DNS Introduction

DNS

DNS services

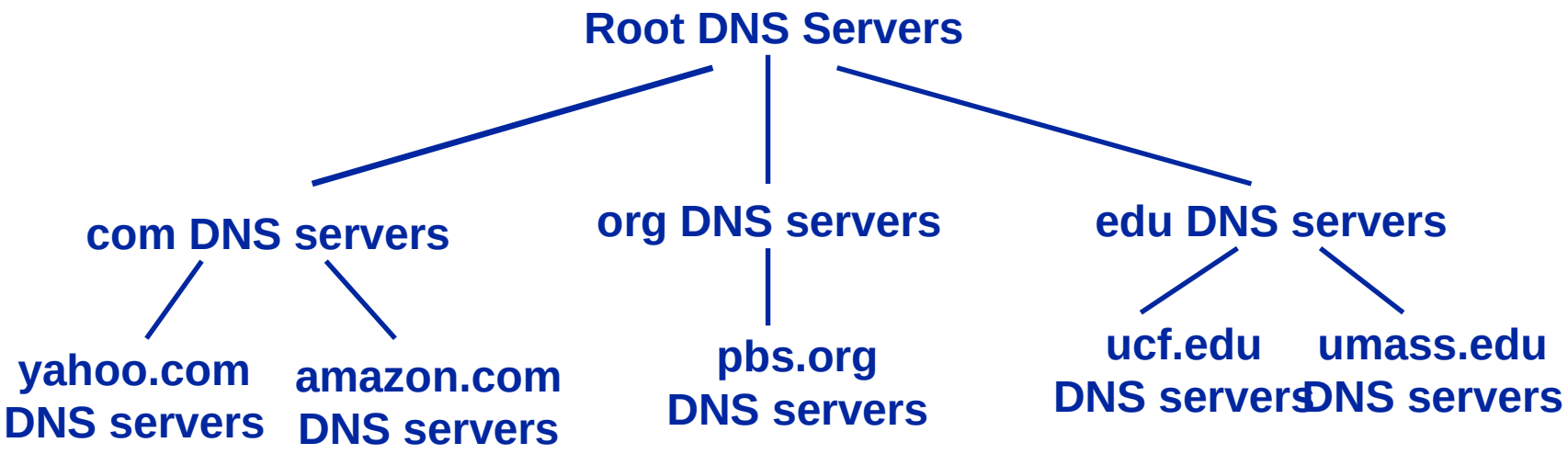
- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
  - Many names for a single host
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

Why not centralize DNS?

