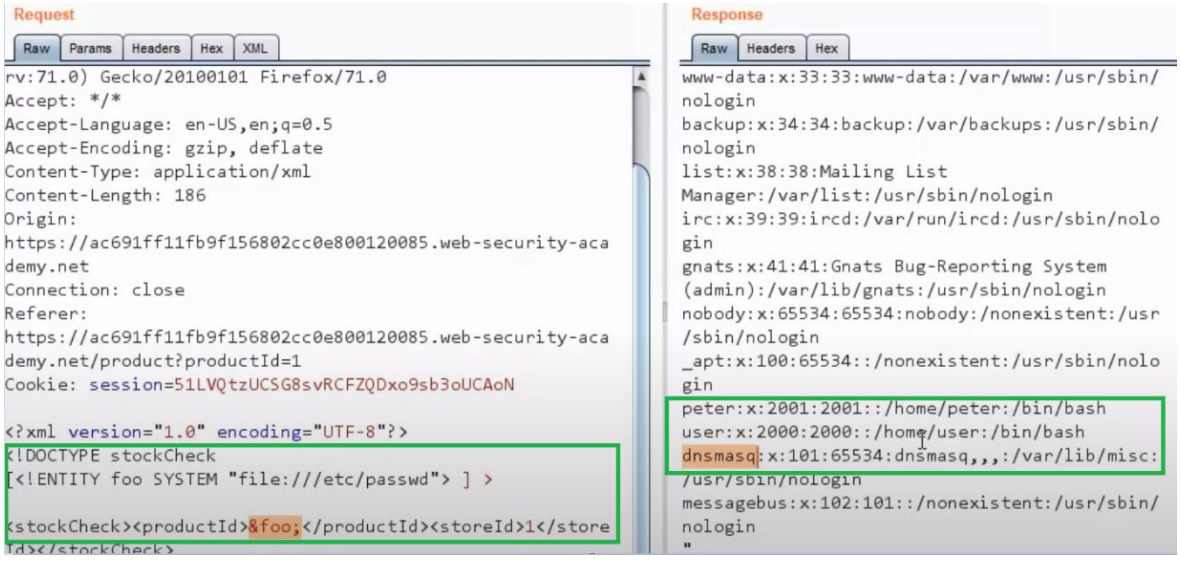
XXE napad (engl. *XML external entity*) je vrsta napada u kojoj ranjivost Web aplikacije omogućava napadaču upad u obradu XML podataka same aplikacije. Princip odvijanja napada je sličan SQL ubrizgavanju jer se također ubacuju zlonamjerni znakovni nizovi unutar XML tagova [29]. Napadač na taj način dolazi do osjetljivih podataka s kojima može manipulirati. Web aplikacija Matix ne sadrži niti obrađuje nikakve podatke u XML obliku te stoga nije moguće implementirati i pokazati napad nad dosadašnjom aplikacijom. Iz tog razloga će se kratko objasniti jedan izrađen primjer napada nad Web trgovinom.

Prilikom izvođenja XXE napada, potrebno je snimati mrežni promet i uhvatiti sve poslane i primljene zahtjeve od aplikacije do preglednika, i obratno. U popisu proizvoda se odabrao jedan od njih i kliknulo na detalje gdje je prikazana cijena istog, a sve to je alat Burp Suite snimio kao što je prikazano na slici pod brojem 33.



Slika 33. Podaci o proizvodu u XML obliku

Nakon što se uhvatio poslani zahtjev, implementirao se zlonamjerni kod koji će omogućiti napadaču ispis korisnika i njihovih lozinki. Implementirani kod, zajedno s rezultatom napada je prikazan na slici pod brojem 34.



Slika 34. XXE napad

Na slici pod brojem 34 je s lijeve strane prikazan ubačeni zlonamjerni kod unutar XML tagova koji omogućuje ispis korisnika i njihovih lozinki koji su označeni zelenim pravokutnikom na desnoj strani slike.

Zaštitu od XXE napada nije moguće implementirati nad Web aplikacijom Matix jer ona ne sadrži niti obrađuje ikakve podatke u XML obliku.

Kao općenita zaštita od XXE napada, preporuča se korištenje što manje složenih formata podataka, redovito ažurirati XML procesore i biblioteke koje se koriste prilikom rada Web aplikacije. Također se preporuča verificiranje prenesenih XML i XSL (engl. *EXtensible Stylesheet Language*) datoteka pomoću XML validatora [29].

5.5. Prekinuta kontrola pristupa

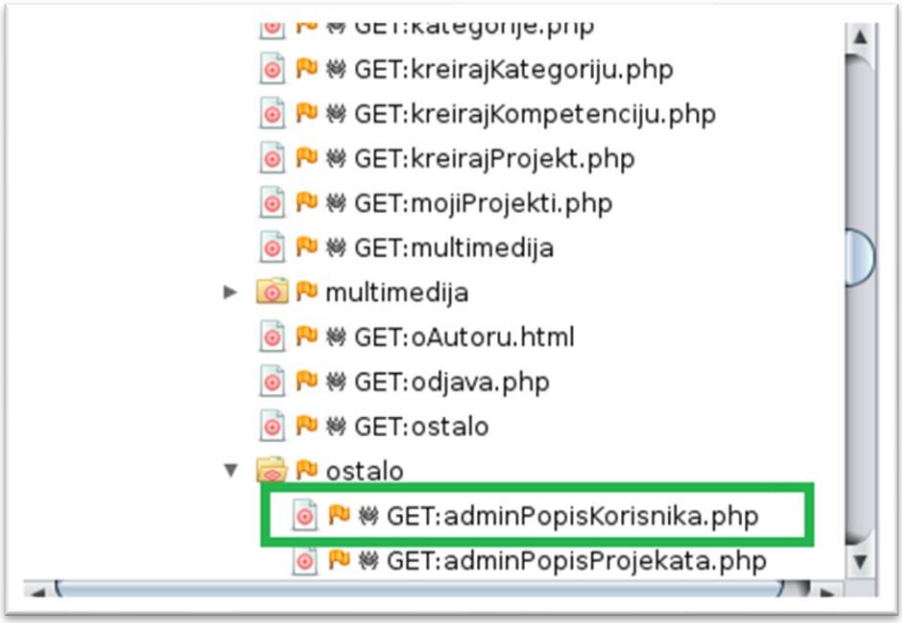
Prekinuta kontrola pristupa (engl. *Broken access control*) je propust koja je nastala pri implementaciji Web aplikacije, a omogućava korisnicima pristup i funkcionalnosti koje ne bi smjeli imati sa svojom razinom korisničke uloge [29]. Drugim riječima, napadač u ovom slučaju „izigra“ Web aplikaciju i dolazi do osjetljivih podataka i ovlasti koje mogu narušiti rad Web aplikacije i time ugroziti cijeli sustav.

Ranjivost u Web aplikaciji Matix je moguća ručnim unosom poveznice do koje korisnik može doći, a za to nema dovoljnu razinu ovlasti. Uzet će se za primjer neregistrirani korisnik koji kada ručno unese poveznicu do stranice za popis korisnika, preglednik će ga preusmjeriti na zadanu stranicu na koju bi samo administrator trebao moći pristupiti.

Navedena stranica sadrži osjetljive podatke i omogućuje blokiranje/odblokiranje korisnika te stoga predstavlja veliku ranjivost ukoliko napadač unese zadanu poveznicu.

5.5.1. Napad prekinutom kontrolom pristupa

Kao demonstracija napada prekinutom kontrolom pristupa, koristit će se alat OWASP Zed Attack Proxy (ZAP) [41]. Alat je već ugrađen u operacijski sustav Kali Linux, a sve što je potrebno je otvoriti Web aplikaciju Matix i odabrati skeniranje cijele aplikacije. Važno je za napomenuti kako nije potrebno biti prijavljen u sustav za skeniranje aplikacije. Potencijalni napad prekinutom kontrolom pristupa je moguć nakon što skeniranje završi, a alat izbaci sav popis stranica i direktorija koje Web aplikacija koristi. Napadač sada zna putanje i direktorije, a sve što je potrebno jest ručno unijeti jednu od poveznica. Za primjer će se uzeti navedena stranica s popisom korisnika. Prikazana poveznica je na slici pod brojem 35.



Slika 35. Dostupan popis korisnika

Kao što je vidljivo na slici pod brojem 35, poveznica do stranice je localhost/Php/nesigurna/ostalo/adminPopisKorisnika.php.

Kada napadač unese poveznicu, preusmjerava se na popis korisnika gdje može manipulirati njezinim sadržajem, odnosno dobiva funkcionalnosti koje ne bi trebao imati.

