HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Napredne DNS Cache Poisoning tehnike

CCERT-PUBDOC-2003-04-15

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

U ovom dokumentu opisana je problematika sigurnosti DNS sustava (engl. ***D**omain **N**ame **S**ystem*) s obzirom na tzv. *DNS cache poisoning* napade. Iako su poznate različite tehnike *DNS cache poisoning* napada, sve one imaju jednaki cilj¸- lažiranje DNS zapisa koji će neovlaštenom korisniku omogućiti neautorizirani dolazak do mrežnog prometa drugih korisnika.
Opisani su osnovni principi napada zajedno s primjerima i odgovarajućim grafičkim prikazima. Također je priložena i usporedna analiza sigurnosnih karakteristika poznatijih DNS poslužitelja s obzirom na *DNS cache poisoning* napade te opasnosti koje prijete Internet korisnicima od istih.

2. Kratka povijest *DNS Cache Poisoning* ranjivosti

1993. godine, Christoph Schuba javno je objavio rad pod nazivom "*Addressing Weaknesses in the Domain Name System*", (http://ftp.cerias.purdue.edu/pub/papers/christoph-schuba/schuba-DNS-msthesis.pdf). Osim osnovnih pojmova i načela rada DNS sustava, dokument ukazuje i na niz sigurnosnih nedostataka unutar ***D**omain **N**ame **S**ystem* protokola. Jedna, od tada prvi put spomenutih tehnika, bila je *DNS Cache Poisoning* metoda, koja je omogućavala umetanje lažnih DNS zapisa u *cache* memoriju ranjivog DNS poslužitelja.
U svojoj prvoj verziji, *DNS Cache Poisoning* tehnika bazirala se na jednostavnom umetanju lažiranih zapisa unutar legitimnih DNS paketa, koje bi ranjivi DNS poslužitelj bez ikakvih provjera pohranio u svoju *cache* memoriju. Mogućnost popunjavanja *cache* memorije DNS poslužitelja proizvoljnim zapisima tada je prvi puta ukazala na ozbiljne propuste unutar DNS sustava.
1997. godine CERT (http://www.cert.org) objavljuje sigurnosnu preporuku pod oznakom CA-1997-22 koja opisuje ranjivost `BIND` (engl. ***B**erkeley **I**nternet **D**omain **N**ame*) DNS poslužitelja na *DNS Cache Poisoning* napade. Sigurnosni rizik objavljene ranjivosti bio je iznimno velik budući su u to vrijeme gotovo svi DNS poslužitelji na Internetu koristili upravo `BIND` programski paket.
Problem je bio vrlo jednostavan. `BIND` poslužitelj nije koristio slučajne vrijednosti *Transaction ID* polja za identifikaciju pojedinih konekcija, već su se nove vrijednosti dobivale sekvencijalnim povećavanjem prethodnih. Budući da je, osim izvorišne i odredišne IP adrese te izvorišnog i odredišnog porta, *Transaction ID* polje jedini element koji `BIND` poslužitelj koristi za identifikaciju pojedinih konekcija, opisani propust omogućavao je relativno jednostavno provođenje *DNS Cache Poisoning* napada. Slanjem pažljivo osmišljenih kombinacija lažiranih DNS upita i pripadajućih lažiranih odgovora bilo je moguće *cache* memoriju ranjivog DNS poslužitelja popuniti malicioznim DNS zapisima. Kako bi se uklonili opisani propusti, kod sljedećih verzija `BIND` DNS poslužitelja dodano je slučajno generiranje *Transaction ID* polja.
Pet godina poslije, 2002. godine, Vagner Sacramento otkrio je nove sigurnosne nedostatke u implementaciji DNS protokola kod `BIND` poslužitelja. Tada provedene analize pokazale su da će `BIND` poslužitelj pri primanju `n` upita za razrješavanje istog DNS imena, proslijediti istih `n` upita nadređenim DNS poslužiteljima kako bi došao do odgovarajućeg odgovora. Iako bi za razrješavanje primljenog upita teoretski bio dovoljan jedan jedini upit, tadašnja inačica `BIND` poslužitelja je u tu svrhu nepotrebno koristila višestruke DNS konekcije, ovisno o broju upita klijenata.
Iako na prvi pogled ozbiljnost problema ne dolazi dovoljno do izražaja, detaljnije analize pokazale su da isti olakšava provođenje *DNS cache poisoning* napada i do 1000 puta. S obzirom na prirodu problema, na njegovu analizu moguće je primijeniti poznati matematički fenomen poznat pod imenom "*Birthday Paradox*", čiji će detaljniji opis i matematički smisao biti opisan u jednom od narednih poglavlja.
Povezivanjem ovog problema s istraživanjem Michala Zalewskog, vezanog uz analizu načina generiranja sekvencijalnih brojeva TCP/IP protokola kod različitih operacijskih sustava ("*Strange Attractors and TCP/IP Sequence Number Analysis*", http://razor.bindview.com/publish/papers/tcpseq.html), CERT je ukazao na potencijalne probleme sličnog karaktera i kod `BIND` DNS poslužitelja.
Istraživanje Michala Zalewskog pokazalo je da se kod većine operacijskih sustava uz relativno jednostavne analize može s priličnom pouzdanošću predvidjeti buduće vrijednosti sekvencijalnih brojeva TCP paketa. Mogućnost predikcije ovih vrijednosti danas se smatra ozbiljnim sigurnosnim

nedostatkom, budući da isti čini implementacije TCP/IP stoga ranjivim na brojne napade (*Man In The Middle* i sl.).
CERT smatra da se isti problem može vrlo jednostavno primijeniti i na `BIND` DNS poslužitelj, što će biti opisano u nastavku ovog dokumenta.

