SQL injekcija kao vrsta kibernetičkog napada

Ćubela, Patrik

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2022./ 2023.

Patrik Ćubela

SQL injekcija kao vrsta kibernetičkog napada

Završni rad

Mentor: dr. sc. Vedran Juričić, izv.prof.

Zagreb, rujan 2023.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

_____

(potpis)

Sadržaj

1. Uvod

Za uspješno međusobno povezivanje u današnjem svijetu, web stranice i aplikacije obrađuju goleme količine osjetljivih podataka, pa je sigurnost ovih sustava postala izuzetno važna. Razvijanje tih aplikacija može se koristiti u pozitivne svrhe i tako olakšati kvalitetu privatnog i poslovnog života, ali novi razvoj pruža i nove prilike. Unatoč značajnom napretku u mjerama kibernetičke sigurnosti, svijet okružuju razne kibernetičke prijetnje s kojima se suočavaju organizacije i pojedinci. Napadači nastavljaju iskorištavati ranjivosti u web aplikacijama, pri čemu je SQL injekcija jedna od najraširenijih i najštetnijih tehnika.

Svrha ovog rada je sagledati područje kibernetičkih napada, a glavni fokus je na SQL injekciju, njena obilježja, potencijalni rizici i utjecaj koje može imati na organizacije ili pojedince. Razumijevajući principe napada SQL injekcijom, shvaća se i ozbiljnost prijetnje koju predstavlja i potencijalne katastrofe koje nosi sa sobom. Opisivanjem djelovanja SQL napada, naglašene će biti zlonamjerne metode koje manipuliraju korisnički unosi za ubacivanje lažnog SQL koda. Istraživanje različitih vrsta i varijanti napada SQL injekcijom, istaknuti će njihove različite karakteristike i obilježja. Osim toga, istraživanje praktičnih mjera za sprječavanje ranjivosti SQL injekcije može pomoći u jačanju sigurnosti web aplikacija i zaštiti vrijednih podataka od ugrožavanja. Za mjeru zaštite pokazat će se učinkovite obrambene strategije i najbolje prakse koji ih se je dobro pridržavati. Predstavit će se primjeri značajnih incidenata u kojima su napadi SQL injekcijom rezultirali značajnim nepogodnostima.

Stjecanjem kvalitetnijeg razumijevanja SQL injekcije i povezanih rizika, organizacije i pojedinci mogu ojačati svoje web aplikacije protiv mogućih napada, čuvajući svoje osjetljive podatke i očuvajući povjerenje svojih korisnika. Prije svega, ovaj rad za cilj ima podići svijest o rješavanju ranjivosti SQL injekcije i pružiti uvid u ublažavanje ove sveprisutne prijetnje.

2. Kibernetički napadi

Kibernetički napadi su radnje koje se čine korištenjem računalne mreže, uglavnom u zlonamjerne svrhe osim kada je riječ o obrani ili namjernom testiranju sa svrhom pronalaska ranjivosti. Koriste se za izmjene, brisanja, oštećivanja ili ograničavanja dostupnosti računalnih podataka ili softvera koje prati neka zloćudna namjera. To uključuje djelomično ili potpuno ometanje normalnog rada ciljanog računala, računalnog sustava ili mreže, uključujući bilo koju povezanu fizičku infrastrukturu koja se oslanja na računala. Također može prouzročiti fizičku štetu i izvan područja računala, računalnih sustava ili mreža (Roscini i Trust, 2014). Znači kada se radi o bilo kakvom neovlaštenom pristupu nekoj mreži kojoj pristupa treće strana, radi se o kibernetičkom napadu. Najpoznatiji termin za osobu koja se služi takvom aktivnosti je haker (engl. *hacker*) (Shruti, 2021).

Karakteristika kibernetičkih napada je da se mogu odvijati u mrežnom području a da u većini slučajeva nisu usko povezani s tom fizičkom lokacijom. Kao rezultat toga, specifično okruženje u kojem se događaju ne može se točno prepoznati, tako da je jako teško odrediti napad na temelju određene geografske lokacije ili neke određene društvene ili kulturne skupine (Jahankhani, Hamid, Al-Nemrat, i Hosseinian-Far, 2014). S razvojem tehnoloških mogućnosti očigledno je da se paralelno razvijaju i nove prilike za izvršavanje ovakvih malicioznih radnji. Zbog ovakvih karakteristika kibernetički kriminal napada svoje mete svakodnevno i postaje sve veći problem, jer je ponekad gotovo ne moguće locirati otkuda napad dolazi ili se locira kada je već prekasno.

Cisco u definiciji kibernetički napada napominje kako namjernu i zlonamjernu radnju mogu provoditi pojedinci ili grupa s ciljem infiltriranja u informacijski sustav drugog pojedinca ili organizacije. Tipično, osoba koja radi ovu aktivnost ima skrivene motive i namjerava steći neku vrstu koristi nad ometanjem mreže žrtve (Cisco.com, 2018). Kao i svaki oblik kriminala ili napada, počinitelje potiču razni motivi zbog kojih su voljni ići dosta daleko kako bi napravili što veću štetu za svoju korist. Prepoznavanje povoda kibernetičkog napada ponekad ima priliku spriječiti takvu situaciju prije nego se ona dogodi.

2.1. Povod kibernetičkih napadača

Kako bi se lakše borilo protiv kibernetičkog napada i uspjelo njihovo spriječavanje u pravo vrijeme, presudno je razmotriti znanje koji kibernetičkih kriminalci imaju i koji je povod njihovog zločina. Upoznavajući se s njihovim procesima i motivacijama koje ih pokreću na ovakvu vrstu ponašanja, mogu se poduzeti značajni koraci u njihovom sprječavanju. S obzirom na to da rizici kibernetičke sigurnosti predstavljaju neke od najznačajnijih problema i izazova ekonomskih i nacionalnih sigurnosti 21. stoljeća, ključno je razumjeti povode koji stoje iza kibernetičkih napadača (The White House, 2009). Unatoč tome, motivi koji pokreću kibernetičke napade s namjerom izazivanja ekonomskih posljedica mogu se razlikovati od onih koji predstavljaju prijetnju nacionalnoj sigurnosti. Osim toga, u brojnim slučajevima, prava namjera i primarni cilj kibernetičkog napada mogu ostati skriveni ili zamagljeni, čak i ako napadač preuzme odgovornost ili ga se uhvati (Shakarian P., Shakarian, J. i Ruef, 2013).

Da bi bilo moguće držati korak i shvatiti određene povode ovakvog kriminala, dolazi se do strategije da se kibernetičke napadače kategorizira, prepoznajući da se pojedinačni napadač može uklopiti u više kategorija. Andress i Winterfeld (2011) za kategorizacija navode tri glavne vrste pojedinaca s unutarnjim pristupom (engl. *insider*). Klasificiraju se na sljedeći način:

1) Nezadovoljni zaposlenici koji bi mogli izvršiti osvetničke radnje ili predstavljati rizik za interne sustave.
2) Osobe vođene financijskim motivima koji bi mogli zloupotrijebiti resurse tvrtke ili iskorištavati sustav za osobnu korist (iako je vrijedno napomenuti da neki od njih mogu imati etičke namjere ili druge razloge).
3) Osobe koje nenamjerno ili nesvjesno pomažu vanjskim napadima, iako sami po sebi nisu primarni napadači.

Iz ove kategorizacije se dobije u uvid kakve vrste napada mogu postojati unutar same tvrtke, sustava, organizacije ili slično. Neki od savjeta su za sprječavanje ovakvih nepovoljnih scena se omogućava dobrom i primjerenom kontrolom zaposlenika te kvalitetnim povjerenjem među svih osoba unutar sustava. Navedeni načini ako svi surađuju imaju priliku iskorijeniti prijetnju u potpunosti. Nakon prve raspodijele koja obuhvaća osobe s unutarnjim pristupom, postoje i pojedinci koji mogu vršiti zločinačke aktivnosti izvan sustava, to jest osobe s vanjskim pristupom (engl. *outsiders*). Han i Dongre (2014) osobe s vanjskim pristupom svrstavaju na temelju motivacija, profesionalnih vještina ili njihove organizacije, pa ih se dijeli u sljedeće skupine:

1) Organizirani napadači
2) Hakeri
3) Amateri

Organizirani napadači se sastoje od raznih skupina poput terorističkih organizacija, hakerskih skupina ili napadača kojima je meta nacionalna država. Terorističke organizacije čine pojedinci koji žele izraziti svoj politički stav ili nanijeti psihičku i fizičku štetu svojim metama, s namjerom postizanja političkih ciljeva ili utjerivanja straha u svoje protivnike ili širu javnost. Hakerske skupine (engl. *hacktivists*) imaju cilj prenijeti političku poruku i iako u većini slučajeva dolazi do određene razine štete, njihova primarna motivacija je podizanje svijesti radi vlasitith ideja, a ne usađivanje straha u svrhu pokretanja promjena. Napadači na nacionalne države angažirani su u prikupljanju informacija i provođenju sabotaže na vladu. Oni su obično opsežno obučeni, dobro financirani i kvalitetno organizirani, često potpomognuti značajnim resursima. Njihovi napadi često su sofisticirani i skrojeni prema određenim ciljevima, najčešće imaju višestruke motive (Cohen i sur., 1998).

Hakeri imaju mnogo definicija koje ih opisuju, a u nekim kontekstima hakiranje ima i pozitivnu ulogu. Uglavnom je interpretacija da je haker netko tko prodire u računalne sustave drugih ljudi bez dopuštenja kako bi saznao informacije ili učinio nešto što se protivi zakonu (Dictionary.cambridge.org., n.d.). Hakiranje obuhvaća aktivnosti kao što su špijunaža (neovlašteno stjecanje tajni iz osobnih, političkih ili kriminalnih razloga), ucjena (zahtijevanje novca, imovine ili ustupaka prijetnjama), krađa (nezakonito stjecanje vrijednih podataka, informacija, intelektualnog vlasništva itd.) i vandalizam (izazivanje namjerne štete zbog pobune ili inata) (Shakarian P., Shakarian, J. i Ruef, 2013). Zbog ovakvih aktivnosti termin hakiranje se automatski percipira negativno, jer u većini slučajeva i je. Uz negativne strane hakiranja važno je napomenuti da se u nekim prilikama nastoji koristiti u dobre svrhe. Tako da u određenim situacijama se ne provodi sa zloćudnom namjerom. Pojam za takvog pojedinca je haker "bijelog šešira" (engl. *„White hat“*) koji identificira ranjivosti u računalnim sustavima ili mrežama s ciljem povećanja njihove sigurnosti, često uz odgovarajuću autorizaciju ili kao dio ugovornog sporazuma sa stranom koja ga je angažirala. S druge strane, haker "crnog šešira" (engl. *„Black hat“*) odnosi se na zlonamjerno iskorištavanje ciljnog sustava sa sudjelovanjem u ilegalnim aktivnostima koji se općenito smatraju klasičnim hakerskim aktivnostima (Shakarian P., Shakarian, J. i Ruef, 2013).

Amaterima se smatraju hakeri s manje stručnosti i iskustva, obično im se dodjeljuje naziv „noob“ ili početnik. Oslanjaju se na alate koji već postoje ili upute koje su javno dostupne na

internetu. Njihovi motivi se kao i kod ostalih razlikuju, jer neki su vođeni znatiželjom ili uzbuđenjem izazova, dok drugi žele pokazati i poboljšati svoje vještine kako bi ispunili uvjete za pridruživanje određenoj grupi, uglavnom nekoj od ove dvije skupine koje su navedene (Andress i Winterfeld, 2011). Iako amateri mogu imati bezopasne namjere, alati koje koriste često su jednostavni, ali moćni. Unatoč njihovim ograničenim vještinama i ograničenjima, imaju potencijal nanijeti značajnu štetu, ili ako steknu više iskustva, mogli bi na kraju prijeći u ozbiljniju prijetnju (Han i Dongre, 2014).

