CARNet

HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Preuzimanje sjednice

NCERT-PUBDOC-2011-06-327

Nacionalni CERT

Sadržaj

1 Uvod

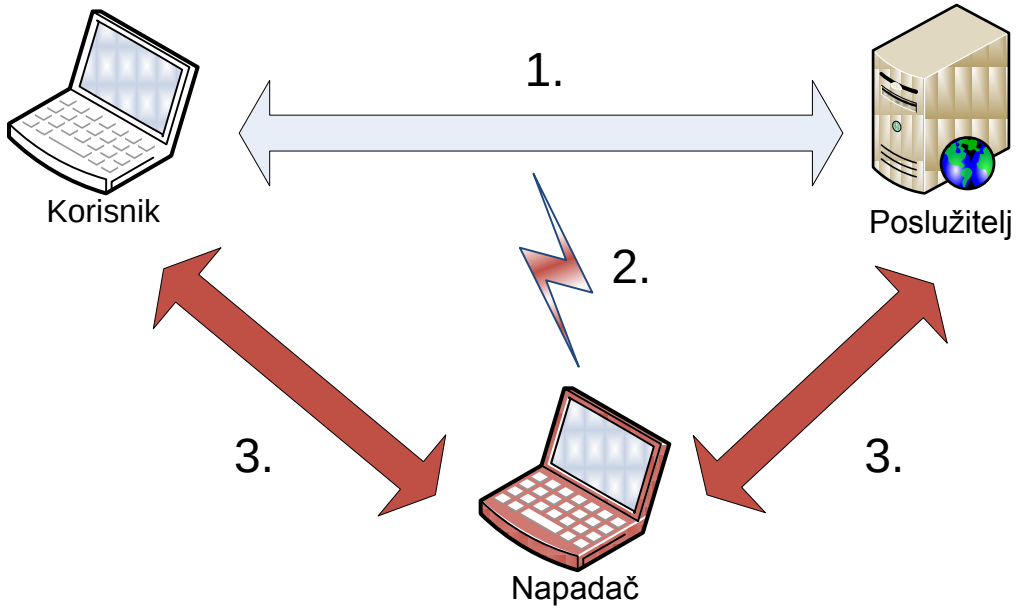
Općeprihvaćenost društvenih mreža i drugih servisa koji na udaljenim računalima pohranjuju korisničke podatke dovodi prosječnog korisnika u situaciju da su njegovi vrlo povjerljivi podaci dostupni na vrlo jednostavan način. Internetsko bankarstvo, socijalne mreže, elektronička pošta samo su neke od tih usluga. Dostupnost informacija sa tih usluga može dovesti do izravne financijske štete (internetsko bankarstvo) ili neizravne (društvene mreže). Kako bi izbjegli takve potencijalne opasnosti korisnici ih moraju biti svjesni te naučiti kako ih izbjeći.

Krađa sjednice jedna je od takvih izravnih prijetnji, a može dovesti do neposrednog otkrivanja povjerljivih informacija. Ona može biti izvedena na više razina, od razine TCP konekcije, do aplikacijske razine.

Aplikacijska razina trenutno je najopasnija za prosječnog korisnika zbog jednostavnosti napada u nekim slučajevima. Jednostavnost napada proizlazi iz gotovih aplikacija koje omogućavaju napadačima sa malo znanja krađu sjednice, a time i povjerljive informacije.

2 Krađa Sjednice

U daljnjem tekstu ćemo se osvrnuti na dva osnovna i vjerojatno najpopularnija načina krađe sjednice, krađu TCP sjednice i krađu HTTP sjednice pomoću Cookiea. U oba slučaja napadač prvo mora saznati različite informacije o sjednici te na temelju njih pokušati izvesti napad. Ovisno o tipu napada, na temelju podataka sakupljenih različitim metodama, napadač postavlja sebe kao drugi kraj komunikacije s poslužiteljem. Također, ovisno o cilju napada, promet može biti i proslijeđen žrtvi i na taj način korisniku svoje prisustvo učiniti nevidiljivim. Na slici 1. vidimo jedan uobičajeni tijk napada. U prvom koraku korisnik (žrtva) uspostavlja komunikaciju sa poslužiteljem, dok napadač sluša tu komunikaciju. Kada sakupi dovoljno podataka pokušava napad, a ukoliko je uspješan, odgovori poslužitelja će od tog trenutka stizati njemu. Ukoliko želi biti transparentan žrtvi, odgovore poslužitelja će proslijediti žrtvi kako ona uopće ne bi bila svjesna napada [1].



