Network vulnerabilities

INGI2347: COMPUTER SYSTEM SECURITY (Spring 2014)

Marco Canini

UCL Université catholique de Louvain

Lecture slides adapted from UCL INGI2347 by Gildas Avoine and Upenn CSE331 by Steve Zdancewic

Announcements

- In-class exercises on 10 & 11 Feb
- Starting 10 Feb, Mondays class moved to BARB 13
- First challenge announced at the end of this lecture

Plan for today

Lecture 3

- Network basics

- Spoofing
- Sniffing
- Session Hijacking
- Denial of Service (DoS)

DARPA Internet Design Goals

"to develop an effective technique for multiplexed utilization of existing interconnected network" – D. Clark, The Design Philosophy of the DARPA Internet Protocols

- Secondary goals:
  - Tolerate loss of networks or gateways
  - Support multiple types of communications service
  - Accommodate a variety of networks
  - Permit distributed management of its resources
  - Be cost effective
  - Permit host attachment with a low level of effort
  - Used resources must be accountable
- Primarily designed for a benign and trustworthy environment

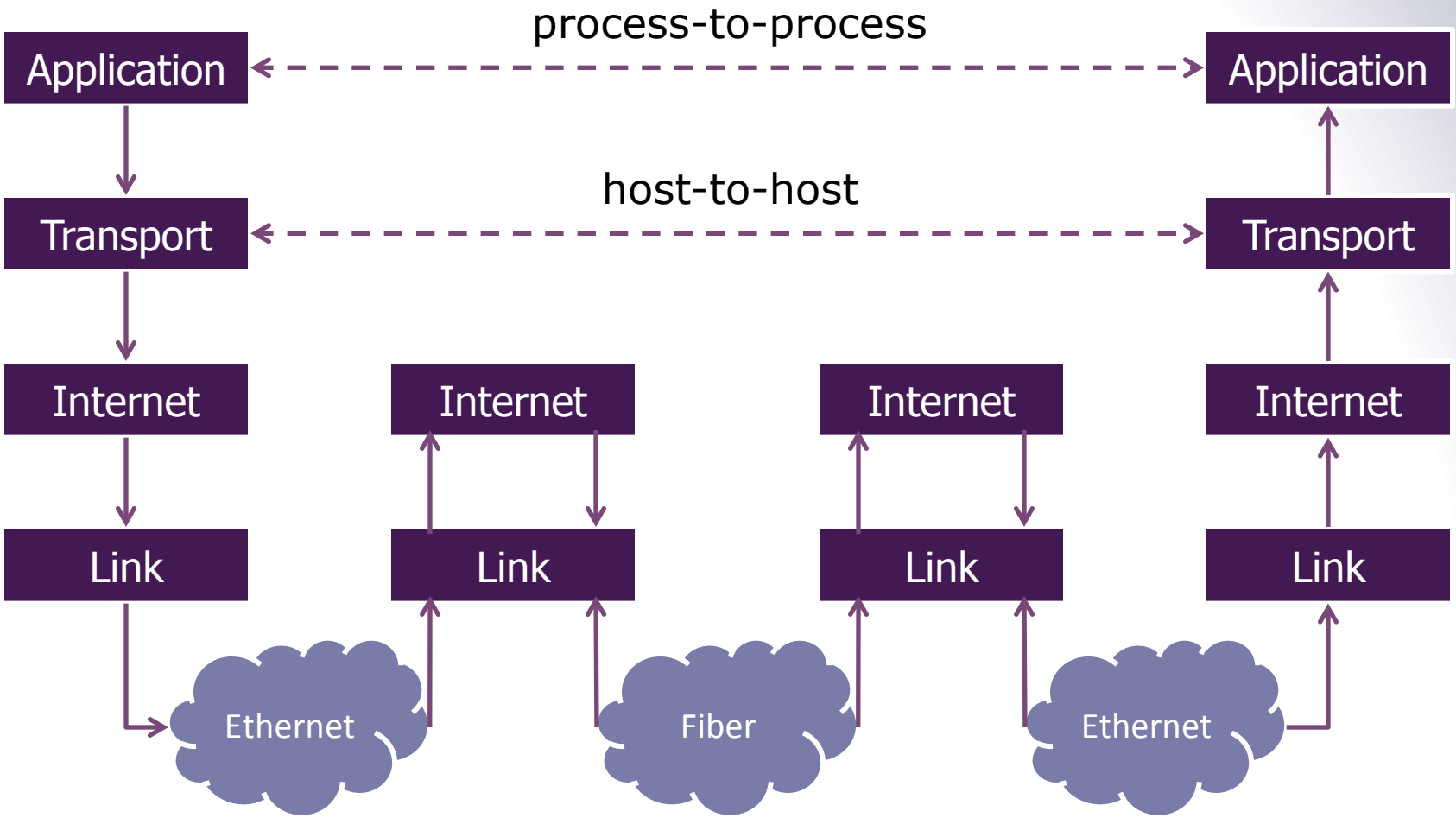
Layering

A result of abstraction in network design

- Higher level services are implemented by using services at lower levels
- Advantages
  - Decompose problems
  - Modular changes

<table>
<tr><th>TCP/IP Layered Model</th><th>OSI Layered Model</th></tr>
<tr><td rowspan="3">Application
(e.g. SMTP, HTTP, Telnet)</td><td>Application</td></tr>
<tr><td>Presentation</td></tr>
<tr><td>Session</td></tr>
<tr><td>Transport
(e.g. TCP, UDP)</td><td>Transport</td></tr>
<tr><td>Internet
(e.g. IP, ARP, ICMP)</td><td>Network</td></tr>
<tr><td rowspan="2">Link
(e.g. Ethernet, PPP, X25)</td><td>Data Link</td></tr>
<tr><td>Physical</td></tr>
</table>

