UCF Stands For Opportunity

CIS3360: Security in Computing

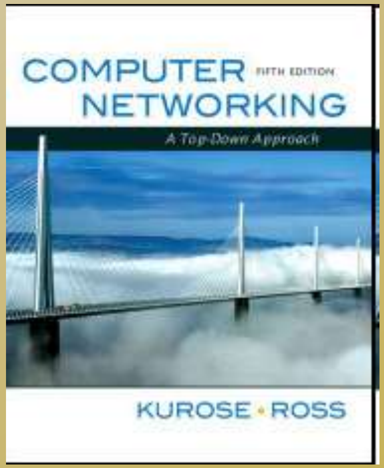
Chapter 6 : Network Security II

Cliff Zou

Spring 2012

DNS Introduction

- DNS introduction content is mainly from the reference book:
  - Computer Networking: A Top Down Approach Featuring the Internet, J. Kurose & K. Ross, Addison Wesley, 5th ed., 2009



DNS: Domain Name System

People: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit) – 128.119.40.12
  unique ID
- "name", e.g., www.yahoo.com - used by humans

Q: How to map between IP addresses and name ?

Domain Name System:

- *distributed database* implemented in hierarchy of many *name servers*

DNS

DNS services

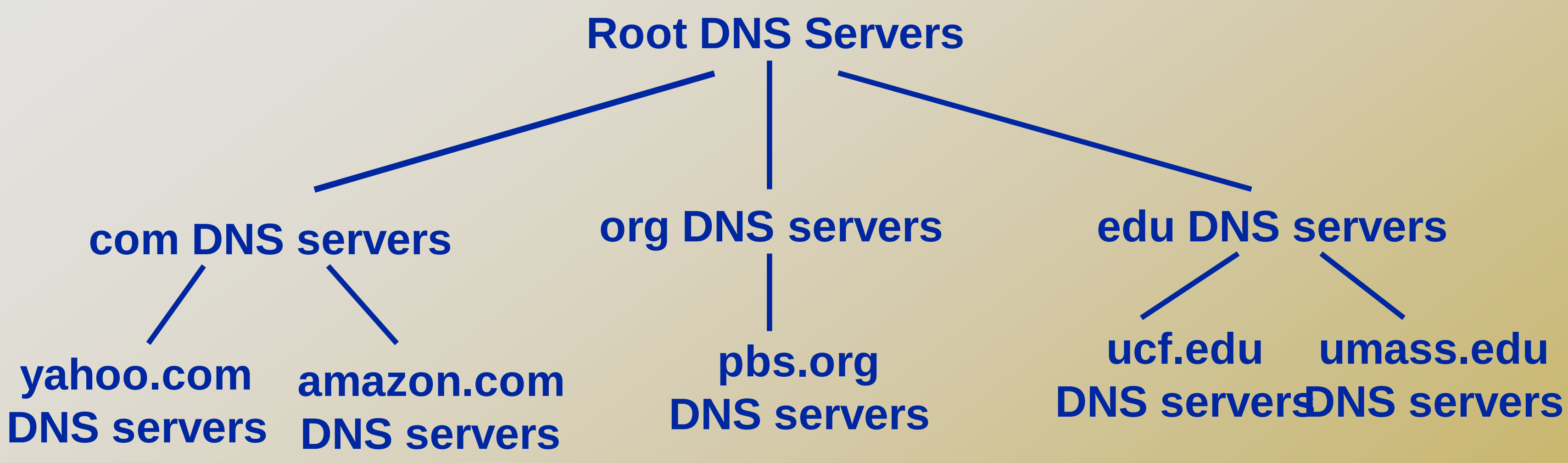
- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
  - Many names for a single host
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

Why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

doesn't *scale!*

Distributed, Hierarchical Database



Client wants IP for www.amazon.com; 1st approx:

- Client queries a root server to find com DNS server
- Client queries "com" DNS server to get amazon.com DNS server
- Client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: Root name servers

13 root name servers worldwide

TLD and Authoritative Servers

- Top-level domain (TLD) servers: responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
  - Network solutions maintains servers for com TLD
  - Educause for edu TLD
- Authoritative DNS servers: organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail).
  - Can be maintained by organization or service provider (paid by the organization)

