



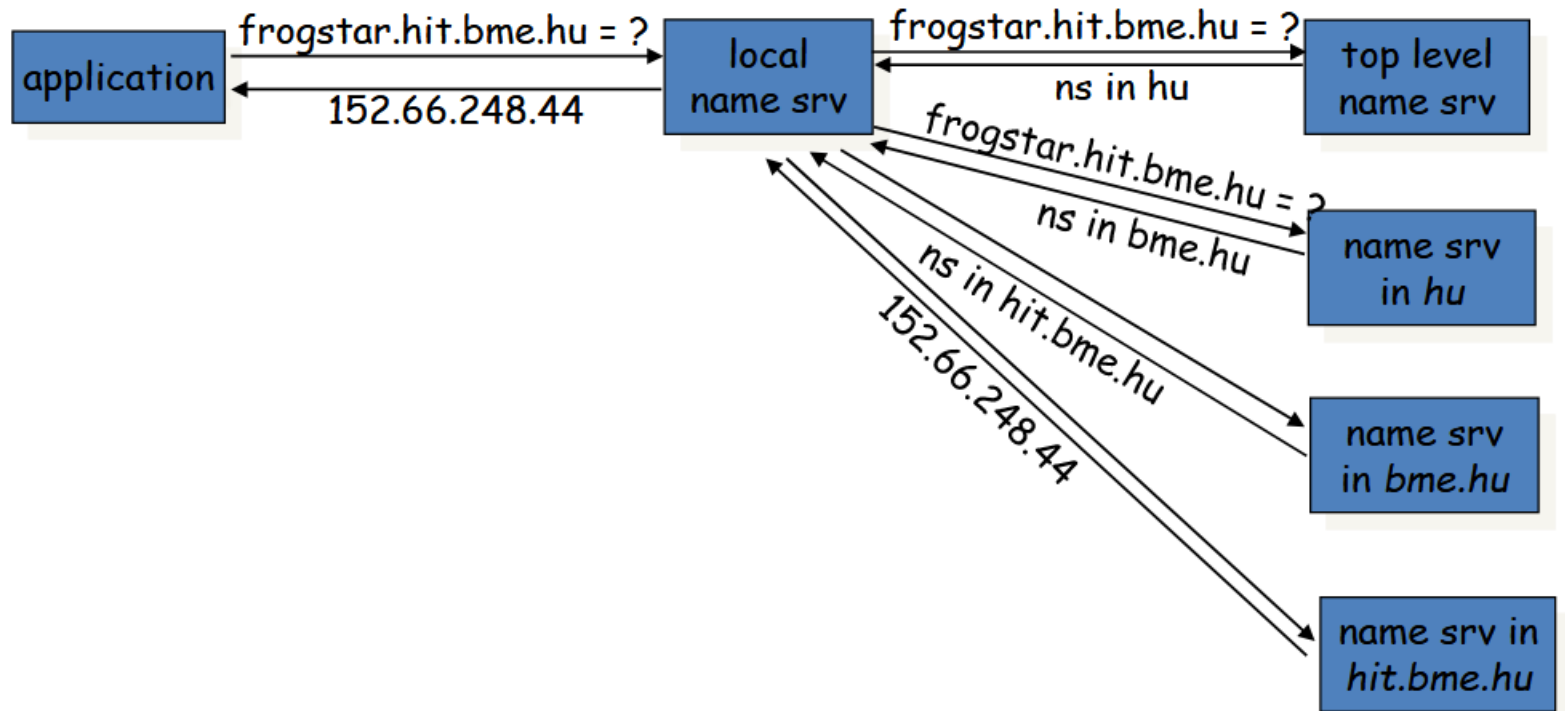
DNS Security

Boldizsár Bencsáth, PhD, OSCP - Tamás Holczer, PhD
Laboratory of Cryptography and System Security
Department of Networked Systems and Services
{bencsath,holczer}@CrySyS.hu

DNS – Domain Name System

- The DNS is a distributed database that provides mapping between hostnames and IP addresses
- the DNS name space is hierarchical
 - top level domains: com, edu, gov, int, mil, net, org, ae, ..., hu, ... zw
 - top level domains may contain second level domains
e.g., bme within hu, epfl within ch, ...
 - second level domains may contain third level domains, etc.
- each domain has name servers
 - usually (not always) a name server knows the IP address of the top level name servers
 - if a domain contains sub-domains, then the name server knows the IP address of the sub-domain name servers
 - when a new host is added to a domain, the administrator adds the (hostname, IP address) mapping to the database of the local name server

DNS Operations



- Single DNS reply may include several mappings
- Received information is cached by the name server
- `/etc/nsswitch.conf`, `/etc/hosts` `/etc/resolv.conf`

Simple configuration

- /etc/resolv.conf (static /dhcp / systemd)
 nameserver 152.66.248.12
 nameserver 152.66.249.12
- /etc/hosts
 127.0.0.1 localhost
 127.0.1.1 shamir-ng.crysys.hu shamir-ng
 10.105.0.152 ldap.crysys.hu
 10.105.0.157 ldap-2.crysys.hu
- /etc/nsswitch.conf
 hosts: files dns
 ...

