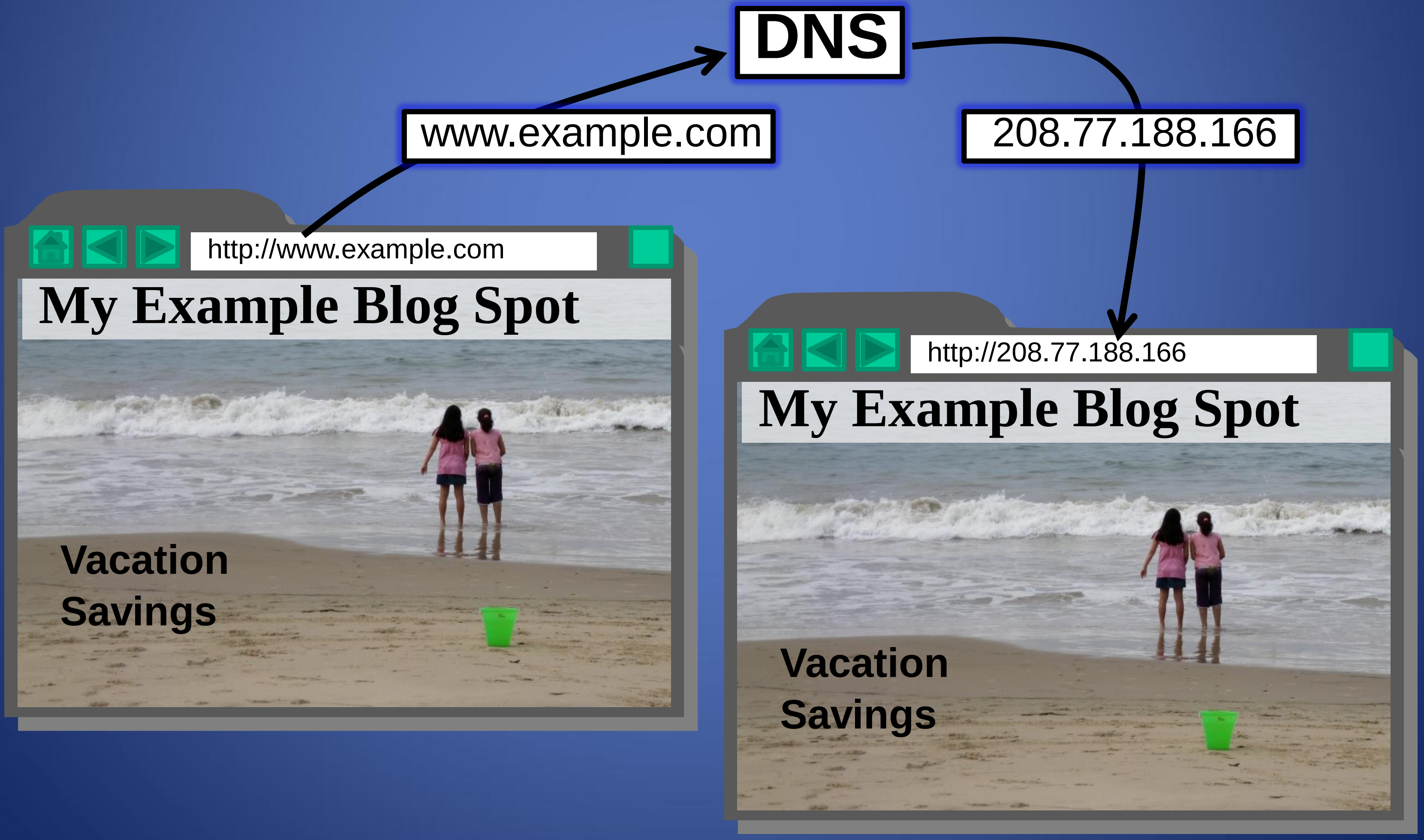
Computer Networks: Domain Name System

Domain Name System

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



DNS

www.example.com

208.77.188.166

http://www.example.com

My Example Blog Spot

Vacation Savings

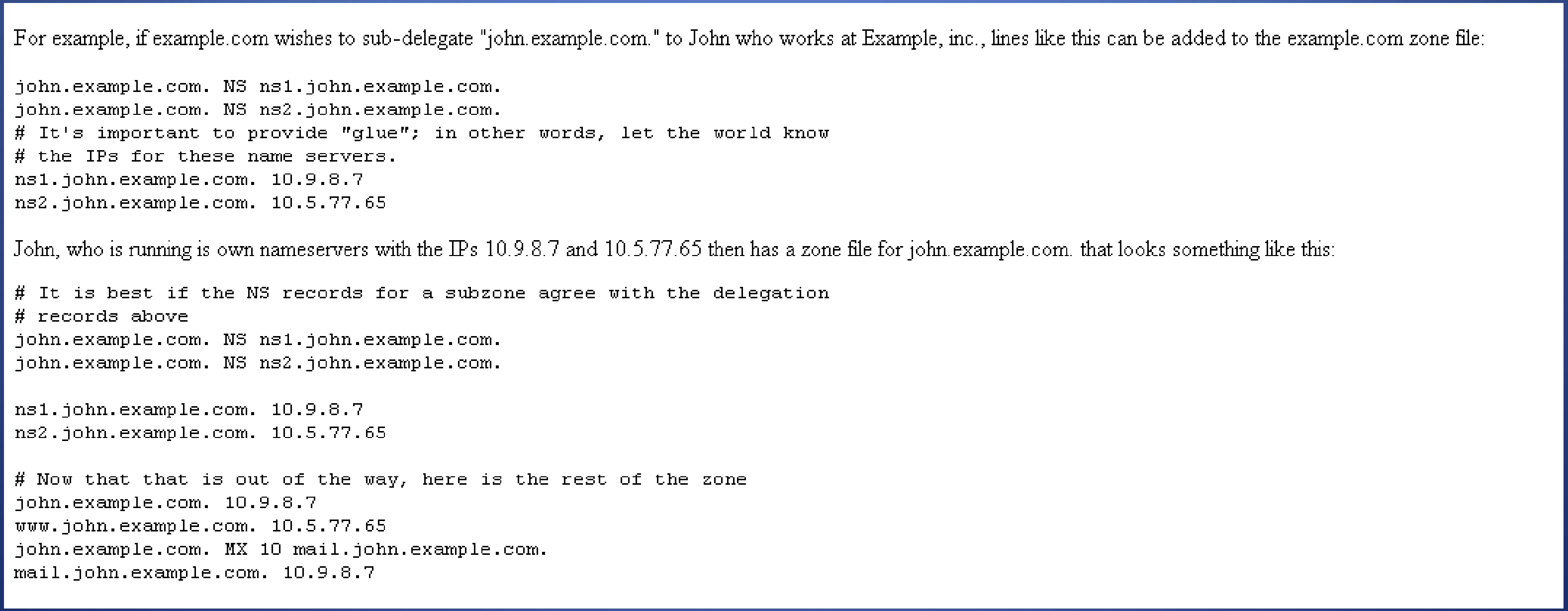
http://208.77.188.166

My Example Blog Spot

Vacation Savings

Domain Name System

- 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



For example, if example.com wishes to sub-delegate "john.example.com." to John who works at Example, inc., lines like this can be added to the example.com zone file:

```
john.example.com. NS ns1.john.example.com.
john.example.com. NS ns2.john.example.com.
# It's important to provide "glue"; in other words, let the world know
# the IPs for these name servers.
ns1.john.example.com. 10.9.8.7
ns2.john.example.com. 10.5.77.65
```

John, who is running is own nameservers with the IPs 10.9.8.7 and 10.5.77.65 then has a zone file for john.example.com. that looks something like this:

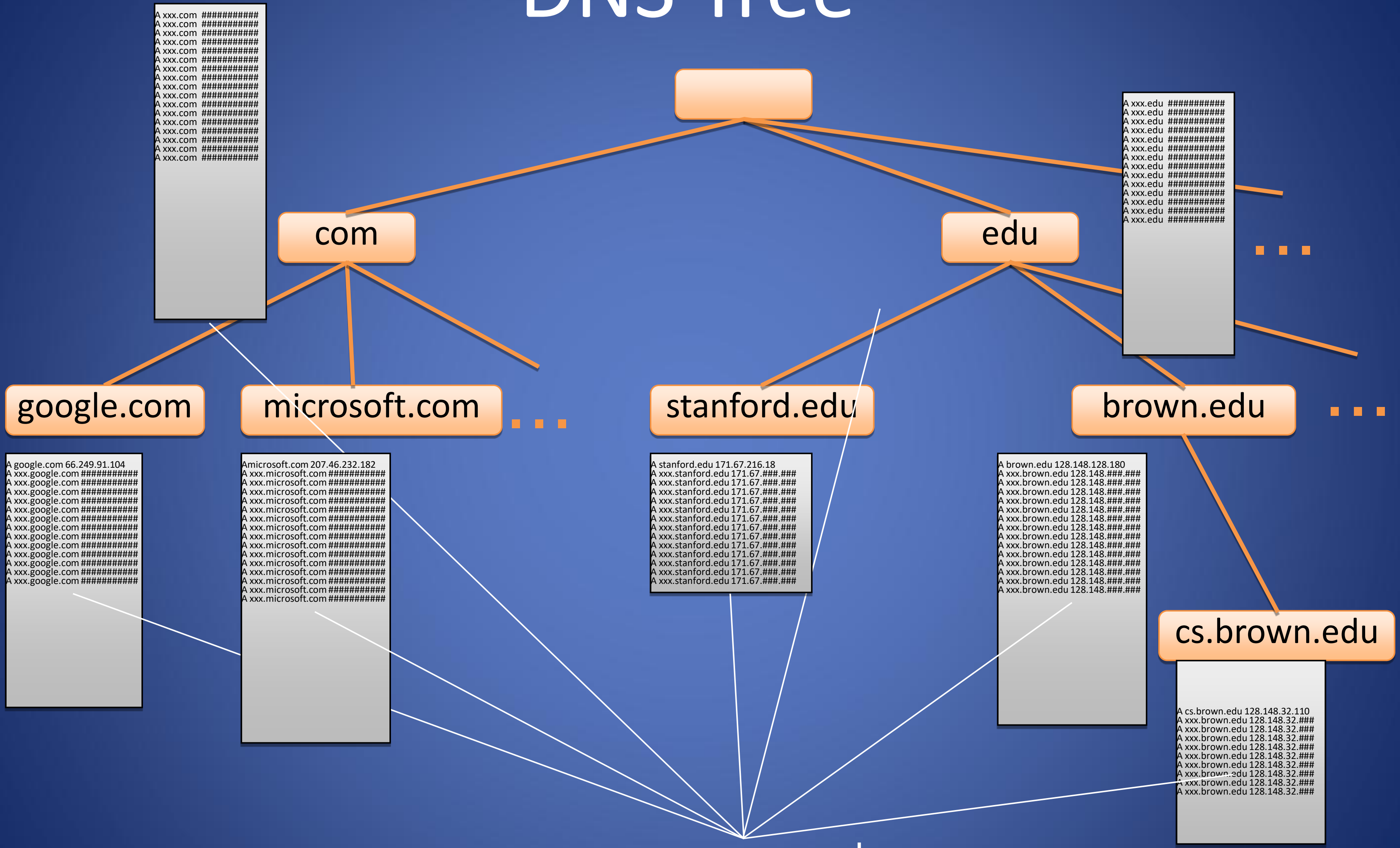
```
# It is best if the NS records for a subzone agree with the delegation
# records above
john.example.com. NS ns1.john.example.com.
john.example.com. NS ns2.john.example.com.

ns1.john.example.com. 10.9.8.7
ns2.john.example.com. 10.5.77.65

# Now that that is out of the way, here is the rest of the zone
john.example.com. 10.9.8.7
www.john.example.com. 10.5.77.65
john.example.com. MX 10 mail.john.example.com.
mail.john.example.com. 10.9.8.7
```

