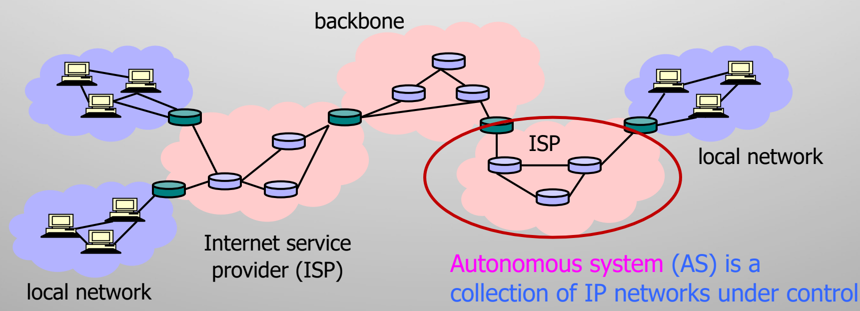
CS 203 / NETSYS 240

Network Threats/Attacks

1

•1

Internet Structure



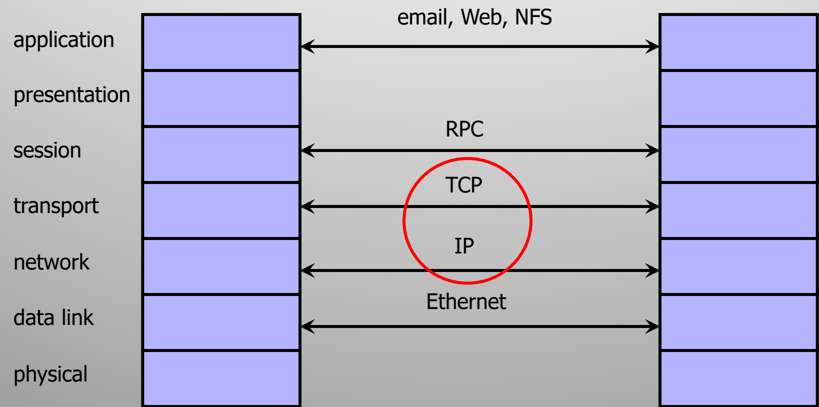
Autonomous system (AS) is a collection of IP networks under control of a single administrator (e.g., ISP)

- TCP/IP for packet routing and connections
- Border Gateway Protocol (BGP) for <u>external</u> route discovery
- Domain Name System (DNS) for IP address discovery

2

•2

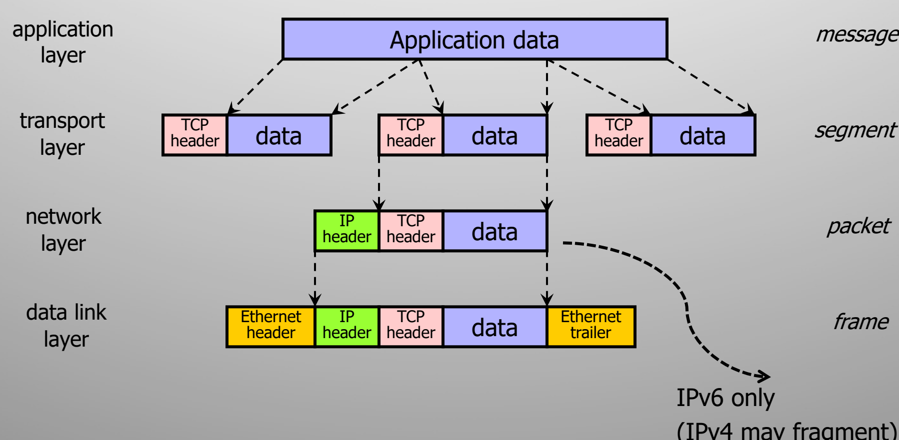
OSI Protocol Stack



3

•3

Data Formats



4

•4

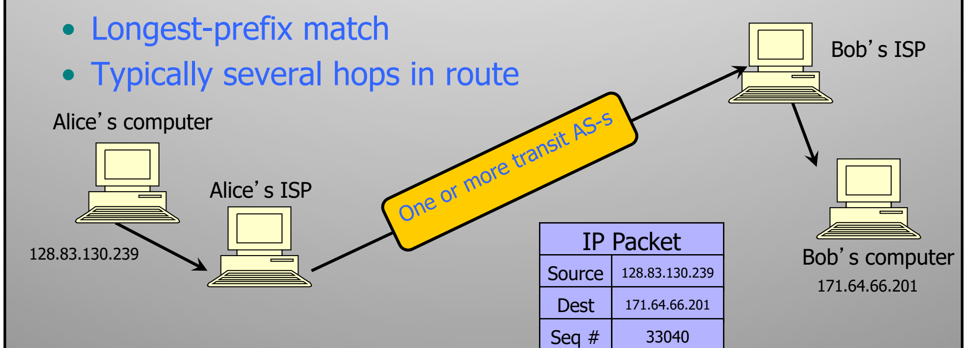
TCP (Transmission Control Protocol)

