

CS 4910: Intro to Computer Security

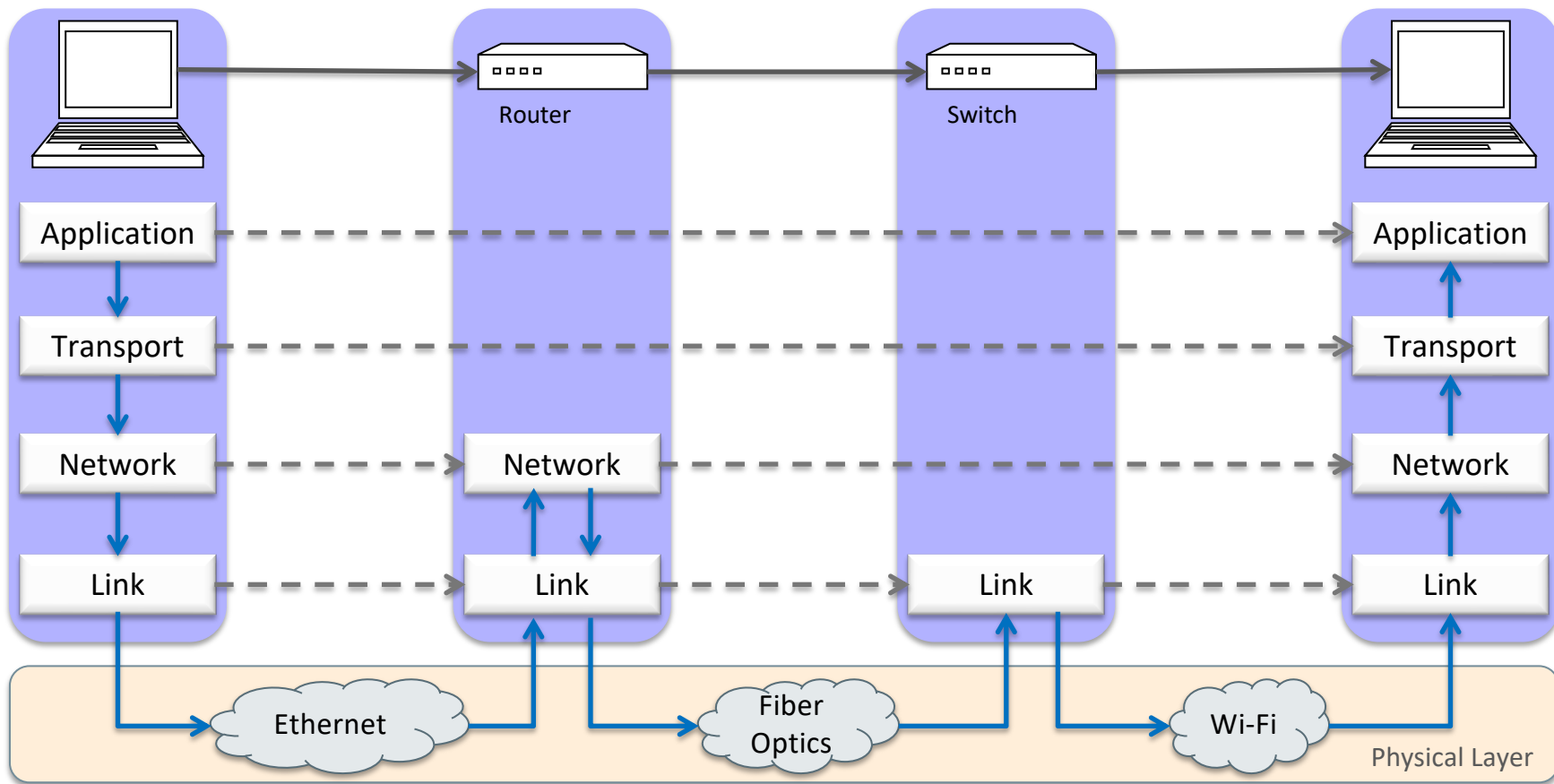
Network Security III:
DNS Attack

Instructor: Xi Tan

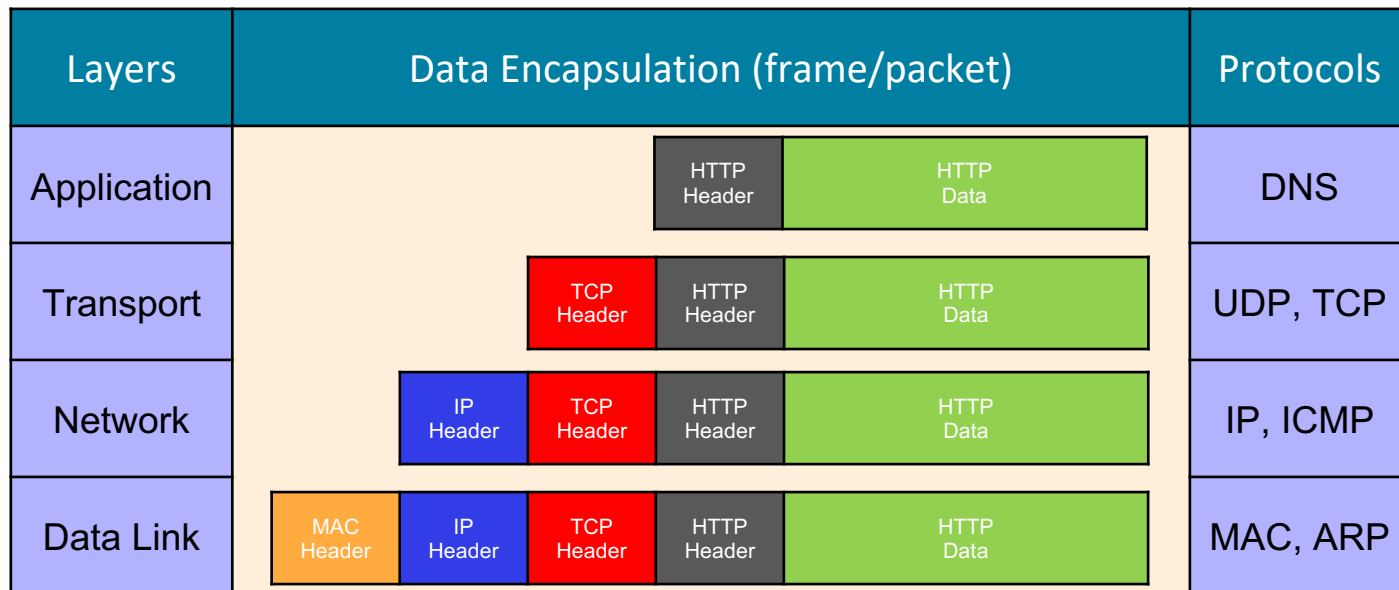
Updates

- Lab 2:
 - Task 1: Packet sniffing and spoofing
 - Task 2: Not required
 - **Deadline: 3/31**
- Homework 3
 - **Deadline: 04/07**
- Research Paper:
 - **Deadline: 04/14**

Recall: Network Layers



So far ...



?

TCP flood

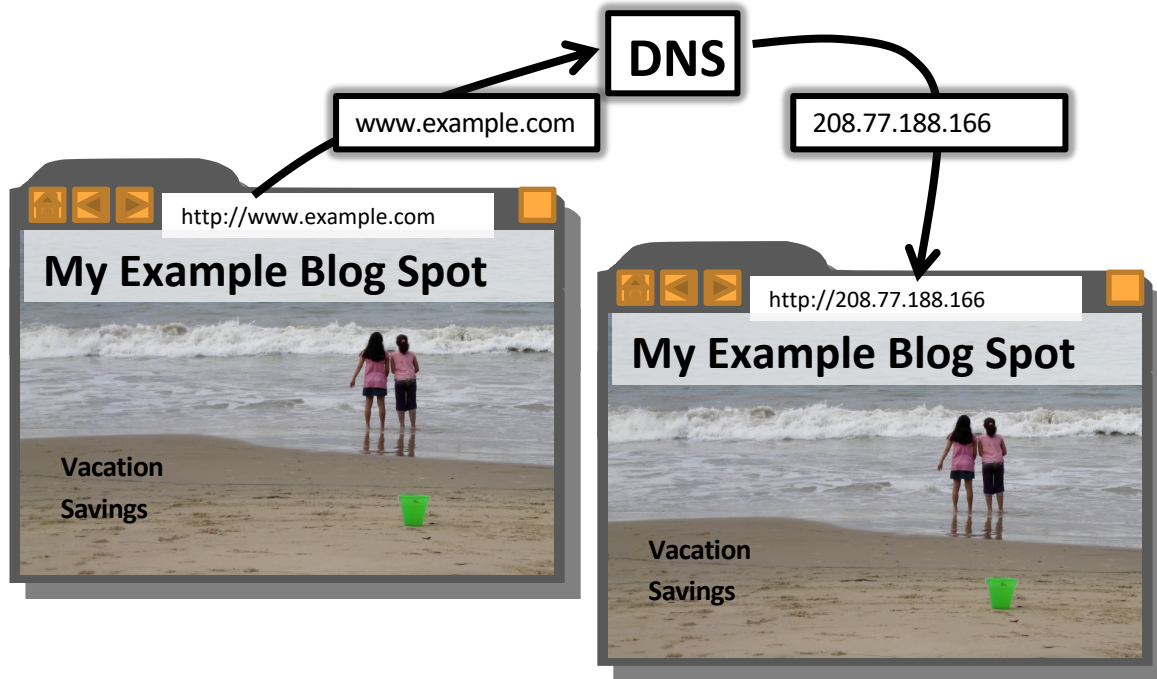
IP spoofing,
ICMP floodMAC spoofing,
ARP spoofing

Next

- Computer Network Concepts
- Network Attacks
 - MAC Spoofing, ARP Spoofing
 - IP Spoofing
 - Denial of Service
 - DNS Cache Poisoning
- Network Security

Domain Name System (DNS)

- The **domain name system (DNS)** is an application-layer protocol for mapping domain names to IP addresses



DNS

- DNS provides a distributed database over the internet that stores **various resource records**, including:
 - Address (A) record: IP address associated with a host name
 - Mail exchange (MX) record: mail server of a domain
 - Name server (NS) record: authoritative server for a domain

Resource records [\[edit \]](#)