Data Flow

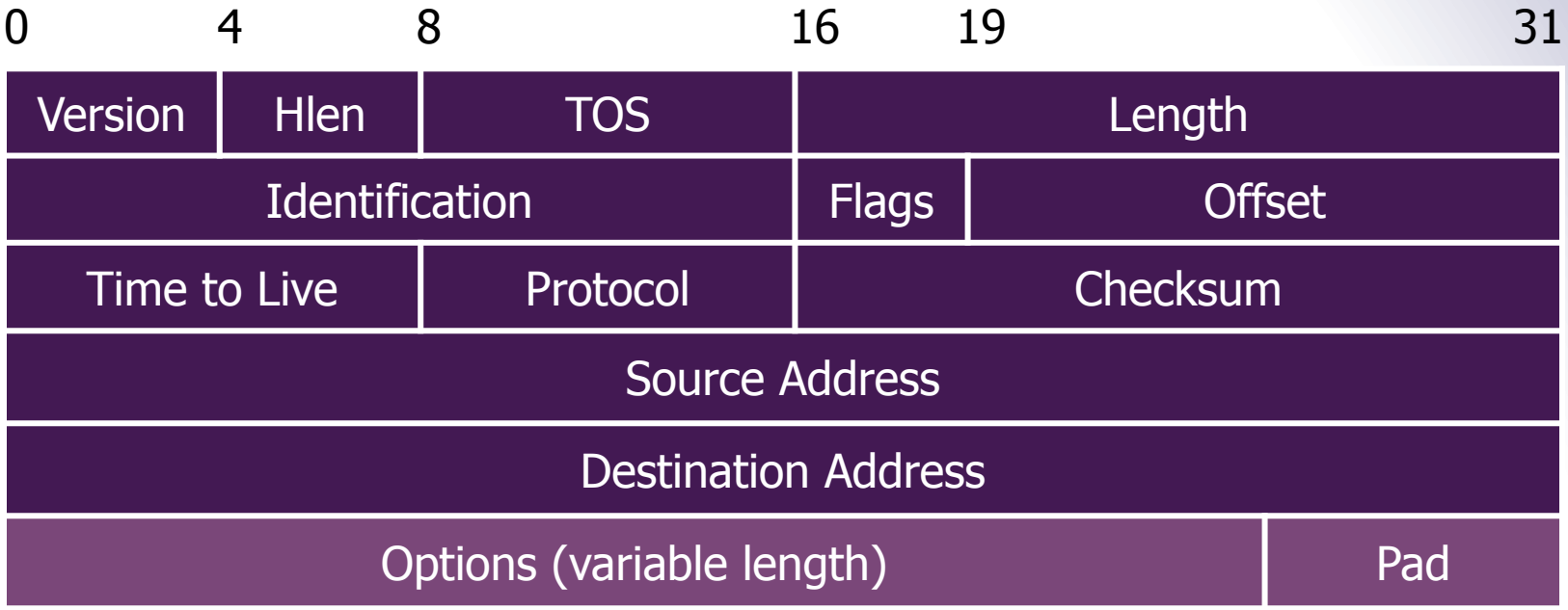

process-to-process
Application
Application
host-to-host
Transport
Transport
Internet
Internet
Internet
Internet
Link
Link
Link
Link
Ethernet
Fiber
Ethernet

Recall IP

How does the router know where to forward a packet?

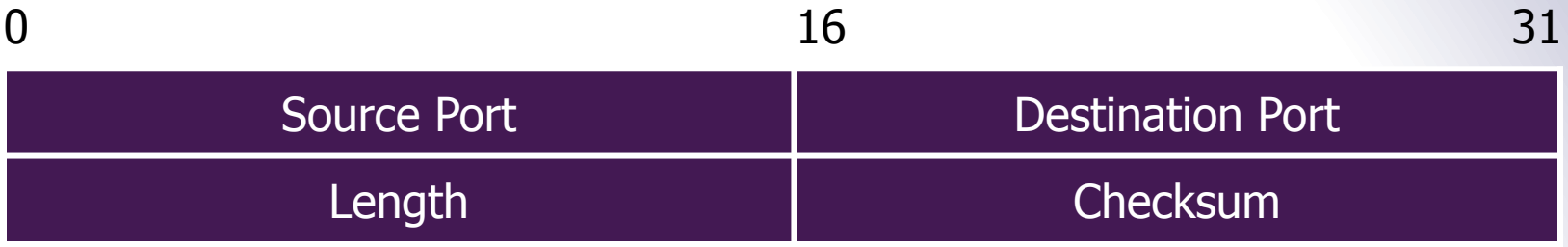
- IP adopts a datagram approach
  - Every packet (datagram) contains the destination IP address
  - Packets can be sent at anywhere at any time
  - Best-effort delivery (unreliable)

Recall IP



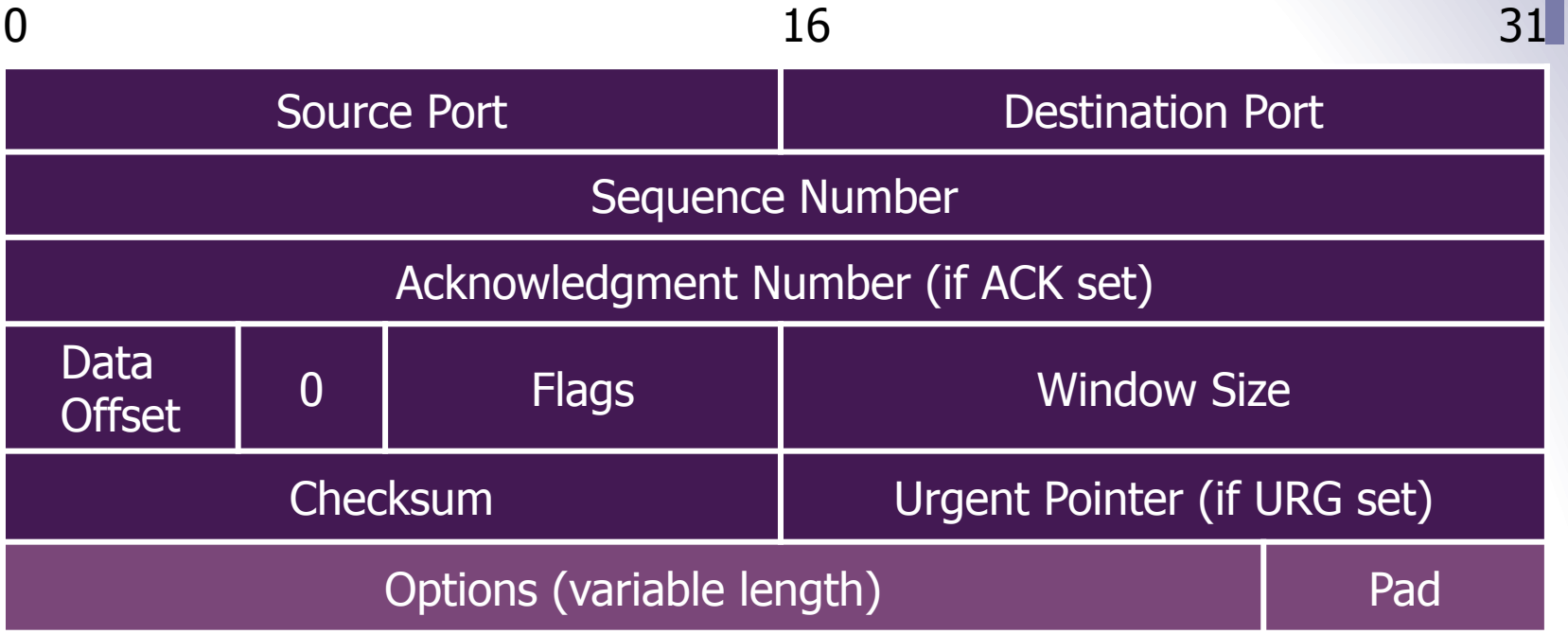
- Can you trust the contents?

