Na sledecem primeru cemo vam pokazati losu praksu pri izradi veb aplikacija.

Naime, aplikacija koju trenutno vidite korisnicima nudi mogucnost rezervisanja avio karata i

iznajmljivanja vozila. Kako bi koristili usluge aplikacije korisnici prethodno moraju da se

registruju i ostave svoje podatke. Jedna od funkcionalosti koja im se nudi jeste povezivanje sa drugim registrovanim korisnicima kako bi bili u mogucnosti da organizuju zajednicka putovanja sto ce biti kljucno za demonstraciju.

Sama aplikacija realizovana je kao veb aplikacija koja se sastoji iz frontenda realizovanog u Angular okruženju, backenda kreiranog pomoću rest servisa i komunikacija izmedju njih vrsi se slanjem http zahteva i odgovora koji se dobavljaju iz baze podataka.

Popust koji programeri često prave jeste slanje viška podataka preko mreže. Veoma je bitno obratiti

pažnju na to koji podaci su neophodni za realizaciju konkretne funkcionalnosti i samo njih prosleđivati.

U ovom primeru mi smo prikazali koje su slabosti slanja celokupnog objekta i kako ih zlonamerni korisnici mogu iskoristiti. Prilikom dodavanja prijatelja, potrebno je učitati sve korisnike koji su registrovani i odgovaraju pretrazi ulogovanog korisnika. Umesto da preko mreže prosledimo samo ime i prezime korisnika, slali smo

ceo objekat, te osoba koja želi da izvrši napad na naš sistem, može presresti odgovor i time iščitati sve podatke koje imamo o korisnicima a zatim ih zloupotrebiti.

Druga mana aplikacije jeste loša kofiguracija kriptografije odnosno korišćenje slabe kriptografske funkcije sažimanja.

Naime, funkcije poput MD5, SHA0 i SHA1 se danas ne mogu smatrati bezbednim jer

postoje napadi koji su sposobni da pronađu koliziju za relativno kratko vreme. Umesto default-ne

algoritma koji koristi ASP.NET Core Identity, u datom primeru koristili smo SHA1.

Ukoliko napadač iskoristi prvu slabost sistema, presretne podatke i pročita hash-ovanu vrednost lozinke, na veoma jednostavan način će moći da dobije čist vrednost lozinke koristeći neke od postojećih aplikacija i sajtova za dekripciju. Mi smo konkretno koristili hashes.com sajt.

Ovo su samo neki od primera lose konfiguracije, o kojima je veoma vazno razmisljati pri samom planiranju i izradi aplikacije.

Naravno, ovaj primer je veoma jednostavan, ali kada pricamo o ozbiljnim aplikacijama i sistemima potrebno je uvideti koliko je

vazno ulagati u bezbednost sistema. Kolega ce u nastavku objasniti jos neke slabosti i resenja za iste.