Security in Informatics and Communications (2025/2026)

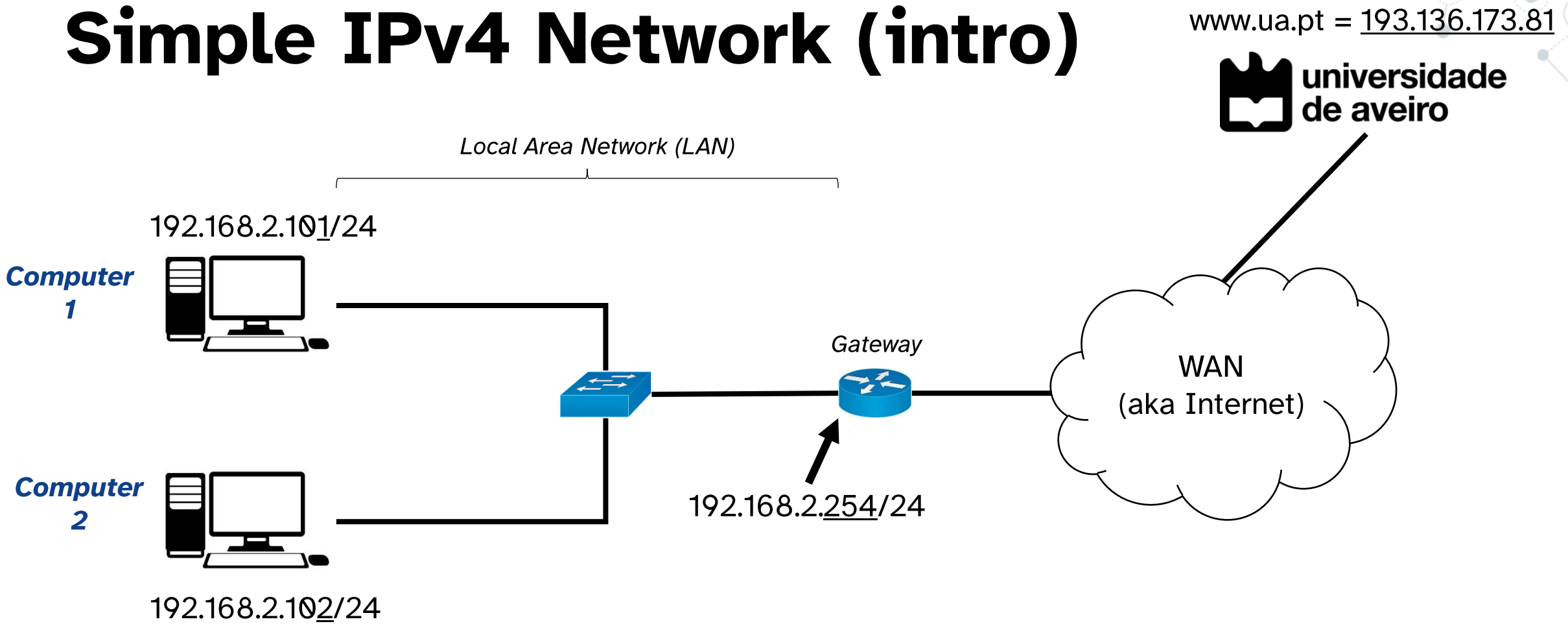
Practical Lesson #2:

Local Network Vulnerabilities

André Zúquete (andre.zuquete@ua.pt)
Vítor Cunha (vitorcunha@ua.pt)