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

doesn't *scale!*

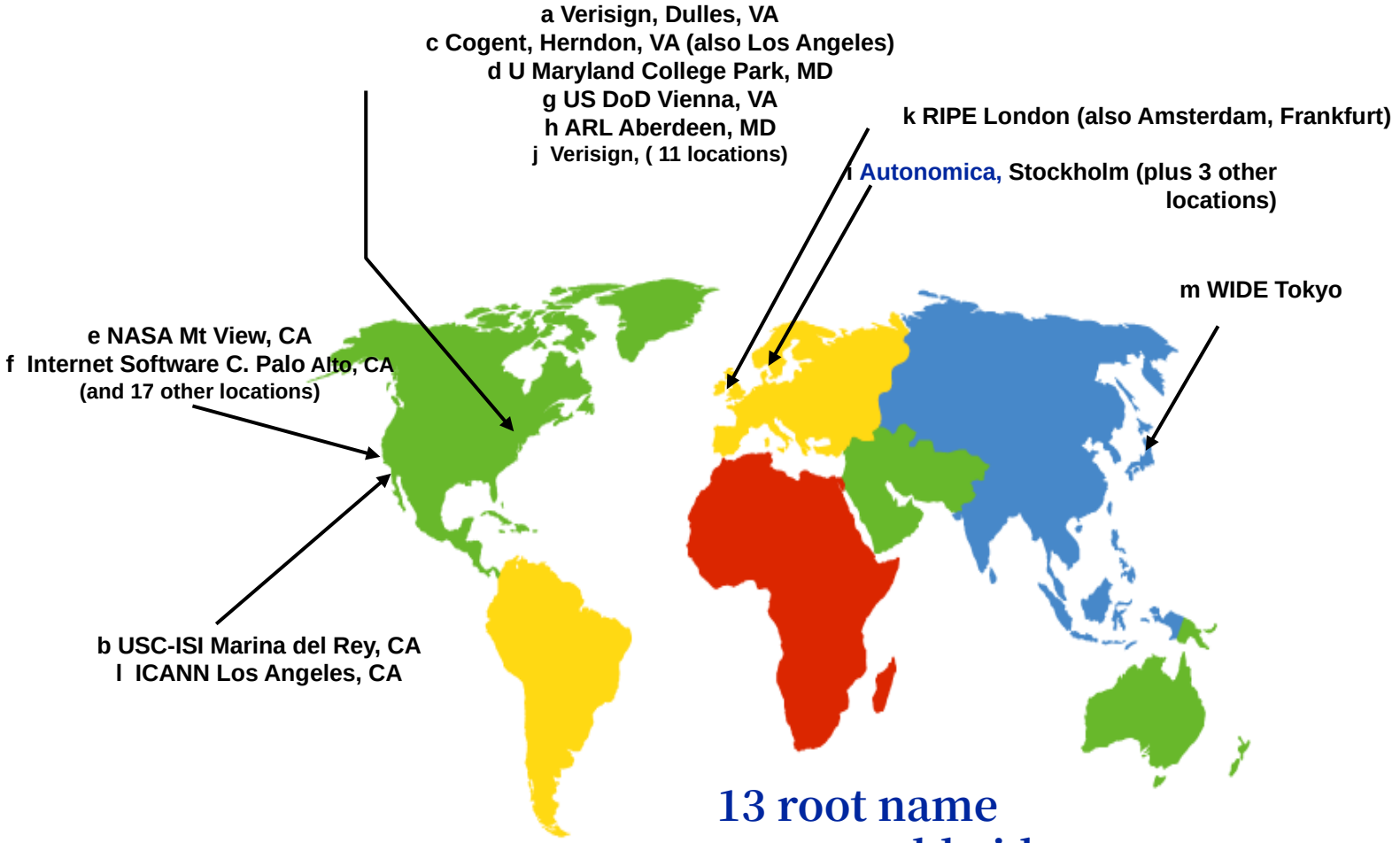
Distributed, Hierarchical Database



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

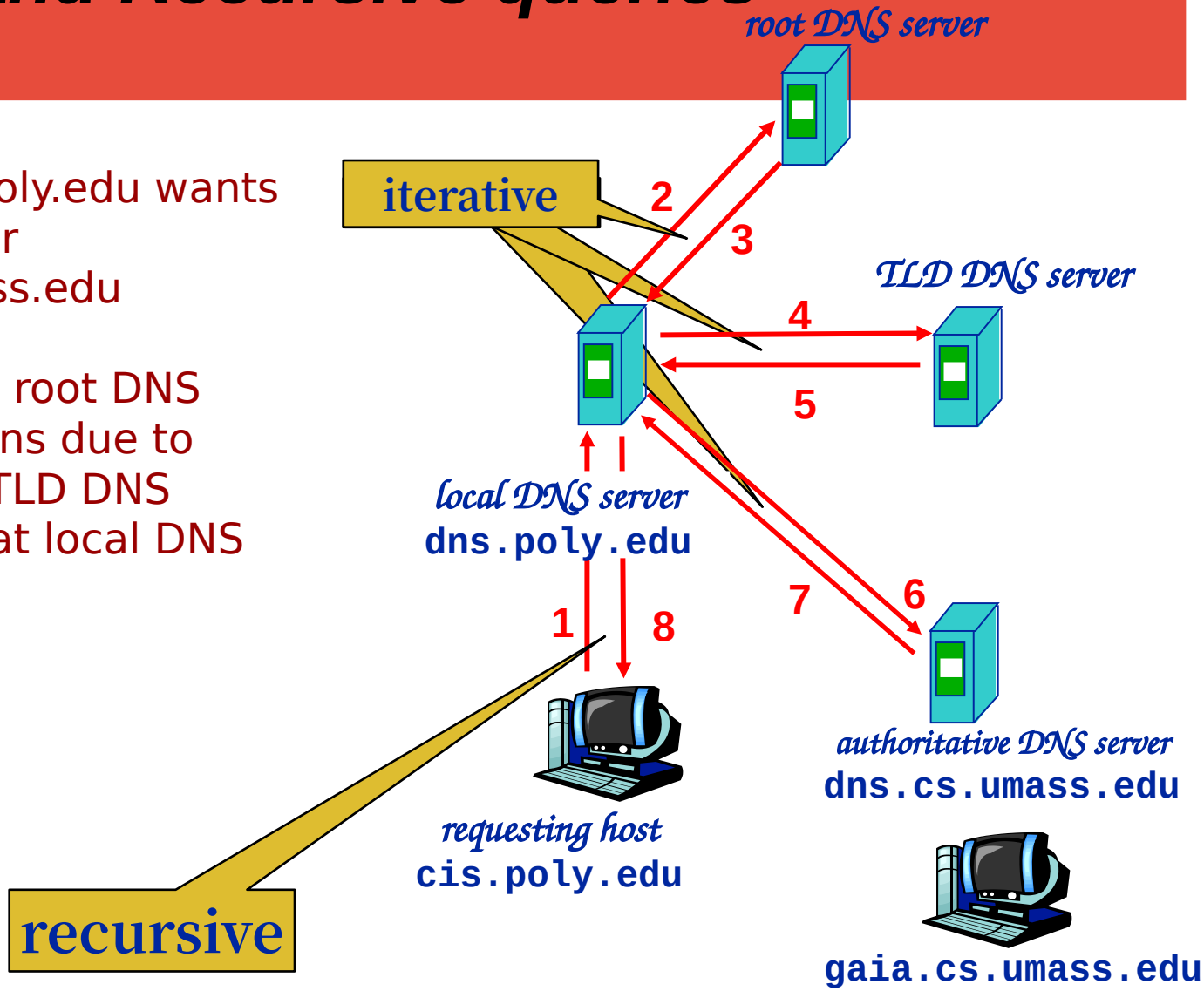
- Top-level domain (TLD) servers: responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
  - Network solutions maintains servers for com TLD
  - Educause for edu TLD
- Authoritative DNS servers: organization's DNS servers, providing authoritative hostname to IP mappings 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