3. *Birthday Attack*

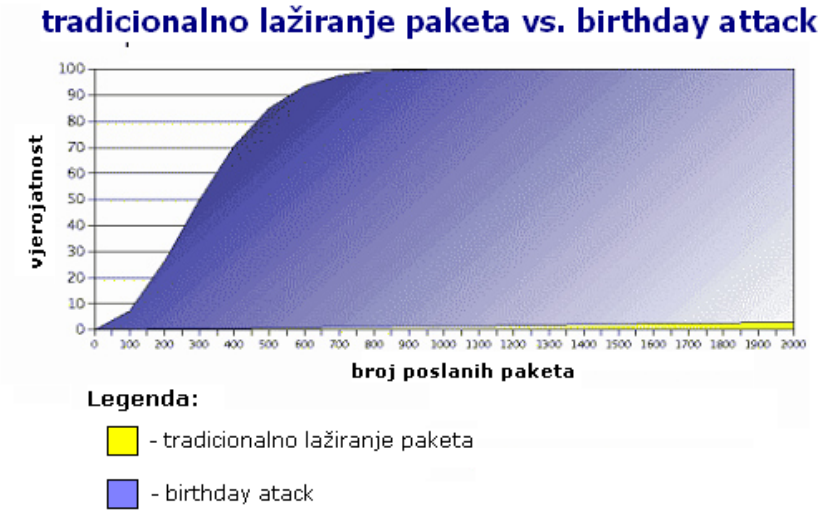
Iako se u ovom slučaju odnosi na DNS servis, *Birthday Attack* je općeniti pojam koji se u terminologiji računalne sigurnosti odnosi na skupinu *brute-force* napada sličnog karaktera. Pojam je preuzet iz teorije vjerojatnosti, gdje *birthday paradox* pravilo kaže da je vjerojatnost da dvoje ili više ljudi u skupini od 23 imaju rođendan na isti dan jednaka 50 ili više posto.
Općenita definicija teorema kaže da ukoliko neka funkcija kao ulaz prima slučajnu vrijednost i na izlazu daje jednu od `k` jednako vjerojatnih vrijednosti, tada će se ista vrijednost na izlazu pojaviti nakon $1.2\sqrt{k}$ ponavljanja. Spomenuti *Birthday paradox* je samo specijalni slučaj ovog teorema gdje je k=365 (broj dana u godini).
Sličan koncept može se primijeniti i na pseudo-slučajno generiranje *Transaction ID* polja kod `BIND` DNS poslužitelja. Kod klasičnog lažiranja DNS paketa (engl. *DNS spoofing*) napadač na jedan lažirani upit šalje `n` lažiranih odgovora. Vjerojatnost da u takvom scenariju napadač pogodi valjanu vrijednost *Transaction ID* polja, koja je neophodna za uspješno provođenje napada, je `n/65535`. Za jedan poslani paket ova vjerojatnost iznosi 0.0015%, što daje vrlo male izglede da se napad uspješno realizira. Za razliku od upravo opisanog tradicionalnog lažiranja paketa, *birthday attack* bazira se na slanju `n` lažiranih DNS odgovora za `n` prethodno lažiranih DNS upita.
Ovakvim pristupom vjerojatnost uspješnog provođenja napada znatno raste i moguće ju je matematički izraziti sljedećim izrazom:

$$p = 1 - \left(1 - \frac{1}{t}\right)^{\frac{n \times n\ (n-1)}{2}}$$

gdje je

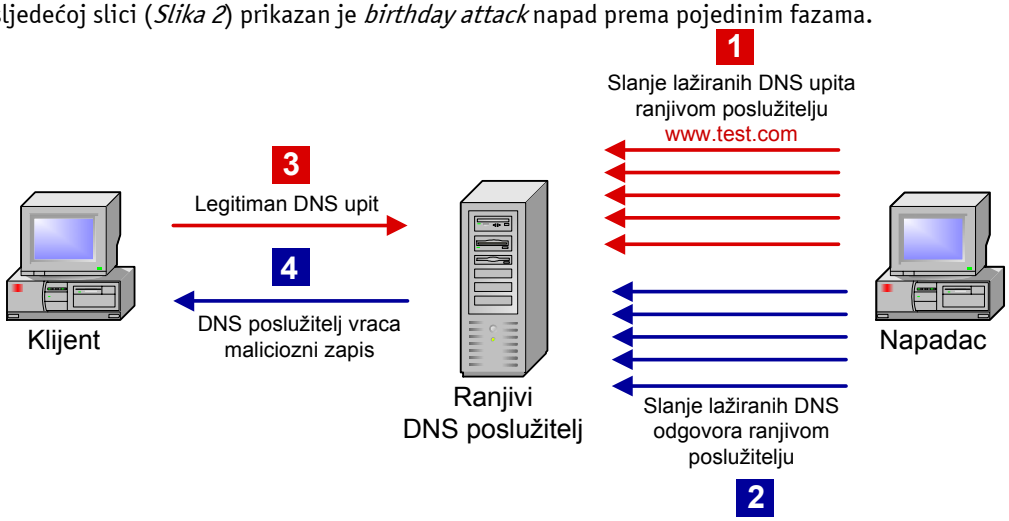
- `t` – ukupni broj mogućih vrijednosti;
- `n` – broj lažiranih paketa.

Međusobni odnos između tradicionalnog i *birthday attack* lažiranja DNS paketa, prikazan je grafičkim prikazom na sljedećoj slici (*Slika 1*).



