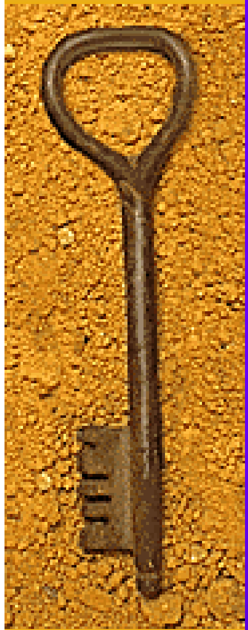
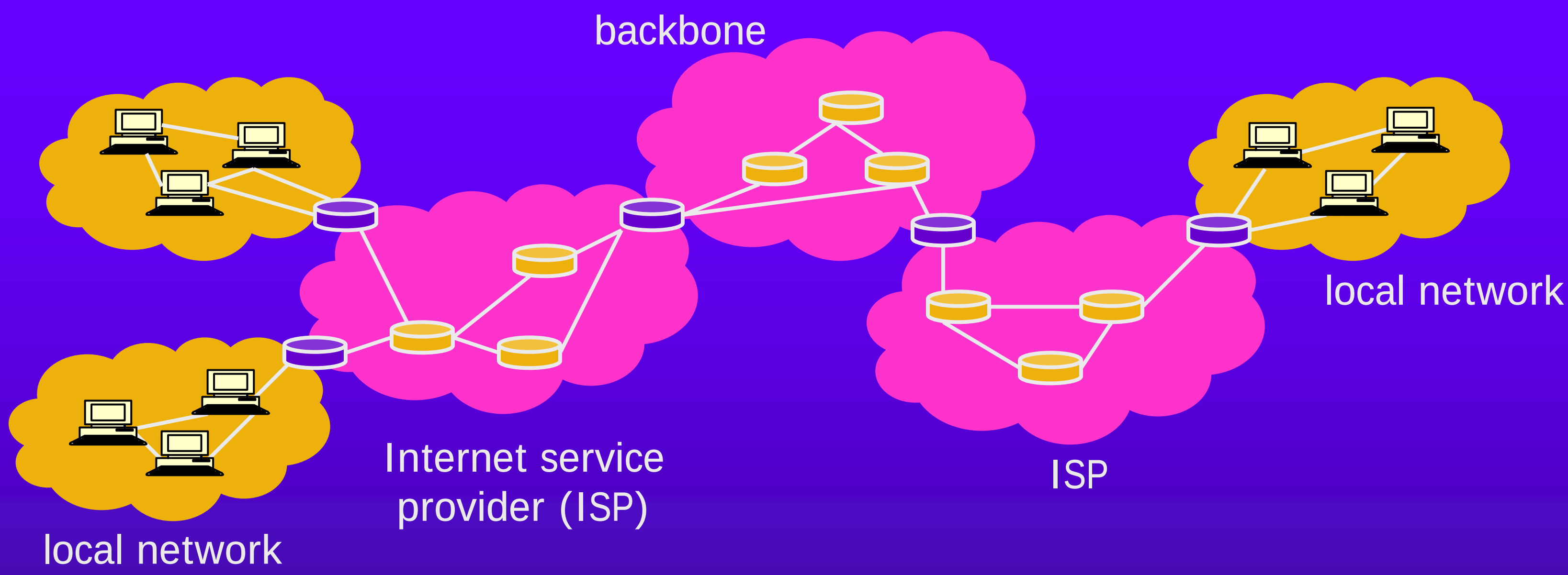
Attacks on TCP/IP and DNS



Agenda

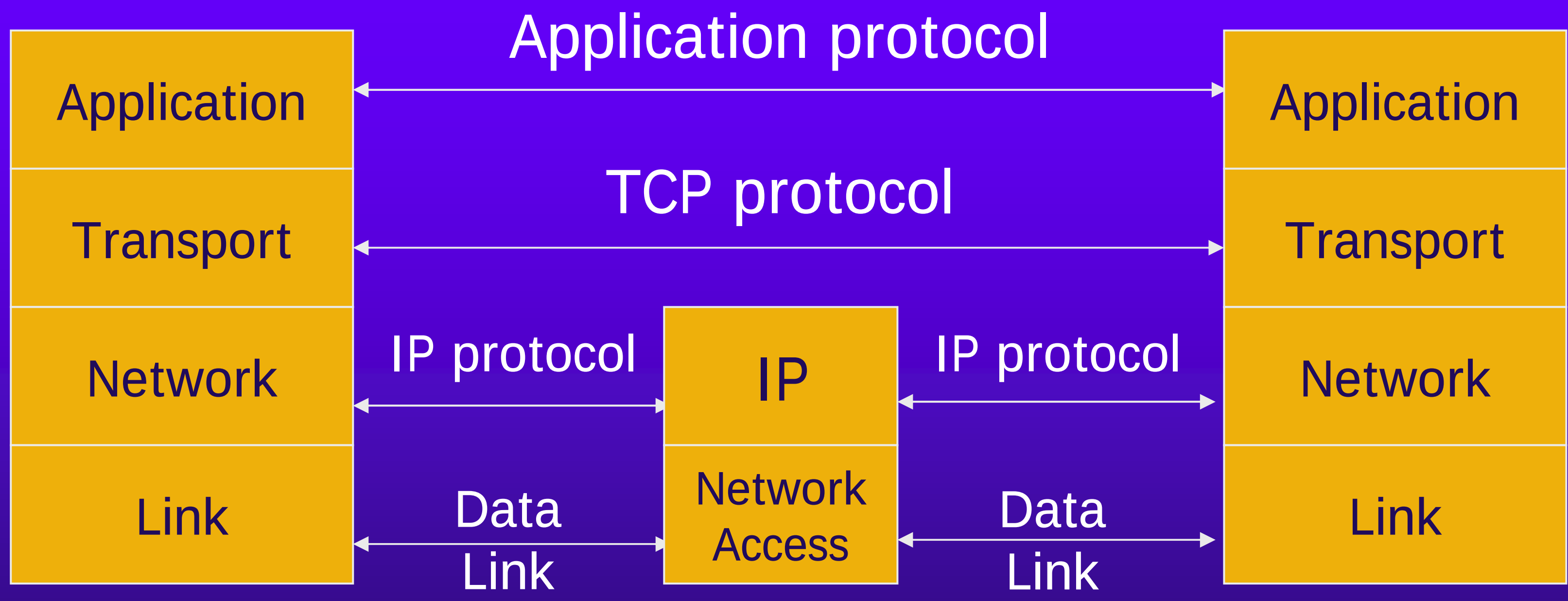
- Brief Introduction to TCP/IP network
- Security Issues in TCP/IP
- DNS Security
- Router Security

Internet Infrastructure

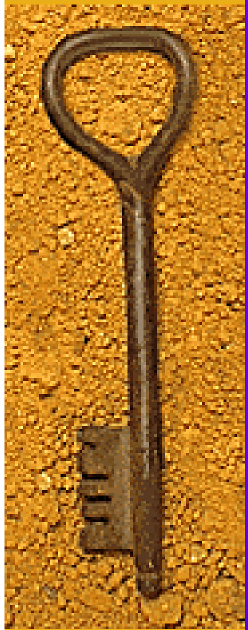


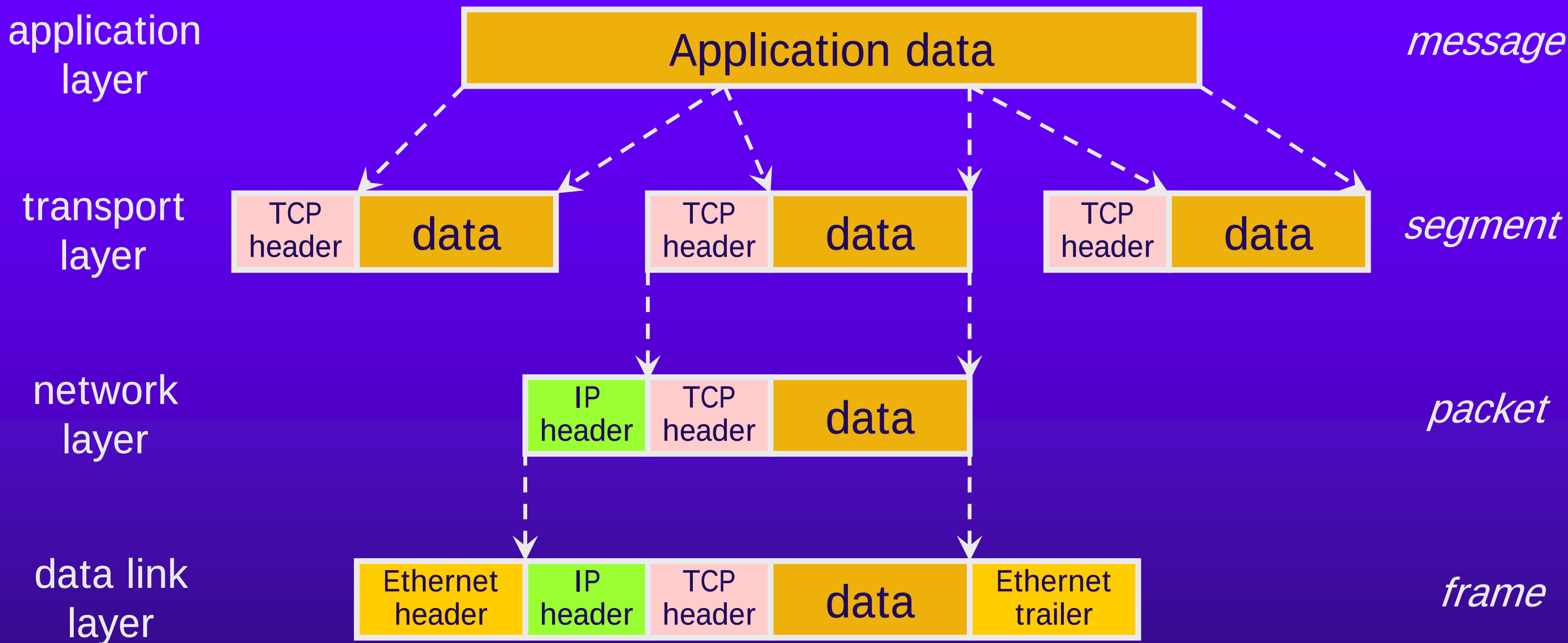
- TCP/IP for packet routing and connections
- Border Gateway Protocol (BGP) for route discovery
- Domain Name System (DNS) for IP address discovery

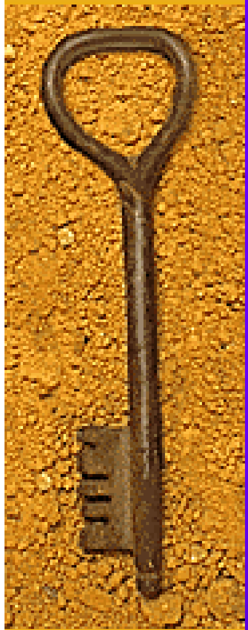
TCP Protocol Stack



Data Formats

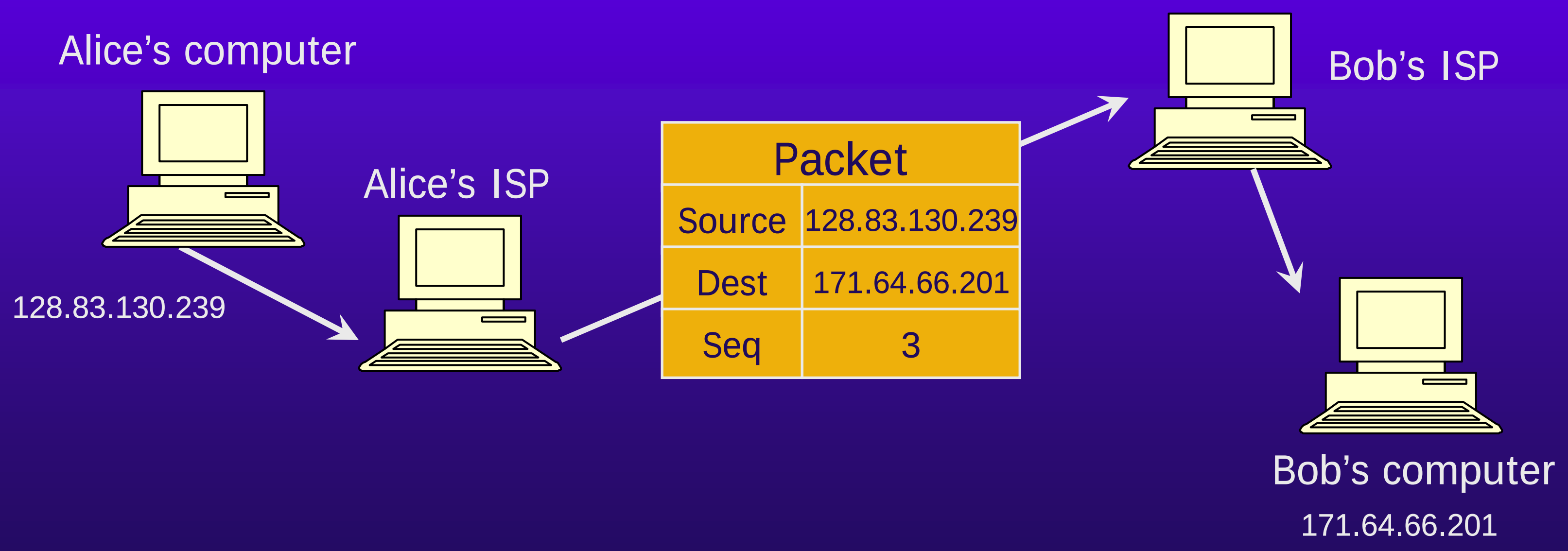


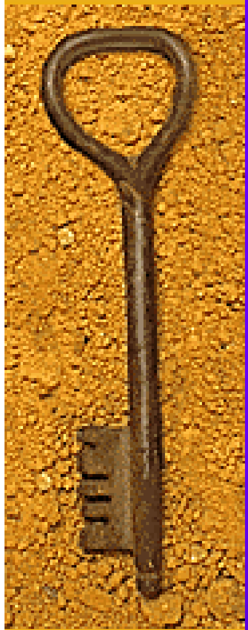




IP (Internet Protocol)