Uz podjelu kibernetičkih napadača koji imaju unutarnji pristup ili vanjski pristup, postoji i podjela gdje se proučavaju tri kategorije na temelju motivacija. One se sastoje od političke, ekonomske i društveno-kulturalne motivacije. Političke motivacije obuhvaćaju različite radnje kao što su ciljano uništenje, ometanje ili kontrolu žrtve, sudjelovanje u špijunaži i korištenje prosvjeda, političkih izjava ili akcija osvete. Drugo su ekonomski motivi gdje su uključene različite aktivnosti, poput krađe intelektualnog vlasništva ili vrijedne ekonomske imovine (financijska sredstva ili podaci o kreditnim karticama), sudjelovanje u prijevarama i špijunaži, razne sabotaže te korištenje ucjena sa svrhom da počinitelj dobije od žrtve neku financijsku korist (Gandhi i sur., 2011). Društveno-kulturalne motivacije obuhvaćaju zabavu, znatiželju i žudnju za publicitetom ili osobnom potvrdom vlastitog ega. Ukorijenjeni društveno-kulturalni problemi su jedni od glavnih povoda mnogih kibernetičkih napadača, istodobno se uz njih mogu prekriti i druge vrste motiva kao na primjer politički povod ili utjecaj na vladinu politiku (Cohen i sur., 1998). Uočava se da postoje raznoliki povodi iz kojih se manifestiraju nedobronamjerne želje kako bi se zadovoljile vlastite potrebe napadača. Raznolikost motiva raznih pojedinaca isto tako donosi i različite vrste napada koje imaju na raspolaganju.

2.2. Tipovi kibernetičkih napada

Danas se u svijetu javljaju mnoge vrste kibernetičkih napada. Razumijevanje različitih tipova ovakvih akcija čine internet mrežu i sustave sigurnijima olakšavajući njihovu zaštitu (Shruti, 2023). Virusi, crvi, trojanski konj i razni slični oblici zloćudnih programa mogu služiti kao sredstva u izvođenju zlobnih planova da se nanese šteta drugima. Zajedno sa zloćudnim programima i softverima postoje i specifičnije kombinacije pojedinih vrsta napada. U tom kontekstu, ukratko će se objasniti neki od najznačajnijih kibernetičkih napada koji mogu utjecati na određeni broj ljudi ili organizacija ovisno o njihovom opsegu i povodu.

2.2.1. Buffer overflow

Ranjivosti prekoračenja međuspremnika (engl. *buffer overflow*) prisutna je od početka računarstva, a nažalost nastavlja biti problem i danas. Obično ih iskorištavaju internetski crvi za širenje, a karakteristika je da utječe na sve vrste softvera što čini ovu ranjivost opasnom. U većini slučajeva proizlaze iz neispravnih unosa ili neuspjeha u dodjeljivanju dovoljno prostora na mjesto pohrane. Ako transakcija prebriše izvršni kod, ona ima potencijal potaknuti nepredvidljivo ponašanje u programu, što rezultira pogrešnim ishodima, kao što su problemi pristupa memoriji ili kompletnog rušenja programa. Primjer takve situacije je C programski jezik visoke razine, jer pretpostavlja da programer snosi odgovornost za osiguranje integriteta podataka. Dok mogućnost C-a daje programerima veći stupanj kontrole i povećava učinkovitost njihovih programa, također može učiniti programe podložnima prelijevanju međuspremnika i curenju memorije ako se ne postupa s odgovarajućim oprezom. Poznati primjer je kada programer želi staviti više bajtova u međuspremnik nego što on ima kapaciteta, iako je radnja u pravilu dopuštena postoji vjerojatnost rušenja programa. U slučaju viška podataka dolazi do prelijevanja i prepisivanja preko susjedne memorije, što potencijalno dovodi do pada programa i s tim će prebrisati drugi podaci. Ovakav postupak može koristiti treća strana da našteti određenom serveru. Takvi se događaji nazivaju buffer overflow (Erickson, 2003).

2.2.2. Malware

Malware je zlonamjerni softver koji je danas jedna od glavnih i ozbiljnih prijetnji na internetu. Zlonamjerni softver dolazi u širokom rasponu varijacija kao što su virus, crv, trojanski konj, rootkit, backdoor, botnet, spyware, adware, ransomware i slično. Ove klase zlonamjernog softvera međusobno se ne isključuju, što znači da jedna instanca zlonamjernog softvera može pokazivati karakteristike povezane s više klasa istovremeno. Jednom kada se uspije ubaciti u sustav, zlonamjerni softver može učiniti raznolike štete. Uglavnom blokira pristup ključnim komponentama mreže, instalira druge maliciozne softvere, prikuplja i krade informacije ili ugrožava određene komponente i onemogućuje kompletni rad sustava (Gandotra, Bansal, i Sofat, 2014). Najbolje zaštite od zlonamjernog programa su kvalitetnim i dobro testiranim antivirusnim programima te zaštitom sigurnosti na računalu ostalim poznatim načinima.

2.2.3. Prijevare

Definitivno jedna od najiritantnijih i najfrustriranijih vrsta napada su prijevare. Korisnici ovih oblika napada se služe različitim taktikama temeljene na e-pošti i ostalim porukama kako bi iskorištavali pojedince da kradu povjerljive i osjetljive podatke. Vrste prijevara koje su vrlo česte i na koje treba paziti su sljedeće (Ttu.edu., 2017).

- Spam: odnosi se na neželjenu e-poštu, trenutne poruke ili poruke društvenih medija. Spam poruke je relativno lako prepoznati i mogu uzrokovati štetu ako ih se otvore ili se stupi u kontakt s njima.
- Phishing: e-pošta koju šalje internetski kriminalac, ali je problem u tome što je prerušena u e-poštu iz legitimnog, pouzdanog izvora. Poruka žrtvu namami da otkrijete osjetljive ili povjerljive informacije.
- Spear Phishing: događa se kada kriminalci dobiju podatke o svojoj meti s web-stranice ili s prostora društvenih mreža. Kada dobiju te podatke, prilagođavaju phishing bliskim temama osobe kojoj šalje kako bi što lakše napravili krađu identiteta ili srodne štete.
- Spoofing: opisuje kriminalca koji se predstavlja kao drugi pojedinac ili dio neke organizacije, s namjerom prikupljanja osobnih ili poslovnih podataka.
- Pharming: djeluje kao zlonamjerna web stranica koja nalikuje legitimnoj web stranici, a koristi se za prikupljanje korisničkih imena i lozinki.

Prijevare se poigravaju sa ljudskim emocijama kako bi potaknule suosjećanje, ljubaznost, strah, brigu, tjeskobu, uzbuđenje i tako dalje. Najbolji načini da se od njih zaštiti su svakako osobne edukacije, jer ako se poznaje princip na temelju čega rade, veće šanse su da ih se potpuno izbjegne i tako zaštiti vlastita sigurnost. Na primjer važno je znati da zahtjev za korisničkim imenom, lozinkom ili drugim osobnim podacima vjerodostojne institucije i organizacije nikada neće tražiti putem e-pošte (Ttu.edu., 2017).

2.2.4. Man in the middle (MITM)

Čovjek u sredini (engl. *Man in the middle*) ili skraćeno MITM je opći koncept koji se koristi za opisivanje situacije u kojoj se napadač ubacuje u komunikaciju između korisnika i aplikacije. Namjera iza takvog napada može biti prisluškivanje ili lažno predstavljanje jedne od uključenih strana, stvarajući iluziju redovne razmjene informacija. Svrha ove aktivnosti je ukrasti osobne podatke, kao što su informacije za prijavu, podaci o računu i brojevi kreditnih kartica. Mete su obično korisnici financijskih aplikacija, SaaS (engl. *Software as a Service*)

poslovanja, web stranica za e-trgovinu i drugih web stranica na kojima je potrebna prijava. Dvije su uobičajene ulazne točka koje omogućavaju MITM napade, jedna je nakon što zlonamjerni softver probije sigurnost, onda nakon toga napadač može instalirati softver za obradu svih podataka inficiranog korisnika, a druga situacija je korištenje nesigurne javne veze za izvršenje MITM napada. Savjet zaštite je izbjegavati WiFi veze koje nisu zaštićene lozinkom, nesigurne web stranice i probati izbjegavati nezaštićene javne veze (Imperva, 2019).

2.2.5. Zero day exploit

Prijetnja nultog dana (engl. *Zero day exploit*) je usmjeren na iskorištavanje ranjivosti softvera za koje se zna da su nepoznati stvoritelju softvera ili antivirusnim programima. Napadač uglavnom detektira koje su to ranjivosti prije bilo koje druge strane i brzo ju iskorištava da načini štetu. Ovi napadi su ekstremno opasni jer u većini slučajeva obrana nije postavljena pa ih to čini ozbiljnom sigurnosnom prijetnjom. Tipično su ciljani web preglednici, privitci e-pošte ili određene datoteke kao što su Flash, Word, Excel ili PDF. Budući da je znanje o ranjivostima jako vrijedno, postoje tržišta koja trguju s ranjivostima nultog dana. Po definiciji, još uvijek ne postoje rješenja ili antivirusni potpisi za ovakve napde, što je i njihov cilj da budu teški za otkrivanje. Međutim, postoji nekoliko načina za otkrivanje dosad nepoznatih softverskih ranjivosti koji mogu smanjiti zlonamjerne prilike. Strategije koje imaju priliku otkriti neistražene slabosti su skeniranje ranjivosti (engl. *Vulnerability scanning*) koji može detektirati neke slabosti kako bi ih se znalo pokrpati. Nadalje, postoji inicijativa nultog dana (engl. Zero Day Initiative) kojoj je cilj stvoriti raznoliku zajednicu istraživača koji proučavaju ranjivosti i koji mogu otkriti sigurnosne prijetnje prije hakera kako bi upozorili na pravu obranu u pravo vrijeme (Imperva, 2019). Sve u svemu, ranjivost nultog dana je vrlo opasna radnja protiv koje se nije ugodno boriti.

2.2.6. Denial of service (DOS)

„Mrežni napadi koji su usredotočeni na onemogućavanje komunikacije nazivamo napadima s uskraćivanjem usluge (engl. Denial of Service – DoS). DoS napad sprječava pristup korisnicima da komuniciraju u okviru napadnute računalne mreže ili onemogućavaju poslužitelja da poslužuje normalne usluge“ (Džanko i sur., 2017). DoS to postiže preplavljivanjem prometa ili slanjem pretjerane količine informacija na server svoje žrtve. Ciljani su uglavnom web poslužitelji visokoprofitnih organizacija kao što su bankarske, trgovačke i medijske tvrtke ili vladine i trgovačke organizacije. Iako DoS napadi obično ne rezultiraju krađom ili gubitkom značajnih informacija ili druge imovine, koštaju puno vremena

i novca kako bi se učinjena šteta popravila. Buffer overflow je isto oblik DoS napada, zbog karakteristike napada da pošalje više prometa na mrežnu adresu nego za što je sustav originalno programiran, što uzrokuje preopterećenje i onemogućuje normalno funkcioniranje ciljanog sustava, programa ili slično (Paloalto Networks, 2019). Radi se na različitim sustavima koji su razvijeni radi ranog otkrivanja i prevencije DoS napada na drugačijim dijelovima mrežne infrastrukture. Ulaže se uzastopni trud za izradu novijeg i boljeg modela za komuniciranje. Suvremene tehnologije za sigurnost u zadnje vrijeme su razvile mehanizme za obranu od većine oblika DoS napada, ali uz to je evoluirao i novi oblik zvan DDos (engl. *Distributed Denial of Service*) koji koristi više pojedinaca da vrši napade s raznih mjesta raspoređenih na cijelom Internetu. Zbog jedinstvenih karakteristika DDoS-a, on se smatra visokom prijetnjom i predstavlja veliku zabrinutost za organizacije kojih je strah biti meta ovakvog oblika napada (Džanko i sur., 2017).