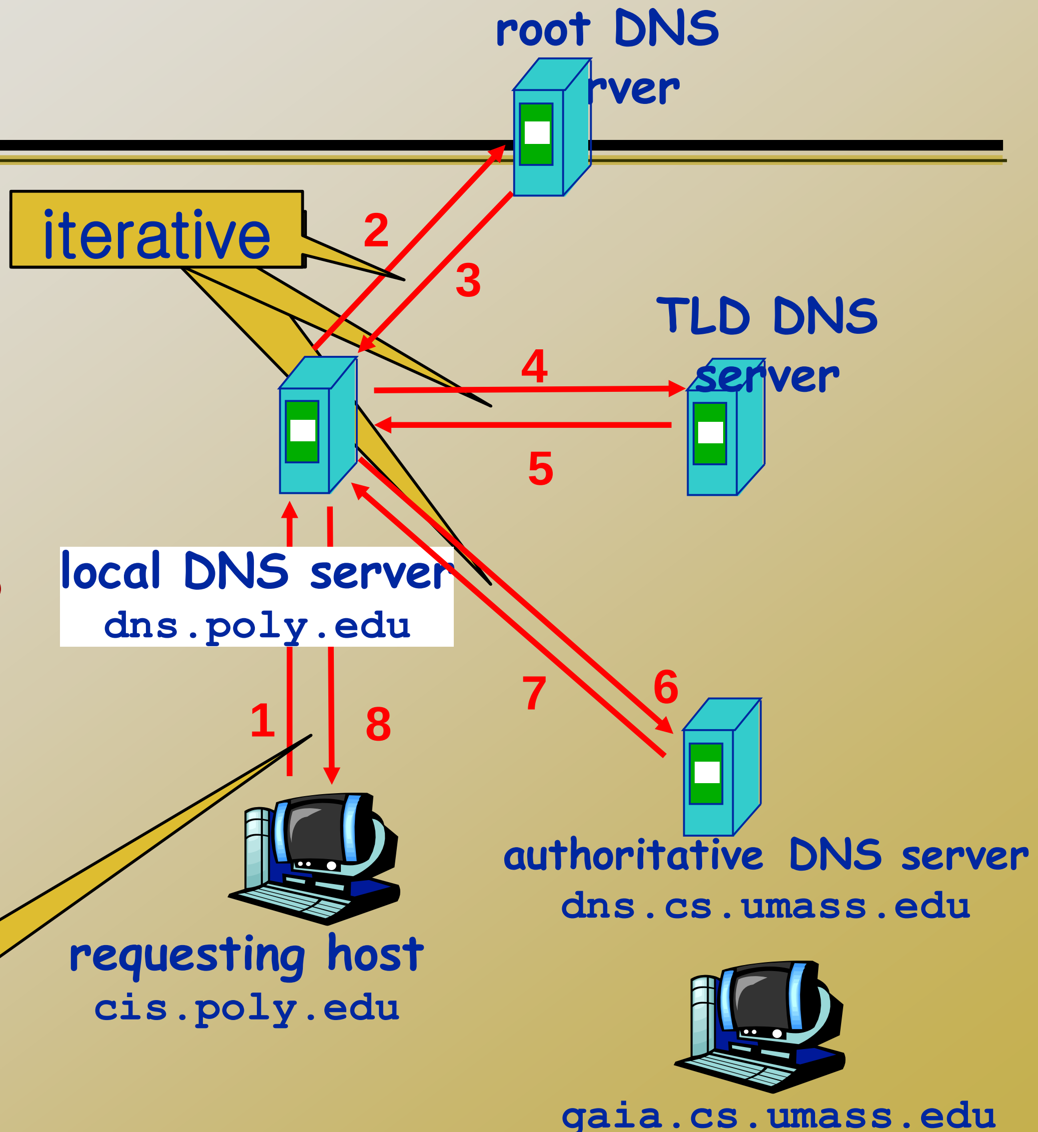
Local Name Server

- Does not strictly belong to hierarchy
- Each ISP (residential ISP, company, university) has one
  - Also called "default name server"
- When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a proxy (cache), forwards query into hierarchy