Example DNS entries from http://www.maradns.org/tutorial/recordtypes.html

Name Servers

- Domain names:
  - Two or more labels, separated by dots (e.g., cs166.net)
  - Rightmost label is the top-level domain (TLD)
- Hierarchy of authoritative name servers
  - Information about root domain
  - Information about its subdomains (A records) or references to other name servers (NS records)
- The authoritative name server hierarchy matches the domain hierarchy: root servers point to DNS servers for TLDs, etc.
- Root servers, and servers for TLDs change infrequently
- DNS servers refer to other DNS servers by name, not by IP: sometimes must bootstrap by providing an IP along with a name, called a glue record

DNS Tree



```
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
A xxx.com  ############
```

```
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
A xxx.edu  ############
```

```
A google.com 66.249.91.104
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
A xxx.google.com ###########
```

```
Amicrosoft.com 207.46.232.182
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
A xxx.microsoft.com ###########
```

```
A stanford.edu 171.67.216.18
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
A xxx.stanford.edu 171.67.###.###
```

```
A brown.edu 128.148.128.180
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
A xxx.brown.edu 128.148.###.###
```

```
A cs.brown.edu 128.148.32.110
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
A xxx.brown.edu 128.148.32.###
```

resource records

Namespace Management

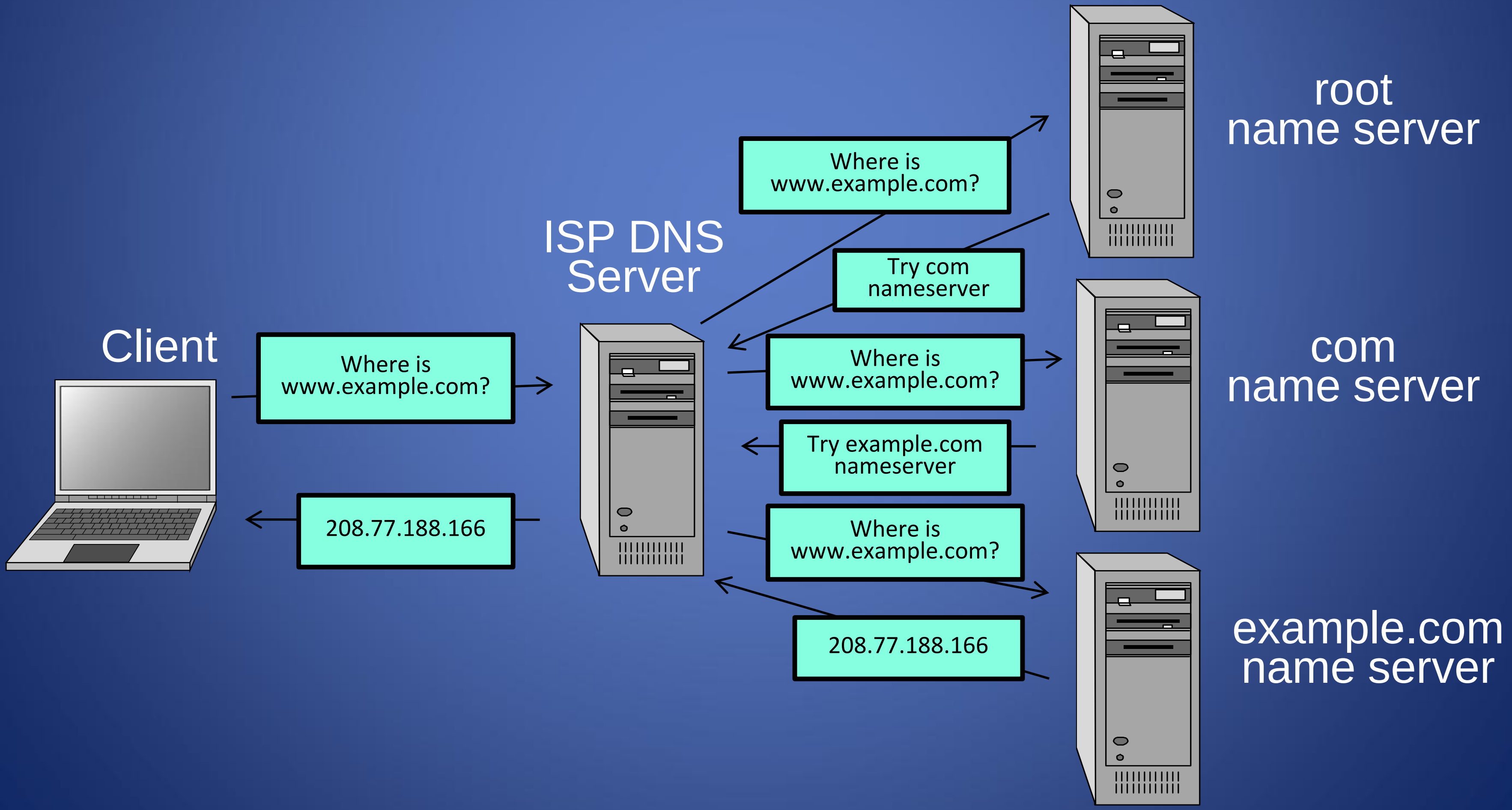
- ICANN: Internet Corporation for Assigned Names and Numbers
- ICANN has the overall responsibility for managing DNS. It controls the root domain, delegating control over each top-level domain to a domain name registry
- Along with a small set of general TLDs, every country has its own TLD -- (cTLDS) – controlled by the government.
- ICANN is the governing body for all general TLDs
- Until 1999 all .com, .net and .org registries were handled by Network Solutions Incorporated.
- After November, 1999, ICANN and NSI had to allow for a shared registration system and there are currently over 500 registrars in the market
- Also since 1999, ICANN has created additional gTLDs including some which are sponsored by consortiums or groups of companies.

Top Level Domains

- Started in 1984
- Originally supposed to be named by function
  - .com for commercial websites, .mil for military
- Eventually agreed upon unrestricted TLDs 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.

