Imperial College
London

CO331 – Network and Web Security

7. Network Security

Dr Sergio Maffeis
Department of Computing
Course web page: http://www.doc.ic.ac.uk/~maffeis/331

WiFi hotspot

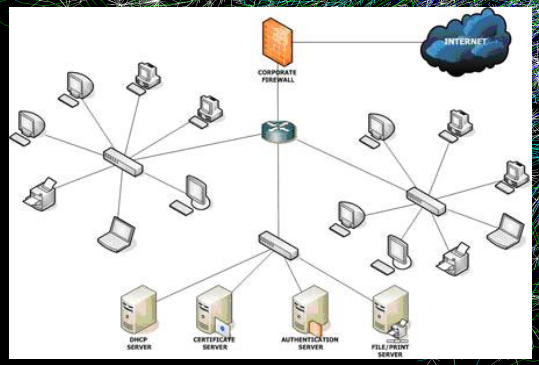
Home LAN

Internet

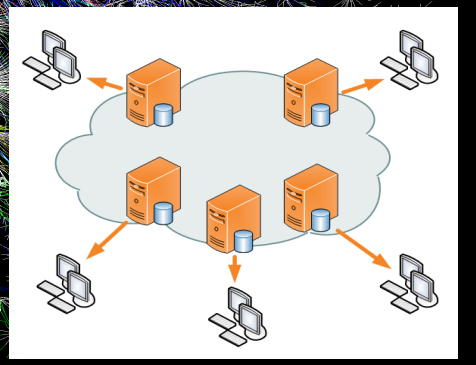
/ˈɪntənɛt/

noun

a global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.



Corporate network



Content Delivery Network

Network addresses

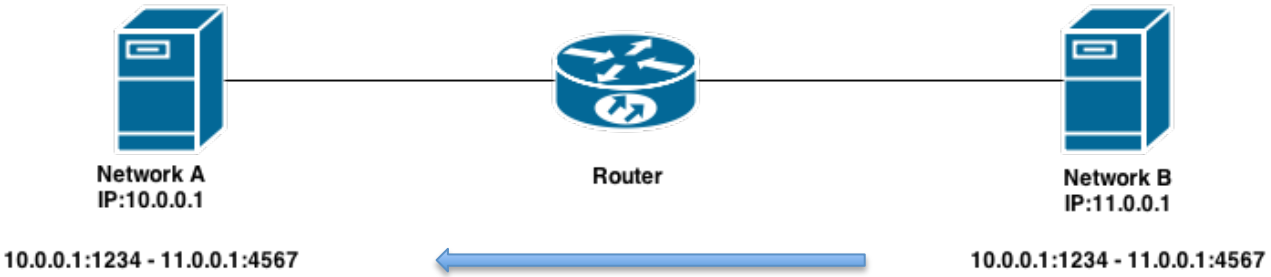
Imperial College London

- *Hosts* are the machines connected to the network
- Each internet host needs an IP address, such as 155.198.140.14
- Network services are multiplexed through the same IP address using *ports*
  - 155.198.140.14**:80**
  - Common services **tend** to be hosted on standard ports
    - SSH: 22, DNS: 53, HTTP: 80, HTTPS: 443
    - We shall see that this does not always hold in practice
- One machine can have multiple IPs
  - Over time: connect at home, at work, on the go
  - At the same time
    - Client with wireless and Ethernet connections
    - *Dual-homed host* (firewall, gateway) connecting two different networks
- Multiple machines may share the same IP
  - Home router connecting desktop, laptop, iPhone
  - Port- or name-based virtual hosting of websites
- *Packet* (or *datagram*): message that is sent as a single unit on the network
  - Typically composed of *headers* + *payload*

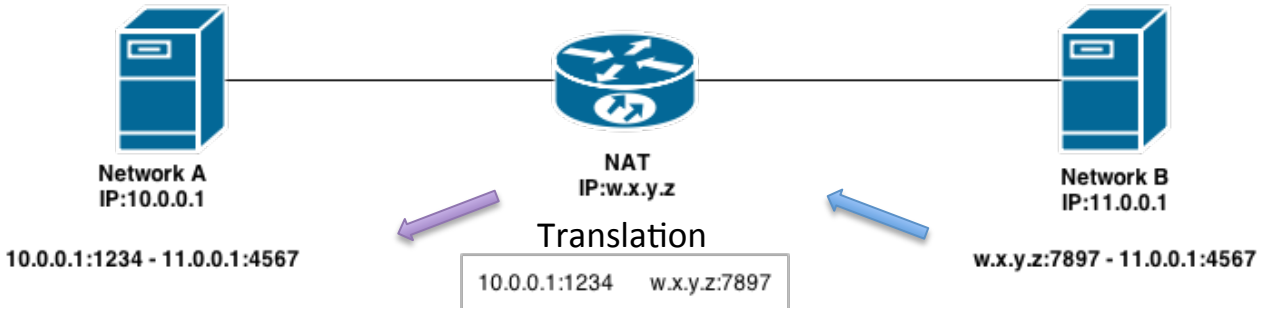
Network intermediaries

Imperial College London

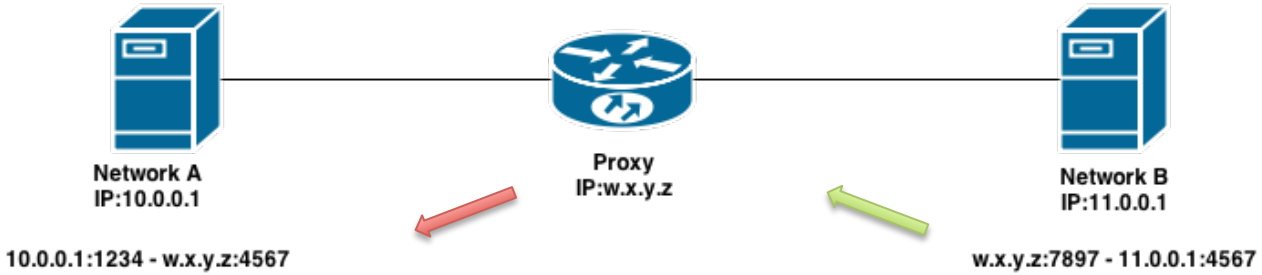
- Router
  - Connects two different networks
  - Does not modify packet addresses



- Network Address Translator (NAT)
  - Exposes a local network via the ports of 1 IP address
  - Modifies packet's IP addresses to effect the mapping



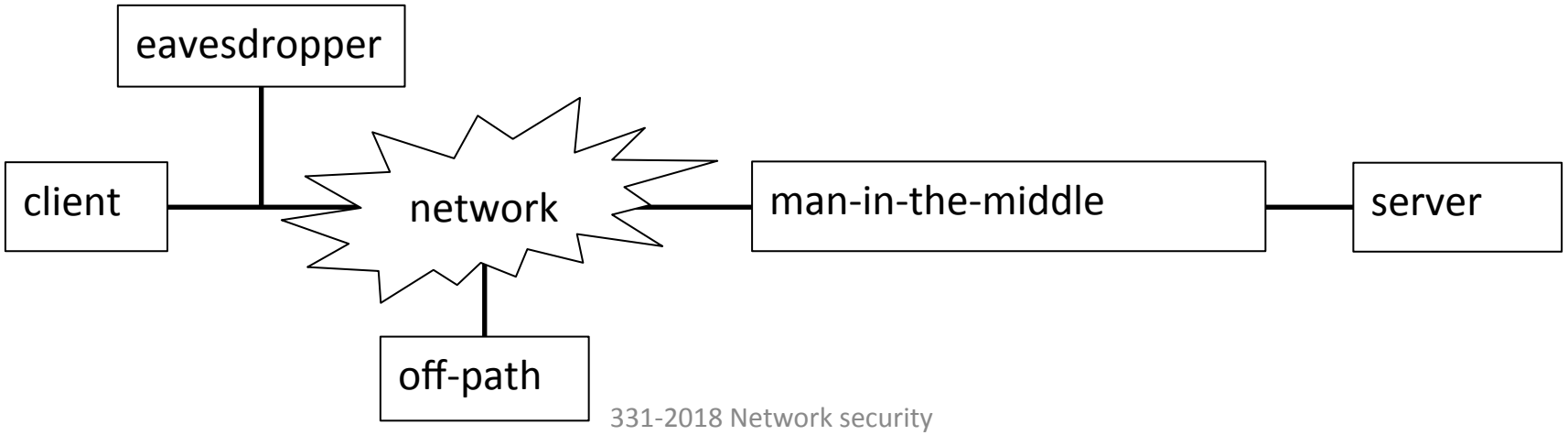
- Proxy
  - A and B communicate to proxy, not directly to each other
  - There are 2 independent packets