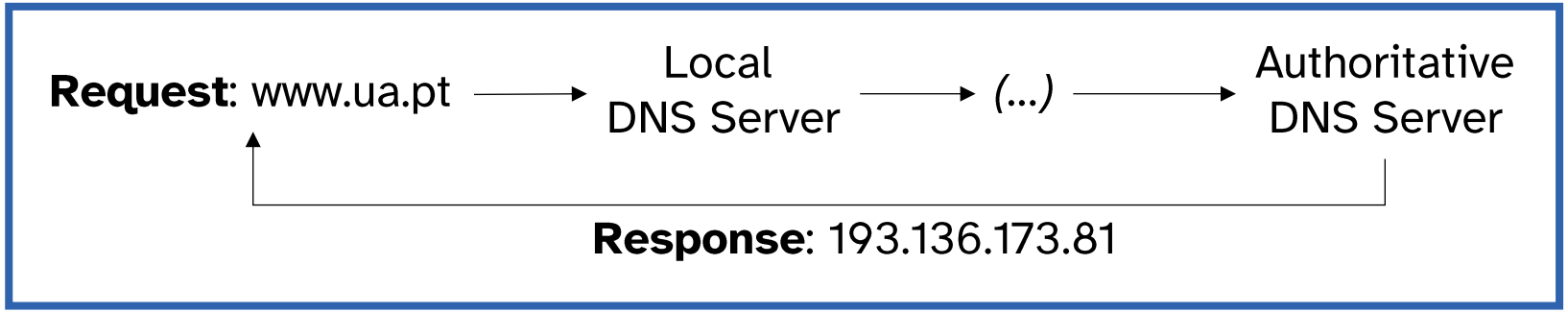
universidade de aveiro
deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

Simple IPv4 Network (intro)



Routing Table

0.0.0.0 via 192.168.2.254
192.168.2.0/24 iface eth0



OSI Model

OSI Model			
Layer		Protocol data unit (PDU)	Function[3]
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4. Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1. Physical	Symbol	Transmission and reception of raw bit streams over a physical medium

Source: https://en.wikipedia.org/wiki/OSI_model

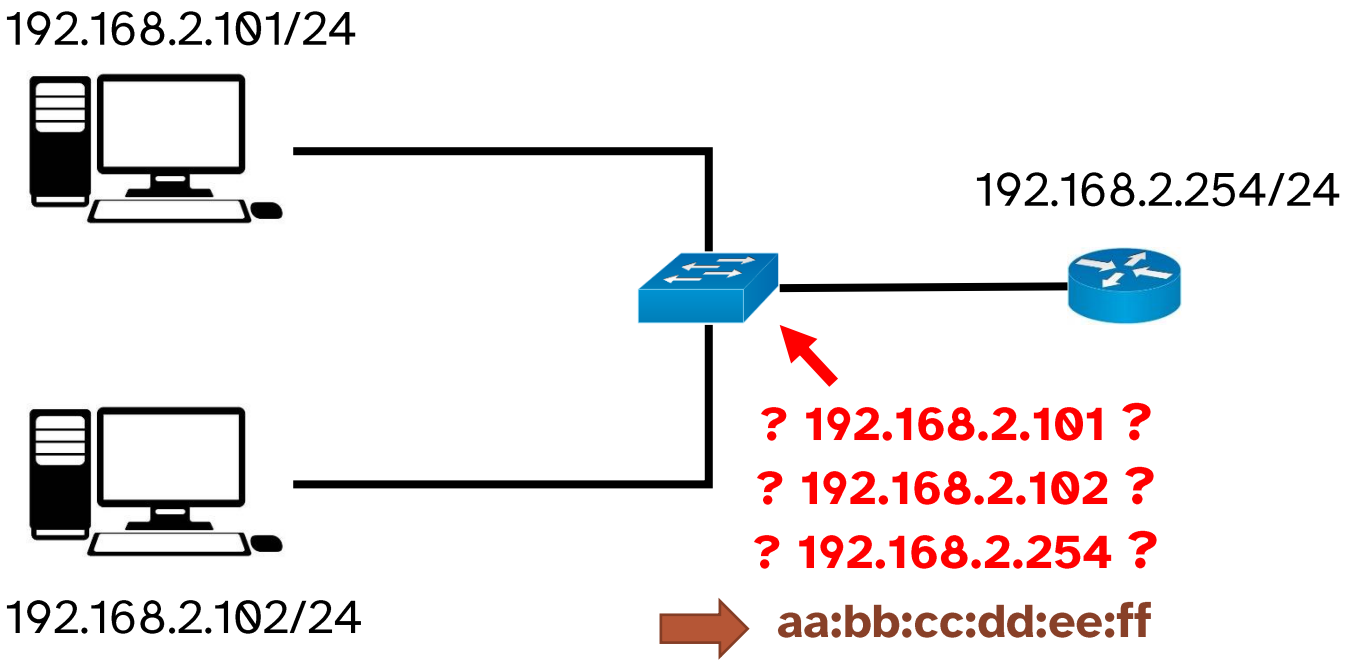
OSI Model

OSI Model			
Layer		Protocol data unit (PDU)	Function[3]
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4. Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1. Physical	Symbol	Transmission and reception of raw bit streams over a physical medium

Today's class!