- Sender: break data into segments
  - Sequence number is attached to every packet
- Receiver: reassemble segments
  - Acknowledge receipt; lost packets are re-sent
- Connection state maintained by both sides

•5

IP (Internet Protocol)

- Connectionless
  - Unreliable, “best-effort” protocol
- Uses addresses (and prefixes thereof) used for routing
  - Longest-prefix match
  - Typically several hops in route



•6

ICMP (Control Message Protocol)

- Provides feedback about network operation
  - Out-of-band (control) messages carried in IP packets
  - Error reporting, congestion control, reachability, etc.
- Example messages:
  - Destination unreachable
  - Time exceeded
  - Parameter problem
  - Redirect to better gateway
  - Reachability test (echo / echo reply)
  - Timestamp request / reply

7

•7

Security Issues in TCP/IP

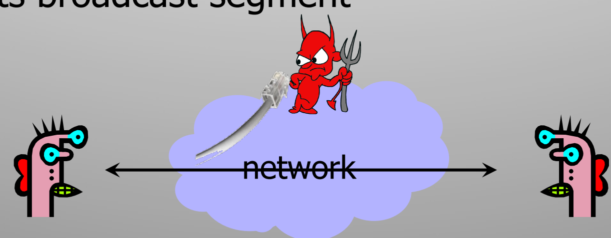
- Network packets pass by and thru untrusted hosts
  - Eavesdropping (packet sniffing)
- IP addresses are public
  - E.g., Ping-of-Death, Smurf attacks
- TCP connection requires state
  - SYN flooding
- TCP state easy to guess
  - TCP spoofing and connection hijacking

8

•8

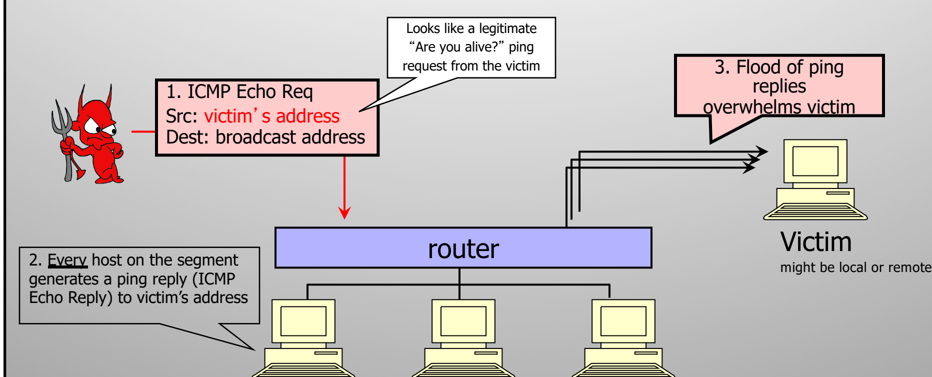
Packet Sniffing

- Many old applications send data unencrypted
  - Plain ftp, telnet send passwords in the clear (as opposed to sftp and ssh)
- Network Interface Card (NIC), e.g., Ethernet device, in "promiscuous mode" can read all data on its broadcast segment



Solution: encryption (e.g., IPsec), improved routing

"Smurf" Attack



Solution: reject external packets to broadcast addresses

"Ping of Death"

- When an old Windows machine receives an ICMP packet with payload over 64K, it crashes and/or reboots
  - Programming error in older versions of Windows
  - Packets of this length are illegal, so programmers of old Windows code did not account for them

Solution: patch OS, filter out ICMP packets

11

•11

"Teardrop" and "Bonk"

- TCP packets contain "Offset" field
  - # bytes since the start of TCP connection (unidirectional)
- Attacker sets Offset field to:
  - overlapping values
    - Bad/old implementation of TCP/IP stack crashes when attempting to re-assemble the fragments
  - … or to very large values
    - Target system crashes

Solution: use up-to-date TCP/IP implementation

12

•12

“LAND”