Slika 1. Tijek napada

3 Krađa TCP Sjednice

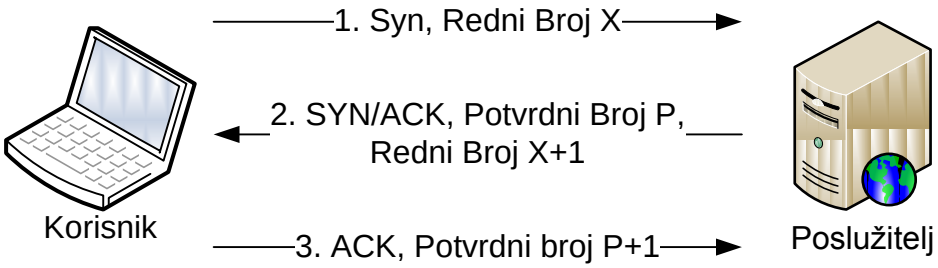
Krađa TCP sjednice temelji se na presretanju već uspostavljene veze između bilo koja dva računala. Napadač se zatim predstavlja kao jedna od strana u komunikaciji, na način da druga uopće ne primijeti promjenu. Napadi na TCP sjednicu mogu se podijeliti na dva osnovna tipa:

1. Postavljanjem sebe u postojeću komunikaciju (eng. Middle Man Attack)
2. Slijepi napad (eng. Blind Attack)

Bilo koji od ova dva napada počinje mijenjanjem izvorišne adrese IP paketa (eng. IP Spoofing) koji želimo poslati poslužitelju. Napadač preuzima IP adresu žrtve, kako bi poslužitelj i dalje mislio da komunicira sa žrtvom koja je uspostavila komunikaciju. Kada je ovakav napad izveden unutar LAN (najčešće WLAN) mreže, napadač ujedno koristi i lažnu MAC adresu kako bi zavarao mrežnu opremu i primao pakete namjenjene računalu sa preuzetom IP adresom. Sama promjena IP adrese u paketima je jednostavna, problem nastaje pri odabiranju ostalih parametara u paketu kao što su redni brojevi. Upravo u načinu na koji se odabiru i predviđaju redni brojevi razlikujemo ove napade [2].

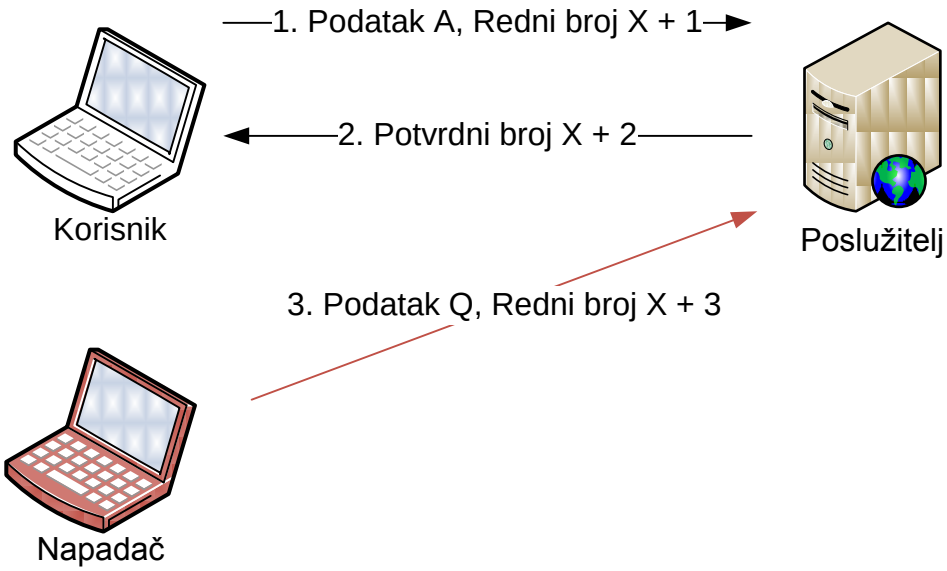
3.1 Umetanjem sebe u postojeću komunikaciju

Za ovu tehniku je potrebno snimati mrežni promet između korisnika i poslužitelja. Nakon snimanja paketa napadač može pristupiti TCP zaglavlju paketa kako bi vidio redni broj te samim time predvidjeti koji sljedeći broj poslužitelj očekuje.