3. SQL injekcija

SQL Injekcija (engl. *SQL injection*) ili skraćeno SQLi je ranjivost koja nastane kada napadač ima mogućnost utjecati na funkcioniranje strukturiranog upitnog jezika (engl. *Structured Query Language*). Zlonamjerni počinitelji utječu na upite tako da vraćaju informacije iz baza podatka koje se u normalnom slučaju ne bi vraćale ili pokazivale (Clarke i sur., 2009). Strukturirani upitni jezik (SQL) koristi se za ispitivanje, upravljanje i nadgledanje sustava baza podataka kao što su Microsoft SQL Server, MySQL, Oracle i slično. Popularan je upitni jezik koji se često koristi u svim vrstama aplikacija i često je korišten jer se dobro integrira s različitim programskim jezicima. Iako temeljna primjena SQL-a ostaje dosljedna u svim podržanim sustavima baza podataka, važno je napomenuti da svaki sustav može imati specifične složenosti i razlike. Sustavi baza podataka se naširoko koriste kako bi ponudili pozadinske funkcije za razne web aplikacije. U kontekstu ovih aplikacija, podaci koje daje korisnik često se koriste za dinamičku konstrukciju SQL naredbi koje izravno komuniciraju s bazom podataka.

Napad SQL injekcijom odnosi se na akciju kojoj je cilj potkopati izvornu svrhu aplikacije podnošenjem SQL naredbi koje zločinac daje izravno u pozadinsku bazu podataka (engl. *backend database*). Učinak uspješnog napada SQL injekcijom može varirati ovisno o web aplikaciji i načinu na koji postupa s podacima prije konstruiranja SQL izjave. Potencijalne sigurnosne posljedice mogu varirati od zaobilaženja autentifikacije i otkrivanja informacija do omogućavanja distribucije štetnog koda korisnicima aplikacije (Cisco.com, n.d.). U pitanju je vrlo destruktivna ranjivost koja može ozbiljno utjecati na poslovanje. Ima potencijal razotkriti sve osjetljive informacije pohranjene u bazama podataka određene aplikacije, što uključuje vrijedne podatke poput korisničkih imena, lozinki, imena, adresa, telefonskih brojeva i podataka o kreditnoj kartici. Također nije ograničeno da samo utječe na web aplikacije, nego predstavlja opasnost za bilo koji oblik koda koji prima podatke od nepouzdanog izvora i koristi taj unos za stvaranje dinamičkih SQL izjava. Najčešći napadi se događaju kada dobivene vrijednosti s web prostora ili drugog ulaznog parametra nisu kvalitetno osigurane, a zbog nedostatak odgovornosti i razumijevanje elemenata od strane programera može otvoriti vrata za ubacivanje malicioznih SQL izraza (Clarke i sur., 2009).

3.1. Primjer SQLi

Prije pokazivanja primjera prikazat će se slikovni prikaz SQL injekcije na slici 1, sa svrhom boljeg razumijevanja kako funkcionira ovaj oblik napada i boljeg shvaćanja teme.

Slika 1 Prikaz SQL injekcije (InfoSec Insights, 2020).

Za primjer će se uzeti SQL upit prikazan kao programski kod 1, cilj je pronaći retke u tablici "TABLE_USERS" gdje je korisničko ime jednako "ADMINISTRATOR", a lozinka je jednaka "ROOT".

```
SELECT * FROM TABLE_USERS WHERE USERNAME =
'ADMINISTRATOR' and PASS = 'ROOT'
```

Programski kod 1 SQL upit (Sadeghian, Zamani and Abdullah, 2013).

Napad SQL injekcije počinje kada haker otkrije ranjivost sustava kojeg podupire SQL, zatim ubrizgava neku varijabilnu vrijednost kao maliciozan SQL upit (engl. *Malicious Request*). Modificirani upit (engl. *Modified Query*) je rezultat konkatenacije to jest ulanjčavanja vanjske varijable i glavne SQL izjave. Zatim baza podataka potvrđuje i izvršava modificirani zloćudni upit. Pokretanjem će se vratiti izlazni upit (engl. *Query Output)* koji će izvući sve retke iz "TABLE_USERS" sa korisničkim imenom "ADMINISTRATORA" ignorirajući njihove lozinke (Sadeghian, Zamani and Abdullah, 2013). Završni upit je prikazan kao programski kod 2

```
SELECT * FROM TABLE_USERS WHERE USERNAME =
'ADMINISTRATOR' and PASS = '' OR 1=1 --'
```

Programski kod 2 SQL injekcija tautologijom (Sadeghian, Zamani and Abdullah, 2013).

Prikazana injekcija prisiljava SQL naredbu da vrati rezultat ignorirajući sve "WHERE" uvjete. Napadač to radi postavljanjem dodatnog uvjeta konkatenacije s kriterijem tautologije "OR" i "1=1". Svrha korištenja tautologije je vraćanje uvjeta koji je u svakom slučaju istinit. Ovakvim upitom hakeru je omogućena eksfiltracija podataka (engl. *Data exfiltration*) što može dovesti do neovlaštenog pristupa osjetljivim informacijama i potencijalnih katastrofa u sigurnosti za bilo koji entitet koji se oslanja na napadnutu aplikaciju.

3.2. Utjecaj napada SQLi

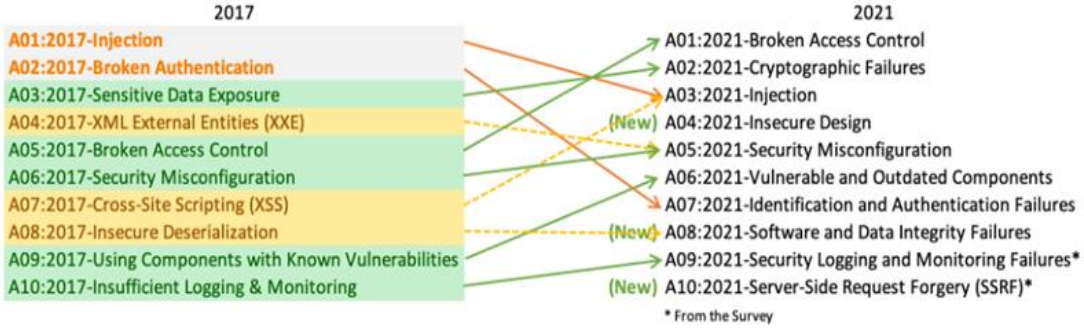
Učinci uspješne radnje SQL injekcijom se razlikuju ovisno o aplikaciji koja je napadnuta i načinu kako aplikacija obrađuje podatke dostavljene korisniku. Stranica Cisco.com (2016) navodi da izvršavanje SQL injekcije povezuje i utječe na sljedeće radnje:

- Neovlašteni pristup osjetljivim podacima
    - Otkrivanje informacija: omogućuje napadaču da dobije, bilo izravno ili neizravno, osjetljive informacije u bazi podataka.
    - Zaobilaženje autorizacije: mogućnost prijave na aplikaciju, potencijalno s administrativnim ovlastima, bez unosa važećeg korisničkog imena i lozinke.
    - Ugroženi integritet podataka: uključuje promjenu sadržaja baze podataka. Dolazi do narušavanje web-stranice ili se povećavaju prijetnje za umetanje zlonamjernog sadržaja na inače bezopasne web-stranice.
    - Ugrožena dostupnost podataka: brisanje informacija s namjerom nanošenja štete ili brisanje podataka dnevnika ili revizije u bazi podataka.
- Izvršenje naredba daljinskim upravljanjem:
    - Izvođenjem naredbi kroz bazu podataka haker ima priliku ugroziti operativni sustav. Često se iskorištavaju postojeće pohranjene procedure koje su unaprijed definirane i s time se izvršavaju naredbe operativnog sustava od glavnog računala. Najpoznatija varijanta ovog napada koristi pohranjenu proceduru xp_cmdshell koja je uobičajena za instalacije Microsoft SQL Servera ili stvaranje vanjskog poziva procedure na Oracle bazama podataka.

3.3. Ozbiljnost SQLi

Općenito su napadi injekcijom usredotočeni na iskorištavanje ranjivosti povezanih s ubrizgavanjem, a to je široki raspon slabosti u području kibernetičke sigurnosti koje obuhvaćaju neke od najvećih rizika za sigurnost aplikacija. U pitanju su toliko rašireni i potencijalno destruktivni napadi da ih OWASP (Open Web Application Security Project) navodi kao treću najrizičniju kategoriju sigurnosti za web aplikacije za 2021. godinu, a 2017. je injekcija uzela prvo mjesto na toj listi kao što se može vidjeti na slici 2.

Iako postoje brojni načini na koje se napadi mogu izvoditi, temeljni element koji dijele gotovo sve ovakve napade je sposobnost zlonamjernih hakera da ubace neprovjereni korisnički unos izravno u aplikacijski kod koji se izvršava (Invicti, 2022).



Slika 2 Top deset napada (OWASP, 2021).

SQL injekcija je pogotovo postao česti problem s web stranicama koje se oslanjanju na baze podataka. Greške sustava se lako otkrivaju i iskorištavaju. Zbog tih značajki, svaka stranica ili softverski paket, pa iako imaju minimalnu bazu korisnika vjerojatno će biti žrtva pokušaja ovakvog napada. SQLi je vrlo česta s aplikacijama koje koriste PHP i ASP zbog prevladavanja starijih funkcionalnih sučelja. Dok aplikacije s J2EE i ASP.NET imaju manju vjerojatnost da budu mete ovakvih ranjivosti, razlog je taj što oni imaju pogodnosti dostupnih programskih sučelja. Procjenjivanje ozbiljnost napada ograničena je vještinom i maštom pojedinca koji izvršava injekciju, te kvaliteti obrane i protumjera (OWASP, 2013). SQL injekcije su jedne od najstarijih i najopasnijih ranjivosti web aplikacija i mnogi resursi tretiraju ovo kao vrlo opasan napad. Kroz godine otkrili su se mnoge vrste ranjivosti SQLi, a općenito se tretira sa sve većom ozbiljnošću (Invicti, 2022).

3.4. Primjer povijesnih napada SQLi

Prikupljanje preciznih informacija o broju organizacija koje su bile ugrožene i ciljane kroz povijesne napade SQL injekcijom je izazovno. Koriste se razne vrste SQL injekcije i često se znaju vrlo dobro prikriti. Mediji se često fokusiraju na učinkovitost ovakvih napada i širenju panike koju napad stvori, zbog toga su ovakve teme dobile značajan publicitet u zadnje vrijeme. Prikazat će se neki od javno dostupnih izvora gdje je SQL injekcija bila glavni krivac:

- Siječanj, 2006. godine: Web stranica vlade Rhode Islanda bila je meta ruskog napadača koji je izveo napad SQL injekcijom, što je rezultiralo neovlaštenim

preuzimanjem podataka o kreditnim karticama koje pripadaju pojedincima uključenim u transakcije s vladinim agencijama (CIS, 2011).

- Prosinac, 2006. godine: američki diskontni trgovac, postao je žrtva uspješnog kibernetičkog napada u kojem su počinitelji uspjeli infiltrirati u TJX baze podataka i nezakonito doći do milijuna podataka o platnim karticama (Ljubičić, Jakšić and Poščić, 2020).
- Kolovozu, 2007. godine: Web-mjesto Ujedinjenih naroda je napadnuto ubrizgavanjem SQLi kako bi se prikazale anti-američke poruke (Ljubičić, Jakšić and Poščić, 2020).
- Siječanj, 2008. godine: značajan broj osobnih računala bio je kontaminiran štetnim kodom koji je izvodio automatizirane napade iskorištavanjem ranjivosti u bazi podataka Microsoft SQL Servera, koristeći tehnike zlonamjernog ubacivanja SQL koda (CIS, 2011).
- Srpanj, 2008. godine: Malezijska web stranica antivirusnog alata "Kaspersky" bila je ciljana i komprimirana ubrizgavanjem SQLi (Ljubičić, Jakšić and Poščić, 2020).
- Zima, 2008. godine: Jedna od najvećih provala SQL napadom u povijesti. Albert Gonzalez, čovjek koji je poznat po svojim hakerskim pothvatima, vodio je hrabar napad na Heartland Payment Systems, tvrtku koja obrađuje transakcije kreditnim i debitnim karticama. Tijekom nekoliko tjedana, Gonzalezova grupa koristila je zlokobnu tehniku hakiranja, kako bi se zaobišla obranu i infiltrirala sustav. Komprimirano je 130 milijuna brojeva kreditnih i debitnih kartica (Moes, 2023).
- Srpanj, 2010. godine: Koristeći SQLi, istraživač računalne sigurnosti iz Južne Amerike uspio je probiti sigurnost poznate torrent platforme, The Pirate Bay, dobivši neovlašteni pristup osjetljivim informacijama. Iskorištavanjem ove ranjivosti, istraživač je došao do detalja povezanih s korisničkim računima, uključujući IP adrese i evidenciju torrenta koje je svaki korisnik postavio na stranicu (CIS, 2011).
- Srpanj, 2010. godine: Počinitelji podrijetlom iz Japana i Kine iskoristili su SQL ranjivost unutar web stranice tvrtke New Beat za internetsku trgovinu. Nezakonito su pribavljeni osobni podaci od 12.191 korisnika. Nakon toga, prijavljeno je oko 300 slučajeva zlouporabe kreditnih kartica, uključujući ukradene podatke iz ovog konkretnog slučaja (CIS, 2011).