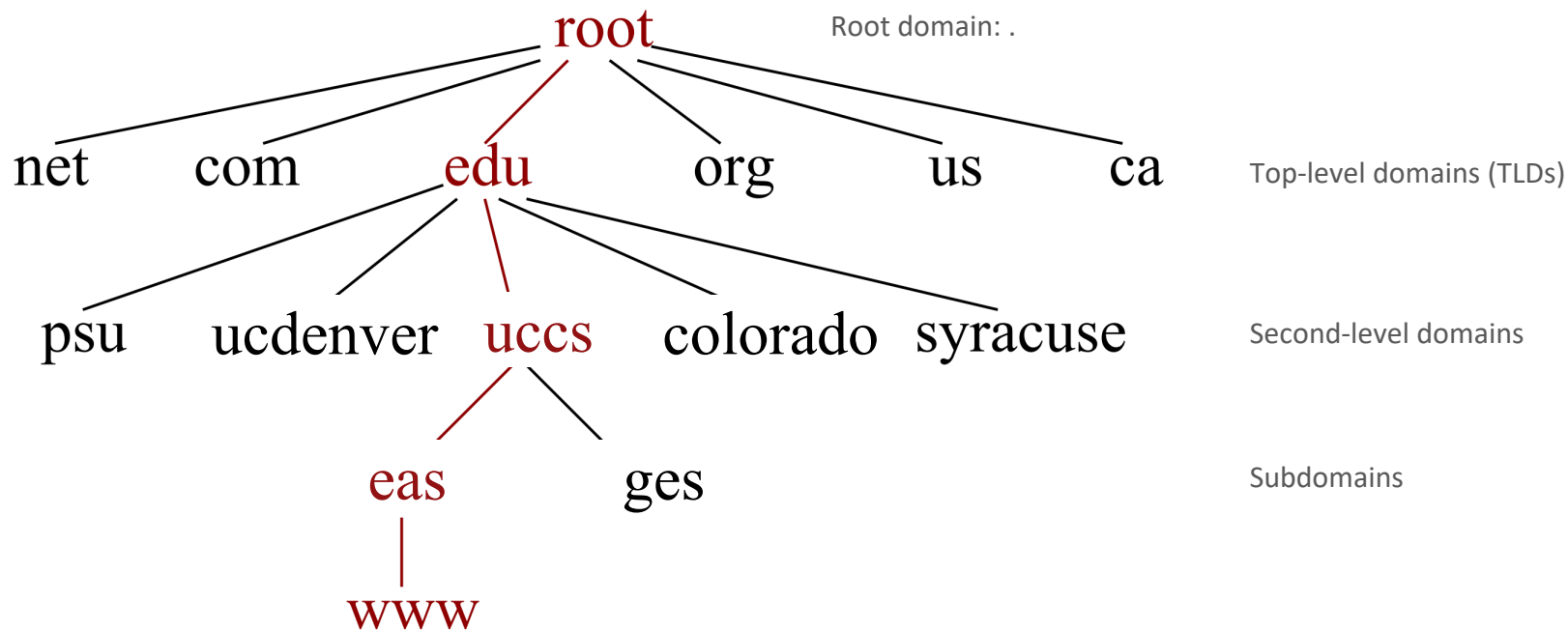
Type ↕	Type id. (decimal) ↕	Defining RFC ↕	Description ↕	Function ↕
A	1	RFC 1035 ^[1]	Address record	Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host, but it is also used for DNSBLs , storing subnet masks in RFC 1101, etc.
AAAA	28	RFC 3596 ^[2]	IPv6 address record	Returns a 128 -bit IPv6 address, most commonly used to map hostnames to an IP address of the host.
AFSDB	18	RFC 1183	AFS database record	Location of database servers of an AFS cell. This record is commonly used by AFS clients to contact AFS cells outside their local domain. A subtype of this record is used by the obsolete DCE/DFS file system.
APL	42	RFC 3123	Address Prefix List	Specify lists of address ranges, e.g. in CIDR format, for various address families. Experimental.