- Connectionless
  - Unreliable, “best-effort” protocol
- Uses numeric addresses for routing
  - Typically several hops in the route

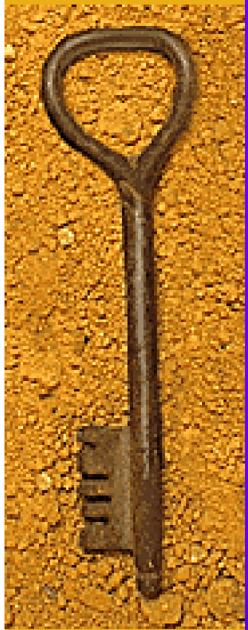




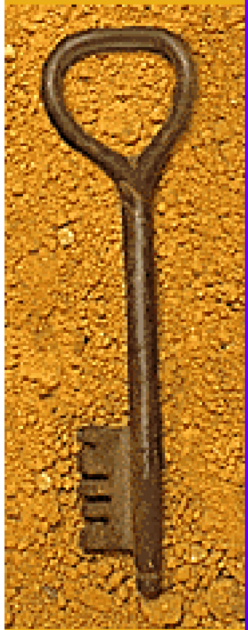
User Datagram Protocol

- IP provides routing
  - IP address gets datagram to a specific machine
- UDP separates traffic by port
  - Destination port number gets UDP datagram to particular application process, e.g., 128.3.23.3, 53
  - Source port number provides return address
- Minimal guarantees (... mice and elephants)
  - No acknowledgment
  - No flow control
  - No message continuation

Transmission Control Protocol

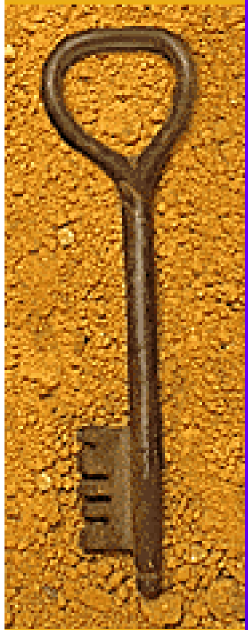


- Sender: break data into packets
  - Sequence number is attached to every packet
- Receiver: reassemble packets in correct order
  - Acknowledge receipt; lost packets are re-sent
- Connection state maintained on both sides



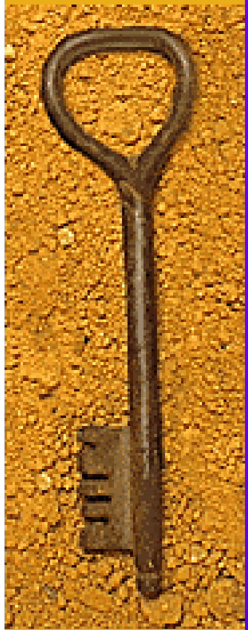
ICMP (Control Message Protocol)

- Provides feedback about network operation
  - "Out-of-band" 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)
  - Message transit delay (timestamp request / reply)



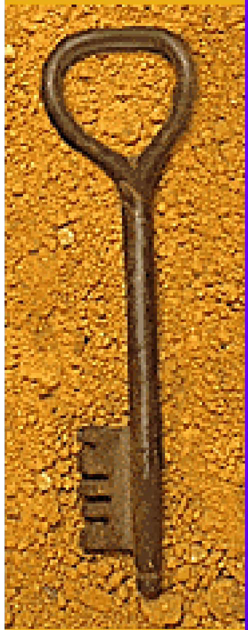
Agenda

- Brief Introduction to TCP/IP network
- Security Issues in TCP/IP
- DNS Security
- Router Security



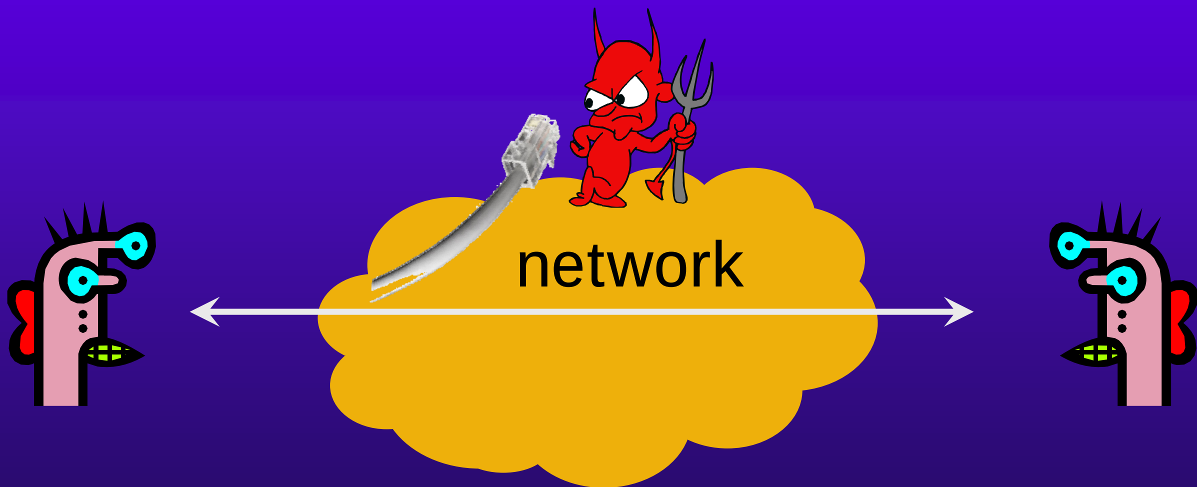
Security Issues in TCP/IP

- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public: Smurf attacks
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding

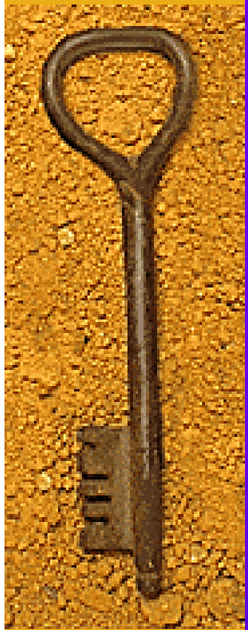


Packet Sniffing

- Many applications send data unencrypted
  - ftp, telnet send passwords in the clear
- Network interface card (NIC) in "promiscuous mode" reads all passing data
- Also in Switch, Router . . .

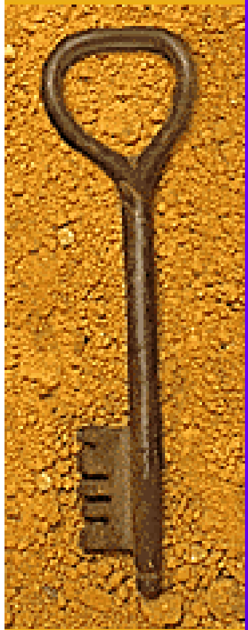


Solution: encryption (e.g., IPSec), improved routing



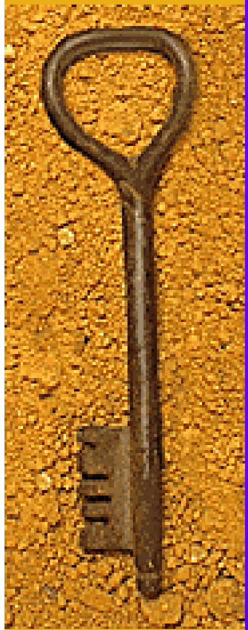
Problem with Switches? Flood it!

- The switch stores MAC addresses locally
- Dsniff keeps sending the switch bogus MAC address
- Eventually the switches memory fills and it turns into a hub
- Then, just run any sniffer you want to get data from the network



Security Issues in TCP/IP

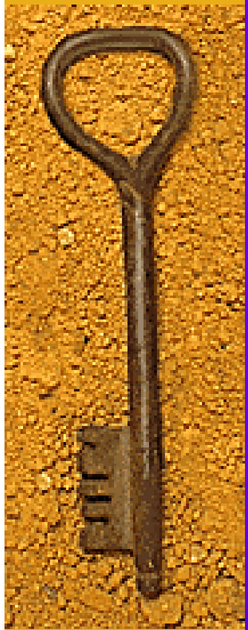
- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public: Smurf attacks
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding



“Ping of Death”

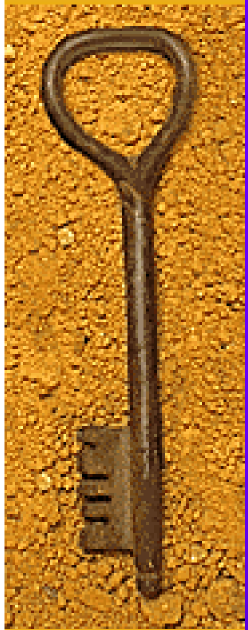
- If an old Windows machine received an ICMP packet with a payload longer than 64K, machine would crash or reboot
  - Programming error in older versions of Windows
  - Packets of this length are illegal, so programmers of Windows code did not account for them

Solution: patch OS, filter out ICMP packets



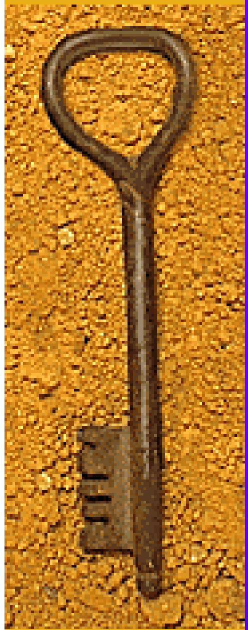
Security Issues in TCP/IP

- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public: Smurf attacks
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding

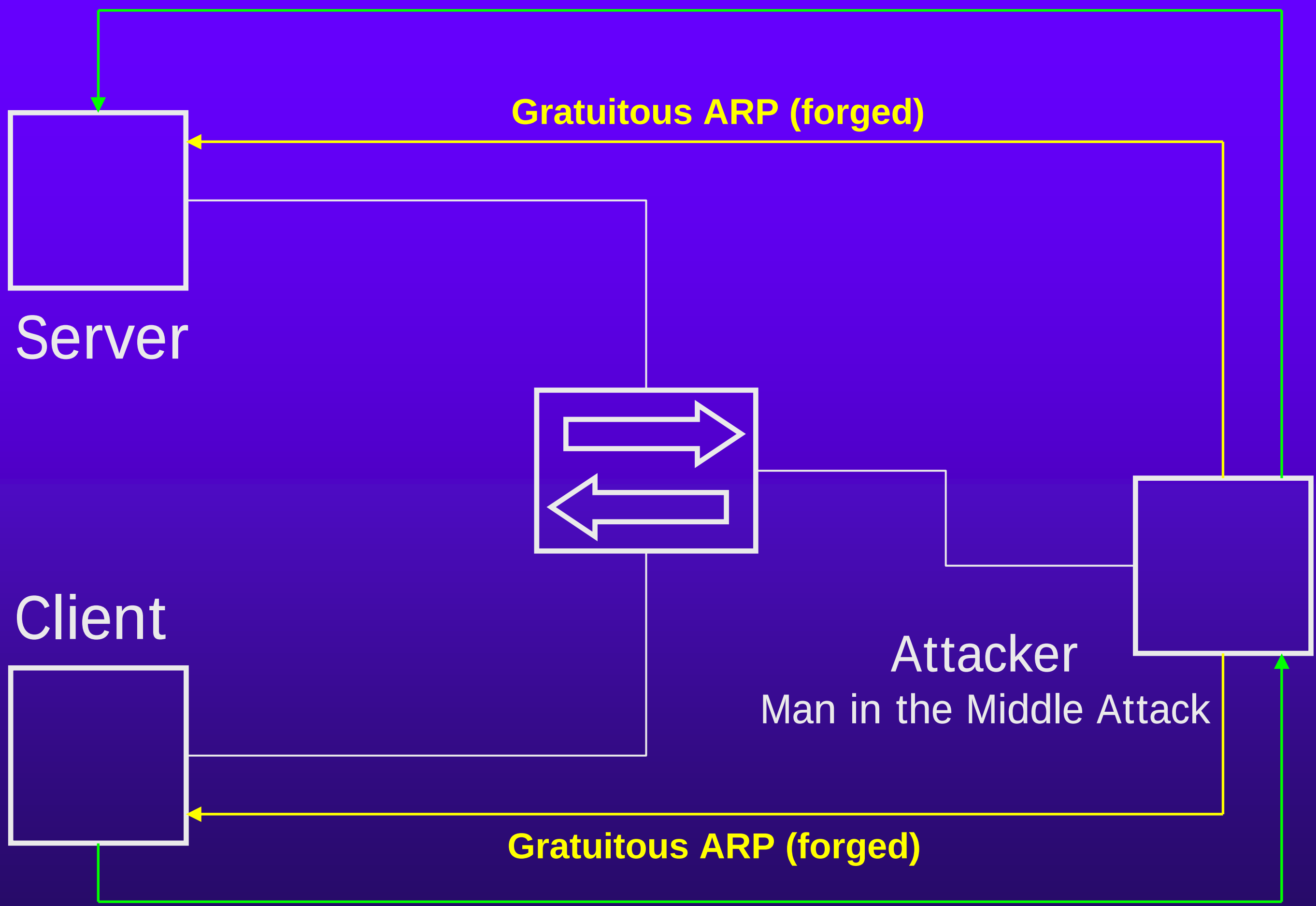


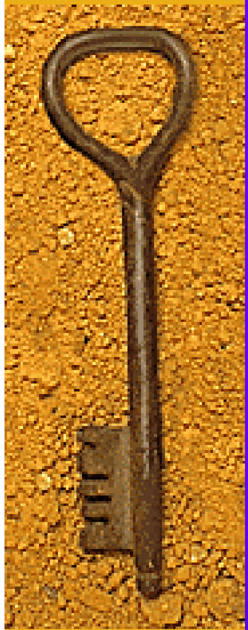
ARP Poisoning

- ARP is stateless (we all knows how it works and what the problems are)
- Some operating systems do not update an entry if it is not already in the cache, others accept only the first received reply (e.g. Solaris)
- The attacker can forge spoofed ICMP packets to force the host to make an ARP request. Immediately after the ICMP it sends the fake ARP reply