- Ožujak, 2011. godine: Ubrizgavanje SQL koda korišteno je za ciljanje i kompromitiranje web stranice "mysql.com (CIS, 2011).
- Lipanj, 2011. godine: Grupa napadača optužena je za korištenje SQLi za probijanje sigurnosti Sony web stranice, čime su naštetili osobnim podacima od raznih korisnika. Otprilike milijun korisničkih računa bilo je pogođeno ovim incidentom, što je rezultiralo stjecanjem korisničkih lozinki, kupona i ključeva proizvoda za Sony usluge i robu (CIS, 2011).
- Lipanj, 2011. godine: Web stranica poznate pjevačice Lady Gage postala je žrtva ubacivanja SQLi. Napad je rezultirao neovlaštenim prikupljanjem podataka od nekoliko tisuća fanova, uključujući njihove e-mail adrese. Vjeruje se da su te informacije kasnije iskorištene za dodatne napade, vjerojatno u obliku lažnih e-mailova slanih obožavateljima koji nude ekskluzivne proizvode povezane s pjevačicom. Umjesto da dobiju ekskluzivan sadržaj kao što je obećano, fanovi su nesvjesno preuzeli zlonamjerne programe (CIS, 2011).
- Ljeto, 2012. godine: Organizirana skupina hakera poznata kao D33Ds Company dobro orkestriranim napadom SQL injekcije, rezultirala je međunarodnim kibernetičkim napadom koji utjecao je na nevjerojatnih 450.000 Yahoo! korisnika diljem svijeta (Moes, 2023).
- Jesen, 2015. godine: Grupa mladih hakera, predvođena 17-godišnjakom, iskoristila je ranjivost SQLi kako bi zaobišla obranu britanske telekomunikacijske tvrtke TalkTalk koja je pretrpjela veliku štetu zbog curenja informacija za 157 000 korisnika, uključujući brojeve bankovnih računa (Moes, 2023).
- 2020. godine: Meta je bila Estonska zdravstvena baza podataka. Napadom SQLi, kibernetički kriminalci potencijalno su ugrozili zdravstvene podatke gotovo svih građana Estonije, napad je zahvatio cijelu naciju. Financijska šteta ostaje neotkrivena, ali učinak je bio razoran, razotkrivajući privatne zdravstvene podatke i poljuljavši povjerenje javnosti (Moes, 2023).

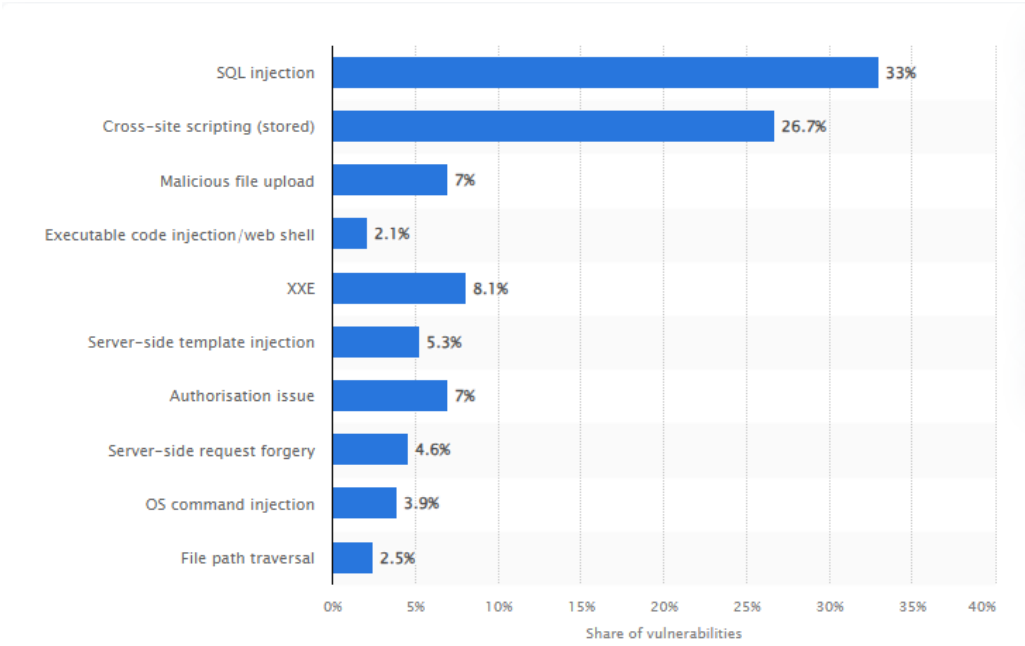
Prethodni primjeri naglašavaju da nitko ili nikakva organizacija nije iznimka od ovakvih zlonamjernih radnji. Može se i uočiti kako SQL injekcija može nadopunjavati i druge tipove kibernetičkih napada. Najčešći povodi su naštetiti aplikacijama radi ostvarivanja financijskih koristi ili nekog pojedinačnog zadovoljstva. Treba imati na umu da u nekim slučajevima velike tvrtke pokušavaju sakriti taj problem pa vjerojatno postoji i još drugačijih slučaja SQL injekcije.

3.5. Značajne činjenice SQLi

Tijekom postojanja relacijskih baza podataka, oblici napada SQLi bili su stalna prijetnja. Otkriće SQLi se javlja 25. prosinca 1998., kada ga je prvi put pronašao i opisao sigurnosni istraživač Jeff Forristal u magazinu Phrack 54. Nakon toga uspostavljen je rječnik uobičajenih ranjivosti i izloženosti (engl. *Common Vulnerabilities and Exposures*) ili skraćeno CVE, koji je ključan u praćenju i obavještavanju o poznatim softverskim propustima. Nevjerojatno je da su SQL injekcije dosljedno zadržale svoje mjesto na popisu 10 najznačajnijih CVE ranjivosti. U razdoblju od 2003. do 2011. zabilježeno je ukupno 3.260 ranjivosti povezanih s SQLi. Ovi podaci naglašavaju stalni značaj rješavanja i zaštite od SQLi prijetnji tijekom godina (Team, I.S., 2013). Dodatno vrijeme i trud za ugradnju sigurnog kodiranja, statičkog i dinamičkog testiranja ili skeniranje sustav za sprječavanje upada (engl. *Intrusion prevention system*) može biti daleko manji od cijene uspješnog napada SQL injekcijom. Za uspješnu provedbu SQLi napada cijene posljedica se procjenjuju od 196.000 dolara do više niza napada koji koštaju oko stotine milijuna dolara. (Hyslip i Horner, 2017). U 2012. godini Prema predstavniku Barclaycarda nevjerojatnih 97% ranjivosti podataka je rezultat SQLi. U kratkom razdoblju, od kraja 2011. do početka 2012., više od milijun web stranica pretrpjelo je ovakav napad, a 2008. je čak došlo do značajnog ekonomskog poremećaja zbog SQLi. (Team, I.S., 2013).

Iako je zaštita od ove vrste zlonamjernog softvera jača nego ikad, još uvijek postoji visok rizik. Gotovo sve web stranice i javni sustavi zahtijevaju pozadinske baze podataka za pohranu podataka i ispravno funkcioniranje. Stoga nije čudno da je 42% napada upravo SQL injekcijom, a prijetnja se proteže i na interne sustave sa manjom mjerom oko 12% (Čitaković, 2023). SQLi se može koristiti i za mješovite napade uz crve i trojance. Takav napad s miješanim prijetnjama kombinira različite vrste zlonamjernog softvera, poput trojanaca, virusa, crva ili slično. Jedan značajan primjer bio je olujni crv (engl. *Storm Worm*), pokrenut u siječnju 2007. godine. Ime je dobio jer je početni neželjeni email bio naslovljen: "230 mrtvih u oluji koja je zahvatila Europu" (Danchev, 2008). Crv u kombinaciji sa SQLi je ubacio zlonamjerne domene u ranjiva mjesta na internetu kako bi se proboj dalje proširio. To je rezultiralo više od milijun infekcija i koštalo je milijune dolara za popravak (Čitaković, 2023). Unatoč napretku u sigurnosti otvorenih web aplikacija, u 2021. godini OWASP je izvijestio o nevjerojatnih 274.000 slučajeva ranjivosti SQLi (OWASP, 2021). Iste godine u trećem mjesecu, dogodio se značajan i široko publiciran sigurnosni proboj kada je grupa WikiLeaks-style napala društvenu mrežu GAB. Uspjeli su ukrasti značajnih 70 gigabajta podataka. Ovi kompromitirani podaci uključivali su lozinke i privatne objave, što je izazvalo ozbiljnu zabrinutost za privatnost i

sigurnost korisnika. Uz to, osnivač društvene medijske platforme Gab otkrio je da je ova krađa podataka uključivala i privatni račun bivšeg predsjednika Donalda Trumpa. Smatra se da je napad omogućila nepažnja, jer platforma nije izbrisala tokene za prijavu pohranjene u preglednicima i mobilnim aplikacijama (Goodin, 2021). Na slici 3 SQL injekcija je prikazana kao glavni izvor kritičnih ranjivosti web-aplikacija pronađenih u 2022. godini s 33%, a slijedi ga napad skriptiranja na više stranica (engl. *Cross-site scripting (stored)*) s 26.7% (Vailshery, 2023).

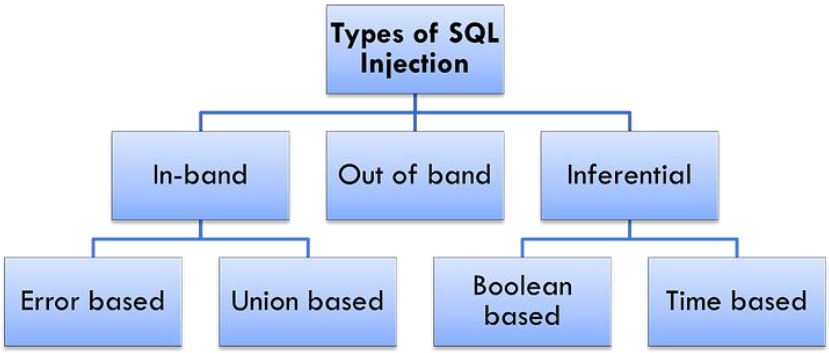


Slika 3 Globalna distribucija ranjivosti web aplikacija u 2022. (Vailshery, 2023)

Suvremene organizacije i dalje se suočavaju sa značajnim rizicima koji su rezultat zastarjelih sustava i nedovoljnih sigurnosnih mjera. Među najranjivijim sektorima, obrazovne institucije čine 35% ranjivosti, a odmah ih slijede vladine organizacije koje čine 32%. Dok 21% ranjivosti prosječno utječe na ostale industrije (VentureBeat, 2022). Zbog svih ovih statistika nije iznenađujuće da su Ministarstvo domovinske sigurnosti Sjedinjenih Država, organizacija Mitre i institucija SANS nazvali SQL injekcijom jednom od najopasnijih sigurnosnih ranjivosti (Team, I.S., 2013).

