CSCE 465 Computer & Network Security

Instructor: Abner Mendoza

Vulnerability Analysis

Roadmap

- Why vulnerability analysis?
- Example: TCP/IP related vulnerabilities
  - IP spoofing
  - TCP attacks
    - SYN flooding attack
    - TCP RST attack
    - TCP Session Hijacking
  - ARP cache poisoning
  - ICMP attacks
- Summary

Recap: The Security Life-Cycle

- ***Threats***
- Policy
- Specification
- Design
- Implementation
- Operation and Maintenance

Threat, Vulnerability, and Attack

- A threat is a *potential* violation of security
  - Flaws in design, implementation, and operation
  - "Vulnerability"!
- An attack is any *action* that violates security
  - Active vs. passive attacks

Vulnerability Definition

- *Vulnerability*, *security flaw*: failure of security policies, procedures, and controls that allow a subject to commit an action that violates the security policy
  - Subject is called an *attacker*
  - Using the failure to violate the policy is *exploiting the vulnerability* or *breaking in*

Vulnerability Analysis

- Vulnerability analysis (vulnerability assessment), is a process that defines, identifies, and classifies the security holes (vulnerabilities) in a computer, network, or communications infrastructure
- Vulnerability analysis can forecast the effectiveness of proposed countermeasures and evaluate their actual effectiveness after they are put into use

Typical Steps

- Defining and classifying network or system resources
- Assigning relative levels of importance to the resources
- Identifying potential threats to each resource
- Developing a strategy to deal with the most serious potential problems first
- Defining and implementing ways to minimize the consequences if an attack occurs

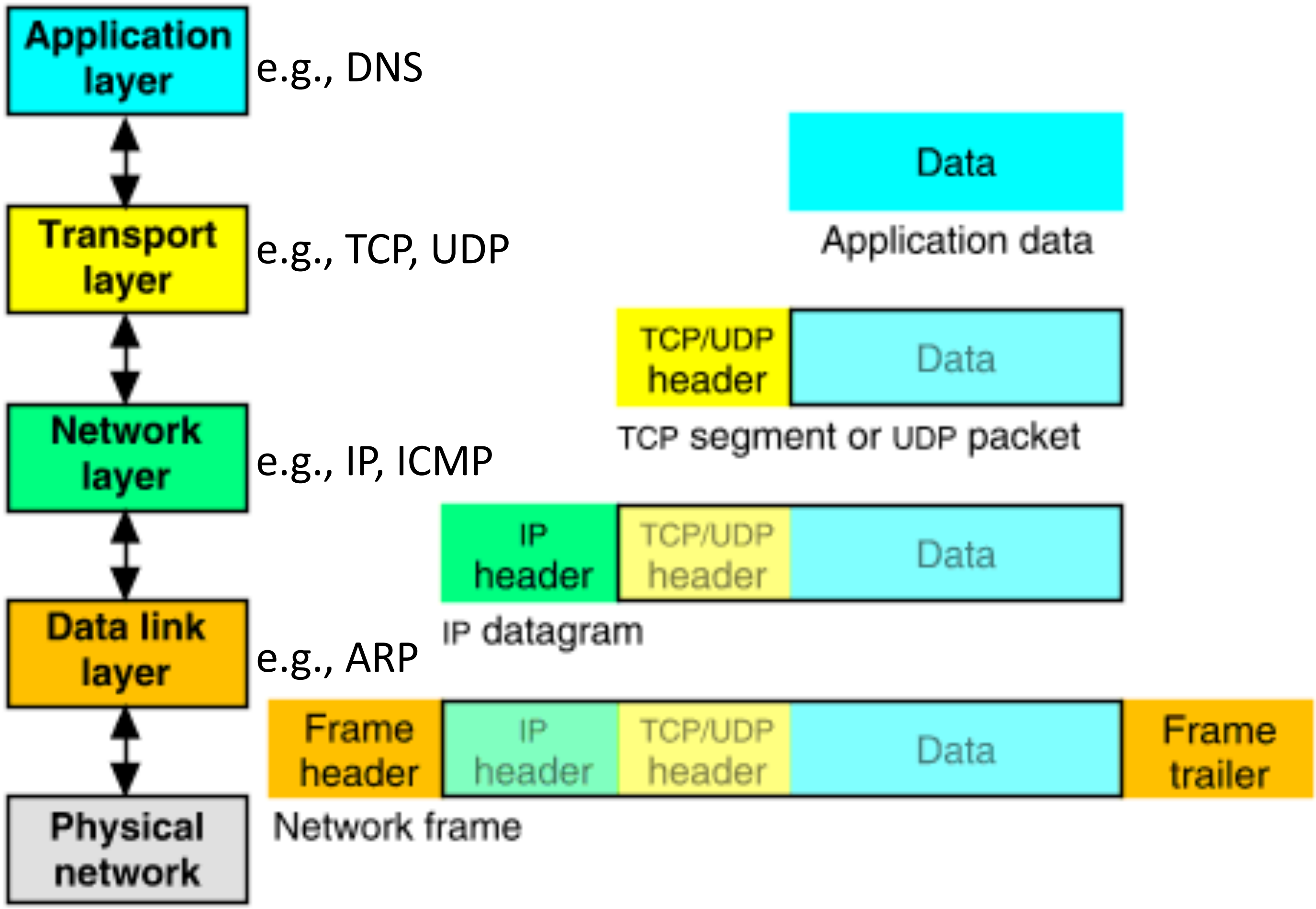
Limited Scope in This Class

- We focus on understanding the generic vulnerabilities inside commonly used TCP/IP protocols

- Homework 4 will be based on this content.

TCP/IP VULNERABILITY EXAMPLES

TCP/IP Stack and Example Protocols



Identify Targets: Port Scanning

- Ports dynamically address ("bind") IP packets to a process
  - Socket data structures keep the mapping information
    - Need to "bind" a socket (port) to a process
- Ports range from 0 to 65535
- Ports 0-1023 are reserved for well-known services
  - Require root (in UNIX) access to listen on those ports
- UDP and TCP ports
  - Usually the same port number is assigned to a service for both UDP and TCP (if the service can use both)
- Tools: NMAP…
  - E.g, run "`nmap -sS 127.0.0.1`"
    - Does a SYN scan
    - More: TCP connect, FIN, Ping, UDP…