Iterative and Recursive queries

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu

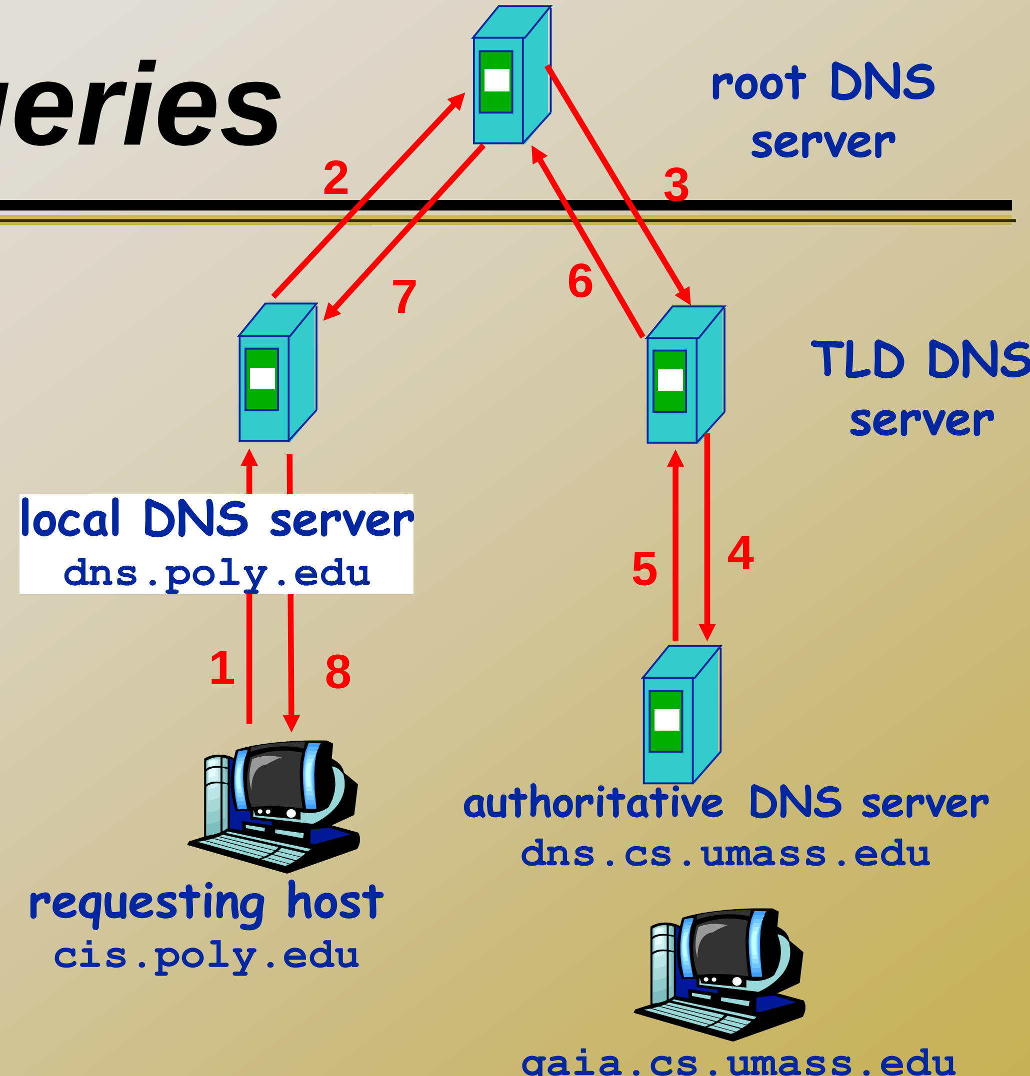
- The query to root DNS rarely happens due to cache of all TLD DNS information at local DNS server



Recursive queries

recursive query:
- DNS client requires DNS server respond with either the requested resource record, or an error message stating that the record or domain name does not exist.

iterative query:
- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”



Reference: http://technet.microsoft.com/en-us/library/cc961401.aspx

DNS: caching and updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time (keep fresh copy)
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