Network capabilities

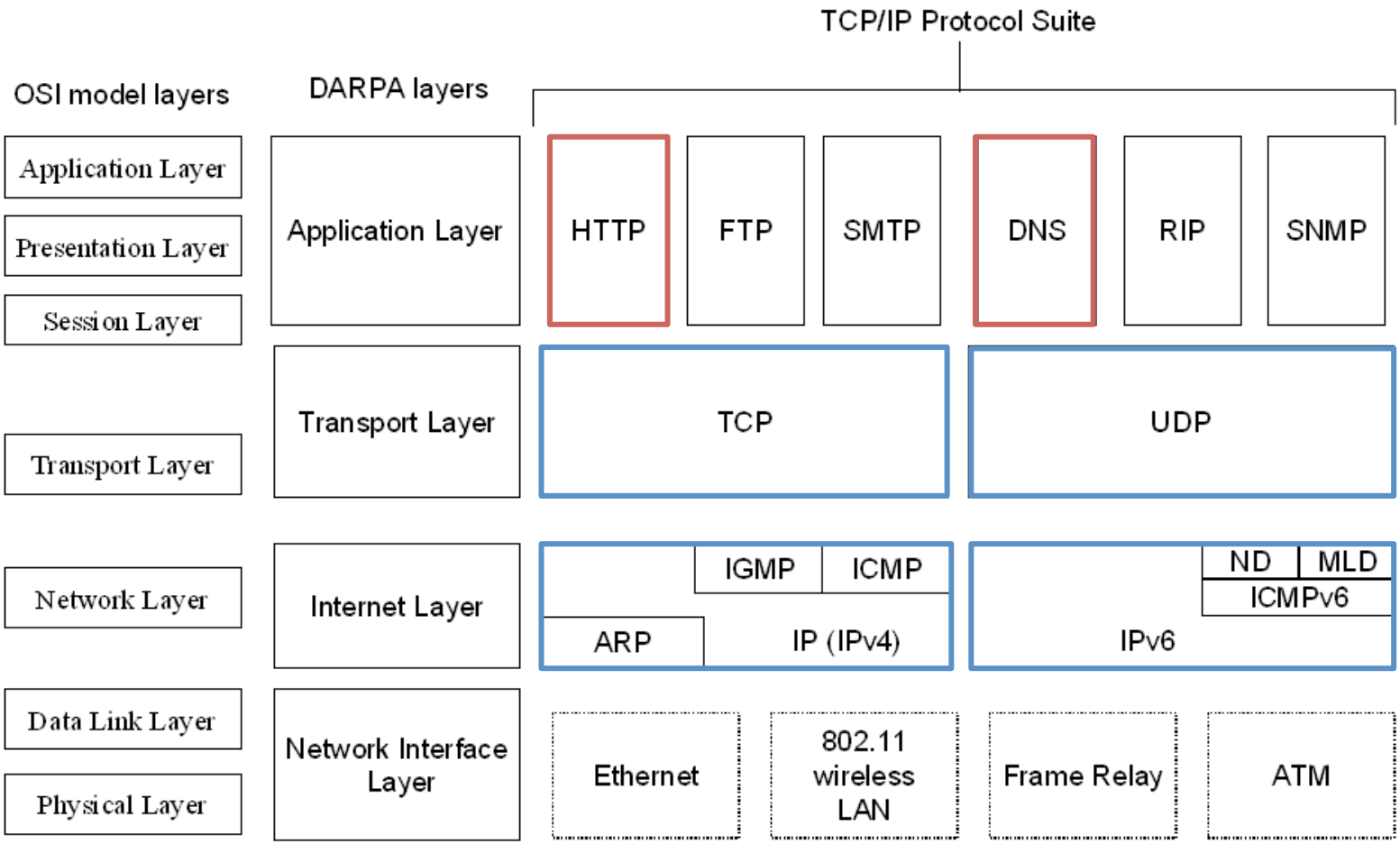
Imperial College London

- Participant
  - Capabilities: send and receive legitimate packets that respect the protocol
  - Examples: web browser, web application
- Eavesdropper
  - Capabilities: read packets sent to others, cannot (or will not) participate
  - Examples: wiretapper, sniffer on a broadcast network (WiFi)
- Off-path
  - Capabilities: participate; create arbitrary packets
  - Examples: machine connected to WiFi, ethernet
- Man in the middle (MITM)
  - Capabilities: participate; read, modify, create or delete packets
  - Example: proxy, ISP, router, WiFi access point



