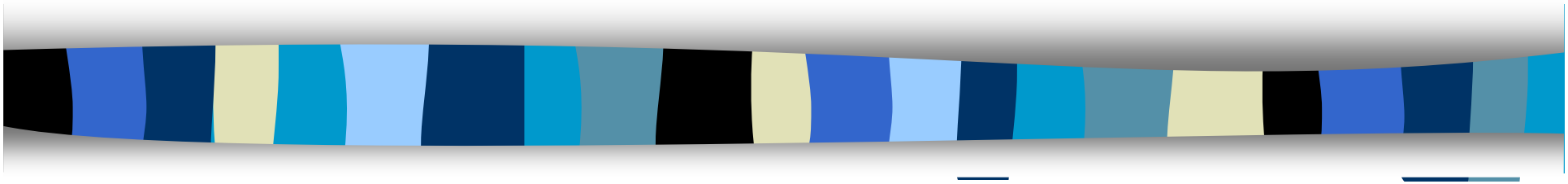
Computer Security
CS 426
Lecture 34



DNS Security

Domain Name System

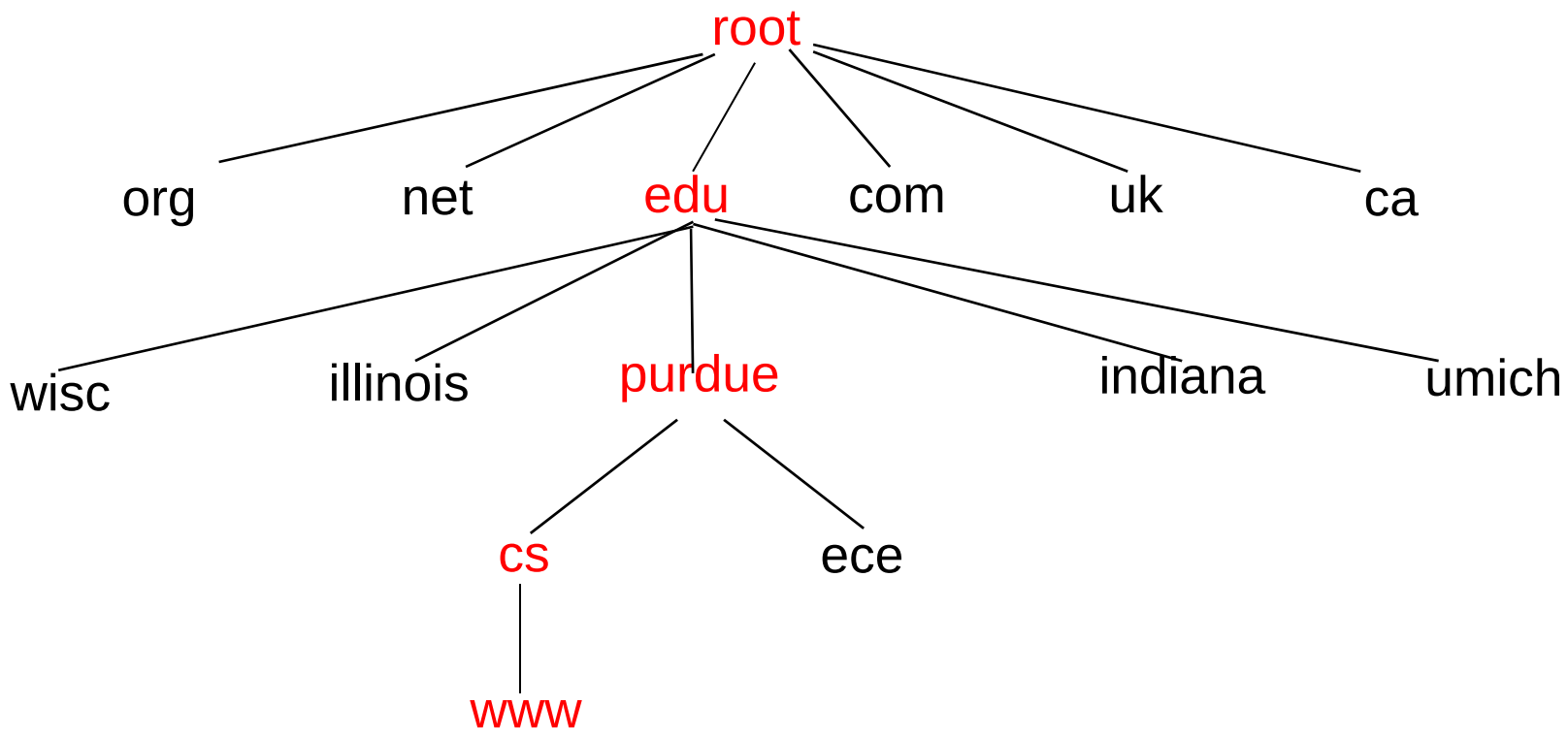
- Translate host names to IP addresses
  - E.g., www.google.com ➔ 74.125.91.103
  - Hostnames are human-friendly
  - IP addresses keep changing
- And back
  - From IP addresses to DNS name
- Analogy: Phone book for the Internet
  - Where they differ?