DNS records

DNS: distributed db storing Resource Records (RR)

RR format: `(name, value, type, ttl)`

Type=A

- **`name`** is hostname
- **`value`** is IP address

Type=NS

- **`name`** is domain (e.g. foo.com)
- **`value`** is IP address of authoritative DNS server for this domain

Type=CNAME

- **`name`** is alias name for some "canonical" (the real) name
  `www.ibm.com` is really **`servereast.backup2.ibm.com`**
- **`value`** is canonical name

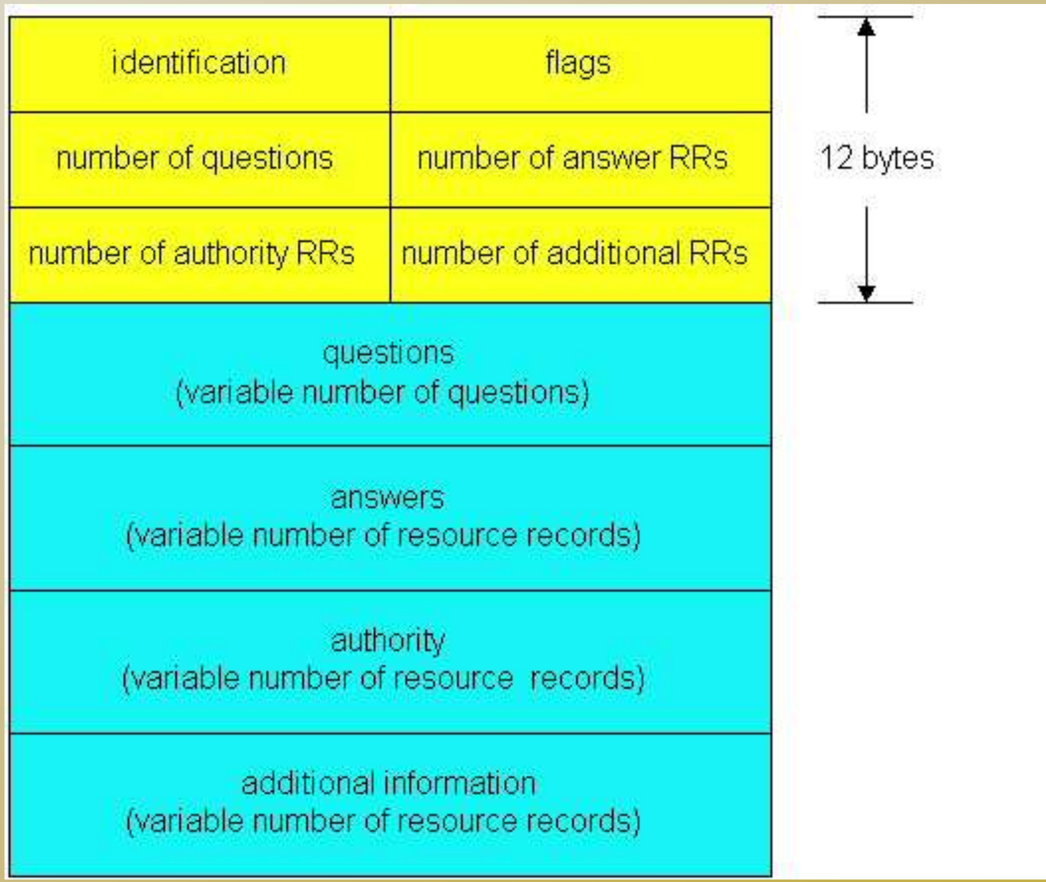
Type=MX

- **`value`** is name of mailserver associated with **`name`**