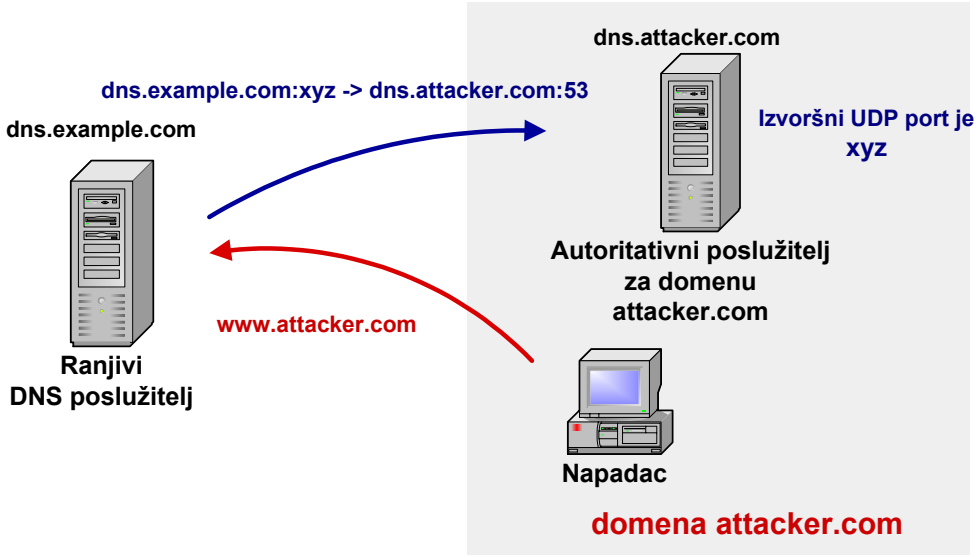
Slika 1: Tradicionalno lažiranje paketa i birthday attack

Kao što se može vidjeti sa priložene slike, *birthday attack* metoda daje neusporedivo bolje rezultate u odnosu na tradicionalne metode lažiranja paketa. Primjenom *birthday attack* metode već oko 700 lažiranih paketa dovoljno je za vjerojatnost uspješnog provođenja napada od 100%, dok će istih 700 paketa tradicionalnom metodom dati vjerojatnost od 1.07% (700/65535). Nagib krivulje također pokazuje da je već oko 300 paketa dovoljno za uspješnost provođenja napada s vjerojatnošću od 50%, što olakšava provođenja napada neovlaštenim korisnicima sa "sporijim" pristupom Internetu.
Na sljedećoj slici (*Slika 2*) prikazan je *birthday attack* napad prema pojedinim fazama.



Slika 2: Birthday attack

U prvom koraku napadač ranjivom poslužitelju šalje nekoliko stotina lažiranih DNS zahtjeva za razrješavanjem imena domene koja se želi kompromitirati, npr. www.test.com (korak 1). Nedugo zatim napadač šalje jednaki broj pripadajućih lažiranih DNS odgovora kojima se pokušava lažirati odgovor DNS poslužitelja autoritativnog za domenu čije se ime razrješava, u ovom primjeru test.com (korak 2).
Za uspješno provođenje napada, vrlo je važno da lažirani DNS odgovori izgledaju kao da zaista dolaze od autoritativnog DNS poslužitelja od kojega se očekuje odgovor za inicirani upit. Ukoliko niti jedan od lažiranih paketa ne odgovara upitu, DNS poslužitelj će odbaciti pakete i napad neće uspjeti.
Prilikom lažiranja paketa treba voditi računa da odgovaraju izvorišna i odredišna IP adresa, zatim izvorišni i odredišni port, i naravno *Transaction* ID polje. Lažiranje IP adresa ne predstavlja poseban problem, budući da su napadaču poznate IP adrese napadnutog i DNS poslužitelja autoritativnog za zonu čije se ime nastoji kompromitirati.
Nešto je veći problem s UDP portovima, budući da je potrebno predvidjeti koji će izvorišni port napadnuti DNS poslužitelj koristiti prilikom iniciranja upita prema autoritativnom DNS poslužitelju. Iako je poznato da je odredišni port rekurzivnog DNS upita 53, za uspješno provođenje napada potrebno je poznavati i upravo spomenuti izvorišni UDP port. Pri određivanju izvorišnog UDP porta pomaže činjenica da `BIND` DNS poslužitelj gotovo uvijek koristi isti izvorišni UDP port za razrješavanje upita koji dolaze od istog klijenta.
Ukoliko napadač ima pristup autoritativnom poslužitelju za domenu s koje provodi napad, dolazak do moguće vrijednosti izvorišnog UDP porta više nije problem. Napadač ranjivom DNS poslužitelju šalje upit za razrješavanje imena računala koje se nalazi unutar domene s koje se provodi napad (*Slika 3*). Napadnuti DNS poslužitelj u svrhu razrješavanja upita, prosljeđuje zahtjev autoritativnom DNS poslužitelju (onome kojemu napadač ima pristup) za domenu u pitanju. Jednostavnom analizom mrežnog prometa napadač može doći do izvorišnog porta koji će napadnuti DNS poslužitelj vrlo vjerojatno koristiti za razrješavanje budućih DNS upita primljenih od strane istog klijenta.



Slika 3: Određivanje izvorišnog porta za lažiranje DNS paketa kod birthday attack

Kao potvrda činjenice da `BIND` DNS poslužitelj vrlo često koristi isti izvorišni port za razrješavanje upita koji dolaze od istog klijenta, priložen je dio mrežnog prometa zabilježenog `tcpdump` programom za analizu i praćenje mrežnog prometa. Na sljedećem primjeru pokazano je kako `BIND` poslužitelj za razrješavanje različitih upita od strane istog klijenta uzastopno koristi isti izvorišni port (33748).

```
10:54:12.423228 192.168.1.2.33748 > 66.218.71.63.53:  21345 [1au] A?
www.yahoo.com.
10:54:21.313293 192.168.1.2.33748 > 216.239.38.10.53:  53735 [1au]
A? www.google.com.
10:54:27.182852 192.168.1.2.33748 > 149.174.213.7.53:  19315 [1au]
A? www.netscape.com.
10:54:43.252461 192.168.1.2.33748 > 66.35.250.11.53:  43129 [1au] A?
www.linux.com.
```

Ovo je samo još jedan primjer sigurnosnog nedostataka koji na prvi pogled nema poseban značaj, ali u dubljem kontekstu sa stanovišta sigurnosti može imati vidljive posljedice. Kada bi se izvorišni portovi odabirali slučajno, vjerojatnost uspješnog provođenja napada bila bi znatno manja.
Konačno, napadač mora osigurati da njegov lažirani DNS paket do napadnutog poslužitelja stigne prije od legitimnog paketa autoritativnog DNS poslužitelja. U tu svrhu moguće je pokrenuti neki blaži oblik napada uskraćivanjem računalnih resursa koji će usporiti vrijeme odgovora ciljnog (autoritativnog) DNS poslužitelja.
Nakon što DNS poslužitelj prihvati lažirani DNS zapis, isti će u *cache* memoriji biti pohranjen vrijeme definirano TTL (engl. ***T**ime **T**o **L**ive*) poljem unutar DNS paketa.