List of DNS records from https://en.wikipedia.org/wiki/List_of_DNS_record_types

DNS

- DNS tree

- the name space is hierarchical



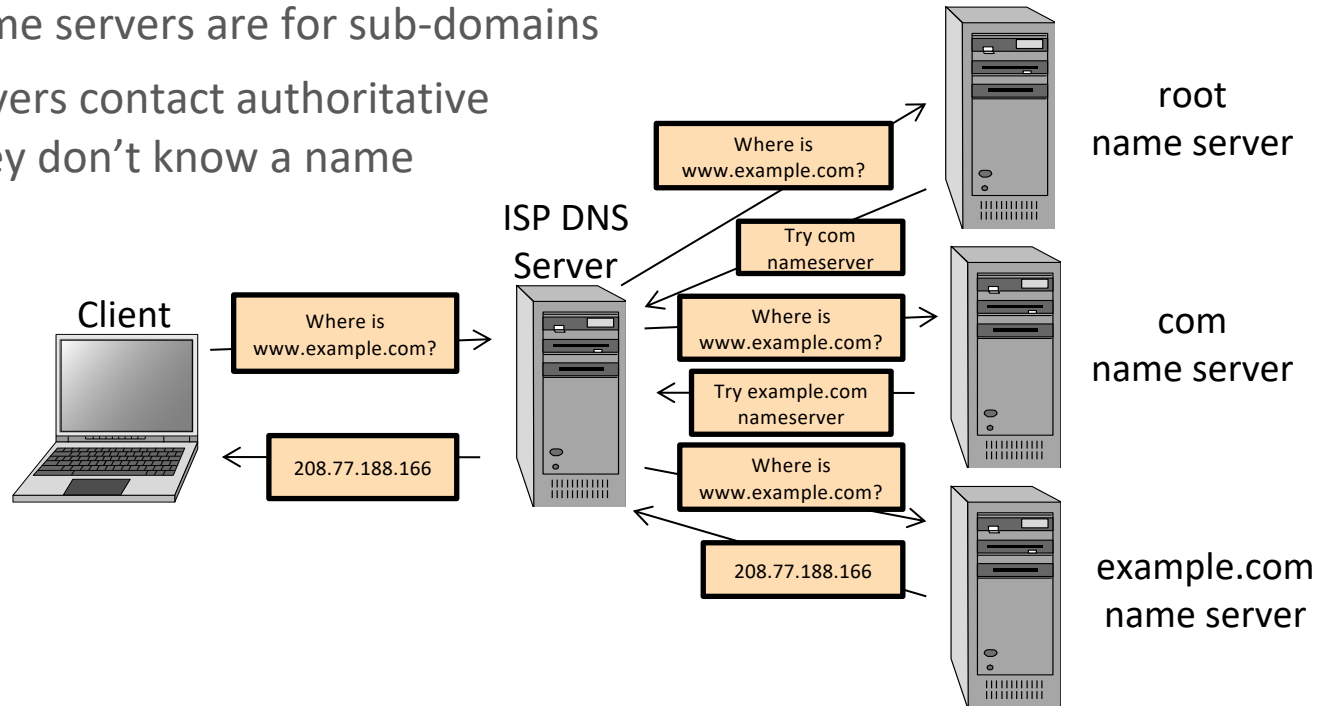
Top Level Domains

- Started in 1984
- Originally supposed to be named by function
 - .com for commercial websites, .mil for military
- Eventually agreed upon unrestricted top-level domain (TLD) for .com, .net, .org, .info
- In 1994 started allowing **country** TLDs such as .it, .us
- Tried to move back to **hierarchy** of purpose in 2000 with creation of .aero, .museum, etc.
- Two primary types of TLD:
 - Generic top-level domains: .com, .net, .edu, .org
 - Country-code top-level domains: .au(Australia), .cn(China), .it (Italy)

DNS

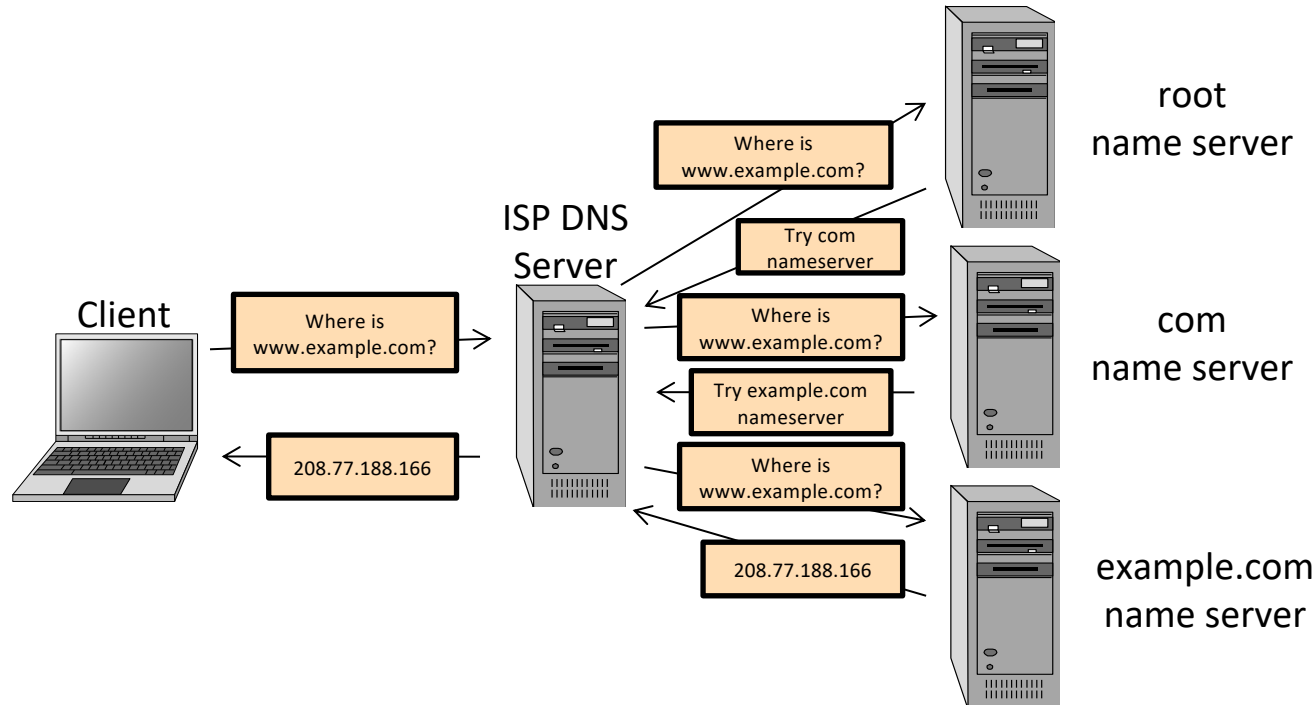
- Hierarchical service

- **root** name servers are for top-level domains
- authoritative name servers are for sub-domains
- **local** name resolvers contact authoritative servers when they don't know a name