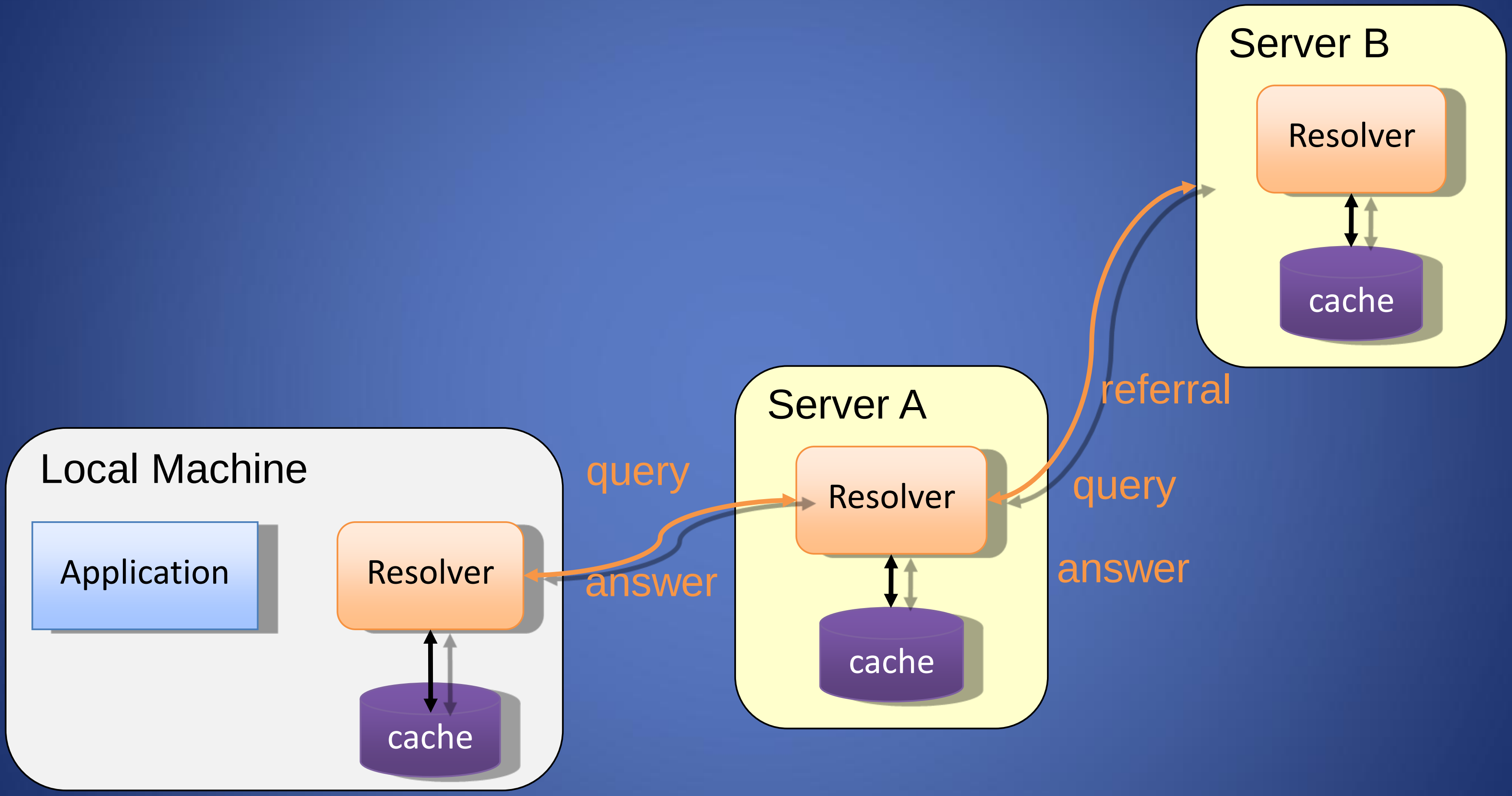
Name Resolution

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

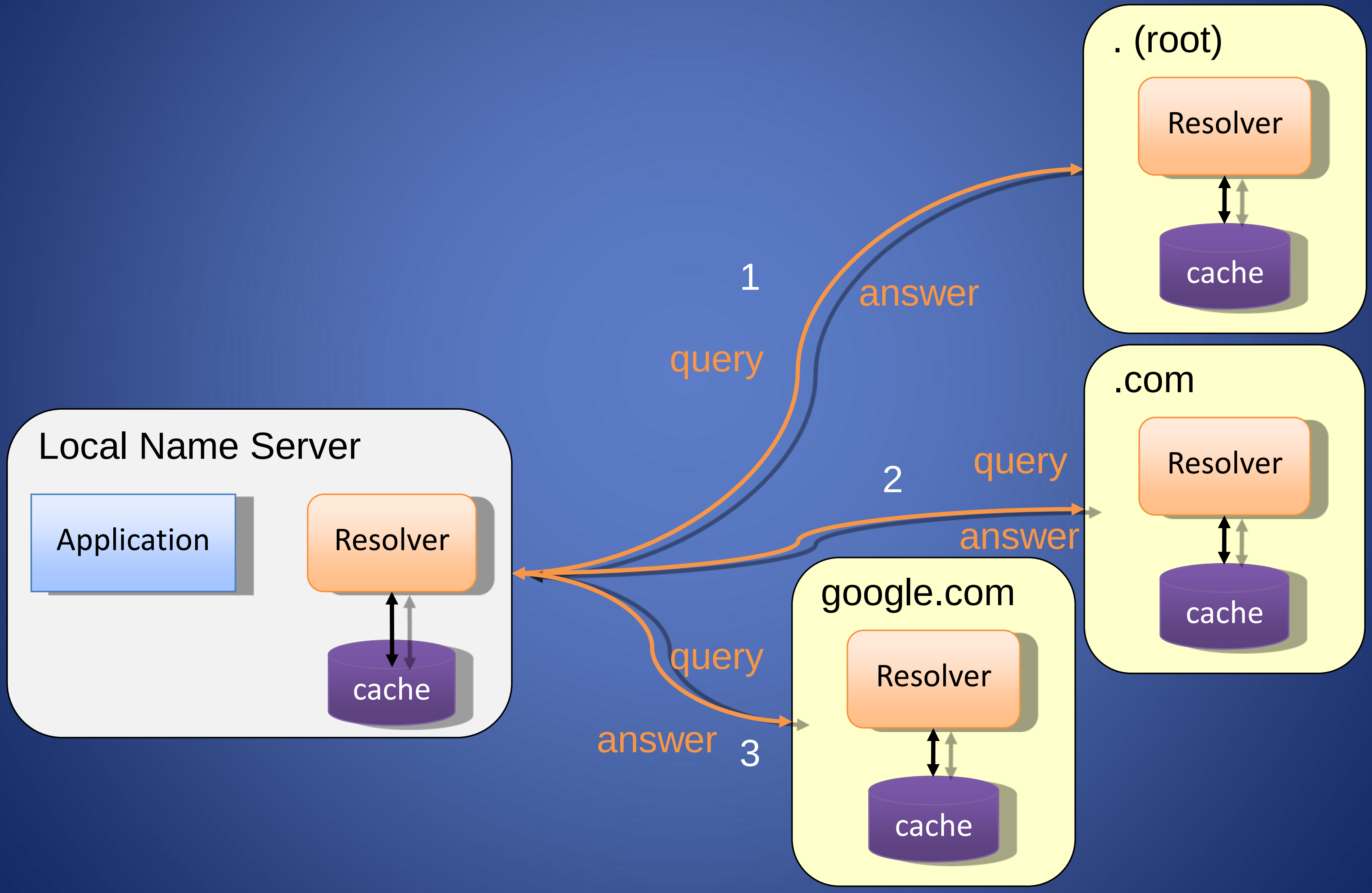


root name server
Where is www.example.com?
ISP DNS Server
Try com nameserver
Client
Where is www.example.com?
Where is www.example.com?
com name server
Try example.com nameserver
208.77.188.166
Where is www.example.com?
example.com name server
208.77.188.166

Recursive Name Resolution



Iterative Name Resolution



Authoritative Name Servers

- Control distributed among authoritative name servers (ANSs)
  - Responsible for specific domains
  - Can designate other ANS for subdomains
- ANS can be master or slave
  - Master contains original zone table
  - Slaves are replicas, automatically updating
- Makes DNS fault tolerant, automatically distributes load
- ANS must be installed as a NS in parents' zone

Dynamic Resolution

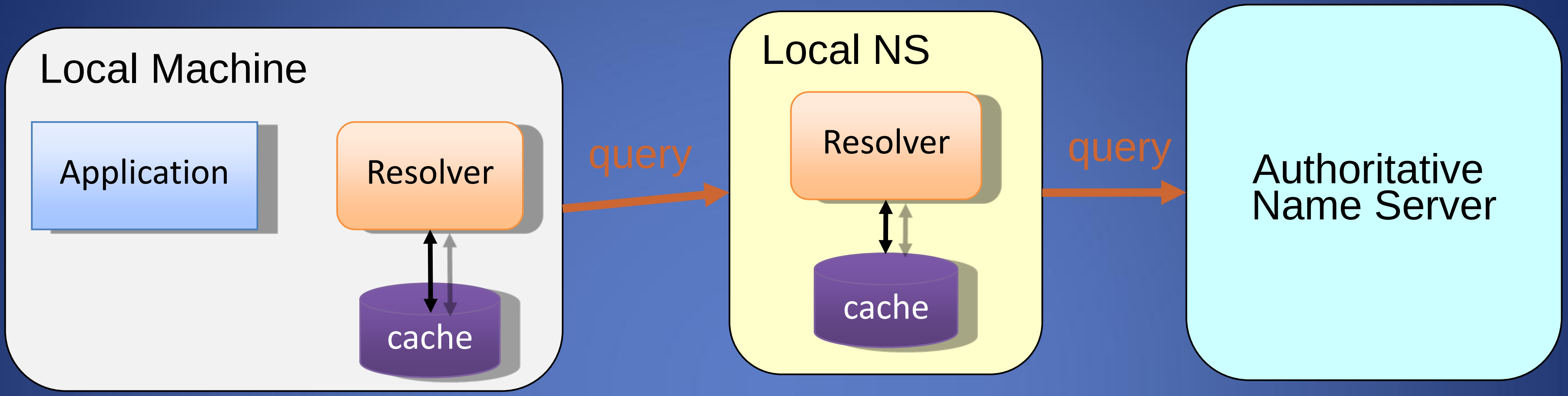
- Many large providers have more than one authoritative name server for a domain
- Problem: need to locate the instance of domain geographically closest to user
- Proposed solution: include first 3 octets of requester's IP in recursive requests to allow better service
- Content distribution networks already do adaptive DNS routing

DNS Caching

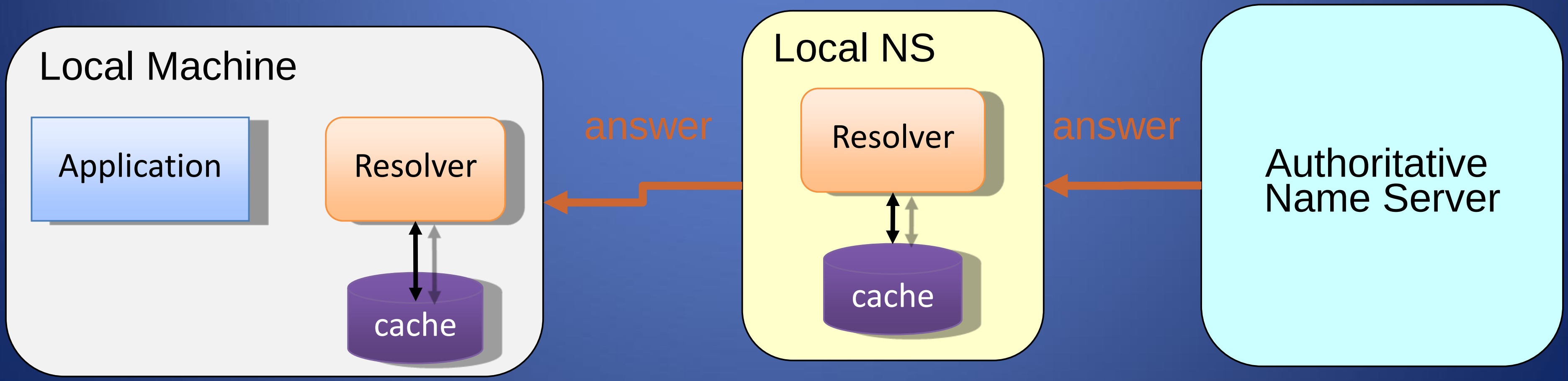
- There would be too much network traffic if a path in the DNS tree would be traversed for each query
  - Root zone would be rapidly overloaded
- DNS servers cache results for a specified amount of time
  - Specified by ANS reply's time-to-live field
- Operating systems and browsers also maintain resolvers and DNS caches
  - View in Windows with command ipconfig /displaydns
  - Associated privacy issues
- DNS queries are typically issued over UDP on port 53
  - 16-bit request identifier in payload

DNS Caching

Step 1: query yourdomain.org

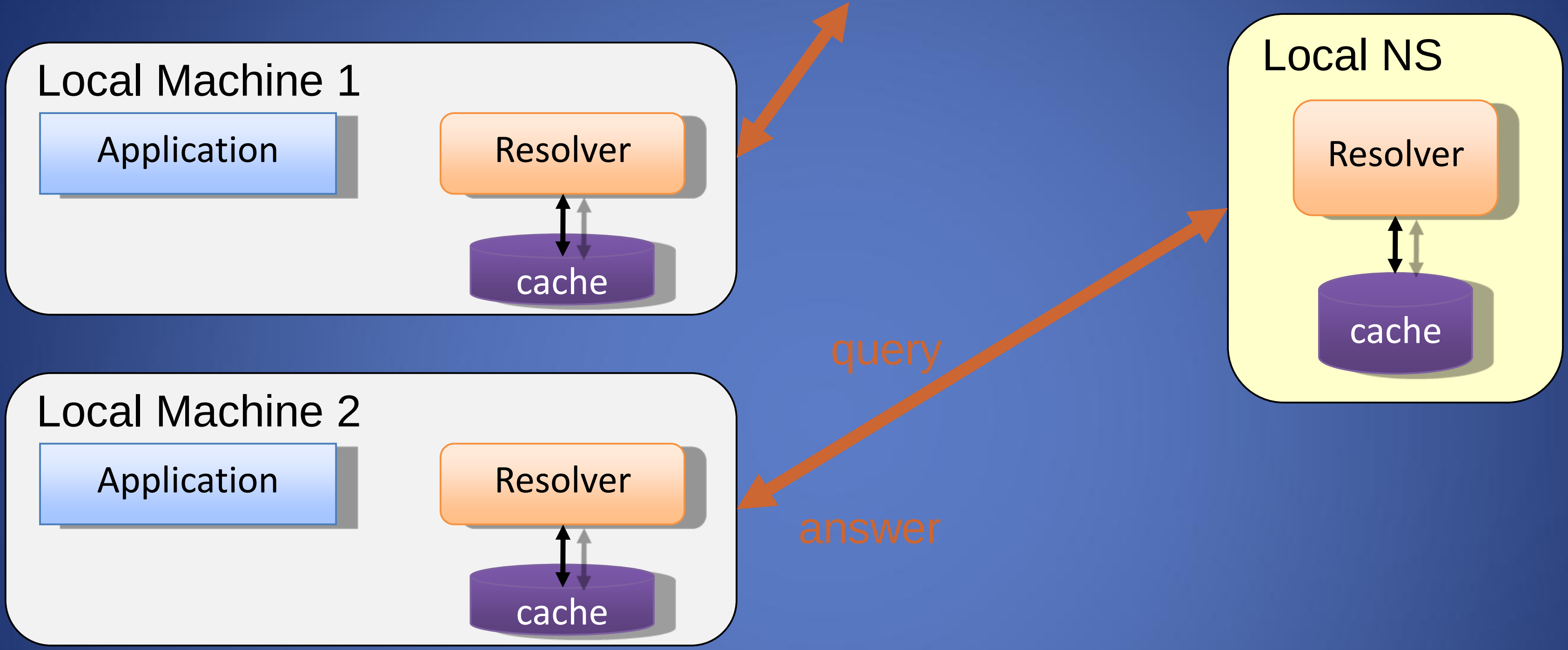


Step 2: receive reply and cache at local NS and host



DNS Caching (con'd)