4. Fazno prostorna analiza

Prema CERT-ovom dokumentu koji opisuje sigurnosni nedostatak `BIND` poslužitelja vezan za iniciranje višestrukih konekcija za isti upit, navodi se kako se ranije spomenuto istraživanje Michala Zalewskog može vrlo jednostavno primijeniti i na ovu problematiku.
U svome radu Zelewski koristi pojam fazno prostorne analize (engl. *Phase space analysis*), gdje se pod faznim prostorom podrazumijeva `n`-dimenzionalni prostor koji opisuje stanje sustava sa `n` varijabli. U sklopu analize također se koristi i pojam atraktora (engl. *Attractor*), oblika koji je specifičan za pojedinu pseudo-slučajnu (PRNG) funkciju i koji otkriva kompleksnu prirodu ovisnosti između pojedinih rezultata dobivenih tom funkcijom.
Koristeći opisanu metodu moguće je analizirati kvalitetu generiranja slučajnih vrijednosti bilo koje pseudo-slučajne funkcije koja se koristi u tu svrhu. Budući da vjerojatnost uspješnog provođenja

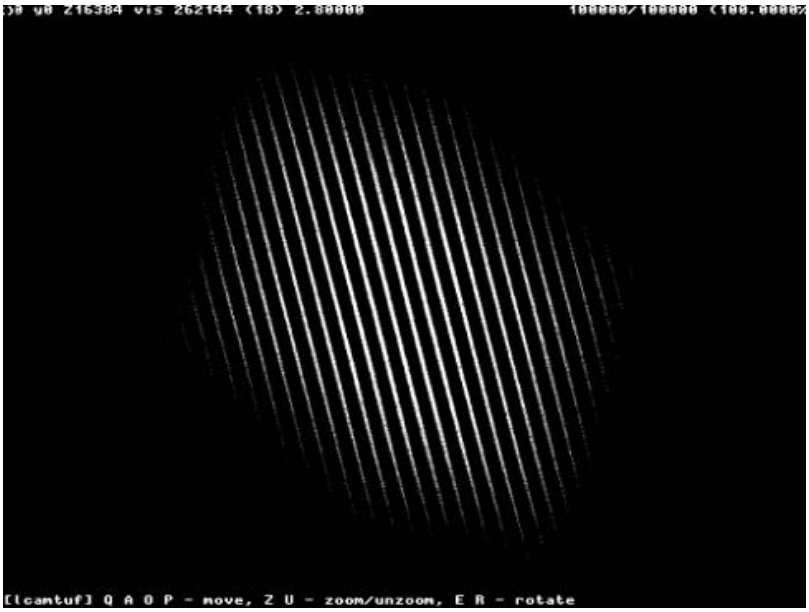
birthday attack uvelike ovisi o mogućnosti predviđanja TID identifikacijskog broja, sa stanovišta sigurnosti DNS servisa svakako valja ispitati način njegovog generiranja. Upravo je to bio zaključak CERT-a kada su se u svom dokumentu referencirali na rad Zelewskog, koji, iako usmjeren prema analizi sekvencijalnih brojeva kod TCP/IP protokola, može izvrsno poslužiti i kod analize ovog problema.
Kao dio istraživanja objavljen je i skup alata koji olakšavaju analizu i interpretaciju dobivenih rezultata. Radi se o sljedećim programima:

- `vseq` – program baziran na linux `svgalib` biblioteci, koji omogućuje vizualizaciju podataka u trodimenzionalnom prostoru, i
- `calprob` – program koji pokušava predvidjeti vjerojatnost pogađanja sljedeće izlazne vrijednosti PRNG funkcije na temelju tri zadnje vrijednosti.

Primjeri korištenja ovih programa dani su u nastavku dokumenta.
U nastavku dokumenta analizirane su karakteristike PRNG funkcija koje se koriste pri generiranju slučajnih *Transaction ID* vrijednosti kod `BIND 8`, `BIND 9` i `djbdns` DNS poslužitelja. U svrhu analize korišteni je upravo spomenuti `vseq` alat.

4.1. BIND 8

U nastavku je dan grafički prikaz dobiven pokretanjem `vseq` programa nad slučajnim vrijednostima TID polja kod `BIND 8` DNS poslužitelja.



Slika 4: Analiza PRNG funkcije za generiranje TID broja kod BIND 8 DNS poslužitelja

Da bi se omogućila jednostavnija interpretacija dobivenih prikaza treba napomenuti kako bi pokretanje `vseq` programa nad izlaznim vrijednostima idealnog PRNG generatora slučajnih brojeva rezultiralo prikazom s potpuno jednoliko distribuiranim oblakom. Bilo koja pravilnost unutar dobivenog grafičkog prikaza (engl. *Attractor*) upućuje na determinističku komponentu PRNG generatora, a samim time i na mogućnosti predikcije budućih vrijednosti.
Gornji prikaz dobiven je analizom vrijednosti TID polja kod 8.4.3 inačice DNS poslužitelja. Paralelne linije jasno upućuju da su neke vrijednosti kod PRNG generatora kod ove inačice BIIND poslužitelja više vjerojatne od drugih. Ukoliko se lažiranje paketa ograniči na TID vrijednosti koje se prema ovakvoj analizi smatraju vjerojatnijima, vjerojatnost uspješnog provođenja napada raste.
Koristeći `calprob` program moguće je na temelju dobivenog prikaza i pripadajućih atraktora analizirati vjerojatnosti uspješnog pogađanja sljedeće izlazne vrijednosti PRNG generatora. Program prihvaća četiri argumenta: datoteku sa slučajno generiranim vrijednostima, radijus analize R1, broj lažiranih paketa i ukupan broj mogućih vrijednosti. Rezultat pokretanja `calprob` programa nad TID vrijednošću `BIND 8.4.9` poslužitelja prikazana je u nastavku:

```
# ./calprob ../bind.nsid.gz 10 1 1
 1: Trying 13085 32386 21499 (-> 36312) - r=10
   + guess3d gave 51 answers, rsort suggests R2 of 0...
   -> SUCCESSFUL (difference 0).

 Data file:          ../bind.nsid.gz
 Failed attempts:    0/1 (0%)
 Average R2:         0
 Average N:          51
 Average error:      0
 Average correct N:  3
 Probability:        100.0000%
```