5.5.2. Zaštita od prekinute kontrole pristupa

Kako bi se zaštitilo od neželjenog pristupa i dobivanja funkcionalnosti koje neki korisnici ne bi smjeli imati, potrebno se kvalitetno osigurati na način da se svaka pojedina stranica zaštiti sljedećim programskim kodom prikazanim na slici pod brojem 36.

```
if(!isset($_SESSION["uloga"])){
        header("Location:../forme/login.php");
        exit();
    }
    elseif(isset($_SESSION["uloga"]) && $_SESSION["uloga"] != 1){ // !== "1"
        header("Location:../pocetna.php");
        exit();
    }
```

Slika 36. Zaštita od prekinute kontrole pristupa

Programski kod prikazan na slici pod brojem 36 se sadrži od IF i ELSE naredbi. Prva selekcija (IF) ispituje postoji li kreirana sesija (odnosno je li korisnik prijavljen u sustav), a ukoliko ne postoji, korisnika se preusmjeruje na obrazac za prijavu.

Druga selekcija ispituje postoji li kreirana korisnička sesija i je li ona sadrži razinu ovlasti administrator (u bazi podataka je administrator zapisan kao uloga s vrijednosti 1). Ukoliko je kreirana korisnička sesija, a korisnička uloga je različita od 1 (od administratora), korisnika se preusmjeruje na početnu stranicu Web aplikacije. Na ovaj način je spriječen pristup stranici svim korisnicima koji imaju razinu ovlasti različitu od administratora, kao i svim koji nisu prijavljeni u sustav.

Kao dodatna zaštita, preporuča se vođenje zapisa o prijavi i radu sustava te informiranju administratora ukoliko se otkrije potencijalni propust. Također je potrebno detaljno testirati ponašanje i rad Web aplikacije prije puštanja iste u rad, kao i onemogućiti zaobilaženje provjera kontrole pristupa [29].

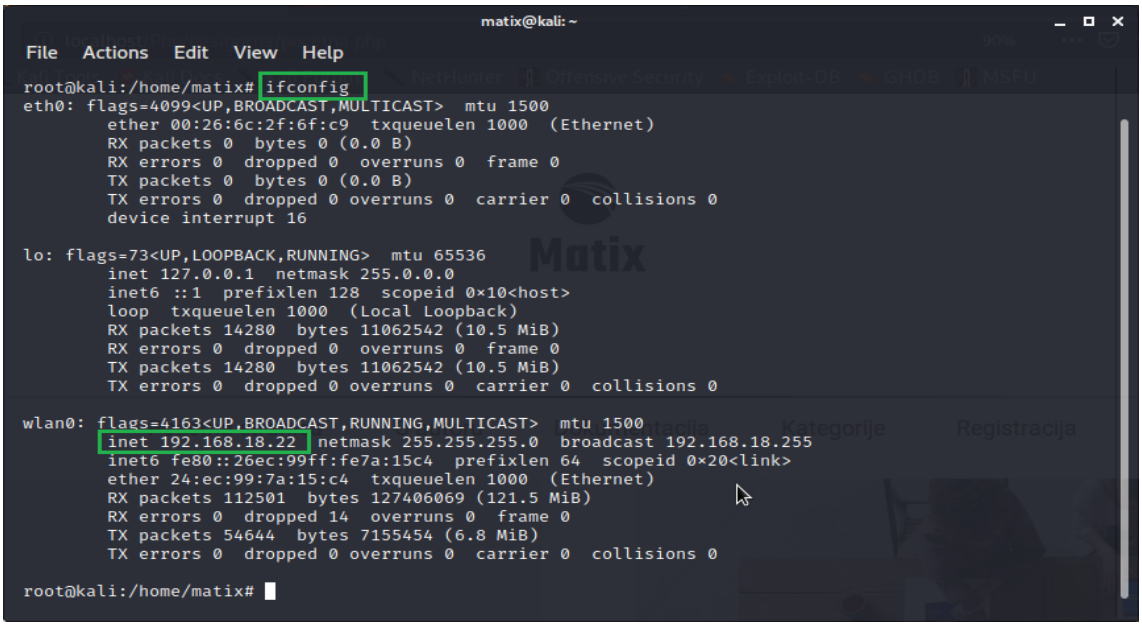
5.6. Neispravno implementirana sigurnosna konfiguracija

Neispravno implementirana sigurnosna konfiguracija (engl. *Security misconfiguration*) je također propust u implementaciji Web aplikacije. Često konfiguracije nisu do kraja odrađene ili su odrađene na način da sadrže greške u radu i time predstavljaju potencijalnu ranjivost. Ukoliko na primjer postoje otvoreni portovi na poslužitelju, moguće je tu ranjivost iskoristiti i napasti poslužitelj na kojem se izvršava Web aplikacija [29].

5.6.1. Napad neispravno implementiranom sigurnosnom konfiguracijom

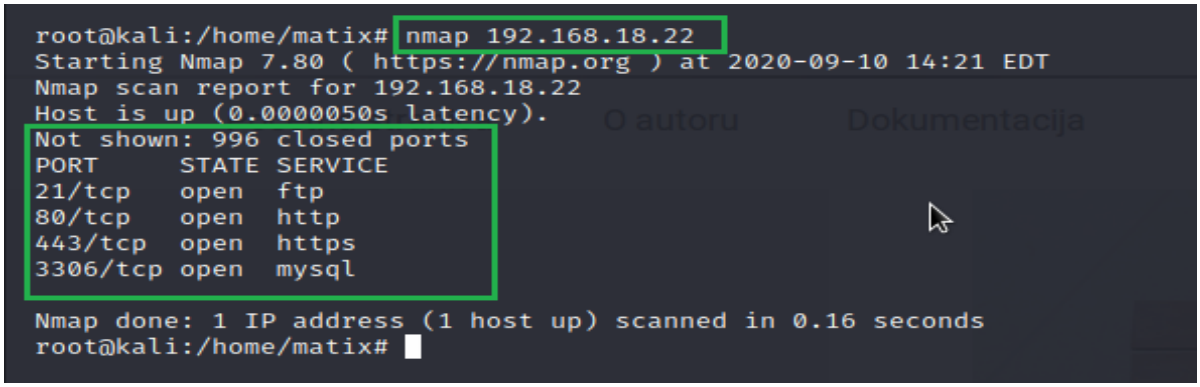
U ovom poglavlju će se pokazati jedan od mogućih napada neispravno implementiranom sigurnosnom konfiguracijom, a to je skeniranje portova pomoću alata „nmap“, koji dolazi ugrađen u operacijski sustav Kali Linux [42], [43]. Razlog skeniranja portova je pronalazak onih koji su otvoreni te samim time i ranjivi za napad na poslužitelj i otkrivanje osjetljivih podataka [29].

Skeniranje se započelo na način da se u Terminal Emulator upisala naredba „ifconfig“ koja će pokazati IP adresu računala. Naredba i njen rezultat je prikazan na slici pod brojem 37.



Slika 37. Naredba ifconfig

Kao što je vidljivo na slici pod brojem 37, IP adresa računala je 192.168.18.22 te će biti potrebna za korištenje alata nmap. Sljedeći postupak u skeniranju portova je upisivanje naredbe nmap 192.168.18.22 u Terminal Emulator. Naredba i njen rezultat su prikazani na slici pod brojem 38.



Slika 38. Skeniranje portova alatom nmap

Kao što je vidljivo na slici pod brojem 38, 996 portova je zatvoreno, a 4 porta su otvorena. Otvoreni portovi su 21,80,443 te 3306. U principu, otvoren port označava postojanje programa koji aktivno sluša i prima, u ovom slučaju, TCP (engl. *Transmission Control Protocol*) pakete. Skeniranjem se zaključilo da postoje 4 otvorena porta koji predstavljaju ranjivost i mogući napad na poslužitelj putem njih. Preporučljivo je zatvoriti one portovi koji se ne koriste, a u ovom slučaju je to port 21.

Port se može zatvoriti sljedećom naredbom u Terminalu: fuser -k -n tcp 21 [44]. Jedni od mogućih napada na poslužitelj su DoS (engl. *Denial of Service*) i DDoS (engl. *Distributed Denial of Service*).