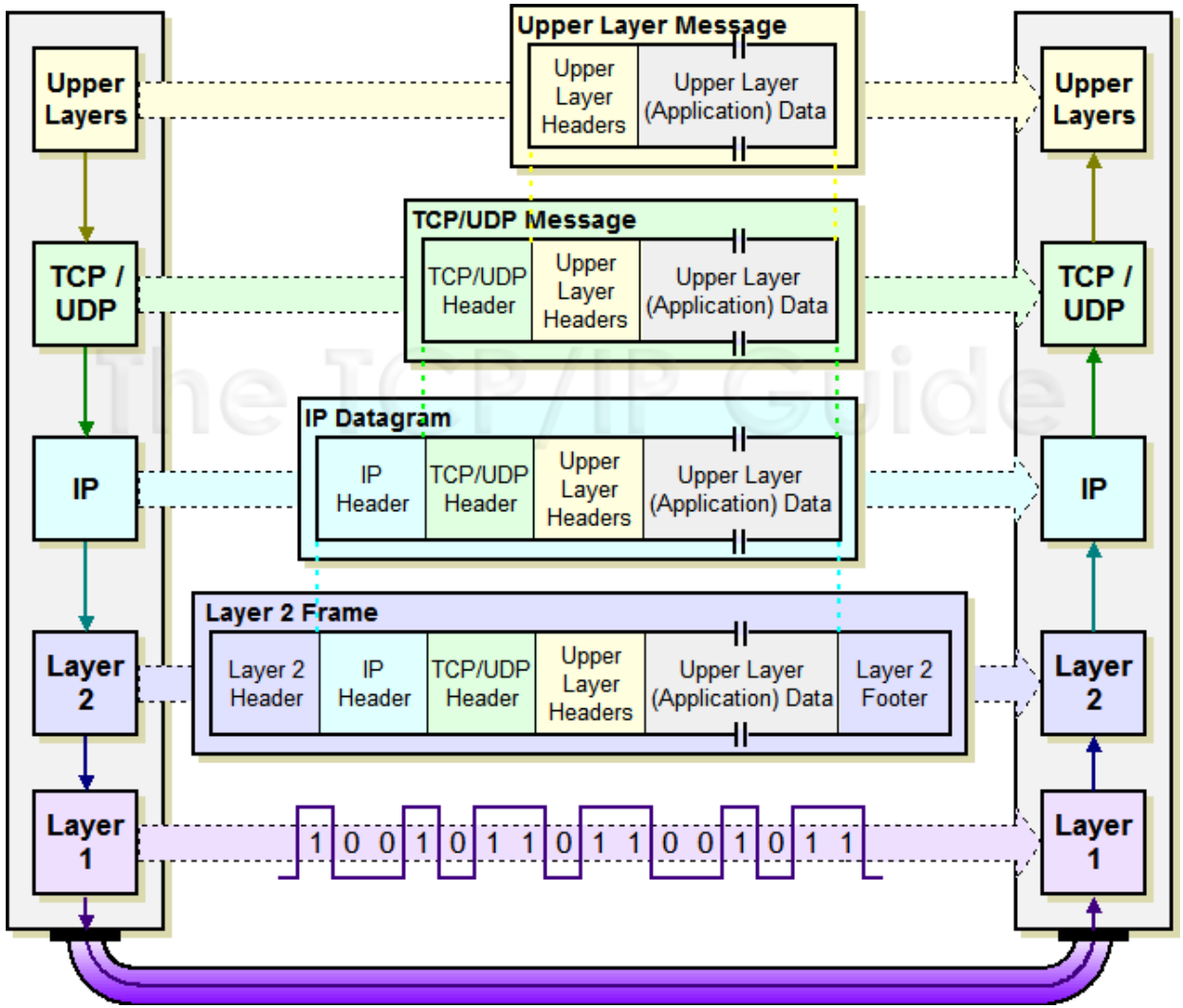
Layers and protocols

Imperial College London



Datagram encapsulation

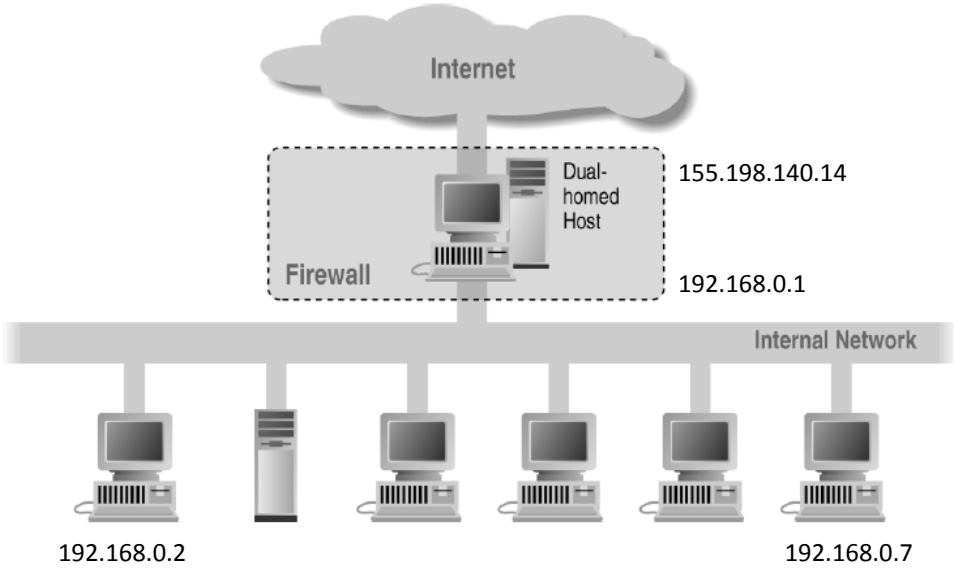
Imperial College London



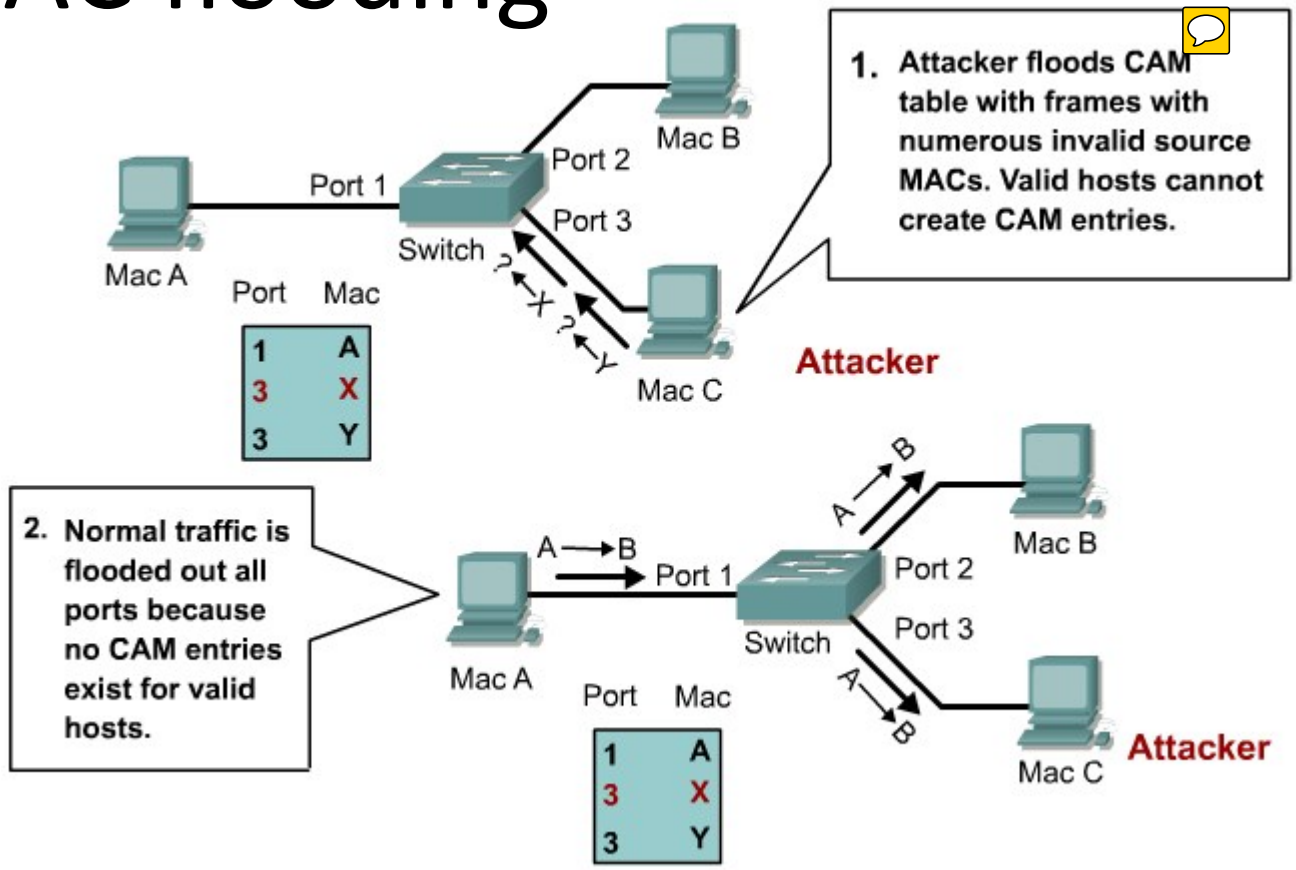
Local Area Network

Imperial College London

- Local Area Network (LAN)
  - Typically based on a broadcast medium, such as cable (Ethernet) or wireless (WiFi)
  - The network interface of each host has a Media Access Control (MAC) address
  - The Address Resolution Protocol (ARP) assigns IP addresses to MAC addresses
    - IP range 192.168.0-255.0-255 is reserved for private networks
  - The Dynamic Host Configuration Protocol (DHCP) tells new hosts their IP and other configuration information
- LAN relies on broadcast medium
  - Conflict resolution requirements prescribe a minimum packet size
  - If padding data is not initialized this may lead to data disclosure
  - Similar to the OpenSSL *Heartbleed* vulnerability



MAC flooding

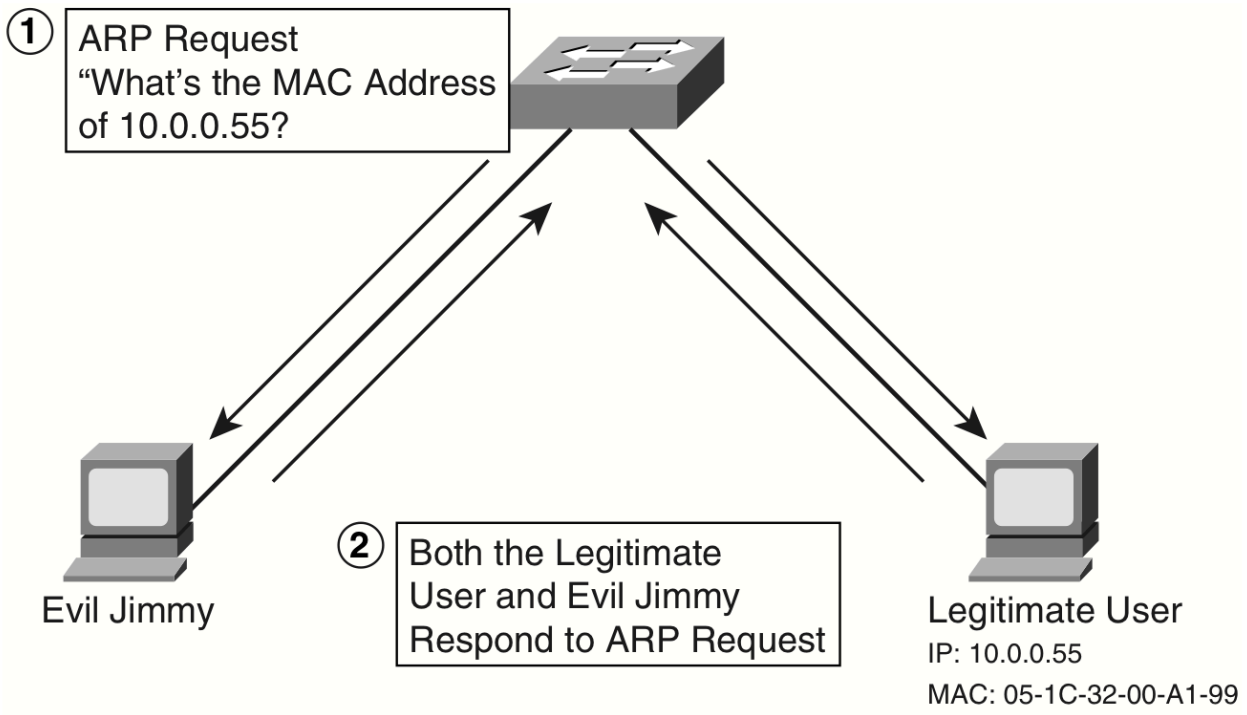


- Network switches cache Port-MAC associations
- Attacker forces switch to broadcast traffic, so he can sniff packets
- Typical countermeasures
  - "Port security": limit ability to flood caches
  - Keep track of authorised MAC addresses in the system

ARP poisoning

Imperial College London

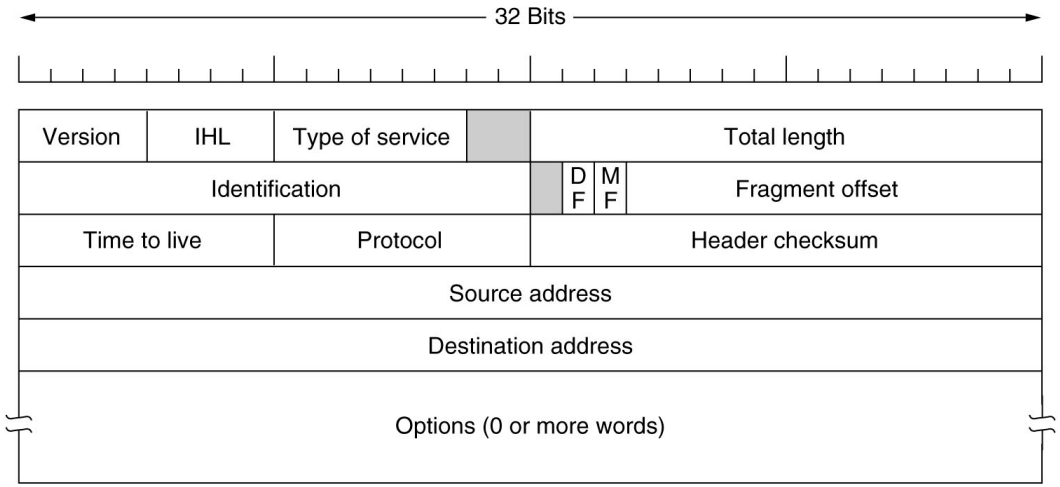
- MAC is easy to spoof: feature to deal with conflicting hardware
  - Attacker can evade MAC-based filtering and access control
  - Off-path attacker spoofing router becomes MITM!
- ARP poisoning
  - Switch needs to find MAC corresponding to an IP
  - Attacker spoofs MAC of victim and replies, like victim does
  - Message is forwarded to both ports that replied (including Evil Jimmy's)
  - Typical countermeasure: static ARP rules or spoofed ARP message detection