Bind / named tld

Named.conf:

```
zone "." {  
    type hint;  
    file "/etc/bind/db.root";  
};
```

. = everybody = . zone

hint = last resort

```
.....  
/etc/bind/db.root  
.....
```

```
; This file holds the information on root name servers needed to  
; initialize cache of Internet domain name servers  
; (e.g. reference this file in the "cache . <file>"  
; configuration file of BIND domain name servers).  
;
```

```
; This file is made available by InterNIC  
; under anonymous FTP as  
; file /domain/named.root  
; on server FTP.INTERNIC.NET  
;-OR- RS.INTERNIC.NET  
;
```

```
; last update: Feb 04, 2008  
; related version of root zone: 2008020400  
;  
; formerly NS.INTERNIC.NET
```

```
...  
; operated by RIPE NCC  
;
```

```
. 3600000 NS K.ROOT-SERVERS.NET.  
K.ROOT-SERVERS.NET. 3600000 A 193.0.14.129  
K.ROOT-SERVERS.NET. 3600000 AAAA 2001:7fd::1  
...
```

Resource Record of Root server

Zone TTL Type Value

DNS definitions

- DNS „**Zone**“, “zone file”: The „domain“, a bunch of records related to the domain. Subdomains might be defined separately -> another zone, but same domain, or included in the domain definition -> same zone, same domain
- **Name server, DNS server** (e.g. bind, djbdns), a program that serves DNS data to the clients and communicates with other servers
- **Authoritative nameserver**: the nameserver who knows the authoritative, genuine information about a domain. Other nameservers might have a copy (cache), but the origin of the information is always coming from authoritative name servers
- **Primary/secondary nameserver**: There might be multiple nameservers. From the client's point of view, it's no matter which server to ask, generally they are used in a round-robin fashion. Generally the origin of the data is the primary nameserver, all the other authoritative servers (secondaries, and later caching servers) receive information from the primary server. Zone transfer to synchronize secondary from primary.

Definitions #2

- **Resolver:** the client who wants to “resolve” a name into an IP address.
- **Recursive DNS server:** a DNS server that will give a usable answer to a client even if the actual query tries to resolve a DNS name and the server is not authoritative for that name. It will recursively (from the root servers) resolve the domain and send back to the client
- **Forwarder DNS server:** forwards the whole request to another server
- **Resource record:** a single entry in the domain “zone file”, e.g. an “A” record is an internet address for a specific host, an “MX” record is the mail exchanger definition for a domain.
- **Delegation:** If the authoritative server does not want to be responsible for every information, it can delegate a specific part of the domain (e.g. a subdomain, a reverse-dns subnet) to a different DNS server

Simple DNS request

- `root@tuzfalmerescit:~# dig www.bme.hu @localhost`
- `; <<>> DiG 9.5.1-P3 <<>> www.bme.hu @localhost`
- `;; global options: printcmd`
- `;; Got answer:`
- `;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59313`
- `;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 0`
- `;; QUESTION SECTION:`
- `www.bme.hu. IN A`
- `;; ANSWER SECTION:`
- `www.bme.hu. 14400 IN A 152.66.115.35`
- `;; AUTHORITY SECTION:`
- `bme.hu. 14400 IN NS ns.bme.hu.`
- `bme.hu. 14400 IN NS ns2.pantel.net.`
- `bme.hu. 14400 IN NS nic.bme.hu.`
- `;; Query time: 4364 msec`
- `;; SERVER: 127.0.0.1#53(127.0.0.1)`
- `;; WHEN: Wed May 5 22:19:44 2010`
- `;; MSG SIZE rcvd: 107`



A simple recursive query

ans/auth/hint

22:19:43.579620 IP 10.105.1.40.43129 > 193.0.14.129.53: 36649 [1au] A? www.bme.hu. (39)
22:19:43.615802 IP 193.0.14.129.53 > 10.105.1.40.43129: 36649- 0/9/9 (504)
22:19:43.634634 IP 10.105.1.40.53377 > 193.0.14.129.53: 24520 [1au] NS? . (28)
22:19:43.684849 IP 193.0.14.129.53 > 10.105.1.40.53377: 24520*- 14/0/21 NS a.root-servers.net.,[domain]

K.Root
UDP rand port

22:19:43.883583 IP 10.105.1.40.43480 > 193.6.16.1.53: 59363 [1au] A? www.bme.hu. (39)
22:19:43.930598 IP 193.6.16.1.53 > 10.105.1.40.43480: 59363- 0/3/5 (190)

.hu. ?

22:19:44.479631 IP 10.105.1.40.60577 > 192.5.5.241.53: 12566% [1au] A? ns2.pantel.net. (43)
22:19:44.503629 IP 10.105.1.40.32234 > 152.66.115.1.53: 25786 [1au] A? www.bme.hu. (39)
22:19:44.503696 IP 10.105.1.40.29619 > 192.5.5.241.53: 46648% [1au] AAAA? ns2.pantel.net. (43)
22:19:44.551306 IP 152.66.115.1.53 > 10.105.1.40.32234: 25786* 1/3/6 A 152.66.115.35 (222)
22:19:44.590675 IP 192.5.5.241.53 > 10.105.1.40.60577: 12566- 0/15/16 (711)
22:19:44.590702 IP 192.5.5.241.53 > 10.105.1.40.29619: 46648- 0/15/16 (711)

22:19:44.634724 IP 10.105.1.40.38747 > 192.26.92.30.53: 51577% [1au] A? ns2.pantel.net. (43)
22:19:44.687762 IP 10.105.1.40.47813 > 192.26.92.30.53: 25972% [1au] AAAA? ns2.pantel.net. (43)
22:19:44.746422 IP 192.26.92.30.53 > 10.105.1.40.38747: 51577- 0/2/3 (107)
22:19:44.775192 IP 192.26.92.30.53 > 10.105.1.40.47813: 25972- 0/2/3 (107)
22:19:44.823264 IP 10.105.1.40.64788 > 212.24.160.1.53: 5666% [1au] A? ns2.pantel.net. (43)
22:19:44.875791 IP 212.24.160.1.53 > 10.105.1.40.64788: 5666*- 1/2/2 A 212.24.160.1 (107)
22:19:44.879600 IP 10.105.1.40.16055 > 212.24.160.1.53: 24986% [1au] AAAA? ns2.pantel.net. (43)
22:19:44.911290 IP 212.24.160.1.53 > 10.105.1.40.16055: 24986*- 0/1/1 (103)

