SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

DNS Rebinding

Edhem Avdagić

Mentor: *Predrag Pale*

Zagreb, svibanj, 2019.

Sadržaj

1. Osnove interneta 3
   - 1.1. IP adrese i domene 3
   - 1.2. DNS – Domain Name System 4
   - 1.3. Web preglednici i same-origin policy 6
2. DNS rebinding 7
   - 2.1. Primjer – FER intranet 7
   - 2.2. Implementacijski problemi 8
   - 2.3. Zašto nije popravljeno? 8
3. Zaključak 9

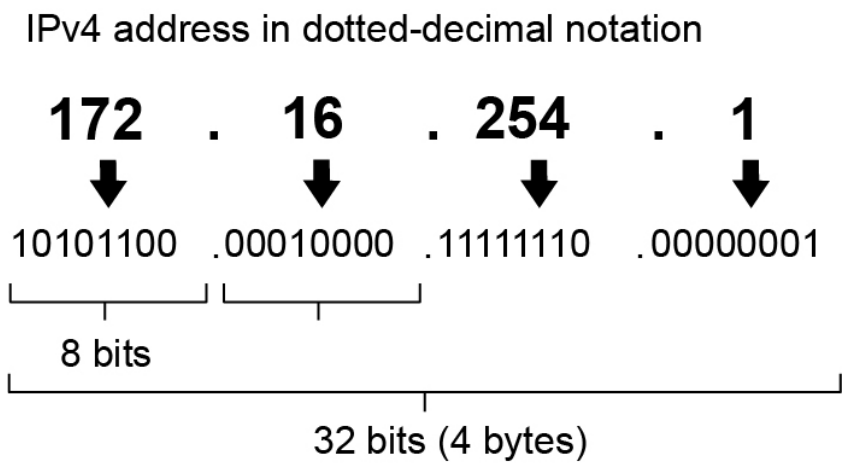
1. Osnove interneta

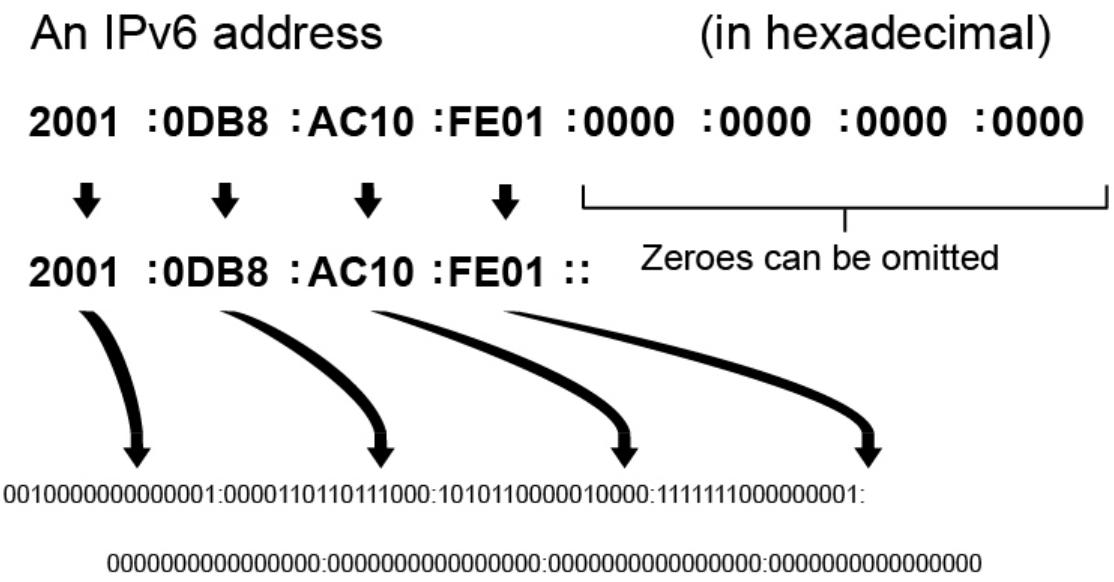
Iako su uveliko olakšala svakodnevni život znanstvenika, računala su u početku imala veliku manu: nemogućnost međusobne komunikacije. Iz tog razloga, 60-ih godina prošlog stoljeća vlada SAD-a počinje financirati projekt izgradnje robusne infrastrukture koja bi omogućila razmjenu podataka i komunikaciju između računala. Rezultat projekta bio je ARPANET koji je do 1980-ih služio za povezivanje kako vladinih, tako i akademskih mreža. Tada na scenu stupa NSFNET (National Science Foundation Network) koji kreira novu infrastrukturu korištenu za uvezivanje akademskih i istraživačkih centara, a početkom 90-ih godina mreža postaje dostupna i komercijalno. Komercijalizacijom interneta nastaju mnogi servisi koje danas uzimamo zdravo za gotovo: email, digitalne novine, video streaming servisi i mnogi drugi. Iako nema nikakav centraliziran autoritet za nadgledanje i kontrolu, ipak postoje dvije standardne definicije za uspostavu komunikacije, IP i DNS.