Internet Protocol

Imperial College London

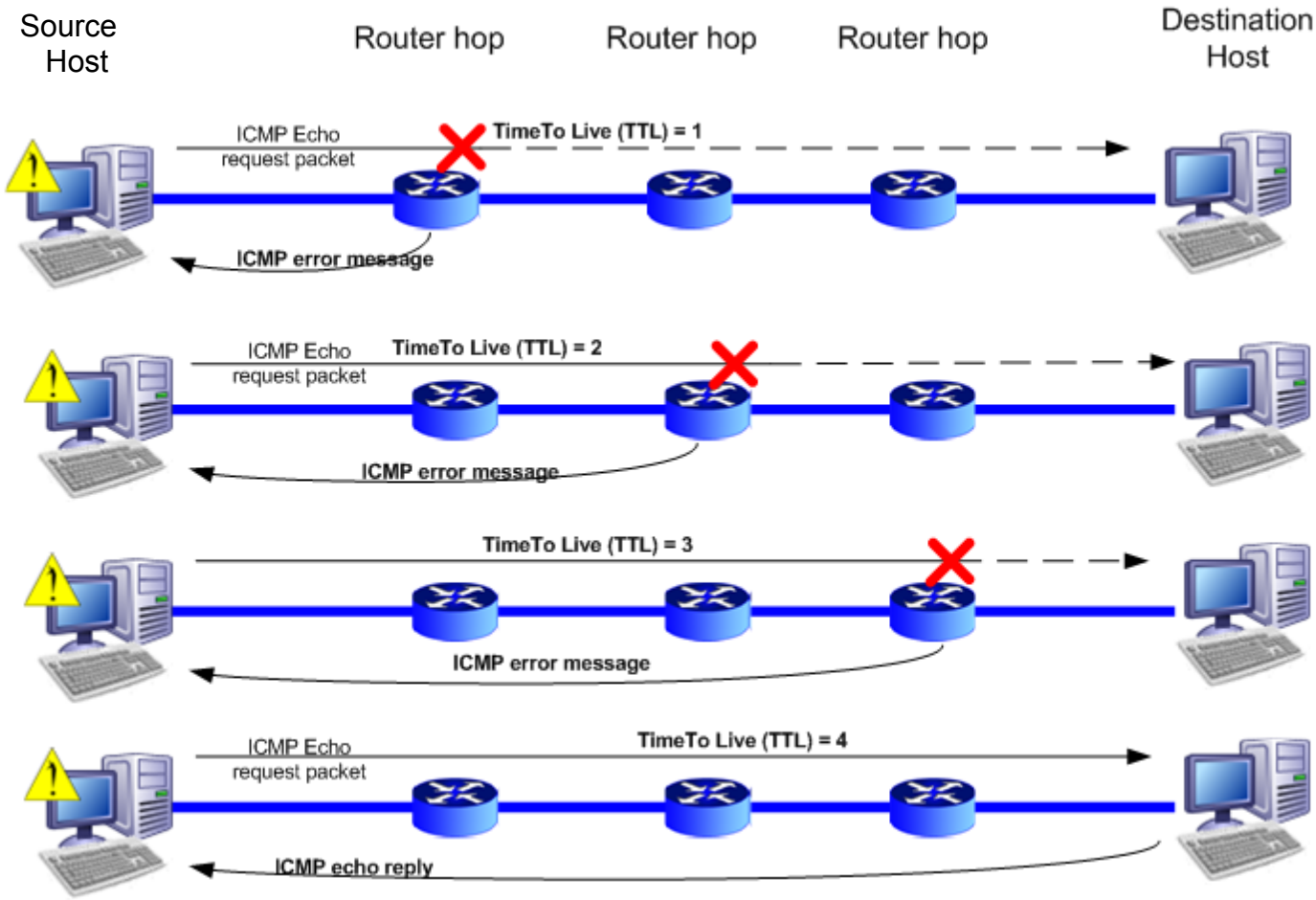
- The IP protocol delivers packets between *Source* and *Destination* hosts
- Structure of IP addresses is hierarchical and guides routing
- Protocol is “best effort”: may drop or reorder packets
- IP packet can be fragmented when it transits on networks with smaller packet sizes
  - **D**on’t **F**ragment, **M**ore **F**ragments flags indicate fragment type
  - *Fragment offset* gives position of fragment in original packet
  - *Identification* differentiates fragments for different packets
  - Various OSs treat duplicate IP fragments in different ways: used for OS fingerprinting
- *Time to live* (TTL) is used to discard packets that take too many steps to reach destination
  - TTL is decremented at each step (“hop”) in the network until it reaches 0
  - At 0, packet is discarded and ICMP error message is sent to source



Traceroute

Imperial College London

- Traceroute algorithm uses TTL to identify hosts/routers on path to target
  - Send packets increasing TTL (1, then 2, then 3, …) until destination is reached
  - Each ICMP error message should be from a host on the path to the destination



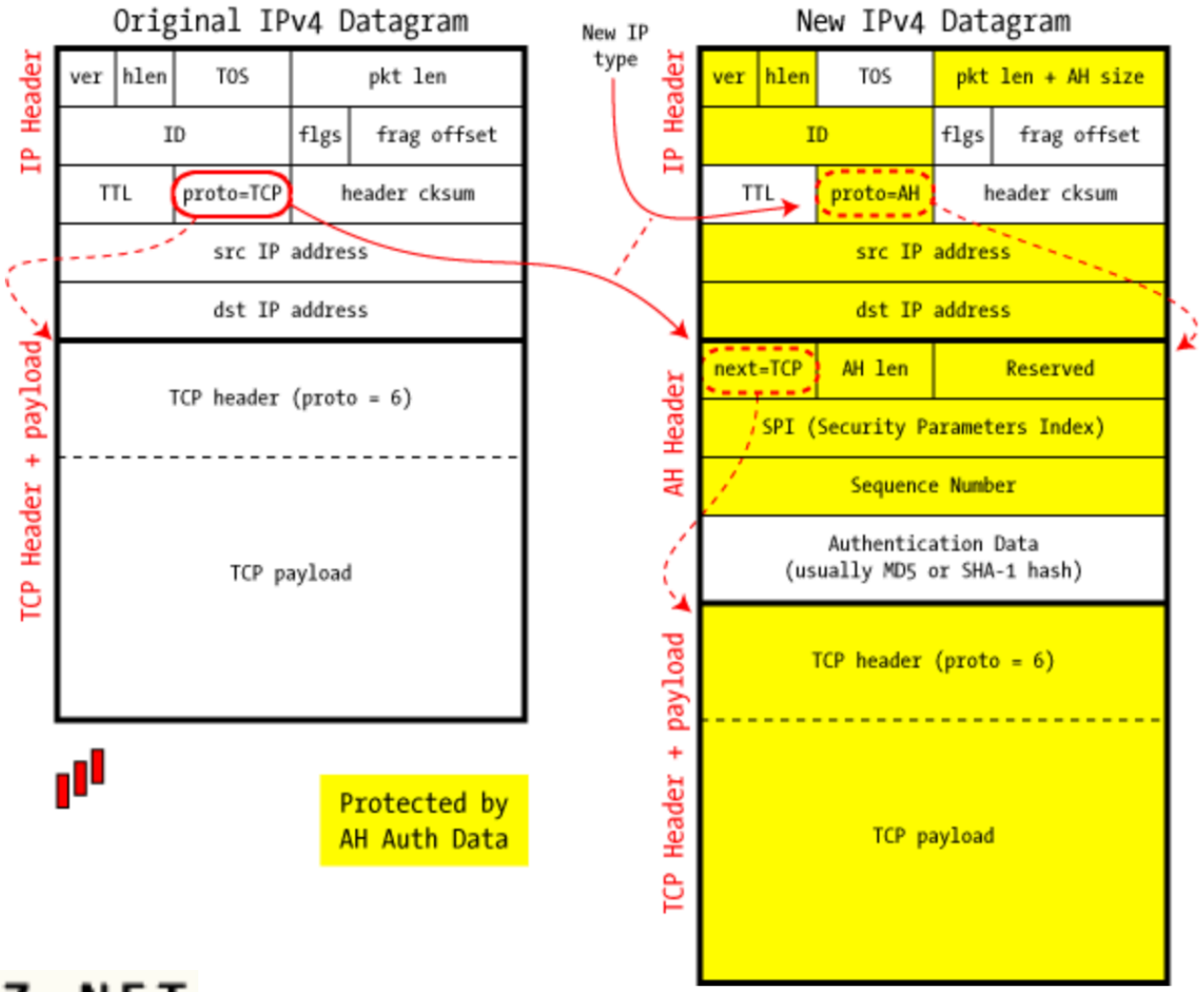
IP security

Imperial College London

- The source IP is not authenticated and easy to spoof
  - Off-path attacker can send packets with target IP as source
    - The target will receive response
    - Used for attacks including idle scanning, DDoS
  - Hard to trace back malware infections
- Packets travel through untrusted hosts
  - MITM attacker can directly read packets and modify payload
- *IPsec* adds security to IP protocol
  - Authentication Header (AH)
    - Authentication and integrity of whole packet
      - Does not interoperate with NAT
    - Allows packet inspection, not blocked by firewalls
  - Encapsulating Security Payload (ESP)
    - Confidentiality of payload
    - Optional authentication
  - Transport mode: protects the IP payload only
  - Tunnel mode: protects also the IP header
  - Limitations
    - Hard to configure correctly
    - Hard to connect to the application layer
  - Used mostly in tunnel mode, for VPNs