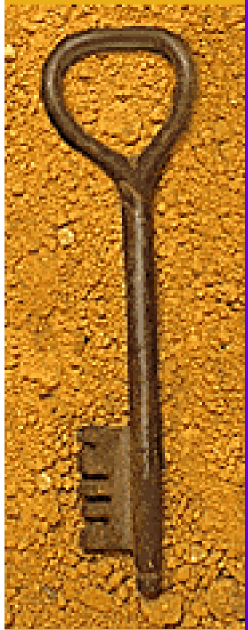
ARP Poisoning: The Scenario





ARP Poisoning: Tools

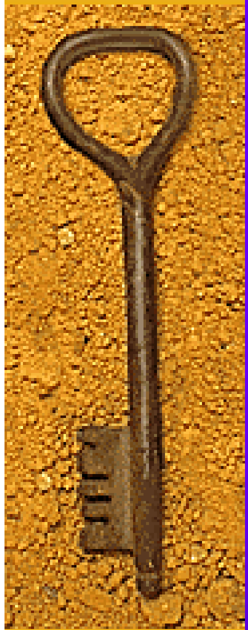
- ettercap (http://ettercap.sourceforge.net)
  - Poisoning
  - Sniffing
  - Hijacking
  - Filtering
  - SSH v.1 sniffing (transparent attack)
- dsniff (http://www.monkey.org/~dugsong/dsniff)
  - Poisoning
  - Sniffing
  - SSH v.1 sniffing (proxy attack)



ARP Poisoning: Countermeasures

- YES - passive monitoring (arpwatch)
- YES - active monitoring (ettercap)
- YES - IDS (detect but not avoid)

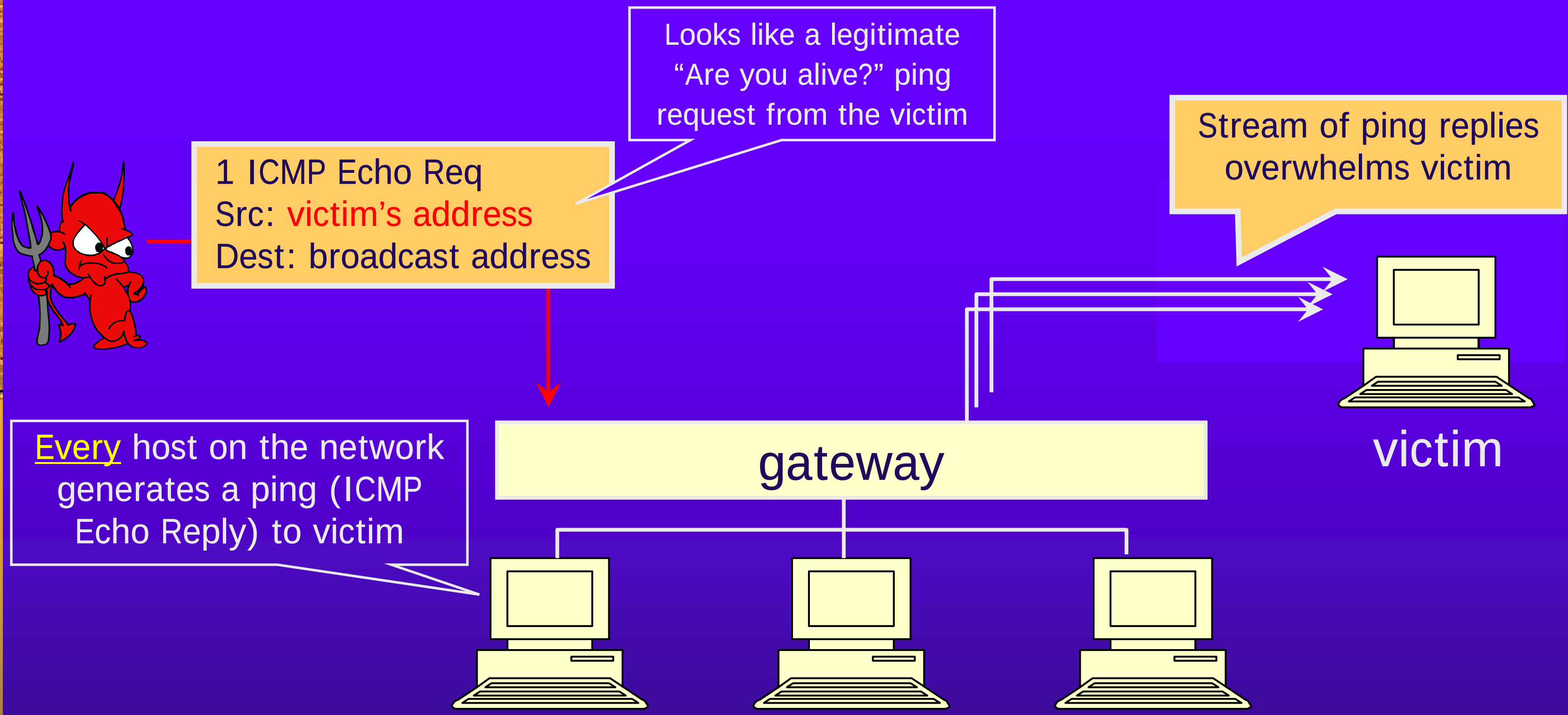
- YES - Static ARP entries (avoid it)
- YES - Secure-ARP (public key authentication)



Security Issues in TCP/IP

- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public
  - Smurf attacks, Source Routing
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding

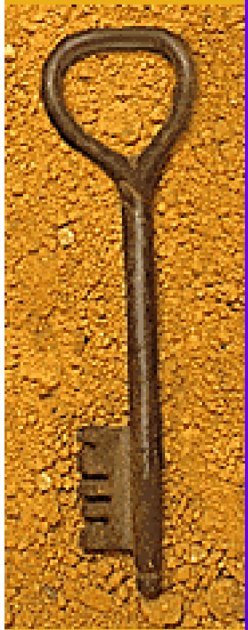
Smurf Attack



Solution: gateway reject external packets to broadcast addresses

Can not stop local smurf attack. But it's easy to detect.

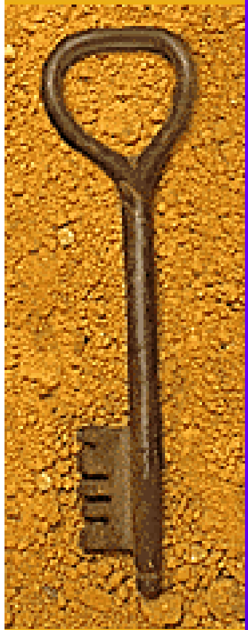
Spoofing with Source Routing

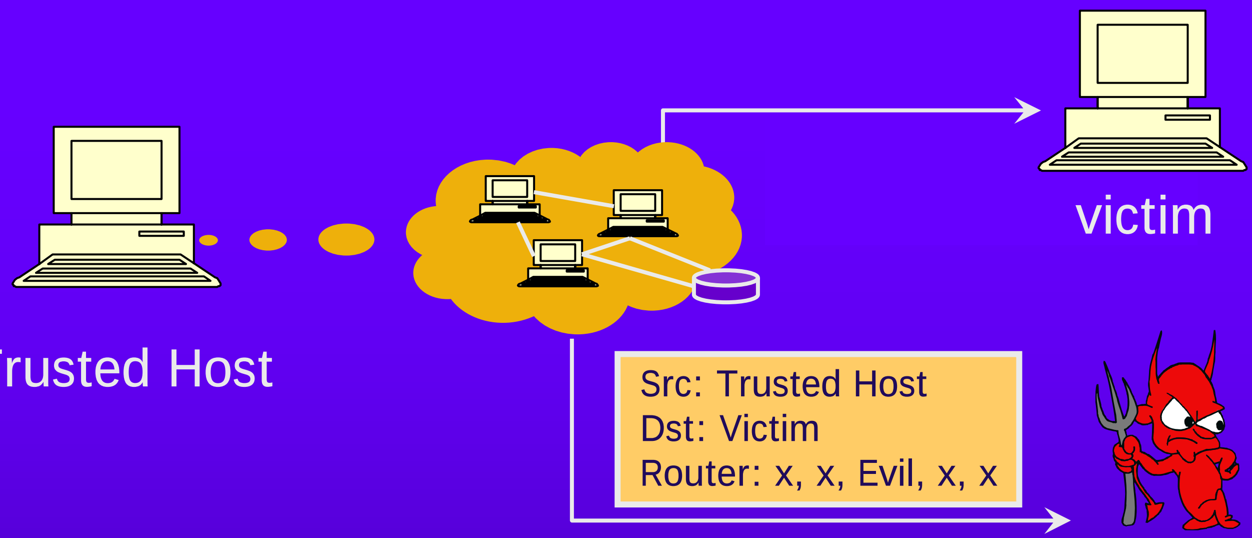


- Source routing allows the attacker to specify a certain path the packet will take on the network
- "loose source routing" allows the attacker to tell the computer some hops but not all

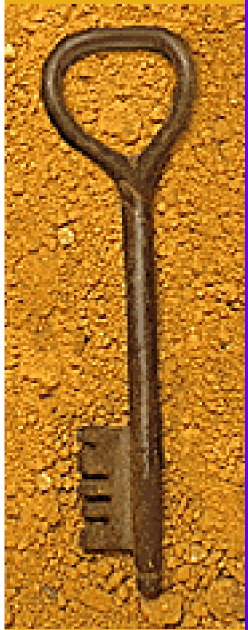
Source IP Address:
Destination IP Address:
Router(Partial Router): x, x, x, x, x

Spoofing with Source Routing





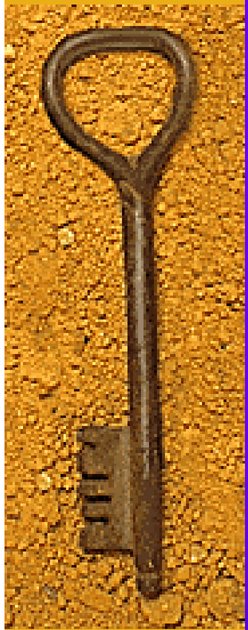
- The attacker sets source routed packets from a fake source IP (trusted by the victim) to the victim
- Include the attackers IP address as one of the hops
- When the victim's computer tries to establish a three-way-handshake the attacker intercepts the SYN-ACK and submits its own ACK
- An open connection has been established between the attacker and victim, the attacker can view the responses from the victim