Recall UDP



- Minimal message-oriented transport protocol
- Provide no guarantees for message delivery
- Port numbers to perform application demultiplexing
  - e.g., port 53 is DNS, port 2049 is NFS, port 138 is Netbios

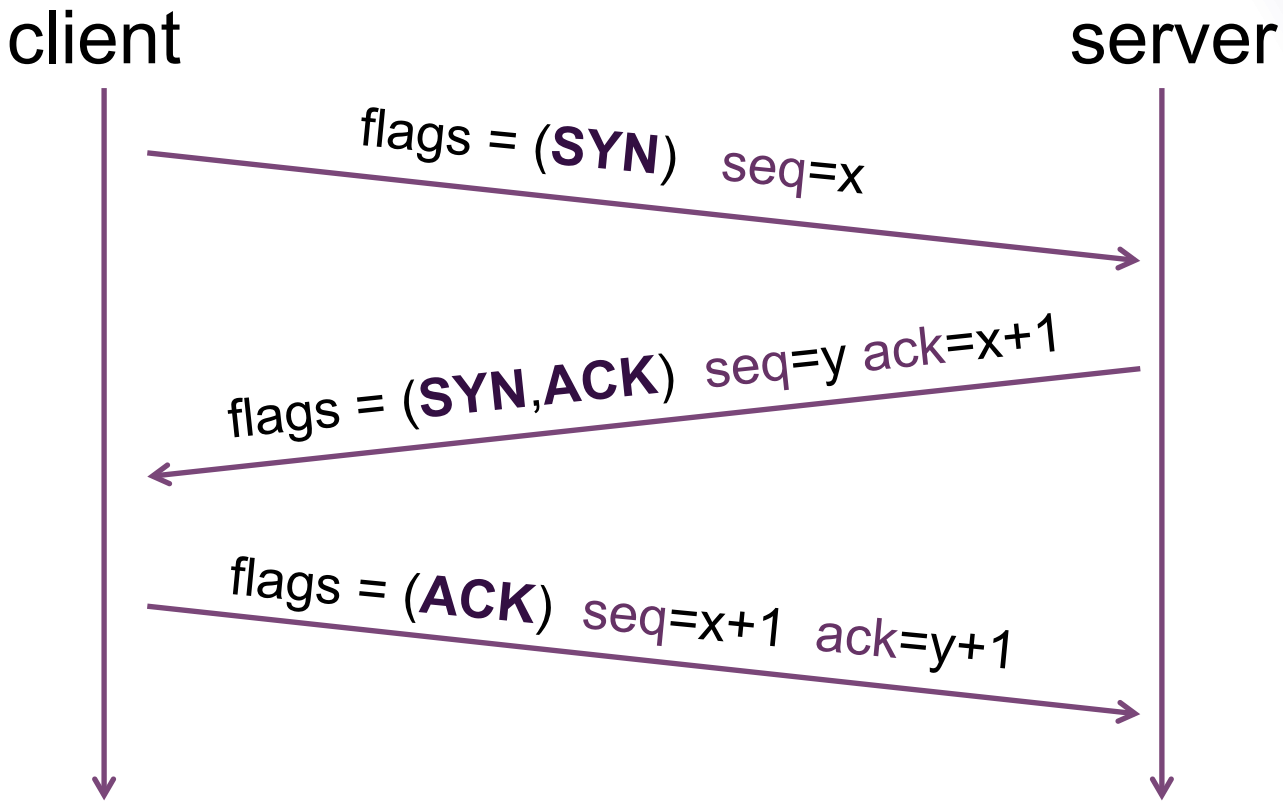
Recall TCP



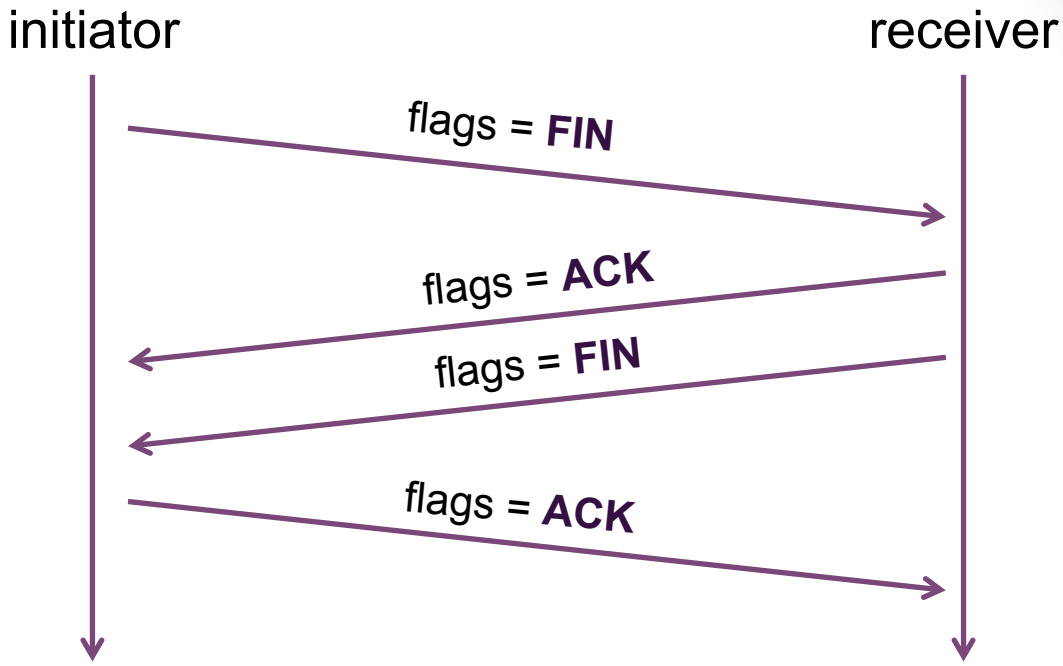
- Reliable, in-order message delivery
  - Sequence numbers identify the order of the bytes sent
  - Ack number specify the sequence number of the next expected byte
  - Sliding window flow control to not overflow receiver

TCP Connection Establishment

- 3-way handshake: SYN ; SYN + ACK ; ACK



TCP Connection Termination



- Each side of the connection terminates independently
  - A connection can be "half-open", in which case one side has terminated, but the other has not

Recall ARP (Address Resolution Protocol)

Problem:

- need mapping between IP addresses and hardware addresses

- Solution: ARP
  - Every host maintains IP-HW address mapping table (cache)
  - Timeout associate with cached info (15 min)

- Sender
  - Broadcasts "Who is IP address X?"
  - Broadcast message includes sender's IP and HW addresses