Identify Targets: OS Fingerprinting

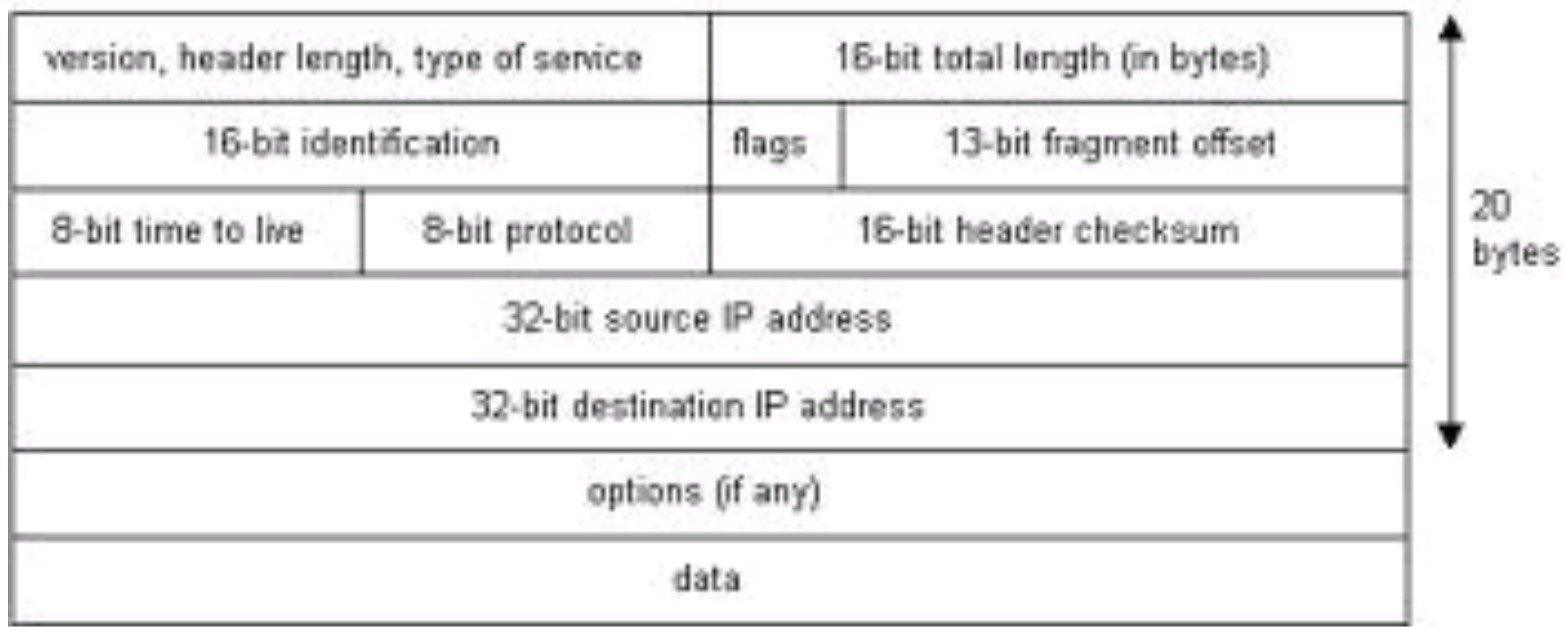
- OS Fingerprinting is a method of detecting the remote host's operating system using information leaked by that host's TCP stack. To do this, we use:
  – the responses it gives to carefully crafted packets (active mode)
    - usually with an invalid/strange set of options (which is where OS vendors usually differ in implementation), and see what happens.
  – or by observing captured network traffic (passive mode).
- These methods are possible because each OS implements their TCP stack differently.
- OS Fingerprinting (ab)uses these differences

Example Methods Used in Nmap

- The FIN probe
  - Send a FIN packet (or any packet without an ACK or SYN flag) to an open port and wait for a response. The correct RFC 793 behavior is to NOT respond, but many broken implementations such as MS Windows, BSDI, CISCO, and IRIX send a RESET back
- IPID sampling
- TCP Initial Window
- TCP Options

EXAMPLE: IP SPOOFING

IP Protocol Header



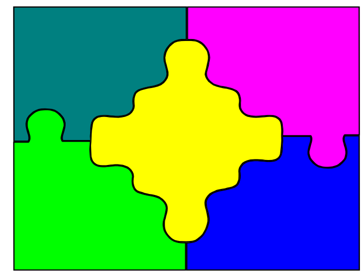
Threat Examples - IP Spoofing

- A common first step to many threats
- Source IP address cannot be trusted!

IP Header	IP Payload
SRC: source DST: destination	
SRC: 18.31.10.8 DST: 128.194.7.237	Is it really from MIT?

Similar to US Mail (or E-mail)

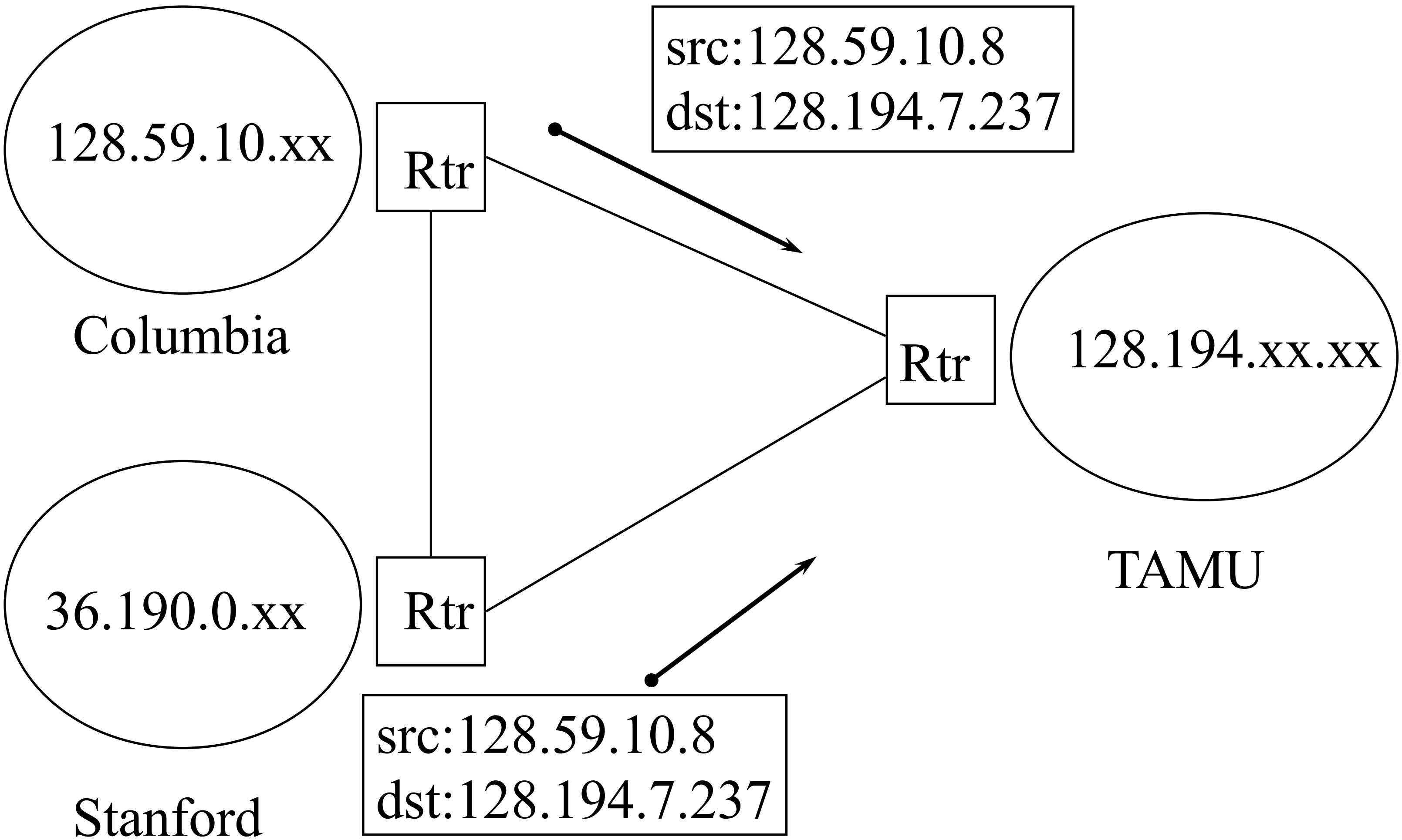
From:
Abner Mendoza
TAMU



To:
William Smith
M.I.B. Corp.

US mail maybe better in the sense that there is a *stamp* put on the envelope at the *location* (e.g., town) of collection...

Most Routers Only Care About Destination Address



Why Should I Care?