IPsec in AH Transport mode

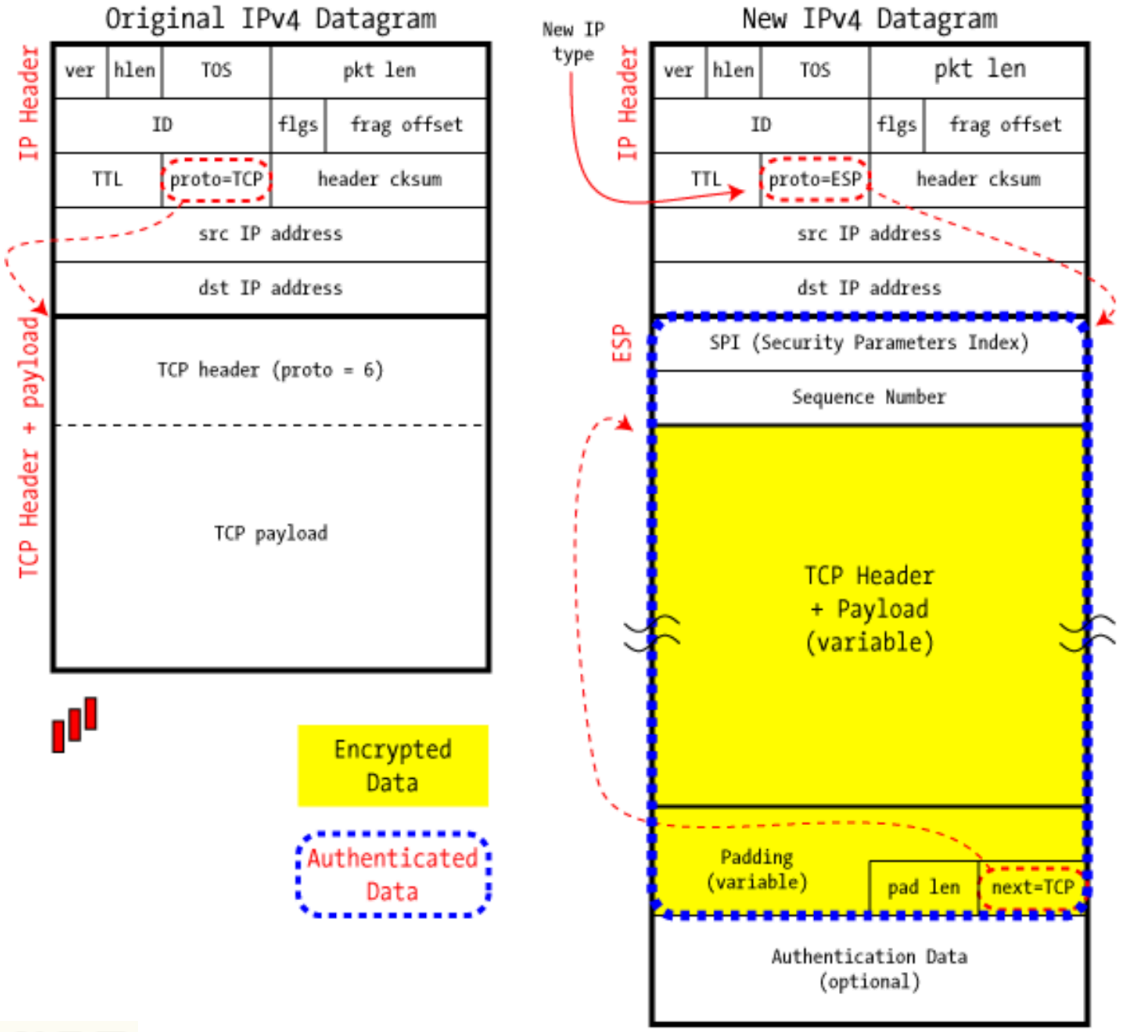
Imperial College London



UNIXWIZ.NET

IPsec in ESP Transport mode

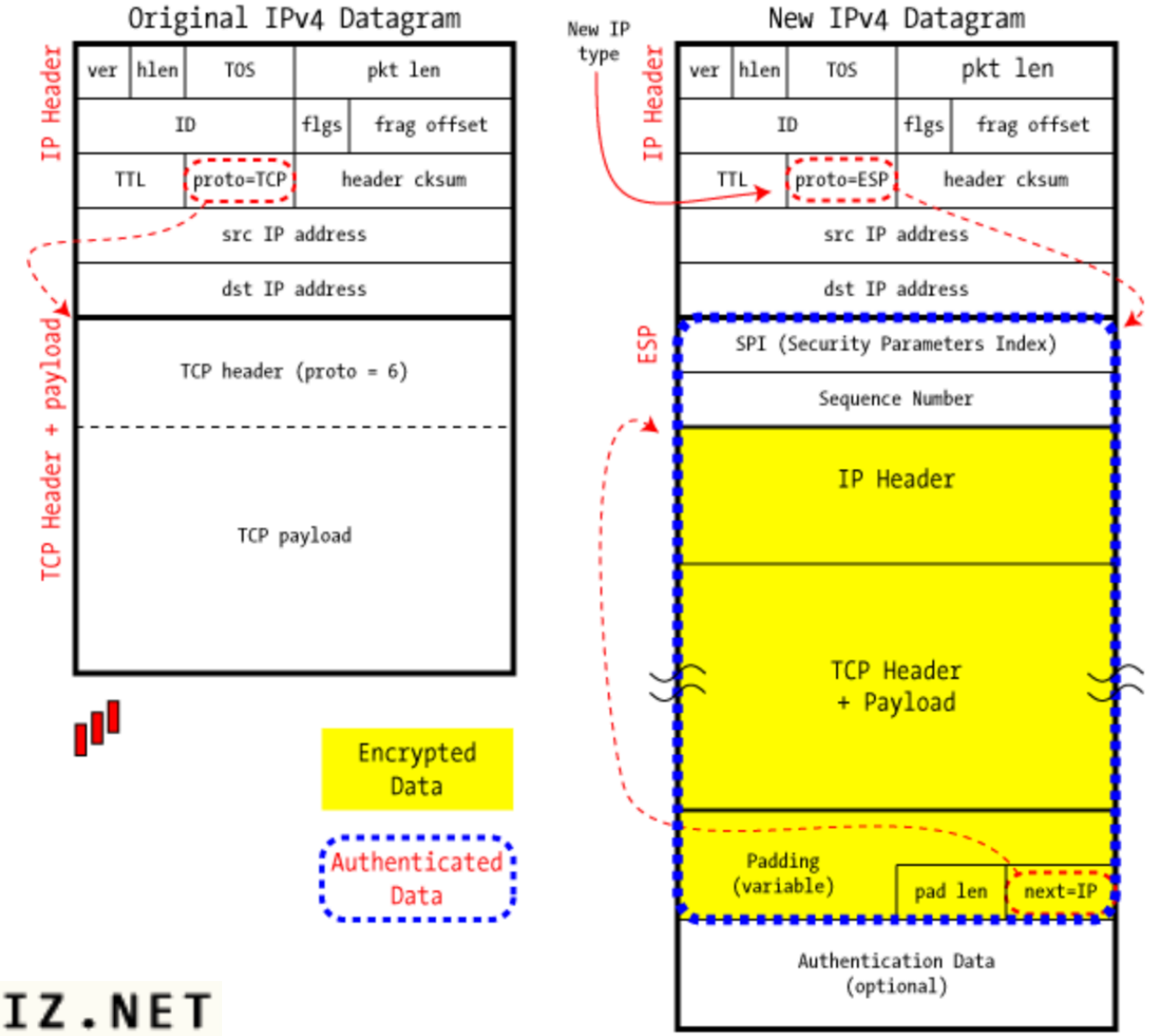
Imperial College London



UNIXWIZ.NET

IPsec in ESP Tunnel mode

Imperial College London

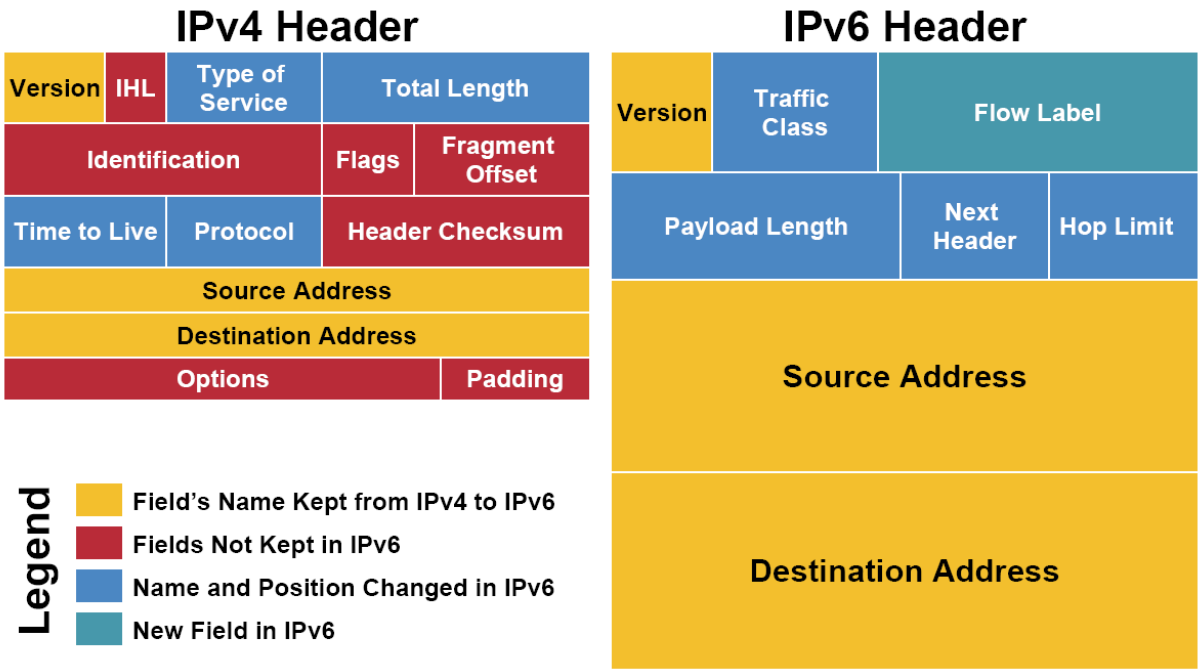


UNIXWIZ.NET

IPv6

Imperial College London

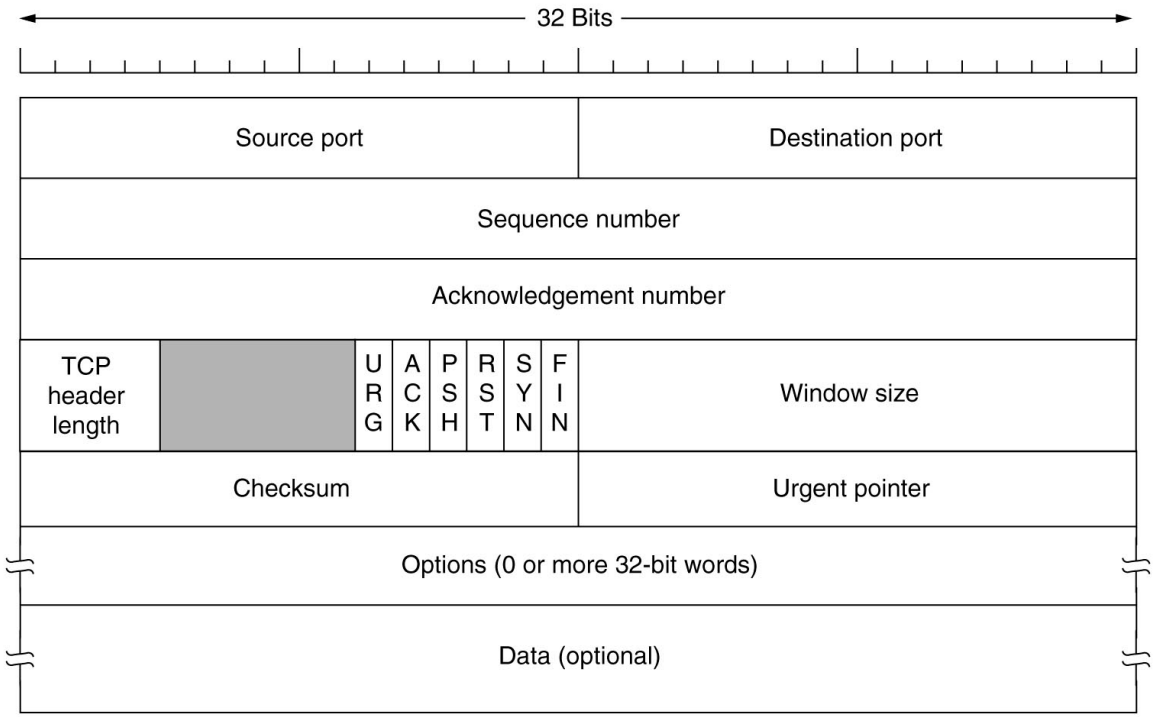
- IPv4: 155.198.140.14 is 32 bits (AFRINIC running out of space by mid 2018?)
- IPv6: 2001:0000:3238:DFE1:EA06:88FF:FECB:AF19 is 128 bits (more than enough!)
- Comparison with IPv4
  - Address space is large enough that there is no more need for NAT