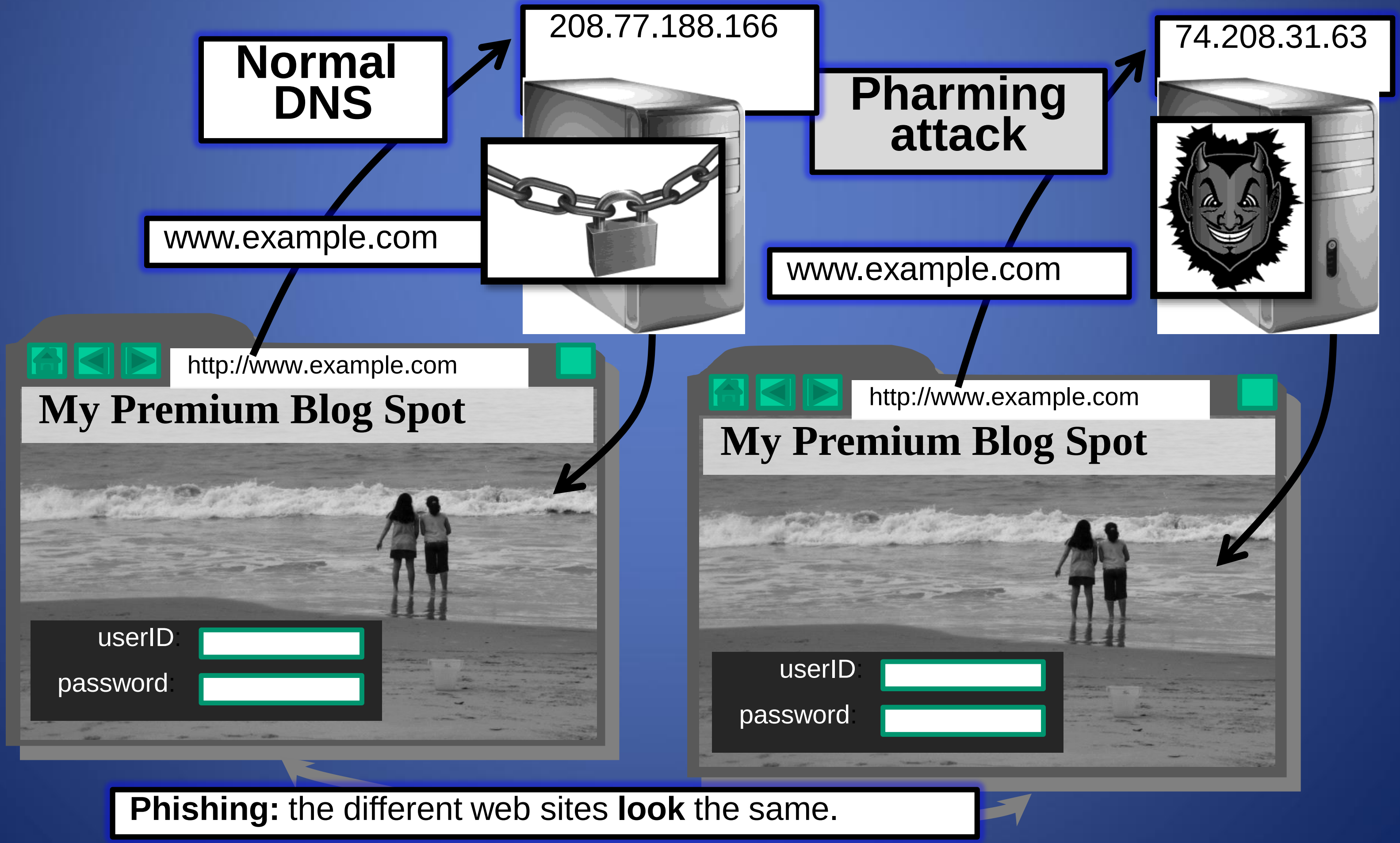
Step 3: use cached results rather than querying the ANS



Step 4: Evict cache entries upon ttl expiration

Pharming: DNS Hijacking

- Changing IP associated with a server maliciously:



Normal DNS

208.77.188.166

www.example.com

Pharming attack

74.208.31.63

www.example.com

http://www.example.com

My Premium Blog Spot

userID:

password:

http://www.example.com

My Premium Blog Spot

userID:

password:

Phishing: the different web sites **look** the same.

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

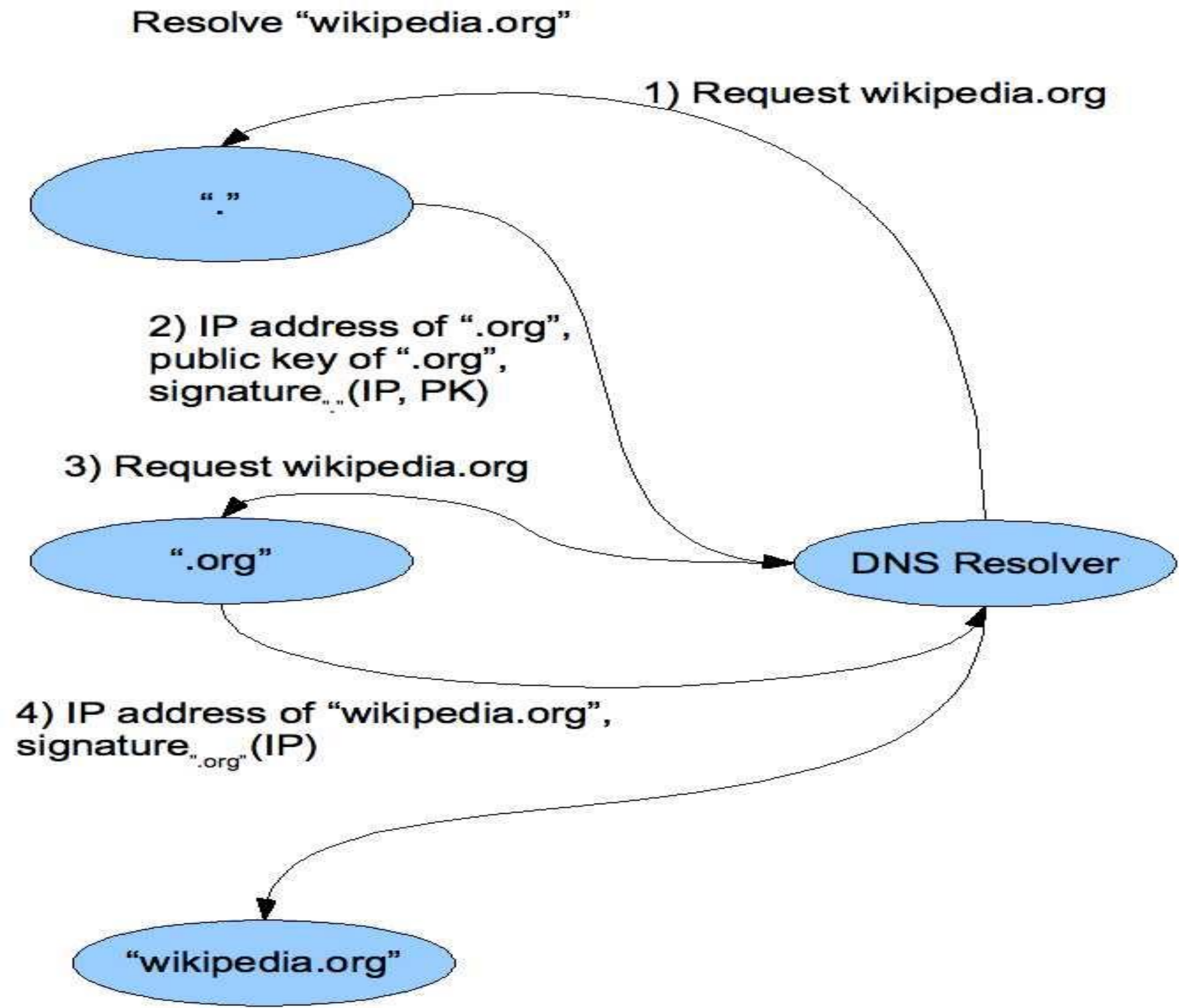
DNS Cache Poisoning Prevention

- Use random identifiers for queries
- Always check identifiers
- Port randomization for DNS requests
- Deploy DNSSEC
  - Challenging because it is still being deployed and requires reciprocity

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


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”

DNSSEC Deployment

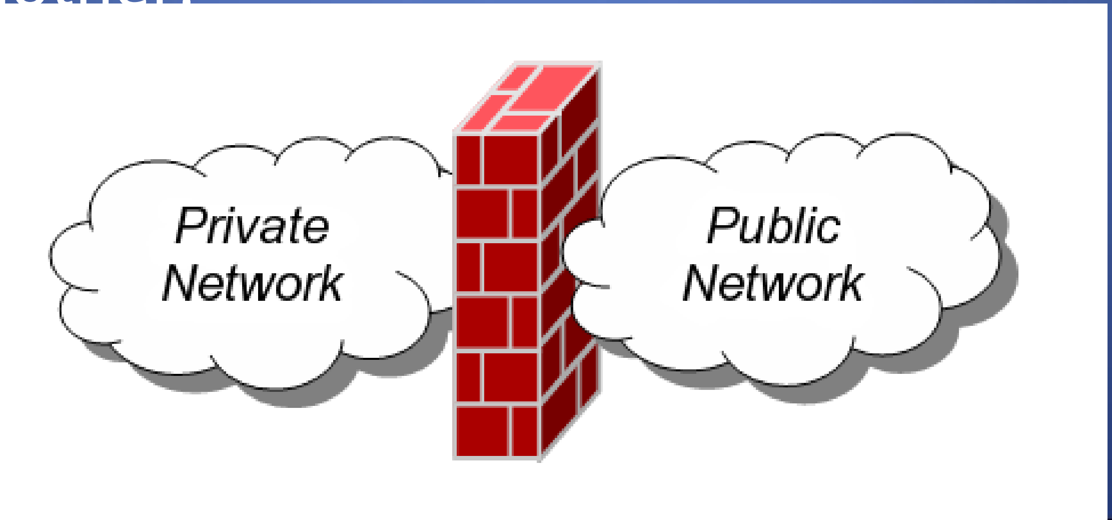
- As the internet becomes regarded as critical infrastructure there is a push to secure DNS
- NIST is in the process of deploying it on root servers now
- May add considerable load to dns servers with packet sizes considerably larger than 512 byte size of UDP packets
- There are political concerns with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

Firewalls, Tunnels, and Network Intrusion Detection

Firewalls

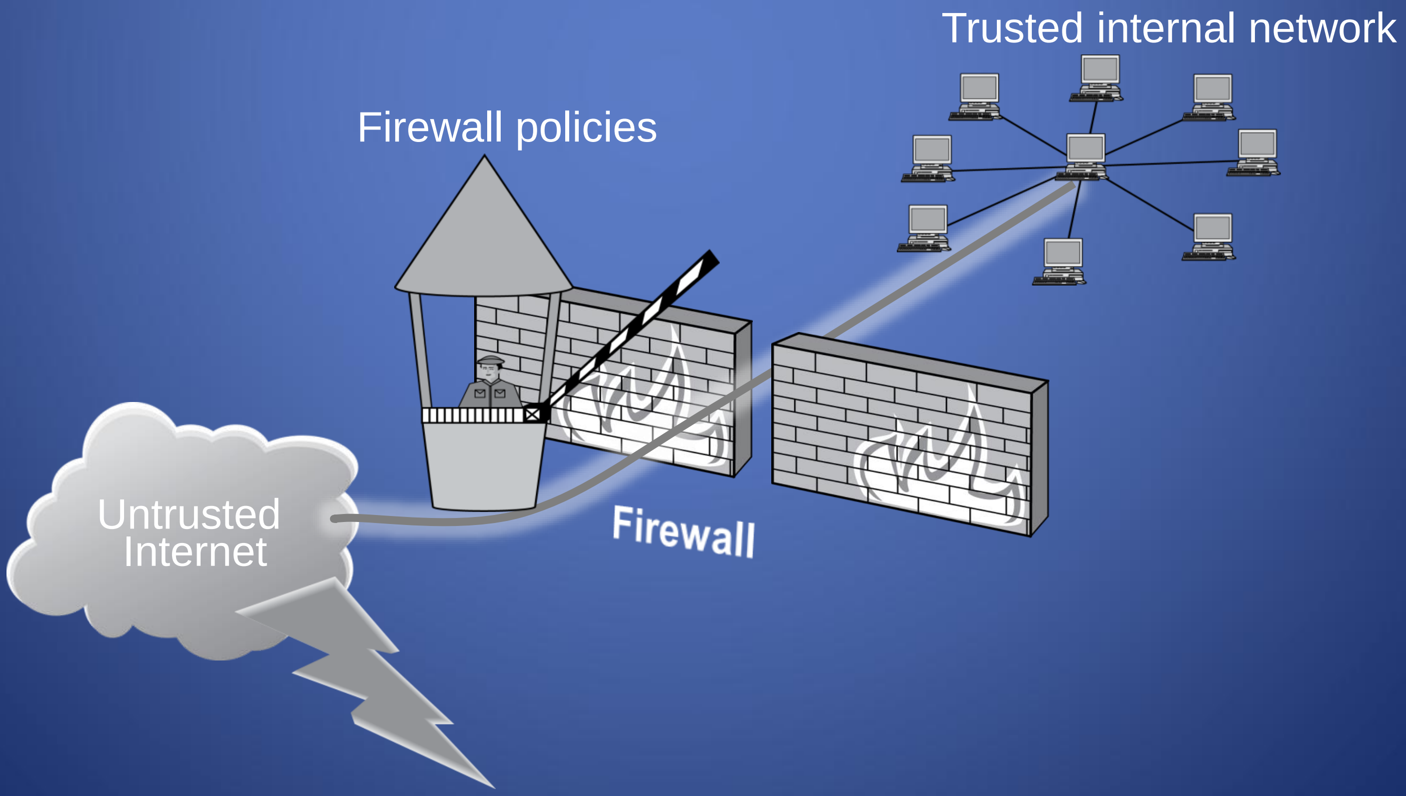
A **firewall** is an integrated collection of security measures designed to prevent unauthorized electronic access to a networked computer system.