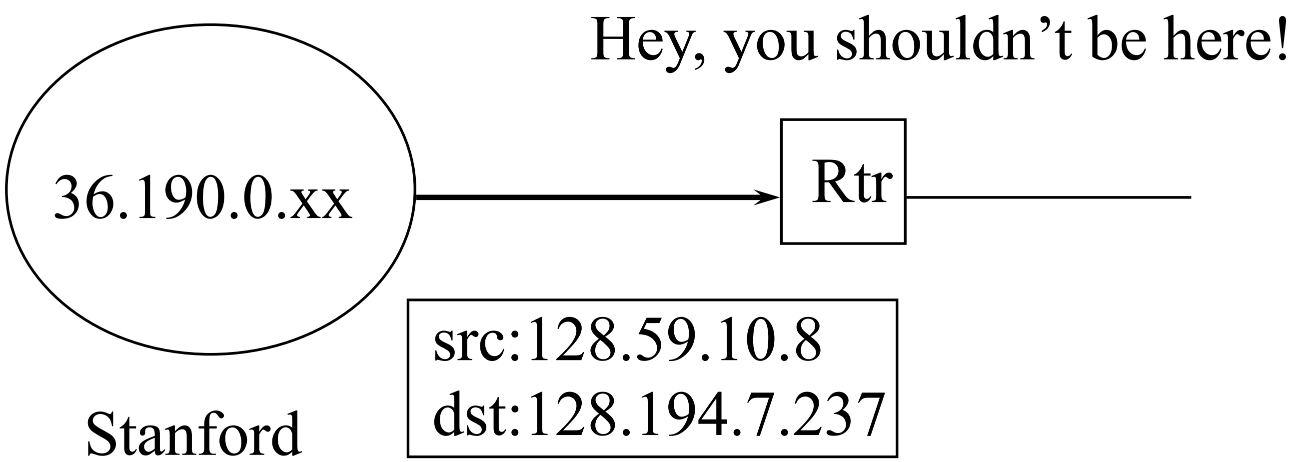
- Attack packets with spoofed IP address help hide the attacking source.
- A *smurf* attack launched with your host IP address could bring your host and network to their knees.
- Higher protocol layers (e.g., TCP) help to protect applications from direct harm, but not enough.

Current IPv4 Infrastructure

- No authentication for the source
- Various approaches exist to address the problem:
  - Router/firewall filtering
  - TCP handshake

Router Filtering

- Decide whether this packet, with certain source IP address, should come from this side of network.



- Local policy

Router Filtering

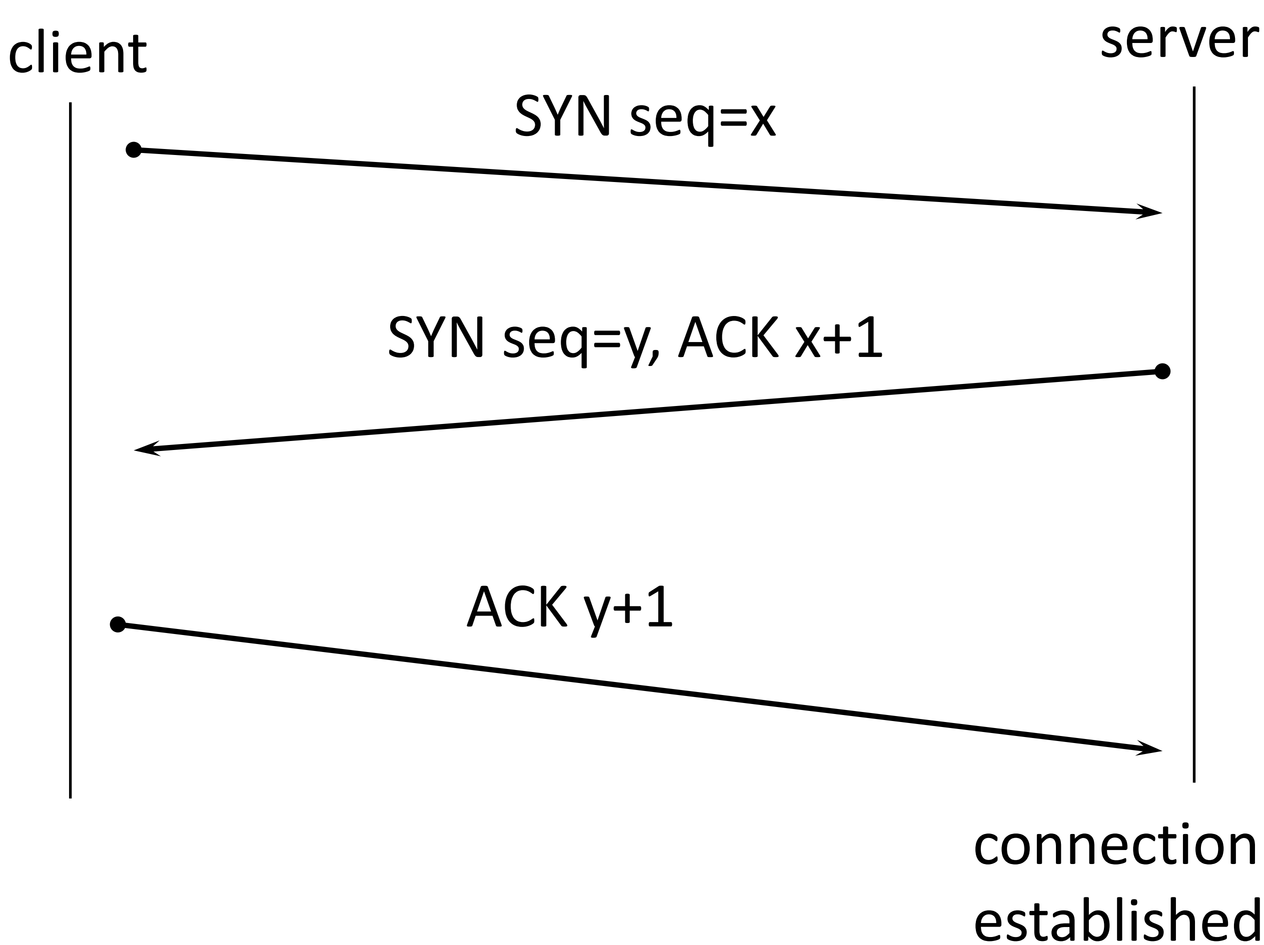
- Very effective for some networks (ISP should always do that!)
  - At least be sure that this packet is from some particular subnet
- Problems:
  - Hard to handle frequent add/delete hosts/subnets or mobile IP
  - Upsets customers should legitimate packets get discarded
  - Need to trust other routers

EXAMPLE: TCP ATTACKS

TCP Protocol Header

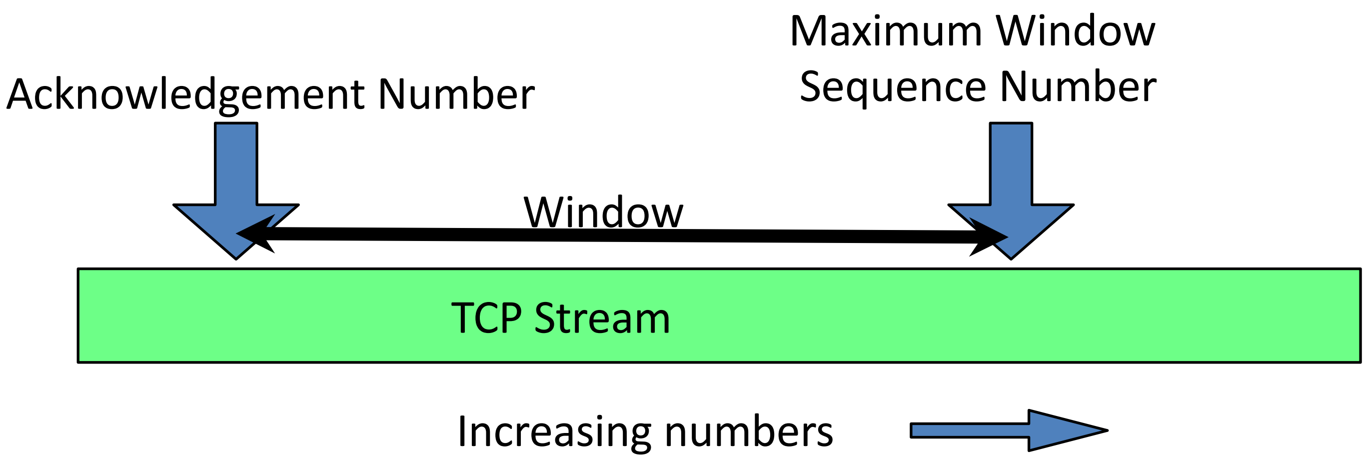
<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">Source Port</td><td colspan="16">Destination Port</td></tr>
<tr><td colspan="32">Sequence Number</td></tr>
<tr><td colspan="32">Acknowledgement Number</td></tr>
<tr><td colspan="4">Data Offset</td><td>-</td><td>-</td><td>-</td><td>-</td><td>CWR</td><td>ECNE</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="16">Window</td></tr>
<tr><td colspan="16">Checksum</td><td colspan="16">Urgent Pointer</td></tr>
<tr><td colspan="32">Options (0 to 10 Words of 32 Bits)</td></tr>
<tr><td colspan="32">TCP Payload</td></tr>
</table>