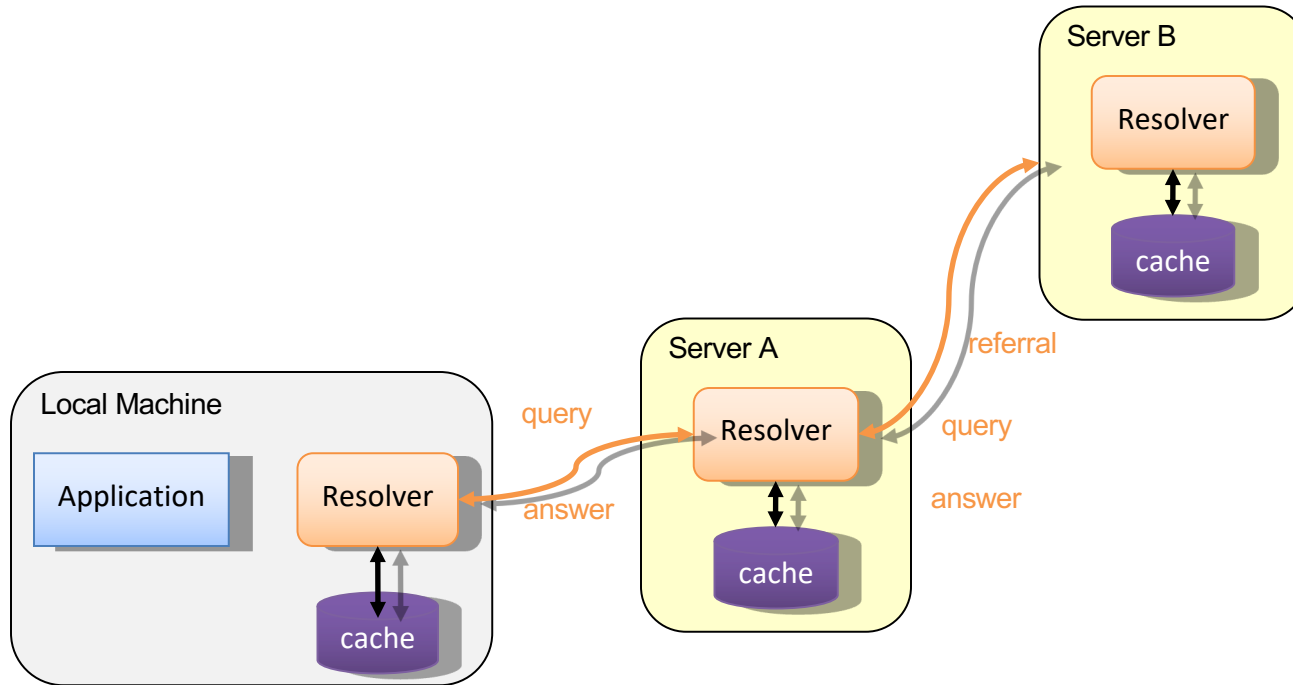


Name Resolution

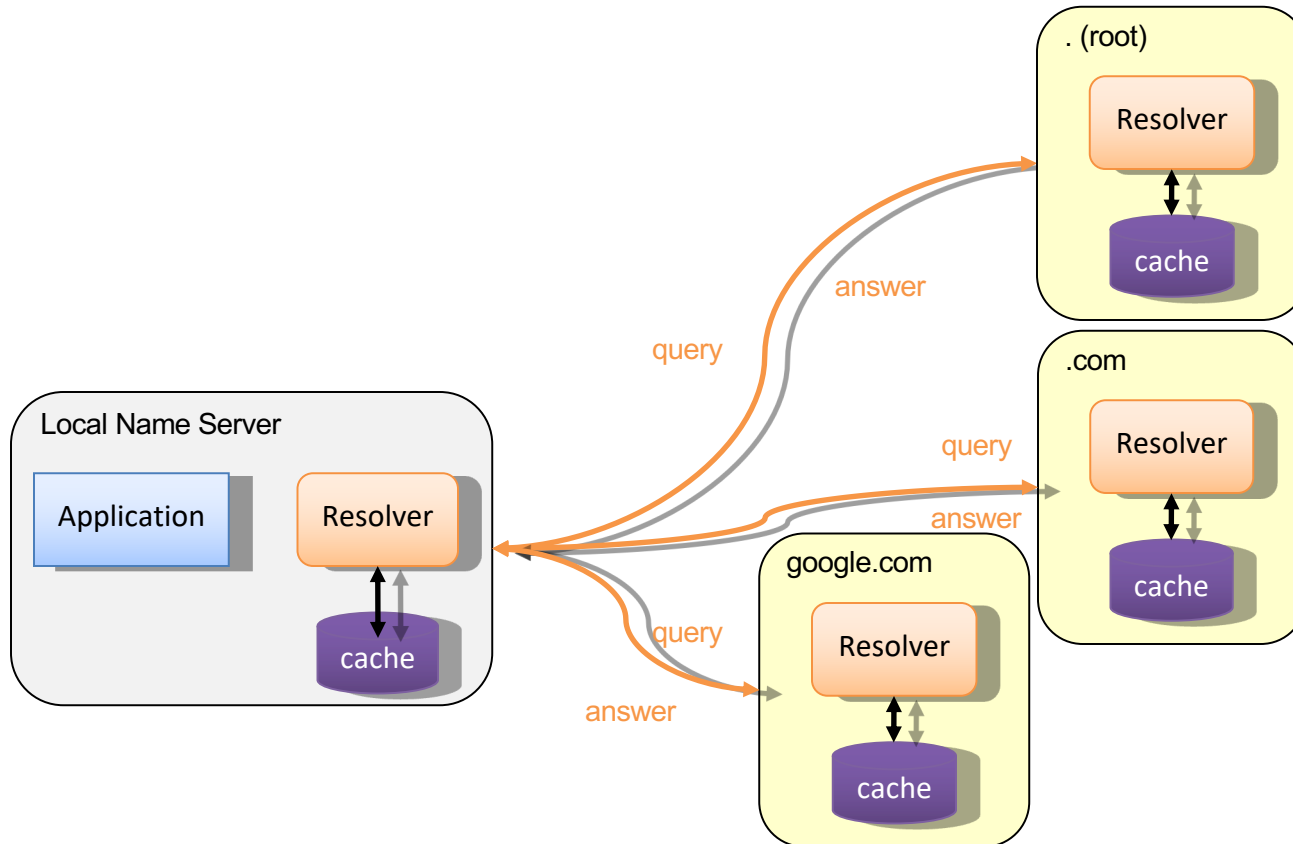
- **Zone**: collection of connected nodes with the same authoritative DNS server
- **Resolution method** when answer not in cache.