DNS is a Distributed Database

- Information is stored in a distributed way
- Highly dynamic
- Decentralized authority

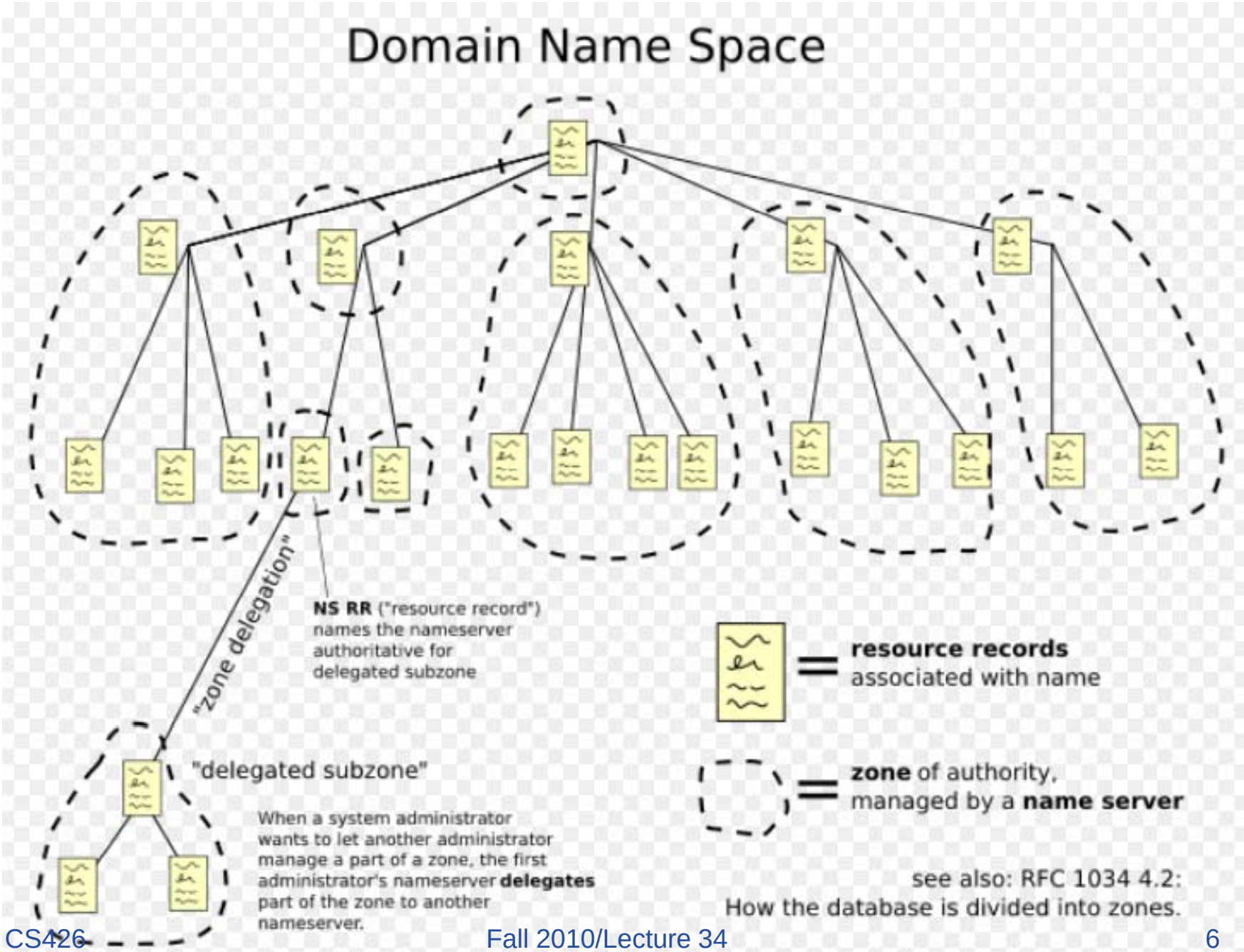
Domain Name System

- Hierarchical Name Space



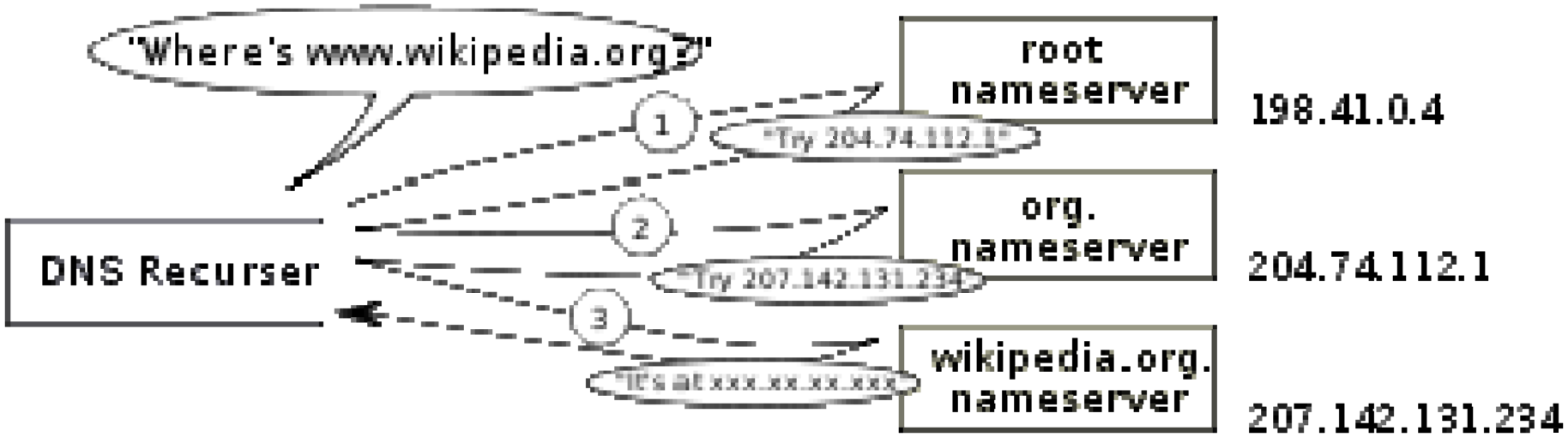
Domain Name Space

- Domain: A node in the DNS tree
- DNS Zones
  - A zone is a group of nodes in the tree, authoritatively served by an *authoritative nameserver*.
  - Each zone may be sub-divided, the parent zone
- Authority servers
  - Answer queries about their zones
  - Provide mapping for leaf nodes or downward delegation
- Hierarchical service
  - *Root name servers* for top-level domains
  - *Authoritative name servers* for subdomains


Domain Name Space
"zone delegation"
NS RR ("resource record") names the nameserver authoritative for delegated subzone
"delegated subzone"
When a system administrator wants to let another administrator manage a part of a zone, the first administrator's nameserver delegates part of the zone to another nameserver.
= resource records associated with name
= zone of authority, managed by a name server
see also: RFC 1034 4.2:
How the database is divided into zones.