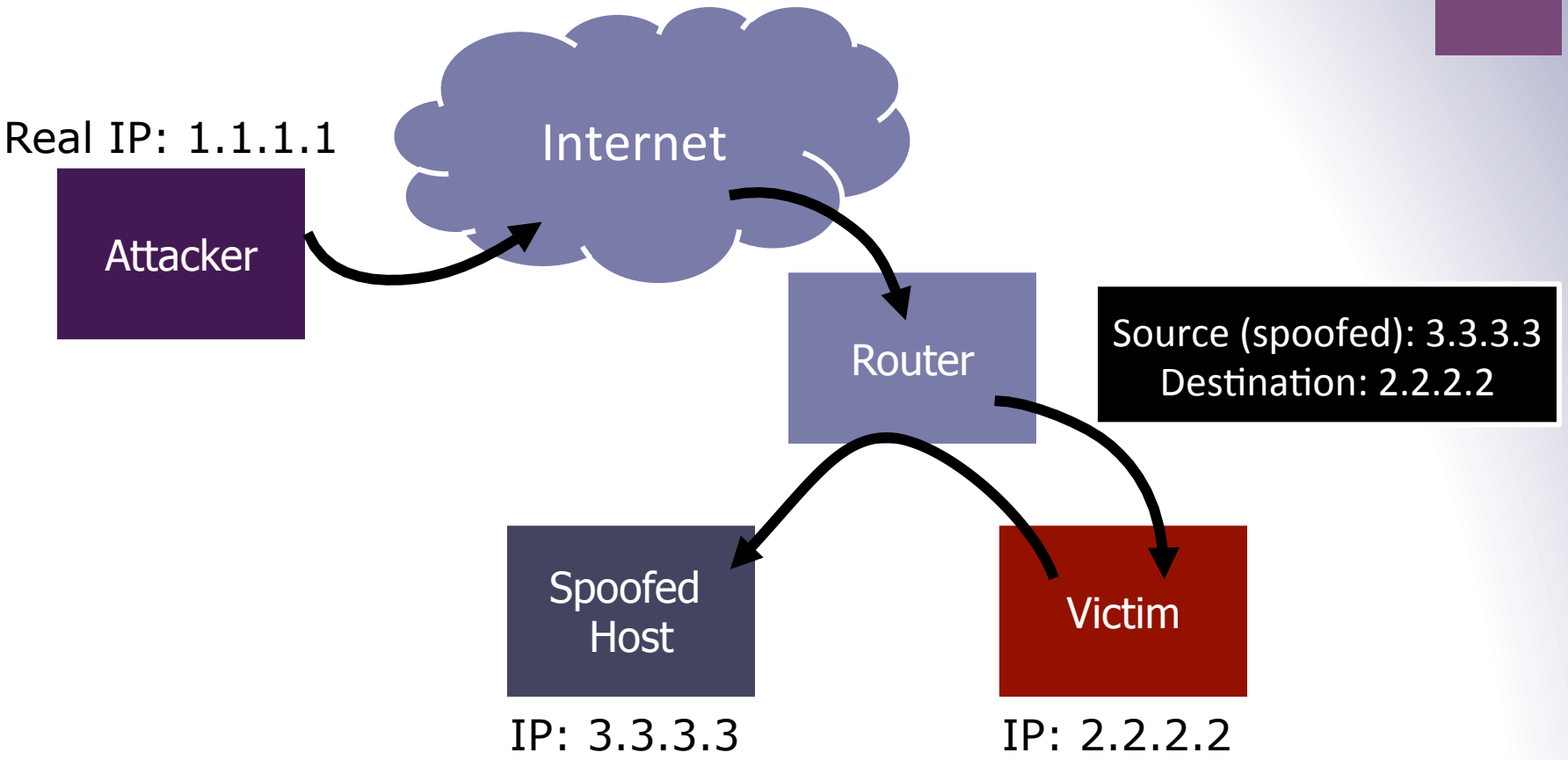
- Receivers
  - Any host with sender in cache "refreshes" timeout
  - Host with IP address X replies "IP X has HW address Y"
  - Target host adds sender (if not already in cache)

Spoofing

IP Spoofing

- Many TCP/IP suite protocols do not authenticate the source or destination of a message
- An attacker can forge a packet's source IP address and abuse the trust of that source
  - E.g., routers and firewalls can filter packets according to their source
  - E.g., certain programs (rlogin, rsh) grant access based on IP address

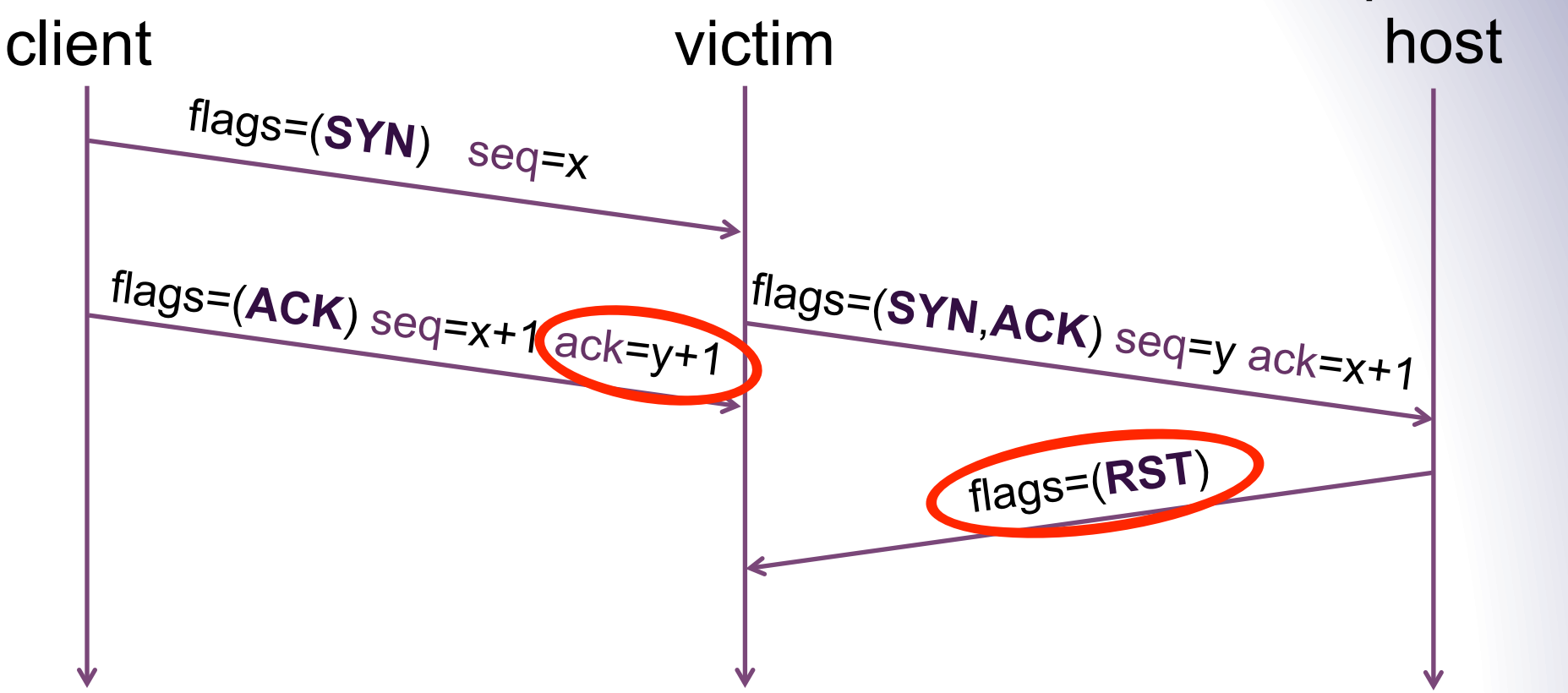
IP Spoofing Example



IP Spoofing

- The response to a forged message is sent to the forged source address
  - Spoofing mainly used when an attacker does not care about response
  - The LAN is a special case as the attacker can observe the response
- Easy to use with protocols based on UDP
- Spoofing a TCP connection is more difficult
  - Recall 3-way handshake: SYN ; SYN + ACK ; ACK

TCP Spoofing



TCP Spoofing

- Initial Sequence Number (ISN) must be guessed
  - Modern TCP implementations defend against spoofing attacks by choosing the ISN at random for every connection
- The spoofed host will respond with a RST packet that resets the handshake
- To succeed the attacker must either:
  - Send the data on the TCP connection before the spoofed host responds
  - Spoof an offline host (or putting it offline, if necessary)

ISN Prediction

- The original standard (RFC 793) requires that the ISN be incremented once every four microseconds
- In some poor TCP implementations the next ISN can be predicted
- Attacker's procedure (ISN prediction):
  - Open a few authentic connections (for example SMTP) to obtain the current ISN and increment samples
  - Launch the forged connection using the last ISN plus an increment obtained from those samples
  - To increase success probability, launch multiple forged connections with different increments hoping that at least one is correct