Slika 2. Početna razmjena potvrdnog broja i rednog broja

Nakon što odredi sve potrebne parametre napadač mora poslati lažirani paket prije žrtve (Slika 3). Kada paket stigne do poslužitelja njegov redni broj u TCP protokolu se poveća i izvorni žrtvin paket više ne vrijedi zato što ne odgovara trenutnom očekivanom broju (koji je sada za jedan veći).



Slika 3. Napadač šalje pakete

Drugi način posredničkog napada je moguće napraviti ukoliko se može uvjeriti žrtvu da je napadačevo računalo izlazni uređaj iz mreže (eng. gateway) te da sve pakete šalje preko napadača. Takav napad je moguće izvršiti pomoću ARP spoofinga, odnosno slanja lažnog ARP paketa koji ima izmijenjen par IP adresa, MAC adresa.

3.2 Slijepi napad

Ovakav tip napada koristi se kada nije moguće presretati podatke koje šalje žrtva čiju sjednicu želimo preuzeti. Šanse za uspjeh ovog napada prilično su male jer je potrebno pogoditi trenutne redne brojeve paketa kojima komuniciraju poslužitelj i žrtva.

4 Krađa sjednice na aplikacijskom nivou

Pri krađi sjednice na aplikacijskom nivou napadač ne preuzima samo sjednice, već i stvara nove koristeći pronađene podatke. Cilj svih postupaka krađe sjednice na aplikacijskom nivou je pronalazak pravog identifikatora sjednice (eng. Session ID). Kada je napadač u posjedu tajnog Session ID-a, može preuzeti postojeću ili stvoriti novu sjednicu.

4.1 Pronalazak podataka o sjednici

Aplikacije koriste identifikator sjednice kako bi identificirali korisnika. Potrebni podatci se obično nalaze na jednoj od tri lokacije.

1. Unutar URL-a koji se šalje u sklopu HTTP GET zahtjeva,
2. Unutar polja koja se predaju aplikaciji u HTTP POST zahtjevu,
3. Unutar Cookiea.

Sva tri mjesta mogu biti dostupna napadačima. URL-ovi su dostupni u povijesti preglednika, isto kao i Cookie-i. Podatci poslani HTTP POST zahtjevom ne nalaze se pohranjeni na računalu (web preglednik ih ne zapisuje), no bilo kojem napadaču s mogućnošću prisluškivanja promet u mreži je na vrlo lak način dostupan. Isto vrijedi i za Cookie i URL-ove. Ukoliko podatci nisu dostupni napadačima, još uvijek postoje izgledne šanse za pogađanje podataka i krađu drugim načinima [3].

- Pogađanje grubom silom je jedna od opcija. Ukoliko je identifikator sjednice predvidljiv ili djelomično predvidljiv napadač može izvesti automatizirani napad koji će pogađati identifikator dok ga ne pogodi. Identifikatori mogu biti prekratki, stvarati se iz javno poznatih podataka (kao korisnička IP adresa i sl.), a svi ti algoritmi koji nisu u potpunosti zasnovani na slučajno odabranim sastavnicama značajno smanjuju potrebno vrijeme za napad.
- Druga mogućnost je cross-site scripting (XSS) napadom uvjeriti korisnikov web-preglednik da pošalje potrebne podatke o sjednici.

4.1.1 Podatci unutar URL-a

URL te svi ostali podaci poslani u HTTP zahtjevu dostupni su svakome tko može preslušavati mrežni promet. Primjer jednog takvog zahtjeva je dan na slici 4.

```
GET / HTTP/1.0
Accept: text/plain
Accept: text/html
Session-Id: SID:ANON:w3.org:j6oAOxCWZh/CD723LGeXlf-01:034
User-Agent: libwww/4.1
```

Slika 4. Primjer Session-Id unutar URL zahtjeva

4.1.2 Unutar polja u HTTP POST naredbi

Kada se informacije o sjednici nalaze unutar polja koji se predaju aplikaciji. Uobičajena praksa je postaviti identifikator sjednice kao jedno od skrivenih polja koja se automatski predaju aplikaciji putem HTTP POST naredbe. Primjer jednog takvog polja unutar HTML stranice je na slici 5.

```
<FORM METHOD=POST ACTION=”/cgi-bin/news.pl”>
<INPUT TYPE=”hidden” NAME=”sessionid” VALUE=”IE60012219”>
<INPUT TYPE=”hidden” NAME=”allowed” VALUE=”true”>
<INPUT TYPE=”submit” NAME=”Read News Article”>
```