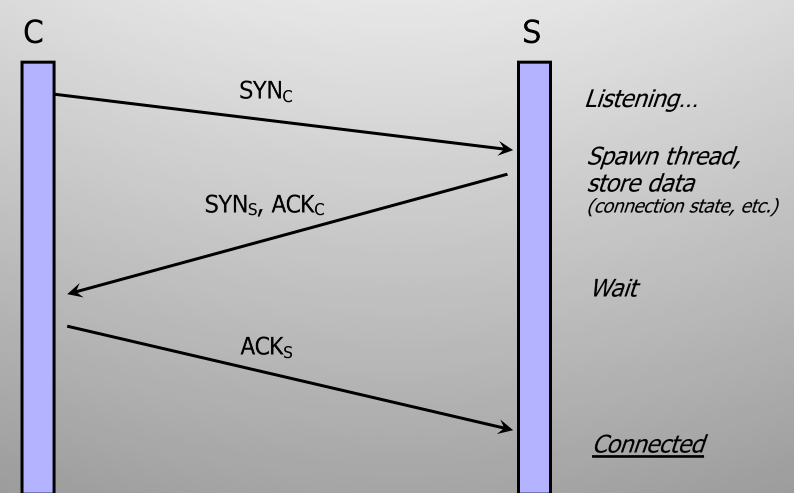
- Single-packet denial of service (DoS) attack
- IP packet with: <source-address,port> equal to <destination-address,port>, SYN flag set
- Triggers loopback in Windows XP SP2 implementation of TCP/IP stack
  - Locks up CPU

Solution: ingress filtering???

13

•13

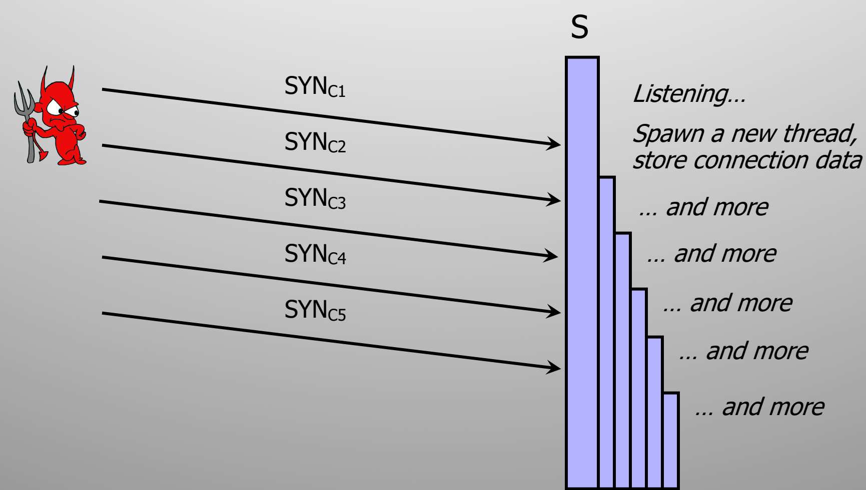
TCP Handshake Reminder



14

•14

SYN Flooding Attack



•15

SYN Flooding Explained

- Attacker sends many connection requests (SYNs) with spoofed source addresses
- Victim allocates resources for each request
  - New thread, connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a classic DoS attack example: ASYMMETRY!!!
  - Common pattern: it costs nothing to TCP client to send a connection request, but TCP server must spawn a thread for each request
  - Other examples of this behavior?

•16

Preventing Denial of Service

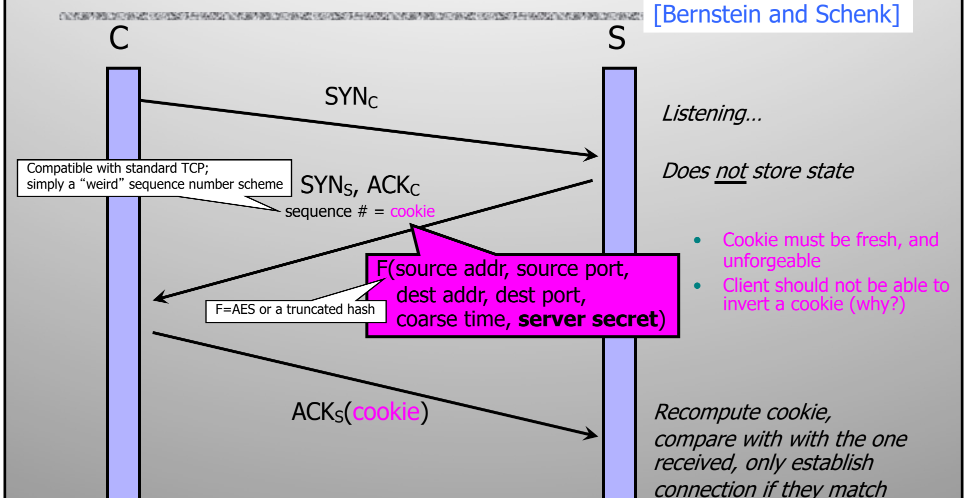
- DoS is caused by asymmetric state allocation
  - If server opens new state for each connection attempt, attacker can initiate many connections from bogus or forged IP addresses
- Cookies allow server to remain stateless until client produces:
  - Server state (IP addresses and ports) stored in a cookie and originally sent to client
- When client responds, cookie is verified

17

•17

SYN Cookies

[Bernstein and Schenk]



More info: http://cr.yp.to/syncookies.html

18

•18

Anti-Spoofing Cookies: Basic Pattern

- Client sends request (message #1) to server
- Typical protocol:
  - Server sets up connection, responds with message #2
  - Client may complete session or not (potential DoS)
- Cookie version:
  - Server responds with hashed connection data instead of message #2
    - Does not spawn any threads, does not allocate resources!
  - Client confirms by returning cookie (with other fields)
    - If source IP address is bogus, attacker can't confirm
    - WHY?