A network firewall is similar to firewalls in building construction, because in both cases they are intended to isolate one "network" or "compartment" from another.



Firewall Policies

To protect private networks and individual machines from the dangers of the greater Internet, a firewall can be employed to filter incoming or outgoing traffic based on a predefined set of rules called **firewall policies**.



Policy Actions

Packets flowing through a firewall can have one of three outcomes:

- **Accepted:** permitted through the firewall
- **Dropped:** not allowed through with no indication of failure
- **Rejected:** not allowed through, accompanied by an attempt to inform the source that the packet was rejected

Policies used by the firewall to handle packets are based on several properties of the packets being inspected, including the protocol used, such as:

- TCP or UDP
- the source and destination IP addresses
- the source and destination ports
- the application-level payload of the packet (e.g., whether it contains a virus).

Blacklists and White Lists

There are two fundamental approaches to creating firewall policies (or rulesets) to effectively minimize vulnerability to the outside world while maintaining the desired functionality for the machines in the trusted internal network (or individual computer).

Blacklist approach

All packets are allowed through except those that fit the rules defined specifically in a blacklist.

This type of configuration is more flexible in ensuring that service to the internal network is not disrupted by the firewall, but is naïve from a security perspective in that it assumes the network administrator can enumerate all of the properties of malicious traffic.

Whitelist approach

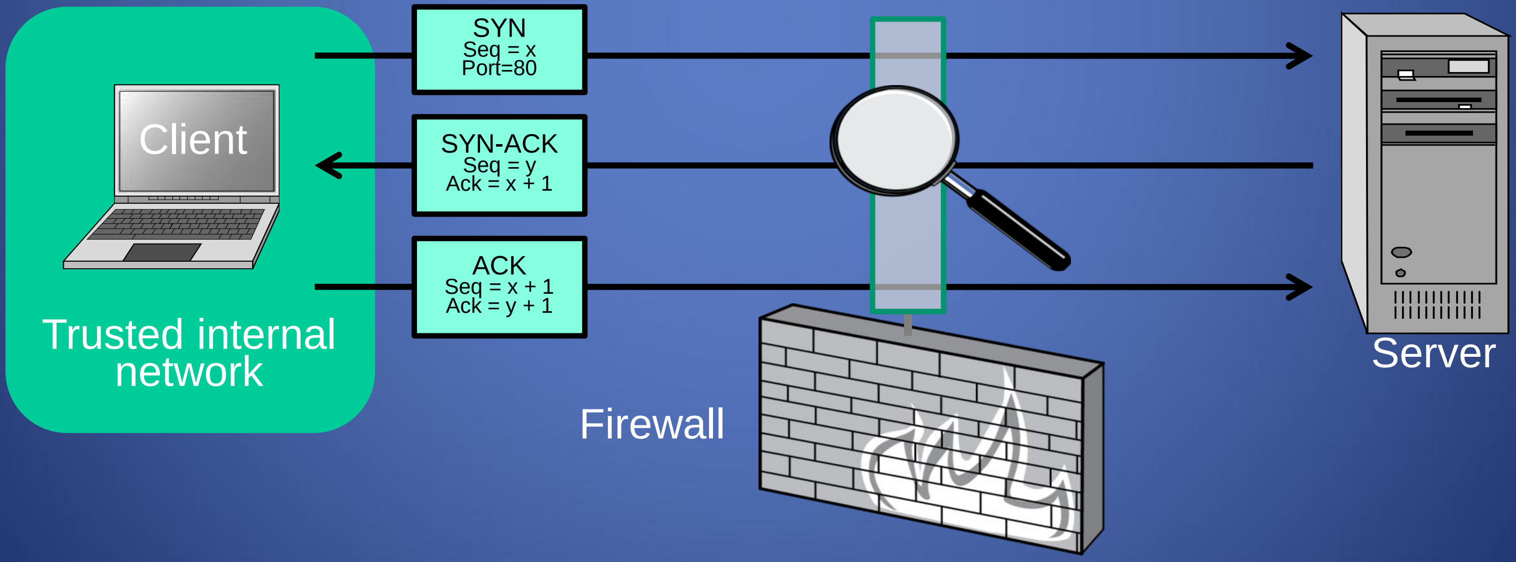
A safer approach to defining a firewall ruleset is the default-deny policy, in which packets are dropped or rejected unless they are specifically allowed by the firewall.

Firewall Types

- **packet filters (stateless)**
  - If a packet matches the packet filter's set of rules, the packet filter will drop or accept it
- **"stateful" filters**
  - it maintains records of all connections passing through it and can determine if a packet is either the start of a new connection, a part of an existing connection, or is an invalid packet.
- **application layer**
  - It works like a **proxy** it can “understand” certain applications and protocols.
  - It may inspect the contents of the traffic, blocking what it views as inappropriate content (i.e. websites, viruses, vulnerabilities, ...)

Stateless Firewalls

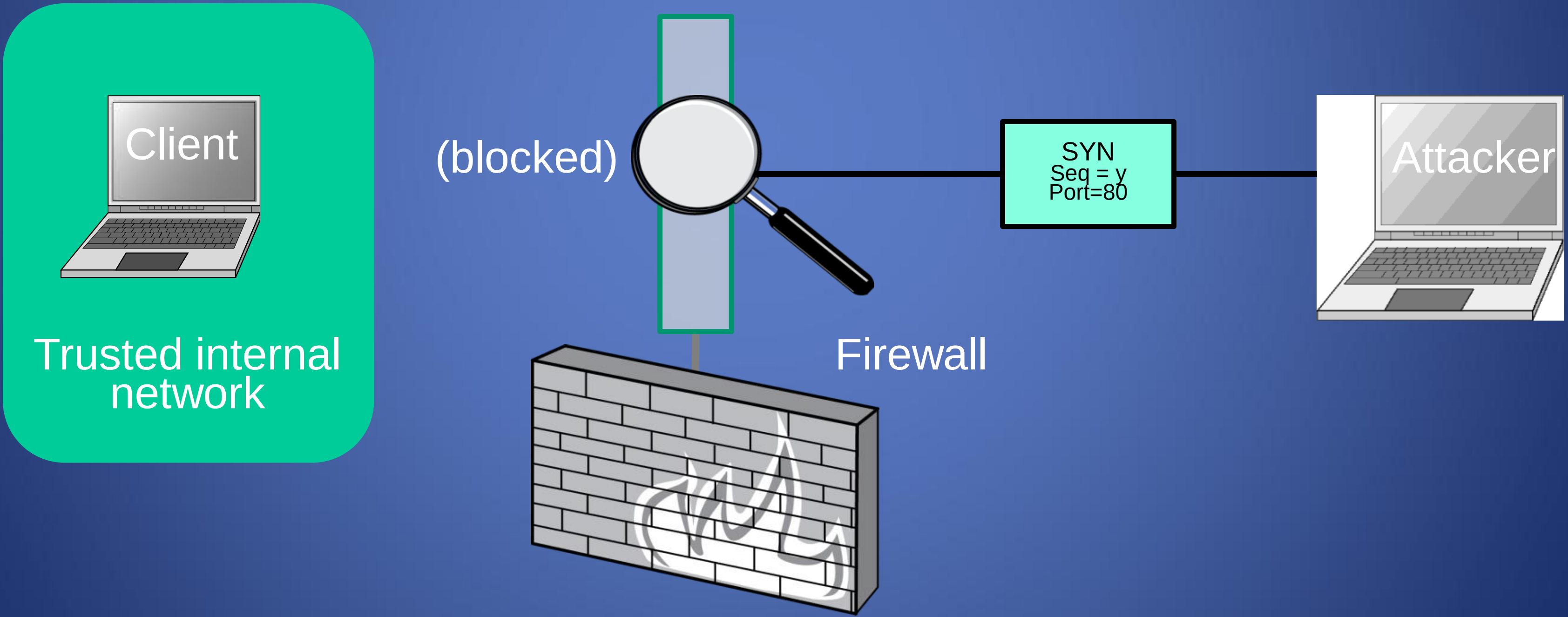
A stateless firewall doesn’t maintain any remembered context (or “state”) with respect to the packets it is processing. Instead, it treats each packet attempting to travel through it in isolation without considering packets that it has processed previously.



Allow outbound SYN packets, destination port=80
Allow inbound SYN-ACK packets, source port=80

Stateless Restrictions

Stateless firewalls may have to be fairly restrictive in order to prevent most attacks.



Client

Trusted internal network

(blocked)

SYN
Seq = y
Port=80

Attacker

Firewall

Allow outbound SYN packets, destination port=80
Drop inbound SYN packets,
Allow inbound SYN-ACK packets, source port=80

Statefull Firewalls

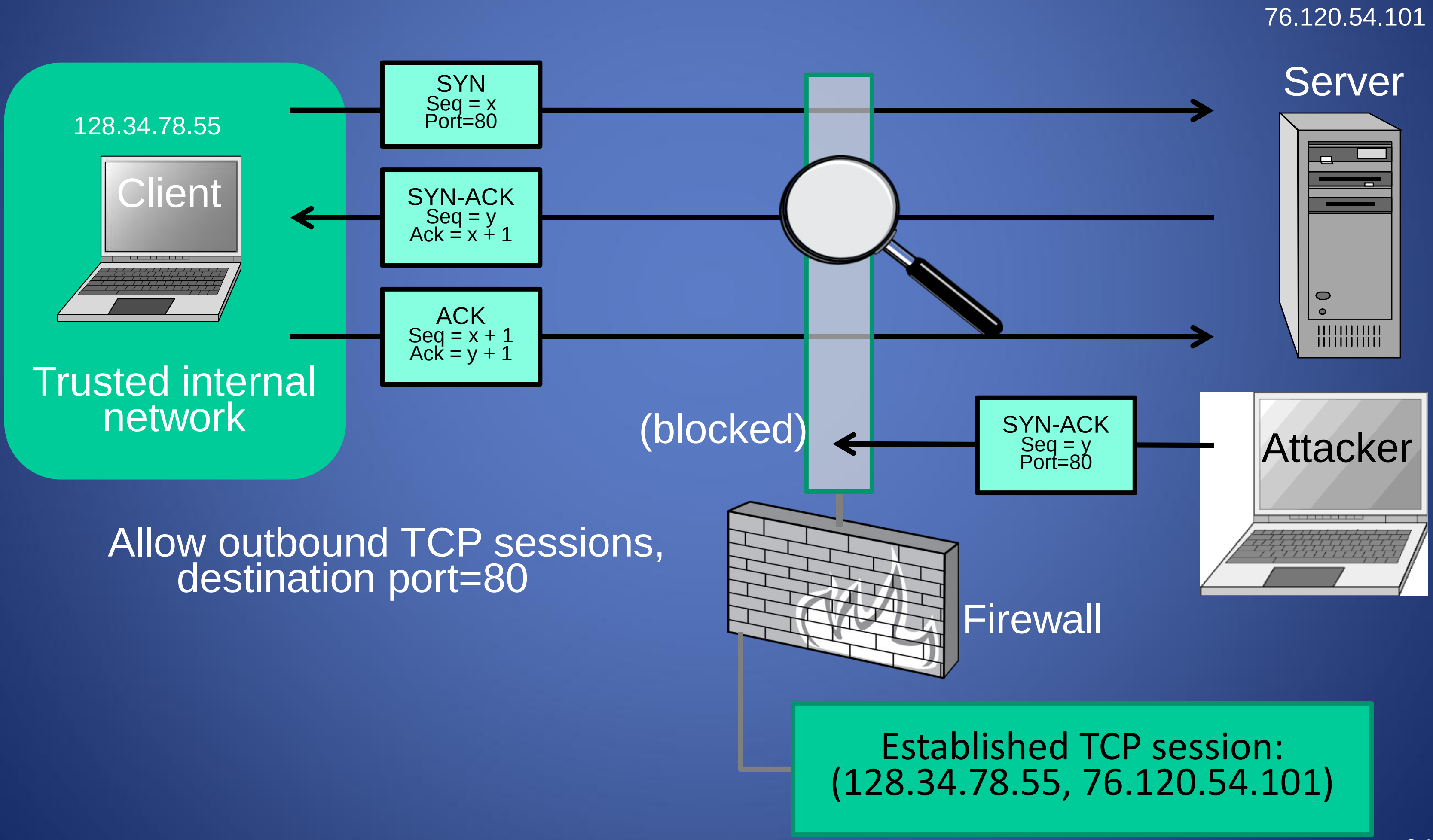
Stateful firewalls can tell when packets are part of legitimate sessions originating within a trusted network.