TCP Handshake



TCP Flow Control

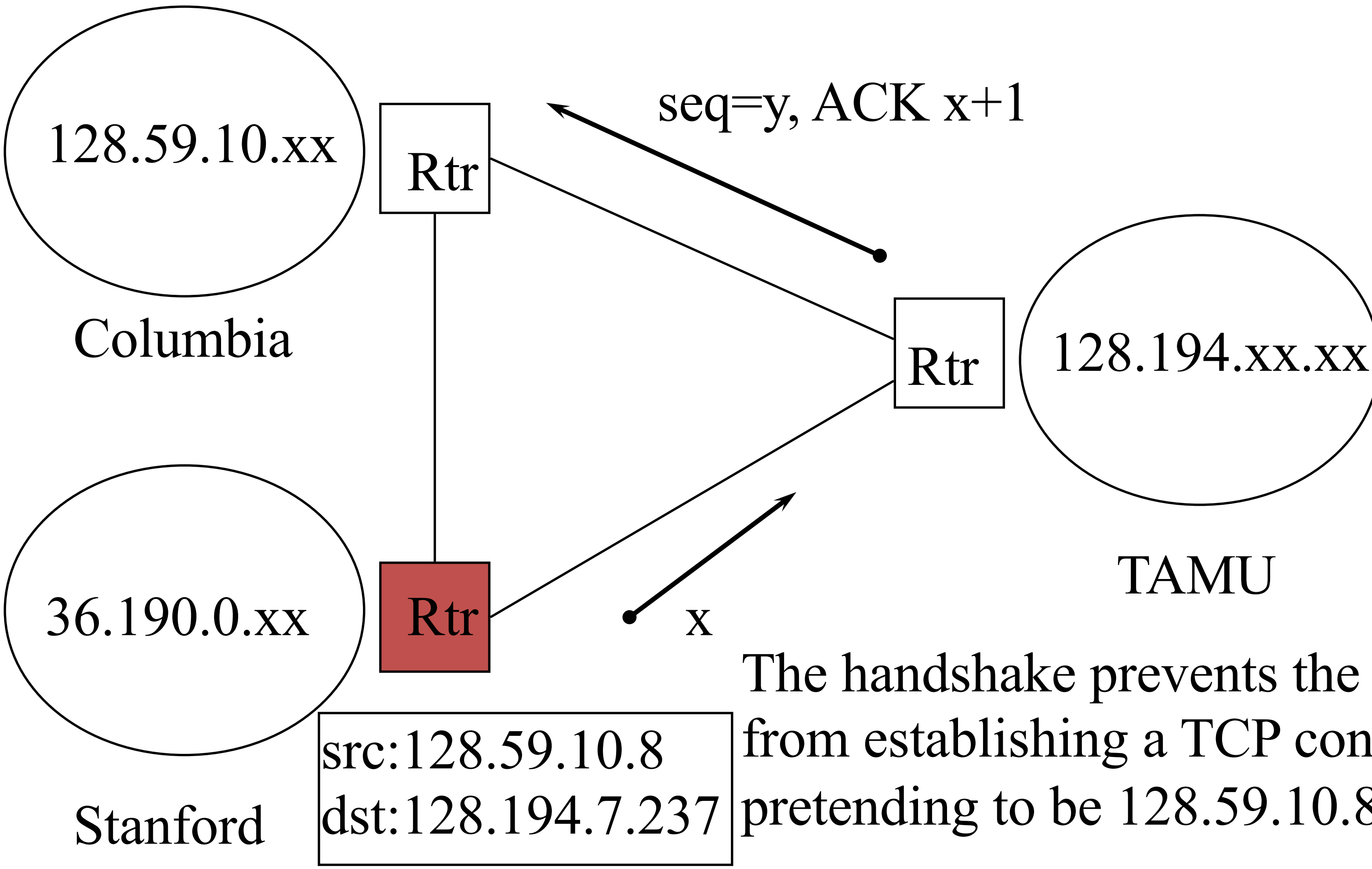
- How much can a sender send at a time?
  - The more can be sent, the more efficient the network is
    - Fewer header bytes, media contention delays, etc...
- TCP "Window"
  - With every ACK, the receiver indicates how many more bytes it is prepared to receive



TCP Sequence Numbers

- Every new connection gets a new initial sequence number (ISN)
  - For both sides of the connection
  - ISNs are exchanged (jargon: streams are "synchronized") in the initial SYN handshake
- TCP packets with sequence numbers outside the window are ignored
  - This makes attacks on TCP applications harder than if they used UDP

TCP Handshake



The handshake prevents the attacker from establishing a TCP connection pretending to be 128.59.10.8

TCP Handshake

- Very effective for stopping most such attacks
- Problems:
  - The attacker can succeed if "y" can be predicted
  - Other DoS attacks are still possible (e.g., TCP SYN-flood)

SYN Flooding Attack

- This exploits how the 3-way handshake of TCP services for opening a session works.
- SYN packets are sent to the target node with incomplete source IP addresses
- The node under attack sends an ACK packet and waits for response
- Since the request has not been processed, it takes up memory
- Many such SYN packets clog the system and take up memory
- Eventually the attacked node is unable to process any requests as it runs out of memory storage space

SYN Flooding Attack

- 90% of DoS attacks use TCP SYN floods
- Streaming spoofed TCP SYNs
- Takes advantage of three way handshake
- Server start “half-open” connections
- These build up… until queue is full and all additional requests are blocked
- Solution?

TCP SYN cookies

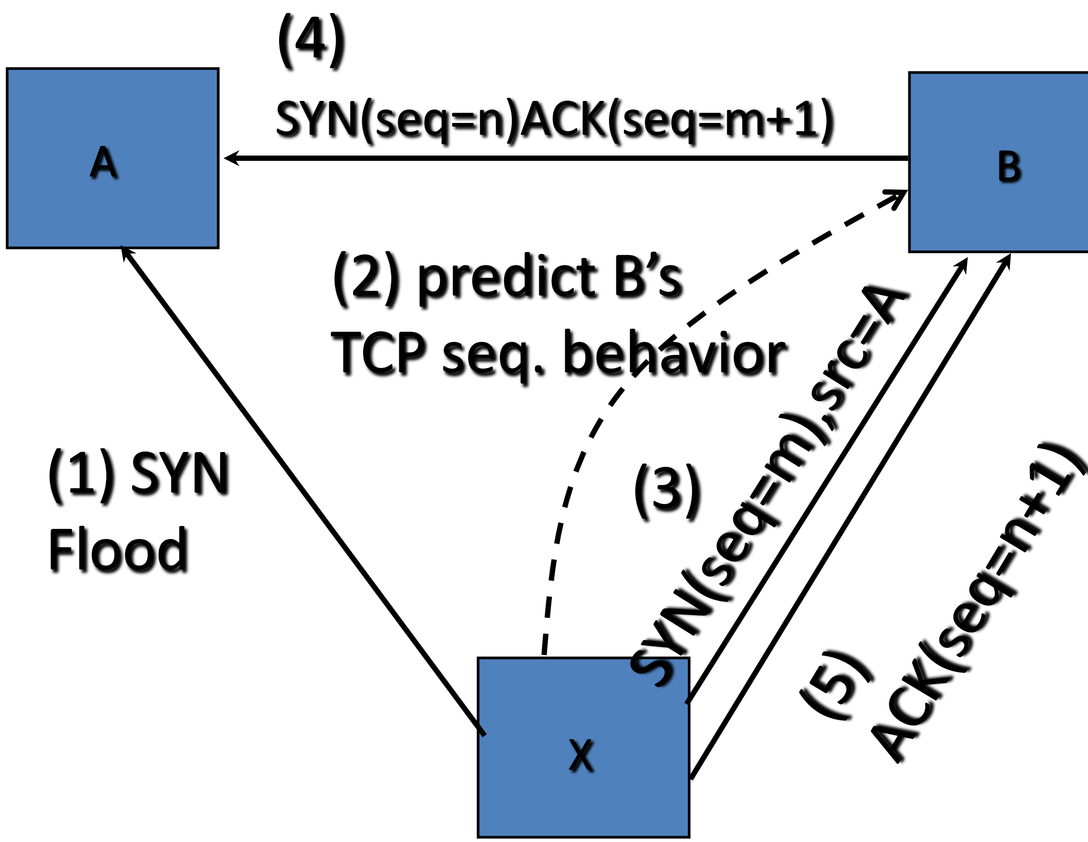
- General idea
  - Client sends SYN w/ ACK number
  - Server responds to Client with SYN-ACK cookie
    - sqn = f(src addr, src port, dest addr, dest port, rand)
    - **Server does not save state**
  - Honest client responds with ACK(sqn+1)
  - Server checks response
  - If matches SYN-ACK, establishes connection