Slika 5. Skriveno HTTP POST polje

4.1.3 Podatci unutar Cookiea

Većina stranica na kojima korisnici imaju izrađene korisničke račune pomoću cookie-a prenosi stanje sjednice. Pri prvom pristupu stranici, poslužitelj će zadati cookie, koji će preglednik tada u svakom slijedećem zahtjevu slati u sklopu zaglavlja. U praktičnom smislu to znači da korisnik neće ponovno morati upisivati korisničko ime i lozinku kako bi pristupio svom korisničkom računu. Količina podataka koja je sadržana u cookie-u ovisi o implementaciji web aplikacije, a jednako tako o web aplikaciji ovisi i vremenski period za koji vrijedi pojedini cookie. Primjer podataka unutar jednog cookiea je dan na slici 6.

```
sessionID=”IE60012219”; path=”/”; domain=”www.example.com”;
expires=”2003-06-01 00:00:00GMT”; version=0
```

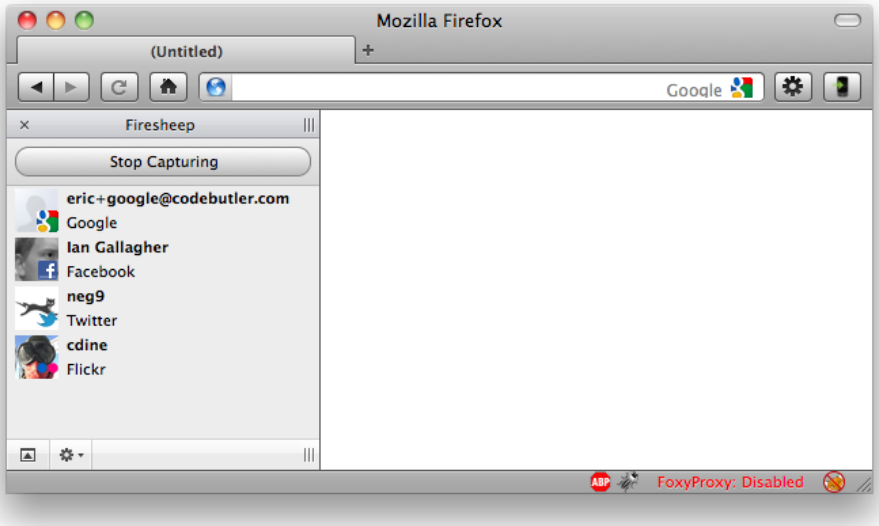
Slika 6. Primjer podataka unutar cookiea.

4.2 Ranjivosti navedenih pristupa

Sva tri pristupa obilježavaju slični nedostaci, ukoliko je mrežni promet vidljiv napadaču, a to je čest slučaj kod bežičnih mreža. Ukoliko napadač uspije reproducirati pakete s poljima koja ispravno imitiraju ona na računalu žrtve, može preuzeti njegov identitet. Ranjivosti koje proizlaze iz ovakvog načina održavanja sjednice uglavnom su poznate te ih je moguće izbjeći. Posljednjih godina, porastom primjene slabo ili nikako zaštićenih bežičnih mrežaproblem je eskalirao. Trenutno postoje lako dostupni gotovi alati koji na nezaštićenim bežičnim mrežama preuzimaju sesije korisnika spojenih na zajedničku pristupnu točku.