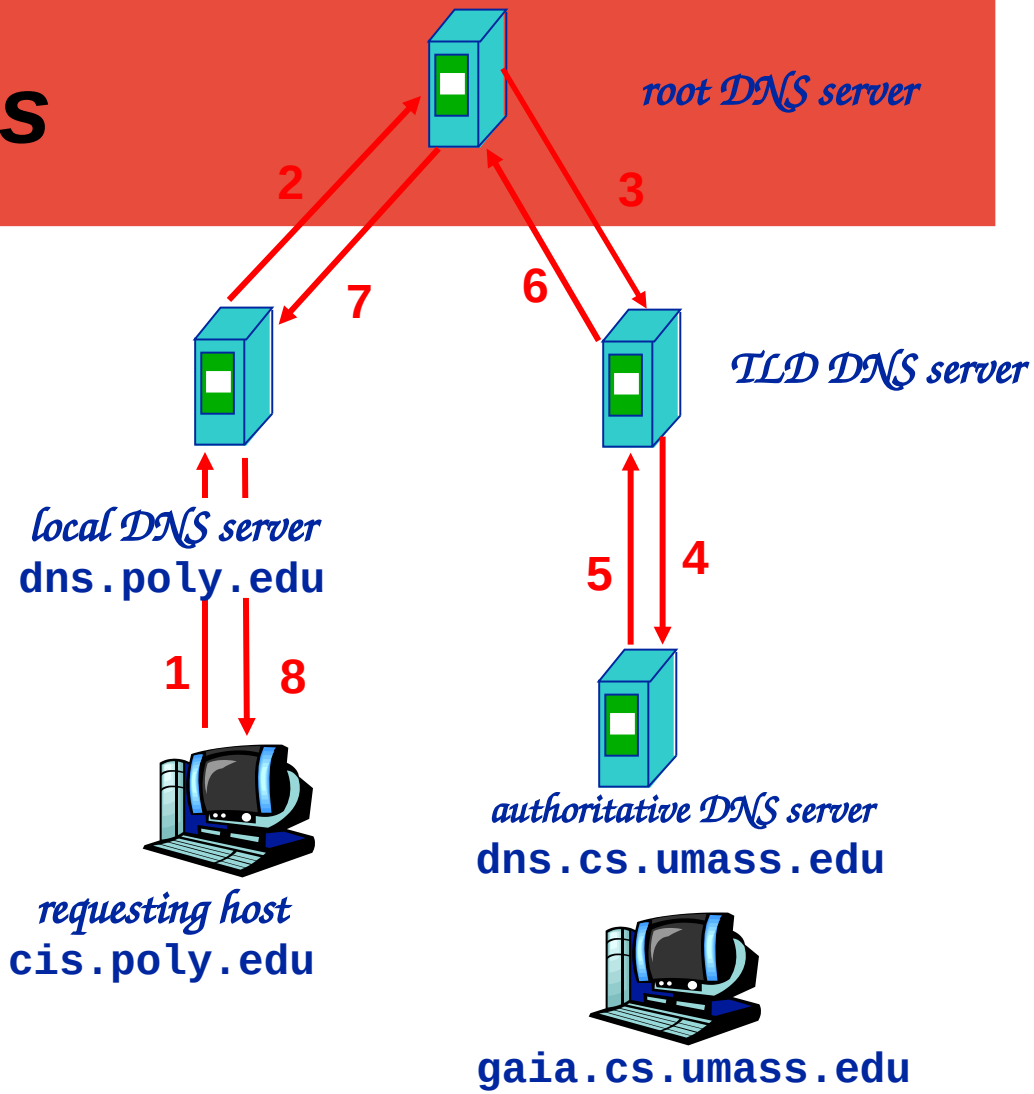
Recursive queries

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

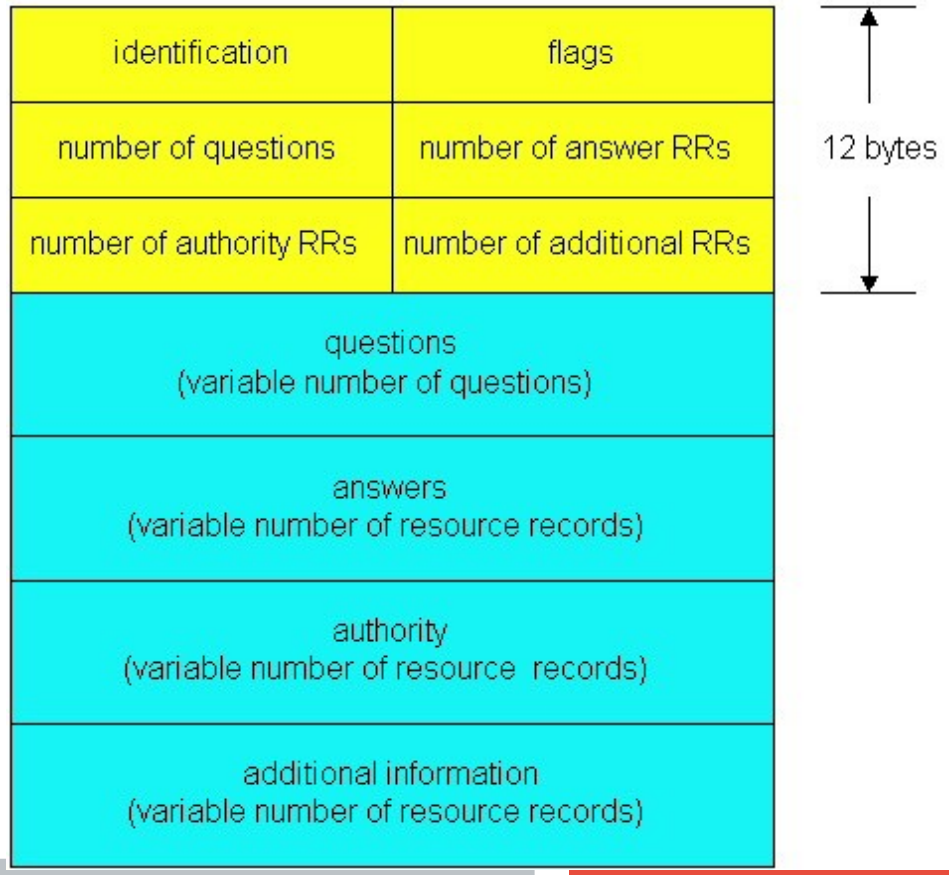
Type=MX
- **value** is name of mailserver associated with **name**