Stateful firewalls maintain tables containing information on each active connection, including the IP addresses, ports, and sequence numbers of packets.

Using these tables, stateful firewalls can allow only inbound TCP packets that are in response to a connection initiated from within the internal network.

Statefull Firewall Example

Allow only requested TCP connections:



Allow outbound TCP sessions, destination port=80

Established TCP session:
(128.34.78.55, 76.120.54.101)

Firewall state table

Tunnels

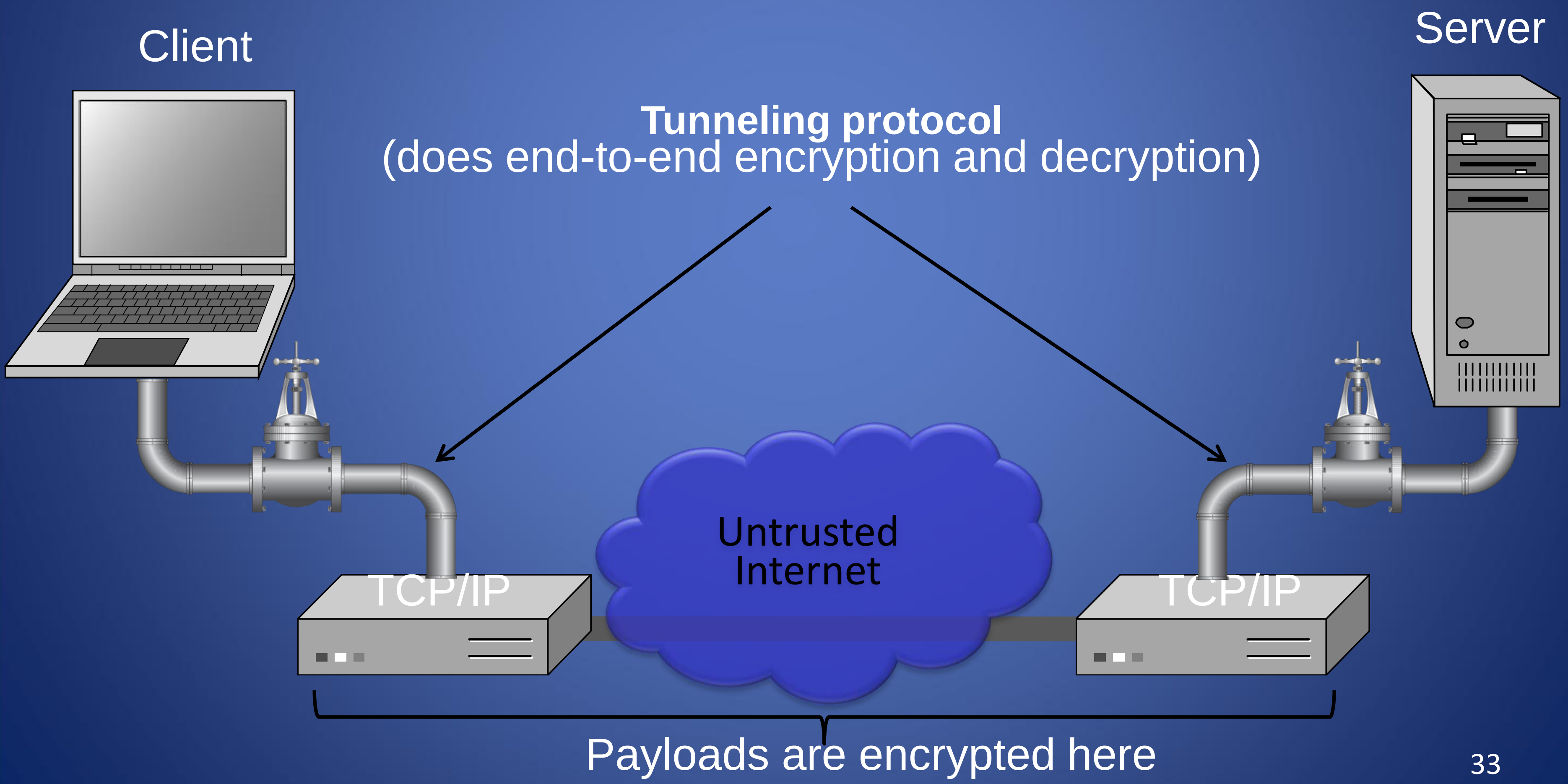
The contents of TCP packets are not normally encrypted, so if someone is eavesdropping on a TCP connection, he can often see the complete contents of the payloads in this session.

One way to prevent such eavesdropping without changing the software performing the communication is to use a **tunneling protocol.**

In such a protocol, the communication between a client and server is automatically encrypted, so that useful eavesdropping is infeasible.

Tunneling Prevents Eavesdropping

Packets sent over the Internet are automatically encrypted.



Client

Server

Tunneling protocol
(does end-to-end encryption and decryption)

TCP/IP

Untrusted
Internet

TCP/IP

Payloads are encrypted here

Secure Shell (SSH)

A secure interactive command session:

1. The client connects to the server via a TCP session.
2. The client and server exchange information on administrative details, such as supported encryption methods and their protocol version, each choosing a set of protocols that the other supports.
3. The client and server initiate a secret-key exchange to establish a shared secret session key, which is used to encrypt their communication (but not for authentication). This session key is used in conjunction with a chosen block cipher (typically AES, 3DES) to encrypt all further communications.
4. The server sends the client a list of acceptable forms of authentication, which the client will try in sequence. The most common mechanism is to use a password or the following public-key authentication method:
    a) If public-key authentication is the selected mechanism, the client sends the server its public key.
    b) The server then checks if this key is stored in its list of authorized keys. If so, the server encrypts a challenge using the client’s public key and sends it to the client.
    c) The client decrypts the challenge with its private key and responds to the server, proving its identity.
5. Once authentication has been successfully completed, the server lets the client access appropriate resources, such as a command prompt

IPSec

- IPSec defines a set of protocols to provide confidentiality and authenticity for IP packets
- Each protocol can operate in one of two modes, **transport mode** or **tunnel mode.**
  - In **transport mode,** additional IPsec header information is inserted before the data of the original packet, and only the payload of the packet is encrypted or authenticated.
  - In **tunnel mode**, a new packet is constructed with IPsec header information, and the entire original packet, including its header, is encapsulated as the payload of the new packet.

Virtual Private Networking (VPN)

Virtual private networking (VPN) is a technology that allows private networks to be safely extended over long physical distances by making use of a public network, such as the Internet, as a means of transport.

VPN provides guarantees of data confidentiality, integrity, and authentication, despite the use of an untrusted network for transmission.

There are two primary types of VPNs, **remote access VPN** and **site-to-site VPN.**

Types of VPNs

Remote access VPNs allow authorized clients to access a private network that is referred to as an **intranet.**

For example, an organization may wish to allow employees access to the company network remotely but make it appear as though they are local to their system and even the Internet itself.

To accomplish this, the organization sets up a VPN endpoint, known as a **network access server, or NAS.** Clients typically install VPN client software on their machines, which handle negotiating a connection to the NAS and facilitating communication.

Site-to-site VPN solutions are designed to provide a secure bridge between two or more physically distant networks.

Before VPN, organizations wishing to safely bridge their private networks purchased expensive leased lines to directly connect their intranets with cabling.

Intrusion Detection Systems

Intrusion

Actions aimed at compromising the security of the target (confidentiality, integrity, availability of computing/networking resources)

Intrusion detection

The identification through intrusion signatures and report of intrusion activities

Intrusion prevention

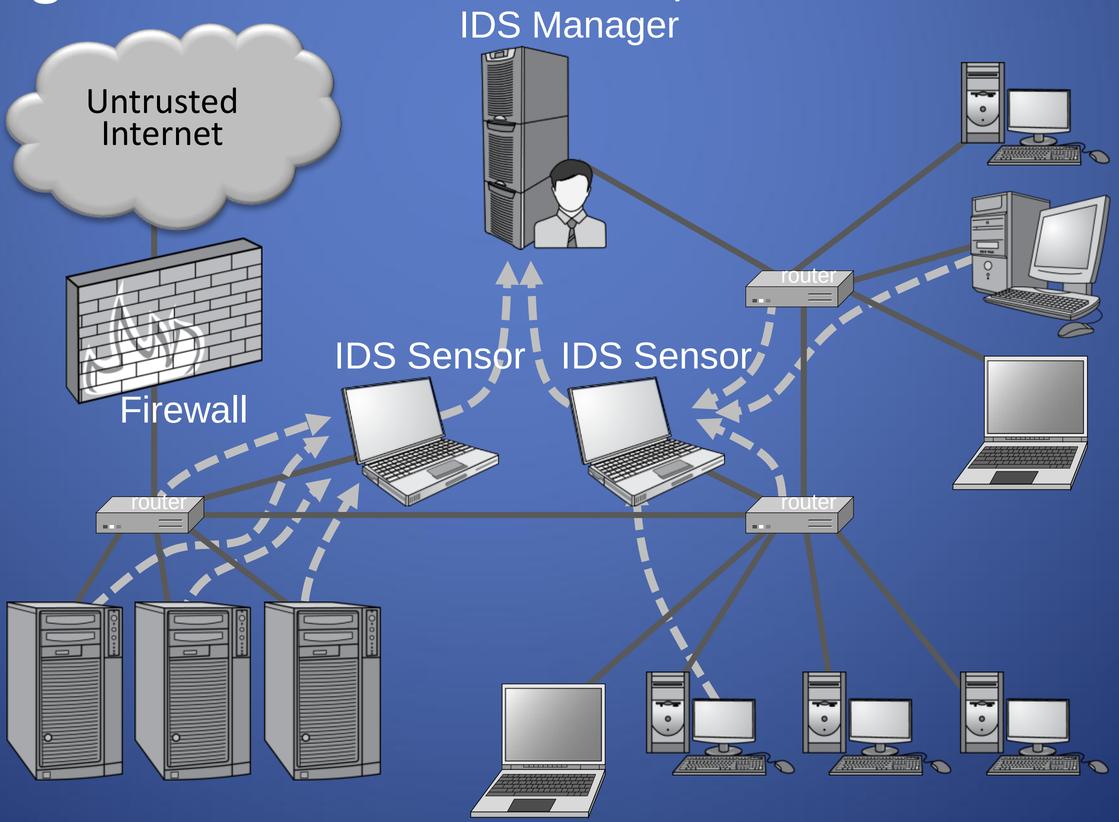
The process of both detecting intrusion activities and managing automatic responsive actions throughout the network

IDS Components

The **IDS manager** compiles data from the IDS sensors to determine if an intrusion has occurred.