Bez dubljih razmatranja dobivenih rezultata može se zaključiti da algoritam predviđa vjerojatnost pogađanja budućih vrijednosti PRNG generatora opisanog slikom od 100%, uz svega tri poznate prethodne vrijednosti.
Budući da je ova analiza potpuno nezavisna u odnosu na ranije opisani *birthday attack* napad, može se zaključiti kako su 8.4.3 inačice BIND poslužitelja ranjive na *Cache Poisoning* napade, čak i ako su instalirane sigurnosne zakrpe koje uklanjaju sigurnosne nedostatke vezane uz *birthday attack* napad.

4.2. BIND 9

U odnosu na inačicu 8 `BIND` poslužitelja, inačica 9 koristi potpuno novi generator slučajnih brojeva. Generator koristi `/dev/random` uređaj dostupan na većini modernih Linux/Unix platformi. Provedene analize pokazale su da `/dev/random` uređaj uz zadovoljavajuću entropiju omogućuje kvalitetno generiranje slučajnih vrijednosti. Naravno, treba uzeti u obzir i razlike u implementaciji između pojedinih operacijskih sustava. U ovom primjeru korištenje Linux distribucija s 2.4.19 jezgrom.
Pokretanje `vseq` programa nad vrijednostima PRNG generatora `BIND 9` poslužitelja dalo je rezultate prikazane sljedećom slikom (*Slika 5*).

Slika 5: Analiza PRNG funkcije za generiranje TID broja kod BIND 9 DNS poslužitelja

`Calprob` program dao je sljedeće rezultate:

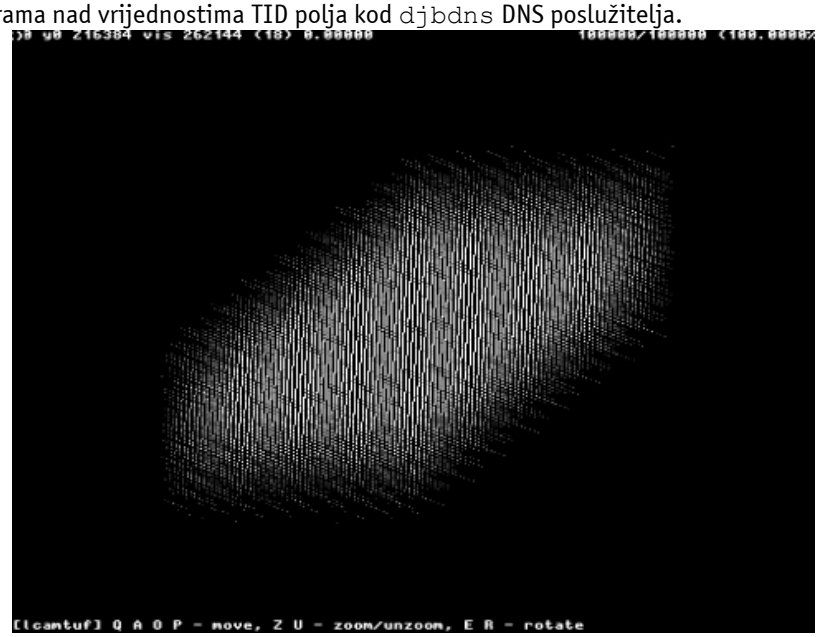
```
./calprob ../bind9.nsid.gz 200 5000 10
...
Data file:          ../bind9.nsid.gz
Failed attempts:    0/10 (0%)
```

```
Average R2:         75
Average N:          522
Average error:      12
Average correct N:  24
Probability:        20.0000%
```

Provedena analiza pokazuje da je buduće vrijednosti PRNG generatora kod BIND 9 poslužitelja moguće predvidjeti s vjerojatnošću od 20% uz 5000 lažiranih paketa. Iako su rezultati znatno bolji nego kod BIND 8 poslužitelja, napadaču se još uvijek ostavlja dovoljno prostora za provođenje napada.
Slanje 5000 paketa zahtjeva nešto brži pristup ciljnom DNS poslužitelju, a i napadaču se u određenoj mjeri otežava utrka s autoritativnim DNS poslužiteljem. Vrlo često neovlašteni korisnici u ovakvim situacijama koriste napade uskraćivanjem računalnih resursa kako bi se povećala učinkovitost provođenja napada.
Kombinirajući fazno prostornu analizu i napade uskraćivanjem računalnih resursa neovlaštenom korisniku omogućuje se pouzdano provođenje napada uz vrlo teško detekciju istoga.

4.3. Djbdns

U ovom poglavlju iznesena je analiza PRNG generatora kod djbdns DNS poslužitelja, alternativnog rješenja popularnom BIND poslužitelju. *Slika 6* prikazuje 3D grafički prikaz dobiven pokretanjem vseq programa nad vrijednostima TID polja kod djbdns DNS poslužitelja.



