ARP
Address Resolution Protocol

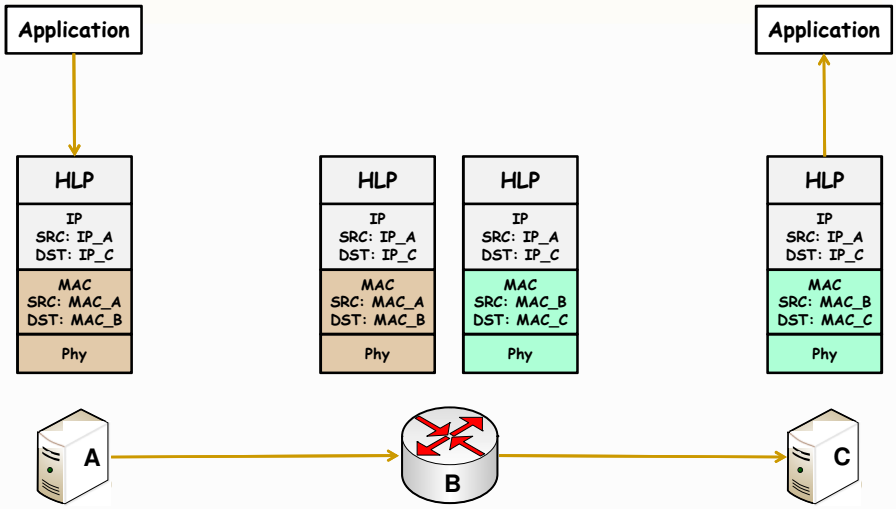
Networking Basics

- Communication in packet networks rely on several layers, with different identifiers
  - Applications use transport (TCP/UDP) ports
  - Hosts use network (IP) addresses
  - Interface Cards use MAC addresses
- Communication is typically made between applications using tuples
  - <IP_Address:Port> and a protocol (TCP, UDP, etc.)

Networking Basics

- When a packet is to be routed, two situations may occur:
  - The destination host is in the same IP network
    - The packet is sent directly to the destination host
  - The destination host is in another IP network
    - The packet is forwarded to a next hop (gateway)
- In both cases, the packet is transmitted between physical interfaces
  - Destination host or gateway

Networking Basics



Networking Basics

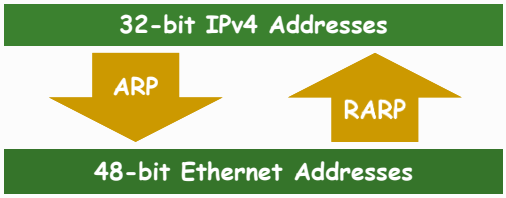
- IP addresses do not change between source and destination
  - End-to-end addressing
- MAC addresses are valid for a single network segment
  - When a packet is routed, the MAC address of the next hop must be found

IP to MAC mapping

- Static configuration
  - MAC entries of all hosts configured statically
    - All hosts "know" the MAC address of all interfaces of all other hosts
  - Doesn't scale!
    - Changing a single interface requires updating all other hosts
- Dynamic configuration
  - ARP (Address Resolution Protocol)

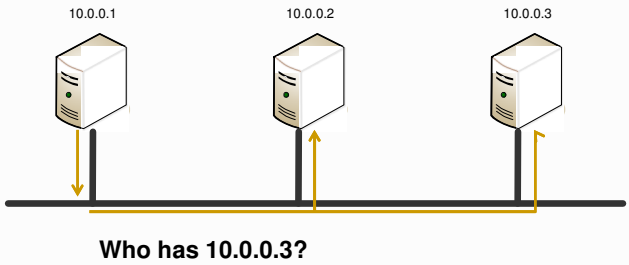
Address Resolution Protocol (RFC 826)

- ARP
  - Find the MAC address of an interface which is in a host with a given IP address
- RARP
  - Finds the IP address of host having an interface with a given MAC



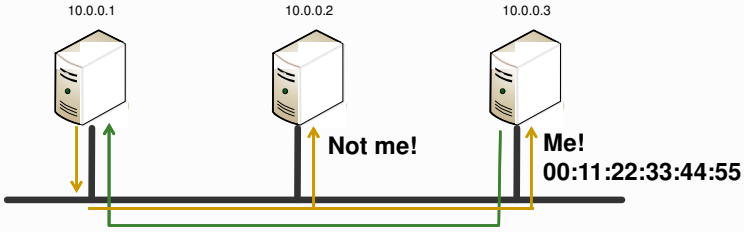
Address Resolution Protocol

- Send ARP Request using broadcast



Address Resolution Protocol

- Reply using ARP Response using unicast



Address Resolution Protocol

- Every packet sent requires two MAC address
  - Source Address is known
  - Destination Address must be determined
- ARP Cache increases performance
  - Caches both known and unknown entries
  - Avoids repeating the discovery process per packet
  - Entries have a large lifetime
    - 2 minutes

ARP Cache

```
$ arp -a
fog.av.it.pt      (193.136.92.154) at 00:1e:8c:3e:6a:a6 [ether] on eth0
atnog.av.it.pt    (193.136.92.123) at 00:15:17:e6:6f:67 [ether] on eth0
guarani.av.it.pt (193.136.92.134) at 00:0c:6e:da:19:87 [ether] on eth0
aeolus.av.it.pt   (193.136.92.136) at bc:ae:c5:1d:c6:53 [ether] on eth0
```


