Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Zaštita računarskih sistema i mreža

- Jedanaesta laboratorijska vežba -

Opis aplikacije

U svim vežbama aplikacija ima istu strukturu i sastoji se iz korisničke aplikacije kojoj studenti pristupaju iz pretraživača i serverske aplikacije pokrenutoj na virtuelnoj mašini kojoj studenti nemaju pristup. Korisnički interfejs i serverska aplikacija komuniciraju preko *HTTP* protokola, dok serverska aplikacija komunicira sa bazom podataka koristeći odgovarajući *JDBC* drajver. Baza koja se koristi je *in-memory h2* baza podataka koja je pokrenuta zajedno sa aplikacijom. Studentima se preporučuje da detaljno prouče SQL sintaksu *h2* baze podataka. Korisničkoj aplikaciji može da se pristupi iz pretraživača računara na kojem je pokrenut *VPN* te nije potrebno udaljeno povezivanje na virtuelnu mašinu. Svi zahtevi poslati sa korisničke aplikacije se upućuju serverskoj aplikaciji i kao odgovor se vraća nova ili osvežava trenutna stranica na korisničkoj aplikaciji. Cilj svih vežbi jeste pronaći lozinke svih registrovanih korisnika.

Napomene:

- U slučaju da stranica po pristupu ne izgleda kao na slici, osvežiti stranicu bez korišćenja keša. (u Google Chrome-u pritisnuti CTRL + SHIFT + R da bi se stranica osvežila bez korišćenja keširanja starih podataka, a potom ponovo kliknuti refresh - F5 - po potrebi).
- Neki karakteri u Word-u, kao što su jednostruki navodnici, ne odgovaraju jednostrukim navodnicima u SQL sintaksi te se studentima preporučuje da ne vrše prost copy/paste upita iz dokumenta u pretraživač, već da prekucavaju kod ili da prvo izvrše copy/paste u jednostavan tekstualni editor (npr. Notepad) pa odatle ponovo kopiraju upit u pretraživač.

Zahvaljujemo se bivšem studentu master inž. Đorđu Madiću koji je pomogao u izradi aplikacija korišćenih za ovu laboratorijsku vežbu, kao i asistentu master inž. Adrianu Milakoviću koji je pomogao u pripremi laboratorijske vežbe!

Vežba 1 – Union-based SQL injection

Ovaj tip napada je pravolinijski, brzo se uči i na skoro identičan način se izvršava za sve baze podataka. Napadač kroz male iteracije otkriva sve više podataka iz baze koje mu se prikazuju na stranici.

Vežba 1.1

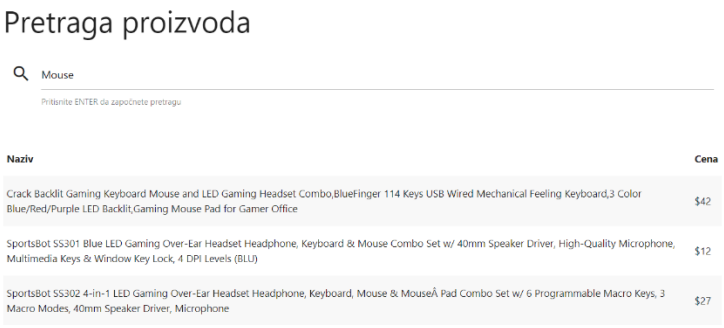
Aplikacija na kojoj se ova vežba zasniva je platforma za gledanje filmova i serija. Korisnici mogu da se registruju na platformu kako bi imali mogućnost pristupa filmovima i serijama. Svi registrovani korisnici nalaze se u bazi podataka. Korisnik može da se uloguje na platormu i tada mu se prikazuje stranica kao na slici, nakon čega može da pokrene zabavu i pristupi stranici sa multimedijalnim sadržajem. Vežba počinje na strani dobrodošlice prijavljenog korisnika. Stranici se pristupa iz pretraživača na `http://192.168.x.1:8091/?id=1`. Ovo je ujedno i jedina stranica platforme kojoj se može pristupiti. Klik na dugme *Pokreni zabavu* ne proizvodi nikakvu akciju. Identifikator prijavljenog korisnika, *id*, postavlja se kao URL parametar. Vrednost parametra služi da se iz baze podataka pročita korisničko ime, koje je vezano za odgovarajući *id*, i prikaže na stranici, tako da se napad zasniva na manipulaciji ovog parametra. Cilj vežbe je da se samo na osnovu ove stranice dođe do detaljnih podataka iz baze.