1.1. IP adrese i domene

Internet Protocol (IP) je osnovni komunikacijski protokol zadužen za usmjeravanje i dostavu paketa podataka na osnovu IP adresa u zaglavljima istih. Kako bi takva komunikacija bila moguća, koriste se usmjerivači (engl. *router*) koji određuju putanju paketa kroz mrežu, a kako IP protokol nema mehanizme pouzdanosti, usmjerivači mogu svoj posao obavljati dosta brzo. S druge strane, bez mehanizama pouzdanosti postoji i mogućnost da neki paketi nikad ne stignu do odredišta, te se za aplikacije kojima je potrebna pouzdanost koriste drugi mehanizmi i protokoli (obično zasnovani na IP protokolu).

Kako bi usmjerivači znali gdje poslati paket odnosno odakle paket dolazi, koriste se IP adrese. IP adrese jedinstveno identificiraju računala/uređaje povezane u mrežu. Prema trenutno najraširenijem standardu, IPv4, IP adresu predstavlja 32-bitni binarni broj, ali zbog sve manjeg broja slobodnih IPv4 adresa, uveden je i IPv6 koji za adrese koristi 128 bitova. Kako bi IP adrese bile razumljivije ljudima, obično se zapisuju u dekadskoj notaciji sa točkama između okteta:

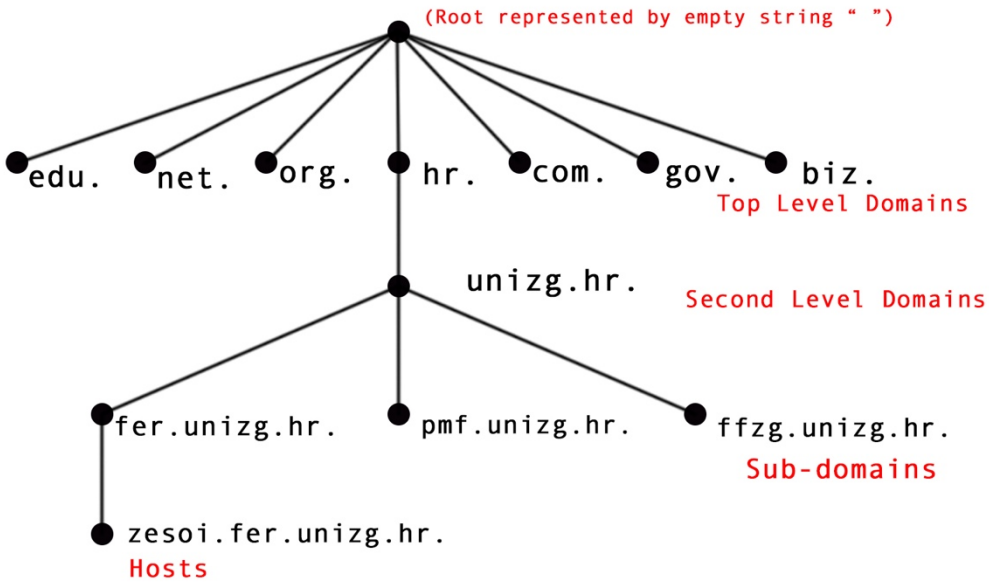




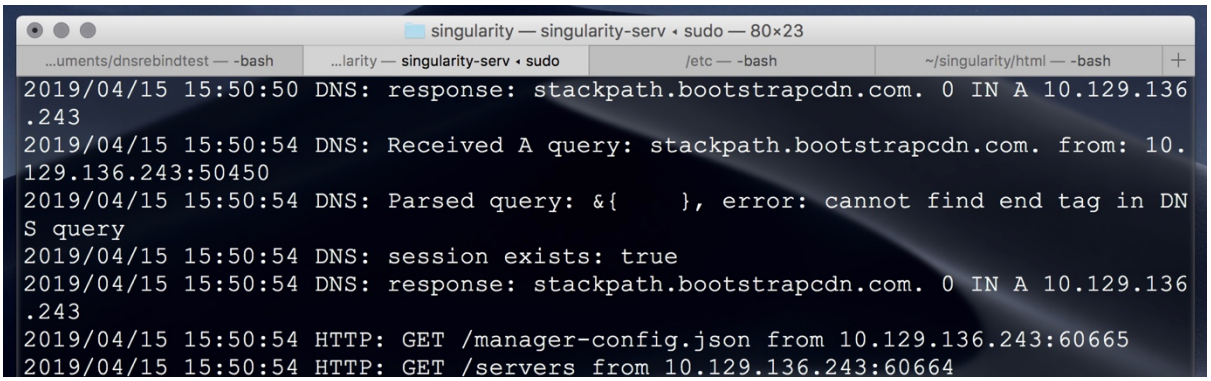
No, kako na internetu postoji mnoštvo web-stranica koje bi korisnik mogao htjeti posjetiti, pamćenje IP adresa za svaku od njih bilo bi veoma neefikasno. Stoga je, još za vrijeme ARPANET-a, uvedena jedinstvena baza podataka koja je mapirala „imena“ web stranica (poslužitelja) i njihove IP adrese. U početku je to bio samo tekstualna datoteka (npr. *host.txt*) koja se nalazila na računalima SRI-ja (*Stanford Research Institute*), no kako se Internet eksponencijalno povećavao, postalo je nemoguće imati jedinstvenu bazu podataka za sve IP adrese. Tako nastaje sistem danas poznat kao DNS (engl *Domain Name System*).

1.2. DNS – Domain Name System

Glavni zadatak DNS-a jeste prevođenje lako pamtljivih domenskih imena (npr. *example.com*) u IP adrese poslužitelja (npr. *192.168.0.34*) potrebne za lociranje servisa i resursa na cijelom internetu. Domenska imena sastavljaju se prema unaprijed određenoj hijerarhiji: na najdesnijoj strani nalazi se tzv. *top-level* domena (npr. *.com, .org, .net*), dok se prema lijevo hijerarhijska važnost smanjuje. Na isti način funkcioniraju i DNS serveri zasluženi za prevođenje domenskih imena: na vrhu je korijenski nameserver koji poslužuje top-level domene ili prosljeđuje zahtjev DNS serveru zaduženom za određenu poddomenu.



Primjera radi, kada pokušamo pristupiti *example.com*, naš web preglednik će zatražiti IP adresu za *example.com* od DNS servera zaduženog za *.com* domene. Postoje dvije daljnje mogućnosti: ako taj DNS server posjeduje *A record* za *example.com*, odnosno ako zna na kojoj adresi se nalazi *example.com*, server će odgovoriti sa tom adresom. Ako pak korijenski DNS server ne posjeduje *A record,* on prosljeđuje query na *nameserver* specificiran u *NS record*-u za tu domenu. Pošto je taj *nameserver* zadužen za tu domenu, on će sigurno posjedovati *A record* za *example.com* te će ga preko korijenskog DNS servera proslijediti korisniku.