Source: https://en.wikipedia.org/wiki/OSI_model

Simple IPv4 Network (layer 2)



ARP table

? 192.168.2.102 at aa:bb:cc:dd:ee:ff
? 192.168.2.254 at aa:bb:cc:dd:ee:fe

Request: 192.168.2.254 → *(broadcast)* → **Destination**

I am at: *aa:bb:cc:dd:ee:ff*

Address Resolution Protocol (ARP)

Packet structure [edit]

The Address Resolution Protocol uses a simple message format containing one address resolution request or response. The size of the ARP message depends on the link layer and network layer address sizes. The message header specifies the types of network in use at each layer as well as the size of addresses of each. The message header is completed with the operation code for request (1) and reply (2). The payload of the packet consists of four addresses, the hardware and protocol address of the sender and receiver hosts.

The principal packet structure of ARP packets is shown in the following table which illustrates the case of IPv4 networks running on Ethernet. In this scenario, the packet has 48-bit fields for the sender hardware address (SHA) and target hardware address (THA), and 32-bit fields for the corresponding sender and target protocol addresses (SPA and TPA). The ARP packet size in this case is 28 bytes.

Hardware type (HTYPE)

This field specifies the network link protocol type. Example: Ethernet is 1.

Protocol type (PTYPE)

This field specifies the internetwork protocol for which the ARP request is intended. For IPv4, this has the value 0x0800. The permitted PTYPE values share a numbering space with those for EtherType.[3][4][5]

Hardware length (HLEN)

Length (in octets) of a hardware address. Ethernet addresses size is 6.

Protocol length (PLEN)

Length (in octets) of addresses used in the upper layer protocol. (The upper layer protocol specified in PTYPE.) IPv4 address size is 4.

Operation

Specifies the operation that the sender is performing: 1 for request, 2 for reply.

Sender hardware address (SHA)

Media address of the sender. In an ARP request this field is used to indicate the address of the host sending the request. In an ARP reply this field is used to indicate the address of the host that the request was looking for. (Not necessarily address of the host replying as in the case of virtual media.) Switches do not pay attention to this field, particularly in learning MAC addresses. The ARP PDU is encapsulated in Ethernet frame, and that is why Layer 2 devices examine it.

Sender protocol address (SPA)

Internetwork address of the sender.

Target hardware address (THA)

Media address of the intended receiver. In an ARP request this field is ignored. In an ARP reply this field is used to indicate the address of the host that originated the ARP request.

Target protocol address (TPA)

Internetwork address of the intended receiver.