DNS Resolver: Recursive Resolver

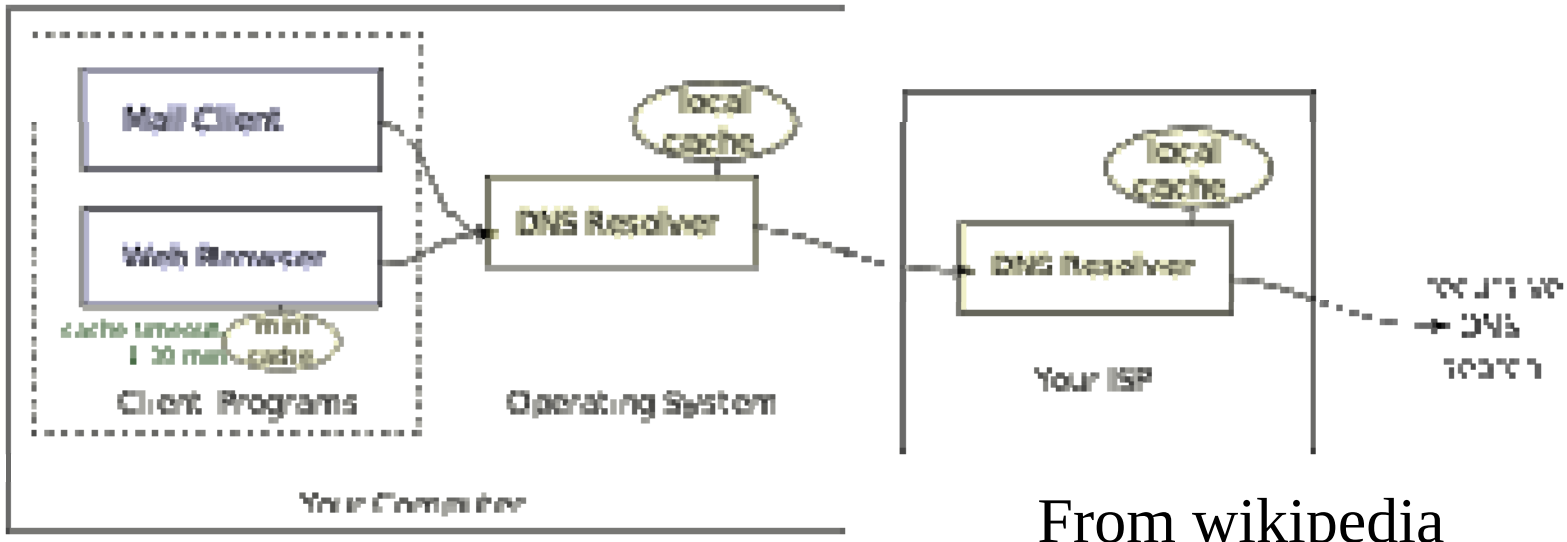
- Recursive resolver
  - Normally thought of as a “DNS server”
  - Accept queries from users, understand the zone hierarchy, interact with the authority servers
  - Cache answers



From wikipedia

DNS Resolver: Stub Resolver

- Stub resolver
  - Not interact with the zone hierarchy
  - Pose basic queries to recursive servers
  - May cache answers
  - PC, client applications



From wikipedia

A Normal DNS Lookup

- Stub resolver asks “www.google.com”
- Assume no previous results cached at the recursive resolver
- Query the root servers (authority servers for “.” zone)
  - Answer: downward delegation
  - com NS a.gtld-servers.net
  - a.gtld-servers.net A 74.292.124.59

NS: Name Server

A: Address

- Query the “.com” zone authority servers
  - Answer: downward delegation
  - google.com NS ns1.google.com
  - ns1.google.com A 122.45.212.57

A Normal DNS Lookup (cont’)

- Query the “google.com” zone authority servers
  - Answer: www.google.com A 24.122.49.76
- The answer is returned to the stub resolver
- The results are cached by the recursive resolver

Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Useful for finding servers as well as addresses
    - NS records for domains
- Negative results are cached
  - Save time for nonexistent sites, e.g. misspelling
- Cached data periodically times out
  - Each record has a TTL field

Inherent DNS Vulnerabilities

- Users/hosts typically trust the host-address mapping provided by DNS
  - What bad things can happen with wrong DNS info?
- DNS resolvers trust responses received after sending out queries
  - How to attack?
- Responses can include DNS information unrelated to the query
- Obvious problems
  - No authentication for DNS responses

Pharming

- Exploit DNS poisoning attack
  - Change IP addresses to redirect URLs to fraudulent sites
  - Potentially more dangerous than phishing attacks
  - No email solicitation is required
- 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"