1. Promeniti vrednost *id* parametra i posmatrati ponašanje stranice.
   (npr. `http:// 192.168.x.1:8091/?id=2`)
   - Posle kog *id*-ja se ne prikazuju imena korisnika? Da li smo sigurni da broj korisnika odgovara poslednjem unetom *id*-ju?
   - Koja su korisnička imena pretraženih korisnika?
2. Pronaći ukupan broj kolona koji se vraća kao rezultat upita unoseći sledeće vrednosti za *id* parametar:
   ```
   -1 union select 1
   -1 union select 1,2
   -1 union select 1,2,3
   -1 union select 1,2,3,4
   ```
   - Objasniti kako funkcioniše upit kojem se prosleđuje sledeća vrednost za *id*.
   - Zašto smo za vrednost *id* parametra izabrali `-1`? Da li smo mogli da izaberemo drugu vrednost?
   - Koliko kolona ima tabela?
   - Koja po redu kolona u tabeli se uzima pri prikazu korisničkog imena na stranici?
3. Pronaći ime tabele koju koristi stranica. Uneti sledeću vrednost za *id* parametar i posmatrati stranicu:
   ```
   -1 union select 1,table_name,3 from information_schema.tables where
   table_schema=schema()
   ```
   - Čemu odgovara vrednost ispisana na stranici?
   - Koristeći sličnu ideju, pronaći ime baze koju koristi stranica.
4. Pronaći imena svih kolona tabele koju koristi stranica. Uneti sledeću vrednost za *id* parametar:
   ```
   -1 union select 1,column_name,3 from information_schema.columns where
   table_name='***'
   ```
   gde je `***` ime tabele koju koristi stranica.
   - Čemu odgovara vrednost ispisana na stranici?
   - Koristeći `limit i offset` pronaći imena svih kolona u tabeli.
     (pomoć: dodati `limit 1 offset 0` na kraj gorenavedenog dela URL-a)
5. Koristeći imena kolona dobijenih u prethodnoj tački, pronaći lozinke svih registrovanih korisnika. (pomoć: koristiti limit/offset ili dodati WHERE deo u SQL upit koji služi da selektuje tačno određenog korisnika)

Vežba 1.2

Napomena: U slučaju da stranica po pristupu ne izgleda kao na slici, osvežiti stranicu bez korišćenja keša. (u Google Chrome-u pritisnuti CTRL + SHIFT + R da bi se stranica osvežila bez korišćenja keširanja starih podataka, a potom ponovo kliknuti refresh - F5 - po potrebi)

Aplikacija na kojoj se ova vežba zasniva je veb prodavnica kompjuterske opreme. Vežba počinje na stranici za pretragu proizvoda kao na slici levo. Stranici se pristupa iz pretraživača na `http://192.168.x.1:8092`. Ovo je ujedno i jedina stranica platforme. Pretraga se vrši po nazivu, gde je moguće uneti i samo deo naziva proizvoda. U slučaju pretrage bez unetog naziva prikazuju se svi proizvodi iz prodavnice. Nakon pritiska tastera *ENTER,* rezultat pretrage prikazuje se u vidu tabele, sa kolonama za naziv i cenu proizvoda, kao na slici desno.