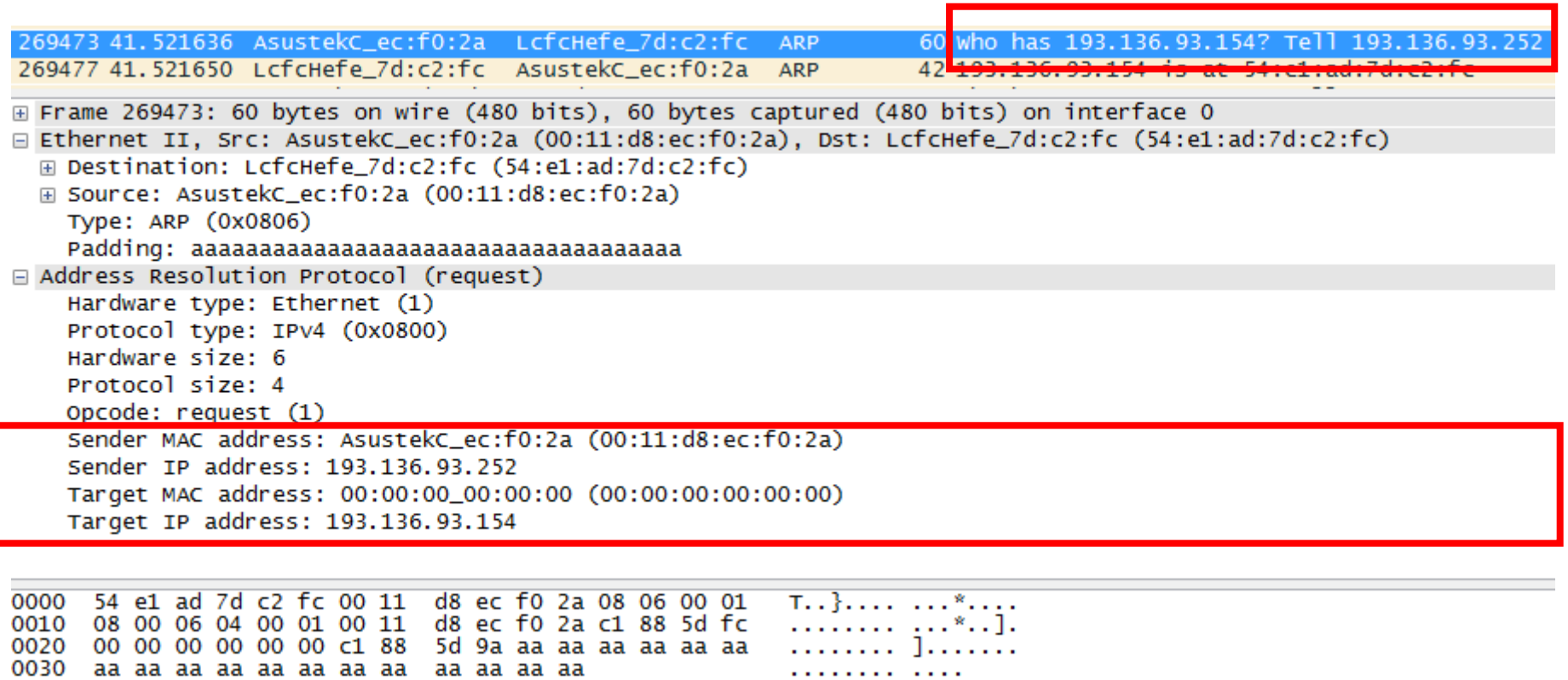
ARP protocol parameter values have been standardized and are maintained by the Internet Assigned Numbers Authority (IANA).[6]

The EtherType for ARP is 0x0806. This appears in the Ethernet frame header when the payload is an ARP packet and is not to be confused with PTYPE, which appears within this encapsulated ARP packet.