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**



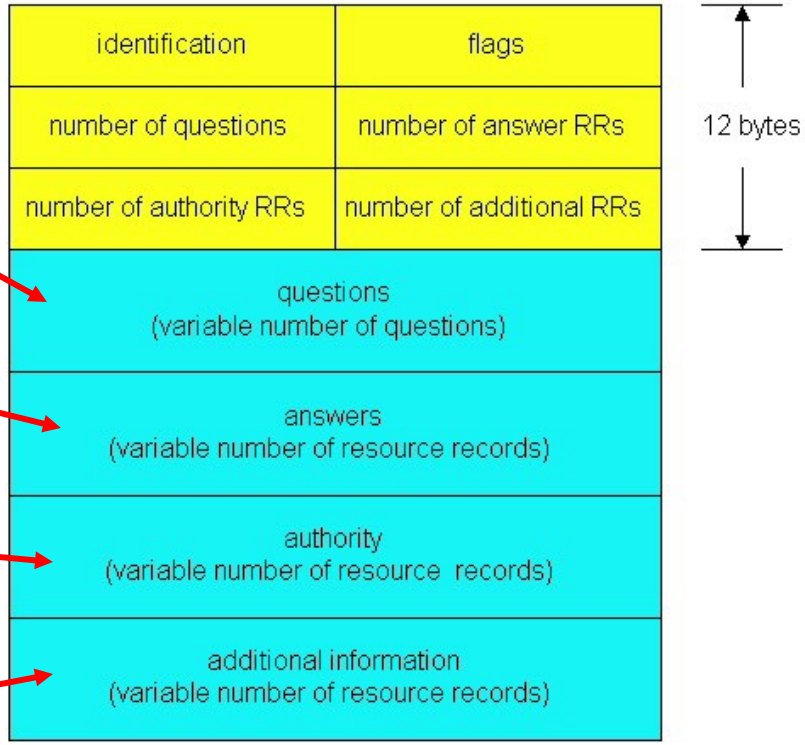
DNS protocol, messages (UDP 53)

Name, type fields for a query

RRs in response to query

records for authoritative servers

additional "helpful" info that may be used



Example using Wireshark

- *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

Inherent DNS Vulnerabilities

- Users/hosts typically trust the host-address mapping provided by DNS
- Obvious problems
  - *Interception of requests or compromise of DNS servers can result in incorrect or malicious responses*
  - *Solution – authenticated requests/responses*
- Delegation of queries allowed by RFC
  - Name server may delegate name to another NS (this is OK)
  - If name is delegated, may also supply IP addr (this is trouble)

DNS Implementation Vulnerabilities

- implementations have had same kinds of vulnerabilities as other software
  - Reverse query buffer overrun in BIND Releases 4.9 (4.9.7 prior) and Releases 8 (8.1.2 prior)
    - gain root access
    - abort DNS service
- MS DNS for NT 4.0 (service pack 3 and prior)
  - crashes on chargen stream
  - telnet ntbox 19 | telnet ntbox 53
- Moral
  - Better software quality is important
  - Defense in depth

Type of DNS attacks

DNS Attacks Target Cache, Recursive and Authoritative Functions

- *Volumetric Attacks*
- *Exploits*
- *Stealth/Slow Drip DoS Attacks*
- *Protocol Abuse*

Volumetric Attacks

- **Direct DNS DoS Attack**
  Flooding of DNS servers with direct requests causing saturation of cache, recursion or authoritative functions. This attack is usually sent from a spoofed IP address.

- **DNS Amplification (DDoS)**

  Use of publicly accessible open DNS servers for flooding a target system with DNS response traffic. The source address of a sent DNS name lookup request is spoofed to be the target's address, which consequently receives the response. For maximizing amplification effect, a request for as much zone information as possible is sent.

Volumetric Attacks

- **DNS Reflection Attack**
  - Flooding authoritative servers or infrastructure components such as firewalls, with the objective often being to exhaust the bandwidth of the network targeted. The attack makes use of the numerous distributed open resolver servers on the Internet and is usually combined with amplification attacks.

- **Bogus Domain Attack**
  - Flooding of the DNS servers with non-existing domain requests implying recursive function saturation. This attack consumes resources on the DNS server for the recursion process and reduces its efficiency in answering legitimate queries. This attack is sometimes called a NXDOMAIN attack..

Exploits

- **Zero-Day Vulnerability**
  - Zero-dayz attacks take advantage of DNS security holes in software for which no solution is currently available.
- **DNS-based Exploits**
  - Attacks exploiting bugs and/ or flaws in DNS services, protocol or on operating system running DNS services.
- **Protocol Anomalies**
  - DNS attacks based on malformed queries intending to crash the service.
- **DNS Rebinding**
  - Combination of javascript and IP subnet discovery in order to attack local network IP devices through the browser. This attack is mainly used for discovery of unsafe devices (targeting IoT) on the network, and for data exfiltration.

Stealth/Slow Drip DoS Attacks

- **Sloth Domain Attack**
  Attacks using queries sent to hacker's authoritative domain that very slowly answers requests, just before the time out, to cause capacity exhaustion on victim's recursive server.
- ***Phantom Domain Attack***
  Attacks targeting DNS resolvers by sending them subdomains for which the domain server is unreachable, causing saturation of cache server capacity.
- **Pseudo-Random Subdomain Attack (PRSD)**
  Attacks using random query name as a subdomain of the victim's domain, causing saturation of its authoritative server capacity. This attack uses either open relay DNS or DNS recursive farm at ISP in order to also exhaust resources of servers waiting for answers from the authoritative server.