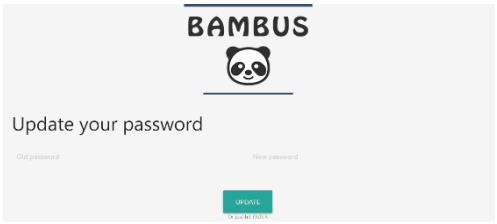
1. Polje za pretragu se koristi da se iz baze pronađu svi proizvodi u čijem nazivu postoji uneta reč iz polja za pretragu, tako da se napad izvršava kroz polje za pretragu. Uneti sledeće vrednosti u polje za pretragu i posmatrati ponašanje stranice:
    - `Mouse`
    - `x`
    - `'`
    - `'--` (`--` je komentar u SQL sintaksi *h2* baze podataka)
    - `' and false--`
    - `' and true--`

    Da li može da se pretpostavi kako izgleda SQL upit?
2. Uneti sledeću vrednost u polje za pretragu:
   `' and false union select 1,table_name from information_schema.tables where table_schema=schema() --`
    - Šta predstavljaju podaci u rezultatima pretrage?
    - Koristeći sličnu ideju, pronaći ime baze koju koristi stranica.
3. Pronaći imena svih kolona svih tabela koje koristi stranica. Uneti sledeći naziv u polje za pretragu:
   `' and false union select 1,column_name from information_schema.columns where table_name='***' --`
   gde je *** ime tabele koju koristi stranica.
4. Koristeći imena kolona i tabela dobijenih u prethodnim tačkama, pronaći korisnička imena i lozinke svih registrovanih korisnika.

Vežba 2 – Blind SQL injection

Napomena: U slučaju da stranica po pristupu ne izgleda kao na slici, osvežiti stranicu bez korišćenja keša. (u Google Chrome-u pritisnuti CTRL + SHIFT + R da bi se stranica osvežila bez korišćenja keširanja starih podataka, a potom ponovo kliknuti refresh - F5 - po potrebi)

Ovaj tip napada bukvalno se prevodi „*SQLi* na slepo“. Napad spada u teže napade za ručno izvršavanje, a težini najviše doprinosi što se napad veoma sporo izvršava. Korisnik ni u jednom trenutku ne vidi podatke iz baze na stranici, već na drugi način mora da „pogađa“ podatke, pri čemu na stranici može da se vidi da li je podatak „pogođen“ ili nije. Laboratorijska vežba je prilagođena *content-based* metodi jer su SQL upiti koji se koriste prilikom napada za nijansu jednostavniji. Vežba može da se reši i *time-based* metodom.



Aplikacija na kojoj se ova vežba zasniva je veb prodavnica hrane i opreme za kućne ljubimce. Korisnici mogu da se registruju kako bi mogli da pretražuju hranu i opremu. Svi registrovani korisnici se nalaze u bazi podataka. Ulogovani korisnici imaju mogućnost promene svoje tekuće lozinke. Vežba počinje na stranici za promenu lozinke prijavljenog korisnika sa korisničkim imenom „dogwhisperer“ kao na slici. Stranici se pristupa iz pretraživača na `http://192.168.x.1:8093`. Ovo je ujedno i jedina stranica platforme kojoj je moguće pristupiti. U polja je potrebno uneti staru i željenu novu lozinku korisnika, a potom pritisnuti dugme *Update*. U slučaju uspešne promene lozinke prikazuje se poruka “You successfully changed your password!”, dok se u slučaju neuspešne promene lozinke prikazuje poruka “Your old password is not correct!”. Na ovaj način poruka o uspehu ili grešci napadaču govore da li je *inject*-ovani upit vratio *true* ili *false*. Napadač koristi povratnu vrednost da bi beležio da li je podatak “pogođen” ili nije. Pored promene lozinke, korisnik na ovoj stranici može i da resetuje bazu podataka. Resetovanjem baze podataka ona se vraća u početno stanje. Voditi računa da su se svaki put, prilikom ispisa poruke uspeha, podaci u bazi uspešno promenili te je potencijalno potrebno resetovati bazu pre nastavljanja napada.