23:09:37.957083 IP 10.105.1.97.17843 > 199.7.91.13.53: 63544 [1au] A? bme.hu. (35)
23:09:37.958685 IP 10.105.1.97.35470 > 192.228.79.201.53: 61901 [1au] NS? . (28)
23:09:38.060477 IP 199.7.91.13.53 > 10.105.1.97.17843: 63544- 0/9/13 (600)
23:09:38.061413 IP 10.105.1.97.48638 > 193.239.148.48.53: 12453 [1au] A? bme.hu. (35)
23:09:38.063879 IP 193.239.148.48.53 > 10.105.1.97.48638: 12453- 0/3/5 (186)
23:09:38.064840 IP 10.105.1.97.25858 > 152.66.115.1.53: 33889 [1au] A? bme.hu. (35)
23:09:38.066248 IP 152.66.115.1.53 > 10.105.1.97.25858: 33889* 1/3/6 A 152.66.115.203 (218)
23:09:38.066589 IP 10.105.1.97.33431 > 192.33.4.12.53: 48806% [1au] AAAA? ns2.pantel.net. (43)
23:09:38.067593 IP 10.105.1.97.19470 > 128.63.2.53.53: 36151% [1au] A? ns2.pantel.net. (43)
23:09:38.090028 IP 192.33.4.12.53 > 10.105.1.97.33431: 48806- 0/15/16 (735)
23:09:38.091050 IP 10.105.1.97.19489 > 192.5.6.30.53: 63379% [1au] AAAA? ns2.pantel.net. (43)
23:09:38.145268 IP 192.228.79.201.53 > 10.105.1.97.35470: 61901*- 14/0/23 NS b.root-servers.net., NS l.root-servers.net., NS i.root-servers.net., NS d.root-servers.net., NS h.root-servers.net., NS g.root-servers.net., NS m.root-servers.net., NS k.root-servers.net., NS a.root-servers.net., NS c.root-servers.net., NS j.root-servers.net., NS e.root-servers.net., NS f.root-servers.net., RRSIG (857)
23:09:38.197162 IP 128.63.2.53.53 > 10.105.1.97.19470: 36151- 0/15/16 (735)
23:09:38.197896 IP 10.105.1.97.65050 > 192.52.178.30.53: 19095% [1au] A? ns2.pantel.net. (43)
23:09:38.219579 IP 192.52.178.30.53 > 10.105.1.97.65050: 19095- 0/6/3 (606)
23:09:38.220239 IP 10.105.1.97.34661 > 62.77.203.11.53: 50127% [1au] A? ns2.pantel.net. (43)
23:09:38.221862 IP 62.77.203.11.53 > 10.105.1.97.34661: 50127*- 1/0/0 A 212.24.160.1 (48)
23:09:38.230684 IP 192.5.6.30.53 > 10.105.1.97.19489: 63379- 0/6/3 (606)
23:09:38.231373 IP 10.105.1.97.26758 > 213.163.34.67.53: 7331% [1au] AAAA? ns2.pantel.net. (43)
23:09:38.233656 IP 213.163.34.67.53 > 10.105.1.97.26758: 7331*- 0/1/0 (85)

Part 1 of the query

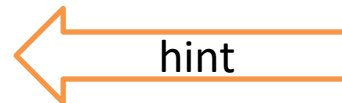
```
root@tuzfalmeresclt:~# dig www.bme.hu @k.root-servers.net in a
```

```
; <<>> DiG 9.5.1-P3 <<>> www.bme.hu @k.root-servers.net in a
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58230
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 8
;; WARNING: recursion requested but not available
```

```
;; QUESTION SECTION:
;www.bme.hu.          IN      A
```

```
;; AUTHORITY SECTION:
hu.      172800 IN      NS      ns.nic.hu.
hu.      172800 IN      NS      ns1.nic.hu.
hu.      172800 IN      NS      ns2.nic.fr.
hu.      172800 IN      NS      ns2.nic.hu.
hu.      172800 IN      NS      ns3.nic.hu.
hu.      172800 IN      NS      ns-se.nic.hu.
hu.      172800 IN      NS      ns-com.nic.hu.
```

```
;; ADDITIONAL SECTION:
ns.nic.hu.      172800 IN      A      193.239.148.62
ns1.nic.hu.     172800 IN      A      193.239.149.3
ns2.nic.fr.     172800 IN      A      192.93.0.4
ns2.nic.hu.     172800 IN      A      193.6.16.1
ns3.nic.hu.     172800 IN      A      195.70.35.250
ns-se.nic.hu.   172800 IN      A      77.72.229.251
ns-com.nic.hu.  172800 IN      A      194.0.1.12
ns.nic.hu.      172800 IN      AAAA   2001:738:4:8000::62
```



Part 2 of the query

```
root@tuzfalmeresclt:~# dig . @k.root-servers.net in ns
```

```
; <<>> DiG 9.5.1-P3 <<>> . @k.root-servers.net in ns
```

```
;; global options: printcmd
```

```
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9443
```

```
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 15
```

```
;; WARNING: recursion requested but not available
```

```
;; QUESTION SECTION:
```

```
;-                IN      NS
```

```
;; ANSWER SECTION:
```

```
.      518400 IN      NS      a.root-servers.net.
.      518400 IN      NS      b.root-servers.net.
.      518400 IN      NS      c.root-servers.net.
.      518400 IN      NS      d.root-servers.net.
.      518400 IN      NS      e.root-servers.net.
.      518400 IN      NS      f.root-servers.net.
.      518400 IN      NS      g.root-servers.net.
.      518400 IN      NS      h.root-servers.net.
.      518400 IN      NS      i.root-servers.net.
.      518400 IN      NS      j.root-servers.net.
.      518400 IN      NS      k.root-servers.net.
.      518400 IN      NS      l.root-servers.net.
.      518400 IN      NS      m.root-servers.net.
```

```
;; ADDITIONAL SECTION:
```

```
a.root-servers.net. 518400 IN      A      198.41.0.4
b.root-servers.net. 518400 IN      A      192.228.79.201
c.root-servers.net. 518400 IN      A      192.33.4.12
d.root-servers.net. 518400 IN      A      128.8.10.90
e.root-servers.net. 518400 IN      A      192.203.230.10
f.root-servers.net. 518400 IN      A      192.5.5.241
g.root-servers.net. 518400 IN      A      192.112.36.4
h.root-servers.net. 518400 IN      A      128.63.2.53
i.root-servers.net. 518400 IN      A      192.36.148.17
j.root-servers.net. 518400 IN      A      192.58.128.30
k.root-servers.net. 518400 IN      A      193.0.14.129
l.root-servers.net. 518400 IN      A      199.7.83.42
m.root-servers.net. 518400 IN      A      202.12.27.33
a.root-servers.net. 518400 IN      AAAA   2001:503:ba3e::2:30
f.root-servers.net. 518400 IN      AAAA   2001:500:2f::f
```

```
;; Query time: 10 msec
```

```
;; SERVER: 193.0.14.129#53(193.0.14.129)
```

```
;; WHEN: Wed May 5 22:25:29 2010
```

```
;; MSG SIZE rcvd: 492
```