4.2.1 FireSheep

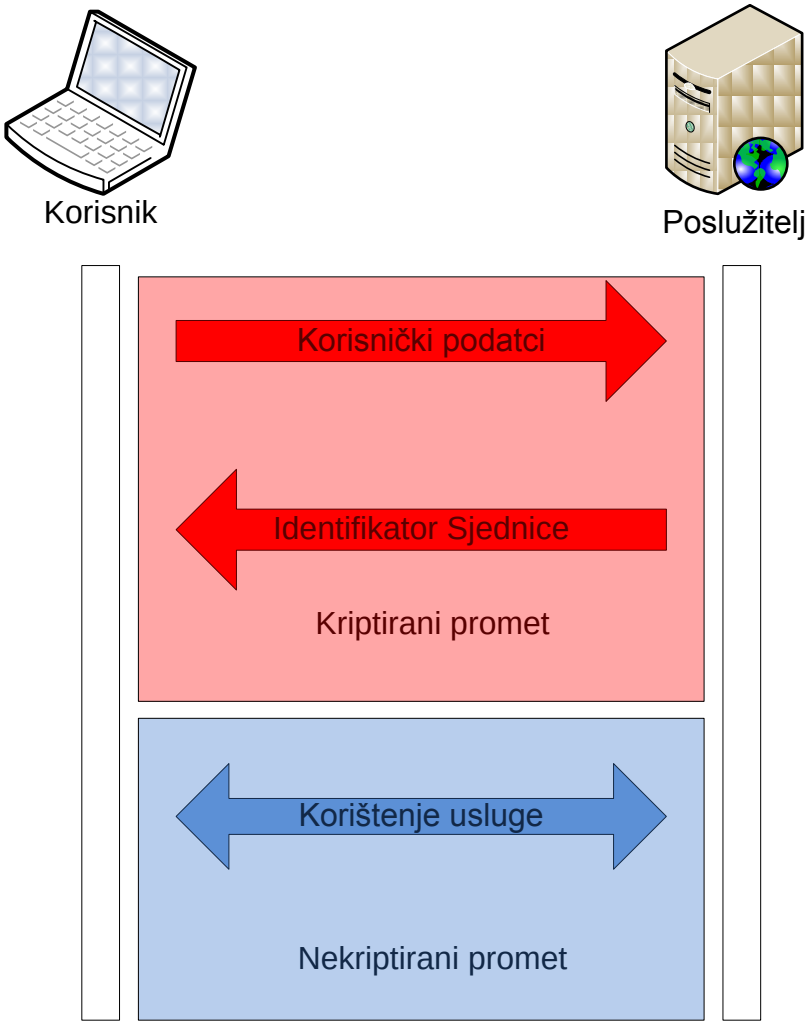
Dodatak za web-preglednik Firefox pomoću kojega je moguće vrlo jednostavno snimati promet na mreži te automatski prepoznavati i reproducirati cookie popularnih internetskih servisa poput Facebooka, eBaya, MySpacea, Twittera, Flickr, Google itd. Nudi grafičko sučelje (Slika 7.) i jednostavno upravljanje u nekoliko klikova mišem. FireSheep vrši napad reprodukcijom postojećih cookie-a, no ne i mogućnost njihovog mijenjanja. Kako bi FireSheep radio potrebno je moći slušati promet na mreži, a ta mogućnost ovisi o pogonskim programima mrežnih adaptera. Na Windows operacijskom sustavu sa standardnim pogonskim programima to nije moguće, dok je na Linux operacijskim sustavima to moguće korištenjem posebno pisanih pogonskih programa.



Slika 7. Grafičko sučelje Firesheep alata

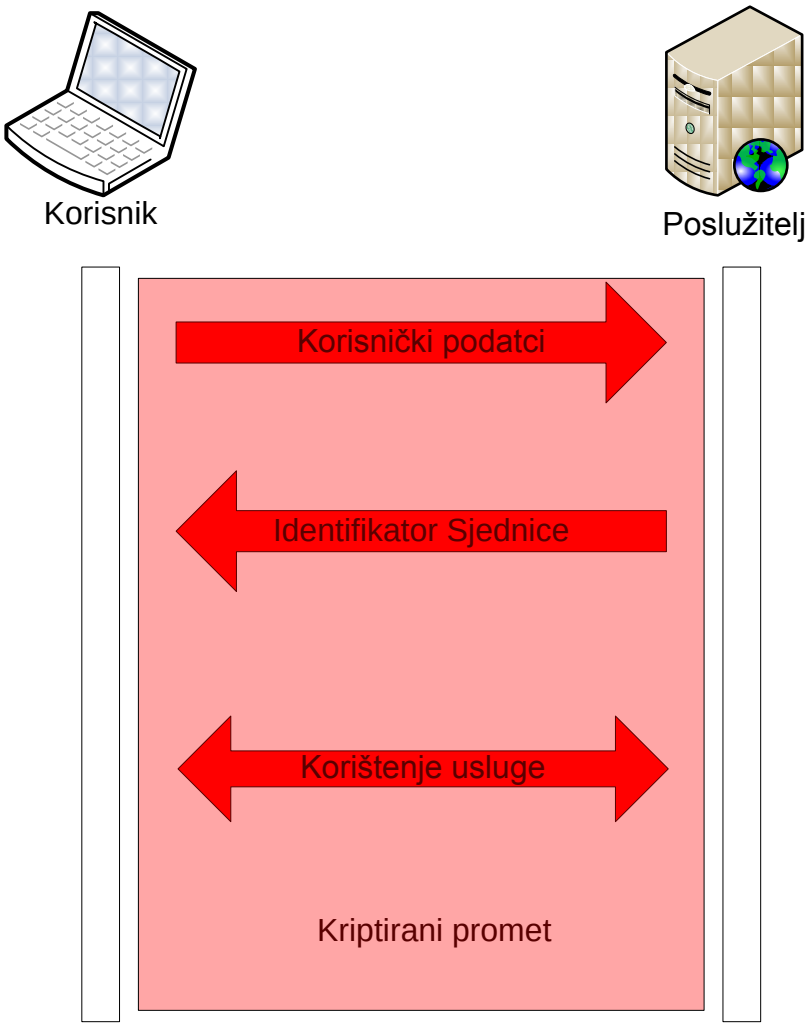
4.3 Zaštita sjednice na aplikacijskoj razini