This determination is based on a set of **site policies,** which are rules and conditions that define probable intrusions.

If an IDS manager detects an intrusion, then it sounds an **alarm**.



Intrusions

An IDS is designed to detect a number of threats, including the following:

- **masquerader:** an attacker who is falsely using the identity and/or credentials of a legitimate user to gain access to a computer system or network
- **Misfeasor:** a legitimate user who performs actions he is not authorized to do
- **Clandestine user:** a user who tries to block or cover up his actions by deleting audit files and/or system logs

In addition, an IDS is designed to detect automated attacks and threats, including the following:

- **port scans:** information gathering intended to determine which ports on a host are open for TCP connections
- **Denial-of-service attacks:** network attacks meant to overwhelm a host and shut out legitimate accesses
- **Malware attacks:** replicating malicious software attacks, such as Trojan horses, computer worms, viruses, etc.
- **ARP spoofing:** an attempt to redirect IP traffic in a local-area network
- **DNS cache poisoning:** a pharming attack directed at changing a host's DNS cache to create a falsified domain-name/IP-address association

Possible Alarm Outcomes

Alarms can be sounded (positive) or not (negative)

Intrusion Attack

No Intrusion Attack

Alarm Sounded

No Alarm Sounded

NYPD
03539480
NYPD
03539480

The Base-Rate Fallacy

It is difficult to create an intrusion detection system with the desirable properties of having both a high true-positive rate and a low false-negative rate.

If the number of actual intrusions is relatively small compared to the amount of data being analyzed, then the effectiveness of an intrusion detection system can be reduced.

In particular, the effectiveness of some IDSs can be misinterpreted due to a statistical error known as the **base-rate fallacy.**

This type of error occurs when the probability of some conditional event is assessed without considering the “base rate” of that event.

Base-Rate Fallacy Example

Suppose an IDS is 99% accurate, having a 1% chance of false positives or false negatives. Suppose further…

An intrusion detection system generates 1,000,100 log entries.

Only 100 of the 1,000,100 entries correspond to actual malicious events.

Because of the success rate of the IDS, of the 100 malicious events, 99 will be detected as malicious, which means we have **1 false negative.**

Nevertheless, of the 1,000,000 benign events, 10,000 will be mistakenly identified as malicious. That is, we have **10,000 false positives!**

Thus, there will be 10,099 alarms sounded, 10,000 of which are false alarms. That is, roughly 99% of our alarms are false alarms.

IDS Data

In an influential 1987 paper, Dorothy Denning identified several fields that should be included in IDS event records:

Subject: the initiator of an action on the target

Object: the resource being targeted, such as a file, command, device, or network protocol

Action: the operation being performed by the subject towards the object

Exception-condition: any error message or exception condition that was raised by this action

Resource-usage: quantitative items that were expended by the system performing or responding to this action

Time-stamp: a unique identifier for the moment in time when this action was initiated

Types of Intrusion Detection Systems

Rule-Based Intrusion Detection

Rules identify the types of actions that match certain known profiles for an intrusion attack, in which case the rule would encode a **signature** for such an attack. Thus, if the IDS manager sees an event that matches the signature for such a rule, it would immediately sound an alarm, possibly even indicating the particular type of attack that is suspected.

Statistical Intrusion Detection

A **profile** is built, which is a statistical representation of the typical ways that a user acts or a host is used; hence, it can be used to determine when a user or host is acting in highly unusual, anomalous ways.

Once a user profile is in place, the IDS manager can determine thresholds for anomalous behaviors and then sound an alarm any time a user or host deviates significantly from the stored profile for that person or machine.

Wireless Networks

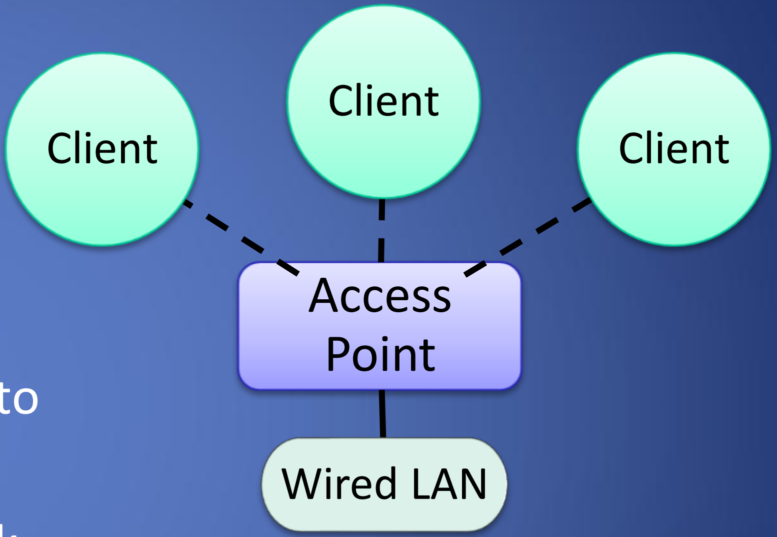
Welcome to Wireless

- Radio waves
  - No need to be physically plugged into the network
  - Remote access
- Coverage
  - Personal Area Network (PAN)
  - Local Area Network (LAN)
  - Metropolitan Area Network (MAN)

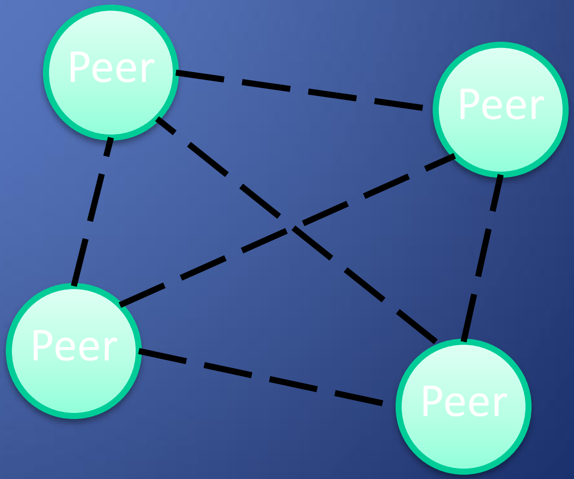
- Security concerns
  - Radio signals leaking outside buildings
  - Detection of unauthorized devices
  - Intercepting wireless communications
  - Man-in-the-middle attacks
  - Verification of users
  - Restricting access

Types of Wireless Networks

- Infrastructure
  - Client machines establish a radio connection to a special network device, called access point
  - Access points connected to a wired network, which provides a gateway to the internet
  - Most common type of wireless network



- Peer-to-peer
  - Multiple peer machines connect to each other
  - Typically used in ad-hoc networks and internet connection sharing



SSID

- Multiple wireless networks can coexist
  - Each network is identified by a 32-character service set ID (SSID)
  - Typical default SSID of access point is manufacturer’s name
  - SSIDs often broadcasted to enable discovery of the network by prospective clients
- SSIDs are not signed, thus enabling a simple spoofing attack
  - Place a rogue access point in a public location (e.g., cafe, airport)
  - Use the SSID of an ISP
  - Set up a login page similar to the one of the ISP
  - Wait for clients to connect to rogue access point and authenticate
  - Possibly forward session to ISP network
  - Facilitated by automatic connection defaults

Eavesdropping and Spoofing

All wireless network traffic can be eavesdropped

MAC-based authentication typically used to identify approved machines in corporate network

MAC spoofing attacks possible, as in wired networks

Sessions kept active after brief disconnects

If ISP client does not explicitly end a session, MAC spoofing allows to take over that session

Captive Portal

Protocol

- DHCP provides IP address
- Name server maps everything to authentication server
- Firewall blocks all other traffic
- Any URL is redirected to authentication page
- After authentication, regular network services reinstated
- Client identified by MAC address
- Used by wireless ISPs

Security issues

- A MAC spoofing and session stealing attack may be performed if client does not actively disconnect
- A tunneling attack can bypass captive portal if DNS traffic beyond firewall is not blocked before authentication

Wardriving and Warchalking

Driving around looking for wireless local area networks

Some use GPS devices to log locations, post online

Software such as NetStumbler for Windows, KisMac for Macs and Kismet for Linux are easily available online

Use antennas to increase range

Legality is unclear when no information is transmitted, and no network services are used

Warchalking involves leaving chalk marks (derived from hobo symbols) on the side walk marking wireless networks and associated information

Wired Equivalent Privacy

Goals

- Confidentiality: eavesdropping is prevented
- Data integrity: packets cannot be tampered with
- Access control: only properly encrypted packets are routed

Design constraints

- Inexpensive hardware implementation with 90's technology
- Compliance with early U.S. export control regulations on encryption devices (40-bit keys)

Implementation and limitations

- Encrypts the body of each frame at the data-link level
- Legacy IEEE 802.11 standard to be avoided

WEP Protocol

Setup

- Access point and client share 40-bit key K
- The key never changes during a WEP session

Encryption

- Compute CRC-32 checksum of message M (payload of frame)
- Pick 24-bit initialization vector V
- Using the RC4 stream cipher, generate key stream S(K,V)
- Create ciphertext

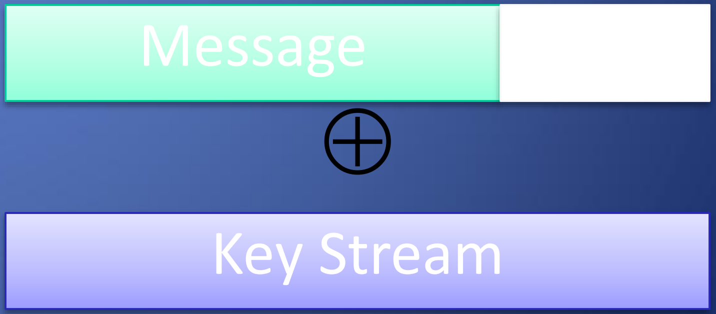
  $C = (M \mid\mid crc(M)) \oplus S(K,V)$

Client authentication

- Access point sends unencrypted random challenge to client
- Client responds with encrypted challenge

Transmission

- Send V || C



Message Modification Attack

- Message modification
  - Given an arbitrary string Δ, we want to replace message M with $M' = M \oplus \Delta$
  - Man-in-the middle replaces ciphertext C with $C' = C \oplus (\Delta \,||\, crc(\Delta))$
- Targeted text replacement
  - Possible if we know position of text in message
  - E.g., change date in email
- Reason of vulnerability
  - CRC checksum distributes over XOR
  - Not a cryptographic hash function

IP Redirection Attack

Attacker convinces access point to decrypt packet

Method

- Eavesdrop inbound IP packet
- Resend packet to external machine controlled by attacker
- Receive packet decrypted by access point
- Repeat with outbound packets

Guess destination address

- Within LAN subnet

Change destination address

- Modify original destination D to external machine D′ controlled by attacker
- Use above message modification method

Change packet checksum

- Difference between new checksum and old known $x' - x = (D'_H + D'_L) - (D_H + D_L)$
- Guess $x' \oplus x$

Success after few attempts

Reused Initialization Vectors

- Repeated IV implies reused key stream
  - Attacker obtains XOR of two messages
  - Attacker can recover both message and key stream
  - Recovered key stream can be used by attacker to inject traffic
- Default IV
  - Several flawed implementations of IV generation
  - E.g., start at zero when device turned on and then repeatedly increment by one
- Random IV
  - Small length (24 bits) leads to repetition in a short amount of time even randomly generated
  - E.g., collision expected with high probability after $2^{12} \approx 4{,}000$ transmissions