1.3. Web preglednici i same-origin policy

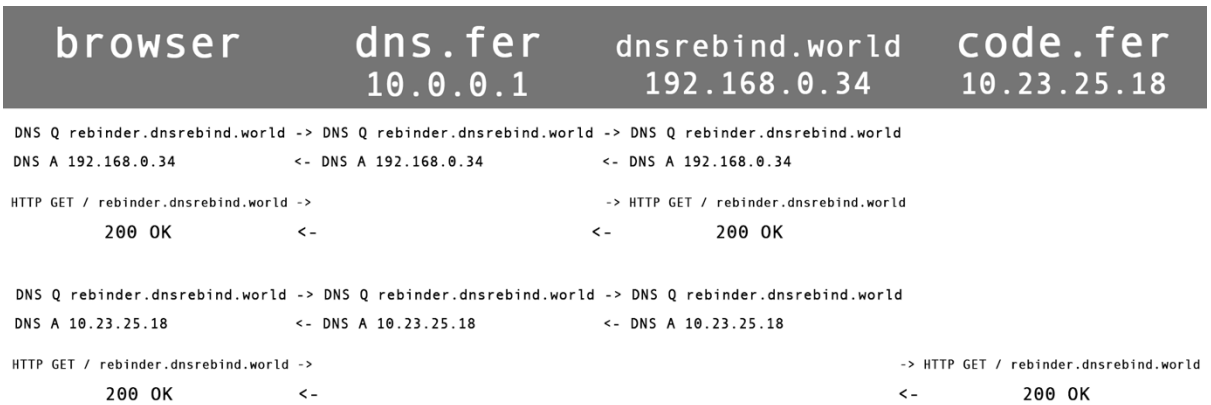
Web preglednici (browseri) su programi pomoću kojih korisnik može prikazivati sadržaj web stranica, kako tekstualni, tako i multimedijalni. Ono što je nama u ovom slučaju najzanimljivije jeste tzv. *same-origin policy.* Da bi današnje web stranice funkcionirale kako je predviđeno, obično moraju učitavati sadržaje sa više poslužitelja. Tako se, na primjer, osnovni sadržaj stranice *example.com* nalazi na serverima *example.com*-a, dok se reklame ili video sadržaj učitava sa neke druge adrese.

Također, danas postoje dinamični dijelovi web stranica poput aplikacija u JavaScriptu ili Flashu koje, i nakon što je web stranica učitana, mogu slati zahtjeve za resurse koji su im potrebni. Da bi se osigurala sigurnost od napada poput XSS-a (engl. *Cross-site scripting*), ti sadržaji mogu slati zahtjeve samo na adresu sa koje su i sami učitani.

Uzmimo slijedeću situaciju kao primjer: korisnik se prijavi na web stranicu svoje banke *banka.hr* i u odgovoru servera banke dobije „kolačić“ (engl. *Cookie*) koji identificira tog korisnika tako da, kad idući put pristupi istoj stranici šaljući sa zahtjevom i dobiveni kolačić, nema potrebe za ponovnom autentifikacijom. Ako taj korisnik na drugom „tab-u“ web preglednika otvori *sumnjivo.hr* na kojoj postoji JavaScript kod *napad.js*, ta skripta bi bez problema mogla pristupiti podacima na *banka.hr*.

Iz tog razloga se uvodi same-origin policy koji osigurava da *napad.js* ne može slati zahtjeve na *banka.hr*, odnosno da može slati zahtjeve samo na *sumnjivo.hr*.

Problemi nastaju u implementaciji *same-origin policy*-ja. Pošto ista domena može pokazivati na više IP adresa (radi *load-balancing*-a kod stranica koje dobijaju velike broje zahtjeva), za provjeru „origin-a“ se ne koristi IP adresa već domensko ime. Implikacije toga istražit ćemo u slijedećem poglavlju.