TCP RST Attack

- Send a RST (TCP RESET flag) packet with a spoofed IP address to either side of a valid connection
  - Need to guess a sequence number inside the appropriate window
    - Or sniff traffic to know which number to use
  - The range can be guessed fairly efficiently for RST attacks
  - Sequence numbers: 32 bits
  - Window size: up to 16 bits
  - Number of guesses 32-16 = 16 bit address space
    - 65535 RST attempts, ~ 4 min on DSL connection
    - Faster connection or zombies, faster RST
    - This is the brute force RST attack

IP Spoofing & TCP Session Hijacking

- X establishes a TCP connection with B assuming A's IP address



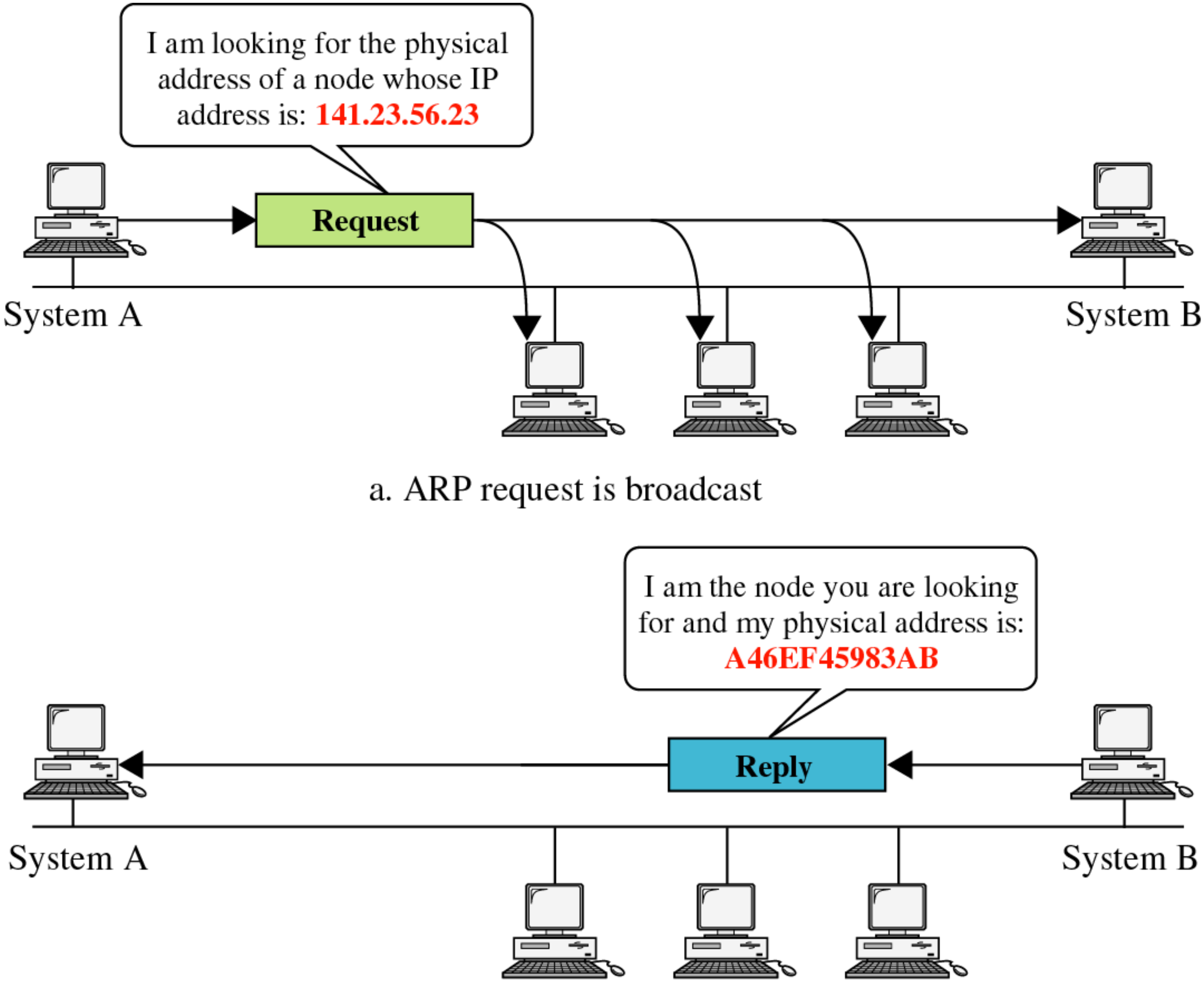
EXAMPLE: ARP POISONING

ARP Poisoning

- ARP = Address Resolution Protocol
- ARP is used by routers extensively to find the destination node. Routers have IP addresses (32-bits). In order to deliver the packet to the destination node, the router broadcasts the IP address of the destination and obtains the MAC address (48-bits).

ARP Protocol

<table>
<tr><td colspan="2">Hardware Type
(16 bits)</td><td colspan="2">Protocol Type
(16 bits)</td></tr>
<tr><td>HA Length
(8 bits)</td><td>PA Length
(8 bits)</td><td colspan="2">Operation
(16 bits)</td></tr>
<tr><td colspan="4">Sender Hardware Address
(Octets 0-3)</td></tr>
<tr><td colspan="2">Sender Hardware Address
(Octets 4-5)</td><td colspan="2">Sender Protocol Address
(Octets 0-1)</td></tr>
<tr><td colspan="2">Sender Protocol Address
(Octets 2-3)</td><td colspan="2">Target Hardware Address
(Octets 0-1)</td></tr>
<tr><td colspan="4">Target Hardware Address
(Octets 2-5)</td></tr>
<tr><td colspan="4">Target Protocol Address
(Octets 0-3)</td></tr>
</table>