Slika 6: Analiza PRNG funkcije za generiranje TID broja kod djbdns DNS poslužitelja

U odnosu na BIND poslužitelj u ovom slučaju moguće je primijetiti puno manje pravilnosti unutar dobivenog prikaza. Iako se na prvi pogled PRNG generator djbdns poslužitelja čini kvalitetnijim u odnosu na BIND 9 poslužitelj, calprob program pokazuje drugačije. Uz iste parametre kao i kod BIND 9 poslužitelja rezultati su sljedeći:

```
./calprob ../djbdns.nsid.gz 200 5000 10
...
 Data file:         ../djbdns.nsid.gz
 Failed attempts:   0/10 (0%)
 Average R2:        33
 Average N:         623
 Average error:     22
 Average correct N: 31
 Probability:       30.0000%
```

Može se primijetiti da je uz isti broj lažiranih paketa (5000) vjerojatnost uspješnog provođenja napada 30%, 10% više nego što je to slučaj kod `BIND 9` poslužitelja.
Kao prednost `djbdns` poslužitelja u odnosu na `BIND` treba spomenuti podatak da `djbdns` ne koristi iste izvorišne portove za razrješavanje upita primljenih od istog klijenta. Ovakav pristup napadaču uvelike otežava provođenje napada, bez obzira što je PRNG generator dao nešto slabije rezultate.
Primjer slučajnog odabira portova priložen je u nastavku:

```
22:42:41.790753 192.168.1.2.16075 > 64.58.77.85.53:  36904 A? www.yahoo.com.
22:42:53.876719   192.168.1.2.53928   >   216.239.38.10.53:      1776   A?
www.google.com.
22:43:07.996666   192.168.1.2.59368   >   207.200.73.80.53:      16261   A?
www.netscape.com.
22:43:18.290976 192.168.1.2.9183 > 66.35.250.10.53:  5110 A? www.linux.com.
```

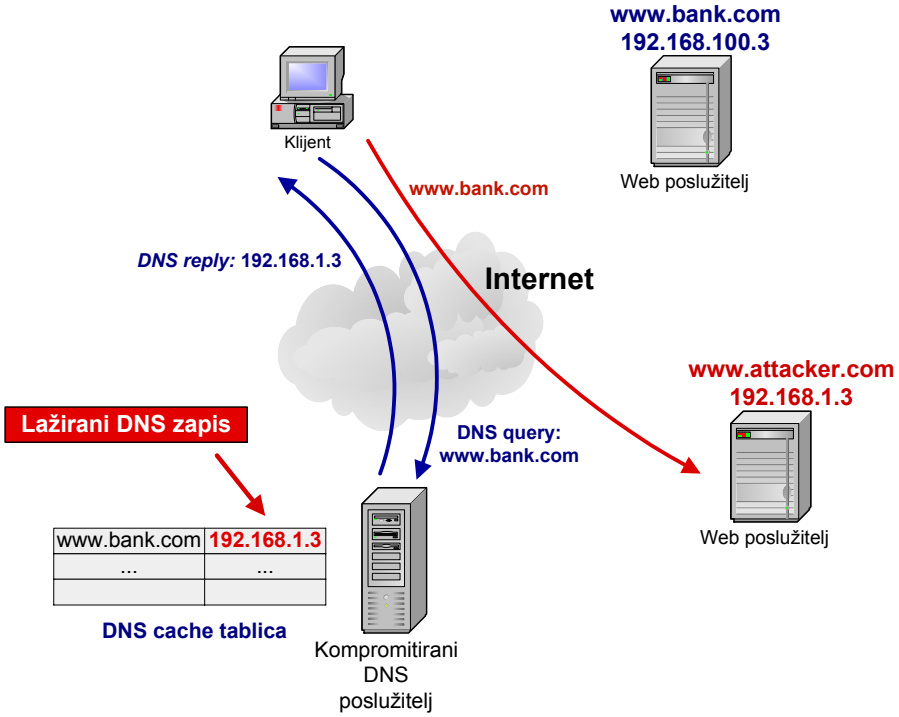
Kombiniranjem slučajnog generiranja izvorišnog porta sa zadovoljavajućim karakteristikama PRNG generatora postignut je zadovoljavajući nivo zaštite od *DNS Cache Poisoning* napada kod `djbdns` poslužitelja.

5. Opasnosti DNS Cache Poisoning ranjivosti

DNS Cache Poisoning napad sam po sebi nema poseban značaj i gotovo uvijek se koristi kao priprema za provođenje nekog drugog napada, na neki od servisa koji se bazira na DNS sustavu. S obzirom na važnost DNS sustava mogući su različiti scenarij, od kojih su neki opisani u nastavku poglavlja.

5.1. Preusmjeravanje Web prometa

Provođenje *DNS Cache Poisoning* napada s ciljem preusmjeravanja Web prometa može imati različite posljedice, ovisno o prioritetu i važnosti prometa koji se preusmjerava. Ukoliko se radi o bankovnim transakcijama ili nekom drugom prometu visokog sigurnosnog rizika, tada posljedice mogu biti katastrofalne kako na strani klijenta, tako i na strani davatelja usluga.
Osnovna ideja preusmjeravanja Web prometa pomoću DNS *Cache Poisoning* metode je da se kreira Web *site* što sličniji onome koji se želi kompromitirati putem lažiranja DNS zapisa. Pod pojmom "što sličniji", smatra se onaj nivo sličnosti koji kod korisnika neće izazvati sumnju da se radi o nelegitimnim Web stranicama.
U sljedećem koraku se jednom od ranije opisanih *DNS Cache Poisoning* metoda lažira DNS zapis koji će sve upite za razrješavanjem imena legitimnog Web poslužitelja preusmjeriti na IP adresu Web poslužitelja neovlaštenog korisnika, na kojem se nalazi lažirana kopija legitimnih Web stranica (*Slika 7*). Opisani napad još je poznat i pod imenom engl. *Domain Hijacking*.