Za zaštitu sjednice koja se održava između poslužitelja i preglednika u prvom redu se koristi enkripcija. Promet koji je kriptiran onemogućava napadaču napad prisluškivanjem. Nažalost većina trenutno popularnih servisa omogućava samo kriptiranje faze predaje korisničkih podataka (Slika 8.)



Slika 8. Kriptiranje autentifikacije

Ovakav pristup uglavnom samo zaštićuje korisničke podatke, ali ne sprječava napad preuzimanjem sjednice koji, ukoliko je uspješan, napadaču omogućava potpunu kontrolu nad korisničkim računom žrtve. Jedini pravi način zaštite korisnika je korištenje HTTPS odnosno TLS/SSL kriptiranja u cjelokupnoj komunikaciji sa poslužiteljem (Slika 9.). U tom slučaju se pojavljuju problemi sa opterećenjem poslužitelja koji mora sav promet kriptirati, odnosno dekriptirati prije komunikacije s korisnikom.

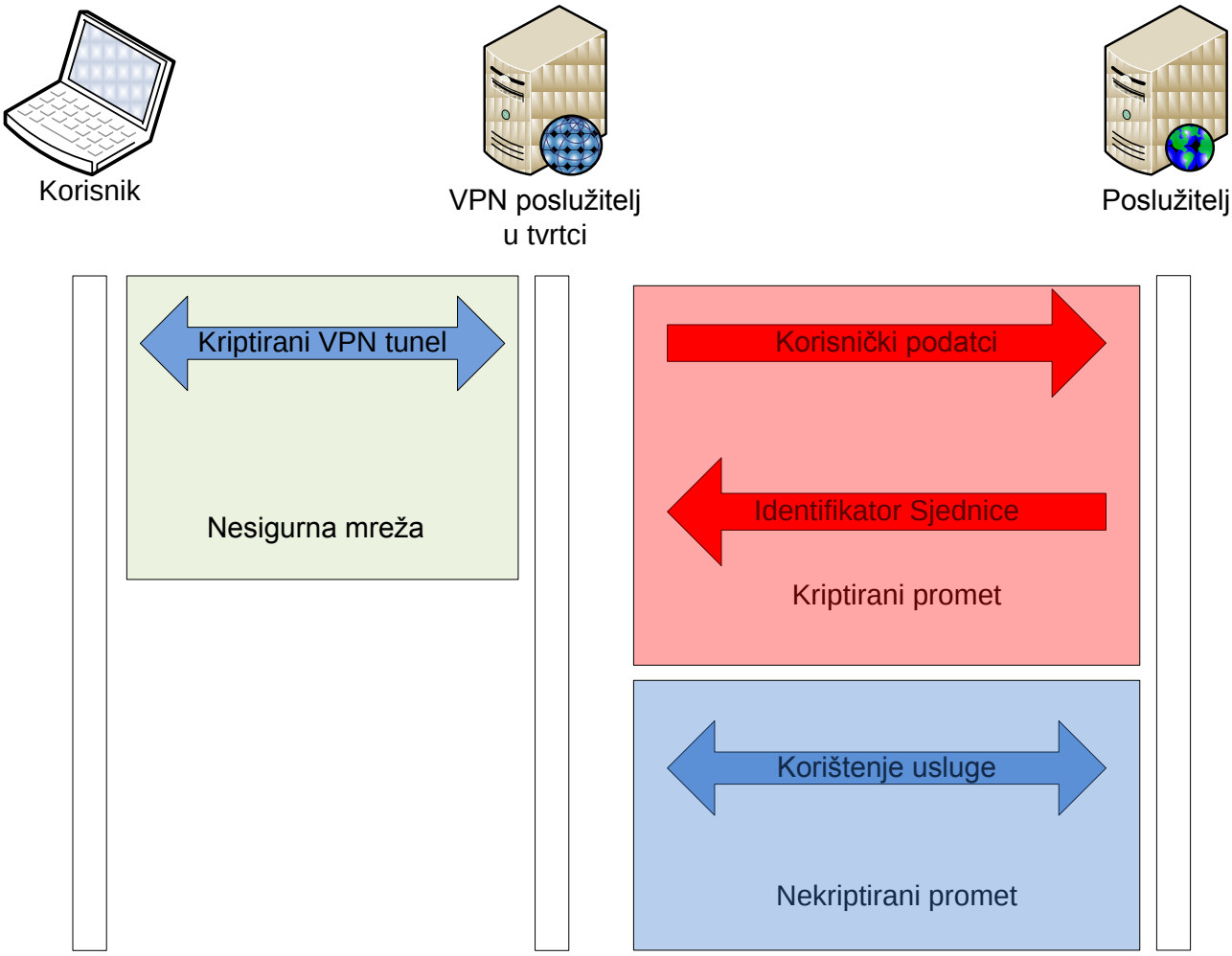


Slika 9. Cjelokupni kriptirani promet

4.4 Preporuke kako se zaštiti

Svaki korisnik će se naći u situaciji kada bi trebao pristupiti nekom internetskom servisu preko nezaštićene ili nedovoljno zaštićene mreže. Kako bi i povećali sigurnost korištenja tih usluga treba se pridržavati nekih smjernica:

- Ukoliko je moguće uvijek koristiti HTTPS, postoje dodatci za web-preglednike koji omogućavaju u nekim slučajevima takvu komunikaciju kroz cijelo korištenje usluge, a ne samo pri pristupanju usluzi. Jedan od tih dodataka je Force-TLS [4]