DoS je napad uskraćivanja usluga, pri čemu napadač ima za cilj onesposobiti normalno funkcioniranje Web aplikacije ili usluge. Osnovna obilježja koja sugeriraju da je određena Web stranica ili aplikacija pod DoS napadom su: dugo vrijeme pristupanja i učitavanja stranice, nedostupnost i nemogućnost pristupa određenoj Web stranici/aplikaciji te povećani broj primljenih spam poruka putem elektroničke pošte. Dakle, kao što i ime samo govori, napadač pokušava spriječiti korisnicima usluga korištenje tih istih usluga. Važno je za napomenuti kako DoS napad dolazi s jedne izvorišne IP adrese. Glavna karakteristika DoS napada je slanje enormnog broja paketa na poslužitelje kako bi se oni preopteretili i time izgubili mogućnost normalnog održavanja i upravljanja sustava [45].

DDoS možemo nazvati „velikim bratom“ DoS napada. Glavna razlika je u tome što pri DDoS napadu, više sustava napada žrtvu, a pri DoS napadu, jedan sustav (s jedne IP adrese) napada jednu žrtvu [46]. Drugim riječima, DDoS je mnogo složeniji, zahtijeva veću i složeniju mrežu računala za generiranje velikog broja prometa koji će se slati žrtvama. Na taj način je moguće provaliti i u stotine računala putem Interneta. Važno je za napomenuti da se promet, odnosno paketi šalju s više lokacija odjednom pa ga je teže spriječiti i ući u trag za razliku od DoS napada. Također, DDoS napad može napraviti više štete nego DoS napad. Naravno, pri izvođenju napada se koriste lažne IP adrese kako bi se napadač zaštitio od otkrivanja svojeg identiteta.

5.6.2. Zaštita od neispravno implementirane sigurnosne konfiguracije

Kako bi se kvalitetno zaštitilo, potrebno je kontrolirati koji su portovi na poslužitelju otvoreni i samim time ranjivi. Ukoliko i dođe do napada na Web poslužitelj, zaštititi od njih se može na sljedeće načine. Neki od njih su korištenje raznih tehnologija i anti-DoS servisa koji sprječavaju i brane korisnika od DoS napada. Ako korisnik ili poslovni subjekt shvati da je pod napadom, potrebno se što prije javiti pružatelju internetskih usluga, koji će analizirati i utvrditi može li se korisnikov promet nekako preusmjeriti. Drugi način koji pružatelji internetskih usluga koriste je preusmjeravanje prekomjernog prometa u jednu posebnu rutu, koja se naziva „crna rupa“. Ovako se može spriječiti pad i nedostupnost neke Web stranice/aplikacije. Naravno, potrebno je vatrozid držati na visokoj razini i redovito ga ažurirati, kao i pravilno konfigurirati usmjerivače da odbijaju lažne dolazne promete [29], [46].

Kako bi se kvalitetno osiguralo od DDoS napada, potrebno je dovesti razinu sigurnosti poslužitelja na najvišu moguću razinu. Popis svih procesa pokrenutih na poslužitelju, popis svih mrežnih priključaka, filtriranje paketa i onemogućavanje procesa koji bi mogli naštetiti normalnom radu poslužitelja su neki od osnovnih načina kako se razina sigurnosti poslužitelja može poboljšati. Distribuiranje Web stranice preko višestrukih poslužitelja te povezivanje na Web preko više pristupnih točka su također metode koje je poželjno provesti, ukoliko za to postoje resursi. Također, postoje razne online stranice koje pružaju zaštitu od DDoS napada. Jedna od njih je „Cloudflare“, koja nudi osnovan paket i besplatnu zaštitu za jednostavne Web stranice i projekte uz prethodnu registraciju putem e-maila. Uz to, postoje i paketi za male poduzetnike, organizacije i velika poduzeća [47]. Cijena za manje organizacije se kreće od 200 američkih dolara mjesečno, a za velike organizacije je potrebno kontaktirati Cloudflare te oni na temelju složenosti organizacije određuju cijenu i izrađuju plan zaštite. Važno je za napomenuti da ne postoji 100% učinkovita obrana od DDoS napada, ali navedene metode obrane se mogu bar donekle suprotstaviti i oslabiti napad.

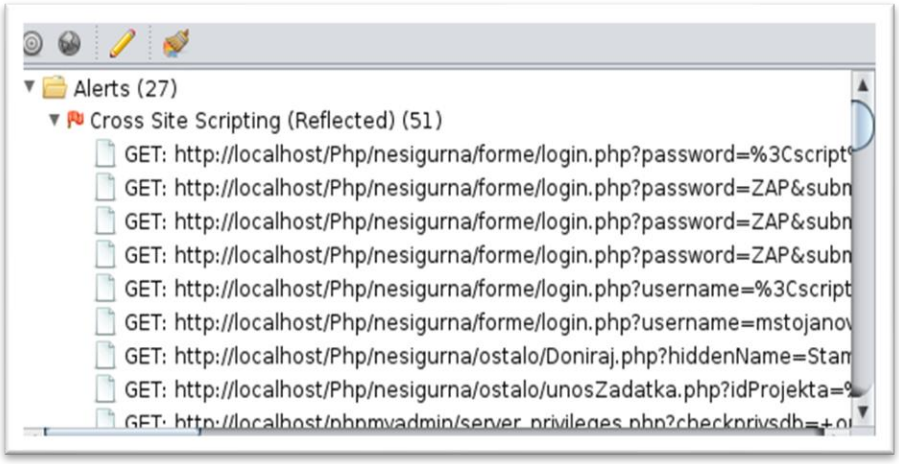
5.7. XSS

XSS (engl. *Cross-site scripting*) je vrsta napada u kojoj napadač ubacuje zlonamjerni kod (najčešće je to JavaScript kod) u ranjivu Web aplikaciju [29], [48], [49]. Nakon toga se maliciozni kod prosljeđuje Web poslužitelju koji, kada ga zaprimi, generira novi izlazni sadržaj u kojem će se izvršavati taj maliciozni skriptni kod. Jednom kada to napravi, napadač može pristupiti i manipulirati osjetljivim podacima, presresti i ukrasti kolačiće (engl. *Cookies*), manipulirati Web preglednikom na način da preusmjeri korisnika na zlonamjerne i nesigurne Web aplikacije.

Ukoliko zlonamjerni kod ostane trajno pohranjen na Web poslužitelj, radi se o ustrajnom XSS napadu, a ukoliko zlonamjerni kod nije trajno pohranjen na Web poslužitelj, nego navodi korisnika na posebne Web stranice u kojima je taj zlonamjerni kod već implementiran, onda se radi o neustrajnom XSS napadu. Također je moguć i treći slučaj u kojem se karakterizira tumačenje neke Web stranice koju je korisnik otvorio kao datoteku koja se nalazi na lokalnom računalu [48], [49], [50]. Drugim riječima, maliciozni kod koji se izvršava na strani poslužitelja, može manipulirati Web preglednikom te omogućiti izvođenje zlonamjernog koda i na korisnikovom računalu te na taj način ugroziti sigurnost čitavog operacijskog sustava. Ova vrsta XSS napada se naziva DOM baziran XSS napad.

5.7.1. XSS napad

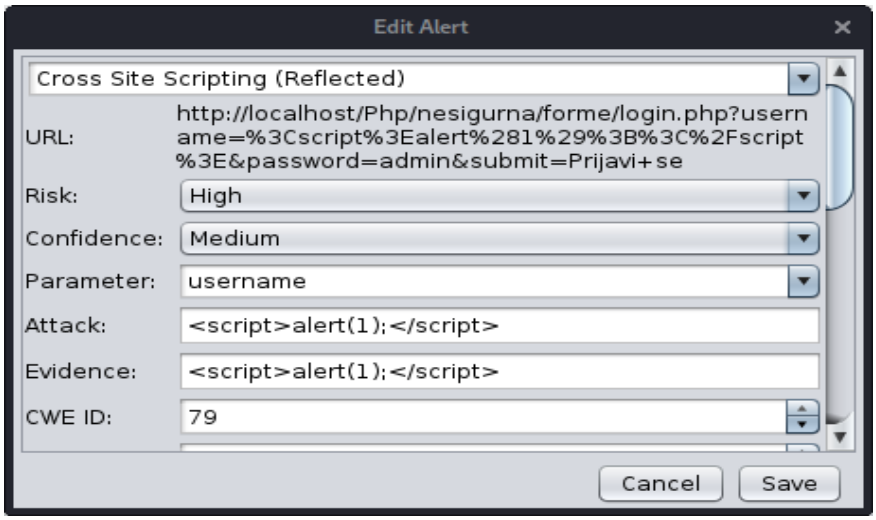
Analizom Web aplikacije pomoću već spomenutog i opisanog alata - OWASP ZAP [41], pokazalo se kako je aplikacija ranjiva putem XSS napada na obrascu za prijavu, registraciji, obrascu za doniranje novaca projektu te stranici za unos radnih zadataka. Ranjivost putem XSS napada, koja je dobivena korištenjem alata OWASP ZAP je prikazana na slici pod brojem 39.