Part 3 of the query

```
root@tuzfalmeresc.lt:~# dig www.bme.hu @193.6.16.1 in a
```

```
; <<>> DiG 9.5.1-P3 <<>> www.bme.hu @193.6.16.1 in a
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22160
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 4
;; WARNING: recursion requested but not available
```

```
;; QUESTION SECTION:
;www.bme.hu.                IN      A
```

```
;; AUTHORITY SECTION:
bme.hu.      86400 IN      NS      nic.bme.hu.
bme.hu.      86400 IN      NS      ns.bme.hu.
bme.hu.      86400 IN      NS      ns2.pantel.net.
```

```
;; ADDITIONAL SECTION:
ns.bme.hu.   86400 IN      A      152.66.116.1
ns.bme.hu.   86400 IN      AAAA    2001:738:2001:8001::2
nic.bme.hu.   86400 IN      A      152.66.115.1
nic.bme.hu.   86400 IN      AAAA    2001:738:2001:2001::2
```

```
;; Query time: 8 msec
;; SERVER: 193.6.16.1#53(193.6.16.1)
;; WHEN: Wed May 5 22:28:09 2010
;; MSG SIZE rcvd: 179
```

Part 4 of the query

```
root@tuzfalmerescit:~# dig ns2.pantel.hu @192.5.5.241 in a
```

.hu instead of .net

```
; <<>> DiG 9.5.1-P3 <<>> ns2.pantel.hu @192.5.5.241 in a
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43225
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 8
;; WARNING: recursion requested but not available
```

```
;; QUESTION SECTION:
;ns2.pantel.hu.          IN      A
```

```
;; AUTHORITY SECTION:
```

```
hu.      172800 IN      NS      ns3.nic.hu.
hu.      172800 IN      NS      ns2.nic.fr.
hu.      172800 IN      NS      ns.nic.hu.
hu.      172800 IN      NS      ns-com.nic.hu.
hu.      172800 IN      NS      ns-se.nic.hu.
hu.      172800 IN      NS      ns1.nic.hu.
hu.      172800 IN      NS      ns2.nic.hu.
```

Why?

```
;; ADDITIONAL SECTION:
```

```
ns.nic.hu.      172800 IN      A      193.239.148.62
ns1.nic.hu.     172800 IN      A      193.239.149.3
ns2.nic.fr.     172800 IN      A      192.93.0.4
ns2.nic.hu.     172800 IN      A      193.6.16.1
ns3.nic.hu.     172800 IN      A      195.70.35.250
ns-se.nic.hu.   172800 IN      A      77.72.229.251
ns-com.nic.hu.  172800 IN      A      194.0.1.12
ns.nic.hu.      172800 IN      AAAA   2001:738:4:8000::62
```

```
;; Query time: 119 msec
;; SERVER: 192.5.5.241#53(192.5.5.241)
;; WHEN: Wed May 5 22:29:10 2010
;; MSG SIZE rcvd: 311
```

Part 5 of the query

```
root@tuzfalmeresc.lt:~# dig ns2.pantel.net @212.24.160.1 in a
```

```
; <<>> DiG 9.5.1-P3 <<>> ns2.pantel.net @212.24.160.1 in a
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21461
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1
;; WARNING: recursion requested but not available
```

```
;; QUESTION SECTION:
ns2.pantel.net.          IN      A
```

```
;; ANSWER SECTION:
ns2.pantel.net.         86400  IN      A      212.24.160.1
```

```
;; AUTHORITY SECTION:
pantel.net.             86400  IN      NS      ns1.pantel.net.
pantel.net.             86400  IN      NS      ns2.pantel.net.
```

```
;; ADDITIONAL SECTION:
ns1.pantel.net.         86400  IN      A      212.24.164.1
```

```
;; Query time: 51 msec
;; SERVER: 212.24.160.1#53(212.24.160.1)
;; WHEN: Wed May 5 22:32:10 2010
;; MSG SIZE rcvd: 96
```

Part 6 of the query

```
root@tuzfalmeresclt:~# dig ns2.pantel.net @192.26.92.30 in any

; <<>> DiG 9.5.1-P3 <<>> ns2.pantel.net @192.26.92.30 in any
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61660
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;ns2.pantel.net.                IN      ANY

;; AUTHORITY SECTION:
pantel.net.      172800 IN      NS      ns1.pantel.net.
pantel.net.      172800 IN      NS      ns2.pantel.net.

;; ADDITIONAL SECTION:
ns1.pantel.net.  172800 IN      A      212.24.164.1
ns2.pantel.net.  172800 IN      A      212.24.160.1

;; Query time: 159 msec
;; SERVER: 192.26.92.30#53(192.26.92.30)
;; WHEN: Wed May 5 22:30:14 2010
;; MSG SIZE rcvd: 96
```


Location of the 13 root servers (205 sites) (2023: 12 at 1698 sites)