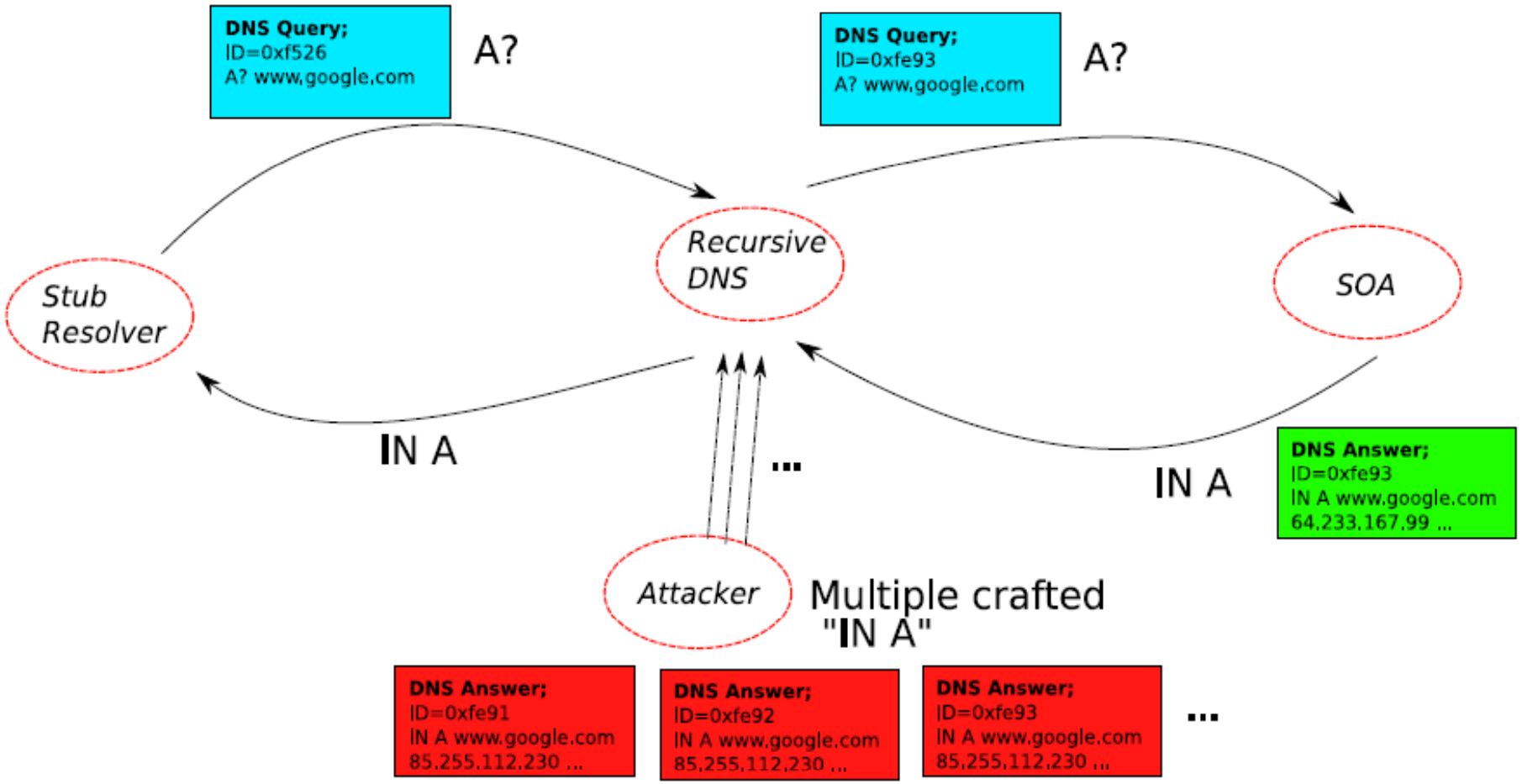
DNS cache poisoning (Vulnerability 1) (Chris Schuba in 1993)

- DNS resource records (see RFC 1034)
  - An “A” record supplies a host IP address
  - A “NS” record supplies name server for domain
- Example
  - 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 yahoo through cache goes to 1.2.3.4

Defense Using The Bailiwicks Rules

- The bailiwick system prevents foo.com from declaring anything about com, or some other new TLD, or www.google.com
- Using the bailiwicks rules
  - The root servers can return any record
  - The com servers can return any record for com
  - The google.com servers can return any record for google.com

DNS cache poisoning: Racing to Respond First



DNS Cache Poisoning

- Attacker wants his IP address returned for a DNS query
- When the resolver asks ns1.google.com for www.google.com, the attacker could reply first, with his own IP
- What is supposed to prevent this?
- Transaction ID
  - 16-bit random number
  - The real server knows the number, because it was contained in the query
  - The attacker has to guess

DNS cache poisoning (Vulnerability 2)