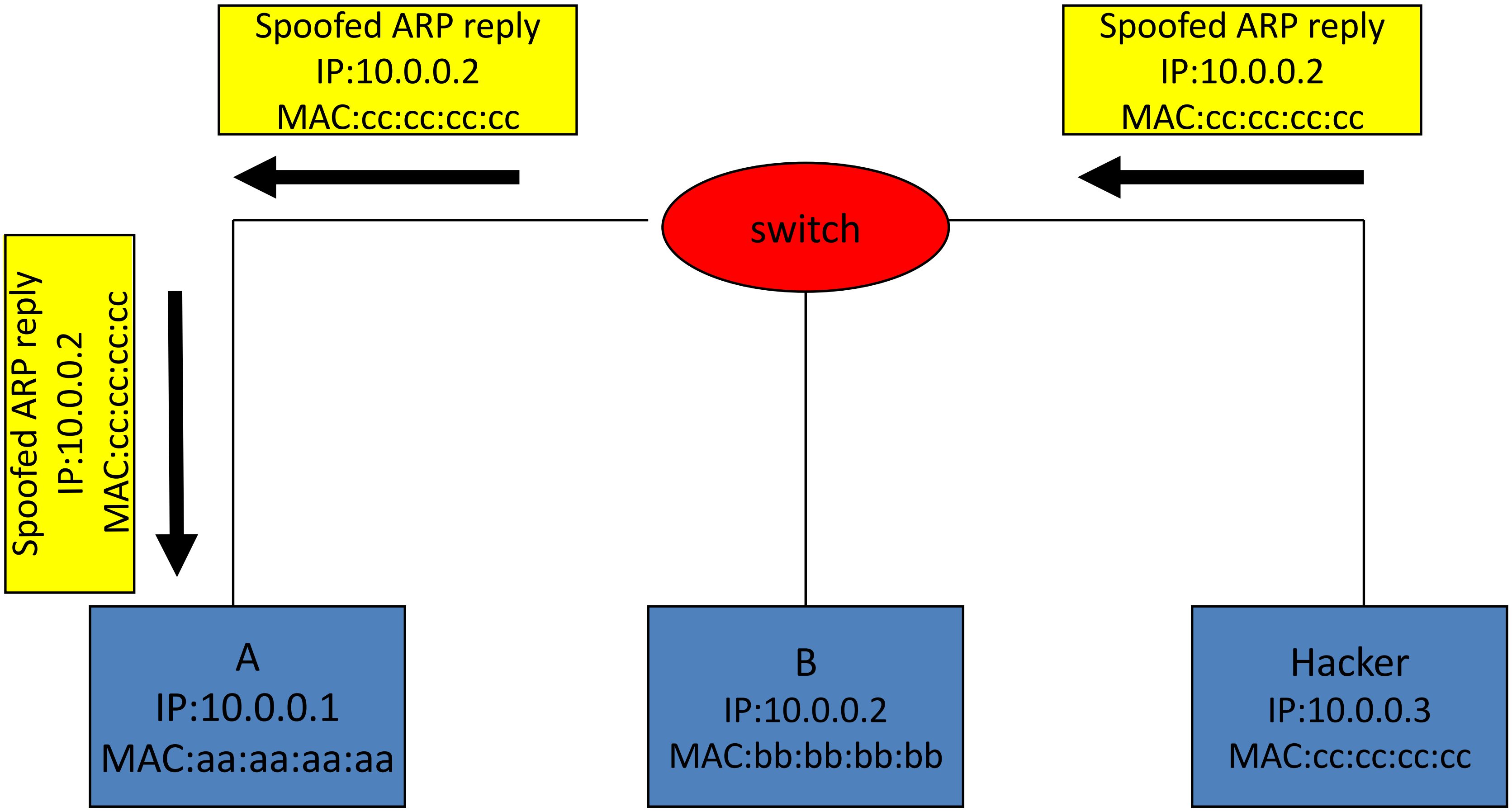


a. ARP request is broadcast

b. ARP reply is unicast

ARP Poisoning

- Hosts store the IP-to-MAC address mapping in the ARP table. ARP Poisoning means that the ARP communication is intercepted by redirection from a router.
- Example:
  - Assume router's IP is 10.1.1.0
  - Host's IP is 10.1.1.1
  - Malicious host with IP 10.1.1.2 spoofs 10.1.1.1 and replies to requests from 10.1.1.0 with its MAC address
  - From this point on all packets meant for 10.1.1.1 is routed to 10.1.1.2 because the router has the MAC address of 10.1.1.2 in its routing table

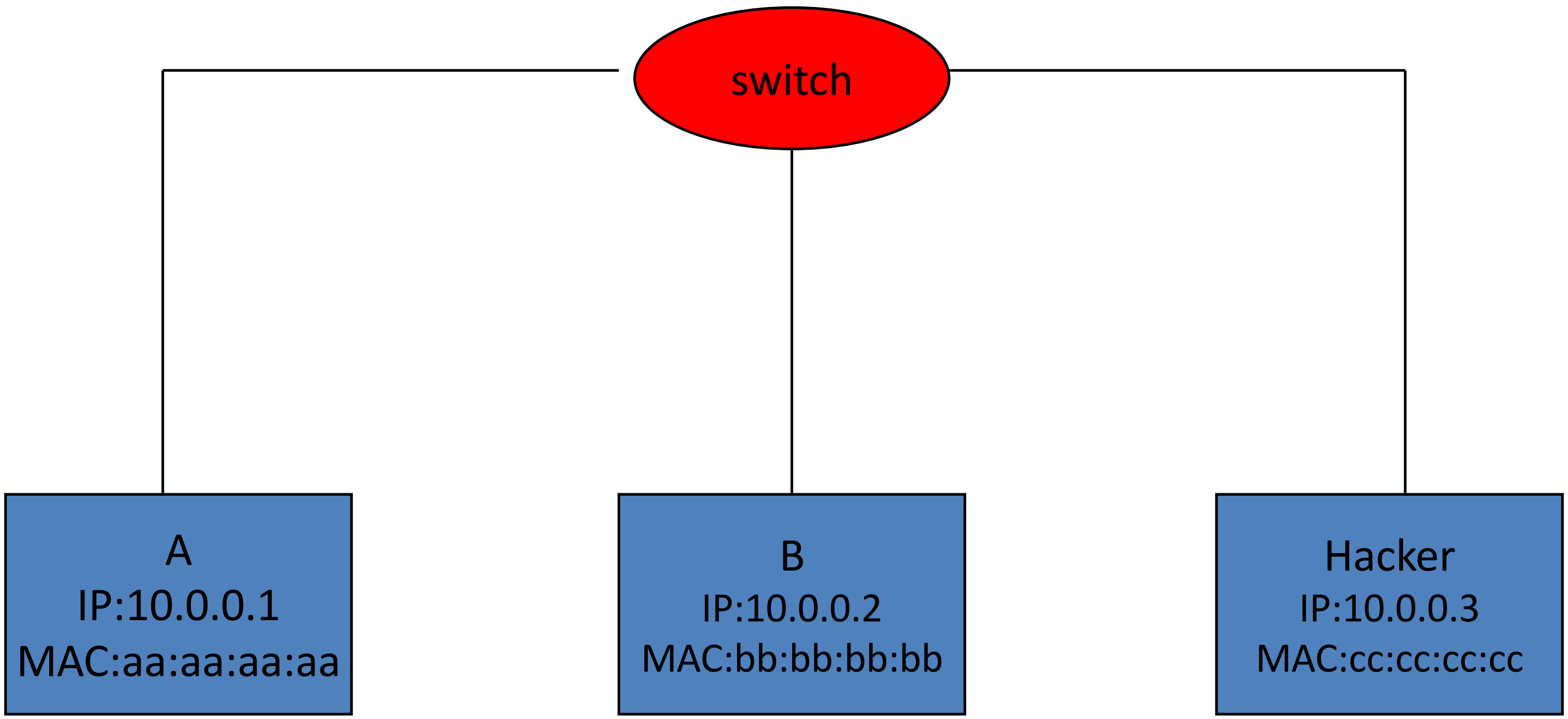


ARP cache

IP	MAC
10.0.0.2	bb:bb:bb:bb

ARP cache

IP	MAC
10.0.0.1	aa:aa:aa:aa



ARP cache

IP	MAC
10.0.0.2	cc:cc:cc:cc

ARP cache

IP	MAC
10.0.0.1	aa:aa:aa:aa

A's cache is poisoned

Defenses against ARP Spoofing

- No Universal defense.
- Use static ARP entries
  - Cannot be updated
  - Spoofed ARP replies are ignored.
  - ARP table needs a static entry for each machine on the network.
  - Large overhead
    - Deploying these tables
    - Keep the table up-to-date
  - Someone point out that Windows still accepts spoofed ARP replies and updates the static entry with the forged MAC.
    - Sabotaging the purpose of static routes.