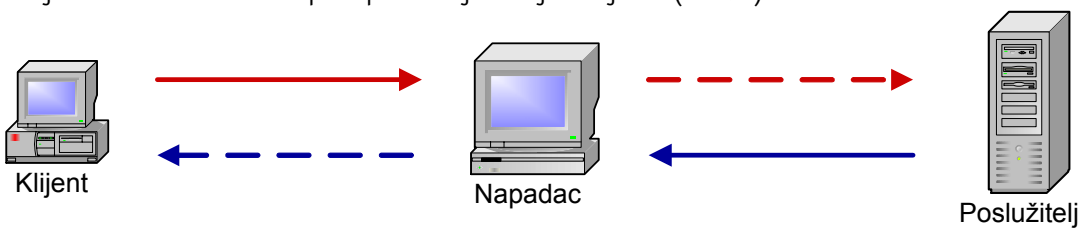


Slika 7: Preusmjeravanje Web prometa putem DNS Cache Poisoning metode

Ukoliko je sadržaj legitimnog Web poslužitelja vjerno kopiran i ukoliko je napad pažljivo osmišljen, vrlo je velika vjerojatnost da korisnik na klijentskom računalu neće primijetiti nikakvu razliku. Vrijeme koje je lažirani DNS zapis pohranjen u *cache* memoriji kompromitiranog DNS poslužitelja neovlaštenom korisniku je dovoljno da prikupi povjerljive informacije, koje je kasnije moguće iskoristiti u druge maliciozne svrhe (npr. korisničke zaporke, brojevi kreditnih kartica, itd...).
Na sličan način moguće je preusmjeriti i bilo koji drugi promet koji se bazira na DNS servisu (Mail, FTP, ...).

5.2. *Man in The Middle* napad

Kod *Man in The Middle* napada neovlašteni korisnik presreće komunikaciju između dvije točke i analizira promet koji se između njih razmjenjuje. Nakon presretanja i analize prometa napadač mora održavati komunikaciju s obje točke kako iste ne bi primijetile prekid veze.
Primjer *Man In The Middle* napada prikazan je na sljedećoj slici (*Slika 8*).



Slika 8 Man In The Middle napad

Presretanje i lažiranje TCP konekcija (engl. *Session hijacking*) nije nimalo trivijalan problem, budući da napadač na neki način mora osigurati da prvi prima pakete namijenjene ciljnom računalu. Ukoliko to ne bi bio slučaj napadač ne bi bio u mogućnosti presresti željenu komunikaciju.
Jedna od tehnika koja u tome može pomoći je upravo *DNS cache poisoning*. Lažiranjem DNS zapisa napadač može željeni promet preusmjeriti na svoju IP adresu nakon čega je moguće održavati komunikaciju s drugom stranom. Analizom presretnutog prometa moguće je doći do povjerljivih podataka koje klijent i poslužitelj razmjenjuju.

6. Mogućnosti zaštite

U ovom poglavlju biti će opisane mogućnosti zaštite od *DNS Cache Poisoning* napada sa različitih stajališta. S obzirom da lažiranje DNS zapisa ima različite utjecaje na pojedine segmente DNS sustava, mogućnosti zaštite će također ovisiti o kojem se dijelu sustava radi. Internet klijenti, administratori DNS poslužitelja i vlasnici domena imaju različite uloge u DNS sustavu pa će i njihove odgovornosti biti nešto drugačije.

Općenito gledano, kod provođenja napada koji se baziraju na lažiranju DNS zapisa postoje dvije mete napada. Prva je domena čije je ime lažirano i čiji se promet preusmjerava na adresu neovlaštenog korisnika, a druga je krajnji korisnik, koji je najčešće žrtva napada.

Za vlasnike domena (engl. *Domain owners*) vrlo je malo moguće postići u smislu zaštite od kompromitiranja imena njihove domene. Kod Web poslužitelja svakako se preporučuje korištenje SSL (engl. ***S**ecure **S**ocket **L**ayer*) protokola koji omogućuje međusobnu autentikaciju Web klijenta i poslužitelja. Korištenje SSL protokola posebno je važno u slučajevima kada se Web servisom razmjenjuju povjerljivi korisnički podaci kao što su zaporke, brojevi kreditnih kartica i sl., budući da SSL protokol, osim autentikacije sudionika, omogućuje i enkripciju podataka.

Iako u razvoju, kao jedan od mogućih načina zaštite od *DNS Cache Poisoning* napada je korištenje *DNSSec* (engl. ***DNS Sec**urity*) dodataka DNS protokolu. Radi se o skupu dodataka koji su dodani DNS protokolu kako bi se osigurao integritet i autentičnost podataka koji se razmjenjuju između DNS poslužitelja. Više informacija o *DNS Security* projektu moguće je naći na adresi http://www.dnssec.net/.

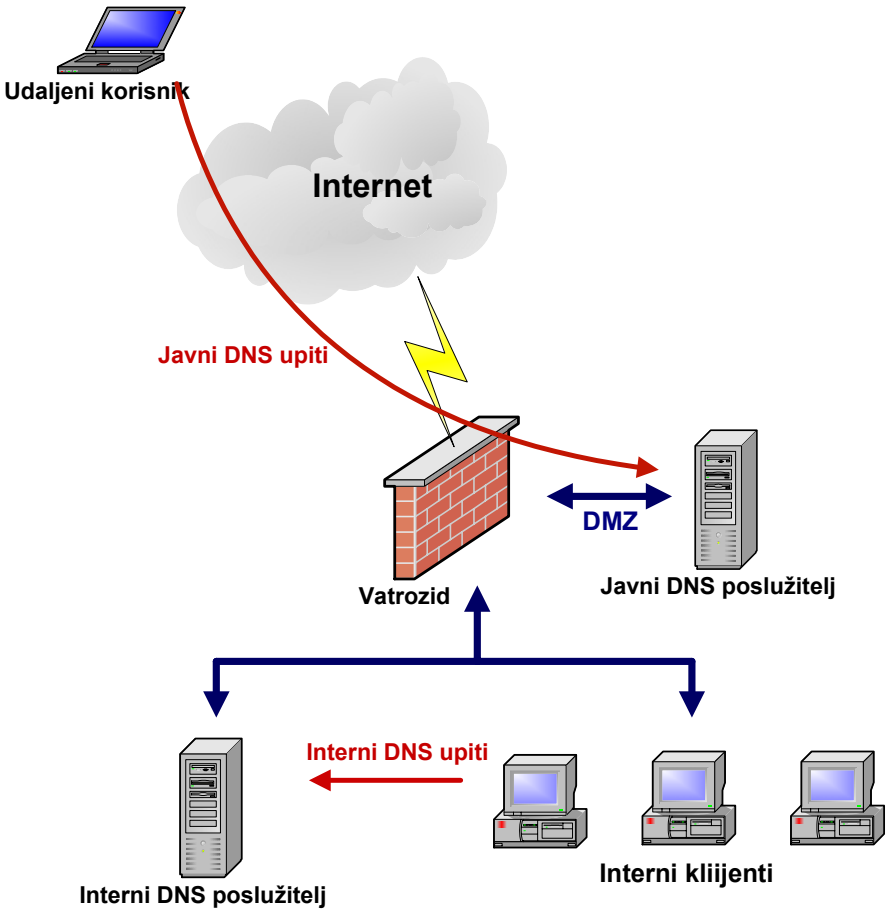
Čak i ukoliko napadač uspješno lažira DNS zapis neke druge domene, vlasnici iste vrlo će teško uvidjeti problem budući da nije vezan uz njihov DNS poslužitelj. Napad kojim neovlašteni korisnik lažira tuđa imena domene, usmjeren je prema DNS poslužitelju koji nije direktno pod nadležnosti domene o čijem se imenu radi, već prema DNS poslužitelju kojeg upotrebljava legitimni korisnik.