19

•19

Passive Defense: Random Deletion

SYN$_C$ →

half-open connections

121.17.182.45
231.202.1.16
121.100.20.14
5.17.95.155

- If SYN queue is full, delete random entry
  - Legitimate connections have a chance to complete
  - Fake addresses will be eventually deleted. WHY?
- Easy to implement

20

•20

TCP Connection Spoofing

- Each TCP connection has associated state
  - Sequence number, port number
- TCP state is easy to guess
  - Port numbers standard, seq numbers are *predictable*
- Can inject packets into existing connections
  - If attacker knows initial sequence number and amount of traffic, can guess current number
  - Guessing a 32-bit seq number is not practical, BUT…
  - Most systems accept a *large window* of sequence numbers (to handle massive packet losses)
  - Send a flood of packets with likely sequence numbers

21

•21

DoS by Connection Reset

- If attacker can guess the current sequence number for an existing connection, can send Reset packet to close it
- Especially effective against long-lived connections
  - For example, BGP route updates
    - Adjacent BGP routers keep long-lived TCP connections

23

•23

User Datagram Protocol (UDP)

- UDP – alternative to TCP, connectionless protocol
  - Simply sends datagram to application process at the specified port of the IP address
  - Source port number provides return address
  - Applications: media streaming, broadcast
- No acknowledgements, no flow control
- So…. DoS simply by UDP packet flooding is easy

24

•24

Countermeasures

- Above transport layer: Kerberos
  - Provides authentication, protects against application-layer spoofing
  - Does not protect against connection hijacking
- Above network layer: SSL/TLS and SSH
  - Protects against connection hijacking and injected data
  - Does not protect against DoS by spoofed packets
- Network (IP) layer: IPsec
  - Protects against hijacking, injection, DoS using connection resets, IP address spoofing
  - But muddled/poor key management…
- Below network layer?

25

•25

IP Routing

- Routing of IP packets is based on IP addresses
  - 32-bit host identifiers (128-bit in IPv6)
- Routers use a forwarding table (FIB)
  - Entry = [**destination, nxt hop, interface, metric**]
  - Table look-up for each IP packet to decide how to route it
- Routers learn routes to hosts and networks via routing protocols
  - Host identified by its IP address, network – by IP prefix
- BGP (Border Gateway Protocol) is the core Internet protocol for establishing inter-AS routes

26

•26

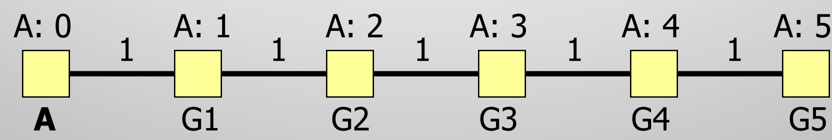
Distance-Vector Routing

- Each node keeps vector with distances to all nodes
- Periodically sends distance vector to all neighbors
- Neighbors reciprocate; node updates its vector based on received information
  - Bellman-Ford algorithm: for each destination, router picks the neighbor advertising the cheapest route, adds his entry into its own routing table and re-advertises
  - Used in RIP (routing information protocol)
- Split-horizon update
  - Do not advertise a route on an interface from which you learned the route in the first place!

27

•27

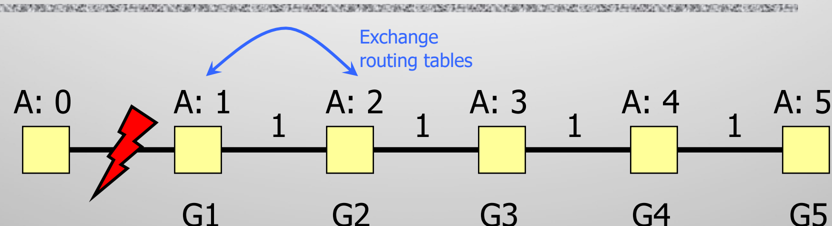
Good News Travels Fast



- G1 advertises route to network A with distance 1
- G2-G5 quickly learn the good news and install the routes to A (via G1) in their local routing tables

•28

Bad News Travels Slowly



- G1’s link to A goes down
- G2 is advertising a pretty good route to G1 (cost=2)
- G1’s packets to A are forever looping between G2 and G1
- G1 is now advertising a route to A with cost=3, so G2 updates its own route to A via G1 to have cost=4, and so on
  - G1 and G2 are slowly counting to infinity
  - Split-horizon updates only prevent two-node loops