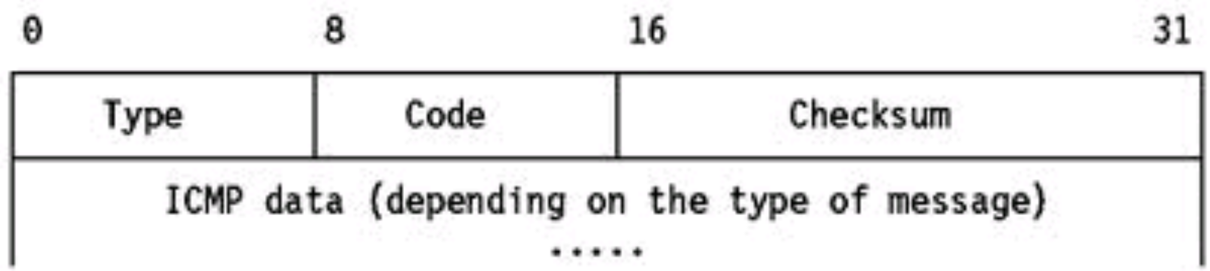
- Port Security
  - Also known as port binding or MAC Binding.
  - A feature on some high-end switches.
  - Prevents changes to the MAC tables of a switch.
    - Unless manually performed by a network administrator.
  - Not suitable for large networks and networks using DHCP.

- Arpwatch
  - A free UNIX program which listens for ARP replies on a network.
  - Build a table of IP/MAC associations and store it in a file.
  - When a MAC/IP pair changes (flip-flop), an email is sent to an administrator
- RARP (Reverse ARP)
  - Requests the IP of a known MAC.
  - Detect MAC cloning.
  - Cloning can be detected, if multiple replies are received for a single RARP

EXAMPLE: ICMP ATTACKS

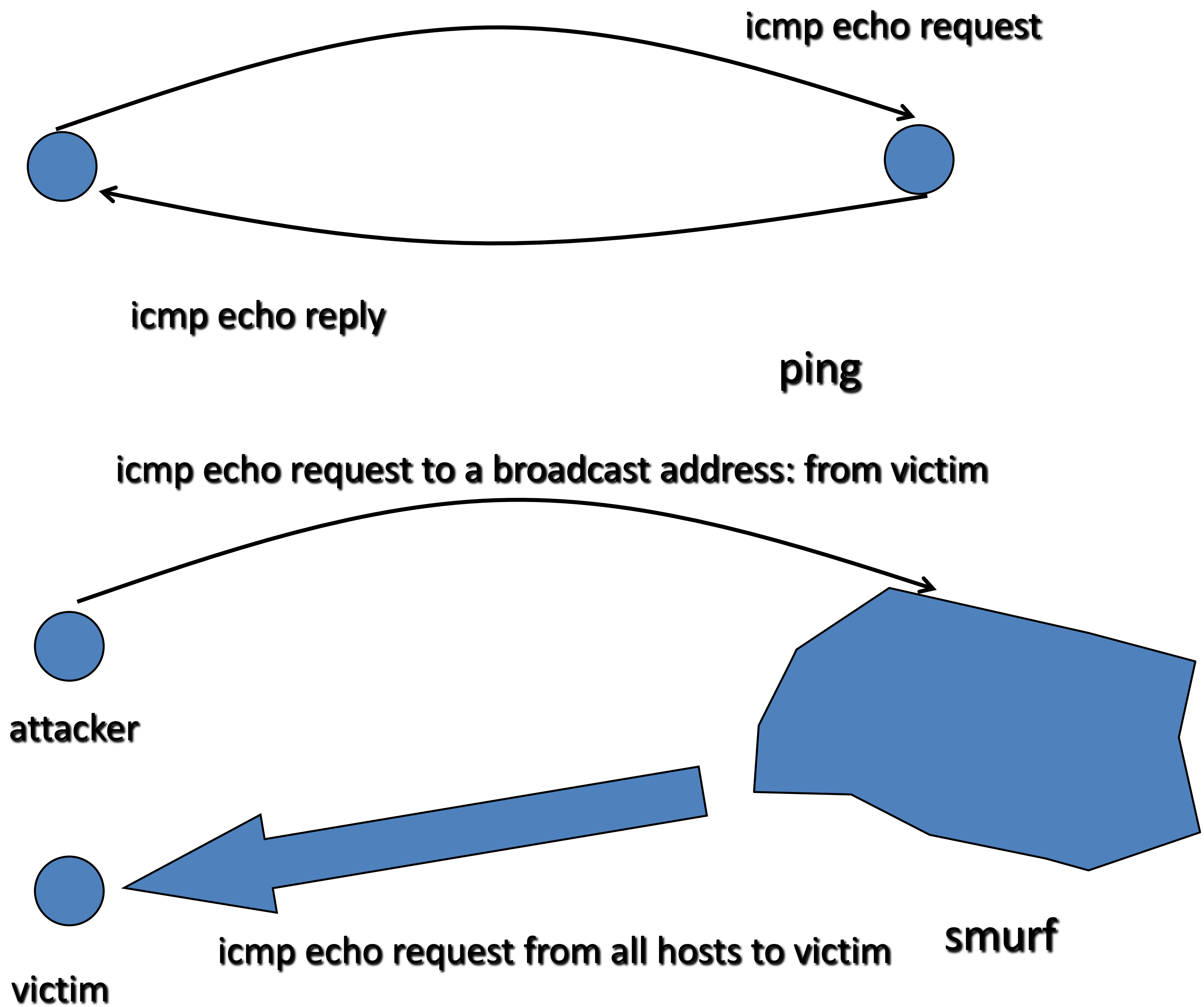
ICMP

- Internet Control Message Protocol (IP management)
- Error handling and debugging protocol
- Not authenticated!
- Encapsulated inside an IP header
- Message types
  - 40 assigned
  - 255 possible
  - about two dozen in use



Basic ICMP Message Type

- **0 Echo Reply**
- 3 Destination Unreachable
- 4 Source Quench
- **5 Redirect**
- **8 Echo**
- 11 Time Exceeded
- 12 Parameter Problem
- 13 Timestamp
- 14 Timestamp Reply
- 15 Information Request
- 16 Information Reply