4. Najčešće vrste SQL injekcije

Identificiranje i klasificiranje svih oblika injekcija vezano za SQL napade može biti dosta izazovno. Slični ili isti napadi mogu imati različite nazive ovisno o specifičnim svojstvima sustava. Postoje različite tehnike napada, uglavnom vođene ciljem napadača. U odnosu na to, najčešća podjela je na temelju metoda koje se koriste za pristup pozadinskim podacima i potencijalnu štetu koja se može uzrokovati. Takva podjela se općenito dijeli u sljedeće kategorije: In-band SQLi (klasični), inferencijalni (engl. inferential) SQLi ili slijepi (eng. blind) i out of band SQLi. Kategorije su prikazane slikovito na slici 3.



Slika 4 Vrste SQL injekcije (Çelik, 2021).

4.1. In-band SQLi

In-band SQLi odnosi se na vrstu napada kada napadač koristi isti komunikacijski kanal za ubacivanje zlonamjernog SQL koda i dohvaćanje rezultata. To znači da napadač može izravno promatrati učinak ubačenog koda i prikupiti informacije iz baze podataka putem iste veze koju je koristio za ubacivanje koda. Na primjer, ako napadač izvede napad koristeći web preglednik, rezultat napada bit će prikazan u istom web pregledniku. In-band se još naziva i klasičnom SQLi jer se često smatra najjednostavnijim i najčešće korištenim (Invicti, 2022). Postoje dvije podvrste in-band SQLi, jedan od njih je temeljen na pogreškama (engl. *error-based*), a drugi je temeljen na operatoru za uniju (engl. *union*).

4.1.1. Error-based SQLi

Ova podvrsta SQL napada koristi se za slanje neočekivanog unosa ili logički neispravnih upita za generiranje nekih pogrešaka, kako bi se na temelju tih grešaka pojedinac opskrbio dodatnim informacijama da pojača svoju injekciju (Rai i sur., 2021). Kada prosječni korisnici naiđu na ovakve zbunjujuće pogreške aplikacija, često ih ignoriraju i ne razmišljaju puno o tome. Problem je što u pravim rukama ovakve informacije mogu otkriti jako puno o tome kako funkcionira sustav. Pogreške i informacije prikazane u njima uvelike pomažu zlonamjernoj strani odabrati način napada i kakvu vrstu upita da pošalje u bazu podataka. Praćenjem te komunikacije između aplikacije i baze podataka gleda se ponašanje i u slučaju greške, napadač ima potrebno znanje da bi dalje naštetio aplikaciji (Bedeković i sur., 2022). Vraćanjem pogreška napadaču na prvu ruku se može činiti bezopasnim. Međutim, ovisno o strukturi aplikacije i vrsti baze podataka, napadač može upotrijebiti primljeni niz pogreške da dobije informacije o vrsti i verziji baze podataka kako bi mogao koristiti određene tehnike napada za specifičnu verziju. Istodobno, ima se mogućnost vađenja podatka iza baza, iako je proces duži i složeniji od izravnog prikazivanja rezultata upita, napadač može manipulirati pogreškama kako bi izvukao podatke iz baze podataka (Invicti, 2022).

```
https://www.example.com/category.php?id=2'
```

Programski kod 3 Error-based SQLi (Çelik, 2021).

Programski kod 3 prikazuje parametar "id" koji se prosljeđuje stranici putem upisanog URL-a. Prisutnost apostrofa (') nakon parametra je uobičajeni pokazatelj da se radi o zlonamjernom kodu i potencijalnoj ranjivosti SQL injekcije. Ako poslužitelj na ovaj URL odgovori SQL pogreškom, to pokazuje da se poslužitelj povezao s bazom podataka na nesiguran način. Nakon ovog koraka, neke od SQL naredbi mogu se pokrenuti za daljnje petljanje ili uništavanje baze podataka. Iako su pogreške vrlo korisne tijekom razvijanja i poboljšanja web aplikacija, treba ih onemogućiti na aktivnoj web stranici ili ih umjesto toga zabilježiti u datoteku s ograničenim pristupom jer u protivnom bit će podloženi ovakvom napadu.

4.1.2. Union SQLi

Operator unije UNION koristi se za kombiniranje dva ili više rezultata SELECT naredbi. Svaka SELECT izjava mora imati isti broj stupaca, a stupci u svakoj izjavi moraju imati isti redoslijed i trebali bi sadržavati slične tipove podataka. Kada je aplikacija ranjiva i

rezultat upita se prikazuje unutar odgovora aplikacije, operator unije se može koristiti za dohvaćanje podataka iz drugih tablica koje nisu vidljivi u bazi podataka. Ova podvrsta napada se često koristi nakon što su baze podataka već otkrivene, na primjer napad temeljen na pogrešci redovito prethodi napadu temeljenom na uniji. Smatra se najopasnijom vrstom SQLi jer nedobronamjernoj strani omogućuje izravno dobivanje gotovo svih informacija iz baze podataka (Bedeković i sur., 2022).

```
https://example.com/category.php?id=3 'UNION+SELECT+NULL,NULL,NULL—
```

Programski kod 4 Union SQLi (Çelik, 2021).

Cilj injekcije programskog koda 4 je manipulirati SQL upitom i počiniti određenu akciju neovlaštenim skupom stupaca. Neovlašteni skup stupaca je predstavljenih NULL vrijednostima koji napadač želi ubaciti u originalni upit. Izrađeni maliciozni upit može izvesti dodatne radnje, izdvojiti osjetljive informacije ili manipulirati rezultatima.

4.2. Inferential SQLi

Inferencijalna ili slijepa (engl. Inferential or blind) SQLi dobiva takav naziv zbog oblika interakcije s bazom podataka u kojoj se manipuliraju podaci bez izravnog uvida da se radi o zlonamjernoj radnji. Za razliku od in-band SQLi, ovaj napada to ne čini na zahtjev upita ili s bilo kakvim vizualnim znakovima da se nešto događa u bazama podataka. U prethodnim primjerima upiti se postavljaju logičkim iskazima kako bi uvjet bio istina (engl. TRUE) ili laž (engl. FALSE). Za razliku od prethodnih napada ovdje se koristi uvjet koji bi mogao biti istinit i s time bi mogao otkriti neke informacije bez slanja povratnih informacija. Umjesto primanja vidljivih rezultata ili poruka o pogrešci, napadač se oslanja na ponašanje aplikacije koristeći Booleovu logiku kako bi utvrdio je li umetnuti SQL kod generirao istinito ili lažno stanje, a baza različito reagira na istinitu ili lažnu izjavu. Izradom specifičnih upita s logičkim uvjetima, napadač ponavljajući upit izvlači informacije ili izvodi radnje koristeći istinite ili netočne odgovore primljene od ranjive aplikacije (Bedeković i sur., 2022). To omogućuje daljnji uvid može li počinitelj napasti s drugim vrstama injekcija ili će jednostavno prikupiti sve željene podatke. Kod inferencijalnog SQLi isto tako postoje dvije podvrste napada, prvi je baziran na Booleovim operatorima (engl. Boolean-based), a drugi je baziran na vremenu (engl. Time-based).

4.2.1. Boolean-based SQLi

Ova tehnika se temelji na analizi Booleove izjave temeljene na Da/Ne upitima. Prisiljava različite odgovore od aplikacije i ako je traženi upit istina, web stranica će vratiti različite podatke u usporedbi s lažnim upitom (Rai i sur., 2021).

```
http://www.example.com/?id=1' AND (length(database())) = 8 – +
http://example.com/categoryx.php?id=1 OR 17-7=10
```

Programski kod 5 Boolean-based SQLi (Çelik, 2021).

U prvom dijelu upita programskog koda 5, cilj je manipulirati SQL upitom kako bi se provjerila duljinu naziva baze podataka. Ubacivanjem ovog upita se želi utvrditi sastoji li se duljina imena baze podataka od 8 znakova. Ako aplikacija odgovori kako se očekuje, sugerira da je umetnuti upit važeći i da je uvjet istinit. Nakon toga u drugom dijelu upita napadač pokušava izvršiti napad SQL injekcije temeljen na logičkim vrijednostima ubacivanjem pravog uvjeta u upit. Tako se želi zaobići mehanizme provjere autentičnosti ili manipulirati logikom izvornog upita kako bi se izvukle osjetljive informacije.

4.2.2. Time-based SQLi

Još jedan način izvođenja napada inferencijalnim SQLi su vremenski napadi. Postoje određene varijacije, a osnovna ideja iza ove vrste napada je izazvati kašnjenja u izvršavanju SQL upita i promatrati vrijeme odgovora aplikacije. Iskorištavanjem vremenskog kašnjenja, napadač može zaključiti informacije o strukturi i sadržaju baze podataka. Ponekad rezultati zlonamjernog logičkog upita ne čine nikakvu akciju, pa napadač stvara dodatnu razliku u odgovoru koji dobije iz baze podataka. Na primjer, MySQL poslužitelj podržava funkcije poput SLEEP() i BENCHMARK() koji dopušta vremensko kašnjenje (Bedeković i sur., 2022). Injekcija SQL-a temeljena na vremenu može biti prilično tiha i neprimjetna jer obično ne proizvodi nikakve vidljive poruke o pogrešci. Treba proučavati razne obrane kojih je dobro implementirati za zaštitu od ovakve podvrste napada za izbjegavanje neugodnih situacija s vremenskim kašnjenjem. Osim toga, redovite sigurnosne provjere i testiranje prodora mogu pomoći u prepoznavanju i ublažavanju ovakve ranjivosti.

```
SELECT * FROM products WHERE id=1; IF SYSTEM_USER='sa' WAIT FOR DELAY '00:00:15'
SELECT * FROM card WHERE id=1-IF(MID(VERSION(),1,1) = '5', SLEEP(15), 0)
```

Programski kod 6 Tme-based SQli (Çelik, 2021).

Prvi gornji primjer programskog koda 6 je vremenski utemeljena SQL injekcije koja cilja Microsoft SQL server bazu podataka. Napad pokušava prikupiti informacije promatrajući vrijeme odgovora na upit. Ako se primijeti kašnjenje, to bi moglo značiti da je korisnik vlasnik specifičnog računa ‘sa’, otkrivajući vrijedne informacije napadaču kao što su ispitivanje je li korisnik admin sustava ili slično. Kao i prethodni primjer, donji dio upita programskog koda 6, ima za cilj izvući informacije promatrajući vrijeme odazivanja. Ako se primijeti kašnjenje, to sugerira da je izvučeni prvi znak niza verzije poslužitelja baze podataka '5', pružajući potencijalno korisne informacije za napadača. Pomoću takvog upita dolazi se do informacije da je verzija baze podataka jednaka MYSQL 5 (Çelik, 2021).