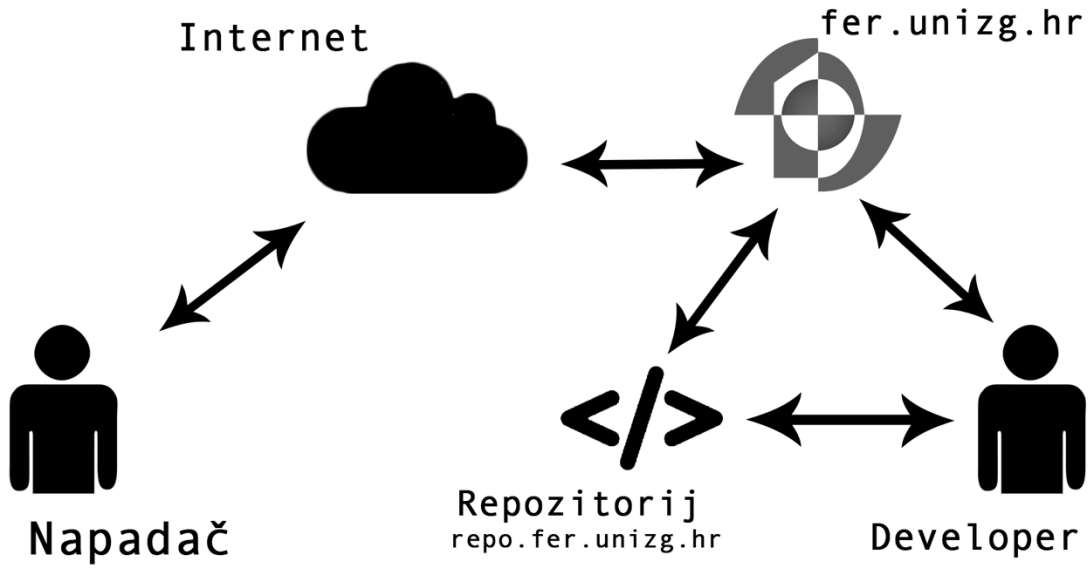


2. DNS Rebinding

Napad poznat kao *DNS Rebinding* je vrsta napada kojom se pokušava zaobići *same-origin policy.* Pošto *SOP* koristi domenska imena za prepoznavanje „origin-a“, ovaj napad iskorištava DNS da bi ga zaobišao. Da bi napad bio izvediv, napadač mora imati domenu (npr. *napadac.hr*) preko koje će ga izvoditi te DNS server zadužen za tu domenu. Kada korisnik pokuša posjetiti *napadac.hr*, napadačev DNS server dobit će DNS upit (engl. *query*) od korisnika. Na prvi query server će odgovoriti sa stvarnom adresom stranice *napadac.hr* sa koje će korisnik učitati (vrlo vjerovatno maliciozni) sadržaj poput *payload.js,* ali će server odgovoriti sa jako kratkim TTL-om (engl. *Time To Live,* koliko dugo će korisnikov browser čuvati odgovor u svojoj cache memoriji).

Nakon isteka zadanog TTL-a, odnosno kad browser „zaboravi“ IP adresu *napadac.hr*-a, maliciozna skripta *payload.js* poslat će novi zahtjev na *napadac.hr*, ali pošto je TTL istekao, browser će morati poslati novi DNS query. Ovog puta, napadačev DNS server odgovara sa drugačijom IP adresom, primjerice lokalnom adresom korisnika (127.0.0.1 ili 0.0.0.0) ili nekom drugom adresom za koju znamo da korisnik ima pristup. Na taj način, skripta dobija pristup svemu na IP adresi iz drugog odgovora, tj. u slučaju da je to 127.0.0.1, skripta ima kontrolu nad lokalnom mrežom korisnika.