Protocol Abuse

- **DNS Tunneling**
  The DNS protocol is used to encapsulate other protocols or data in order to remotely control malware or/and the exfiltration of data.

- **DNS Cache Poisoning**
  Attacks introducing data into a DNS resolver’s cache, causing the name server to return an incorrect IP address for further requests, diverting traffic to the attacker’s computer.

- **DNS Hijacking - Farming**
  Hosted on local computer, malware alters TCP/IP configurations to point to a malicious DNS server, causing traffic to be redirected to a phishing website.

Protocol Abuse

- **DNS Hijacking - Phishing**
  DNS records are modified at the registrar level (after the compromission of administrator's credentials) and users are redirected to malicious website since using valid domains.

- **Subdomain Hijacking**
  Attack aiming to reuse an existing DNS entry (generally a CNAME) associated to a public cloud resource that has been suppressed.

- **Domain Squatting**
  Attack using registered domain names with a typo in order to get capture or redirect legitimate traffic to another web site..

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 Example

- DNS resource records (see RFC 1034)
  - An “A” record supplies a host IP address
  - A “NS” record supplies name server for domain
- Example
  - www.evil.org NS ns.yahoo.com /delegate to yahoo
  - ns.yahoo.com A 1.2.3.4 / address for yahoo
- Result
  - If resolver looks up www.evil.org, then evil name server will give resolver address 1.2.3.4 for yahoo
  - Lookup for yahoo through cache goes to 1.2.3.4

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

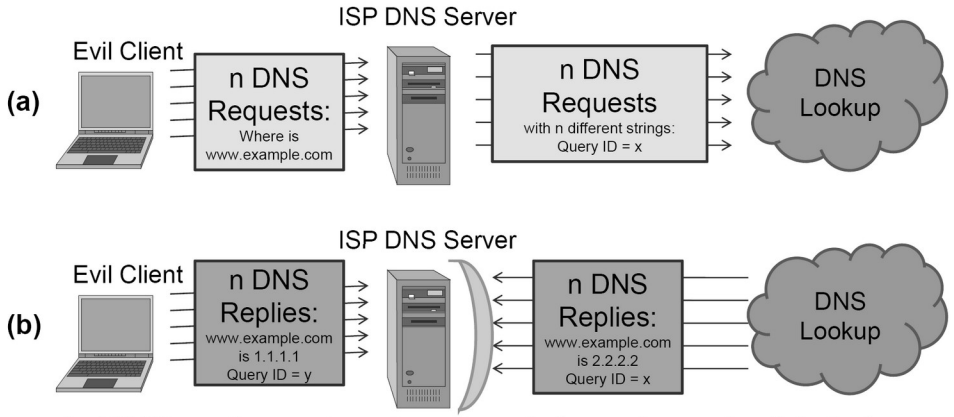


Figure 6.7: A DNS cache poisoning attack based on the birthday paradox: (a) First, an attacker sends n DNS requests for the domain she wishes to poison. (b) The attacker sends n corresponding replies for her own request. If she successfully guesses one of the random query IDs chosen by the ISP DNS server,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

DNS Pharming (="farming" and "phishing")

- DNS poisoning attack (less common than phishing)
  - Change IP addresses to redirect URLs to fraudulent sites
  - Potentially more dangerous than phishing attacks
  - No email solicitation is required
- Pharming can be conducted either by changing
  - the hosts file on a victim's computer (incorrect entries in a desktop computer's hosts file via a malware) or
  - by exploitation of a vulnerability in DNS server software, or
  - By compromising local-routers (difficult to detect!!)
    - misconfiguration of existing settings
    - wholesale rewrite of firmware
    - on purpose by the admin !!!

DNS pharming

- DNS poisoning attacks have occurred:
  - January 2005, the domain name for a large New York ISP, Panix, was hijacked to a site in Australia.
  - In November 2004, Google and Amazon users were sent to Med Network Inc., an online pharmacy
  - In March 2003, a group dubbed the "Freedom Cyber Force Militia" hijacked visitors to the Al-Jazeera Web site and presented them with the message "God Bless Our Troops"

DNS Rebinding Attack

- A malicious web page causes visitors to run a client-side script that attacks machines elsewhere on the network.
- In theory, the same-origin policy prevents this from happening: client-side scripts are only allowed to access content on the same host that served the script.
- Comparing domain names is an essential part of enforcing this policy, so DNS rebinding circumvents this protection by abusing the Domain Name System (DNS).

DNS Rebinding: how it works

- The attacker registers a domain (such as attacker.com) and delegates it to a DNS server that is under the attacker's control.
- The server is configured to respond with a very short time to live (TTL) record, preventing the DNS response from being cached.
- When the victim browses to the malicious domain, the attacker's DNS server first responds with the IP address of a server hosting the malicious client-side code. *For instance, they could point the victim's browser to a website that contains malicious JavaScript or Flash scripts that are intended to execute on the victim's computer.*
- The malicious client-side code makes additional accesses to the original domain name (such as attacker.com). These are permitted by the same-origin policy.
- However, when the victim's browser runs the script it makes a new DNS request for the domain, and the attacker replies with a new IP address. For instance, they could reply with an internal IP address or the IP address of a target somewhere else on the Internet.

DNS Rebinding Protection

- DNS servers in the chain can filter out private IP addresses and loopback IP addresses:
  - External public DNS servers (e.g. OpenDNS) can implement DNS filtering.
  - Local system administrators can configure the organization's local nameserver(s) to block the resolution of external names into internal IP addresses. (This has the downside of allowing an attacker to map the internal address ranges in use.)

- A firewall (e.g. dnswall), in the gateway or in the local pc, can filter DNS replies that pass through it, discarding local addresses.