  - Fewer headers, enables better routing
- IPv6 includes IPsec natively
  - Will work more efficiently (packet size) and protect more (headers)
  - Up to the network rather than application layer to use IPsec, so applications should not rely on it



Transmission Control Protocol

Imperial College London

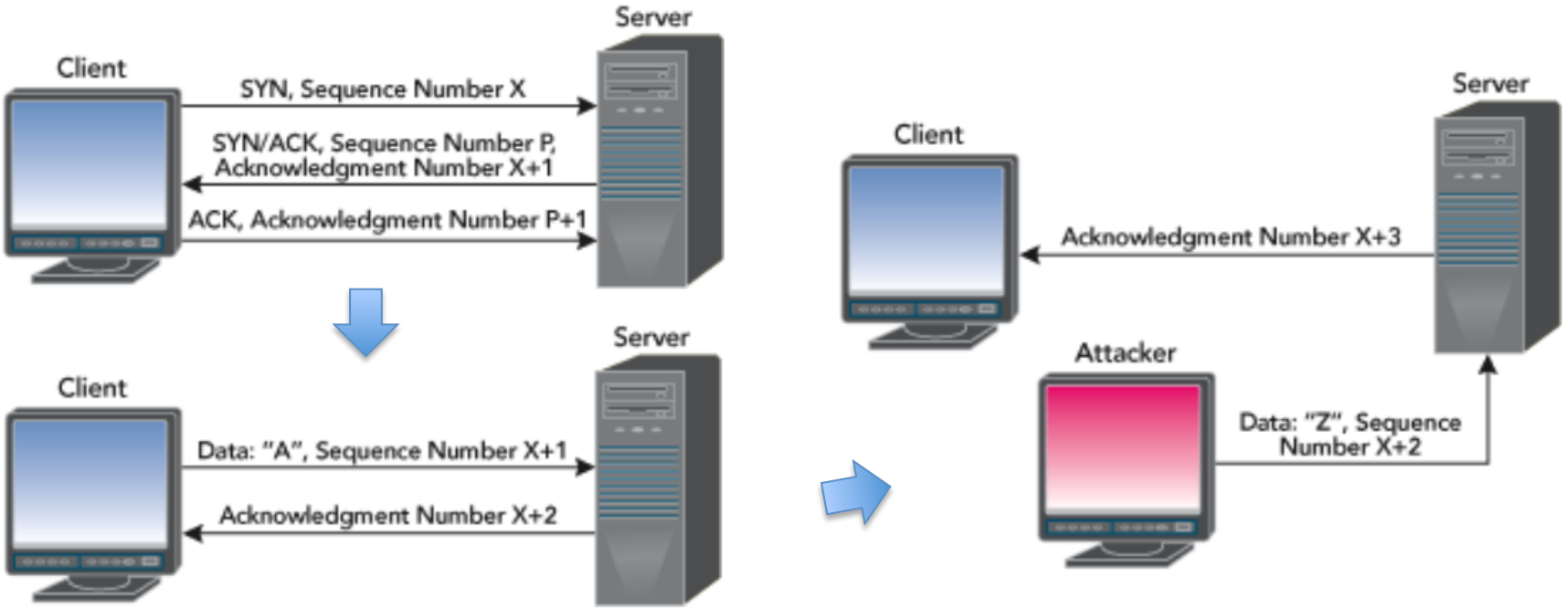
- TCP establishes a reliable connection to a service on the target destination port
  - Source port is chosen at random by OS, to receive responses
  - TCP adds sequence numbers and re-requests lost packets
  - Everything is delivered to the application, and in the right order
- TCP is the channel used to send HTTP data