ISN Prediction

```
14:18:25.90   kevin.1000   >   bob.514:      S   1382726990
14:18:26.09   bob.514      >   kevin.1000:   S   2021824000   ack   1382726991
14:18:26.17   kevin.1000   >   bob.514:      R   1382726991
14:18:26.50   kevin.999    >   bob.514:      S   1382726991
14:18:26.69   bob.514      >   kevin.999:    S   2021952000   ack   1382726992
14:18:26.77   kevin.999    >   bob.514:      R   1382726992
14:18:27.01   kevin.998    >   bob.514:      S   1382726992
14:18:27.17   bob.514      >   kevin.998:    S   2022080000   ack   1382726993
14:18:27.25   kevin.998    >   bob.514:      R   1382726993
14:18:27.54   kevin.997    >   bob.514:      S   1382726993
14:18:27.71   bob.514      >   kevin.997:    S   2022208000   ack   1382726994
14:18:27.79   kevin.997    >   bob.514:      R   1382726994
14:18:28.05   kevin.996    >   bob.514:      S   1382726994
14:18:28.22   bob.514      >   kevin.996:    S   2022336000   ack   1382726995
14:18:28.30   kevin.996    >   bob.514:      R   1382726995
```

128,000

128,000

128,000

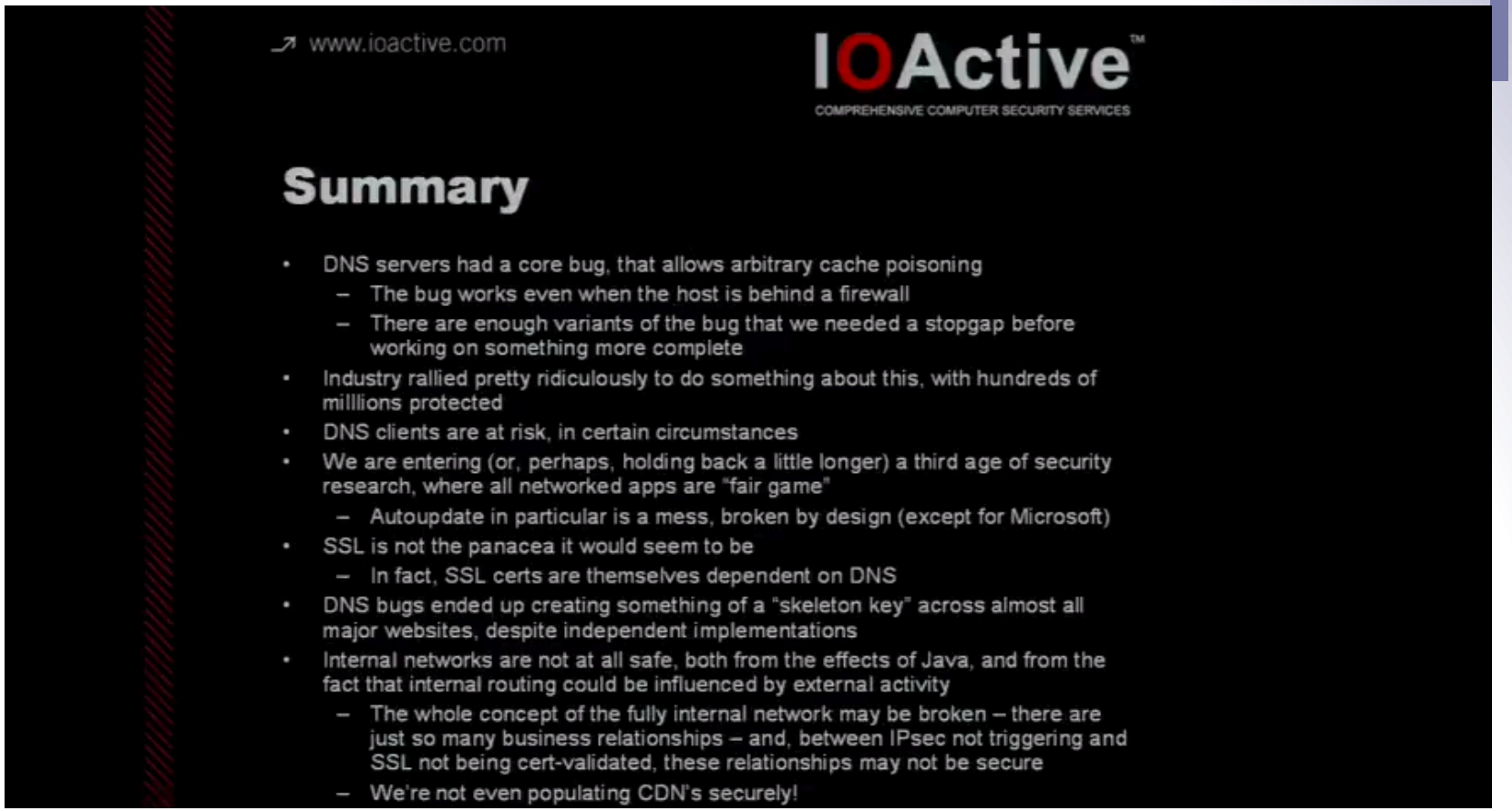
ARP Spoofing

- Recall: ARP maps between IP addresses and hardware addresses
- ARP is very simple and insecure
  - client: who knows the Ethernet address of 1.2.3.4?
  - anybody: 1.2.3.4 has Ethernet address 01:02:03:04:05:06
- It is easy to forge responses (even non-solicited) to redirect traffic!

DNS Spoofing

- Domain Name System maps names to IP addresses
- DNS is a distributed system
  - Clients (e.g. the browser) contact a recursive nameserver
  - Recursive nameserver maintain a cache of previously resolved names
  - For a new request, the nameserver asks another server and recurses until an authoritative answer is found
- An attacker can attempt to inject a false translation to a nameserver, and hence poison its cache
  - Exploit flaws in the DNS software
  - Flood a server with forged answers that guess the Query ID, a DNS packet field that uniquely identifies every request
- A poisoned server can cause client traffic to be diverted to another computer, usually under control of the attacker