•29

Overview of BGP

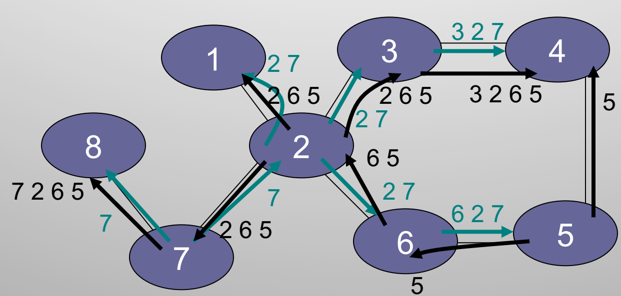
- BGP is a path-vector INTER-AS protocol
- Just like distance-vector, but routing updates, for each entry *also* contain an AS-level path to destination
  - List of traversed AS-s and a set of network prefixes belonging to the first AS on the list
- Each BGP router receives UPDATE messages from neighbors, selects one "best" path for each prefix, and advertises to its neighbors
  - Can be shortest path, but doesn't have to be
  - AS doesn't have to use the path it advertises!

30

•30

BGP Example

[Wetherall]



- AS 2 provides transit service for AS 7
  - Traffic to and from AS 7 travels through AS 2

31

•31

Some BGP Statistics

- BGP routing tables contain about 125,000 address prefixes mapping to about 17-18,000 paths
- Approx. 10,000 BGP routers
- Approx. 2,000 organizations are AS-es
- Approx. 6,000 organizations own prefixes
- Average route length (AS hops) is about 3.7
- 50% of routes have length less than 4 AS-s
- 95% of routes have length less than 5 AS-s

32

•32

BGP Misconfiguration

- **Blackholing:** Domain advertises good routes to addresses it does not know how to reach
  - Result: packets go into a network "black hole"
- April 25, 1997: "The day the Internet died"
  - AS 7007 (Florida Internet Exchange) de-aggregated the BGP route table and re-advertised all prefixes as if it originated paths to them
  - In effect, AS 7007 was advertising that it has the best route to <u>every host on the Internet</u>
  - Huge network instability as incorrect routing data propagated and routers crashed under traffic

33

•33

BGP Security

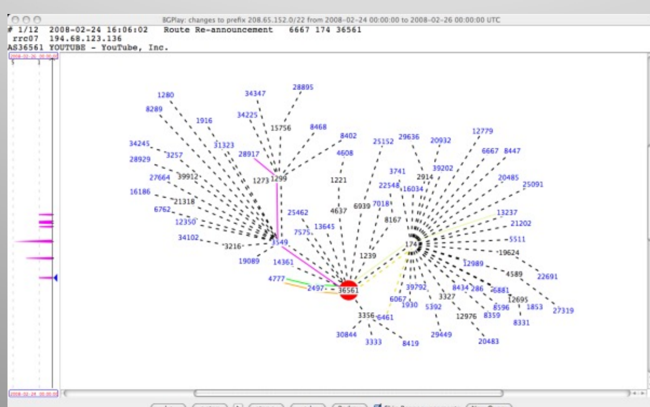
- BGP update messages contain no authentication or integrity protection
  - However, today BGP updates are sent over secure tunnels
- Attacker may falsify advertised routes
  - Modify IP prefixes associated with the route
    - Can blackhole traffic to certain IP prefixes
  - Change AS path
    - Either attract traffic to attacker's AS, or divert traffic away
    - Interesting economic incentive: an ISP wants to dump its traffic on other ISPs without routing their traffic in exchange
  - Re-advertise/propagate AS path without permission
    - For example, multi-homed customer may end up advertising transit capability between two large ISPs

34

•34

YouTube (Normally)

- AS36561 (YouTube) advertises 208.65.152.0/22

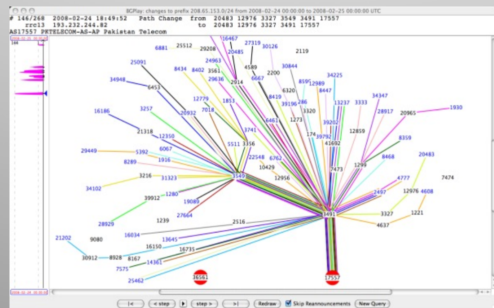


35

•35

YouTube (February 24, 2008)

- Pakistan government wants to block YouTube
  - AS17557 (Pakistan Telecom) advertises 208.65.153.0/24
  - All YouTube traffic worldwide directed to AS17557