TCP security issues

Imperial College London

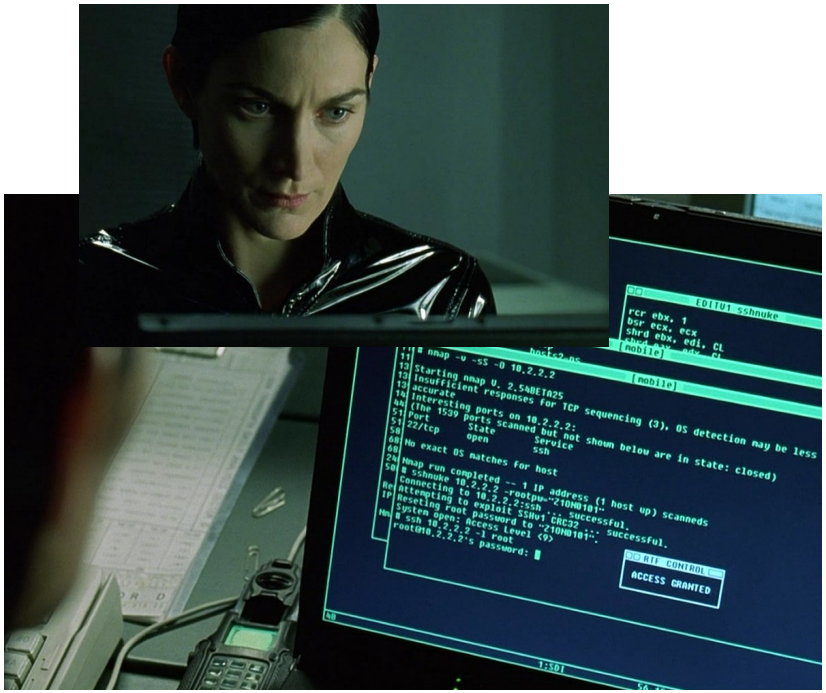
- TCP state easily accessible
- Sequence numbers are predictable: previous number + bytes exchanged
- MITM attacker can read current sequence number and inject new packets
  - TCP session hijacking
- Off-path attacker can try and guess the right sequence number
  - Blind spoofing attack
  - See recommended reading: *Off-Path Hacking*
- Typical countermeasures
  - Time-delay, and discard race-condition packers
  - Use IDS, HTTPS



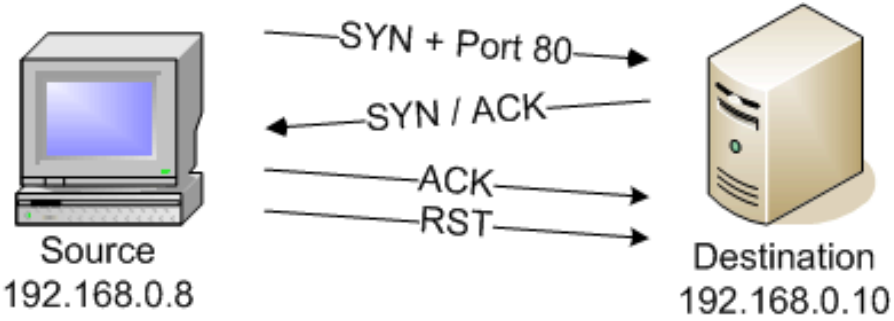
Port scanning

Imperial College London

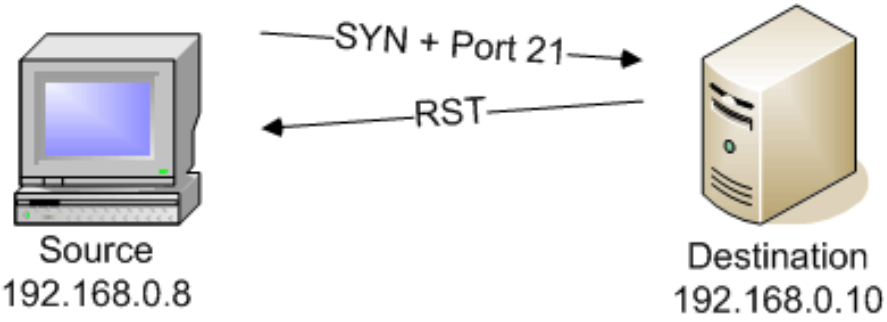
- (Unauthorised) port scanning **is a crime**
  - We will practice safely in the lab with Nmap
- Different services/protocols require different kind of scans
- Examples
  - TCP `connect()` scan (below)
  - TCP idle scan (next slide)



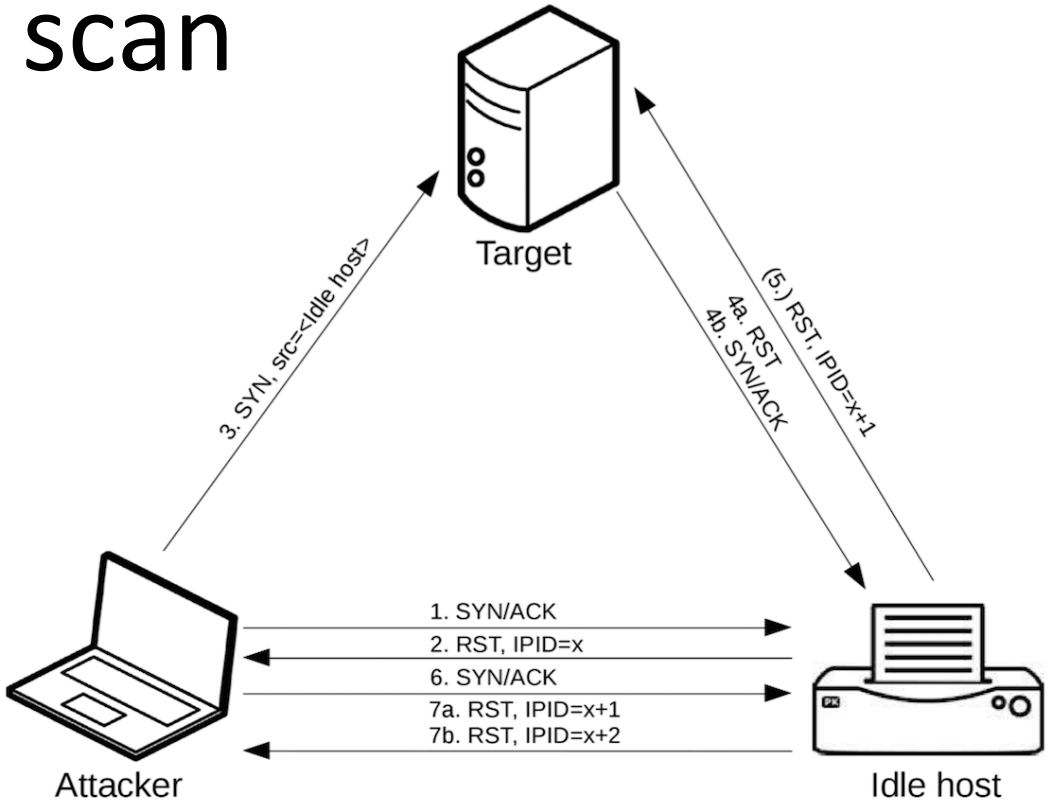
a) HTTP port is open



b) FTP port is closed



TCP idle scan

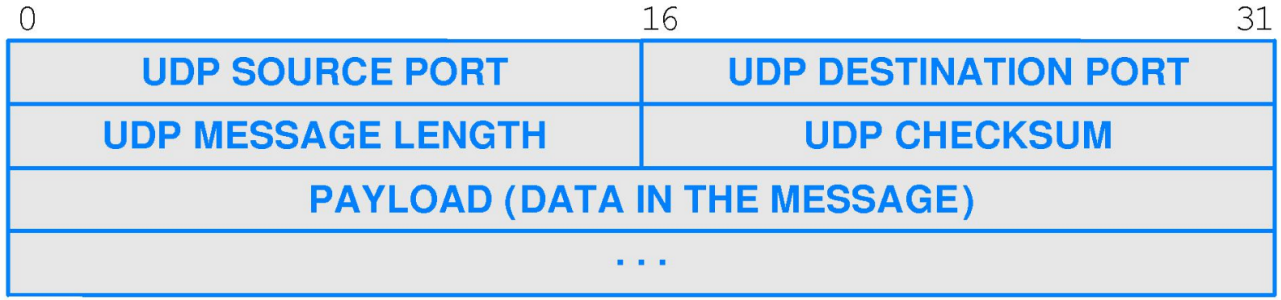


0. Identify an idle host (printer, ftp server, …)
1. & 2. Check available IPID on idle host
3. Spoof source IP when connecting to target
4. Idle host receives scan result it is not waiting for
5. 4a (closed): no response; 4b (open): RST packet and IPID increment
6. & 7. Check new available IPID on idle host

User Datagram Protocol

Imperial College London

- *Connectionless protocol*
  - Low overhead, low latency: faster than TCP
  - Can be used for broadcasting, multicasting packets
  - Up to the application layer to make sense of a stream of UDP packets
    - No guarantee data reaches destination: routers may drop UDP packets if there's a conflict
    - No integrity: checksum is optional
    - Packets may be received in different order than they are sent, receiver may also get duplicates
- Usage
  - Streaming media: voice and video
    - WebRTC (RTP) can use UDP or TCP
  - Network management: SNMP, DHCP, DNS
  - Applications that need low-level control on network packets