Extra



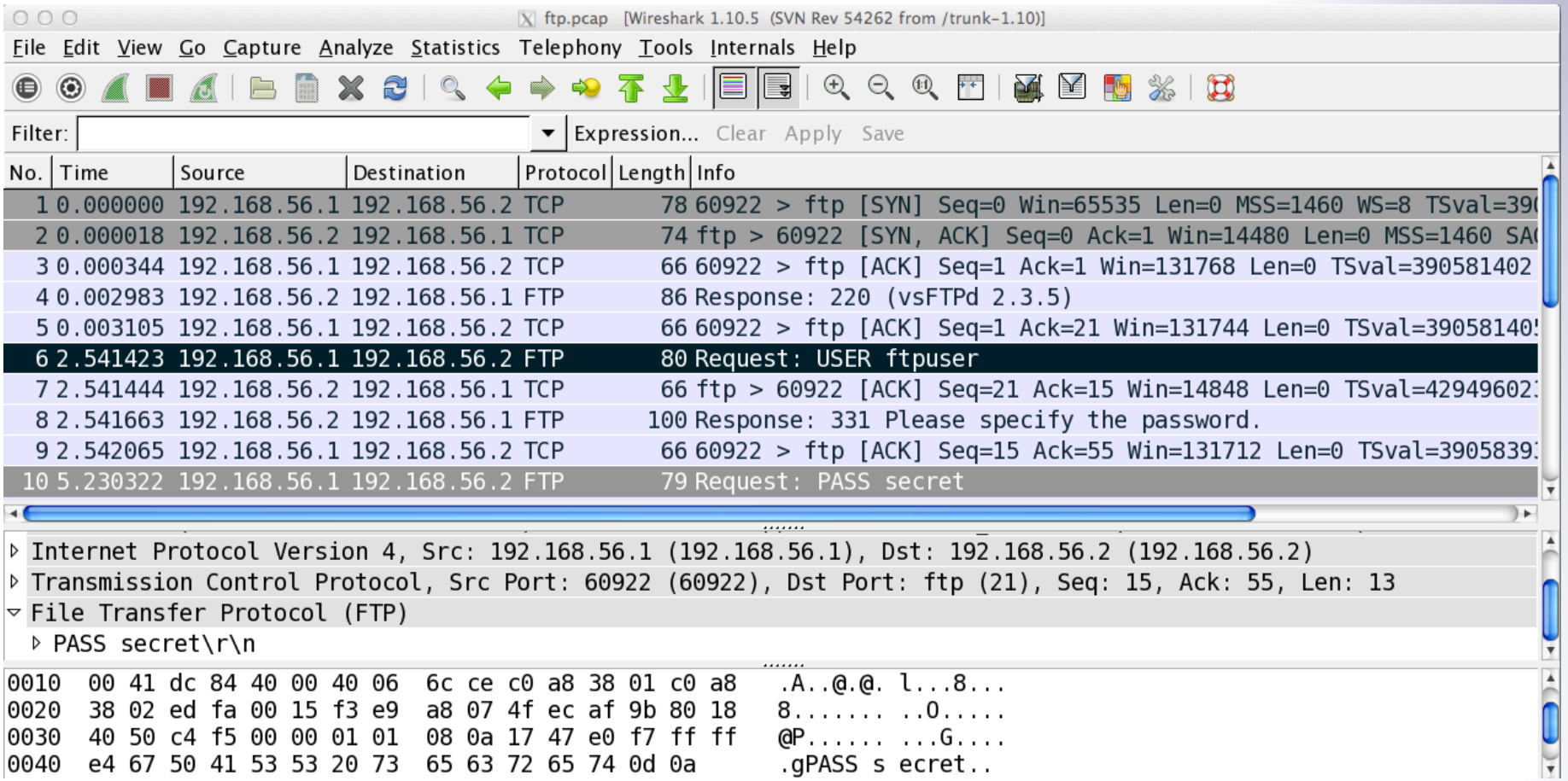
- Watch Dan Kaminsky's 2008 talk on an extremely serious vulnerability in DNS https://vimeo.com/17247507

Sniffing

Sniffing

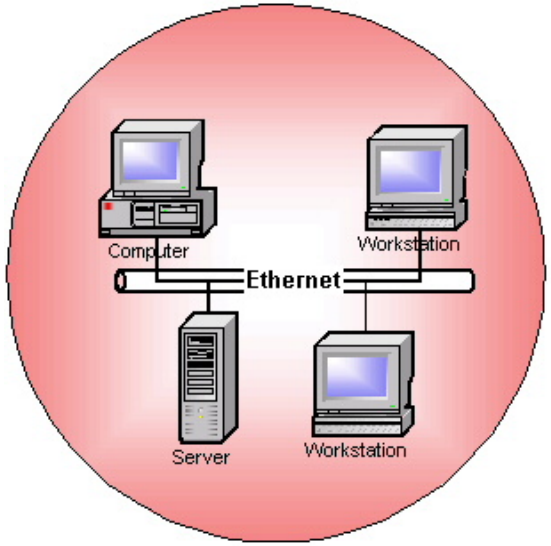
- Many protocols use clear text authentication
- By eavesdropping traffic on a network segment, we can obtain usernames and passwords
- A password gives access to a remote system from which we can sniff and obtain new passwords

Sniffing Tool: Wireshark

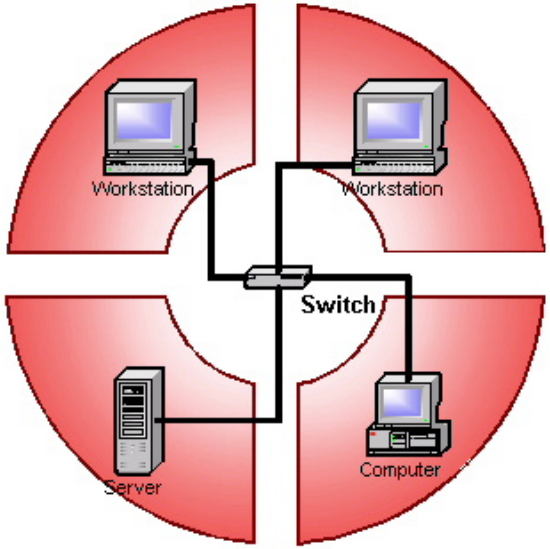


Sniffing

- Switched networks limit sniffing possibilities, but switches can be manipulated



Shared Ethernet: : 1 collision domain for multiple nodes. The possibility of collisions. Non-deterministic



Switched Full Duplex Ethernet: 1 collision domain per node. Use of switch. No possibility of collisions. Deterministic.

Source: Industrial Ethernet University

Session Hijacking

Session Hijacking

- Instead of stealing a password, an attacker can wait until a user authenticates and then steal his session
- This technique can be applied at several levels:
  - Modem
  - TCP
  - HTTP

Hijacking Modem Session

- The modem gives access to a serial line
  - E.g. remote access
- A user may drop the line without quitting the online session
- The terminal's session remains active for a while
- The next user (or hacker) who connects to the modem finds the preceding user's session

Hijacking TCP Session

- If an attacker can spy on a TCP connection, he can insert a TCP packet with correct sequence numbers
- Inserting an additional packet in a TCP connection creates a packet avalanche:
  - The source, who has never sent the packet, doesn't agree with the acknowledged sequence number and emits an acknowledgement
  - The destination, who has seen the packet, insists on the sequence number and also sends an acknowledgement
- An attacker can also take over the session, terminating the connection only on one end

Hijacking HTTP "Session"

- HTTP doesn't have the concept of a session
- It is made of independent requests/responses
- Websites use artificial means to recognize requests belonging to a session:
  - Cookies
  - Personalized URLs
- If an attacker can spy on these data, he can create requests that would be part of the same session

Protection Against Session Hijacking

- Encryption of data traffic (in particular session key)
- Require two-factor authentication (e.g., via SMS)
- Change the value of the cookie with each and every request, thus reducing the attack window
- Use random session keys to prevent guessing

Denial of Service

Denial of Service (DoS)

- An attempt to make a machine or network resource unavailable to its intended users, either temporarily or indefinitely
- Many methods of attack exist, classifiable as:
  - Consumption of computation resources (e.g. CPU, memory, bandwidth)
  - Disruption of configuration (e.g. routing information)
  - Disruption of state information (e.g. sending TCP RST packets)
  - Disruption of physical components
  - Obstructing communication media between the victim and its users
- DoS attacks typically involve forged IP addresses