Slika 39. XSS ranjivosti na obrascima

Na slici pod brojem 39, alat je izbacio da postoji 51 ranjivost putem XSS napada, razlog tomu je što alat svako polje u obrascu računa kao potencijalnu ranjivost.

OWASP ZAP je također izbacio i način napada na navedene obrazac, a to je zlonamjerni kod „<script>alert(1)</script>. Dakle, unosom ovog koda izvršit će se JavaScript kod te će se ispisati broj 1 na ekran. Važno je za napomenuti kako alert(1) sam po sebi ne radi nikakvu štetu, nego samo predstavlja XSS ranjivost. Drugim riječima, ovdje alert(1) predstavlja sve moguće zlonamjerne kodove koji stvarno mogu naštetiti radu Web aplikacije, a realizirati se na isti način unutar <script> taga. Na slici pod brojem 40 je prikazan JavaScript kod kojim će se demonstrirati XSS napad.



Slika 40. Zlonamjerni kod za izvršavanje XSS napada

Kao što je prikazano na slici pod brojem 40, u obrazac za prijavu će se unijeti sljedeći kod: <script>alert(1)</script>. Nakon unosa koda, na ekran se ispisuje broj 1 kao rezultat XSS napada, kao što je prikazano na slici pod brojem 41.

Slika 41. Izvršavanje alert naredbe

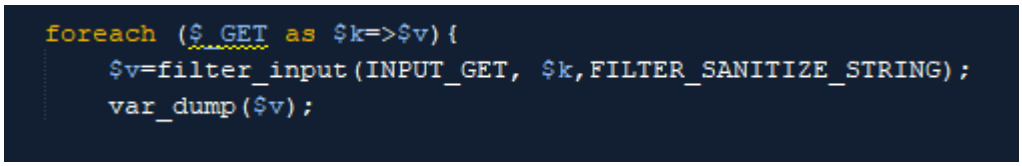
Na slici pod brojem 41 je prikazan rezultat XSS napada, odnosno izvršavanje alert(1) naredbe, gdje korisniku iskoči na ekran prozor s brojem jedan.

5.7.2. Zaštita od XSS napada

Kao zaštita od potencijalnog XSS napada nad obrascima prilikom preuzimanja korisničkih podataka, implementiran je sljedeći kod. Dakle, nakon što se podaci preuzmu putem GET metode (u ovom slučaju je GET, no može biti i POST), koristi se funkcija *filter_input* [51] koja prima sljedeće parametre [52]:

1. Metoda kojom se dohvaćaju podaci (GET)
2. Naziv ključa koji sadrži vrijednost koja se provjerava ($v)
3. Očekivana vrijednost unosa (FILTER_SANITIZE_STRING)

Na slici pod brojem 42 je prikazan implementirani programski kod.

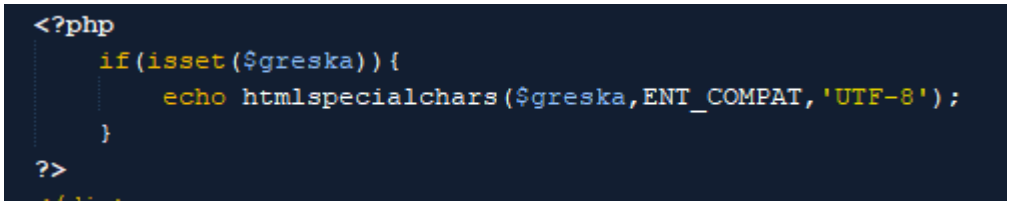


Slika 42. Zaštita od XSS napada

Također, potrebno se zaštititi i prilikom ispisa podataka s funkcijom htmspecialchars() [51], koja izbjegava neželjene znakove prilikom ispisa. Funkcija prima sljedeće parametre [52]:

1. Varijabla koja se provjerava
2. ENT_COMPAT za konvertiranje dvostrukih navodnika
3. Način kodiranja(UTF-8)

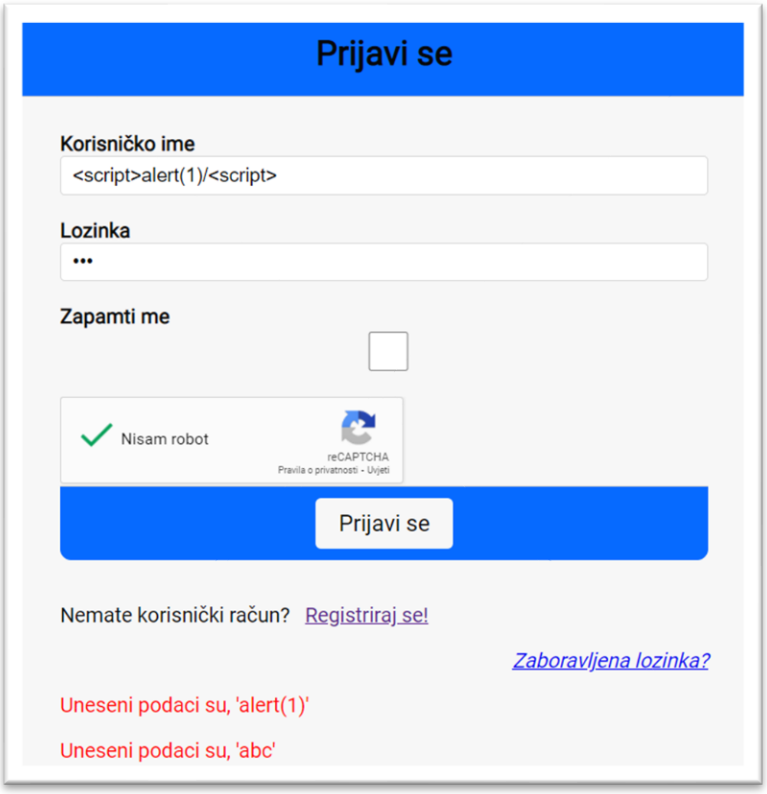
Na slici pod brojem 43 je prikazan implementirani programski kod koji ispisuje eventualne greške nastale prilikom prijave u sustav, a one su označene varijablom „greska“.



Slika 43. Zaštita prilikom ispisa podataka

U ovom slučaju, ukoliko se na primjer u obrazac upiše <script>alert(1)</script>, koji može potencijalno sadržavati zlonamjerni JavaScript kod, neće se izvršavati.

Na slici pod brojem 44 je prikazan ispis svih unesenih podataka u obrazac za prijavu. Vidljivo je kako se broj 1 ne ispisuje na ekran, odnosno ne izvršava se JavaScript kod jer je implementirana zaštita od XSS napada.



Slika 44. Zaštita od XSS napada

Kao što je vidljivo na slici pod brojem 44, <script> tag se ignorirao te se ne izvršava i uzima u obzir JavaScript kod prilikom ispisa na ekran, dok je unesena lozinka „abc“ ispisana na ekran normalno jer ne sadržava potencijalni zlonamjerni kod.

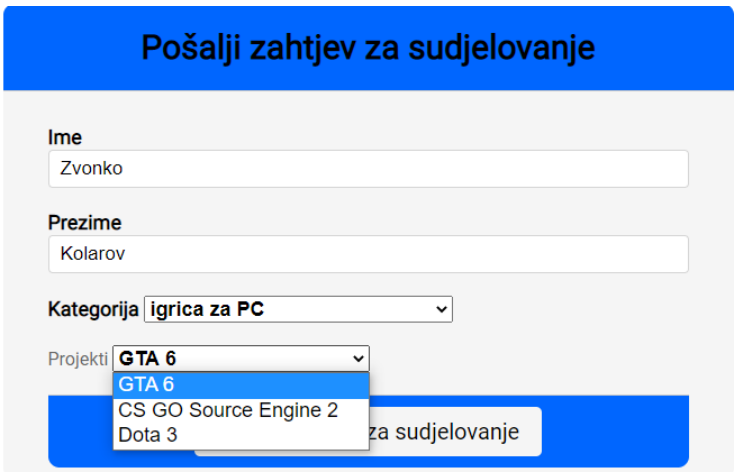
Također, preporučljivo je izbjegavanje otvaranja sumnjivih i neprovjerenih poveznica. Ukoliko poveznica sadrži nekakve heksadecimalne znakove, vrlo velika vjerojatnost je da je ta poveznica nesigurna i da se iza nje možda krije potencijalni XSS napad. Još jedan mogući način od zaštite je u postavkama Web preglednika onemogućavanje izvršavanja JavaScript koda na strani klijenta. Također je poželjno kodirati Web stranice pri njihovoj izradi, čime se može onemogućiti/smanjiti izvršavanje zlonamjernog koda [29] [52].