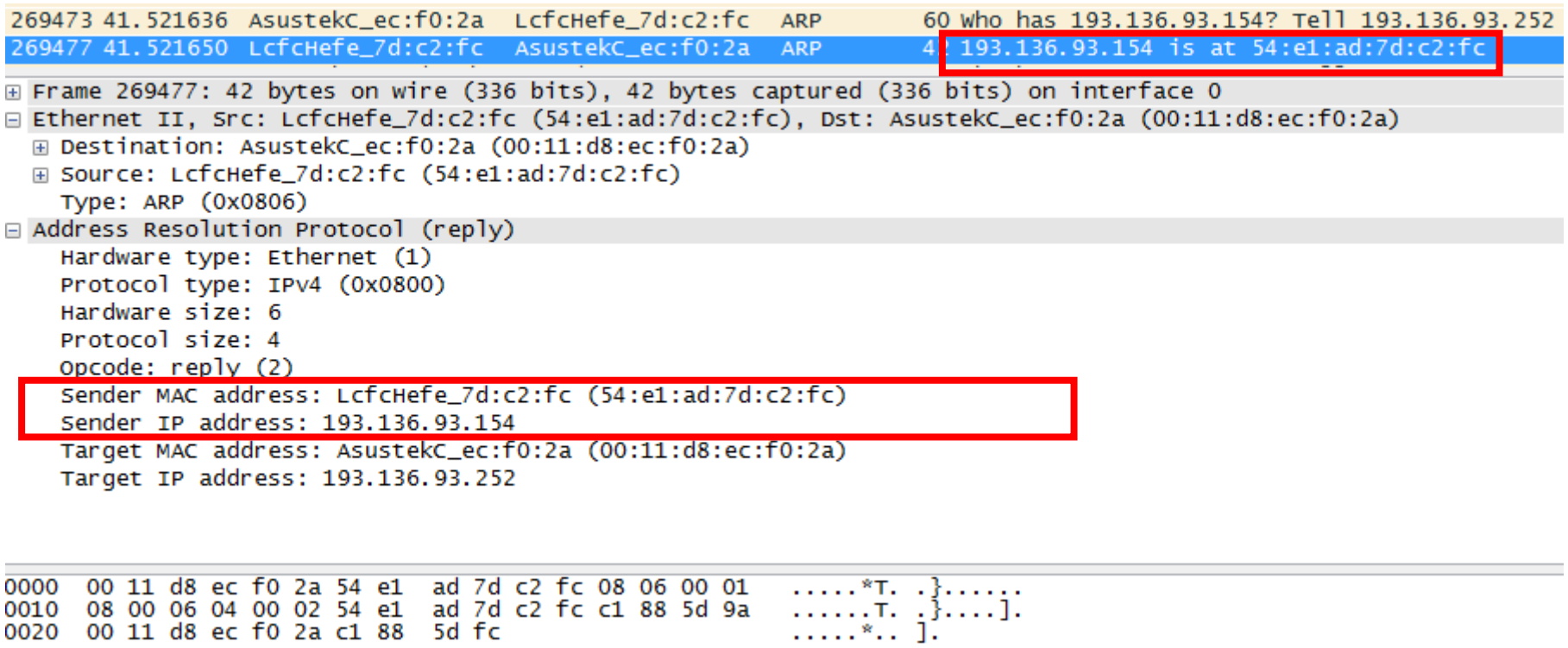
Internet Protocol (IPv4) over Ethernet ARP packet		
octet offset	**0**	**1**
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