Historical example

Ping of death (PoD)

- One of the earliest denial of service attack
  - Up to 1997, affected Unix, Linux, Mac, Windows, printers, and routers
- Attack based on sending a malformed ICMP packet

Recall ICMP (Internet Control Message Protocol)

- Collection of error & control messages
- Sent back to the source when Router or Host cannot process packet correctly
- Error Examples:
  - Destination host unreachable
  - Reassembly process failed
  - TTL reached 0
  - IP Header Checksum failed
- Query Example:
  - Ping uses Echo Request/Reply to test the reachability of a host

Recall IP Fragmentation

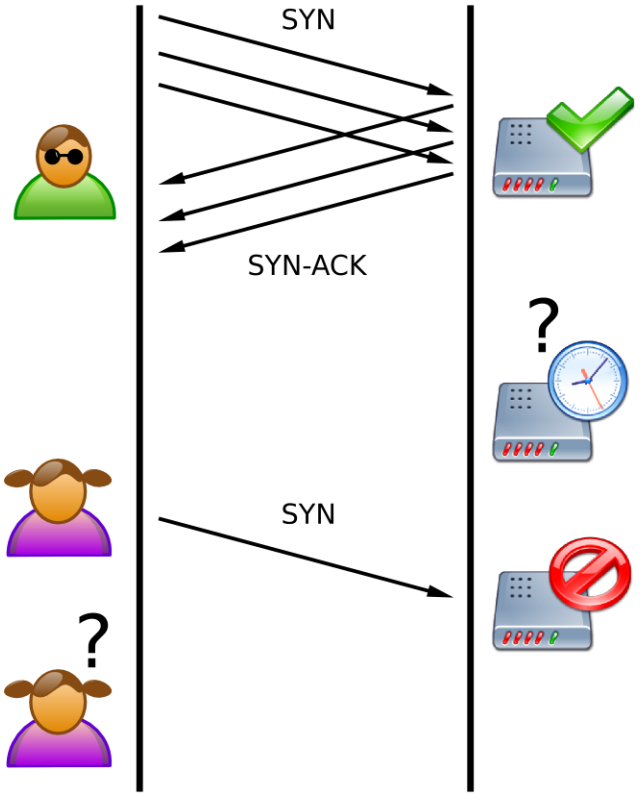
- Split datagram into fragments
  - Why? Maximum Transmission Unit (MTU) may vary for different networks
- Each fragment is a complete datagram (own header)
  - Identification: uniquely identifying the group of fragments
  - Flags: all fragments except last have the MF flag set
  - Offset: relative to the beginning of the original unfragmented IP datagram
- Receiving host reassembles them

Ping of death

- Ping size is normally 56 bytes (84 with header)
- Historically sending a ping larger than the maximum IPv4 packet size (65,535 bytes) could crash a target
- How to send a packet large than maximum length?
  - Exploit fragmentation (problem has nothing to do with ICMP)
  - Maximum offset (13 bits) addresses 65,528 bytes
  - Send packet with maximum offset and more data than 8 bytes
  - When packet is reassembled, a buffer overflow can occur, which often causes a system crash
- Prevented with stronger checks at receiver host

SYN Flooding

- Send a large number TCP/SYN packets
  - Recall 3-way handshake: SYN ; SYN + ACK ; ACK



Source: Wikipedia

SYN Flooding

- On receiving the SYN packet, the server allocates necessary memory for the connection and enters it in a queue of half open connections
- Once the queue overflows, the server cannot accept any new connection
- The attacker can forge the source address of his SYN packets to remain anonymous
- Recent versions of operating systems (Windows, Unix, Linux) are protected against such attacks

Protections Against SYN Flooding

- Increase the size of the queue
- Reduce timeout while server is waiting for an ACK
- Drop the oldest SYN in the queue
- Filtering e.g. on IP addresses
- SYN Cache
  - Allocate minimal state on the server, reply with SYN + ACK and complete the connection creation once the ACK is received
  - Host compute a hash based on some secrete bits, IP addresses and transport ports that determines the location in a global hash table where the incomplete connection information is stored
  - Still must be prepared to overflow

Protections Against SYN Flooding

SYN cookies

- Once the connection queue is almost filled up, the server uses SYN cookies
- Upon reception of a SYN:
  - The server sends a SYN + ACK containing a SYN cookie
  - The server erases the SYN entry
- Upon reception of an ACK:
  - The server checks whether it contains a valid cookie
  - If so this highly likely means that the client has already sent a SYN and so it is an honest client

SYN Cookie Content

SYN cookies are specific Initial Sequence Numbers:

- *t* is a counter incremented every 64 s modulo 32
- *m* is the Maximum Segment Size encoded on 3 bits
- *s* is the result of a cryptographic hash function computed on t and the IP address and port number of the server and client

ISN =

t **mod** 32	*m*	*s*
5 bits	3 bits	24 bits

32 bits

SYN Cookie Check

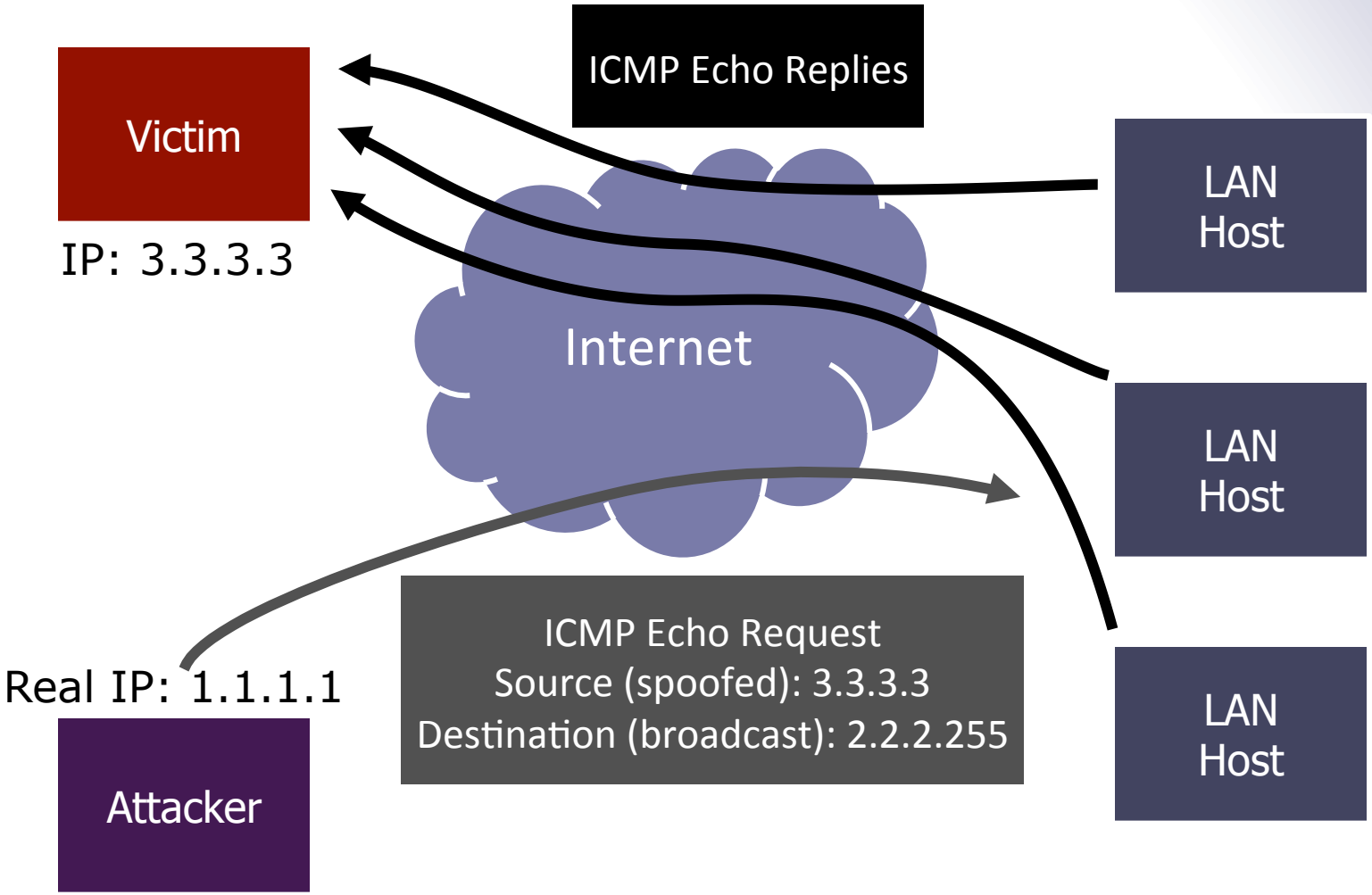
Upon reception of an ACK, the server carries out the following operations:

- Check that the received value *t* is valid with respect to the current time
  - Otherwise, this means the connection is expired
- Recompute *s* to check its validity
- Decode the value *m*, which allows the server to reconstruct the SYN queue entry

Smurf Attack

- Drown the target with the help of traffic amplifiers
- Typical case: ICMP Echo Request (ping)
- The attacker sends a ping packet with the target address as source address
- The target machine sends its response to the victim
- If the attacker sends the packet to a broadcast address, all machines in the network will reply to the victim

Smurf Example



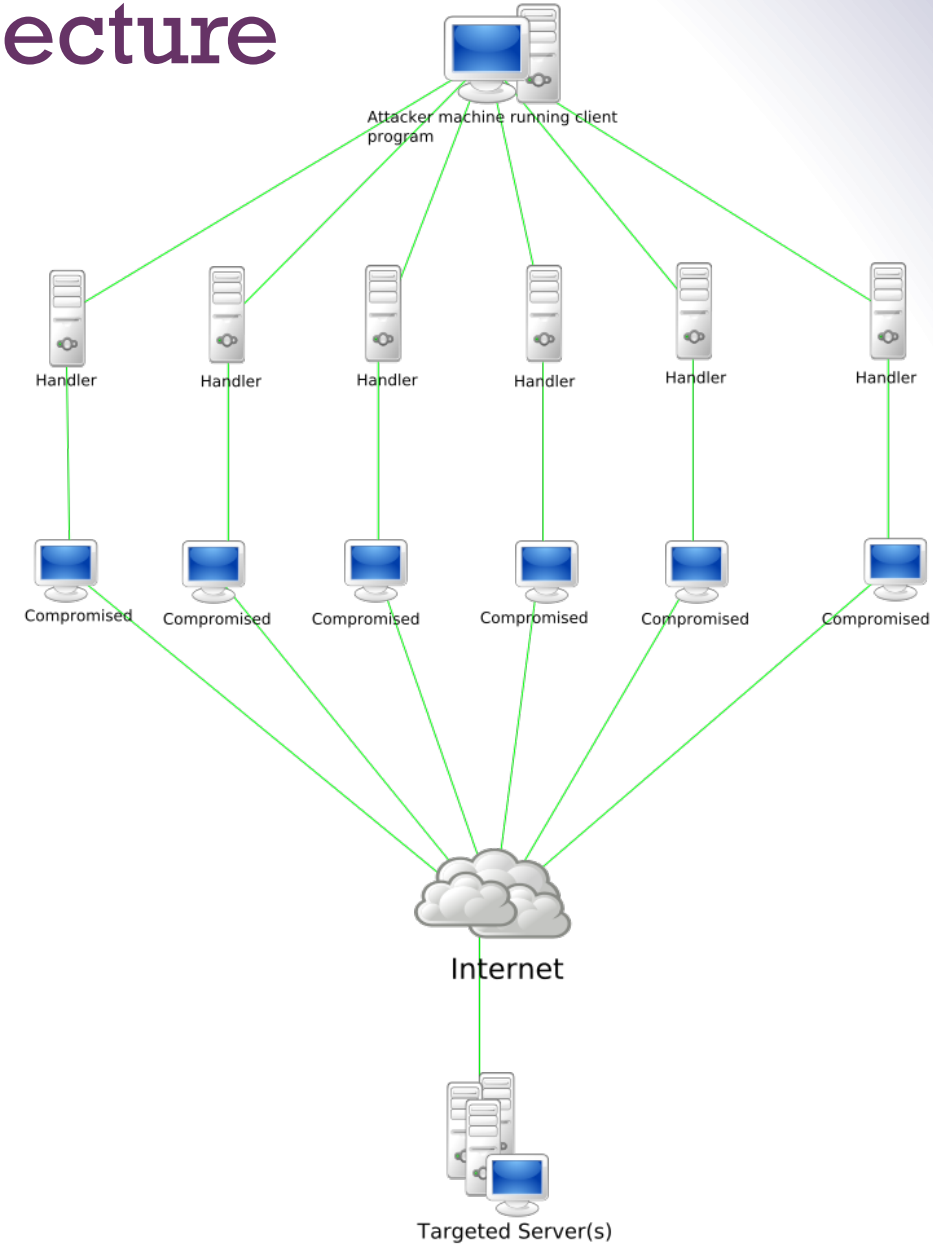
Protections Against Smurf Attack

- Configure individual hosts and routers not to respond to ping requests to broadcast addresses
- Configure routers not to forward packets directed to broadcast addresses
- Traffic amplification is more effective when application replies are much larger than requests

DDoS: Distributed Denial of Service

- To increase the efficiency of Denial of Service, attackers hack into several machines and install agents on them to form a botnet
- Several master machines control the botnet
- The attacker sends commands to the masters which in turn execute the attack through the agents

DDoS: Architecture



Source: Wikipedia

DDoS: Characteristics

- The power (bandwidth) of the attack is multiplied by the agents
- It is more difficult to trace the attackers
  - 2 intermediate layers
  - Encrypted control traffic (even peer-to-peer)
- Since attack comes from several sources, it is much more difficult to filter it

DDoS Recent Examples

- Mar 2013 – DDoS attack against Spamhaus
  - reportedly the largest in history, peaked at 300 Gbps [CloudFlare]
- Oct 2012 – HSBC experienced a downtime of many of its website worldwide
- Aug 2012 – WikiLeaks was hit by a DDoS attack
- Jun 2012 – CIA website fallen foul of a DDoS attack
- Mar 2011 – WordPress.com suffered an extremely large DDoS attack
  - Nowadays thousands of hacked WordPress websites are used to carry out DDoS attackes

Any questions?

Stay tuned

Next time you will learn about

Firewalls