Anycast request (AS25152 – K-Root)

- Nearest root server: k.root-servers.net in Budapest, BIX: 193.0.14.129

inetnum: 193.0.14.0 - 193.0.15.255

netname: RIPE-NCC-K-ROOT

descr: Subnet for k.root-servers.net

descr: See <http://k.root-servers.org/> for details

country: NL

admin-c: AP110-RIPE

tech-c: RNDS-RIPE

status: ASSIGNED PI

mnt-by: RIPE-DNS-MNT

source: RIPE # Filtered

- **Sites: 18**

Global: 5

Local: 13

London, UK *; **Amsterdam, NL ***; **Frankfurt, DE**; Athens, GR *; Doha, QA; Milan, IT *; Reykjavik, IS *; Helsinki, FI *; Geneva, CH *; Poznan, PL; Budapest, HU *; Abu Dhabi, AE; **Tokyo, JP**; Brisbane, AU *; **Miami, FL, US ***; Delhi, IN; Novosibirsk, RU; Dar es Salaam, TZ

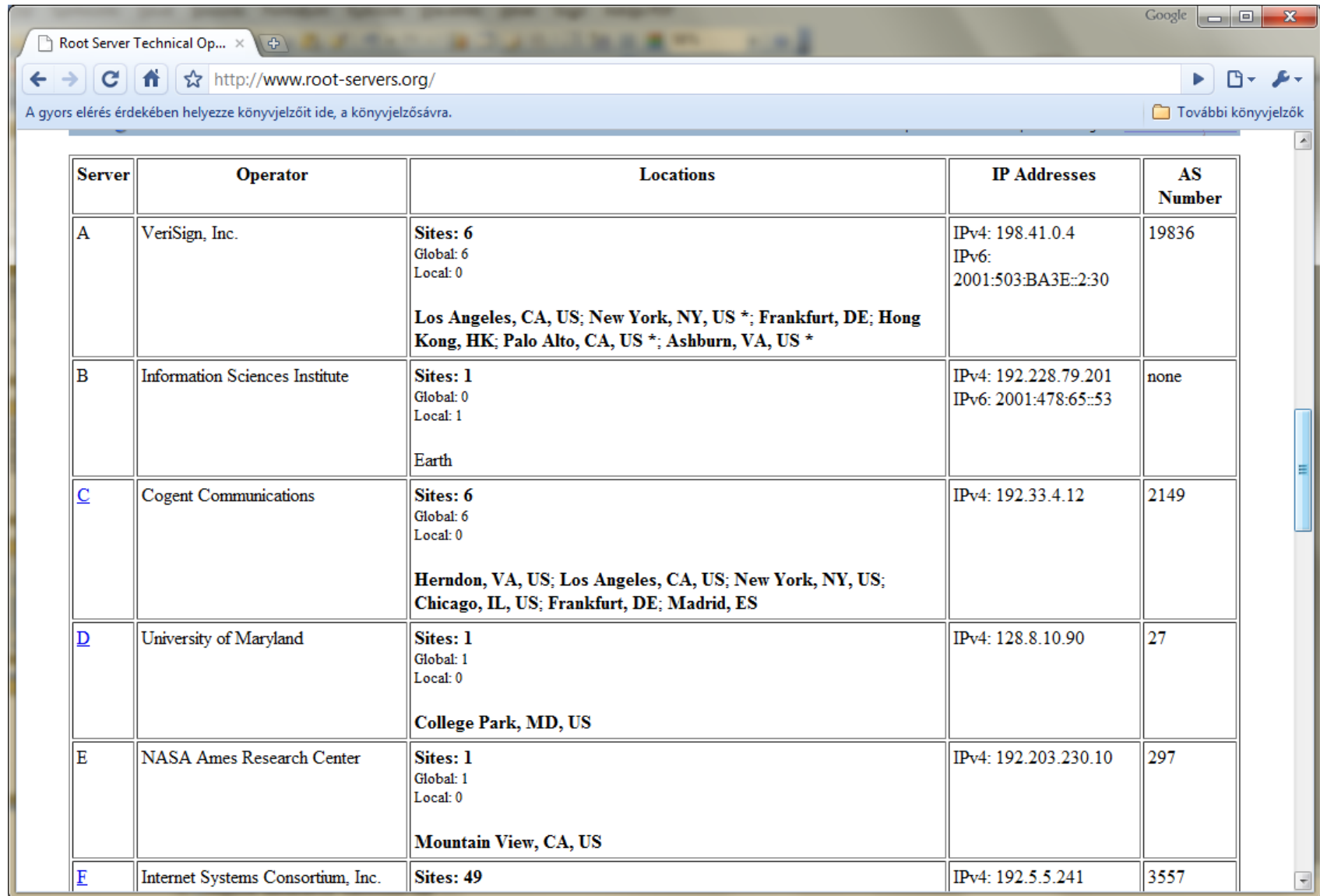
- IPv4: 193.0.14.129

IPv6: 2001:7fd::1

Partial BGP data on AS25152

remarks:	Node	Location	Type	IPv6	Community	Route-Set
remarks:	----	-----	----	----	-----	-----
remarks:	ams-ix	NL, Amsterdam	Global	Yes	25152:1	RS-KROOT-AMS-IX
remarks:	linx	UK, London	Global	Yes	25152:2	RS-KROOT-LINX
remarks:	tokyo	JP, Tokyo	Global	No	25152:3	RS-KROOT-TOKYO
remarks:	nap	US, Miami	Global	Yes	25152:5	RS-KROOT-NAP
remarks:	denic	DE, Frankfurt	Global	No	25152:11	RS-KROOT-DENIC
remarks:	delhi	IN, Delhi	Local	No	25152:4	RS-KROOT-DELHI
remarks:	bix	HU, Budapest	Local	Yes	25152:6	RS-KROOT-BIX
remarks:	mix	IT, Milan	Local	Yes	25152:7	RS-KROOT-MIX
remarks:	ficix	FI, Helsinki	Local	Yes	25152:8	RS-KROOT-FICIX
remarks:	isnic	IS, Reykjavik	Local	Yes	25152:9	RS-KROOT-ISNIC
remarks:	poznan	PL, Poznan	Local	No	25152:10	RS-KROOT-POZNAN
remarks:	cern	CH, Geneva	Local	Yes	25152:12	RS-KROOT-CERN
remarks:	grnet	GR, Athens	Local	Yes	25152:13	RS-KROOT-GRNET
remarks:	qtel	QA, Doha	Local	No	25152:14	RS-KROOT-QTEL
remarks:	nskix	RU, Novosibirsk	Local	No	25152:15	RS-KROOT-NSKIX
remarks:	emix	UA, Abu Dhabi	Local	No	25152:16	RS-KROOT-EMIX
remarks:	apnic	AU, Brisbane	Local	Yes	25152:17	RS-KROOT-APNIC
remarks:	tix	TZ, Dar es Sal.	Local	No	25152:18	RS-KROOT-TIX

Root servers



Server	Operator	Locations	IP Addresses	AS Number
A	VeriSign, Inc.	Sites: 6 Global: 6 Local: 0 Los Angeles, CA, US; New York, NY, US *; Frankfurt, DE; Hong Kong, HK; Palo Alto, CA, US *; Ashburn, VA, US *	IPv4: 198.41.0.4 IPv6: 2001:503:BA3E::2:30	19836