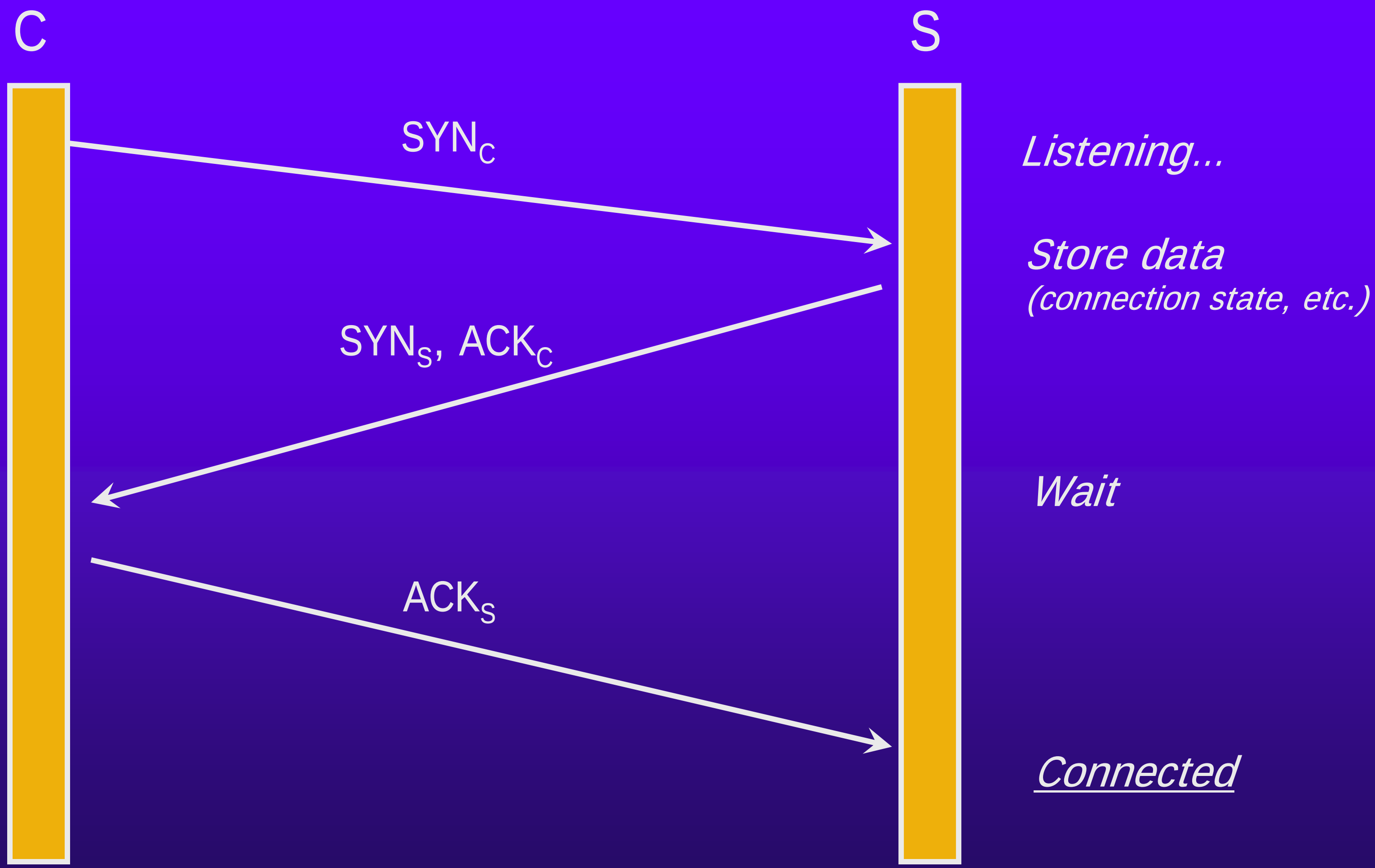


Security Issues in TCP/IP

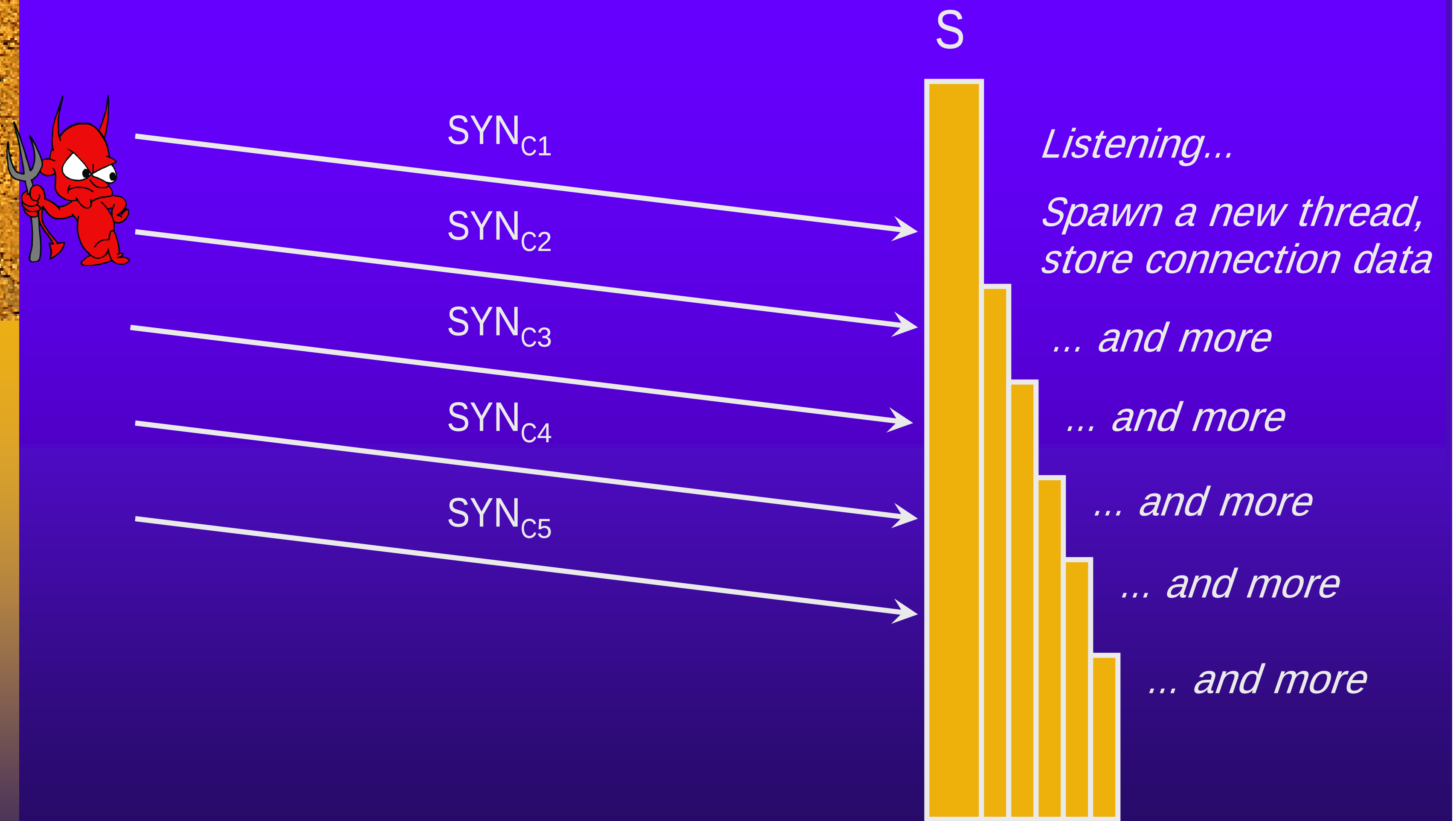
- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public: Smurf attacks
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding

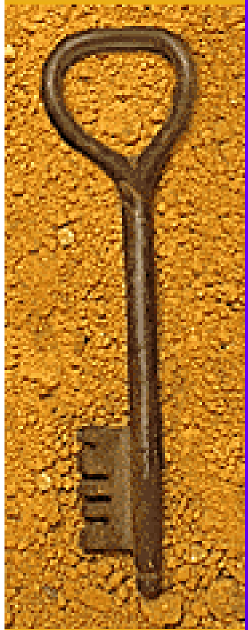
TCP Handshake





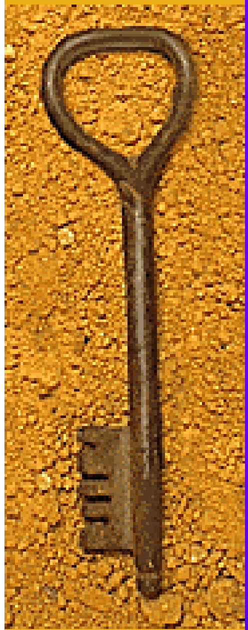
SYN Flooding Attack





SYN Flooding Explained

- Attacker sends many connection requests with spoofed source addresses
- Victim allocates resources for each request
  - Connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a classic denial of service (DoS) attack
  - Common pattern: it costs nothing to TCP initiator to send a connection request, but TCP responder must allocate state for each request (asymmetry!)



Preventing Denial of Service

$\xrightarrow{SYN_C}$

half-open connections

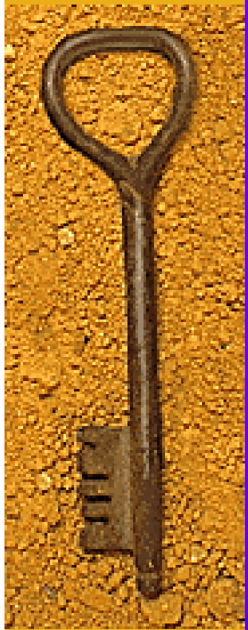
121.17.182.45
231.202.1.16
121.100.20.14
5.17.95.155

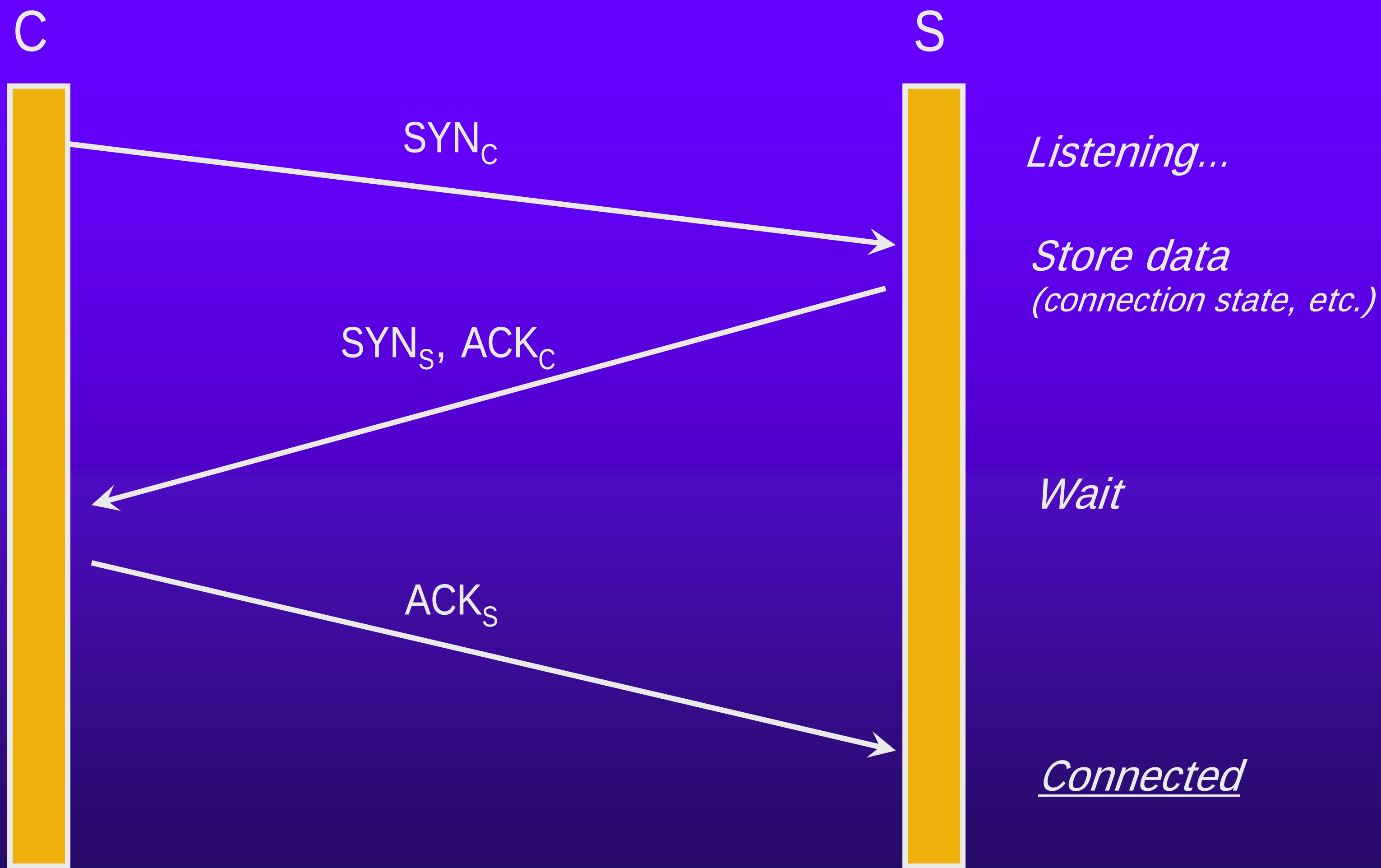
- If SYN queue is full, delete random entry
  - Legitimate connections have a chance to complete
  - Fake addresses will be eventually deleted
- Easy to implement

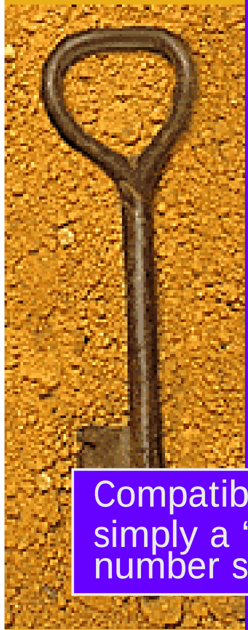
Another Defense: SYN-ACK Cookies

- DoS is caused by asymmetric state allocation
  - If server opens a state for each connection attempt, attacker can initiate thousands of connections from bogus or forged IP addresses
- Cookies ensure that the server is stateless until client produced at least 2 messages

TCP Handshake







SYN Cookies

[Bernstein and Schenk]

C

S

SYN_C

Listening…

Does not store state

Compatible with standard TCP; simply a "weird" sequence number scheme

SYN_S, ACK_C

sequence # = cookie

F(source addr, source port, dest addr, dest port, coarse time, **server secret**)

F=Rijndael or crypto hash

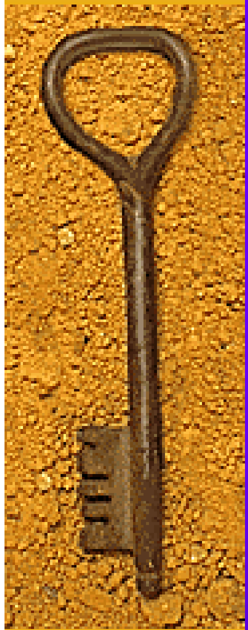
Cookie must be unforgeable and tamper-proof (why?)

Client should not be able to invert a cookie (why?)

ACK_S(cookie)

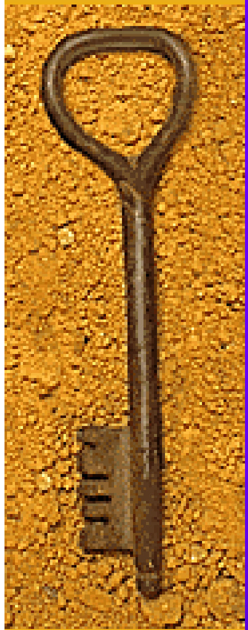
Recompute cookie, compare with with the one received, only establish connection if they match

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



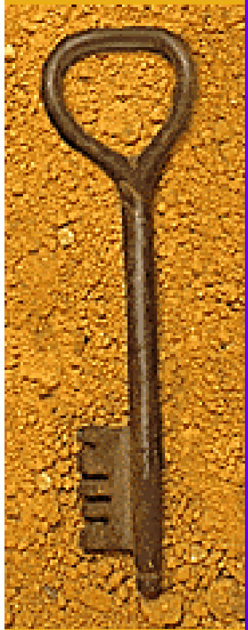
Examples of SYN-ACK Cookies

- SYN cookies are now a standard part of Linux and FreeBSD.
  - But, they are not enabled by default under Linux.
  - To enable, add the following line to your boot scripts
  - echo 1 > /proc/sys/net/ipv4/tcp_syncookies.