ARP Spoofing

- MAC addresses can be modified
  ifconfig eth0 hw ether 00:11:22:33:44:55
- Using a colliding MAC address will allow the reception of network traffic for other hosts
  - Some switches limit MAC addresses to single ports
- Sending ARP packets with spoofed addresses may poison the cache of other stations
  - ARP Poisoning

ARP Poisoning

- Hosts cache information directly from all packets received
  - Besides ARP packets
  - No other verification is done
- New information will replace existing entries
  - Great for allowing network dynamism
  - Very bad for security
- It is possible to send specially crafted packets to create specific entries in remote hosts

ARP Poisoning

- When receiving an ARP Request:

```
▷ Frame 10: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▷ Ethernet II, Src: Apple_1b:1f:42 (e0:f8:47:1b:1f:42), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▽ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Apple_1b:1f:42 (e0:f8:47:1b:1f:42)
    Sender IP address: 10.0.0.3 (10.0.0.3)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.0.2 (10.0.0.2)
```

- 10.0.0.2 will send an ARP Response
- But… 10.0.0.2 will also "learn" that 10.0.0.3 is at e0:f8:47:1b:1f:42

ARP Poisoning

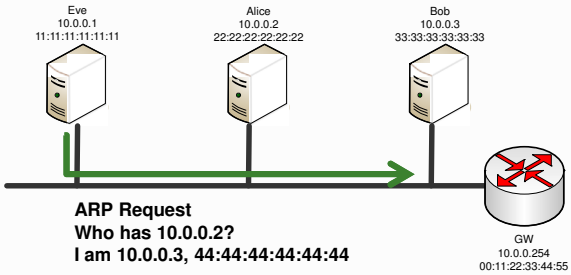
- When receiving an ARP Response

```
▸ Frame 123: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: Tp-LinkT_f2:77:62 (90:f6:52:f2:77:62), Dst: Apple_1b:1f:42 (e0:f8:47:1b:1f:42)
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Tp-LinkT_f2:77:62 (90:f6:52:f2:77:62)
    Sender IP address: 10.0.0.246 (10.0.0.246)
    Target MAC address: Apple_1b:1f:42 (e0:f8:47:1b:1f:42)
    Target IP address: 10.0.0.3 (10.0.0.3)
```

- 10.0.0.3 will learn that 10.0.0.246 is at 90:f6:52:f2:77:62
- …. even if no matching request has been made…
  - Gratuitous ARP

ARP Poisoning: Consequences

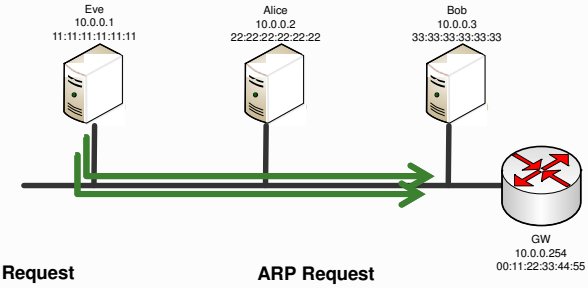
- Hosts can be isolated from the network
  - Create fake entries for all other hosts



- Alice will use 44:44:44:44:44:44 when talking to Bob

ARP Poisoning: Consequences

- Hosts can be denied communication with the outside world

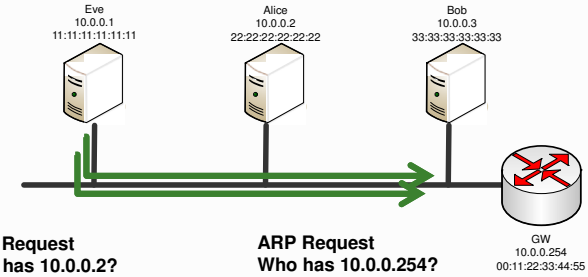


ARP Request
Who has 10.0.0.2?
I am 10.0.0.254, 44:44:44:44:44:44

ARP Request
Who has 10.0.0.254?
I am 10.0.0.2, 44:44:44:44:44:44

ARP Poisoning: Consequences

- Interception of traffic between hosts (MitM)



ARP Request
Who has 10.0.0.2?
I am 10.0.0.254, 11:11:11:11:11:11

ARP Request
Who has 10.0.0.254?
I am 10.0.0.2, 11:11:11:11:11:11

- Then Eve will forward traffic

ARP Poisoning: Avoidance

- Static entries
  - No resolution process is triggered
  - Colliding information from ARP packets is discarded
- Port-based packet filtering at switch ingress
  - Spoofed ARP packets are dropped
  - Only possible in static scenarios
- Network segregation
  - VLANs, WiFi client segregation

ARP Poisoning: Avoidance

- Behavior detection w/ monitoring software
  - Detect ARP Responses without Request
  - Detect repeated Requests from same host
  - Detect MAC changes
  - Network administrator is notified
    - But ARP poisoning is not actually avoided!
    - And it may be difficult to find the attacker's host