DNS protocol, messages

DNS protocol : *query* and *reply* messages, both with same *message format*

msg header
- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



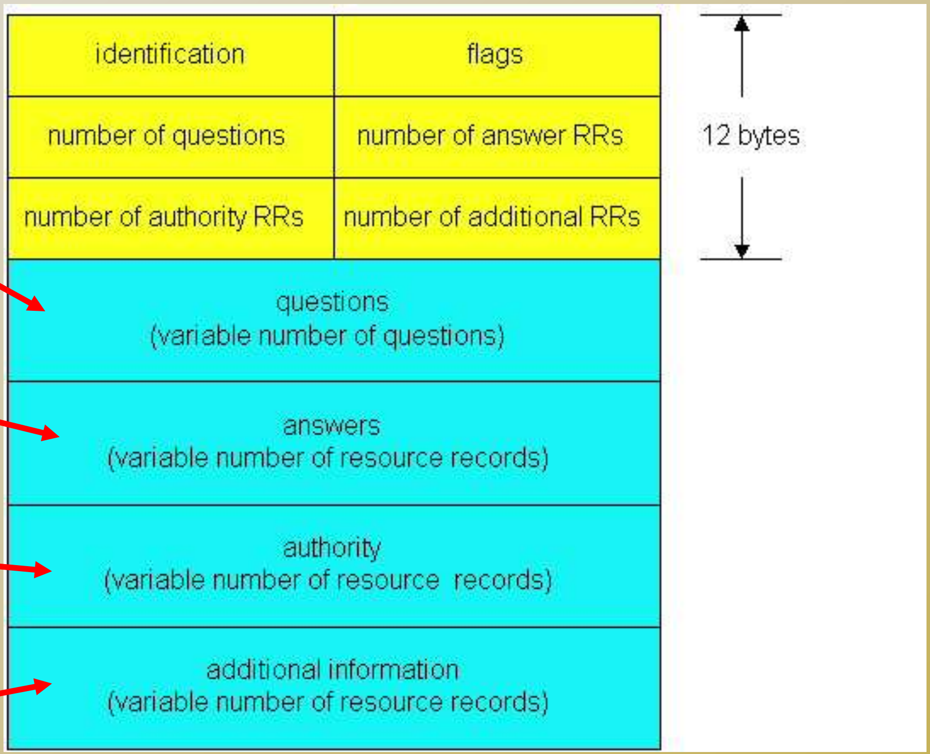
DNS protocol, messages (UDP 53)

Name, type fields for a query

RRs in response to query

records for authoritative servers

additional "helpful" info that may be used



- Let's check a web example using Wireshark!
- Check MX record:
  - nslookup –type=MX cs.ucf.edu   (Under Windows)
  - dig mx cs.ucf.edu   (Under Unix)