5.8. Nesigurna deserijalizacija

Nesigurna deserijalizacija (engl. *Insecure deserialization*) je vrsta napada u kojem Web aplikacija ne provjerava serijalizirani objekt te je na taj način moguće pokrenuti i izvesti zlonamjerni kod na Web poslužitelju i time naštetiti cijelome sustavu [29].

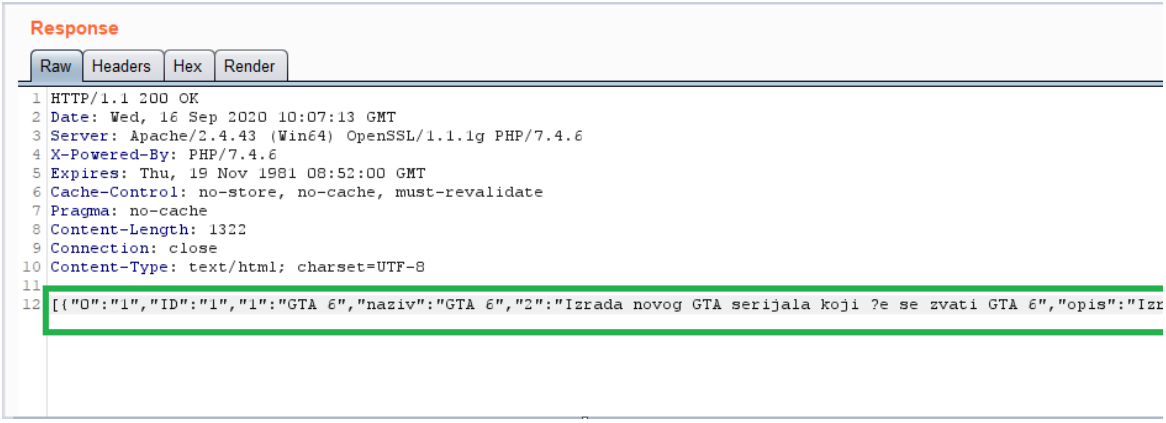
5.8.1. Napad nesigurnom deserijalizacijom

Napad nesigurnom deserijalizacijom će se demonstrirati korištenjem alata Burp Suite. Korisniku će se poslati zahtjev za sudjelovanje na projektu [53]. Prilikom korištenja ovog obrasca, koristi se JSON jer se na temelju odabrane kategorije projekta automatski prikazuju svi projekti u toj odabranoj kategoriji. Na slici pod brojem 45 je prikazan obrazac za slanje zahtjeva za sudjelovanje na projektu.



Slika 45. Korištenje JSON-a

Dakle, cijelo vrijeme Burp Suite snima poslane i primljene zahtjeve te se nakon odabira kategorija i željenog projekta, poslani zahtjev pronađe u Burp Suite-u gdje se serijalizirao objekt s podacima o korisniku kojem se šalje zahtjev, kao i podacima o odabranom projektu. Na slici pod brojem 46 je prikazan odgovor na poslani zahtjev u alatu Burp Suite.



Slika 46. Serijalizirani objekt nakon slanja zahtjeva za sudjelovanje

Napad nesigurnom deserijalizacijom je moguće izvršiti umetanjem zlonamjernog koda u poslani zahtjev. Poslani zahtjev je prikazan na slici pod brojem 47.

Slika 47. Poslan zahtjev za sudjelovanje

Na slici pod brojem 47, zelenim pravokutnikom je označen dio zahtjeva u kojem je moguće ubaciti zlonamjerni kod te na taj način pokrenuti njegovo izvršavanje unutar Web aplikacije i doći do osjetljivih podataka. Također je moguće i jednostavno promijeniti broj nakon parametra podatak=, te također i na taj način dobiti osjetljive podatke koji će se ispisati u odgovoru na zahtjev, kao što je prikazano na slici pod brojem 46, samo će sadržavati druge podatke.

5.8.2. Zaštita od nesigurne deserijalizacije

Najučinkovitija zaštita od nesigurne deserijalizacije je, ukoliko je to moguće, neprihvaćanje serijaliziranih objekata iz nepouzdanih i sumnjivih izvora. Preporuča se nadziranje deserijalizacije te alarmiranje korisnika ukoliko se korisnik učestalo deserijalizira. Dodatno, preporuča se provođenje provjere integriteta nad serijaliziranim objektima [29].

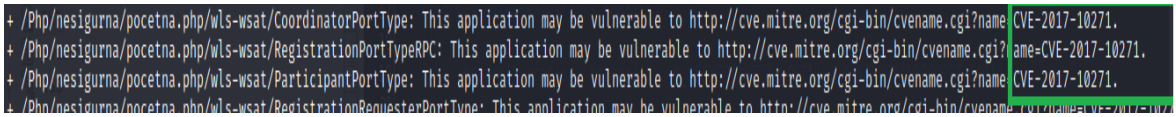
5.9. Korištenje komponenata s već poznatim ranjivostima

Korištenje komponenata s već poznatim ranjivostima (engl. *Using Components with Known Vulnerabilities*), kao što i ime samo govori, odnosi se na Web aplikacije koje koriste komponente s već poznatim ranjivostima čime mogu narušiti sigurnosti cijelog sustava. Ako se neke komponente s ranjivostima kvalitetno ne „pokrpaju“, a puste u rad, zasigurno će ta Web aplikacija biti izložena ovome napadu [29].

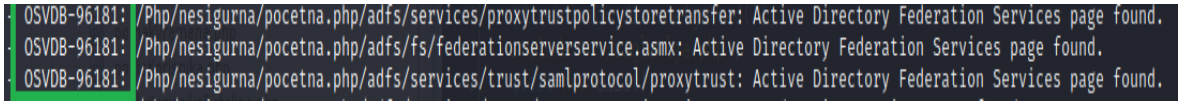
5.9.1. Napad na otkrivanje komponenata s već poznatim ranjivostima

Za otkrivanje komponenata s već poznatim ranjivostima će se koristiti alat „nikto“. Nikto dolazi kao ugrađen u operacijski sustav Kali Linux [54]. Korištenje je vrlo jednostavno, potrebno je u Terminal unijeti sljedeću naredbu: nikto -h te poveznicu do Web aplikacije na poslužitelju. Ono što alat zapravo radi je testiranje poslužitelja na način da otkriva i ispisuje ranjive komponente koje poslužitelj i aplikacija koja se izvršava na poslužitelju koriste.

Sve pronađene ranjive komponente ispisuje kao reference CVE (engl. *Common Vulnerabilities and Exposures*) koje se nalaze u bazi podataka OSVDB (engl. *Open Source Vulnerability Database*). Rezultat korištenja alata sadrži preko sto linija sadržaja pa će se prikazati samo neke otkrivene ranjive komponente zajedno s njihovom CVE/OSVDB referencom. Na slici pod brojevima 48 i 49 su prikazane otkrivene ranjivosti korištenjem alata nikto.



Slika 48. CVE refrenca



Slika 49. OSVDB referenca

Ranjivost CVE-2017-10271 označava kako neprijavljeni korisnik s pristupom internetu može kompromitirati poslužitelj i preuzeti kontrolu nad njim, zajedno sa svim osjetljivim podacima koji su na njemu [55].

OSVDB-96181 ranjivost označava mogućnost otkrivanja i manipuliranja osjetljivih podataka na poslužitelju, kao i moguće zaključavanje korisničkih računa u sustavu [56].

5.9.2. Zaštita od korištenja komponenata s već poznatim ranjivostima

Za izbjegavanja napada trebao bi postojati proces koji će upravljati zakrpama na način da će ukloniti sve nepotrebne značajke, komponente i datoteke. Potrebno je analizirati Web aplikaciju i otkriti potencijalne ranjive komponente, kao i biti upućen u sadržaj baza otkrivenih ranjivosti kako bi se korištenje takvih komponenata izbjeglo.

Također, svaka organizacija bi trebala imati realizirani plan za nadziranje, pokretanje i primjenu ažuriranja ili promjene konfiguracije [29].