- Responding before the real nameserver
  - An attacker can guess when a DNS cache entry times out and a query has been sent, and provide a fake response.
  - The fake response will be accepted only when its 16-bit transaction ID matches the query
  - CERT reported in 1997 that BIND uses sequential transaction ID and is easily predicted
    - fixed by using random transaction IDs

DNS cache poisoning (Vulnerability 3)

- Improve the chance of responding before the real nameserver (discovered by Vagner Sacramento in 2002)
  - Have many (say hundreds of) clients send the same DNS request to the name server
    - Each generates a query
  - Send hundreds of reply with random transaction IDs at the same time
  - Due to the Birthday Paradox, the success probability can be close to 1

DNS cache poisoning (Vulnerability 4)

- Kaminsky Attack
  - Big security news in summer of 2008
  - DNS servers worldwide were quickly patched to defend against the attack

- In previous attacks, when the attacker loses the race, the record is cached, with a TTL.
  - Before TTL expires, no attack can be carried out
  - Posining address for google.com in a DNS server is not easy.

Guess the ID

- Early versions of DNS servers deterministically incremented the ID field
- Vulnerabilities were discovered in the random ID generation
  - Weak random number generator
  - The attacker is able to predict the ID if knowing several IDs in previous transactions
- Birthday attack
  - Force the resolver to send many identical queries, with different IDs, at the same time
  - Increase the probability of making a correct guess

What is New in the Kaminsky Attack?

- The bad guy does not need to wait to try again
- The bad guy asks the resolver to look up www.google.com
  - If the bad guy lost the race, the other race for www.google.com will be suppressed by the TTL
- If the bad guy asks the resolver to look up 1.google.com, 2.google.com, 3.google.com, and so on
  - Each new query starts a new race
- Eventually, the bad guy will win
  - he is able to spoof 183.google.com
  - So what? No one wants to visit 183.google.com

Kaminsky-Style Poisoning

- A bad guy who wins the race for "183.google.com" can end up stealing "www.google.com" as well

- The malicious response
  - google.com NS www.google.com
  - www.google.com A 6.6.6.6
  - OR
  - google.com NS ns.badguy.com

Kaminsky-Style Poisoning (cont')

- Can start anytime; no waiting for old good cached entries to expire
- No "wait penalty" for racing failure
- The attack is only bandwidth limited

- Defense (alleviate, but not solve the problem)
  - Also randomize the UDP used to send the DNS query, the attacker has to guess that port correctly as well.

DNS Poisoning Defenses

- Difficulty to change the protocol
  - Protocol stability (embedded devices)
  - Backward compatible
- Long-term
  - Cryptographic protections
    - E.g., DNSSEC, DNSCurve
  - Require changes to both recursive and authority servers
  - A multi-year process
- Short-term
  - Only change the recursive server
  - Easy to adopt

Short-Term Defenses

- Source port randomization
  - Add 16-bits entropy
  - resource intensive (select on a potentially large pool of ports)
  - NAT could de-randomize the port
- DNS 0x20 encoding
  - From Georgia tech, CCS 2008
- Tighter logic for accepting responses