- koristiti VPN vezu prema nekoj sigurnoj točci, poput tvrtke (Slika 10.). Na opisani način korisnik može biti zaštićen pri korištenju web aplikacija s neadekvatno zaštićenim podacima o sjednici.



Slika 10. Komunikacija putem VPN tunela

- Ne koristiti poslužitelj elektroničke pošte koji radi putem POP protokola u javnim mrežama, budući da se korisnički podatci ne kriptiraju
- Izbjegavati online kupovinu u nepoznatim mrežama čak i ako je prisutna enkripcija, jer je u slučaju da napadač kontrolira računalo s kojeg se transakcija obavlja enkripcija beskorisna (napadač sa samog računala može pristuiti svemu što vidi i sam korisnik)

5 Zaključak

Krađa sjednice je ozbiljan problem koji može pogoditi svakog korisnika web aplikacija te je potrebno provoditi mjere opreza kako ne bi do toga došlo. Krađa sjednice na razini TCP protokola nije toliko uobičajena i vjerojatnost je da će takvi napadi češće biti usmjeravani ka velikim infrastrukturama, dok je krađa sjednice na aplikacijskoj razini opasnost za sve korisnicike. Informiranim korištenjem web aplikacija i omogućavanjem enkripcije gdje god je to moguće, opasnosti se uvelike mogu smanjiti, no ne i u potpunosti ukloniti.

6 Literatura

1) Session Hijacking Exploiting TCP, UDP and HTTP Sessions, Shray Kapoor, 2010

2) Web Based Session Management Best practices in managing HTTP-based client sessions, Gunter Ollmann, 2007

3) An Overview of Session Hijacking at the Network and Application Levels, Mark Lin, 2005

4) Force-TLS, https://addons.mozilla.org/en-US/firefox/addon/force-tls/, Sid Stamm