1. Napad se izvršava kroz vrednost stare lozinke.
    - Uneti vrednost `lozinka` u oba polja i potvrditi. Koja poruka je ispisana na stranici?
    - Uneti vrednost `lozinka` u polje za novu lozinku. U polje za staru lozinku uneti vrednost `' or 1=1--` (Voditi računa o razmacima! S obzirom da se na stranici pri kucanju lozinke ne vide karakteri koji se kucaju, predlaže se da se vrednost otkuca u tekstualnom editoru, a potom prekopira). Koja poruka je ispisana na stranici. Šta se desilo? Da li su podaci u bazi promenjeni i ako jesu, koji?
2. Potrebno je pronaći ime baze. Ime baze može da se nađe pogađanjem karakter po karakter i posmatranjem poruke o uspehu i grešci na ekranu. Pre pogađanja, potrebno je pronaći broj karaktera u imenu baze. Uneti sledeću vrednost u polje za staru lozinku. U polje za novu lozinku uneti bilo šta.

   ```
   ' or select 1 from information_schema.schemata where schema_name=schema()
   and length(catalog_name) = 1--
   ```

    - Koja poruka je ispisana na stranici? Šta to znači?
    - Promeniti vrednost dužine imena baze koju pogađamo na 2 i probati ponovo.
    - Menjati vrednost dužine imena baze dok se ne desi pogodak. Koliko karaktera ima u imenu baze?

   Koristeći prethodno saznanje, u polje za staru lozinku uneti sledeću vrednost:

   ```
   ' or select 1 from information_schema.schemata where schema_name=schema()
   and substring(catalog_name,1,1) = 'A'--
   ```

    - Koja poruka je ispisana na stranici? Šta to znači?
    - Promeniti vrednost prvog slova u imenu baze koje služi za pogađanje na ‘B’ i probati ponovo.
    - Pronaći prvo slovo u imenu baze. (Imati u vidu da *h2* baze podrazumevano ne razlikuju velika i mala slova te je dovoljno isprobavati samo velika slova alfabeta, cifre i donju crtu).

Koristeći prethodno saznanje, u polje za staru lozinku uneti sledeću vrednost:
`' or select 1 from information_schema.schemata where schema_name=schema() and substring(catalog_name,1,2) = 'xA'--`, gde je "x" pronađeno prvo slovo.

- Pronaći drugo slovo imena baze, a zatim i celo ime. Da li je potrebno pogađati sva slova? Da li deo imena baze može da se zaključi posle nekoliko pogodaka? Kako biste Vi nazvali bazu da otežate posao napadaču?

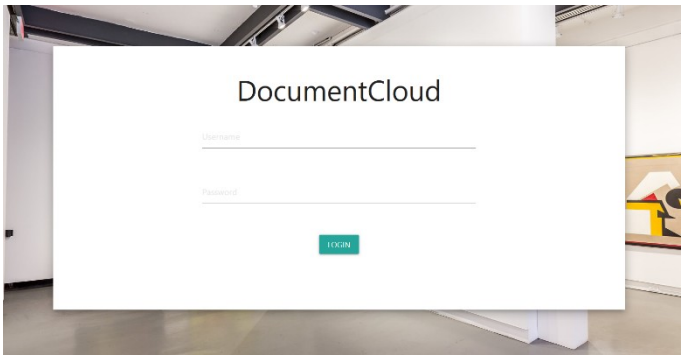
3. Nakon pronalaska imena baze, potrebno je pronaći ime tabele u bazi koja čuva podatke o lozinkama. Koristeći sledeću vrednost za staru lozinku, pronaći broj tabela u bazi:
```
' or select 1 from information_schema.tables where table_schema=schema() and (select count(*) from information_schema.tables where table_schema=schema() = 1)--
```
   - Da li je pronađen broj tabela?

   Koristeći prethodno saznanje, pronaći dužinu imena tabele, a zatim i samo ime koristeći sledeće:
```
' or select 1 from information_schema.tables where table_schema=schema() and length(table_name) = 1--
' or select 1 from information_schema.tables where table_schema=schema() and substring(table_name,1,1) = 'A'--
```
   Po potrebi menjati parametre dok se ne desi pogodak.
   - Koje je ime tabele u bazi? Da li je ovog puta bilo potrebno pogađati sva slova?
4. Nakon pronalaska imena tabele, potrebno je pronaći ime kolone u kojoj se čuvaju lozinke. Uneti sledeću vrednost u polje za staru lozinku:
`' or (select count(*) from information_schema.columns where table_name = '***' and table_schema = schema()) = 1--`, gde je *** ime odgovarajuće tabele.
   - Menjati parametre dok se ne desi pogodak. Koliko kolona postoji u tabeli?