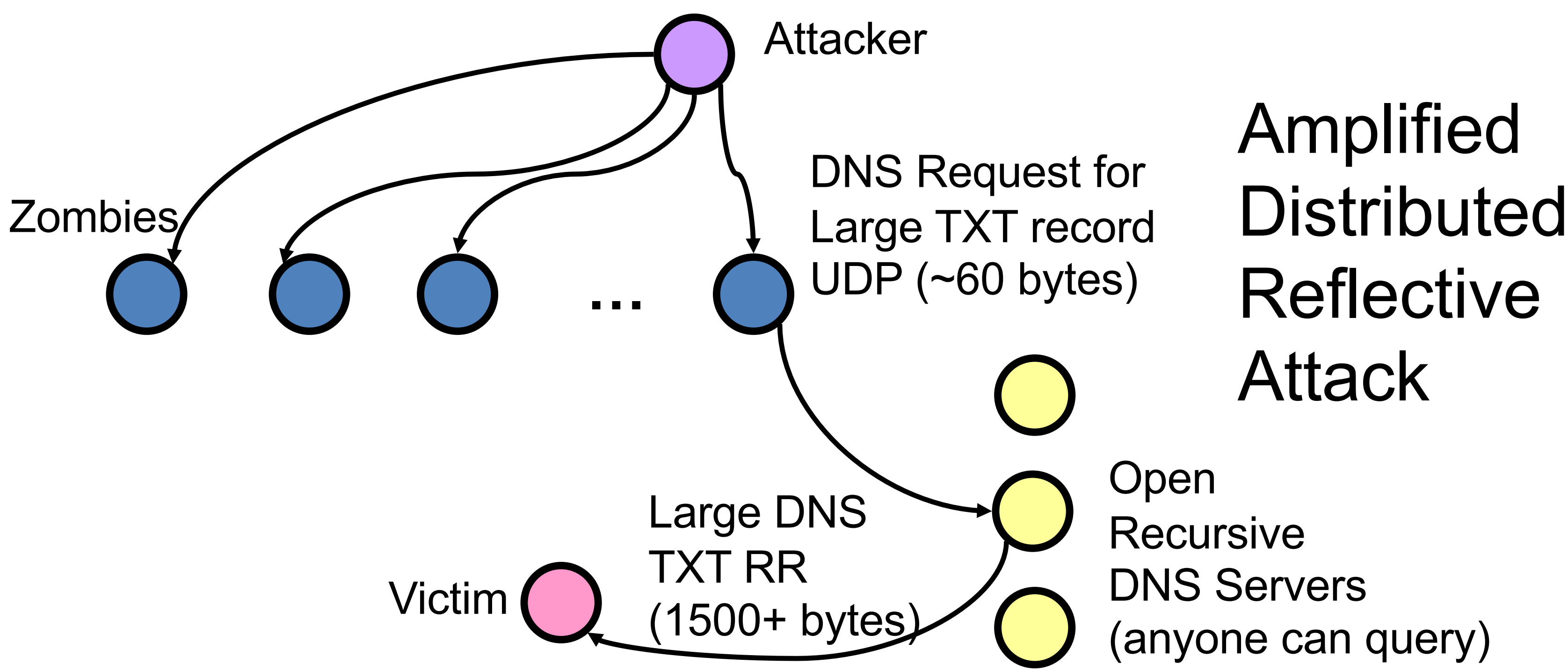

icmp echo request
icmp echo reply
ping
icmp echo request to a broadcast address: from victim
attacker
victim
icmp echo request from all hosts to victim
smurf

Smurf Attack

- Generate *ping* stream (ICMP echo request) to a network *broadcast address* with a *spoofed source IP* set to a victim host
- Every host on the ping target network will generate a ping reply (ICMP echo reply) stream, all towards the victim host
- Amplified ping reply stream can easily overwhelm the victim's network connection
- *Fraggle* and *Pingpong* exploit UDP in a similar way

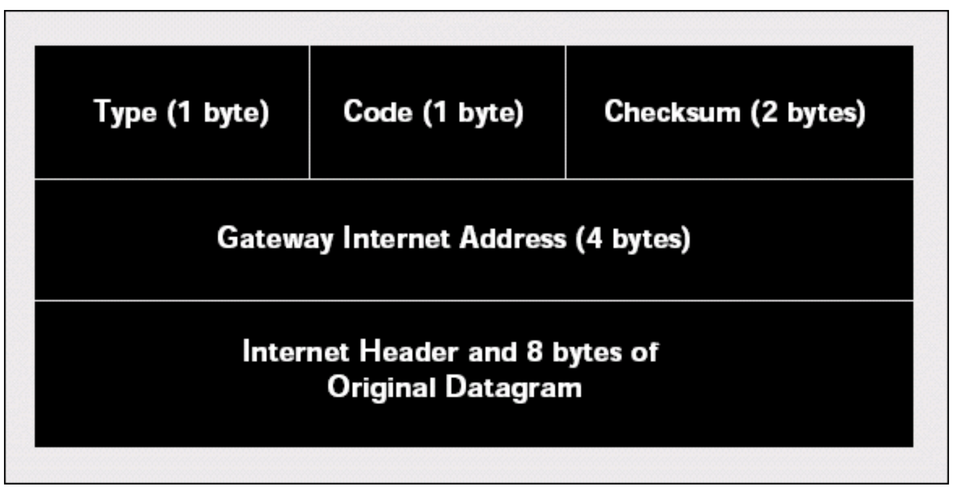
Others: DDoS Using DNS

- Botnets increasingly used for amplified distributed reflective attacks



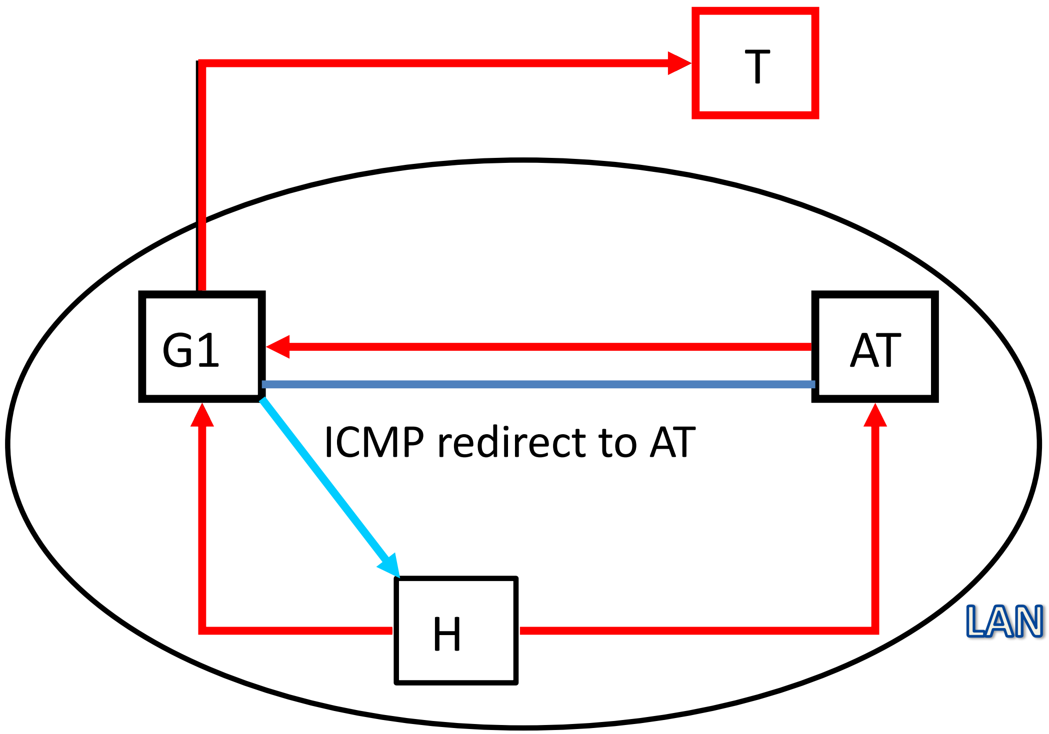
ICMP Redirect Attack

- ICMP message type 5
- “Really, you should send your packets to that gateway first, it will be faster”
  “Uh, OK, I’ll send them there then if you say so”
- Typically used as a scam to perpetrate man-in-the-middle attack or DoS
- Similar ideologically to ARP poisoning



ICMP Redirect Attack

The attacker can forge ICMP redirect packet in order to Redirect traffic to himself



ICMP Redirect - Countermeasures

- YES - Disable the ICMP REDIRECT

- NO - Linux has the “secure redirect” options but it seems to be ineffective against this attack

SUMMARY

Vulnerability

- A vulnerability (or security flaw) is a specific failure of the security controls
- Using the failure to violate the site security: exploiting the vulnerability; the person who does this: an attacker
- Defenders can also do this: vulnerability analysis (assessment), penetration testing…
- Many tools exist such as Nessus, ISS Internet Scanner, SAINT…

Vulnerability Analysis

- Even commonly, widely used TCP/IP protocols have vulnerabilities
- Very likely more in realistic, deployed applications, systems, networks…
- Understanding the vulnerabilities can help us defend against attacks more effectively