Authentication Spoofing

Attacker wants to spoof a legitimate client

- Does not know the secret key K
- Can eavesdrop authentication messages

Attack

- Obtain challenge R and encrypted challenge $C = (R \,||\, crc(R)) \oplus S(K,V)$
- Compute key stream $S(K,V) = (R \,||\, crc(R)) \oplus C$
- Reuse key stream $S(K,V)$ when challenged from access point

DEMO: WARDRIVING AND WEP CRACKING

Wardriving Tools

Netstumbler
wifi scanner

Antenna for db gain

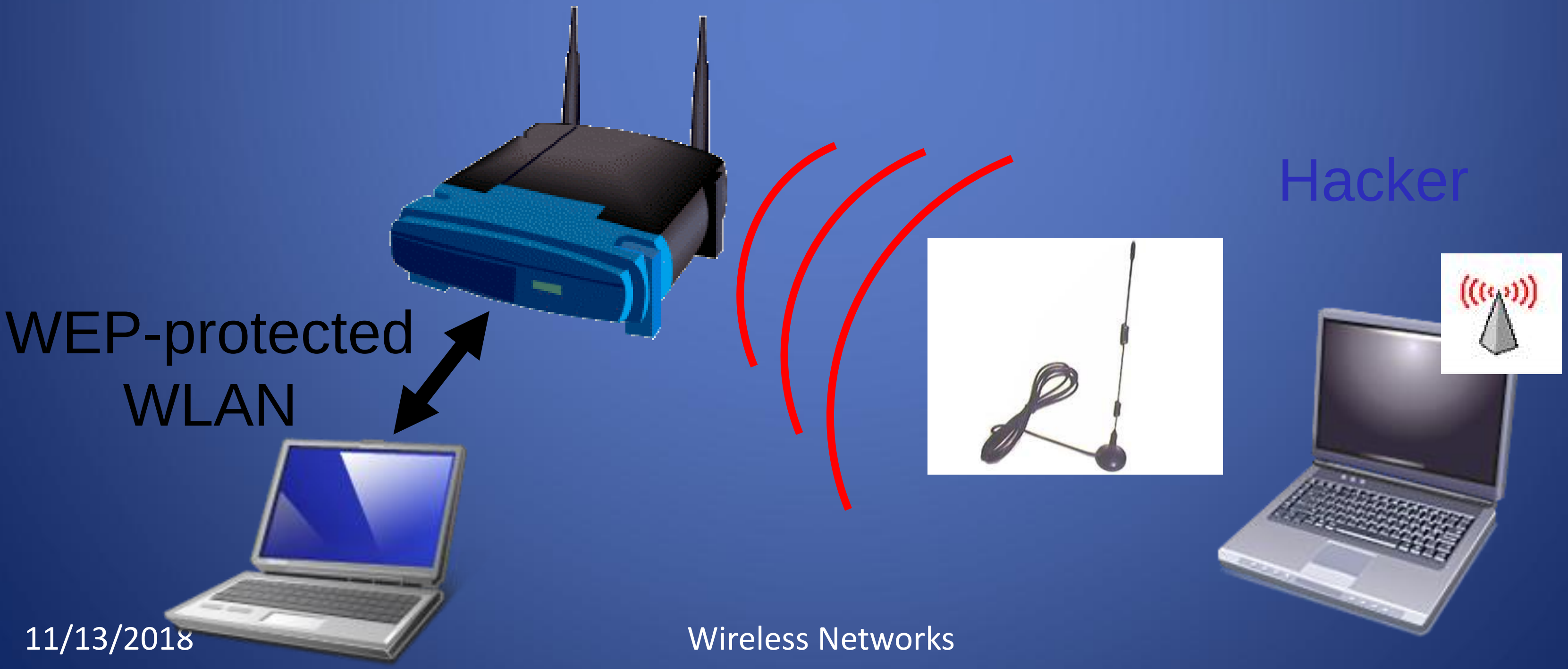
Wireless card with
plug and monitor mode

GPS (optional)

Wardriving Setup

The access point and client are using WEP encryption

The hacker is sniffing using wardriving tools



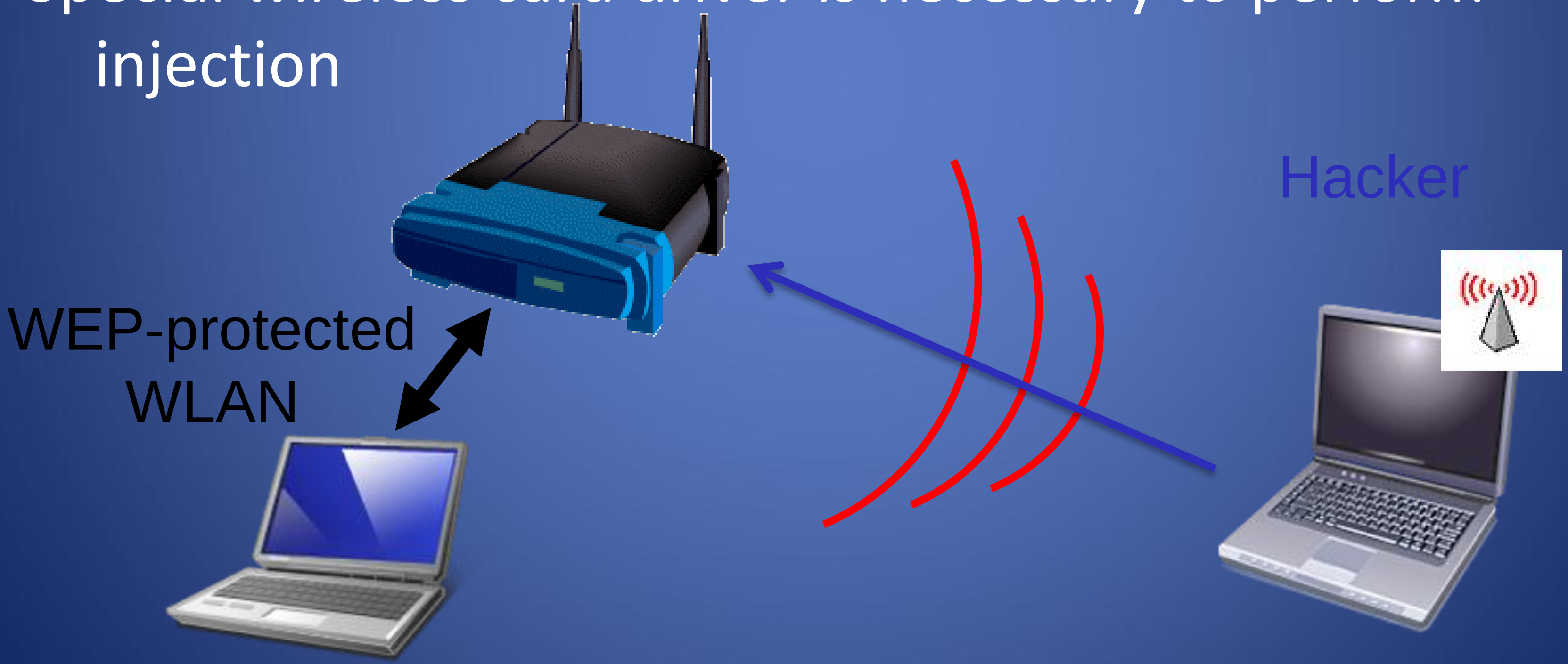
Slow Attack: WEP Sniffing

- To crack a 64-bit WEP key you can capture:
  - 50,000 to 200,000 packets containing Initialization Vectors (IVs)
  - Only about ¼ of the packets contain IVs
  - So you need 200,000 to 800,000 packets
- It can take a long time (typically several hours or even days) to capture that many packets

Fast Attack: Packet Injection

The hacker injects packets to create a more "interesting" packet

Special wireless card driver is necessary to perform injection



Initialization vector (IV)

- One for each packet, a 24-bit value
- Sent in the cleartext part of the message!
- Small space of initialization vectors guarantees reuse of the same key stream
- IV Collision:
  - Attack the XOR of the two plaintext messages
  - IV is often very predictable and introduces a lot of redundancy

Injection Method

Suppose attacker knows one plaintext for one encrypted message, X

$RC4(X) \oplus X \oplus Y = RC4(Y)$

constructing a new message calculating the CRC32

Even without a complete knowledge of the packet, it is possible to flip selected bits in a message and successfully adjust the encrypted CRC

We know ARP, reinject it:

ARP will normally rebroadcast and generate IVs

Reference

Nikita Borisov, Ian Goldberg, David Wagner, Intercepting Mobile Communications: The Insecurity of 802.11. MOBICOM, 2001.

Wi-Fi Protected Access (WPA)

- WEP became widely known as insecure
  - In 2005, FBI publically cracked a WEP key in only 3 minutes!
- Wi-Fi Protected Access (WPA) proposed in 2003
- Improves on WEP in several ways:
  - Larger secret key (128 bits) and initialization data (48 bits)
  - Supports various types of authentication besides a shared secret, such as username/password
  - Dynamically changes keys as session continues
  - Cryptographic method to check integrity
  - Frame counter to prevent replay attacks

WPA2

- WPA was an intermediate stepping-stone
  - Final version: IEEE 802.11i, aka WPA2
- Improvements over WPA are incremental rather than changes in philosophy:
  - Uses AES instead of RC4
  - Handles encryption, key management, and integrity
  - MAC provided by Counter Mode with Cipher Block Chaining (CCMP) used in conjunction with AES
- WPA2 needs recent hardware to operate properly, but this will get better over time

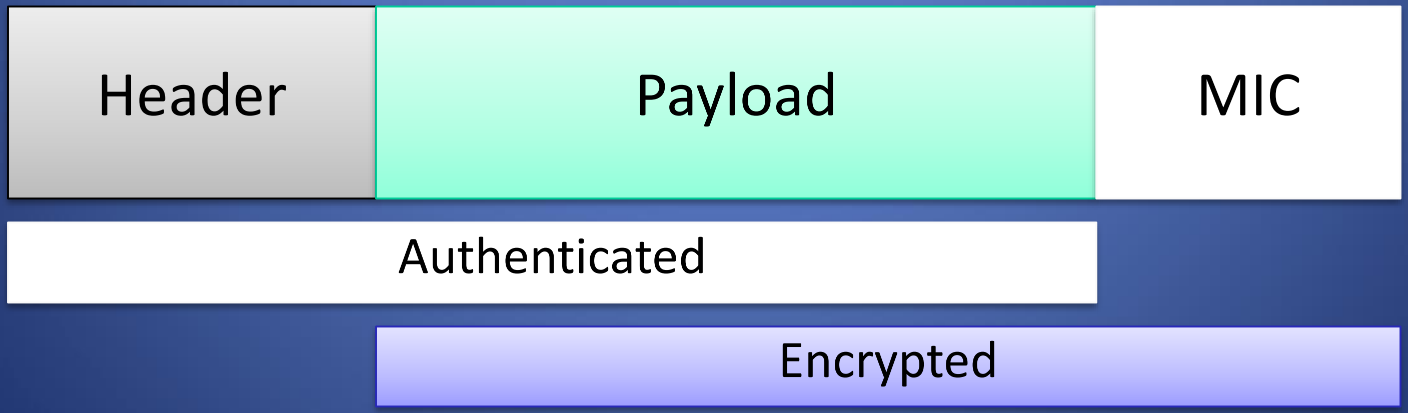
WPA2 Encryption

Counter Mode with Cipher Block Chaining Message Authentication Code Protocol

Compute a 64-bit message integrity code (MIC) on the plaintext header and the payload using the Michael algorithm

Encrypt the payload and MIC

Michael is not a strong cryptographic hash function



Alternatives and Add-Ons

WEP, WPA, and WPA2 all protect your traffic only up to the access point

- No security provided beyond access point

Other methods can encrypt end-to-end:

- SSL, SSH, VPN, PGP, and so on

End-to-end encryption is often simpler than setting up network-level encryption

Most of these solutions require per-application configuration