5.10. Nedovoljna kvaliteta nadzora i vođenja zapisa o prijavi i aktivnosti korisnika

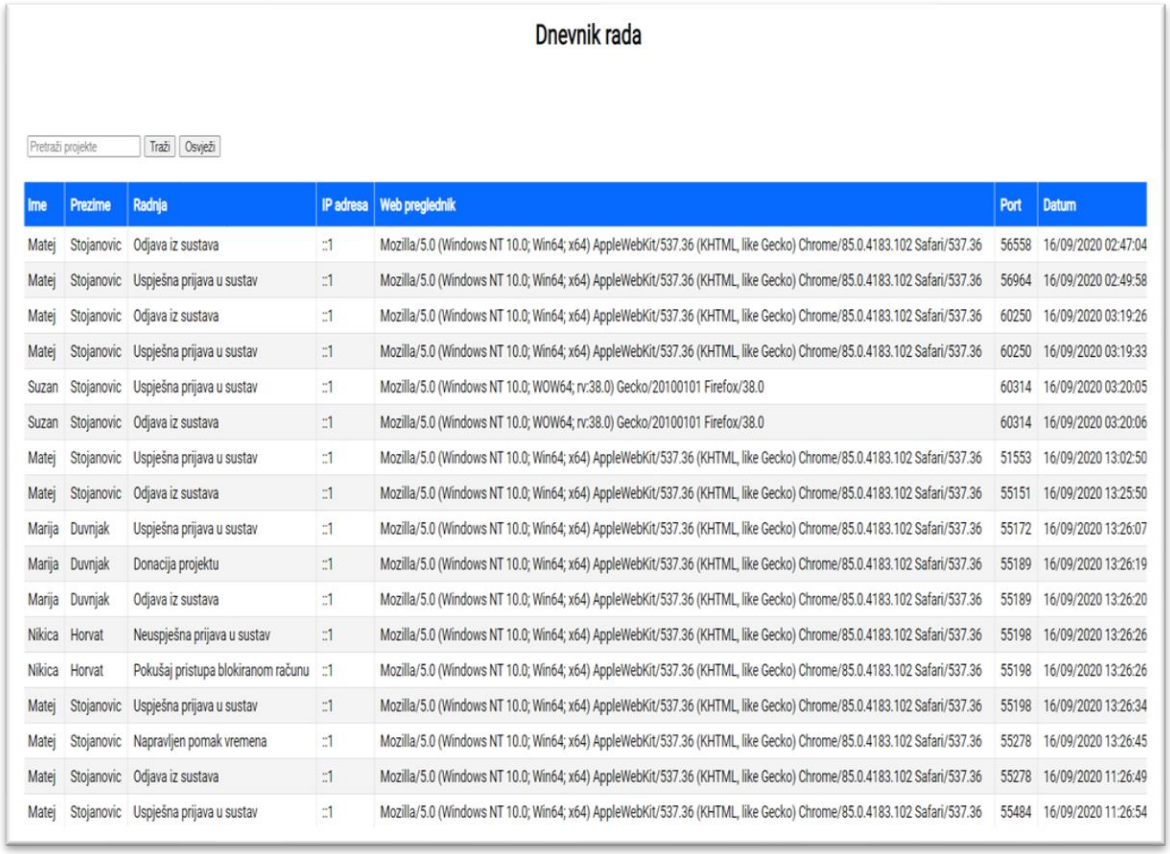
Nedovoljna kvaliteta nadzora i vođenja zapisa o prijavi i aktivnosti korisnika (engl. *Insufficient logging and monitoring*) označava nekvalitetno vođenje dnevničkih zapisa.

Konstantni neuspjeli pokušaji logiranja, nedovoljno kriptirani zapisi, zapisi spremljeni samo na lokalnom računalu te nedovoljno analiziranje zapisa predstavljaju ogroman propust u sigurnosti rada Web aplikacije. Kvalitetno vođenje navedenih može otkriti i spriječiti razne napada i time poboljšati sigurnost svih korisnika [29].

5.10.1. Zaštita od nedovoljne kvalitete nadzora i vođenja zapisa o prijavi i aktivnosti korisnika

Kao zaštita, preporučuje se bilježenje svih podataka o prijavi, kontroli pristupa i greškama prilikom prijave na način da se mogu identificirati potencijalni napadi, kao i zlonamjerni korisnički računi. Dnevnički zapisi o aktivnosti korisnika u sustavu trebaju biti realizirani u formatu na način da se mogu brzo i kvalitetno pronaći i iščitati. Preporučljivo je i uspostavljanje efikasnog nadzora sumnjivih aktivnosti, kao spremanje dnevničkih zapisa na sigurne i neprobojne sustave [29].

Web aplikacija Matix bilježi osnovne korisničke aktivnosti u poseban dnevnik, kojemu može pristupiti samo korisnička uloga administrator. Osnovne aktivnosti su na primjer: korisnička radnja, upit nad bazom, datum aktivnosti, tip aktivnosti (prijava/odjava, rad s bazom, ostalo), te podaci o korisniku koji ju je izvršio. Na slici pod brojem 50 je prikazan dnevnik aktivnosti u koji administrator ima uvid.



Dnevnik rada

Pretraži projekte | Traži | Osvježi

Ime	Prezime	Radnja	IP adresa	Web preglednik	Port	Datum
Matej	Stojanovic	Odjava iz sustava	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	56558	16/09/2020 02:47:04
Matej	Stojanovic	Uspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	56964	16/09/2020 02:49:58
Matej	Stojanovic	Odjava iz sustava	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	60250	16/09/2020 03:19:26
Matej	Stojanovic	Uspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	60250	16/09/2020 03:19:33
Suzan	Stojanovic	Uspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0	60314	16/09/2020 03:20:05
Suzan	Stojanovic	Odjava iz sustava	::1	Mozilla/5.0 (Windows NT 10.0; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0	60314	16/09/2020 03:20:06
Matej	Stojanovic	Uspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	51553	16/09/2020 13:02:50
Matej	Stojanovic	Odjava iz sustava	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55151	16/09/2020 13:25:50
Marija	Duvnjak	Uspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55172	16/09/2020 13:26:07
Marija	Duvnjak	Donacija projektu	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55189	16/09/2020 13:26:19
Marija	Duvnjak	Odjava iz sustava	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55189	16/09/2020 13:26:20
Nikica	Horvat	Neuspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55198	16/09/2020 13:26:26
Nikica	Horvat	Pokušaj pristupa blokiranom računu	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55198	16/09/2020 13:26:26
Matej	Stojanovic	Uspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55198	16/09/2020 13:26:34
Matej	Stojanovic	Napravljen pomak vremena	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55278	16/09/2020 13:26:45
Matej	Stojanovic	Odjava iz sustava	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55278	16/09/2020 11:26:49
Matej	Stojanovic	Uspješna prijava u sustav	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36	55484	16/09/2020 11:26:54

Slika 50. Dnevnik aktivnosti

Na slici pod brojem 50, administrator ima uvid u osnovne korisničke aktivnosti koji se zapisuju prilikom trajanja korisničke sesije. Bilježi se ime i prezime korisnika, radnju koju je napravio, njegova IP adresa (::1 označava adresu lokalnog poslužitelja u IPV6 formatu), Web preglednik koji korisnik koristi, kao i port te datum same radnje [52].

6. Zaključak

Web poslužitelji, Web mjesta i Web aplikacije, kojih je svakim danom sve više i više, sadržavaju osjetljive podatke koje je potrebno kvalitetno spremati, održavati i zaštititi od raznih napada. Zaštita osobnih podataka svaki put kada se oni prikupljaju, pohranjuju ili obrađuju, je jedno od temeljnih prava propisano Poveljom Europske unije o temeljnim pravima te je stoga donesena Opća uredba o zaštiti podataka, čije kršenje može rezultirati visokim novčanim kaznama. U praksi, za provođenje sigurnosnih testiranja, koriste se razni operacijski sustavi, kao i virtualna okruženja poput Virtual Box, kako bi na jednom računalu istovremeno moglo biti pokrenuto više operacijskih sustava. Kali Linux je jedan od najpopularnijih operacijskih sustava za sigurnosno testiranje koji se detaljnije predstavio u radu. Dolazi s velikim brojem ugrađenih alata za provođenje sigurnosnih napada i testiranje probojnosti, a jedan od njih je „sqlpmap“ koji omogućuje automatizirano izvršavanje napada SQL ubrizgavanjem i „nmap“ koji omogućuje skeniranje portova.

Također, rad je obuhvatio i sigurnosne timove koji se bave probijanjem i testiranjem sigurnosti raznih sustava i Web aplikacija. Sigurnosni tim čine crveni, plavi i ljubičasti tim. Crveni tim pokreće napad, plavi tim se brani od njega, ljubičasti tim može predstavljati ili koordinatore koji će nadzirati crveni i plavi tim kako ne bi bilo prevelike razlike u kvaliteti istih, ili može označavati suradnju i dobru komunikaciju između crvenog i plavog time, a upravo te dvije boje čine ljubičastu.