   Uneti sledeću vrednost u polje za staru lozinku:
```
' or select 1 from information_schema.columns where table_name='***' and table_schema = schema() and length(column_name) = 1--
```
   - Menjati parametre za dužinu imena kolone dok se ne desi pogodak.
   - U opštem slučaju moguće je da postoji više kolona u tabeli. U tom slučaju će gornji deo upita da vrati grešku ukoliko više kolona ima istu dužinu imena. Izmeniti upit dodajući mu sledeći deo između vrednosti dužine imena kolone i oznake komentara:
     `and ORDINAL_POSITION = 1`
     Ordinalna pozicija je pozicija kolone unutar tabele u bazi. Koristeći ordinalnu poziciju pronaći dužine imena svih kolona u tabeli.

   Uneti sledeću vrednost u polje za staru lozinku:
```
' or select 1 from information_schema.columns where table_name='***' and table_schema = schema() and substring(column_name,1,1) = 'A' and ORDINAL_POSITION = 1--
```
   - Menjati parametre za podstring koji se traži i ordinalnu poziciju kolone u tabeli dok se ne pronađu imena svih kolona u tabeli. Da li neka imena možemo naslutiti?
5. Nakon pronalaska odgovarajuće kolone, pronaći lozinku korisnika "dogwhisperer" unoseći sledeće vrednosti u polje za staru lozinku i menjajući parametre do pogodka.
```
' or select 1 from TABLE where COL1='dogwhisperer' and length(COL2) = 1--
' or select 1 from TABLE where COL1='dogwhisperer' and substring(COL2,1,1) = 'a'--
```
   gde je TABLE ime odgovarajuće tabele, COL1 ime kolone za korisnička imena, a COL2 ime kolone za lozinke.
   Pretpostaviti da je korisnikova šifra stvarna reč napisana malim slovima i na osnovu toga birati slova!
   Voditi računa da se lozinke menjaju svakim pogotkom! Resetovati bazu nakon svakog pogotka!