Security Issues in TCP/IP

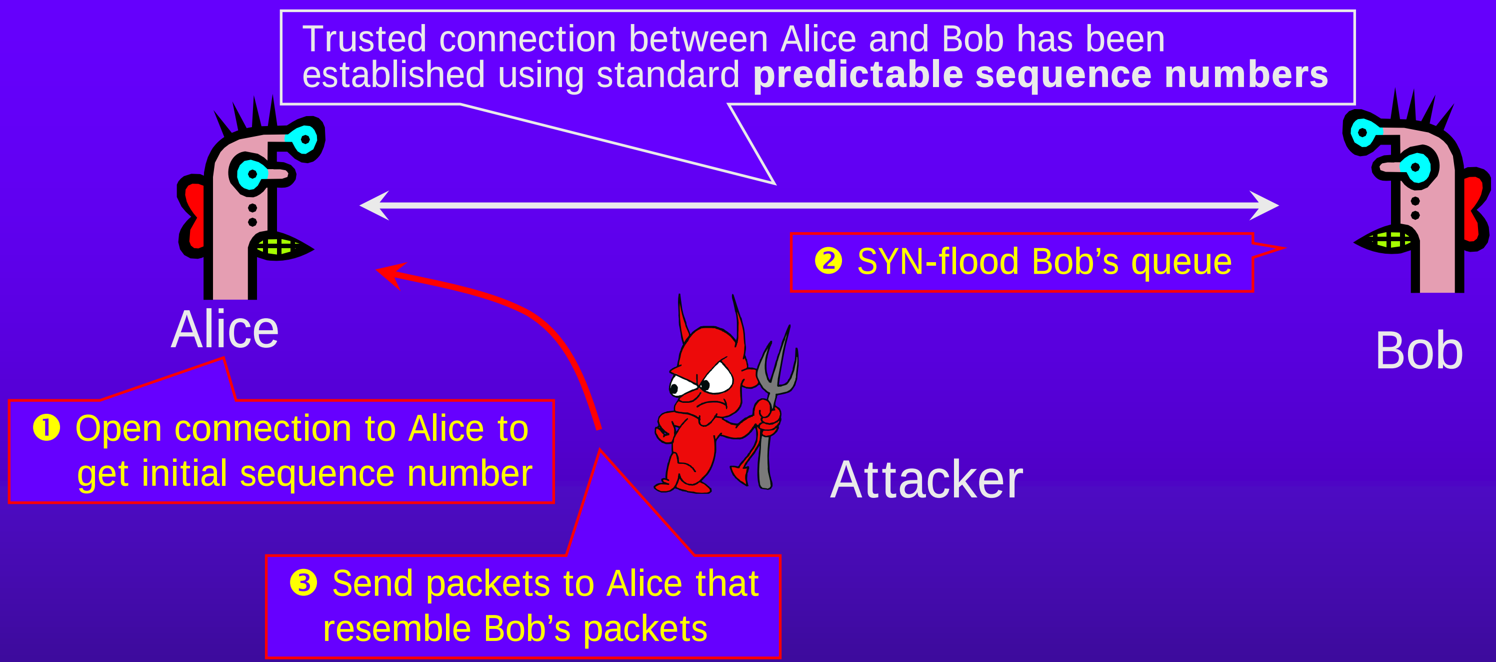
- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public: Smurf attacks
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding



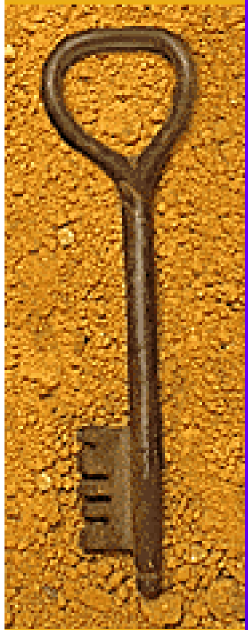
TCP Connection Spoofing

- Each TCP connection has an associated state
  - Port number, Sequence number
- TCP state is easy to guess
  - Port numbers are standard, sequence numbers are often predictable
  - Can inject packets into existing connections
- If attacker knows initial sequence number and amount of traffic, can guess likely current number
  - Send a flood of packets with likely sequence numbers

“Blind” IP Spoofing Attack

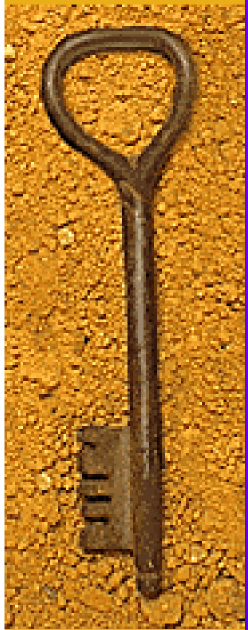


- In order to insert into the communication between Alice and Bob, Attacker can use/forge Bob’s identity if Alice uses IP address-based authentication
  - For example, rlogin and many other remote access programs uses address-based authentication



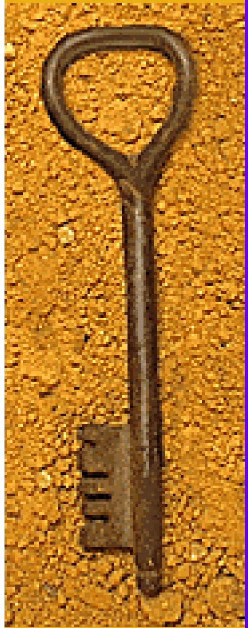
TCP Sequence Numbers

- Need high degree of unpredictability
  - If attacker knows initial seq # and amount of traffic sent, can estimate likely current values
  - Send a flood of packets with likely seq numbers
    - larger bandwidth => larger flood possible
- Reported to be safe from practical attacks
  - Cisco IOS, OpenBSD 2.8-current, FreeBSD 4.3-RELEASE, AIX, HP/UX 11i, Linux Kernels after 1996
  - Solaris 2.6 if strong seq numbers turned on:
    - Set TCP_STRONG_ISS to 2 in /etc/default/inetinit.
  - HP/UX , IRIX 6.5.3, ... if so configured



DoS by Connection Reset

- If attacker can guess current sequence number for an existing connection, can send Reset packet to close it
  - With 32-bit sequence numbers, probability of guessing correctly is $1/2^{32}$ (not practical)
  - Most systems accept large windows of sequence numbers $\Rightarrow$ much higher probability of success
    - Need large windows to handle massive packet losses
- Especially effective against long-lived connections
  - For example, BGP route updates



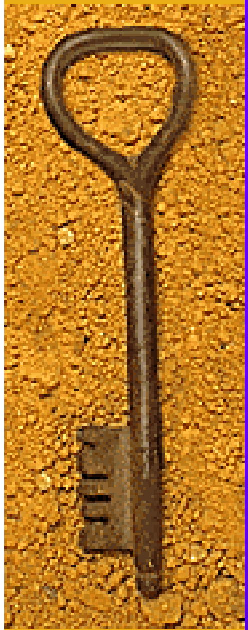
Cryptographic protection

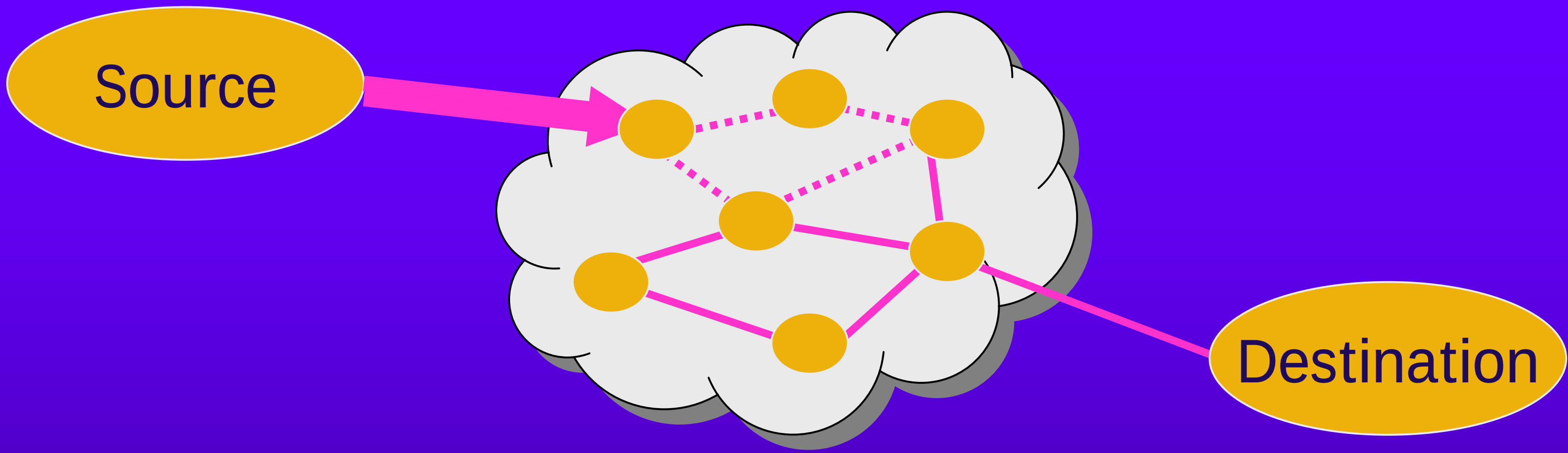
- Solutions above the transport layer
  - Examples: SSL and SSH
  - Protect against session hijacking and injected data
  - Do not protect against denial-of-service attacks caused by spoofed packets
- Solutions at network layer
  - IPSec
  - Can protect against
    - session hijacking and injection of data
    - denial-of-service attacks using session resets

Security Issues in TCP/IP

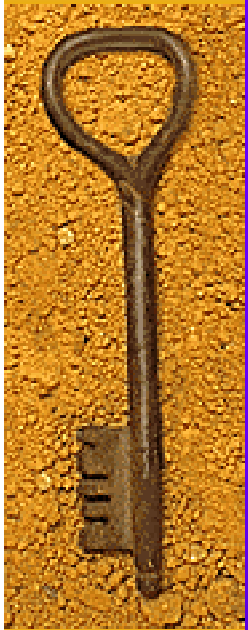
- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public: Smurf attacks
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding

TCP Congestion Control

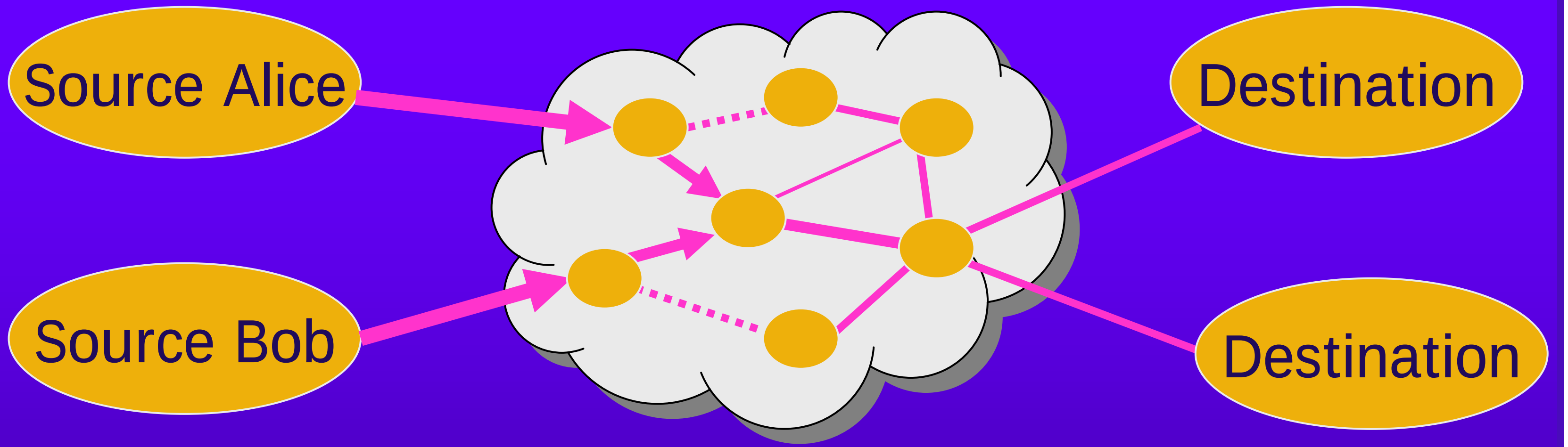




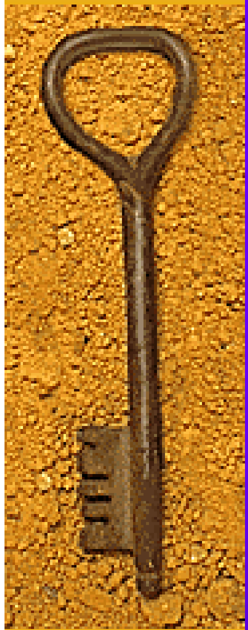
- If packets are lost, assume congestion
  - – Reduce transmission rate by half, repeat
  - – If loss stops, increase rate very slowly



Competition



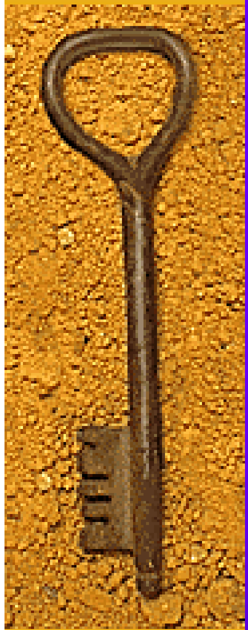
- Amiable Alice yields to malicious Bob
  - Alice and Bob both experience packet loss
  - Alice backs off
  - Bob disobeys protocol, gets better results



TCP Attack on Congestion Control

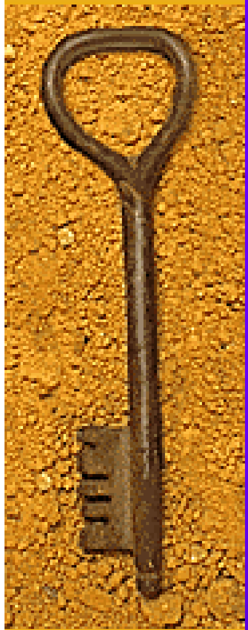
- Misbehaving receiver can trick sender into ignoring congestion control
  - Receiver: duplicate ACK indicates gap
    - Packets within seq number range assumed lost
    - Sender executes fast retransmit algorithm
  - Malicious receiver can
    - Send duplicate ACK
    - ACK before data is received
      - needs some application level retransmission – e.g. HTTP 1.1 range requests … See RFC 2581
  - Solutions
    - Add nonces – ACKs return nonce to prove reception

See: Savage et al., TCP Congestion Control with a Misbehaving Receiver



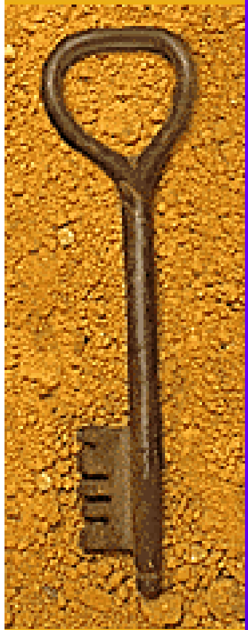
Security Issues in TCP/IP

- Network packets pass by untrusted hosts
  - Eavesdropping (packet sniffing)
- Bug in ICMP implementation: Ping of Death
- ARP info is public: ARP Poisoning
- IP addresses are public: Smurf attacks
- TCP connection requires state: SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
- TCP Congestion Control
  - Trick sender forget congestion control
- UDP data flooding