Source: https://en.wikipedia.org/wiki/Address_Resolution_Protocol

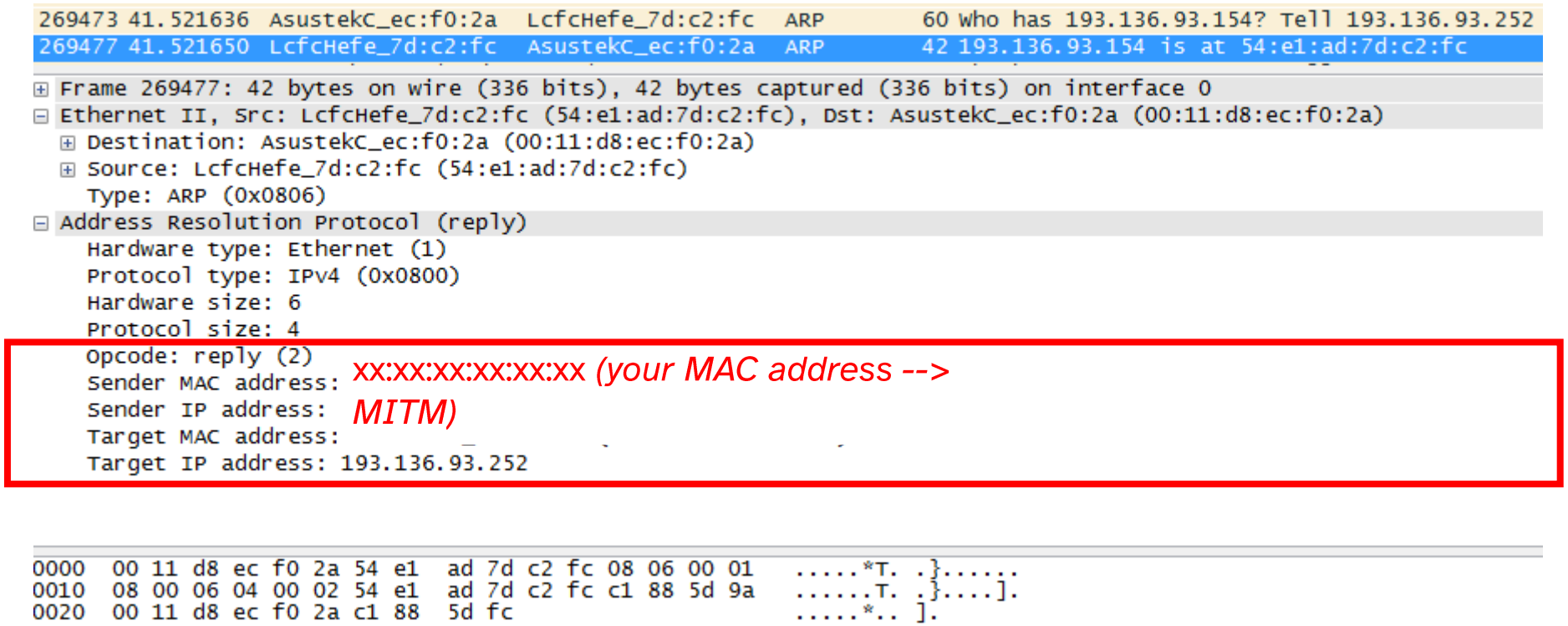
ARP Request


269473 41.521636 AsustekC_ec:f0:2a LcfcHefe_7d:c2:fc ARP 60 Who has 193.136.93.154? Tell 193.136.93.252
269477 41.521650 LcfcHefe_7d:c2:fc AsustekC_ec:f0:2a ARP 42
Frame 269473: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a), Dst: LcfcHefe_7d:c2:fc (54:e1:ad:7d:c2:fc)
Destination: LcfcHefe_7d:c2:fc (54:e1:ad:7d:c2:fc)
Source: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
Type: ARP (0x0806)
Padding: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
Sender IP address: 193.136.93.252
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 193.136.93.154
0000 54 e1 ad 7d c2 fc 00 11 d8 ec f0 2a 08 06 00 01 T..}.... ...*....
0010 08 00 06 04 00 01 00 11 d8 ec f0 2a c1 88 5d fc ....*..].
0020 00 00 00 00 00 00 c1 88 5d 9a aa aa aa aa aa aa ....].......
0030 aa aa aa aa aa aa aa aa aa aa aa aa ....

ARP Reply


269473 41.521636 AsustekC_ec:f0:2a LcfcHefe_7d:c2:fc ARP 60 Who has 193.136.93.154? Tell 193.136.93.252
269477 41.521650 LcfcHefe_7d:c2:fc AsustekC_ec:f0:2a ARP 4 193.136.93.154 is at 54:e1:ad:7d:c2:fc
Frame 269477: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: LcfcHefe_7d:c2:fc (54:e1:ad:7d:c2:fc), Dst: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
Destination: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
Source: LcfcHefe_7d:c2:fc (54:e1:ad:7d:c2:fc)
Type: ARP (0x0806)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: LcfcHefe_7d:c2:fc (54:e1:ad:7d:c2:fc)
Sender IP address: 193.136.93.154
Target MAC address: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
Target IP address: 193.136.93.252
0000 00 11 d8 ec f0 2a 54 e1 ad 7d c2 fc 08 06 00 01 .....*T. .}......
0010 08 00 06 04 00 02 54 e1 ad 7d c2 fc c1 88 5d 9a ......T. .}....].
0020 00 11 d8 ec f0 2a c1 88 5d fc .....*..].

ARP Reply (abusing the protocol)



No authentication whatsoever!

ARP Reply (abusing the protocol *v2*)