Recursive Name Resolution



Iterative Name Resolution

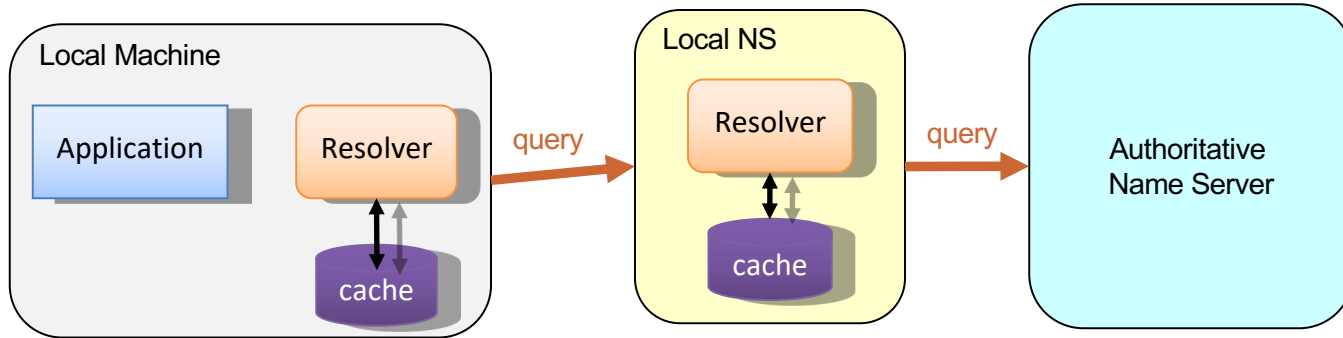


DNS Caching

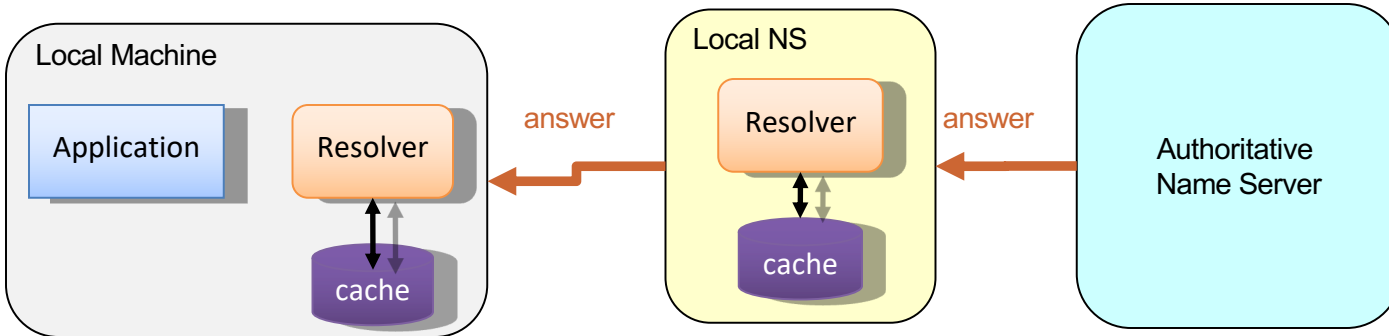
- DNS responses are cached
 - quick response for repeated translations
 - useful for finding servers as well as addresses
- negative results are cached
 - save time for nonexistent sites, e.g., misspelling
- cached data periodically time out

DNS Caching

Step 1: query yourdomain.org

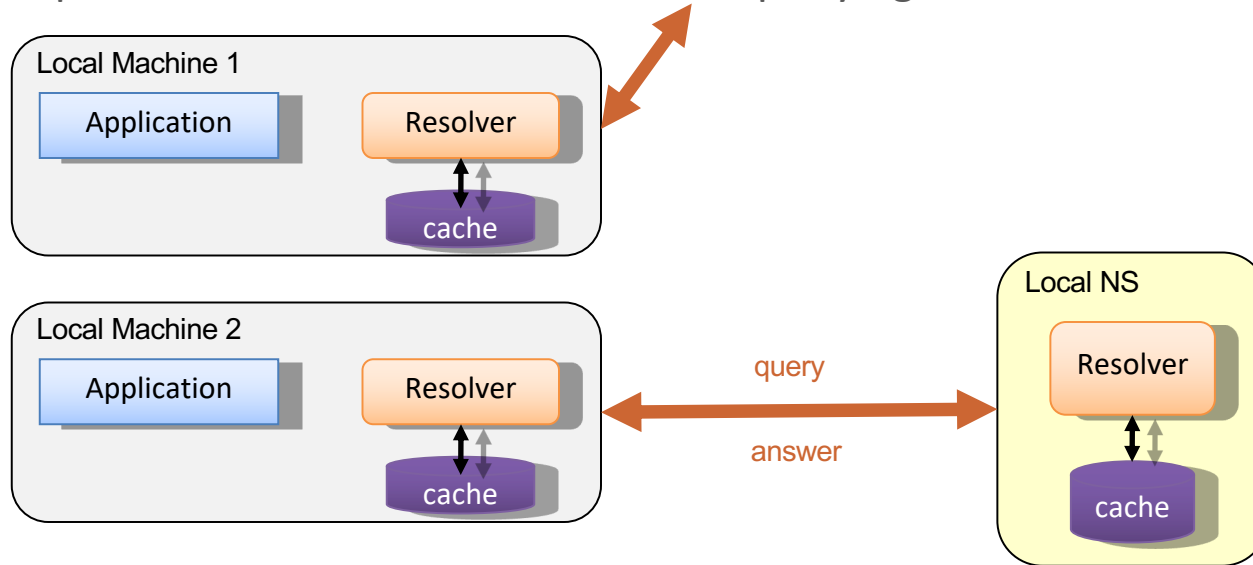


Step 2: receive reply and cache at local name server and host



DNS Caching (con'd)