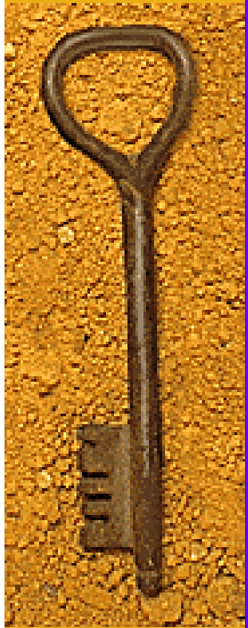
User Datagram Protocol (UDP)

- UDP is a connectionless protocol
  - Simply send datagram to application process at the specified port of the IP address
  - Source port number provides return address
  - Applications: media streaming, broadcast
- No acknowledgement, no flow control, no message continuation
- Denial of service by UDP data flood



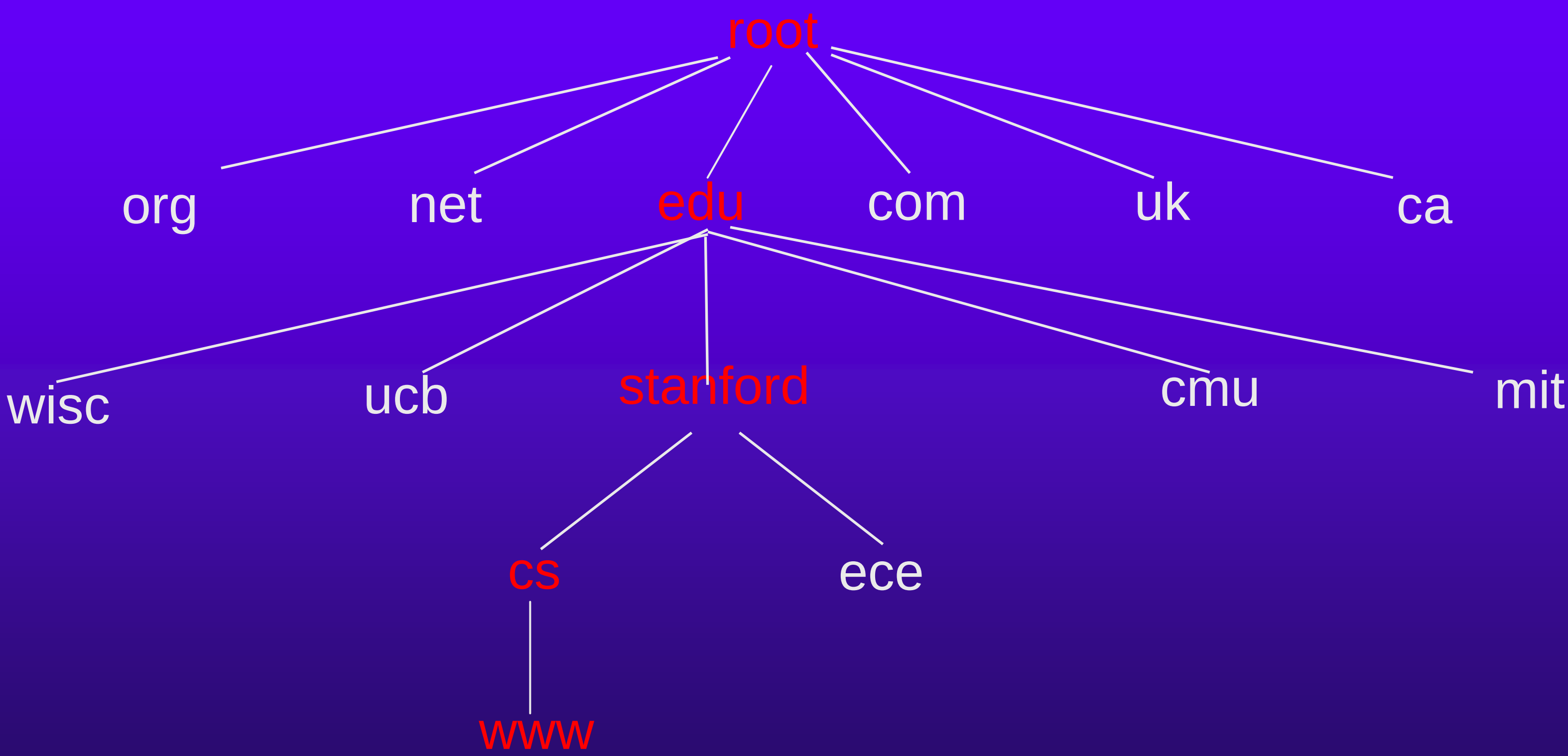
Agenda

- Brief Introduction to TCP/IP network
- Security Issues in TCP/IP
- DNS Security
- Router Security



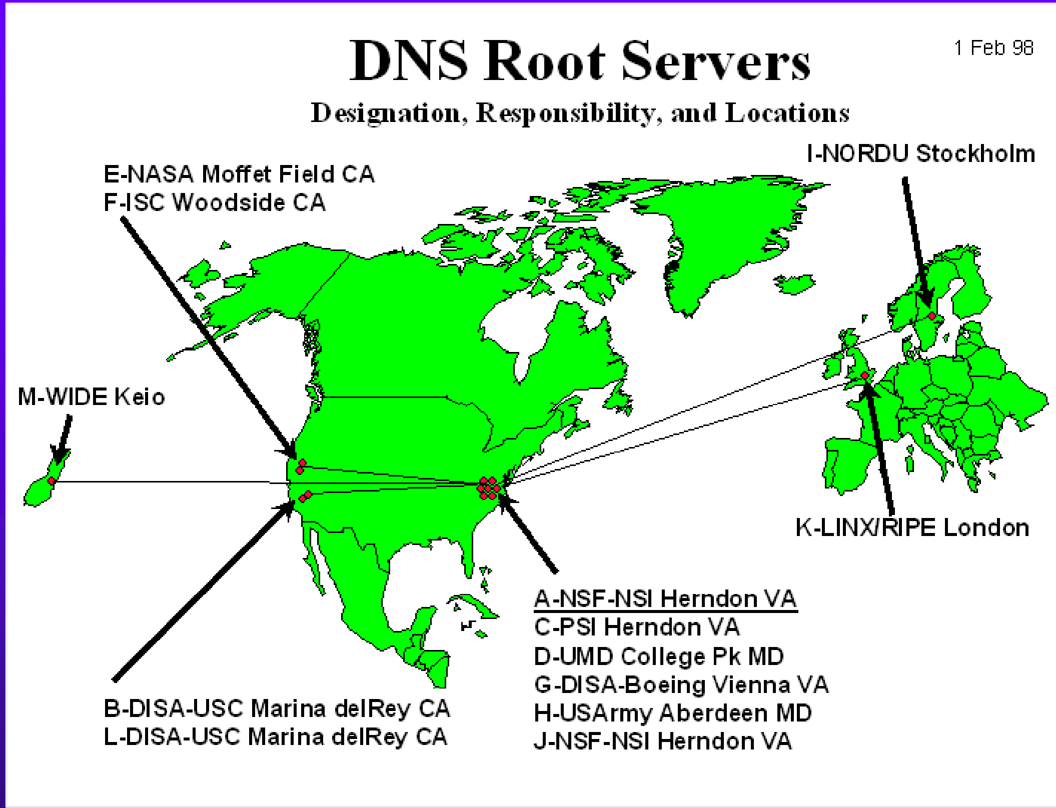
Domain Name System

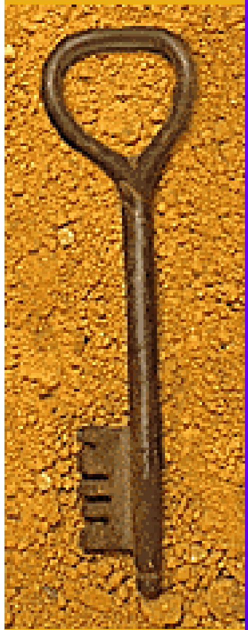
- Hierarchical Name Space



DNS Root Name Servers

- Root name servers
- Local name servers contact root servers when they cannot resolve a name

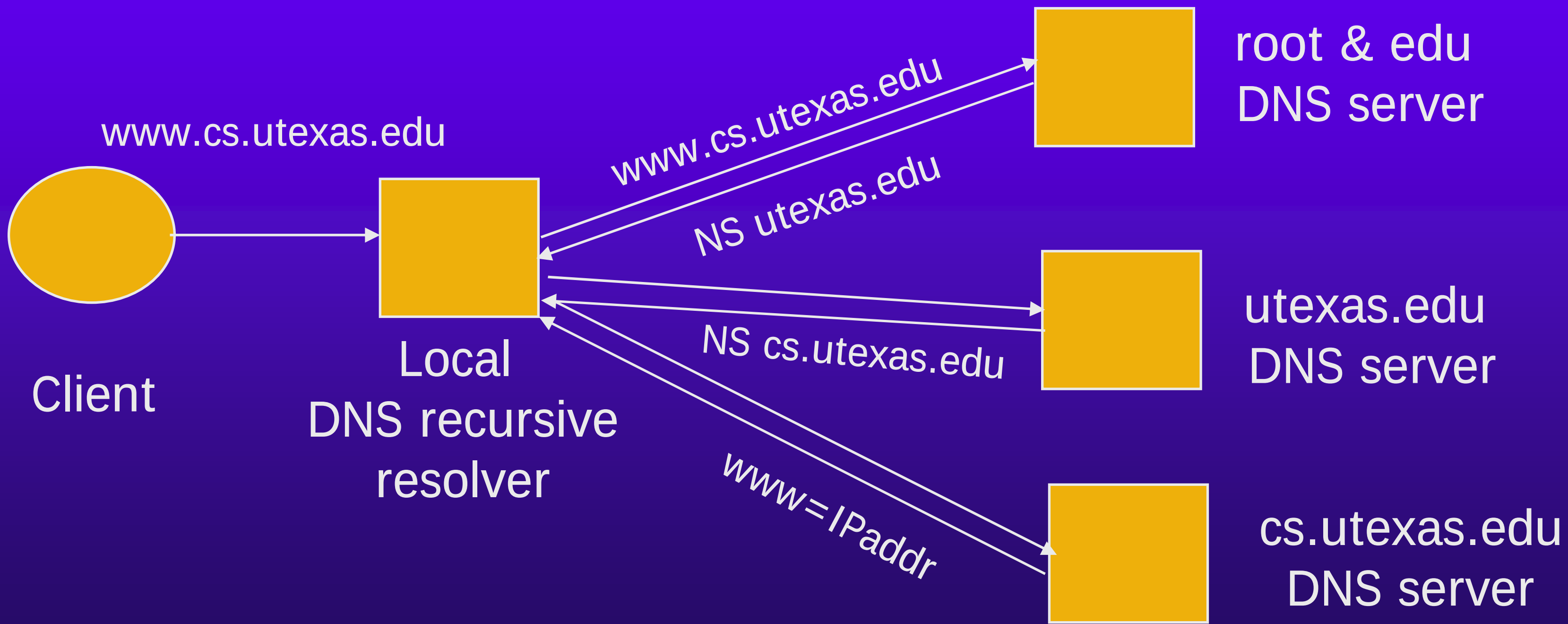


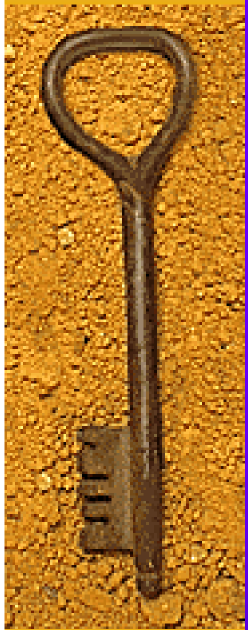


DNS: Domain Name Service

DNS maps symbolic names to numeric IP addresses

(for example, www.cs.utexas.edu ↔ 128.83.120.155)