- Web servers can reject HTTP requests with an unrecognized Host header.

DNS Rebinding Protection

- Web browsers can resist DNS rebinding:
  - Web browsers can implement DNS pinning:
    - the IP address is locked to the value received in the first DNS response.
    - This technique may block some legitimate uses of Dynamic DNS, and may not work against all attacks.
    - However, it is important to fail safe (stop rendering) if the IP address does change, because using an IP address past the TTL expiration can open the opposite vulnerability when the IP address has legitimately changed and the expired IP address may now be controlled by an attacker.
  - The ***NoScript*** extension for Firefox includes ABE, a firewall-like feature inside the browser which in its default configuration prevents attacks on the local network by preventing external webpages from accessing local IP addresses.

Domain Name System Security Extensions

The Domain Name System Security Extensions (DNSSEC) is a suite of Internet Engineering Task Force (IETF) specifications for securing certain kinds of information provided by the Domain Name System (DNS) as used on Internet Protocol (IP) networks.

Guarantees:

adds data origin authentication and data integrity to DNS protocol.

Digitally Sign DNS lookup using Public Key Crypto

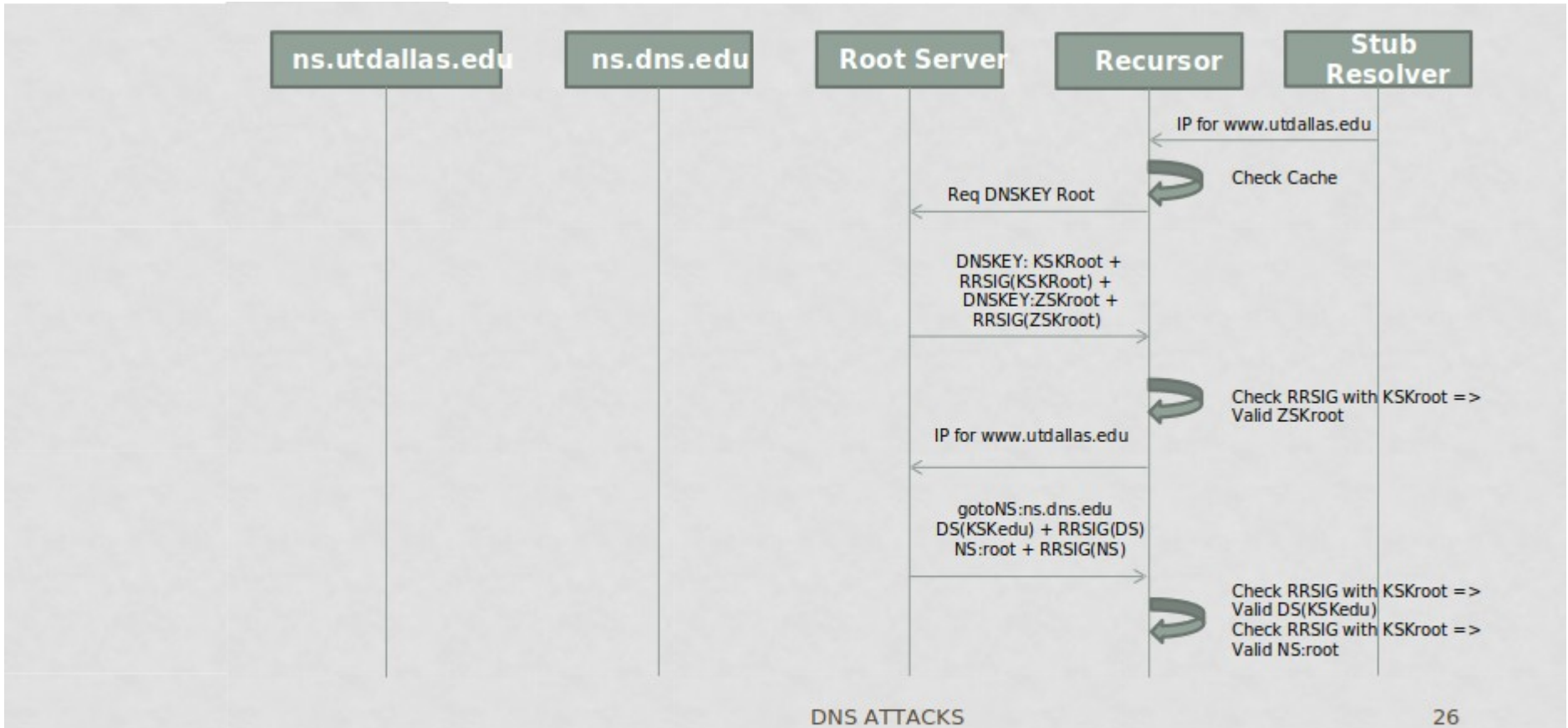
DNSKEY record is authenticated via Chain of Trust starting with trusted root

Its kind of SSL authentication for the DNS.