4.3. Out of band SQLi

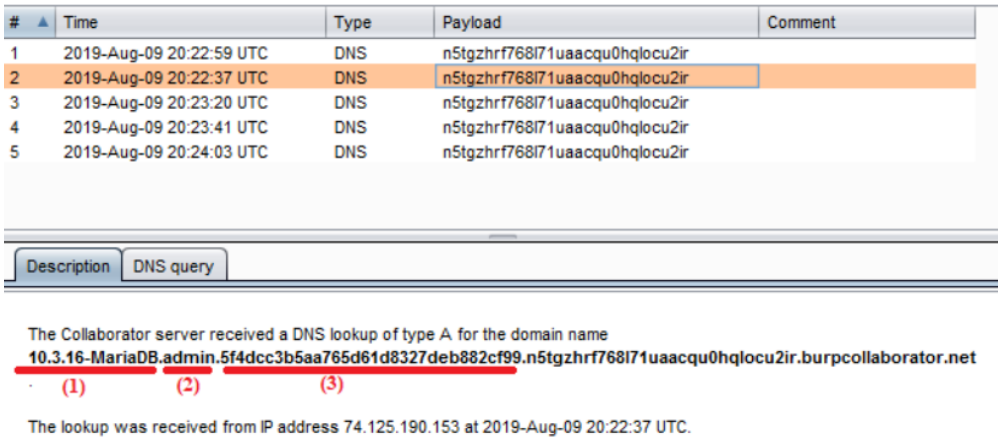
Ranjivost out of band injekcije ili skraćeno OOB SQLi je eksfiltracija podatka iz baze podataka kroz različite izlazne kanale. Ovaj napad karakterizira to što se događa kada napadač ne može koristiti isti kanal za pokretanje napada i prikupljanje rezultata. Tako da tijekom ove injekcije se koristi zajednički kanal za ekstrakciju podataka kroz Domain Name Server (DNS) i HyperText Transfer Protocol (HTTP) kanala. Od ove vrsta SQL injekcije bi se trebalo što prije ispravno obraniti zbog jakih utjecaja koji su u razini s ostalim tradicionalnim metodama. Utječe na sustave baza podataka s nedovoljnom kontrolom kod valjanosti unosa i dopuštenog javnog pristua, bilo DNS ili HTTP protokolu. Moderniji sustavi za upravljanje bazama podatak raspolažu s moćnim aplikacijama, a njihove značajke nadilaze samo jednostavno vraćanje podataka kroz upit. Na primjer, ako trebaju neke informacije koje se nalaze u drugoj bazi podataka trebaju se otvoriti specifične veze kako bi se ti podaci dohvatili. Između ostalog, imaju opciju da pošalju e-poštu kada se dogodi neki određeni događaj i mogućnost komunikacije s datotečnim sustavom. Ove funkcionalnosti se navode jer mogu biti od velike pomoći napadaču, a ponekad se pokažu i najboljim načinima iskorištavanja ranjivosti SQL injekcije, pogotovo kada nije moguće dobiti upit izravnom HTTP komunikacijom (Clarke i sur., 2009).

Tri su faktora uspjeha OOB SQLi. Prvo, sustav baze podataka prihvaća i obrađuje zlonamjerne SQL upite bez odgovarajuće kontrole dezinfekcije na razini web aplikacije. Zatim je sustav baze podataka dopustio komunikaciju na javnoj mreži (DNS ili HTTP protokol). Na kraju, poslužitelj koji se koristi za slušanje je potreban za hvatanje izvučenih informacija iz sustava baze podataka.

```
select
load_file(CONCAT('\\\\',(SELECT+@@version),'.',(SELECT+user),'.',
(SELECT+password),'.','n5tgzhrf768l71uaacqu0hqlocu2ir.burpcollabo
rator.net\\vfw'))
```

Programski kod 7 OOB SQLi (Chun How, 2019).

U upitu napadač pokušava iskoristiti funkciju load_file() za čitanje datoteka iz datotečnog sustava poslužitelja. Umetnuti kod injekcije koristi funkciju CONCAT() za konstrukciju putanje datoteke, a 'n5tgzhrf768l71uaacqu0hqlocu2ir.burpcollaborator.net\\vfw' je naziv domene koji se koristi za out of band SQLi. Predstavlja vanjski poslužitelj kojim upravlja napadač i na koji će se slati izvučene informacije. Svrha je konstruirati put datoteke koji uključuje MySQL verziju, trenutnog korisnika i njegovu lozinku. Izvršavanjem ovog upita, napadač pokušava pročitati datoteku iz datotečnog sustava poslužitelja (*Slika 5*) i poslati dobivene podatke svom kontroliranom vanjskom poslužitelju na daljnju analizu. Primjer demonstrira OOB SQLi za jednog od forka MySQL baze podataka koji se zove MariaDB.



Slika 5 Rezultat OOB SQLi (Chun How, 2019).

Na slici 11 prikazuje se snimljeni DNS zahtjev s verzijom baze podataka, nazivom poslužitelja i trenutnim imenom baze podataka. Točka (.) se koristi kao graničnik za organiziranje prikaza snimljenog zahtjeva. Rezultati koji su otkriveni su (1) snimljena verzija baze podataka, (2) naziv hosta i (3) naziv baze podataka Microsoft SQL (Chun How, 2019).

5. Metode testiranja SQL ranjivosti

SQL napadi mogu biti izuzetno štetni. Kao rezultat toga, važno je poduzeti preventivne mjere zaštite prije nego se nešto zloćudno dogodi. To se postiže prevencijom koju čini skup taktika i postupaka osmišljenih za sprječavanje mogućih napada. Cilj prevencije je pronaći strategiju za izbjegavanje potencijalnih posljedica (Ljubičić, Jakšić and Poščić, 2020). Postoje različiti pristupi prevencije napada SQL injekcijom, uz utvrđivanje obrane jedan od preporučenih pristupa je analiziranje mogućih nedostataka SQL upita, to jest namjerno pronalaženje i testiranje aplikacije kako bi se otkrila moguća ranjivost na SQLi. Imajući to na umu javljaju se određene metode testiranja kao što su testiranje crne kutije (engl. *Black box testing*), testiranje bijele kutije (engl. *White box testing)* ili testiranje sive kutije (engl. *Grey box testing*), ta metoda je ujedno i kompromis između prve dvije metode.

Testiranje crne kutije uključuje testiranje sustava bez prethodnog znanja o njegovom internom radu. Ispitivač tijekom testiranja sustava daje ulaz i promatra izlaz koji se generira. Takav način omogućuje prepoznavanje kako sustav reagira na očekivane ili neočekivane radnje korisnika, njegovo vrijeme odazivanja, probleme s upotrebljivošću i probleme s pouzdanošću. Pozitivna strana ovakvog testiranja je što osobe koji su testeri ne trebaju imati prethodna tehnička znanja, programerske ili slične informatičke vještine Zbog tih pozitivnih strana kod pisanja testnih slučajeva dosta je teško identificirati sve moguće inpute u ograničenom vremenu. Ovu metodu testiranja nije idealno koristiti za velike i komplicirane aplikacije jer potpuna pokrivenost testom nije moguća (Imperva, 2023). Primjer ove metode je upisati SQL upit sa specifičnim znakovima kao što su ' or“, i tražiti greške ili druge anomalije u sustavu Još jedan od mogućih primjera je upisivanje Booleovog upit kao što je OR 1=1 i OR 1=2, pa tražiti razlike u odgovoru aplikacije (Khalil, 2021).

Testiranje bijele kutije podrazumijeva metodu testiranja aplikacije s detaljnim unutarnjim informacijama o njezinom izvornom kodu, arhitekturi i načinu konfiguracije. Pogodnost ove metode je da razotkriva probleme sigurnosnih ranjivosti i probleme s protokom podataka, koje se kod testiranje crne kutije ne može sveobuhvatno ili uopće ispitati. Zajedno s time, omogućuje kontinuirano poboljšanje koda, razvojnih praksi i smanjuje opterećenje komunikacije između testera i programera (Imperva, 2023).

Redoslijed provedbe testiranja je najčešće sljedeći:

- Omogućena prijava na web poslužitelj.
- Omogućena prijava u bazu podataka.
- Mapiranje aplikacije: vidljiva funkcionalnost aplikacije i napredno pretraživanje. svih instanci koje u kodu koji razgovaraju s bazom podataka.
- Pregled koda: pregledavanje putanje koda i ulaznih vrijednosti.
- Testiranje na potencijalnu SQL ranjivost.

Uz mnoge prednosti za metodu testiranje bijele kutije, postoje i određeni nedostatci. Za automatizaciju je potreban veliki napor i skupo je za održavanje za razliku od testiranja crnom kutijom. Dosta je osjetljivo na promjene i nije moguće testirati neku funkcionalnost koja ne postoji u bazi koda. Uspoređivanjem navedenih metoda, vidi se da se testiranje bijele kutije često suprotstavlja testiranju crne kutije, koje uključuje testiranje aplikacije iz korisničke perspektive bez ikakvog znanja o njezinoj implementaciji, ali se i nadopunjuju. Testiranje bijelom kutijom nudi otkrivanje strukturnih problema, skrivenih pogrešaka i problema s određenim komponentama, dok testiranje crnom kutijom provjerava radi li sustav u cjelini prema zadanim očekivanjima (Khalil, 2021).

Testiranje sive kutije je odličan kompromis između dvije metode, jer testiranje bijele kutije pretpostavlja da ispitivač ima potpuno znanje, dok se testiranje crne kutije oslanja na perspektivu korisnika bez uvida u kod. Kompromis sive kutije testira aplikacije i okruženja uz djelomično poznavanje internog rada. Obično se koristi za testiranje prodora, s kraja na kraj (engl. *End-to-end*) načina testiranje i raznih integracije. Koncentrirano je na izvođenju više usmjerenih testova na područjima ili korisničkih putanja za koje se smatra da će najvjerojatnije sadržavati nedostatke. Kombinacijom ove dvije metode se osigurava primijenjeno potrebno znanje o strukturi aplikacije za prepoznavanje ranjivosti i grešaka. Testiranje sive kutije objektivno procjenjuje aplikaciju, otkriva probleme sučelja i pokriva sve aspekte funkcionalnosti aplikacije što je izuzetno korisno za prevenciju mogućih SQL injekcija (Imperva, 2023).

6. Zaštita od SQL injekcije

Za razliku od metoda testiranja, kod obrane je ponekad potrebno koristiti neke druge tehnike i mehanizme koji se ponešto razlikuju od preventivnih tehnika i metoda testiranja. Određeno potencijalno rješenje može biti učinkovito u svojoj trenutnoj primjeni, ali kako se tehnologija razvija, pojavljuju se novi rizici i prepreke. Nadalje, višestruka učinkovita rješenja protiv različitih vrsta napada i sigurnosti moraju se kombinirati, a sustav se mora stalno nadzirati i ažurirati. Programer ili administrator treba zaštititi svoje sustave na različite načine. Poželjno je koristiti različite strategije u sustavima s različitim svrhama, kao što je provjera valjanosti unosa, najmanja privilegija, prilagođena poruka o pogrešci i tako dalje. Iako su ove strategije najučinkovitije sredstvo za sprječavanje i zaštitu od napada ranjivosti SQLi, njihova implementacija ponekad zna biti teška u praksi. Unatoč najvećim pokušajima i trudu, ovi postupci su skloni ljudskim pogreškama i stoga su manje učinkoviti od automatiziranih tehnika, zato što uvijek postoji mogućnost da se ušuljaju neželjene pogreške. Stoga se preporučuju brojni savjeti, pristupi i tehnike koje bi pomogle u borbi protiv SQLi (Ljubičić, Jakšić and Poščić, 2020). OWASP (2021) navodi preporuke i radnje kojih bi se trebalo pridržavati, a razvrstavaju se na glavne i dodatne metode zaštite koje se mogu podijeliti na sljedeći način:

Glavne mjere zaštite:

- Upotreba pripremljenih izjava (s parametriziranim upitima)
- Pohranjene procedure (engl. *Stored procedures*)
- Valjanost unosa
- Izbjegavanje svih korisničkih unosa

Dodatna obrana:

- Provođenje najmanje privilegije (engl. *Least privilege*)
- Izvođenje validacije unosa s liste dopuštenih kao sekundarna obrana

6.1. Pripremljene izjave

Upotreba pripremljenih izjava s vezanim varijablama odnosno parametriziranim upitima je opcija koju bi svi programeri prvo trebali naučiti kako bi kvalitetno raspolagali s upitima u bazi podataka. Jednostavni su za pisanje i lakši za razumijevanje od dinamičkih upita. Parametrizirani upit je vrsta upita koji zahtijeva najmanje jedan parametar, dobar je za obranu jer tjera programere da prvo definiraju sav SQL kod, a zatim da proslijede svaki parametar u

upit. Takav način omogućuje bazi podataka razlikovanje koda u odnosu na podatak, bez obzira na unos korisnika. Ove vrste upita s pripremljenim izjavama osiguravaju da napadač ne može promijeniti namjeru upita, čak i ako umetne SQL naredbe. U jako rijetkim okolnostima, pripremljene izjave mogu naštetiti izvedbi. U toj situaciji je najbolje strogo provjeriti valjanost svih podataka ili kompletno izbjeći sve korisničke unose korištenjem rutine izbjegavanja specifične za konkretnu bazu podataka (OWASP, 2021).