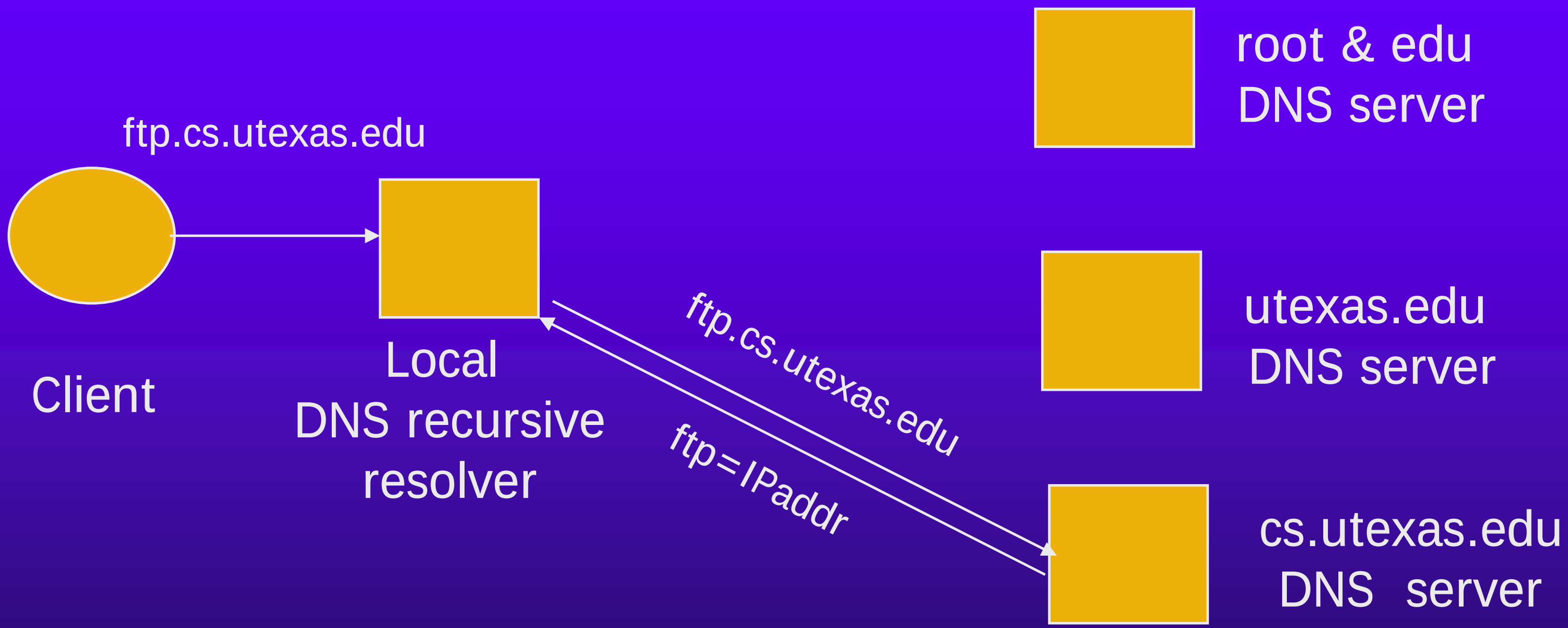


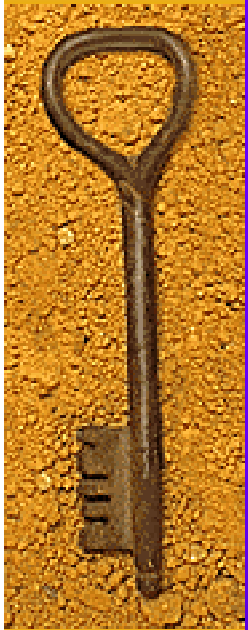


DNS Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Other queries may reuse some parts of lookup
    - NS records for domains
- DNS negative queries are cached
  - Don’t have to repeat past mistakes
    - For example, misspellings
- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record

Cached Lookup Example



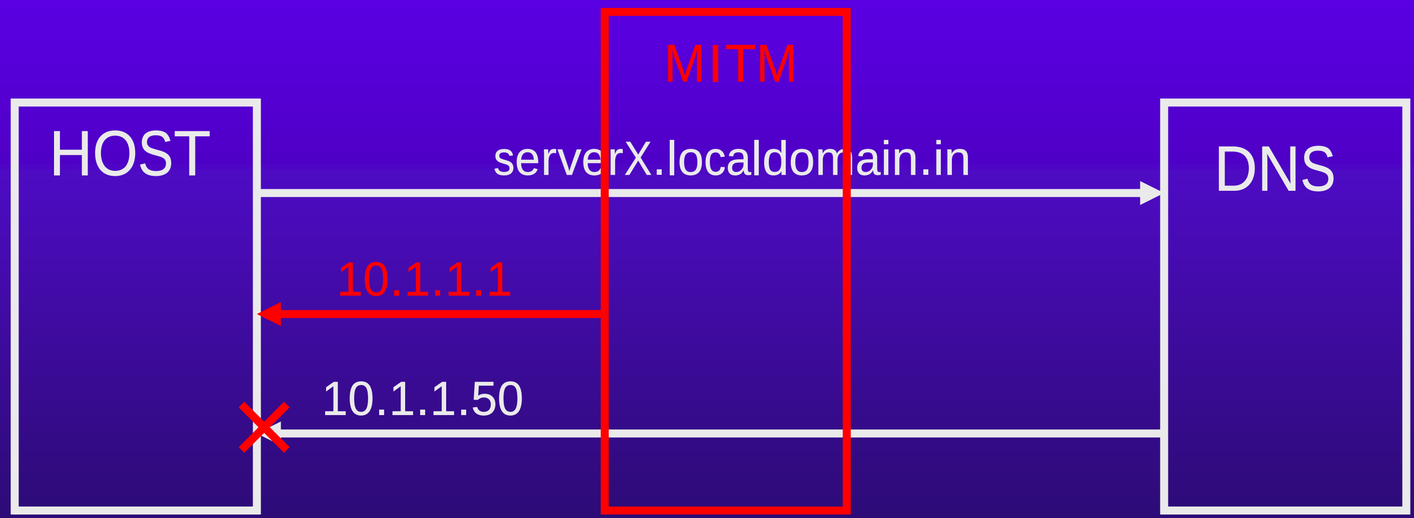


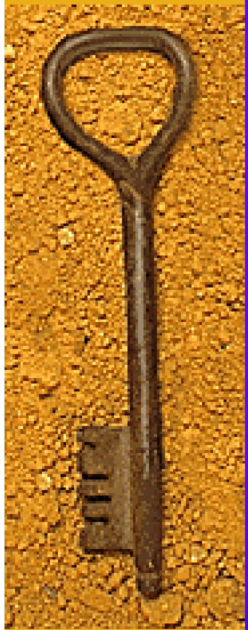
DNS Vulnerabilities

- DNS implementations have vulnerabilities
  - Reverse query buffer overrun in old releases of BIND
    - Gain root access, abort DNS service…
  - MS DNS for NT 4.0 crashes on chargen stream
    - telnet ntbox 19 | telnet ntbox 53
- Denial of service is a risk
  - Oct '02: ICMP flood took out 9 root servers for 1 hour
- Can use "zone transfer" requests to download DNS database and map out the network
- DNS host-address mappings are <u>not</u> authenticated (see next slides DNS-Spoofing)

DNS Spoofing

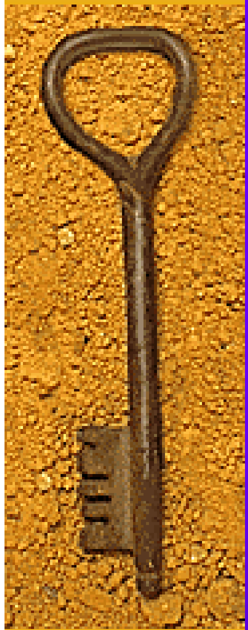
If the attacker is able to sniff the ID of the DNS request, he/she can reply before the real DNS server





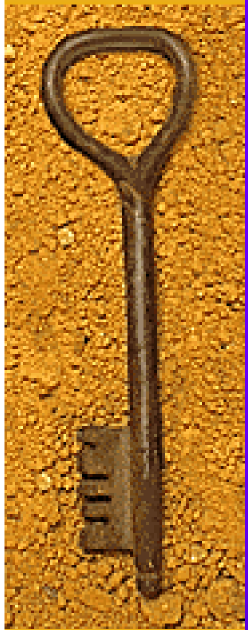
DNS Spoofing: Tools

- ettercap (http://ettercap.sf.net)
  - Phantom plugin
- dsniff (http://www.monkey.org/~dugsong/dsniff)
  - Dnsspoof
- zodiac (http://www.packetfactory.com/Projects/zodiac)



DNS Spoofing: Countermeasures

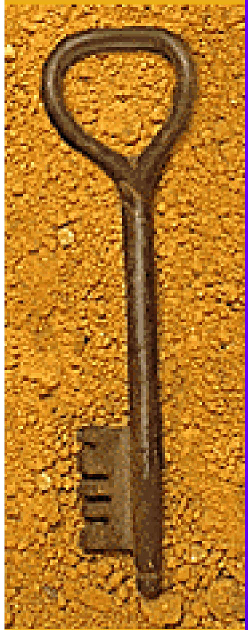
- YES - detect multiple replies (IDS)
- YES - use lmhost or host file for static resolution of critical hosts
- YES - DNSSEC



Defenses Against DNS Spoofing

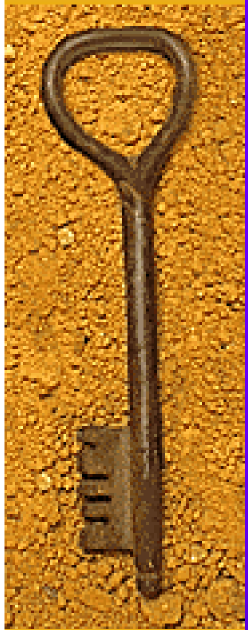
- Double-check reverse DNS
  - Modify rlogind, rshd to query DNS server and check if symbolic address maps to numeric address
  - Cache poisoning still an issue
- Authenticate entries in DNS tables
  - Hard to do; need public-key infrastructure

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



DNS Poisoning

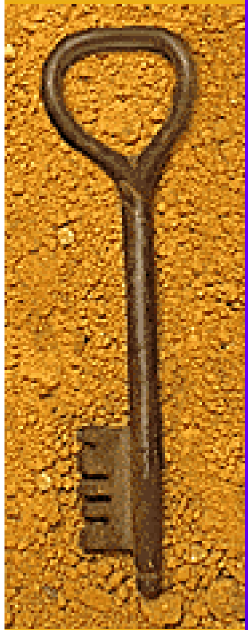
- Type 1 attack
  - The attacker sends a request to the victim DNS asking for one host
  - The attacker spoofs the reply which is expected to come from the real DNS
  - The spoofed reply must contain the correct ID (brute force or semi-blind guessing)



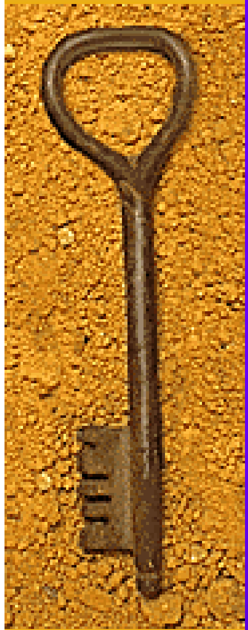
DNS Poisoning

- Type 2 attack
  - The attacker can send a “dynamic update” to the victim DNS
  - If the DNS processes it, it is even worst because it will be authoritative for those entries

DNS Poisoning: Tools

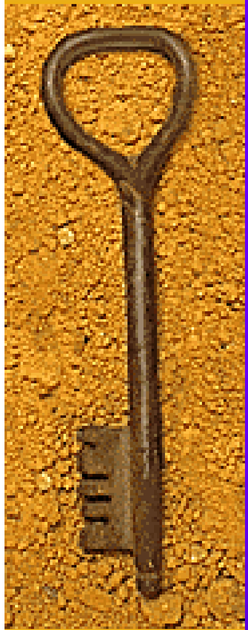


- ADMIdPack
  - http://packetstormsecurity.org/groups/ADM/ADMIDpack/
- Zodiac
  - http://www.packetfactory.com/Projects/zodiac



DNS Poisoning: Countermeasures

- YES - Use DNS with random transaction ID (Bind v9)
- YES - DNSSec (Bind v9) allows the digital signature of the replies.
- NO - restrict the dynamic update to a range of IPs (they can be spoofed)



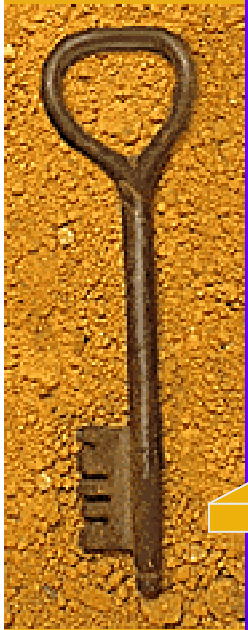
Other DNS Risks

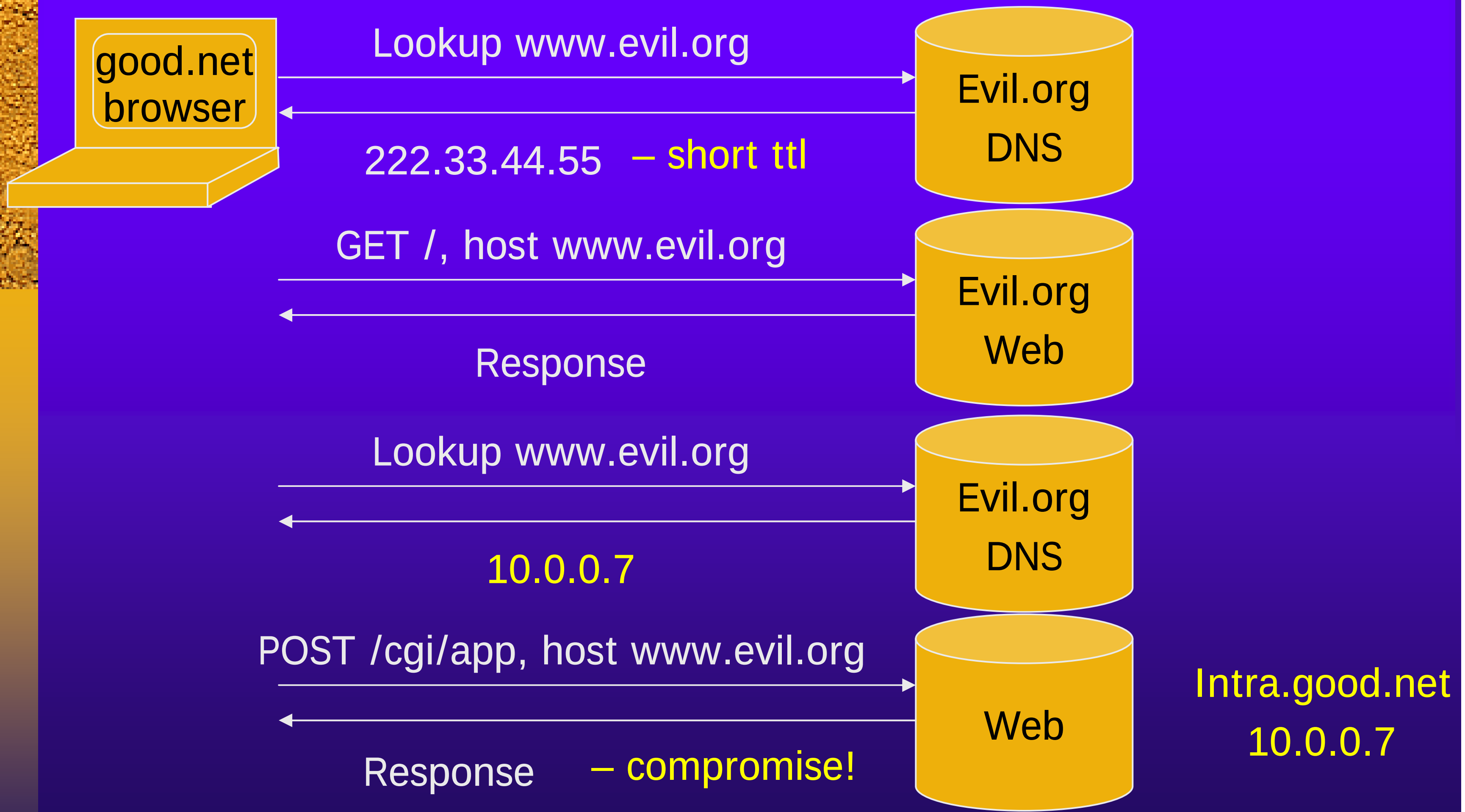
- DNS cache poisoning
  - False IP with a high time-to-live will stay in the cache of the DNS server for a long time
  - Basis of pharming
- Spoofed ICANN registration and domain hijacking
  - Authentication of domain transfers based on email addr
  - Aug ’04: teenager hijacks eBay’s German site
  - 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 has been redirected to yet another company in Canada."
- Misconfiguration and human error