Step 3: use cached results rather than querying the ANS



Step 4: Evict cache entries upon Time-To-Live (TTL) expiration

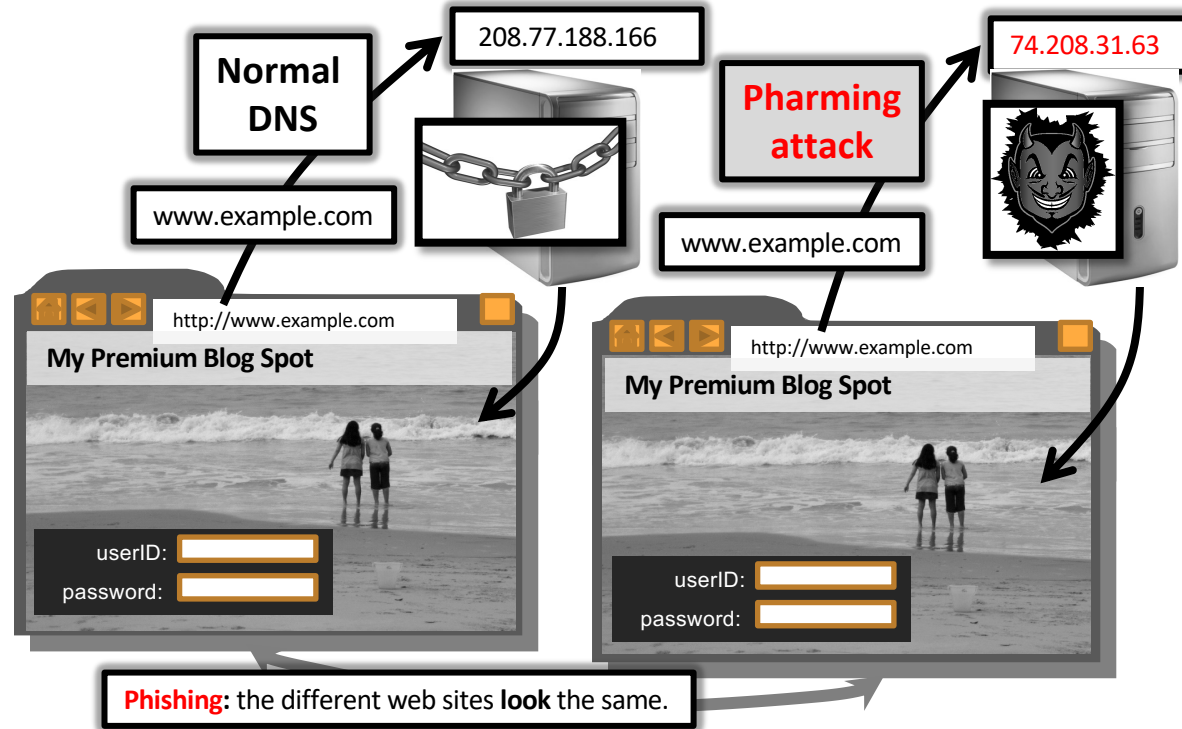
- Common TTL values might be 3600 (1 hour), 86400 (1 day), or even as low as 300 (5 minutes).
- Once the TTL expires, the cached result is **discarded**, and any subsequent request for the same domain will trigger a **new** DNS query to get the updated information.

DNS Cache Poisoning

- DNS is susceptible to **cache poisoning attacks**
 - Basic idea: change IP address in cache to redirect URLs to fraudulent sites
 - this attack is called **pharming/DNS hijacking**
 - example
 - www.yahoo.com NS ns.evil.org (delegate to evil.org)
 - ns.evil.org A 1.2.3.4 (address for evil.org)
 - if resolver looks up www.yahoo.com, the address 1.2.3.4 will be returned
 - **root cause**: 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
 - In 2002, most major DNS software used sequential numbers of query IDs
 - Accepts unsolicited DNS records

DNS Cache Poisoning

- Changing IP associated with a server maliciously:



DNS Cache Poisoning

- DNS cache poisoning

- the problem is DNS messages are **NOT** authenticated
- some DNS poisoning attacks in the past
 - in January 2005, the address of a large ISP Panix was redirected to a site in Australia
 - in November 2004, Google and Amazon users were sent to Med Network Inc., an online pharmacy

- There are also attacks on DNS reverse address lookup and DNS implementations

- example: reverse query buffer overrun in BIND releases 4.9 and 8
 - could gain root access, abort DNS service