B	Information Sciences Institute	Sites: 1 Global: 0 Local: 1 Earth	IPv4: 192.228.79.201 IPv6: 2001:478:65::53	none
C	Cogent Communications	Sites: 6 Global: 6 Local: 0 Herndon, VA, US; Los Angeles, CA, US; New York, NY, US; Chicago, IL, US; Frankfurt, DE; Madrid, ES	IPv4: 192.33.4.12	2149
D	University of Maryland	Sites: 1 Global: 1 Local: 0 College Park, MD, US	IPv4: 128.8.10.90	27
E	NASA Ames Research Center	Sites: 1 Global: 1 Local: 0 Mountain View, CA, US	IPv4: 192.203.230.10	297
F	Internet Systems Consortium, Inc.	Sites: 49	IPv4: 192.5.5.241	3557

Root servers

Root Server Technical Op...				
http://www.root-servers.org/				
A gyors elérés érdekében helyezze könyvjelzőit ide, a könyvjelzősávra.				
További könyvjelzők				
		Mountain View, CA, US		
E	Internet Systems Consortium, Inc.	Sites: 49 Global: 2 Local: 47 Ottawa, Canada *; Palo Alto, CA, US * ; San Jose, CA, US; New York, NY, US *; San Francisco, CA, US * ; Madrid, ES; Hong Kong, HK; Los Angeles, CA, US *; Rome, Italy; Auckland, NZ *; Sao Paulo, BR; Beijing, CN; Seoul, KR *; Moscow, RU *; Taipei, TW; Dubai, AE; Paris, FR *; Singapore, SG; Brisbane, AU *; Toronto, CA *; Monterrey, MX; Lisbon, PT *; Johannesburg, ZA; Tel Aviv, IL; Jakarta, ID; Munich, DE *; Osaka, JP *; Prague, CZ *; Amsterdam, NL *; Barcelona, ES *; Nairobi, KE; Chennai, IN; London, UK *; Santiago de Chile, CL; Dhaka, BD; Karachi, PK; Torino, IT; Chicago, IL, US *; Buenos Aires, AR; Caracas, VE; Oslo, NO *; Panama, PA; Quito, EC; Kuala Lumpur, Malaysia *; Suva, Fiji; Cairo, Egypt; Atlanta, GA, US; Podgorica, ME; St. Maarten, AN *	IPv4: 192.5.5.241 IPv6: 2001:500:2ff	3557
G	U.S. DOD Network Information Center	Sites: 6 Global: 6 Local: 0 Columbus, OH, US; San Antonio, TX, US; Honolulu, HI, US; Fussa, JP; Stuttgart-Vaihingen, DE; Naples, IT	IPv4: 192.112.36.4	5927
H	U.S. Army Research Lab	Sites: 1 Global: 1 Local: 0 Aberdeen Proving Ground, MD, US *	IPv4: 128.63.2.53 IPv6: 2001:500:1::803f:235	13
I	Autonomica	Sites: 34 Stockholm, SE; Helsinki, FI; Milan, IT; London, UK; Geneva, CH; Amsterdam, NL; Oslo, NO; Bangkok, TH; Hong Kong, HK; Brussels.	IPv4: 192.36.148.17	29216