JavaScript/DNS Intranet attack (I)

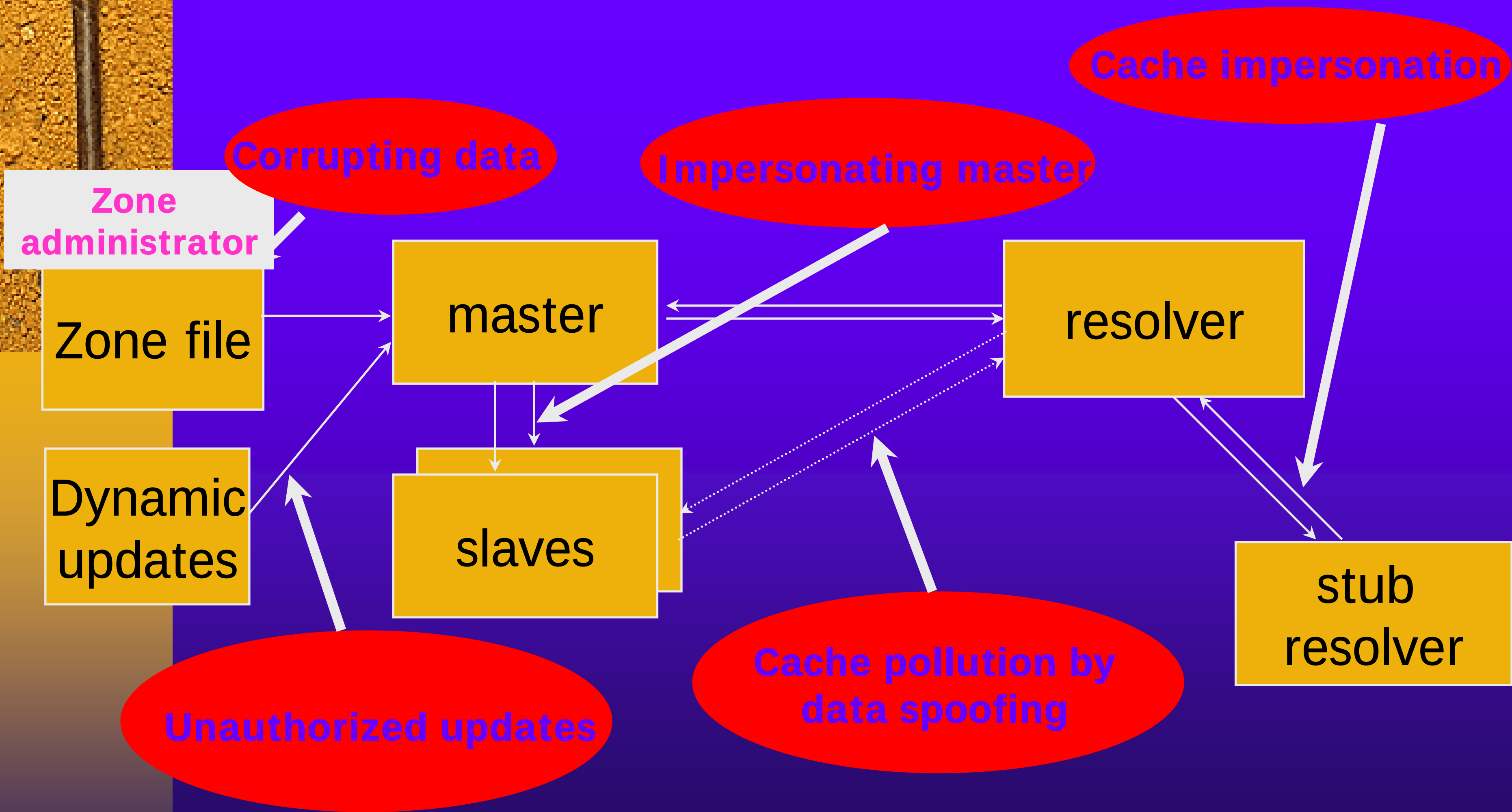
- Consider a Web server intra.good.net
  - IP: 10.0.0.7, inaccessible outside good.net network
  - Hosts sensitive CGI applications
- Attacker at evil.org gets good.net user to browse www.evil.org
- Places Javascript on www.evil.org that accesses sensitive application on intra.good.net
  - This doesn't work because Javascript is subject to "same-origin" policy
  - ... but the attacker controls evil.org DNS

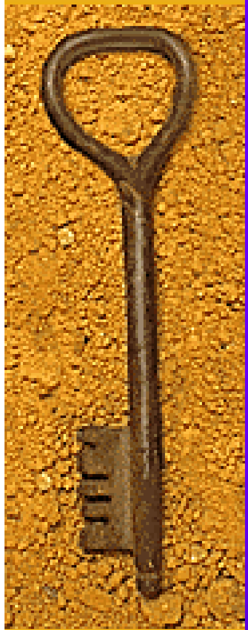
JavaScript/DNS Intranet attack (II)





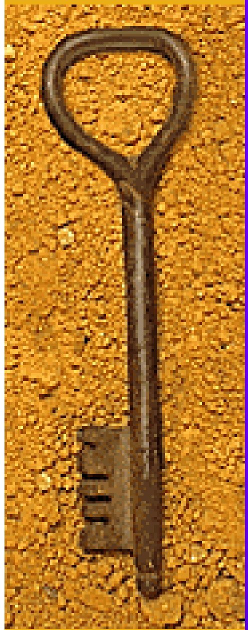
DNS Vulnerabilities: Summary





DNSSEC

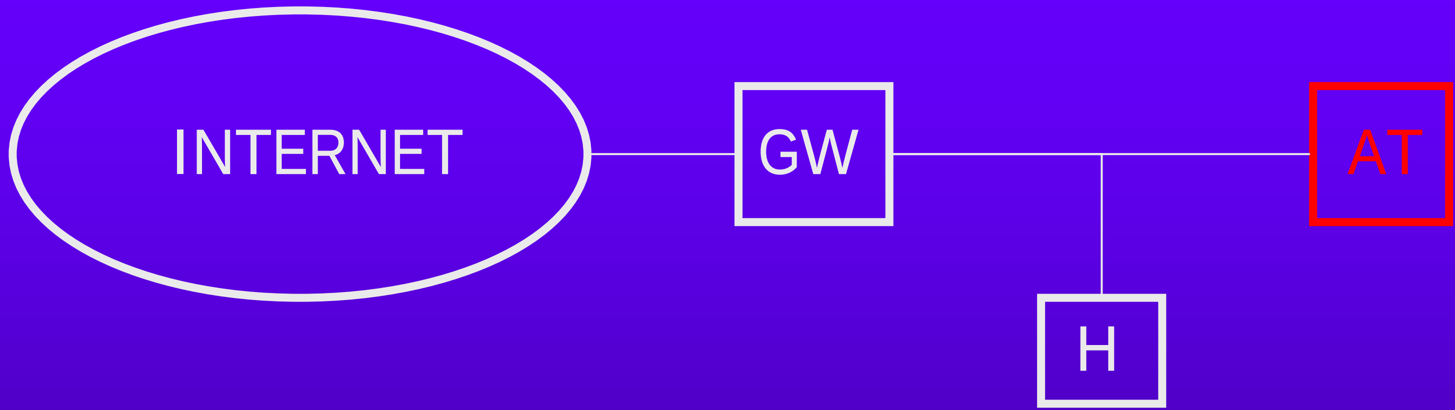
- Goals: authentication and integrity of DNS requests and responses
- PK-DNSSEC (public key)
  - DNS server signs its data (can be done in advance)
  - How do other servers learn the public key?
- SK-DNSSEC (symmetric key)
  - Encryption and MAC: $E_k(m, MAC(m))$
  - Each message contains a nonce to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



Agenda

- Brief Introduction to TCP/IP network
- Security Issues in TCP/IP
- DNS Security
- Router Security

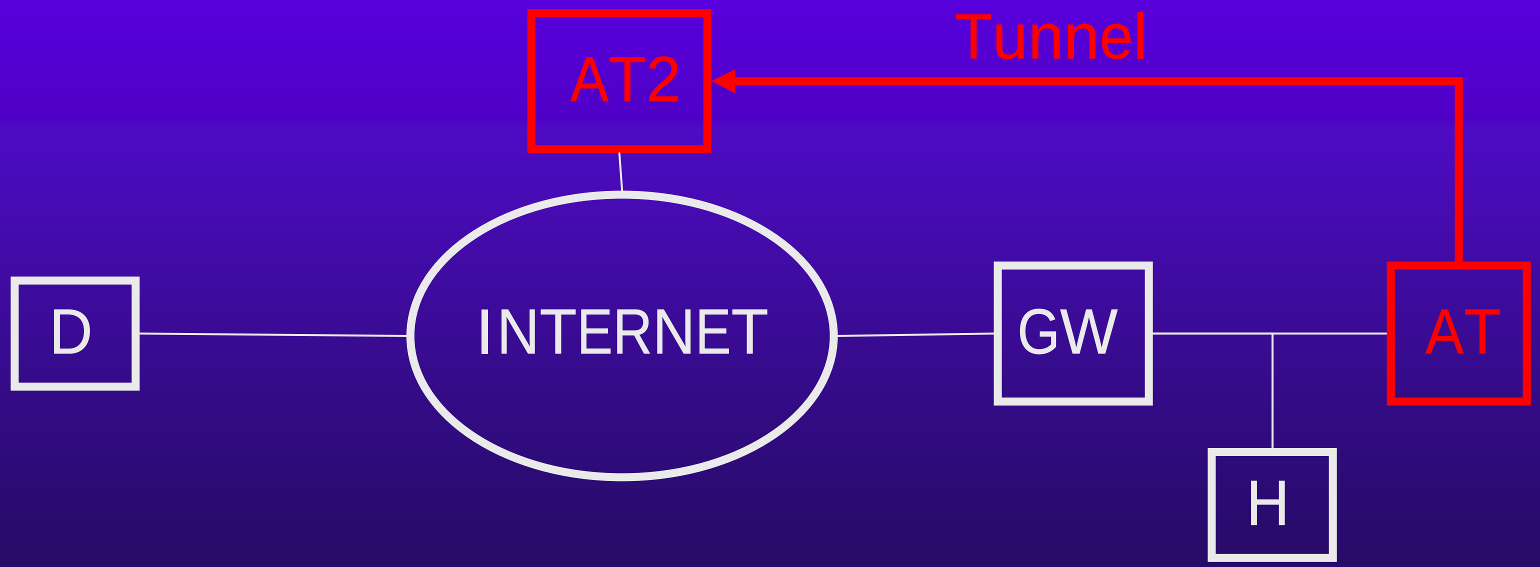
ROUTE mangling

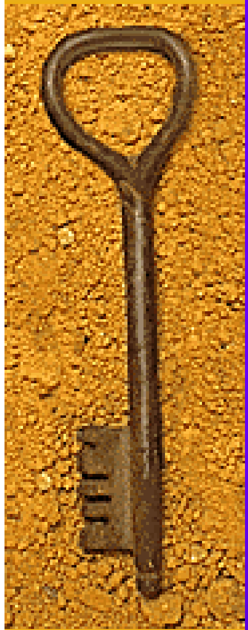


The attacker can forge packets for the gateway (GW) pretending to be a router with a good metric for a specified host on the internet

ROUTE mangling

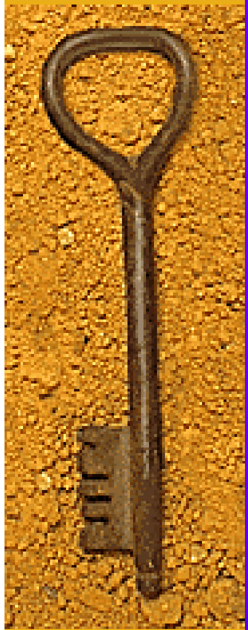
- Now the problem for the attacker is to send packets to the real destination. He/she cannot send it through GW since it is convinced that the best route is AT.





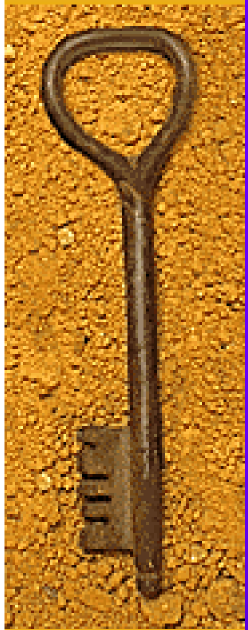
ROUTE mangling: Tools

- IRPAS (Phenoelit) (http://www.phenoelit.de/irpas/)
- Nemesis (http://www.packetfactory.net/Projects/nemesis/)



ROUTE mangling: Countermeasures

- YES - Disable dynamic routing protocols in this type of scenario
- YES - Enable ACLs to block unexpected update
- YES - Enable authentication on the protocols that support authentication



Reading Assignment

- "SYN cookies" by Bernstein
- "IP Spoofing Demystified" from Phrack magazine
- Joncheray's paper about TCP connection hijacking