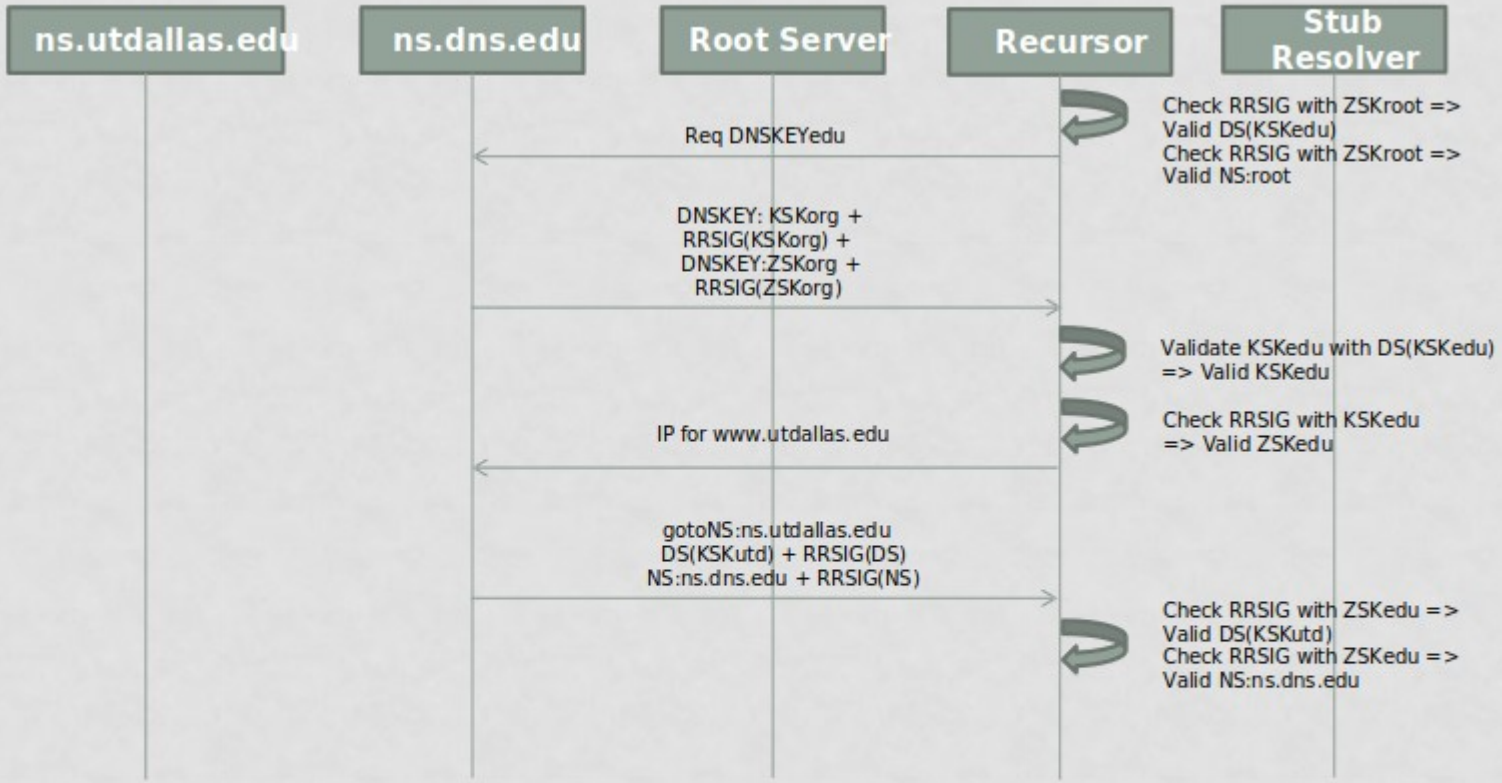
Adds Authenticity of denial of existence

Accomplishes this by signing DNS replies at each step of the way

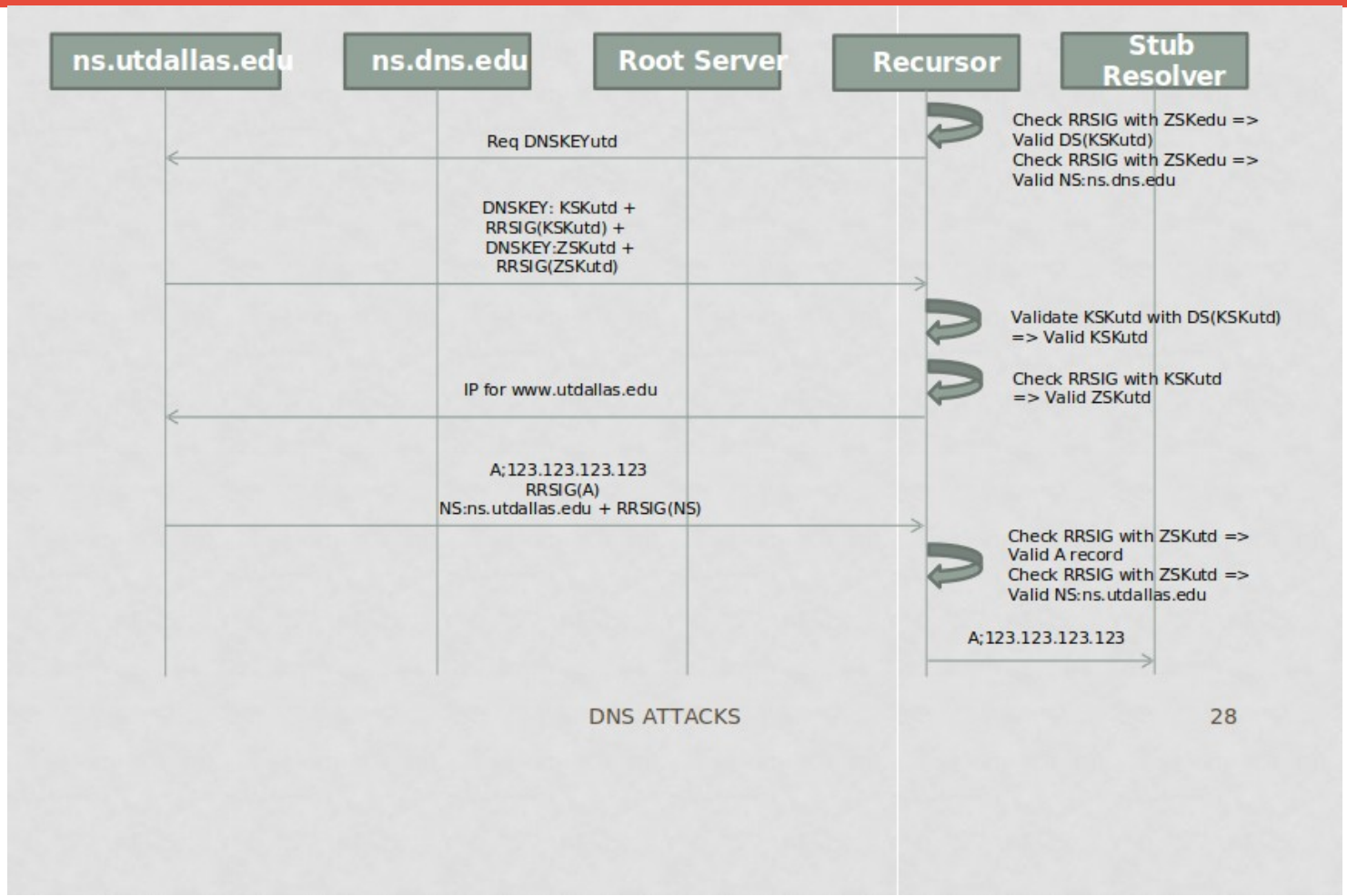
HOW DNSSEC WORKS?