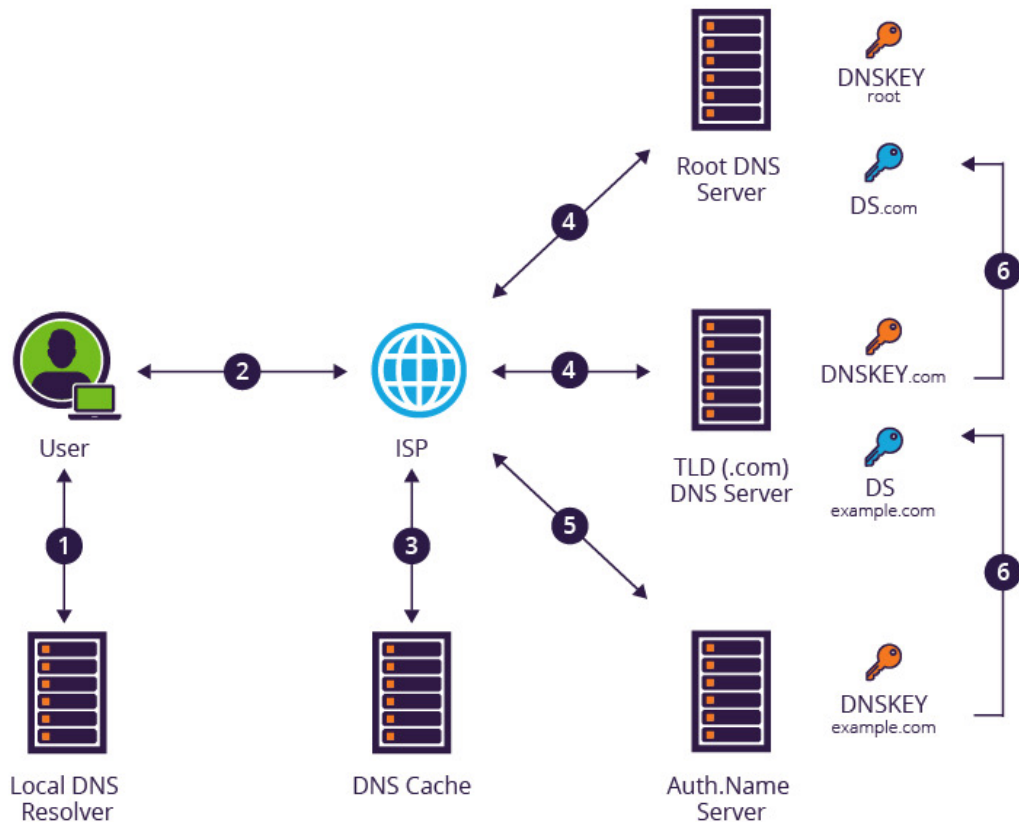
DNS Cache Poisoning Prevention

- Difficult to prevent
 - Relying on a 16-bit number to verify the DNS response
- Possible solutions:
 - Use random identifiers for queries
 - Always check identifiers
 - Port randomization for DNS requests
 - Deploy DNSSEC

DNS Cache Poisoning Prevention - DNSSEC

- Domain Name System Security Extensions (DNSSEC) was developed to protect integrity of DNS records
 - 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

DNSSEC



Example of DNSSEC validation process from <https://www.imperva.com/learn/application-security/dnssec/>

Other Attacks

- Session hijacking attacks
 - host-based session hijacking
 - with root privileges can read and write to local terminal devices
 - network-based session hijacking
 - often performed against TCP
- What harm can be done
 - data injection into unencrypted server-to-server traffic such as email exchange, DNS zone transfers, etc.
 - data injection into unencrypted client-to-server traffic such as ftp file downloads and http responses
 - denial of service attacks such as resetting a connection

Other Attacks

- **TCP session hijacking**
 - each TCP connection has an associated state
 - client and server IP and port numbers, sequence numbers
 - the problems is that it is not difficult to guess state
 - port numbers can be standard
 - sequence numbers are often chosen in a predictable way
- **TCP sequence numbers**
 - need high degree of unpredictability
 - attacker who knows initial sequence numbers and amount of traffic sent can estimate likely current values
 - send a flood of packets with likely sequence numbers




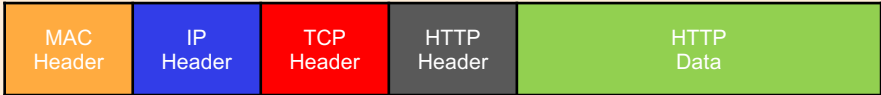
Other Attacks

- TCP sequence numbers (cont.)
 - packets can be injected into existing connection
 - some implementations are vulnerable
 - DoS vulnerability
 - if attacker can guess sequence numbers for an existing connection, it can send a RST packet to close connection (DoS)
 - naively, success probability is $1/2^{32}$ (32-bit numbers)
 - most systems allow for a large window of acceptable sequence numbers resulting in much higher success probability
 - attack is most effective against long lived connections such as BGP

Defenses

- Cryptographic network protection
 - protocol level solutions
 - adding authentication to protocols would solve many problems (various types of spoofing and poisoning)
 - perceived as too expensive for current internet speeds/volumes
 - solutions at network layer
 - use cryptographically random initial sequence numbers, IPsec
 - can protect against session hijacking/data injection and DoS using session resets
 - solutions above transport layer
 - tools such as TLS and SSH
 - protect against session hijacking, but not against RST-based DoS

Network Attacks: Summary

Layers	Data Encapsulation (frame/packet)				Protocols	Attacks
Application					DNS	DNS cache poisoning
Transport					TCP	TCP flood
Network					IP, ICMP	IP spoofing, ICMP flood
Data Link					MAC, ARP	MAC spoofing, ARP spoofing

Next

- Network Security
 - Network Firewalls
 - Intrusion Detection Systems