U radu se predstavio i pojam zero-day koji predstavlja sve neotkrivene ranjivosti i propusti, a koji nisu poznati programerima i kibernetičkim stručnjacima. Napadač može iskoristiti neotkrivene ranjivosti te na taj način ugroziti normalno funkcioniranje Web aplikacija. Postoje mnogi programi nagrađivanja koji nude nagrade za otkrivanje zero day ranjivosti. Nagrade često znaju doseći i nekoliko milijuna dolara, ovisno o kritičnosti i složenosti ranjivosti koje se otkrilo. Otkrivanje sigurnosnih propusta nije nimalo lako te zahtijeva posjedovanje velike količine znanja iz područja sigurnosti Web aplikacija, kao i ustrajnost i strpljivost. Postoje baze otkrivenih ranjivosti u kojima se nalazi veliki broj otkrivenih ranjivosti, kao i ranjivih softvera. ExploitDB sadrži preko 7 000 otkrivenih ranjivosti te je stoga jedna od najpopularnijih i najdetaljnijih baza otkrivenih ranjivosti.

Kao praktični dio, realizirala se Web aplikacija "Matix", koja je bila pokrenuta na lokalnom poslužitelju, a služila je kao podloga za testiranje sigurnosti i provođenje napada. Napadi su se bazirali na OWASP TOP 10 2017 ljestvici koja obuhvaća deset najčešćih napada na Web aplikacije i Web mjesta. Također, kao napadi na Web poslužitelj, obradili su se napadi DoS i DDoS koji šalju veliki broj paketa prema poslužitelju i time onemogućuju normalno funkcioniranje raznih Web aplikacija. Koristili su se i alat Burp Suite te OWASP ZAP s kojima se skeniralo i simuliralo napade na Web aplikaciju Matix. Na kraju se doradila već postojeća ranjiva Web aplikacija i na taj način stvorila nova sigurna Web aplikacija Matix.

Izrađivanje ovog završnog rada je zahtijevalo veliku količinu vremena utrošenu u proučavanje materijala, gledanje videozapisa i razumijevanje raznih napada. Također, proučavanjem ranjivosti, dodatno se shvatilo koliko je bitno zaštititi Web aplikaciju prilikom izrade i implementacije iste. Naposljetku, stekla su se razna znanja koja će definitivno biti korisna u nadolazećoj struci.

Popis literature

[1] Internet users distribution in the world - 2020Q1, [Na internetu]. Dostupno: https://www.internetworldstats.com/stats.htm#links. [Pristupano srpanj 2020]

[2] Total number of Websites, [Na internetu]. Dostupno: https://www.internetlivestats.com/total-number-of-websites/. [Pristupano srpanj 2020]

[3] Simson Garfinkel, Gene Spafford, Web Security, Privacy and Commerce, Second edition, O'Reilly Media, Inc., Sebastopol, 2002.

[4] Hypertext Transfer Protocol - HTTP/1.0, [Na internetu]. Dostupno: http://www.ietf.org/rfc/rfc1945.txt. [Pristupano srpanj 2020]

[5] Hackers attack every 39 seconds, [Na internetu]. Dostupno: https://www.securitymagazine.com/articles/87787-hackers-attack-every-39-seconds. [Pristupano srpanj 2020]

[6] Fireeye cyber-map threat, [Na internetu]. Dostupno: https://www.fireeye.com/cyber-map/threat-map.html. [Pristupano kolovoz 2020]

[7] B. Komarić (2018.), Cyber sigurnost – zašto je trebate shvatiti ozbiljno? – Racunalo.com, [Na internetu]. Dostupno: https://www.racunalo.com/cyber-sigurnost-zasto-je-trebate-shvatiti-ozbiljno/. [Pristupano srpanj 2020]

[8] A. Aldairi i L. Tawalbeh, Cyber Security Attacks on Smart Cities and Associated Mobile Technologies, 2017, [Na internetu]. Dostupno: https://www.researchgate.net/publication/317548513_Cyber_Security_Attacks_on_Smart_Cities_and_Associated_Mobile_Technologies. [Pristupano srpanj 2020]

[9] Zaštita podataka i privatnost na internetu - Europa.eu, [Na internetu]. Dostupno: https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_hr.htm. [Pristupano srpanj 2020]

[10] Zaštita osobnih podataka, [Na internetu]. Dostupno: https://www.europarl.europa.eu/factsheets/hr/sheet/157/zastita-osobnih-podataka. [Pristupano srpanj 2020]

[11] Opća uredba o zaštiti podataka - zakon.hr, [Na internetu]. Dostupno: https://www.zakon.hr/z/1021/Op%C4%87a-uredba-o-za%C5%A1titi-podataka---Uredba-%28EU%29-2016-679. [Pristupano srpanj 2020]

[12] Zaštita osobnih podataka - pisalica.com, [Na internetu]. Dostupno: https://pisalica.com/zastita-osobnih-podataka/. [Pristupano srpanj 2020]

[13] Top 10 operating systems for ethical hackers and penetration testers, [Na internetu]. Dostupno: https://medium.com/lotus-fruit/top-10-operating-systems-for-ethical-hackers-and-penetration-testers-2020-list-b523b611cdbb. [Pristupano kolovoz 2020]

[14] Our Most Advanced Penetration Testing Distribution, Ever, [Na internetu]. Dostupno: https://www.kali.org/. [Pristupano kolovoz 2020]

[15] BlackBox Linux, [Na internetu]. Dostupno: https://linux.backbox.org/. [Pristupano kolovoz 2020]

[16] Parrot Security, [Na internetu]. Dostupno: https://parrotlinux.org/. [Pristupano kolovoz 2020]

[17] DEFT Linux a Linux Distribution For Computer Forensics, [Na internetu]. Dostupno: http://www.linuxandubuntu.com/home/deft-linux-a-linux-distribution-for-computer-forensics. [Pristupano kolovoz 2020]

[18] Samuraiwtf, [Na internetu]. Dostupno: https://github.com/SamuraiWTF/samuraiwtf. [Pristupano kolovoz 2020]

[19] Samurai Web Testing Framework, [Na internetu]. Dostupno: http://www.samurai-wtf.org/. [Pristupano kolovoz 2020]

[20] The Difference Between Red,Blue, and Purple teams, [Na internetu]. Dostupno: https://danielmiessler.com/study/red-blue-purple-teams/. [Pristupano kolovoz 2020]

[21] Red Team VS Blue Team: What's The Difference?, [Na internetu]. Dostupno: https://purplesec.us/red-team-vs-blue-team-cyber-security/. [Pristupano kolovoz 2020]

[22] Cybersecurity Red Team Versus Blue Team - Main Differences Explained, [Na internetu]. Dostupno: https://securitytrails.com/blog/cybersecurity-red-blue-team. [Pristupano kolovoz 2020]

[23] Zero day ranjivosti, [Na internetu]. Dostupno: https://www.cert.hr/wp-content/uploads/2019/04/NCERT-PUBDOC-2010-01-289.pdf. [Pristupano kolovoz 2020]

[24] Zero-day vulnerability database, [Na internetu]. Dostupno: https://www.zero-day.cz/database/. [Pristupano kolovoz 2020]

[25] TOP 10 exploit databases for finding vulnerabilities, [Na internetu]. Dostupno: https://null-byte.wonderhowto.com/how-to/top-10-exploit-databases-for-finding-vulnerabilities-0189314/. [Pristupano kolovoz 2020]

[26] Exploit Database, [Na internetu]. Dostupno: https://www.exploit-db.com/. [Pristupano kolovoz 2020]

[27] Zerodium-Our Exploit Acquisition Program, [Na internetu]. Dostupno: https://zerodium.com/program.html. [Pristupano kolovoz 2020]

[28] Zero-click mobile phone attacks-and how to avoid them, [Na internetu]. Dostupno: https://nakedsecurity.sophos.com/2020/04/30/zero-click-mobile-phone-attacks-and-how-to-avoid-them/. [Pristupano kolovoz 2020]

[29] OWASP, [Na internetu]. Dostupno: https://owasp.org/. [Pristupano kolovoz 2020]

[30] Mirko Maleković, Kornelije Rabuzin, Uvod u Baze podataka, Fakultet organizacije i informatike Sveučilište u Zagrebu, Varaždin, 2016.