Kao možebitni pokazatelj mogu biti učestali napadi uskraćivanjem računalnih resursa kojim napadač želi usporiti procesiranje DNS upita. Iskusniji mrežni administrator može ovakve aktivnosti u određenim okolnostima uspješno povezati s *DNS Cache Poisoning* napadom.

Administratorima DNS poslužitelja preporučuje se redovita administracija DNS poslužitelja. Pod redovitom instalacijom smatra se periodičko provjeravanje sigurnosnih upozorenja i redovita instalacija sigurnosnih zakrpi, kao i instaliranje najnovijih inačica korištenog programskog paketa. Iako je `BIND` poslužitelj danas najrasprostranjeniji DNS poslužitelj, u razmatranje se može uzeti i korištenje nekog drugog programskog paketa. Kao primjer može se spomenuti `djbdns` DNS poslužitelj, čija je analiza dana u poglavlju 4.3.

Treba spomenuti kako `djbdns` DNS poslužitelj dolazi s garancijom o sigurnosti programa, uz novčanu nagradu svakome tko uspije pronaći sigurnosne nedostatke unutar programa. Iako se nagrada odnosi na napade prepisivanjem spremnika (engl. *Buffer overflow*), a ne na *DNS Cache Poisoning* napade opisane u ovom dokumentu, analize su pokazale da `djbdns` program ima zaista dobra svojstva sa stanovišta sigurnosti.

Kao dodatna mjera zaštite, preporučuje se onemogućavanje procesiranja rekurzivnih upita vanjskim korisnicima te korištenje tzv. *split-split* DNS arhitekture prikazane na sljedećoj slici (*Slika 9*).



Slika 9 Split-split DNS arhitektura

Split-split DNS struktura podrazumijeva korištenje dva DNS poslužitelja: jedan za posluživanje vanjskih korisnika i jedan za razrješavanje rekurzivnih upita internih korisnika. U ovakvoj arhitekturi javni DNS poslužitelj nikako ne bi smio dozvoljavati izvršavanje rekurzivnih upita, dok bi interni DNS poslužitelj trebao na odgovarajući način biti zaštićen varozidom.

Split-split DNS arhitekturu treba razlikovati u odnosu na čistu *split* arhitekturu gdje interni DNS poslužitelj sve upite prosljeđuje javnom DNS poslužitelju u demilitariziranoj zoni (engl. *Demilitarized zone - DMZ*), koji dalje razrješava upite. *Split* arhitektura ne pruža nikakvu zaštitu od *DNS Cache Poisoning* napada i sa stanovišta sigurnosti DNS servisa svakako je neprihvatljiva.

Ukoliko postoje ograničenja u pogledu implementacije *split-split* arhitekture, procesiranje rekurzivnih upita svakako treba ograničiti samo na lokalne korisnike.

Uključivanje `allow-recursion` opcije kod `BIND` poslužitelja u ovom slučaju neće u potpunosti riješiti problem, budući da napadač ionako lažira IP adresu DNS paketa te mu je za provođenje napada dovoljno poznavanje IP adresa kojima je dozvoljeno procesiranje rekurzivnih upita. Kao dodatni nivo zaštite moguće je upotrijebiti `listen-on` opciju, koja će DNS poslužitelj povezati samo s mrežnim sučeljem koja nisu dostupna putem javnog Interneta.

Proizvođačima, odnosno programerima DNS poslužitelja, preporučuje se posvećivanje posebne pažnje sigurnosnim svojstvima njihovih proizvoda. Upotreba naprednih i pouzdanih tehnika generiranja slučajnih brojeva (PRNG generatori) i izbjegavanje korištenja istog izvorišnog porta prilikom iniciranja upita samo su neki od primjera kojima se mogu poboljšati sigurnosna svojstva postojećih DNS poslužitelja. U bliskoj budućnosti svakako bi trebalo razmotriti i mogućnosti autentikacije i provjere integriteta pojedinih DNS upita, kako bi se izbjegli problemi lažiranja DNS konekcija. Budući da DNS upiti trenutno koriste UDP protokol, njihovo lažiranje ne predstavlja velik problem.

7. Zaključak

Dokument opisuje problematiku *DNS Cache Poisoning* napada, jednog od ozbiljnijih problema DNS sustava. Opisani su sigurnosni nedostatci pojedinih implementacija DNS protokola zajedno s pripadajućim pojašnjenima. Također su pojašnjeni sigurnosni rizici koji povlače mogućnost lažiranja DNS zapisa te mogućnosti zaštite sa različitih stanovišta Internet korisnika.