Inserting records into DNS

- Example: just created startup "Network Utopia"
- Register name networkuptopia.com at a registrar (e.g., Network Solutions)
  - Need to provide registrar with names and IP addresses of your authoritative name server (primary and secondary)
  - Registrar inserts two RRs into the com TLD server:

```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

- Put in authoritative server `dns1.networkutopia.com`
  - Type A record for www.networkuptopia.com
    - Type CName for networkuptopia.com (alias)
  - Type MX record for networkutopia.com (email)
    - Type A record for the email server
- How do people get the IP address of your Web site?

DNS Security

Cybersquatting

- Cybersquatting is to register a domain in anticipation of that domain being desirable to another organization
  - Intent to sell tot hat organization for big profit
- For example, You can register "hurricane2013.com", or "hurricane-in-Texas.com" if you think there will be a big one in Texas in the near future.
  - Sell it for big profit if it is true!
  - Domain name purchase is cheap!
- Many organizations have to buy all related domain names to prevent cybersquatting
- A legitimate example: http://teaparty.com/
  - suspicious ones for tea party: http://tparty.com/, http://t-party.com/
- http://en.wikipedia.org/wiki/Cybersquatting

Typosquatting

- Register all possible typo domain names for another organization
  - Should a user accidentally enters an incorrect website address, he may be led to an alternative website owned by a cybersquatter.
  - Could lead to phishing attack (malicious), or increase web visits (not very malicious)
- For example, for "bankofamerica.com", a cybersquatter could register:
  - "bankamerica.com", "bankoamerica.com", "bankofamerican.com", "bankfoamerica.com", ……
  - Domain name purchase is cheap!

OS DNS Cache Privacy

- Windows OS maintain a local DNS cache
  - Command "ipconfig/displaydns"
- DNS cache reveals a user's browsing history
  - Even if the user deletes browsing cache and cookies

- Internet Explorer does not have its own DNS cache
- Cross-platform browser, such as Firefox, has its own DNS cache

DNS Vulnerability

- Most DNS queries and responses are in plaintext
- No authentication is done for DNS response
  - You really has no good way to tell if the DNS response you get are trustable or not!
- DNS is mostly relying on UDP packets
  - IP address spoofing is very easy for UDP packets
    - No seq/ack numbers

DNS Cache Poisoning

- Basic idea: give DNS servers false records and get it cached
- DNS uses a 16-bit request identifier to pair queries with answers
- Cache may be poisoned when a name server:
  - Disregards identifiers
  - Has predictable ids