- Result: two-hour YouTube outage

36

•36

Other BGP Incidents

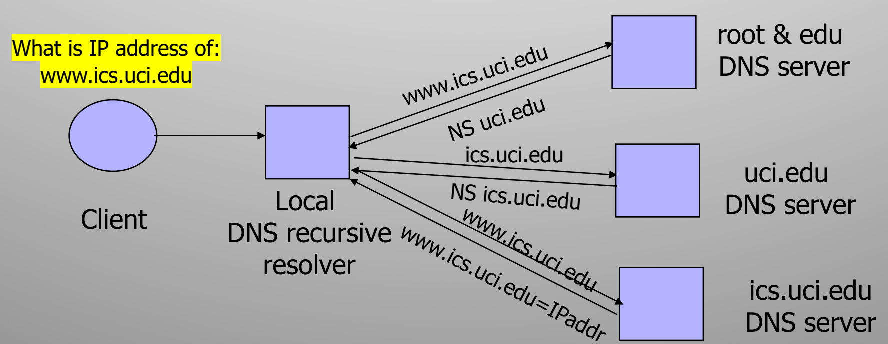
- May 2003: Spammers hijack unused block of IP addresses belonging to Northrop Grumman
  - Entire Northrop Grumman ends up on spam blacklist
  - Took two months to reclaim ownership of IP addresses
- May 2004: Malaysian ISP hijacks prefix of Yahoo California data center
- Dec 2004: Turkish ISP advertises routes to the entire Internet, including Amazon, CNN, Yahoo
- April 2021 FB outage: BGP withdrew routes to IP prefixes for FB's DNS servers: couldn't resolve any FB domain names for over 7 hours.

37

•37

DNS: Domain Name Service

DNS maps symbolic names to numeric IP addresses
(for example, www.uci.edu ↔ 128.195.188.233)

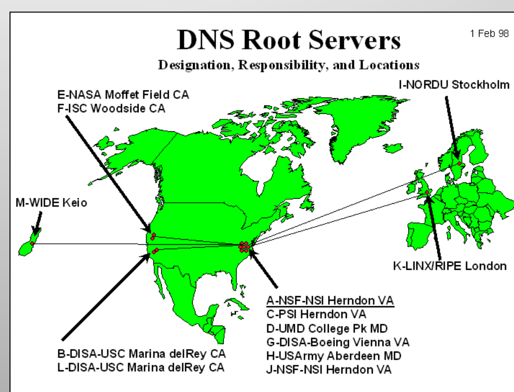


38

•38

DNS Root Name Servers

- Root name servers for top-level domains
- Authoritative name servers for subdomains
- Local name resolvers contact authoritative servers when they do not know a name



Feb 6, 2007: DoS attack on root DNS servers

39

•39

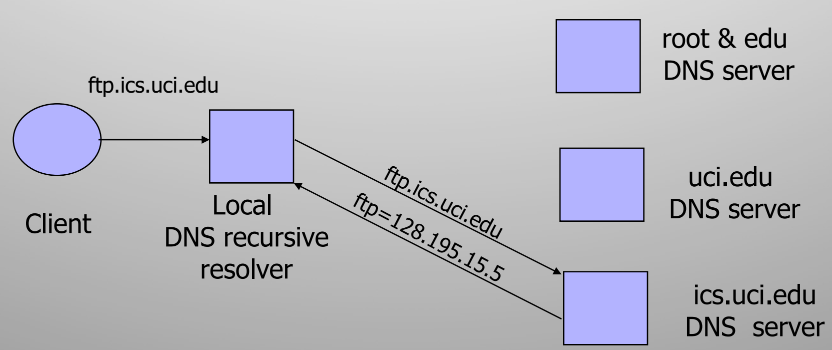
DNS Caching

- DNS responses are cached
  - Quick response for repeated queries
  - Other queries may reuse some parts of lookup
    - NS records for domains
- DNS negative queries are cached
  - So as not to repeat past mistakes, e.g., typos
- Cached data periodically times out
  - Lifetime (TTL) controlled by owner of record
  - TTL passed with every record