6.2. Pohranjene procedure

Pohranjene procedure nisu uvijek sigurne od SQL injekcije. Međutim, ispravno korištenje konstruiranih podataka pohranjenih procedura imaju isti učinak kao upotreba parametriziranih upita kada se implementiraju na siguran način. Razlika između pripremljenih izjava i pohranjenih procedura je u tome što se SQL upit za pohranjenu proceduru definira i pohranjuje u samoj bazi podataka, a zatim poziva iz aplikacije. Zajedničko im je da imaju istu učinkovitost u sprječavanju SQLi, tako da organizacija ili ovlašteni pojedinac treba odabrati koji pristup ima najviše smisla za njihovu situaciju. Pojam implementirana sigurnost (engl. *Implemented safely*) znači da pohranjena procedura ne uključuje nikakvo nesigurno dinamičko generiranje SQL koda i s time se uobičajeno ne radi u pohranjenim procedurama. To se može učiniti, ali treba izbjegavati. Ako se ne može izbjeći onda pohranjena procedura mora koristiti provjeru valjanosti unosa ili slične prikladne korake kako bi osiguralo da se sav korisnički unos koji je spremljen u proceduru ne može biti korišten za injekciju prema SQL dinamičkom upitu (OWASP, 2021).

6.3. Valjanost unosa

Provjera valjanosti unosa je postupak testiranja usklađenosti unosa primljenih od strane aplikacije prema standardu definiranom unutar aplikacije. To može biti jednostavno kao striktno upisivanje parametara, a ima i složenijih opcija poput korištenja regularnih izraza ili poslovne logike za provjeru valjanosti unosa. Ako se izvede dobro jedna je od najmoćnijih kontrola koja se koristi. Postoje dvije različite vrste pristupa provjeri valjanosti unosa. Prva je provjera valjanosti bijele liste (engl. *Whitelisting*) koja se naziva i pozitivna provjera valjanosti, a druga je provjera valjanosti crne liste (engl. *Blacklisting*) ili negativna provjera valjanosti. Kada se provodi provjera valjanosti unosa, trebalo bi se osigurati da je unos u svom kanonskom odnosno najjednostavnijem obliku, prije nego dođe do bilo kakve provjere valjanosti unosa. To može uključivati dekodiranje ulaza u jednostavniji format ili samo odbijanje unosa koji još nisu u najjednostavnijem formatu gdje se nekanonski ne očekuje (Clarke i sur., 2009).

6.3.1. Valjanost crne i bijele liste

Kod provjere valjanosti crne liste, specifični, poznati zlonamjerni znakovi se uklanjaju ili se skroz zamjenjuju u korisničkom unosu. Iako se ovaj pristup često primjenjuje, uglavnom zbog jednostavnosti kojom se može postići, nije učinkovit u usporedbi s provjerom valjanosti bijele liste. Rad s crnom listom nekada neispravno odradi taktike složenog zamagljivanja, što bi moglo omogućiti napadaču da pokvari filtre i potencijalno ubaci SQL injekciju. Ova greška se često događa kao rezultat razvijanja tehnika napada i filtera koji nisu sveobuhvatni ili pravilno implementirani. Alternativno, provjera valjanosti bijele liste ispituje svaki dio korisničkog unosa u odnosu na popis dopuštenih znakova. Ovaj je pristup učinkovitiji u ublažavanju rizika od SQLi jer je restriktivniji što se tiče dopuštenih vrsta unosa. Dobro implementirana valjanost bijele liste bi trebala ispitati svaki dio podataka koje je dostavio korisnik u odnosu na očekivani format podataka (Cisco.com, 2016).

Primjer gdje se koristi valjanost bijele liste je potvrda za ulaznu vrijednost broja kreditne kartice. U takvoj situaciji uključivat će se provjera da unos vrijednosti sadrži samo brojeve, provjera da je dužina između 13 i 16 znamenki i prolazak provjere ispravnosti poslovne logike Luhnove formule (formula za izračunavanje valjanosti broja temeljena na posljednjoj kontrolnoj znamenki kartice). Općenito, provjera valjanosti bijele liste je moćnija od dvije navedene i preporučuje se uvijek koristiti. Međutim, to može biti teško implementirati u situacijama gdje postoje jako složeni unosi ili gdje se cijeli skup mogućih unosa ne može lako odrediti. Zahtjevni primjeri mogu uključivati aplikacije koje su lokalizirane na jezike s velikim skupovima znakova (npr. japanski ili kineski). Kada se pojave takvi primjeri gdje se može koristiti bijela lista, crna lista može pružiti korisnu djelomičnu kontrolu. Ako dođe do toga preporučuje se stavljanje na crnu listu u kombinaciji sa zaštitnim kodiranjem kako bi se osiguralo da je ulaz prošao negdje drugdje (npr. u bazu podataka) radi dodatne provjere kako bi se osiguralo da se njime ispravno raspolaže i da se spriječi moguća SQL injekcija (Clarke i sur., 2009).

6.4. Izbjegavanje svih korisničkih unosa

Zadnju glavnu mjeru zaštite treba koristiti samo kao posljednje sredstvo i u iznimnim situacijama, kada ništa od navedenog nije izvedivo. Svaki sustav za upravljanje baza podataka podržava jednu ili više shema za izbjegavanje određenih znakova kod vrsta upita. Mjera zaštite funkcionira tako da kada se izbjegne sav upisan korisnički unos korištenjem pravilnih uputa,

sustav neće pomiješati taj unos sa SQL kodom koji je napisao programer. Takvim načinom se izbjegavaju određene potencijalne ranjivosti SQLi. Validacija unosa je vjerojatno bolji izbor jer je ova metodologija slaba u usporedbi s drugim obranama i teško je jamčiti da će spriječiti sve SQL injekcije u svim situacijama. Preporuka je uglavnom naknadna ugradnja već prije postavljenog koda ako implementacija provjere valjanosti unosa nije isplativa (OWASP, 2021).

6.5. Dodatna obrana

Pored usvajanja navedenih mjere zaštite, preporučuje se i uvođenje dodatnih obrana kako bi se pružio maksimalni sloj zaštite. Najpoznatija dva pristupa dodatne obrane su najmanja privilegija (engl. *Least privilege*) i validacija unosa s popisa dopuštenih (engl. *Allow-list Input Validation*). Najmanja privilegija služi kako bi se smanjile privilegije dodijeljene svakom računu baze podataka u okruženju, tako da nema dodjeljivanja administratorskih prava računima aplikacije. Razumna je lakoća ovog pristupa i da zvuči jednostavno, ali kada se to radi na ovaj način postaje vrlo nezgodno za normalnu funkciju sustava baza podataka. Trebala bi se pronaći ravnoteža kada se provodi sigurnosnu strategija s najmanjim privilegijama. Što se tiče validacije unosa, osim što je primarna obrana kada ništa drugo nije moguće, provjera valjanosti unosa s popisa dopuštenih može biti i sekundarna obrana koja se koristi za otkrivanje neovlaštenog unosa prije nego što se SQL upitu dalje proslijedi (OWASP, 2021).

7. Zaključak

Nažalost kibernetički napadi se nastavljaju razvijati s tehnološkim razvojem i vrlo vjerojatno će uvijek biti prisutni. Među njima, SQL injekcija ostaje stalna i značajna prijetnja sigurnosti web aplikacija i osjetljivih podataka kojima raspolažu. Posljedice uspješnih napada SQL injekcijom mogu biti razorne, rezultirajući financijskim gubicima, štetom po ugledu, pravnim problemima i gubitkom povjerenja u online platforme. Ozbiljnost i utjecaj ovih napada naglašava potrebu da organizacije poduzmu proaktivne mjere kako bi zaštitile svoje web aplikacije i baze podataka od takvih ranjivosti. Za smanjivanje rizika ključna je implementacija višeslojnog pristupa sigurnosti. To uključuje usvajanje sigurnosnih praksi koje su se navodile kroz ovaj rad. Preporučuje se poticanje kulture svijesti o sigurnosti i obrazovanja unutar organizacija. Važna je obuka programera i administratora o praksama sigurnog kodiranja, rizicima SQL-a i razumijevanje važnosti redovitih sigurnosnih ažuriranja može značajno poboljšati cjelokupnu zaštitu. Za organizacije je ključno da budu oprezne i informirane o prijetnjama u nastajanju i najboljim praksama te da u skladu s tim prilagode svoje sigurnosne mjere.

Zaključno, rješavanje SQL injekcije je kolektivna odgovornost. Razumijevanje rizika, usvajanje snažnih sigurnosnih mjera i poticanjem proaktivnog sigurnosnog načina razmišljanja, organizacije mogu značajno smanjiti vjerojatnost i učinak napada SQL injekcijom. Tako mogu bolje zaštititi svoje vrijedne podatke, održati povjerenje svojih korisnika i osigurati sigurnije digitalno okruženje. Stalnim naporima, suradnjom i predanošću prema sigurnosti, možemo se nadati budućnosti u kojoj kibernetički napadi postaju zastarjeli, a web aplikacije dovoljno jake da pobjede takve prijetnje SQL injekcije.

8. Literatura

1. Andress, J. and Winterfeld, S. (2011). Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners. Waltham, MA: Elsevier. [online] Dostupno na: https://books.google.hr/books/about/Introduction_to_Cyber_Warfare.html?id=McWHysu1eWkC&redir_esc=y [Pristupljeno: 30.06.2023].

2. Bedeković, N., Havaš, L., Horvat, T. and Crčić, D. (2022). The Importance of Developing Preventive Techniques for SQL Injection Attacks. Tehnički glasnik, 16(4), pp.523–529. doi: https://doi.org/10.31803/tg-20211203090618 [Pristupljeno: 08.07.2023].

3. Çelik, Irem. (2021). The Ultimate Guide to SQL Injection. [online] PurpleBox. Dostupno na: https://medium.com/purplebox/sql-injection-da949c39dbe6 [Pristupljeno: 08.07.2023].

4. CIS (2011). Centar Informacijske Sigurnosti · Napadi umetanjem SQL koda. Dostupno na: https://www.cis.hr/files/dokumenti/CIS-DOC-2011-09-025.pdf [Pristupljeno: 11.03. 2023].

5. Cisco.com (2008). Cyber Attack - What Are Common Cyberthreats? [online] Cisco. Dostupno na: https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html [Pristupljeno: 10.06.2023.]

6. Cisco.com. (2016). Understanding SQL Injection. [online] Dostupno na: https://sec.cloudapps.cisco.com/security/center/resources/sql_injection.html [Pristupljeno: 10.06. 2023].

7. Čitaković, Selma. (2023). 10 SQL Injection Attacks Statistics To Know in 2023. [online] Security Escape. Dostupno na: https://securityescape.com/sql-injection-attacks-statistics/ [Pristupljeno: 21.07.2023].

8. Cohen, F., Phillips, C., Painton Swiler, L., Gaylor, T., Leary, P., Rupley, F., & Isler, R. (1998). A Cause and Effect Model of Attacks on Information Systems: Some Analysis Based on That Model, and The Application of That Model for Cyber Warfare in CID. Computers & Security, 17(3): 211-221. [online] Dostupno na: http://dx.doi.org/10.1016/S0167-4048(98)80312-X [Pristupljeno: 01.07.2023].

9. Cyberspace Policy Review (2009). Assuring a Trusted and Resilient Information and Communications Infrastructure. Washington, DC: Executive Office of the President of the United States. [online] Dostupno na: http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final.pdf [Pristupljeno: 17.06.2023.]

10. Danchev, Dancho. (2008). Tracking down the Storm Worm malware. [online] Dostupno na: https://www.zdnet.com/article/tracking-down-the-storm-worm-malware/ [Pristupljeno: 21.07.2023].

11. Dharam, R. and Shiva, S.G. (2012). Runtime monitors for tautology based SQL injection attacks. Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec). doi: https://doi.org/10.1109/cybersec.2012.6246104 [Pristupljeno: 07.07.2023].