Anycast

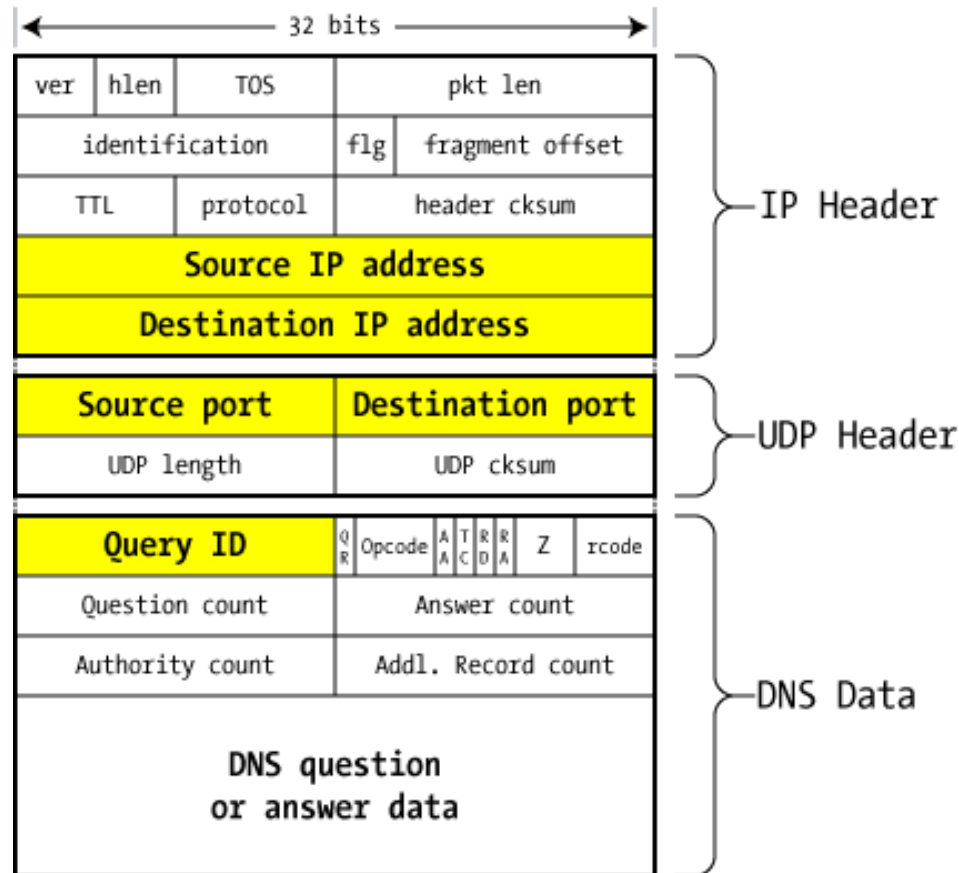
- Setting up identical copies of existing servers.
 - Same IP address.
 - Exactly the same data.
- Works like transmitter antennas for radio.
 - You will talk to (listen to) the nearest one.
 - Standard Internet routing will bring the queries to the nearest server.
 - Provides better service to more users.
 - **Mitigates impact of denial of service attacks.**

Technically, Anycast is provided by different methods, such as providing by the help of BGP routing

E.g. Anycast NTP servers at BME: time.bme.hu (152.66.0.15)

DNS packet structure

- (from <http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html>)



DNS packet on the wire

DNS packet structure

- Source / Destination IP address
 - These reflect the IP addresses of the machines that sent and should receive the packet. It's possible to **forge** the source address, but pointless to forge the destination.
 - Analog in the real world: on an envelope sent in the US Mail, you can put anything you want as the return address — the source address — but if you lie about the recipient, it's not going to go where you want.
- Source / Destination port numbers
 - DNS servers listen on port 53/udp for queries from the outside world, so the first packet of any exchange always includes 53 as the UDP destination port.
 - The source port varies considerably (though not enough, as we'll find shortly): sometimes it's also port 53/udp, sometimes it's a fixed port chosen at random by the operating system, and sometimes it's just a random port that changes every time.
 - As far as DNS functionality is concerned, the source port doesn't really matter as long as the replies get routed to it properly. But this turns out to be the crux of the problem at hand.
- Query ID
 - This is a unique identifier created in the query packet that's left intact by the server sending the reply: it allows the server making the request to associate the answer with the question.
 - A nameserver might have many queries outstanding at one time — even multiple queries to the same server — so this Query ID helps match the answers with the awaiting questions.
 - This is also sometimes called the Transaction ID (TXID).

DNS packet structure 2

- QR (Query / Response) : Set to **0** for a query by a client, **1** for a response from a server.
- Opcode
 - Set by client to **0** for a standard query; the other types aren't used in our examples.
- AA (Authoritative Answer)
 - Set to **1** in a server response if this answer is Authoritative, **0** if not.
- TC (Truncated)
 - Set to **1** in a server response if the answer can't fit in the 512-byte limit of a UDP packet response; this means the client will need to try again with a TCP query, which doesn't have the same limits.
- RD (Recursion Desired)
 - The client sets this to **1** if it wishes that the server will perform the entire lookup of the name recursively, or **0** if it just wants the best information the server has and the client will continue with the iterative query on its own. Not all nameservers will honor a recursive request (root servers, for instance, won't ever perform recursive queries).
- RA (Recursion Available)
 - The server sets this to indicate that it will (**1**) or won't (**0**) support recursion.
- Z — reserved
 - This is reserved and must be zero
- rcode
 - Response code from the server: indicates success or failure
- Question record count
 - The client fills in the next section with a single "question" record that specifies what it's looking for: it includes the name (**www.unixwiz.net**), the type (**A**, **NS**, **MX**, etc.), and the class (virtually always **IN**=Internet).
 - The server repeats the question in the response packet, so the question count is almost always **1**.
- Answer/authority/additional record count
 - Set by the server, these provide various kinds of answers to the query from the client: we'll dig into these answers shortly.
- DNS Question/Answer data
 - This is the area that holds the question/answer data referenced by the count fields above. These will be discussed in great detail later.

Importance of DNS security

- DNS is crucial for the internet. Without DNS there is practically no internet
- DoS attacks against root DNS servers can collapse the net
- Any attack on authoritative DNS servers or just on some caching/recursive servers could redirect traffic to other hosts (man-in-the-middle attack)
- The SSL is also not working without proper DNS
 - Proof of domain ownership: DNS RR
 - DoT (DNS over TLS), DoH (DNS over HTTPS): privacy, integrity
 - CAA record: specify which certificate authorities (CAs) are allowed to issue certificates for a domain
- The ownership of a domain name represents money

DNS spoofing / poisoning –attack methods 1.

- the cache of a DNS name server can be poisoned with false information
- how to do it?
- assume that the attacker wants *www.anything.hu* to map to his own IP address 152.66.249.32
- approach 1 (spoofing answer):
 - attacker submits a DNS query “*www.anything.hu=?*” to ns.victim.hu
 - The victim will start a resolve process
 - a bit later it forges a DNS reply “*www.anything.hu=152.66.249.32*”
 - UDP makes forging easier but the attacker must still predict the **query ID (no port check earlier)**
 - No new trial is possible until the end of TTL (ns.victim.hu won't try it again!)