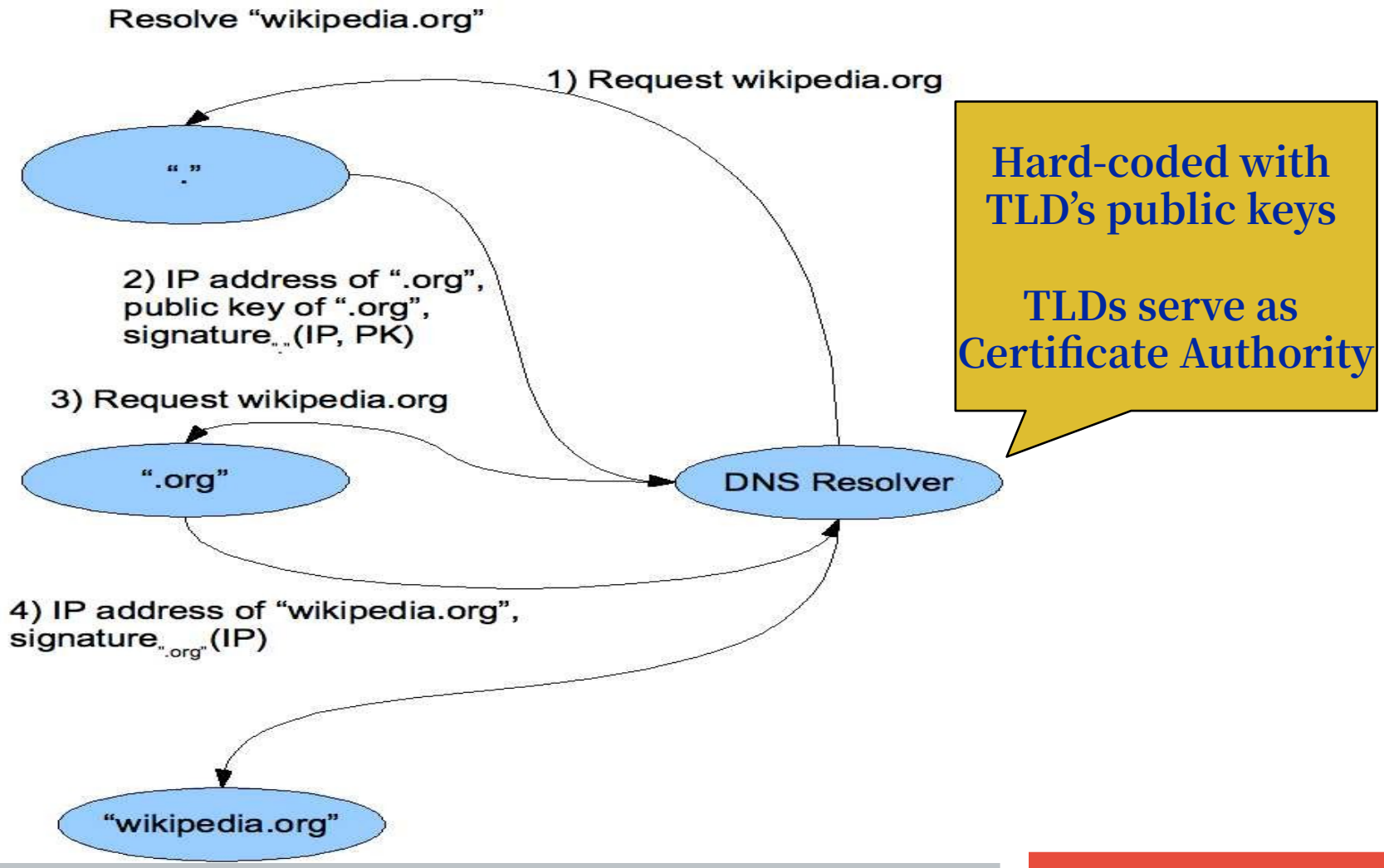
HOW DNSSEC WORKS?



HOW DNSSEC WORKS?



DNS Signing


Resolve “wikipedia.org”
1) Request wikipedia.org
“.”
2) IP address of “.org”,
public key of “.org”,
signature“.”(IP, PK)
3) Request wikipedia.org
“.org”
DNS Resolver
4) IP address of “wikipedia.org”,
signature“.org”(IP)
“wikipedia.org”
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