Vežba 3 – SQL injection login bypass

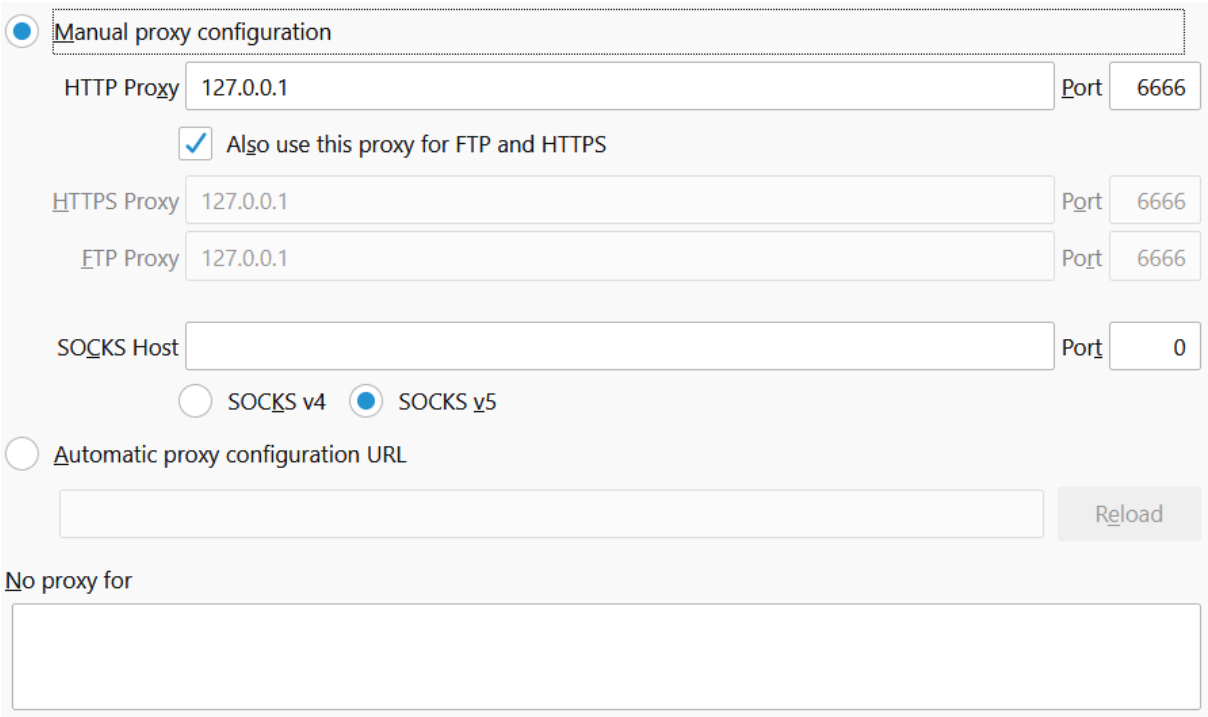


Cilj ovog tipa napada je zaobići prijavu, odnosno prijaviti se ne znajući nijedno korisničko ime ni lozinku u bazi. Napad je vrlo jednostavan i izvršava se na sličan način kao i u prvoj vežbi. Iz tog razloga ova laboratorijska vežba za nijansu komplikuje napad dodavanjem validacije korisničkog unosa. Aplikacija na kojoj se zasniva ova vežba je veb portal za skladištenje dokumenata u *cloud*-u. Vežba počinje na stranici za prijavu korisnika, kao na slici, kojoj se može pristupiti na `http://192.168.x.1:8094/login.html`. Aplikacija se brani od SQLi tako što prilikom unosa korisničkog imena i lozinke dozvoljava jedino slova i brojeve. Time je onemogućen unos specijalnih karaktera kao što su apostrof (’), taraba (#) i crtica (-) koji se često koriste za napad. U aplikaciji postoji propust, jer se validacija korisničkog unosa radi samo na klijentskoj strani, odnosno u pretraživaču, dok server prihvata sve podatke. Kako bi se izvršio napad potrebno je zaobići validaciju, na primer, presretanjem zahteva koji putuje od pretraživača ka serveru i promenom podataka koje zahtev nosi. Koncept je opisan sledećom slikom.

Zahtev se može presresti korišćenjem *HTTP Proxy* servera kao što je Burp Suite. Kada zahtev stigne do *proxy* servera pauzira se i čeka na korisničku akciju, prosleđivanje ili odbacivanje zahteva. Vidljivi su i meta-podaci koje zahtev nosi i korisniku je dozvoljeno da ih promeni.

1. Instalirati alat Burp Suite na mašini na kojoj je pokrenut VPN. Link za preuzimanje alata je dat u nastavku:
   https://portswigger.net/burp/communitydownload
2. Pokrenuti alat i konfigurisati ga kako bi presretao zahteve:
   - U Proxy->Options jezičku dodati novi ili podesiti postojeći *proxy listener* kao na slici:

3. Podesiti pretraživač da radi sa Burp Suite alatom. Za *Firefox* uraditi sledeće:
   - U Firefox meniju, kliknuti na Options, a potom na Advanced i naći sekciju Network settings. Kliknuti na Settings. Podesiti parametre kao na slici i kliknuti OK.



4. U Burp Suite alatu, u Proxy -> Inercept uključiti opciju presretanja (potrebno je da piše Interecpt is on).

5. Na stranici ukucati proizvoljno ime i lozinku i pritisnuti na Login. Burp Suite alat je presreo zahtev i može se videti u Proxy -> Intercept -> Raw jezičku.

6. Izmeniti vrednost korisničkog imena na dnu otvorenog Raw jezička unoseći sledeći deo SQL upita:
   `' or 1=1--`
   Kliknuti na Forward. Zahtev je poslat na server i Burp je presreo odgovor. Isključiti opciju presretanja (Intercept is off), kako Burp dalje ne bi presretao sve zahteve. Koja stranica je prikazana? Šta se desilo?