40

•40

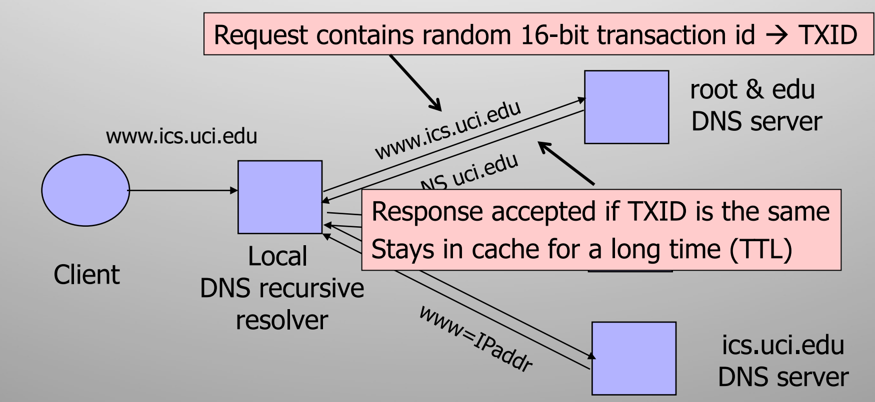
Cached Lookup Example



41

•41

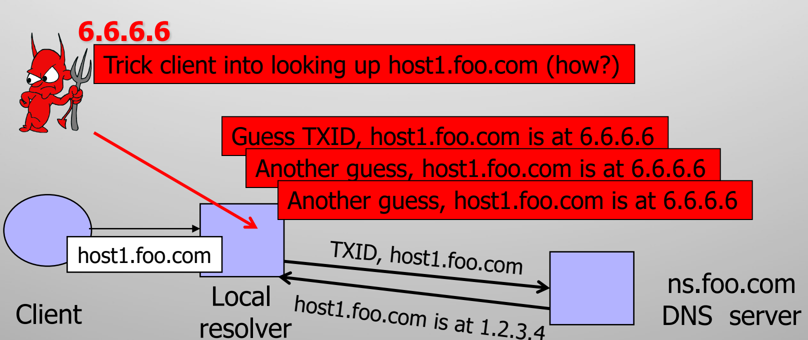
DNS “Authentication”



42

•42

DNS Spoofing



- Several opportunities to win the race
- If attacker loses, has to wait until TTL expires

  … but can try again with host2.foo.com, host3.foo.com, etc.

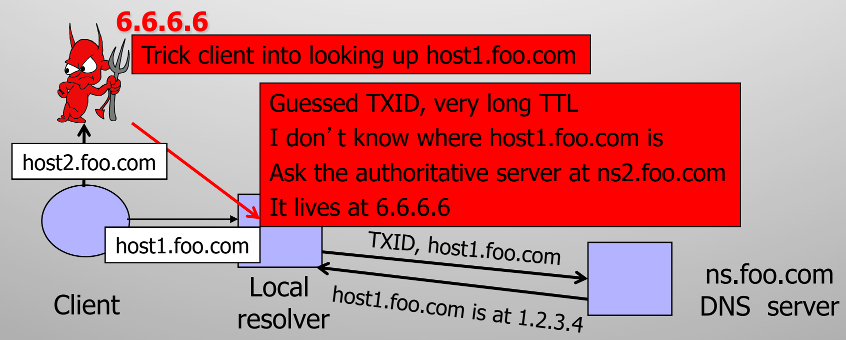
  … but what’s the point of hijacking host3.foo.com?

43

•43

Exploiting Recursive Resolving

[Kaminsky]



- *If attacker wins, all future DNS requests for foo.com will go to 6.6.6.6*
- *The cache is now poisoned… for a very long time!*
- *No need to win future races!*

44

•44

Triggering DNS Lookup

- Any link, any image, any ad, anything can cause a DNS lookup
  - No Javascript required, though it helps
- Mail servers will look up what bad guy wants
  - Upon first greeting: HELLO
  - Upon first learning who they're talking to: MAIL FROM
  - Upon spam check (oops!)
  - When trying to deliver a bounce
  - When trying to deliver a newsletter

45

•45

Reverse DNS Spoofing

- Trusted access is often based on host names
  - E.g., permit all hosts in .rhosts to run remote shell
- Network requests such as rsh or rlogin arrive from numeric (IP) source addresses
  - System performs reverse DNS lookup to determine requester's host name and checks if it's in .rhosts
- If attacker can spoof the answer to reverse DNS query, he can fool target machine into thinking that request comes from an authorized host
  - No authentication for DNS responses and typically no double-checking (numeric → symbolic → numeric)

46

•46

Pharming

- Many anti-phishing defenses rely on DNS
- Can bypass them by poisoning DNS cache and/or forging DNS responses
  - Browser: give me the address of www.paypal.com
  - Attacker: sure, it' s 6.6.6.6 (attacker-controlled site)
- Dynamic pharming
  - Provide bogus DNS mapping for the trusted server, trick user into downloading a malicious script from evil server
  - Force user to download content from the real trusted server, by temporarily providing correct DNS mapping
  - Malicious script and content have the same origin!

    Thus, malicious script can access (sensitive) content

47

•47

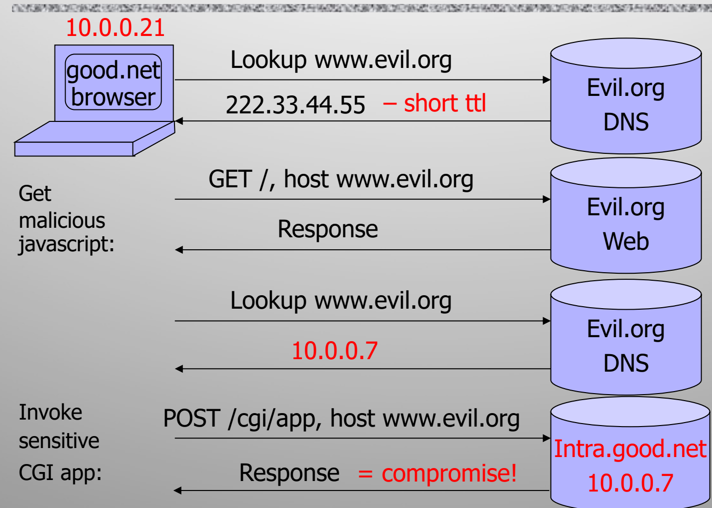
JavaScript/DNS Intranet attack (I)