2.1. Primjer – FER Intranet



Zamislimo mrežu opisanu gornjim dijagramom. Za pristup repozitoriju nije potrebna dodatna autentifikacija jer joj se jedino može pristupiti ako je korisnik već na

intranetu. Naš developer je “unutar” mreže (na intranetu) te ima pristup nekim bitnim podacima u repozitoriju. Ako developer posjeti web stranicu napadača, zbog same-origin policy-ja skripte na napadačevoj stranici neće imati pristup FERovom intranetu. No ako napadač ima kontrolu nad DNS serverom za svoju web stranicu i poznaje topologiju mreže na kojoj se developer nalazi, koristeći DNS rebinding napadač može dobiti pristup repozitoriju i podacima u njemu bez da developer zna što se dogodilo.

2.2. Implementacijski problemi

- Jedna od najbitnijih karakteristika kod analize napada jeste način na koji se širi. Postoje različite vrste distribucije napada. Kod nekih je dovoljno da korisnik bude dio neke mreže, dok drugi zahtjevaju više “truda”. Za uspjeh DNS rebindinga potrebno je uvjeriti žrtvu da klikne na link koji će pokrenuti napad. Postoje različite tehnike poput korištenja reklama ili slanja “phishing” e-mailova za postizanje tog cilja.
- U primjeru sa FERovim intranetom vidjeli smo da je napadač morao poznavati internu strukturu mreže koju pokušava napasti (točnije IP adresu stranice na koju pokušava dobiti pristup). Pored toga, morao je znati da za pristup repozitoriju nije bila potrebna dodatna autentifikacija. To uveliko smanjuje šanse za uspjeh DNS rebindinga, ali ipak nije nemoguće. Korištenjem generičnih imena za host-ove (npr. ***code****.randomcompany.org*) ili objavljivanjem dijagrama mreže na GitHubu ugrožava se sigurnost mreže u pitanju.

2.3. Zašto nije popravljeno?

Iako je DNS rebinding poprilično star napad, idalje ne postoji kompletno rješenje. Razlog tome jeste činjenica da mnogi internetski servisi koriste tehnike slične DNS rebinding-u da bi izvele sasvim legitimne radnje. Jedan od primjera toga jeste glazbena “streaming” platforma *Spotify.* Kod pokretanja Spotify-evog web-playera ili kod prebacivanja sa web-playera u browseru na player u desktop aplikaciji desi se DNS rebinding napad.

Kad bi se u potpunosti uklonila mogućnost DNS rebindinga, mnogi servisi ne bi mogli funkcionisati kako je predviđeno, te u globalu DNS rebinding donosi više koristi. Ono što je bitno primjetiti jeste da DNS rebinding nije problem web preglednika, tj. da se ne bi trebao sprječavati promjenama u web preglednicima. Neka od mogućih rješenja bila bi zaštita servera putem validacije Host headera te blokiranjem “vanjskih” domena koje se prevedu u interne IP adrese pomoću “firewall”-a.

3. Zaključak

Na osnovu svega što smo vidjeli, dalo bi se zaključiti da je DNS rebinding neophodno zlo. No iako je rješavanje problema ovog napada veoma teško, postoje koraci koje možemo poduzeti da se zaštitimo. Dodavanje autentifikacije na što više servisa neophodan je prvi korak. Spomenuta verifikacija Host headera također je veoma korisna. Naime, ako pogledamo Host header legitimnog zahtjeva, obično ćemo pronaći 127.0.0.1, a u slučaju DNS napada nešto slično *rebinder.dnsrebind.world.* Filtriranjem takvih zahtjeva te korištenjem TLS-a (Transport Layer Security) povećava se sigurnost od DNS rebindinga.

Literatura

Brannon Dorsey, Attacking Private Networks from the Internet with DNS Rebinding
David Murphy, Prevent DNS Rebinding Attacks by Adjusting Your Router
Luke Young, There's no place like 127.0.0.1: Achieving Reliable DNS Rebinding
https://github.com/nccgroup/singularity