  - Accepts unsolicited DNS records

DNS Cache Poisoning Procedure

- Eve wants to poison attack an ISP DNS server
  - Eve transmits a DNS query to this server, which in turn queries authoritative DNS on behalf of Eve
  - Eve simultaneously sends a DNS response to the server, spoofing with the authoritative server's IP
  - The ISP's DNS server accepts the forged response and caches a wrong DNS entry
    - All downstream users of this ISP will be directed to the wrong website

DNS Cache Poisoning Prevention

- Use random identifiers for queries
  - Make it hard to guess the ID number
- Always check identifiers
- Port randomization for DNS requests
- Deploy DNSSEC
  - Challenging because it is still being deployed and requires reciprocity

DNS Cache Poisoning against Query ID

- Even if a DNS server checks response IDs and use random IDs, it is still vulnerable to the attack
  - Attacker generates a flux of DNS requests and send the corresponding flux of DNS response back
  - If one of the pair has matched ID, the attack is successful
  - Birthday Paradox: the prob. Of two persons in 23 people share the same birthday is more than 50%!

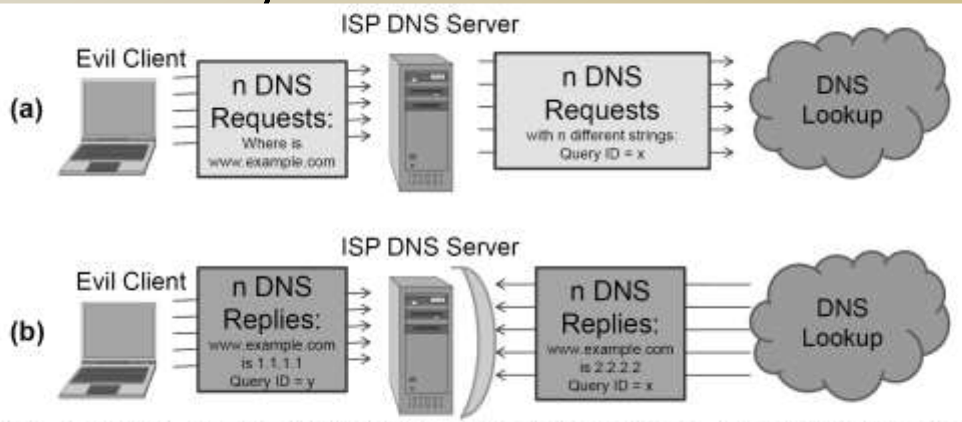


Figure 6.7: A DNS cache poisoning attack based on the birthday paradox: (a) First, an attacker sends n DNS requests for the domain she wishes to poison. (b) The attacker sends n corresponding replies for her own request. If she successfully guesses one of the random query IDs chosen by the ISP DNS server, the response will be cached.

Some Defenses

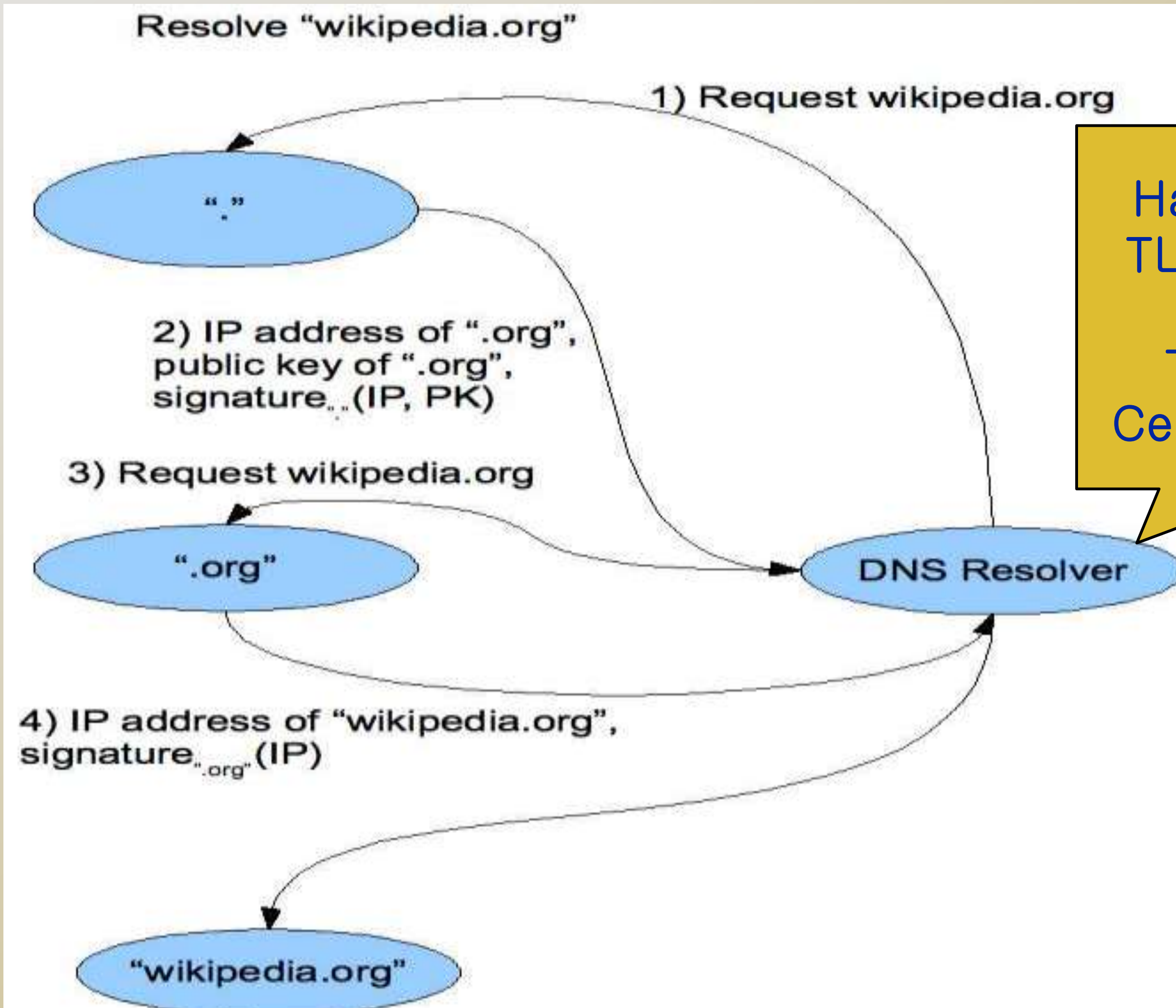
- Fact: Most DNS poisoning target local DNS (LDNS) server
- Solution: Configure LDNS to only accept requests from internal networks
  - Why does it need to server outside users?

- Source-port randomization (SPR)
  - DNS query sent out will have two randomized numbers:
    - Source port number (destination port always 53)
    - Query ID number (16 bits)
  - Check DNS response for both of these numbers

DNSSEC

- Guarantees:
  - Authenticity of DNS answer origin
  - Integrity of reply
  - Authenticity of denial of existence
- Accomplishes this by signing DNS replies at each step of the way
- Uses public-key cryptography to sign responses
- Typically use trust anchors, entries in the OS to bootstrap the process

DNS Signing



Hard-coded with TLD's public keys

TLDs serve as Certificate Authority

DNSSEC Deployment

- As the internet becomes regarded as critical infrastructure there is a push to secure DNS
- NIST is in the process of deploying it on root servers now
- May add considerable load to dns servers with packet sizes considerably larger than 512 byte size of UDP packets
- There are political concerns with the US controlling the root level of DNS