Why approach 1 does not work?

- In practice, it is almost impossible to execute this attack
 - TTL is 1-2 days
 - 2^{16} IDs
 - Very low probability that the attacker wins
 - Only possible in a man-in-the-middle fashion attack, where we can “grap” and delete the original answer and replace it with our version

DNS spoofing / poisoning –attack methods 2.

- the cache of a DNS name server can be poisoned with false information
- how to do it?
- assume that the attacker wants *www.anything.hu* to map to his own IP address 152.66.249.32
- approach 2 (attacker has access to ns.attacker.hu):
 - the attacker modifies its local name server such that it responds a query “www.attacker.hu=?” with “www.anything.hu=152.66.249.32”
 - the attacker then submits a query “www.attacker.hu=?” to ns.victim.hu
 - ns.victim.hu sends the query “www.attacker.hu=?” to ns.attacker.hu
 - ns.attacker.hu responds with [www.anything.hu=152.66.249.32](#)
 - This attack does not work generally due to restrictions on hints/non-glue records and query id checks

Why approach 2 does not work?

- How does a nameserver know that any response packet is "expected"?
 - The response arrives on the same UDP port we sent it from: otherwise the network stack would not deliver it to the waiting nameserver process (it's dropped instead).
 - The **Question** section (which is duplicated in the reply) matches the Question in the pending query.
 - The **Query ID** matches the pending query
 - The Authority and Additional sections represent names that are within the same domain as the question: this is known as "bailiwick checking".
 - This prevents **ns.unixwiz.net** from replying with not only the IP address of **www.unixwiz.net**, but also fraudulent information about (say) **BankOfSteve.com**.

The Kaminsky attack - 2008

- Coordinated defense by all major vendors
- Target: the attacker wants to insert a fake information into a caching DNS server for `www.anything.hu`
- attacker submits a DNS query “a1.anything.hu=?” to ns.victim.hu
- a bit later it forges a DNS reply “a1.anything.hu=152.66.249.32” and in the additional information part a hint www.anything.hu=152.66.249.32, the answer contains a random Query ID.
- If the attacker cannot successfully guess the right Query ID, then the forged answer will be discarded
- Race condition: If the Query ID is right, we have to be faster than the real answer, otherwise the answer will be discarded
- There is a high chance that a single attack will be unsuccessful
- However, if it did not work, go on with a2.anything.hu, a3.anything.hu, etc.
- At some point, the attacker will successfully guess the Query ID and will be faster than the real server.
- In this case the a3423942.anything.hu will be poisoned, but this is not very interesting
- Due to the additional information, www.anything.hu might be inserted into the cache!
- Only works if the www.anything.hu is not already in the cache.

Protection against the Kaminsky attack: port randomization

- The source port of the DNS query is being randomized (e.g. MS: 11 bit)
- The attacker should respond to that particular UDP port
- Original chance: 2^{-16} to find out the query ID
- Randomization might use only 2^{11} random ports (2048) as source ports (MS)
- The new probability for the attacker when using random ports is $2^{-27} \sim 1:134$ million
- The attacked address should not be in the cache at the time of the attack, which is not true for a long time on busy DNS servers!

Information leakage from DNS

- Send a query on something.com to a local DNS server
- If the DNS server answers with the maximum TTL, then the domain was just downloaded at the time of the query
- This way, one can check/prove if a specific domain was used / a web page was downloaded lately, or not
- If recursive queries are disabled, the attacker cannot use this tool (except insiders)

Securing DNS: Restricting zone transfers

- Zone transfer is used to synchronize DNS zone among DNS servers (e.g. from primary to secondary)
 - Initiated by the client
 - » The client does not know when has the zone changed
 - » The primary server can send out notifications so the zone can be fresh very soon at least at the authoritative servers
- Leak information about all existing zones (e.g records for dev or test systems)
- Imagine sending to a country server: IN AXFR
- National registry might need right for zone transfer to check the technical settings of the zone
- For all the others: zone transfer is not necessary

Securing DNS: DNSSEC

- add security, while maintaining backwards compatibility
- All answers in DNSSEC are digitally signed
- By checking the digital signature, a DNS resolver is able to check if the information is identical (correct and complete) to the information on the authoritative DNS server
- new DNS record types: RRSIG, DNSKEY, DS, and NSEC
- When DNSSEC is used, each answer to a DNS lookup will contain an RRSIG DNS record
- The RRSIG record is a digital signature of the answer DNS resource record set
- The digital signature can be verified by locating the correct public key found in a DNSKEY record

2020- new things around DNS

- Domain Name System (DNS) Cookies

DNS Cookies are a lightweight DNS transaction security mechanism that provides limited protection to DNS servers and clients against a variety of increasingly common denial-of-service and amplification/ forgery or cache poisoning attacks by off-path attackers (most new operating systems use DNS cookies)

The default server behavior is to accept, but not require, cookies

In BIND 9.11.0 and later versions, we use DNS Cookies to white-list known clients for Response Rate Limiting (RRL).

- DNS over HTTPS/TLS (sent to reliable DNS server)

Proof of Non-Existence (next lecture)

- NSEC
 - Integrity protection
- NSEC3
 - Integrity protection
 - Opt out for insecure child
- NSEC5
 - Integrity protection
 - Preventing DNSSEC Zone Enumeration

example.com. 300 IN NSEC alice.example.com. A RRSIG NSEC
alice.example.com. 300 IN NSEC edward.example.com. A RRSIG NSEC
edward.example.com. 300 IN NSEC susan.example.com. A RRSIG NSEC
susan.example.com. 300 IN NSEC example.com. A RRSIG NSEC

Questions?

Control question

- What DNS record types do you know?
- How a DNS request is resolved?
- What is the TTL in the records?
- What is DNS poisoning?
- Why the DNS query ID is important?
- How does the Kaminsky attack work?