UDP scans

Imperial College London

- Many different protocols may be run on top of UDP
- Send generic UDP header with no payload to target ports
- If you receive a UDP response, the port is **open**
- If you receive an ICMP error the port is **closed** or **filtered** by a firewall
- If you timeout without a response
  - The port may be **open** and host a service that drops ill-formed packets
  - The port may be **filtered** by a firewall
  - Probe the port again using UDP packets with protocol-specific payloads
- UPD services are harder to scan
  - More time consuming
    - Timeouts for lack of response
    - May take several attempts to resolve **open|filtered** ports
  - Less precise
    - Some UDP custom protocols simply can't be probed

Key TCP/IP threats

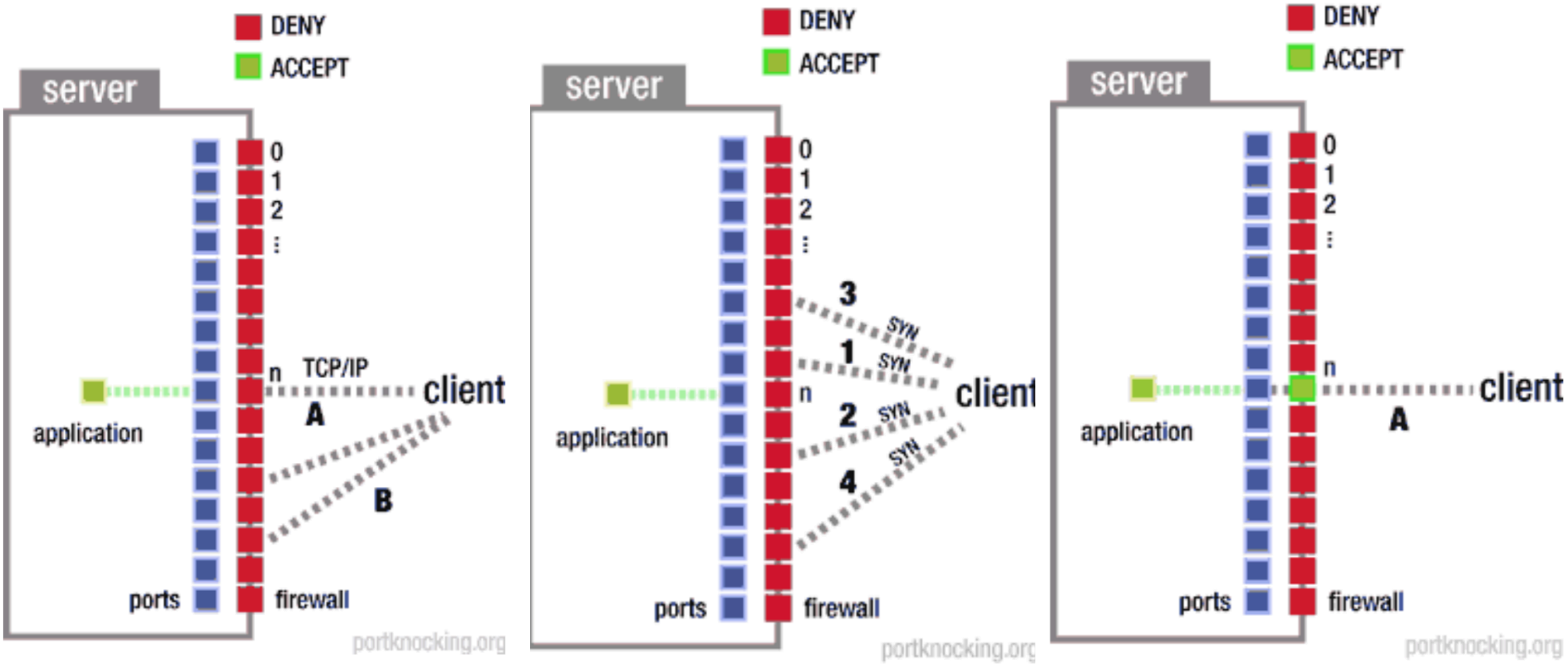
Imperial College London

- Host and port scanning
  - Used by hackers during active information gathering
  - They will try to hide the requests within the normal variance of network traffic
- Port sweep
  - One attacker looks for a specific service on many machines
  - More sensitive than port scanning: likely that service is vulnerable (0-day, unpatched)
- Malicious traffic
  - Targeted attacks via network connections
  - Exploitation of networking stack implementation
  - Exfiltration of data
- Distributed Denial of Service (DDoS)
  - Flood a target with extremely high volume of network traffic
  - Attacker can use a botnet
    - Achieve large volume
    - Diversify behaviour to avoid detection
    - Spread attack traffic to preventing takedown

Port knocking

Imperial College London

- Technique to hide a service from port scanning
  1. Sequential or random scan only finds closed ports
  2. Client shares a secret with server that identifies specific ports to probe in a fixed order (3,1,2,4); server replies to last probe (4) with random port (n) where the service will be provided
  3. Client connect to service on the random port
- Can be used legitimately, or to hide backdoors



Network defenses

Imperial College London

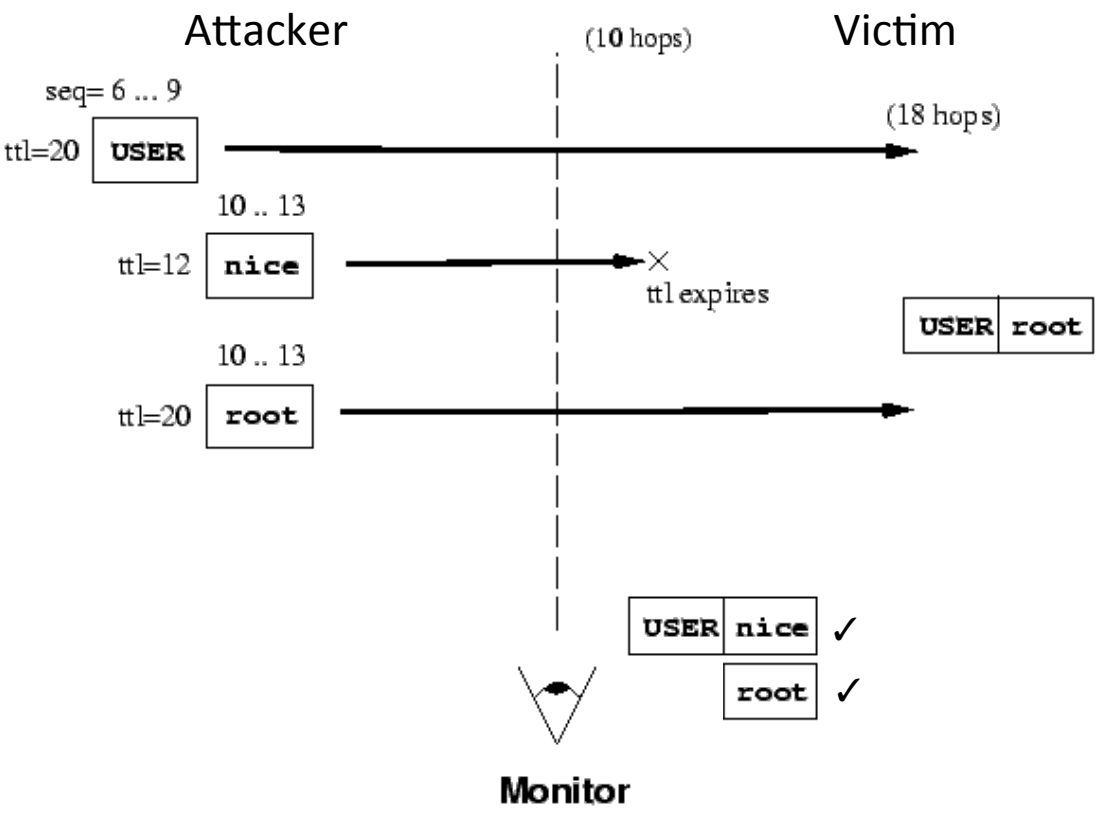
- Firewalls
  - Open source examples: `iptables` (Linux), `pf` (OpenBSD, OSX)
  - Beyond packet filtering, firewalls can keep state, inspect application-layer packets
  - Can protect individual machine from port scanning, malicious traffic
  - Needs kernel-level operations: critical to defend from compromise
  - No help against DDoS or port sweep
- Intrusion Detection Systems
  - Free/open source examples: Snort, Bro
  - Dedicated hardware that inspects network traffic (on- or off-path)
    - Signature based
    - Anomaly detection based (also machine learning techniques)
  - Can detect host and port scanning, port sweep
  - Can filter malicious traffic but must be fast (hence miss attacks)
  - Can rate-limit connections to mitigate DDOS
    - But the source IP is easy to spoof: risk of blocking too many IPs
- Variants: IPS, egress filters, etc..

IDS evasion

Imperial College London

Example of attack against IDS using IP fragmentation:

1. Fragment a suspicious IP packet in 2
2. Traceroute to determine distance to IDS and target
3. Send frag 1 to reach target
4. Send innocuous replacement of frag 2 so that it's seen by the IDS but expires before reaching the target
5. IDS decides that communication is safe
6. Send malicious frag 2 so that it reaches the target
7. IDS does not interpret message from (6) as related to the one in (3)



(*A System for Detecting Network Intruders in Real-Time*, V. Paxson)