- Consider a web server, hostname: intra.good.net
  - IP: 10.0.0.7, inaccessible outside **good.net** network
  - Hosts sensitive CGI applications
- Attacker at evil.org gets good.net user to browse www.evil.org (e.g., via a link in email msg or an ad on 3rd party web page)
- Places JavaScript on www.evil.org, which, when invoked by client browser, accesses sensitive CGI applications on intra.good.net
  - This wouldn't work since JavaScript is subject to same origin policy -- user's browser tries to prevent client-side scripts from different places
  - But, suppose that attacker controls DNS

48

•48

JavaScript/DNS Intranet attack (II)



Javascript's "same origin" policy is now satisfied
(malicious script can access results of cgi script)

49

•49

Other DNS Vulnerabilities

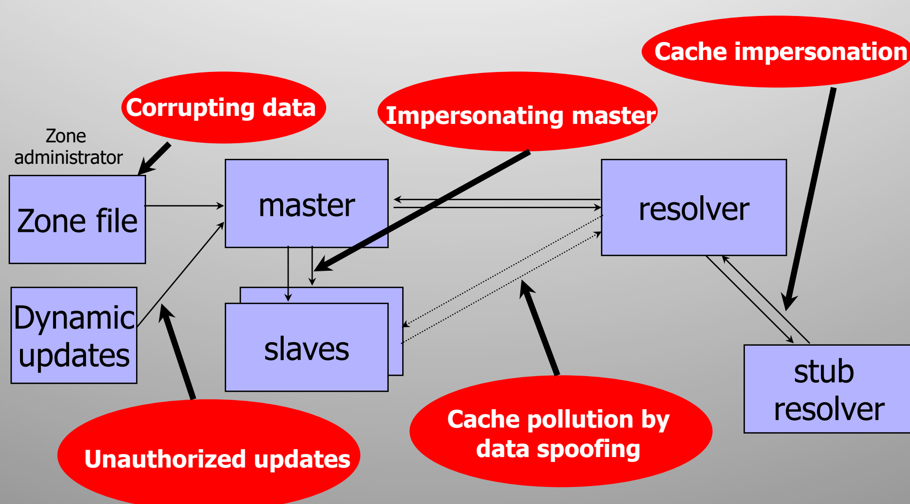
- DNS implementations can also have vulnerabilities
  - Reverse query buffer overrun in old releases of BIND
  - MS DNS for NT 4.0 crashes on chargen stream
- Denial of service
  - Oct 2002: ICMP flood took out 9 root servers for 1 hour
- Can use "zone transfer" requests to download DNS database and map out the network
  - "The Art of Intrusion": NYTimes.com and Excite@Home
  - Solution: block port 53 (zone transfer) on corporate name servers

See http://cr.yp.to/djbdns/notes.html

50

•50

DNS Vulnerabilities: Summary



51

•51

Domain Hijacking and Other Risks

- Spoofed ICANN registration and domain hijacking
  - Authentication of domain transfers based on email addr
  - Aug'04: teen hacker hijacks eBay's German site
  - Jan'05: hijacking of panix.com (oldest ISP in NYC)
    - "The ownership of panix.com was moved to a company in Australia, the actual DNS records were moved to a company in the United Kingdom, and Panix.com's mail was redirected to yet another company in Canada."
  - Many other domain theft attacks
- Misconfiguration and human error

ICANN: Internet Corporation for Assigned Names and Numbers

52

•52

Solving the DNS Spoofing Problem

- Long TTL for legitimate responses
  - Does it really help?
- Randomize port in addition to TXID
  - 32 bits of randomness – makes it harder for attacker to guess TXID
- DNSSEC
  - Cryptographic authentication of host-address mappings

53

•53

DNSSEC

- Goals: authentication and integrity of DNS requests and responses
- PK-DNSSEC (public key)
  - DNS server signs its data (can be done in advance)
  - How do other servers learn the public key?
- SK-DNSSEC (symmetric key)
  - Encryption and MAC: $E_k(m, MAC(m))$
  - Each message contains a nonce to avoid replay
  - Each DNS node shares a symmetric key with its parent
  - Zone root server has a public key (hybrid approach)

MORE INFO: http://www.dnssec.net/presentations

54

•54