DNS-0x20 Bit Encoding

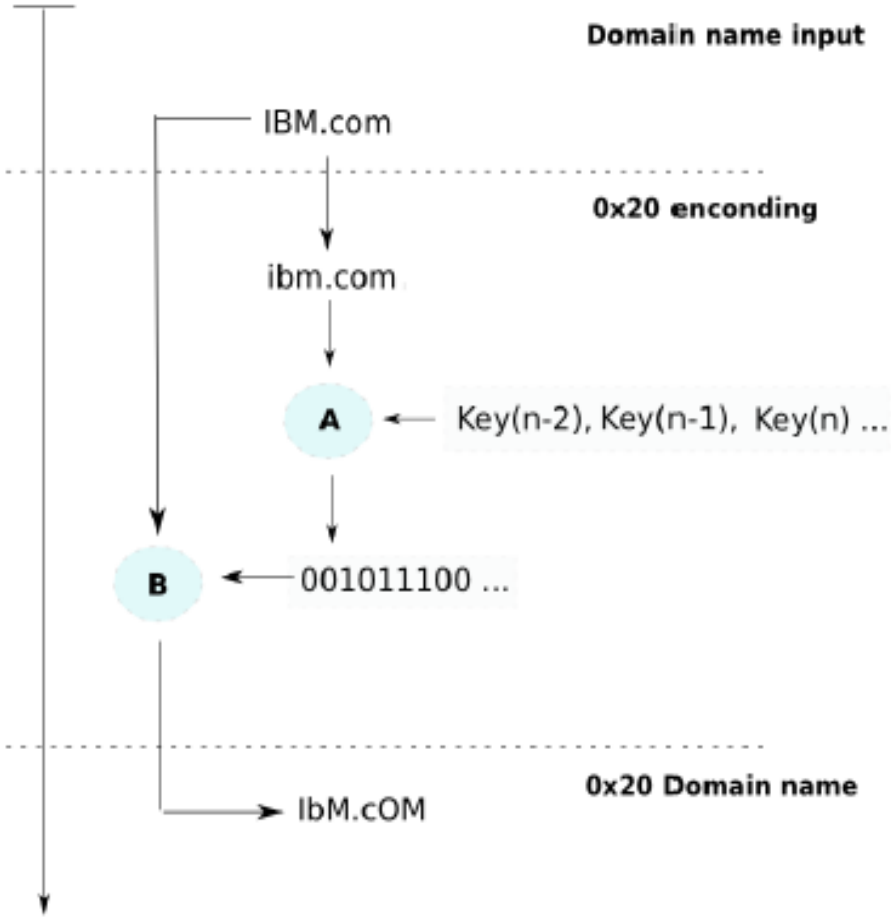
- DNS labels are case insensitive
- Matching and resolution is entirely case insensitive
- A resolver can query in any case pattern
  - E.g., WwW.ExAmpLe.cOM
  - It will get the answer for www.example.com

DNS-0x20 DNS Encoding (cont')

- A DNS response contains the query being asked
- When generating the response, the query is copied from the request exactly into the response
  - The case pattern of the query is preserved in the response
- Open source implementations exhibit this behavior
  - The DNS request is rewritten in place
- The mixed pattern of upper and lower case letters constitutes a channel, which can be used to improve DNS security
  - Only the real server knows the correct pattern

Query Encoding

- Transforms the query into all lowercase
- Encrypt the query with a key shared by all queries on the recursive server (A)
- The cipher text is used to encode the query
  - 0: buff[i] |= 0x20
  - 1: buff[i] &= 0x20



DNS-0x20 Encoding Analysis

- Do existing authority servers preserve the case pattern?
  - Scan 75 million name servers, 7 million domains

- Only 0.3% mismatch observed

Type	Mismatch	Mismatch pct.	Domain scanned
.com TLD	15451	0.327%	4786993
.net TLD	4437	0.204%	2168352

DNS-0x20 Encoding Analysis (cont')

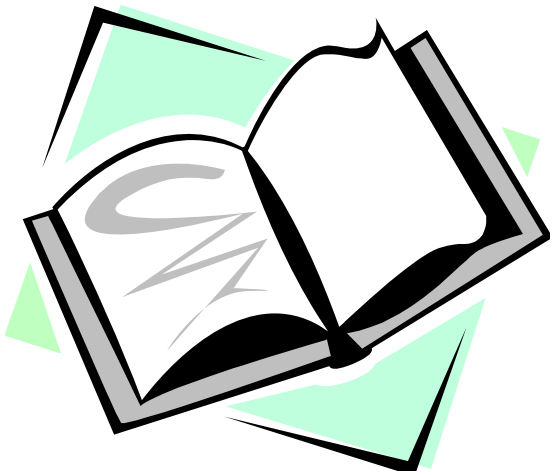
- Not every character is 0x20 capable
- Improve the forgery resistance of DNS messages only in proportion to the number of upper or lower case characters
  - cia.gov 6-bit entropy
  - licensing.disney.com 12-bit entropy
  - 163.com 3-bit entropy
- TLDs are also vulnerable to Kaminsky-style attacks; but they have few 0x20-capable bits

Other DNS attacks

- Attacking home routers/gateways
- Incidence in Mexica in 2008
  - an email sent to users
  - email include URL (HTTP requests) to the HTTP-based interface of wireless routers
  - using the default password to reconfigure the router/gateway

Readings for This Lecture

- Optional:
  - **An Illustrated Guide to the Kaminsky DNS Vulnerability**
  - Dan Kaminsky's Black Hat presentation (PowerPoint)



Coming Attractions …

- Network Security Defenses