[31] SQL injekcija, [Na internetu]. Dostupno: http://sigurnost.zemris.fer.hr/ns/malware/2007_zelanto/sql.html. [Pristupano kolovoz 2020]

[32] Sqlmap, [Na internetu]. Dostupno: http://sqlmap.org/. [Pristupano kolovoz 2020]

[33] Sqlmapproject/sqlmap, [Na internetu]. Dostupno: https://github.com/sqlmapproject/sqlmap. [Pristupano kolovoz 2020]

[34] How to prevent SQL injection attacks, [Na internetu]. Dostupno: https://www.ptsecurity.com/ww-en/analytics/knowledge-base/how-to-prevent-sql-injection-attacks/. [Pristupano kolovoz 2020]

[35] Brute force napadi, [Na internetu]. Dostupno: https://www.cert.hr/wp-content/uploads/2019/04/CCERT-PUBDOC-2007-08-201.pdf. [Pristupano kolovoz 2020]

[36] Brute Force a Website Login Page with Burp Suite, [Video datoteka]. Dostupno: https://www.youtube.com/watch?v=25cazx5D_vw&t=225s. [Pristupano kolovoz 2020]

[37] What is reCAPTCHA, [Na internetu]. Dostupno: https://www.google.com/recaptcha/about/. [Pristupano kolovoz 2020]

[38] Estimating password-cracking times, [Na internetu]. Dostupno: https://www.betterbuys.com/estimating-password-cracking-times/. [Pristupano kolovoz 2020]

[39] 5 Most Common Encryption Algorithms and the Unbreakables of the Future, [Na internetu]. Dostupno: https://blog.storagecraft.com/5-common-encryption-algorithms/. [Pristupano kolovoz 2020]

[40] Secure your site with HTTPS, [Na internetu]. Dostupno: https://support.google.com/webmasters/answer/6073543?hl=en. [Pristupano rujan 2020]

[41] How to | OWASP ZAP finding vulnerabilities, [Video datoteka]. Dostupno: https://www.youtube.com/watch?v=0ThNgh4BGKY&ab_channel=BlackHatEthicalHacking. [Pristupano rujan 2020]

[42] Analiza NMAP alata, [Na internetu]. Dostupno: https://www.cert.hr/wp-content/uploads/2006/04/CCERT-PUBDOC-2006-01-147.pdf. [Pristupano kolovoz 2020]

[43] Basic Guide to NMAP (Kali Linux 2.0), [Video datoteka]. Dostupno: https://www.youtube.com/watch?v=T3XhXPGFdrI&t=640s&ab_channel=JackkTutorials. [Pristupano rujan 2020]

[44] How to close the specific port in linux | Kali linux | Fuser, [Video datoteka]. Dostupno: https://www.youtube.com/watch?v=q_AozvRSlyw&ab_channel=Techbooster. [Pristupano rujan 2020]

[45] What are Denial of Service (DoS) attacks? DoS attacks explained, [Na internetu]. Dostupno: https://us.norton.com/internetsecurity-emerging-threats-dos-attacks-explained.html. [Pristupano kolovoz 2020]

[46] DDoS napad, [Na internetu]. Dostupno: https://www.cert.hr/wp-content/uploads/2019/04/CCERT-PUBDOC-2008-09-240.pdf. [Pristupano kolovoz 2020]

[47] Comprehensive DDoS Protection, [Na internetu]. Dostupno: https://www.cloudflare.com/ddos/. [Pristupano kolovoz 2020]

[48] Analiza XSS sigurnosnih propusta, [Na internetu]. Dostupno: https://www.cert.hr/wp-content/uploads/2006/07/CCERT-PUBDOC-2006-05-157.pdf. [Pristupano kolovoz 2020]

[49] Provjera XSS i SQL Injection ranjivosti Exploit me skupom alata, [Na internetu]. Dostupno: https://www.cert.hr/wp-content/uploads/2019/04/CCERT-PUBDOC-2008-01-215.pdf. [Pristupano kolovoz 2020]

[50] Cross-site scripting (XSS), [Na internetu]. Dostupno: https://docs.kentico.com/k8/securing-websites/developing-secure-websites/cross-site-scripting-xss. [Pristupano kolovoz 2020]

[51] Preventing Cross-site scripting in PHP, [Na internetu]. Dostupno: https://www.virtuesecurity.com/preventing-cross-site-scripting-php/. [Pristupano kolovoz 2020]

[52] PHP NET, [Na internetu]. Dostupno: https://www.php.net/. [Pristupano rujan 2020]

[53] Insecure Deserialization | Modifying serializet objects using Burp Suite, [Video datoteka]. Dostupno: https://www.youtube.com/watch?v=3qvE9lGOAkI&t=213s&ab_channel=ExploitBlizzard. [Pristupano rujan 2020]

[54] Nikto Web Vulnerability Scanner – Web Penetration Testing, [Video datoteka]. Dostupno:

https://www.youtube.com/watch?v=GH9qn_DBzCk&t=206s&ab_channel=HackerSploit. [Pristupano rujan 2020]

[55] CVE-2017-10271 Detail [Na internetu]. Dostupno: https://nvd.nist.gov/vuln/detail/CVE-2017-10271. [Pristupano rujan 2020]

[56] MICROSOFT WINDOWS SERVER 2008/SERVER 2012 ACTIVE DIRECTORY FEDERATION SERVICES UNSPECIFIED ACCOUNT INFORMATION DISCLOSURE, [Na internetu]. Dostupno: https://vuldb.com/?id.9929. [Pristupano rujan 2020]

Popis slika

Slika 1. Napadi uživo diljem svijeta [6] 5
Slika 2. Sigurnosni timovi 8
Slika 3. Postotak zastupljenosti softvera u bazi ranjivosti [23] 12
Slika 4. Početna stranica ExploitDB [26] 13
Slika 5. ZERODIUM program nagrađivanja za računala/poslužitelje [27] 14
Slika 6. ZERODIUM program nagrađivanja za mobilne uređaje [27] 15
Slika 7. Početna stranica Web aplikacije „Matix“ 16
Slika 8. Pregled aktivnih projekata i kategorija projekata 17
Slika 9. Profil registriranog korisnika 18
Slika 10. Obrazac za kreiranje projekta 19
Slika 11. Moderatorovi projekti 19
Slika 12. Popis korisnika 20
Slika 13. Pomak i virtualno vrijeme 20
Slika 14. Upit koji nije zaštićen od SQL ubrizgavanja 22
Slika 15 Otkrivene baze podataka na lokalnom poslužitelj 22
Slika 16. Napad korištenjem sqlmap alata 23
Slika 17. Baza podataka koju Web aplikacija koristi 23
Slika 18. Podaci zapisani u tablici „korisnik“ 24
Slika 19. SQL ubrizgavanje 25
Slika 20. Rezultat napada SQL ubrizgavanje 25
Slika 21. Mysqli_real_escape_string funkcija 26
Slika 22. Napad sqlmap alatom kada je zaštita implementirana 26
Slika 23. Korisnička imena i lozinke 28
Slika 24. Rezultati napada 30
Slika 25. GET zahtjev nakon uspješne prijave u sustav 30
Slika 26. Blokiran korisnički račun 31
Slika 27. Zapisivanje neuspjele prijave u bazu podataka 31
Slika 28. Blokiranje korisničkog računa 32
Slika 29. Google reCAPTCHA 32
Slika 30. Potreban kod za implementaciju reCAPTCHA-e 33
Slika 31. Snimanje mrežnog prometa pomoću Wireshark alata 34
Slika 32. Snimanje mrežnog prometa sa HTTPS prokokolom prilikom prijave 35
Slika 33. Podaci o proizvodu u XML obliku 36
Slika 34. XXE napad 36
Slika 35. Dostupan popis korisnika 38
Slika 36. Zaštita od prekinute kontrole pristupa 38
Slika 37. Naredba ifconfig 40

Slika 38. Skeniranje portova alatom nmap 40
Slika 39. XSS ranjivosti na obrascima 43
Slika 40. Zlonamjerni kod za izvršavanje XSS napada 44
Slika 41. Izvršavanje alert naredbe 44
Slika 42. Zaštita od XSS napada 45
Slika 43. Zaštita prilikom ispisa podataka 45
Slika 44. Zaštita od XSS napada 46
Slika 45. Korištenje JSON-a 47
Slika 46. Serijalizirani objekt nakon slanja zahtjeva za sudjelovanje 47
Slika 47. Poslan zahtjev za sudjelovanje 48
Slika 48. CVE refrenca 49
Slika 49. OSVDB referenca 49
Slika 50. Dnevnik aktivnosti 51

Popis tablica

Tablica 1. Kombinacija korisničkog imena i ispravne lozinke 28