12. Dictionary.cambridge.org. (n.d.). HACKER | meaning in the Cambridge English Dictionary. [online] Dostupno na: https://dictionary.cambridge.org/dictionary/english/hacker [Pristupljeno: 01.07.2023].

13. Džanko, I., Dodig, I., Cafuta, D., Kovačević, R. and Kramberger, T. (2017). ANALIZA NAPADA USKRAĆIVANJEM USLUGE U STVARNOM OKRUŽENJU. Polytechnic and design, 5(1), pp.51–60. [online] Dostupno na: https://hrcak.srce.hr/194802 [Pristupljeno: 05.07.2023].

14. Erickson, Jon. (2003). Hacking: The Art of Exploitation. [online] Google Books. No Starch Press. [online] Dostupno na: https://books.google.hr/books/about/Hacking.html?id=P8ijosP6ti4C&redir_esc=y [Pristupljeno: 03.07. 2023].

15. Gandhi, R., Sharma, A., Mahoney, W., Sousan, W., Zhu, Q., and Laplante, P. (2011). Dimensions of Cyber-Attacks: Cultural, Social, Economic, and Political. IEEE Technology and Society Magazine, 30(1): 28-38. [online] Dostupno na: http://dx.doi.org/10.1109/MTS.2011.940293 [Pristupljeno: 01.07.2023]

16. Gandotra, E., Bansal, D. and Sofat, S. (2014). Malware Analysis and Classification: A Survey. Journal of Information Security, 05(02), pp.56–64. doi: https://doi.org/10.4236/jis.2014.52006 [Pristupljeno: 03.07.2023].

17. Goodin, Dan. (2021). Trump's is one of 15,000 Gab accounts that just got hacked. [online] Ars Technica. Dostupno na: https://arstechnica.com/information-technology/2021/03/gab-the-far-right-website-has-been-hacked-and-70gb-of-data-leaked/ [Pristupljeno: 22.07.2023].

18. Han, C. and Dongre, R. (2014). Technology Innovation Management Review 40 Q&A. [online] Dostupno na: https://www.timreview.ca/sites/default/files/article_PDF/HanDongre_TIMReview_October2014.pdf [Pristupljeno 17.06. 2023].

19. Hyslip, T. and Horner, M. (2017). SQL Injection: The Longest Running Sequel in Programming History. The Journal of Digital Forensics, Security and Law. Dostupno na:https://www.researchgate.net/publication/324227697_SQL_Injection_The_Longest_Running_Sequel_in_Programming_History [Pristupljeno: 21.07.2023].

20. Imperva (2019). What is MITM (Man in the Middle) Attack | Imperva. [online] Learning Center. Dosupno na: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/ [Pristupljeno 03.07.2023].

21. Imperva (2019). What is SQL Injection | SQLI Attack Example & Prevention Methods | Imperva. [online] Imperva. Dostupno na: https://www.imperva.com/learn/application-security/sql-injection-sqli/ [Pristupljeno: 07.07.2023].

22. Imperva (2023). What Is White Box Testing | Types & Techniques for Code Coverage | Imperva. [online] Dostupno na: https://www.imperva.com/learn/application-security/white-box-testing/ [Pristupljeno: 12.07.2023].

23. Imperva. (2019). What is a Zero-Day Exploit | Protecting Against 0day Vulnerabilities | Imperva. [online] Dostupno na: https://www.imperva.com/learn/application-security/zero-day-exploit/ [Pristupljeno 03.07.2023].

24. InfoSec Insights. (2020). What Is SQL Injection? 8 Tips on How to Prevent SQL Injection Attacks. [online] Dostupno na: https://sectigostore.com/blog/what-is-sql-injection-8-tips-on-how-to-prevent-sql-injection-attacks/ [Pristupljeno: 07.07.2023].

25. Invicti. (2022). In-Band SQL Injection | Learn AppSec. [online] Dostupno na: https://www.invicti.com/learn/in-band-sql-injection/ [Pristupljeno: 08.07.2023].

26. Invicti. (2022). Top 5 most dangerous injection attacks. [online] Dostupno na: https://www.invicti.com/blog/web-security/top-dangerous-injection-attacks/ [Pristupljeno: 08.07.2023].

27. Jahankhani, Hamid, Al-Nemrat, A. and Hosseinian-Far, Amin. (2014). Cyber crime Classification and Characteristics. (str. 149-164).[online] Dostupno na: https://www.researchgate.net/publication/280488873_Cyber_crime_Classification_and_Characteristics [Pristupljeno 03.06.2023].

28. Justin Clarke and Lead, J., Rodrigo, T., Alvarez, M., Hartley, D., Hemler, J., Kornbrust, A., Meer, H., O'leary-Steele, G., Revelli, A., Slaviero, M. and Stuttard, D. (2009). SQL Injection Attacks and Defense. [online] Dostupno na: https://doc.lagout.org/security/SQL%20Injection%20Attacks%20and%20Defense.pdf [Pristupljeno: 11.03.2023].

29. Khalil, Rana. (2021). Web-Security-Academy-Series/sql-injection/theory/SQL Injection Complete Guide.pdf at main · rkhal101/Web-Security-Academy-Series. [online] GitHub. Dostupno na: https://github.com/rkhal101/Web-Security-Academy-Series/blob/main/sql-injection/theory/SQL%20Injection%20Complete%20Guide.pdf [Pristupljeno: 11.07.2023].

30. Lee Chun How (2019). A Study of Out-of-Band Structured Query Language Injection. [online] Dostupno na: https://zenodo.org/record/3556347/files/A%20Study%20of%20Out-of-Band%20SQL%20Injection.pdf?download=1 [Pristupljeno: 11.07.2023].

31. Ljubičić, N., Jakšić, D. and Poščić, P. (2020). Napadi ubacivanjem SQL izraza – prevencija i obrana. Zbornik Veleučilišta u Rijeci, 8(1), pp.313–330. [online] Dostupno na: https://hrcak.srce.hr/clanak/348577 [Pristupljeno: 11.03.2023].

32. Moes, Tibor. (2023). SQL Injection Examples (2023): The 6 Worst Attacks Ever. [online] Dostupno na: https://softwarelab.org/blog/sql-injection-examples/ [Pristupljeno: 21.07.2023].

33. OWASP (2013). SQL Injection. [online] OWASP. Dostupno na: https://owasp.org/www-community/attacks/SQL_Injection [Pristupljeno: 08.07.2023].

34. OWASP (2021). OWASP Top Ten. [online] Owasp.org. Available at: https://owasp.org/www-project-top-ten/ [Pristupljeno: 08.07.2023].

35. OWASP (2021). SQL Injection Prevention · OWASP Cheat Sheet Series. [online] Owasp.org. Dostupno na: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html. [Pristupljeno: 12.07.2023].

36. Paloalto Networks (2019). What is a denial of service attack (DoS) ? - Palo Alto Networks. [online] Paloaltonetworks.com. Dostupno na: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos [Pristupljeno: 03.07.2023].

37. Rai, A., Miraz, MD.M.I., Das, D., Kaur, H. and Swati (2021). SQL Injection: Classification and Prevention. [online] IEEE Xplore. doi: https://doi.org/10.1109/ICIEM51511.2021.9445347 [Pristupljeno 08.07.2023].

38. Roscini, M. and Trust, L. (2014). Cyber Operations and the Use of Force in International Law. [online] Google Books. OUP Oxford. Dostupno na: https://books.google.hr/books?hl=hr&lr=&id=yeokAwAAQBAJ&oi=fnd&pg=PP1&dq=Cyber+Operations+and+the+Use+of+Force+in+International+Law&ots=Xoof-EUlQe&sig=eterIfmfOcxrdA8tpsYQ-p8hCDI&redir_esc=y#v=onepage&q=Cyber%20Operations%20and%20the%20Use%20of%20Force%20in%20International%20Law&f=false [Pristupljeno: 03. 06. 2023].

39. Sadeghian, A., Zamani, M. and Abdullah, S.M. (2013). A Taxonomy of SQL Injection Attacks. [online] IEEE Xplore. doi: https://doi.org/10.1109/ICICM.2013.53 [Pristupljeno 07.07.2023].

40. Shakarian, P., Shakarian, J. and Ruef, A. (2013). Introduction to CyberWarfare: A Multidisciplinary Approach. Waltham, MA: Elsevier. [online] Dostupno na: https://books.google.hr/books/about/Introduction_to_Cyber_Warfare.html?id=McWHysu1eWkC&redir_esc=y [Pristupljeno: 30.06.2023].

41. Shruti, Mohan. (2023). 10 Types of Cyber Attacks You Should Be Aware. [online] Simplilearn.com. www.simplilearn.com/tutorials/cyber-security-tutorial/types-of-cyber-attacks [Pristupljeno 03.06.2023].

42. Sujay Vailshery, Lionel. (2023). Most common web application critical risks 2020. [online] Dostupno na: https://www.statista.com/statistics/806081/worldwide-application-vulnerability-taxonomy/ [Pristupljeno: 22.07.2023].

43. Team, I.S. (2013). 14 Years of SQL Injection and still the most dangerous vulnerability. [online] www.invicti.com. Dostupno na: https://www.invicti.com/blog/web-security/sql-injection-vulnerability-history/ [Pristupljeno: 21.07.2023].

44. Ttu.edu. (2017). Scams – Spam, Phishing, Spoofing and Pharming | Be in Charge of Your Digital Life | Cybersecurity Awareness Program: Lubbock | TTU. [online] Dostupno na: https://www.ttu.edu/cybersecurity/lubbock/digital-life/digital-identity/scams-spam-phishing-spoofing-pharming.php [Pristupljeno: 03.07.2023].

45. VentureBeat. (2022). Report: 35% of educational institutions have a SQLi vulnerability. [online] VentureBeat. Dostupno na: https://venturebeat.com/security/report-35-of-educational-institutions-have-a-sqli-vulnerability/ [Pristupljeno: 22.07.2023].

Popis slika

Popis programskih kodova

SQL injekcija kao vrsta kibernetičkog napada

Sažetak

U ovome radu će se objasniti SQL injekcija kao jedna od vrsta kibernetičkog napada, kao i opasnost koju predstavlja svojim djelovanjem. Opisat će se štete koje SQL injekcija može uzrokovati i kakve vrste napada se najčešće koriste. Odredit će se kako uopće dolazi do akcije koja omogućuje zloćudni čin i koje naredbe se upotrebljavaju kako bi se počinila šteta. Analizirat će se nedostatci SQL upita i utvrdit će se razlog zbog kojeg je moguće pogoditi strukturu SQL upita ovakvim napadom. Opisat će se neki od procesa kojima se izvodi izvršavanje napada i pokušaji i pogreške koji nastaju iskorištavanjem ovakve ranjivosti. Naznačit će se kako se može smanjiti mogućnost SQL injekcije i kojih bi se preporuka bilo dobro pridržavati. Prikaz SQL injekcije će se nadopuniti s nekoliko poznatih primjera kako bi se bolje utvrdila potencijalna opasnost ovakve vrste kibernetičkog kriminala.

Ključne riječi: SQL injekcija, vrste SQL injekcije, kibernetički napad, ranjivost web aplikacija

SQL injection as a type of cyberattack

Summary

In this paper, SQL injection will be explained as one of the types of cyberattack as well as the danger it represents through its action. The damage that SQL injection can cause will be described, as will the types of attacks that are most often used. It will be determined how the action that enables the malicious act occurs in the first place and what commands are used to commit the harm. The shortcomings of SQL queries will be analyzed, and it will be determined why it is possible to hit the structure of SQL queries with this kind of attack. This paper will describe some of the processes used to execute the attack and the trial and error that occurs when exploiting this vulnerability. It will be indicated how the possibility of SQL injection can be reduced and which recommendations should be followed. The representation of SQL injection will be complemented with several well-known examples to better determine the potential danger of this type of cybercrime.

Key words: SQL injection, types of SQL injection, cyberattack, web application vulnerability