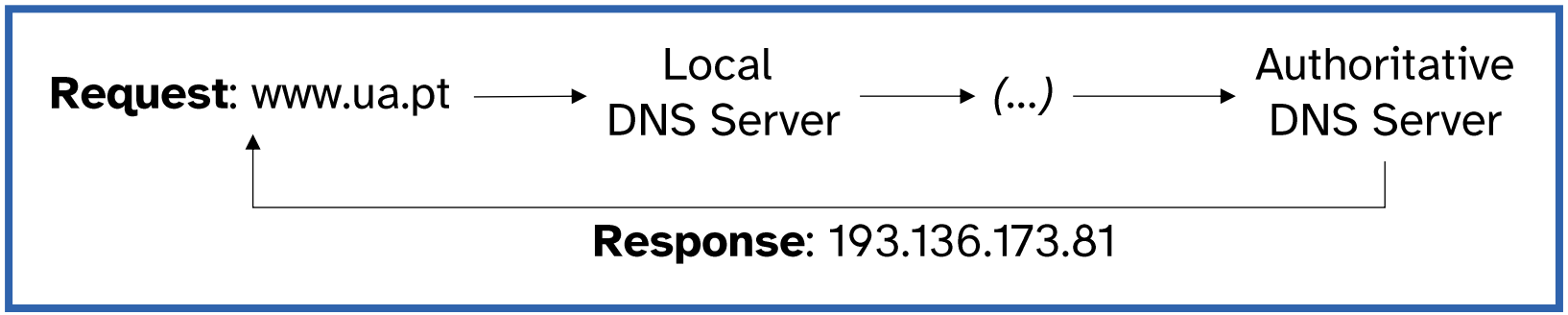
```
269473 41.521636  AsustekC_ec:f0:2a  LcfcHefe_7d:c2:fc  ARP      60 Who has 193.136.93.154? Tell 193.136.93.252
269477 41.521650  LcfcHefe_7d:c2:fc  AsustekC_ec:f0:2a  ARP      42 193.136.93.154 is at 54:e1:ad:7d:c2:fc

⊞ Frame 269477: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
⊟ Ethernet II, Src: LcfcHefe_7d:c2:fc (54:e1:ad:7d:c2:fc), Dst: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
  ⊞ Destination: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
  ⊞ Source: LcfcHefe_7d:c2:fc (54:e1:ad:7d:c2:fc)
    Type: ARP (0x0806)
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: xx:xx:xx:xx:xx:xx (inexistent MAC address --> DoS)
    Sender IP address: 193.136.93.154
    Target MAC address: AsustekC_ec:f0:2a (00:11:d8:ec:f0:2a)
    Target IP address: 193.136.93.252
```

```
0000  00 11 d8 ec f0 2a 54 e1  ad 7d c2 fc 08 06 00 01   .....*T. .}......
0010  08 00 06 04 00 02 54 e1  ad 7d c2 fc c1 88 5d 9a   ......T. .}....].
0020  00 11 d8 ec f0 2a c1 88  5d fc                     .....*.. ].
```

Bonus Information:

While IPv6 does not use ARP, it is still vulnerable to similar attacks in its Neighbor Discovery (ND) protocol!

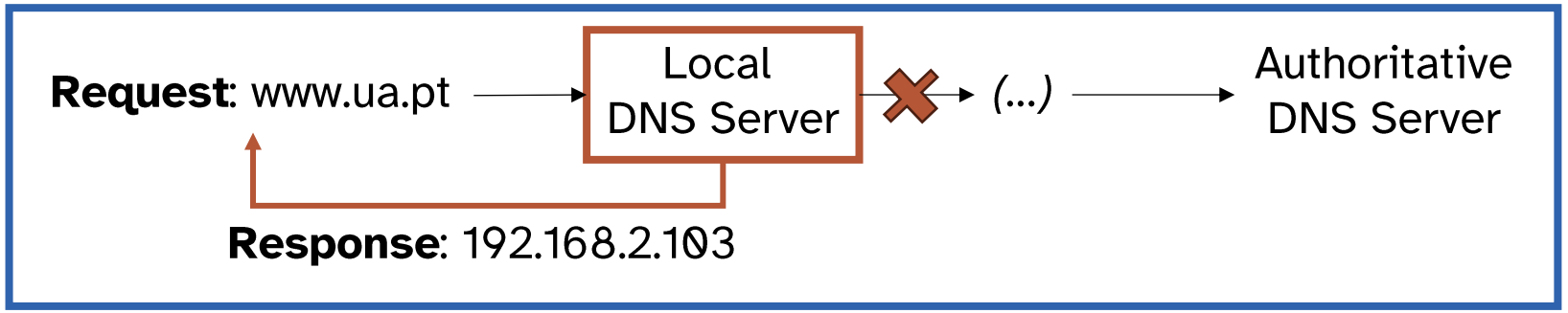
Moving up the OSI Stack – DNS Spoofing



Remember that we need to resolve domain names to IP addresses in order to establish a connection

If we control the LAN and its services (e.g., Local DNS Server)...

Moving up the OSI Stack – DNS Spoofing



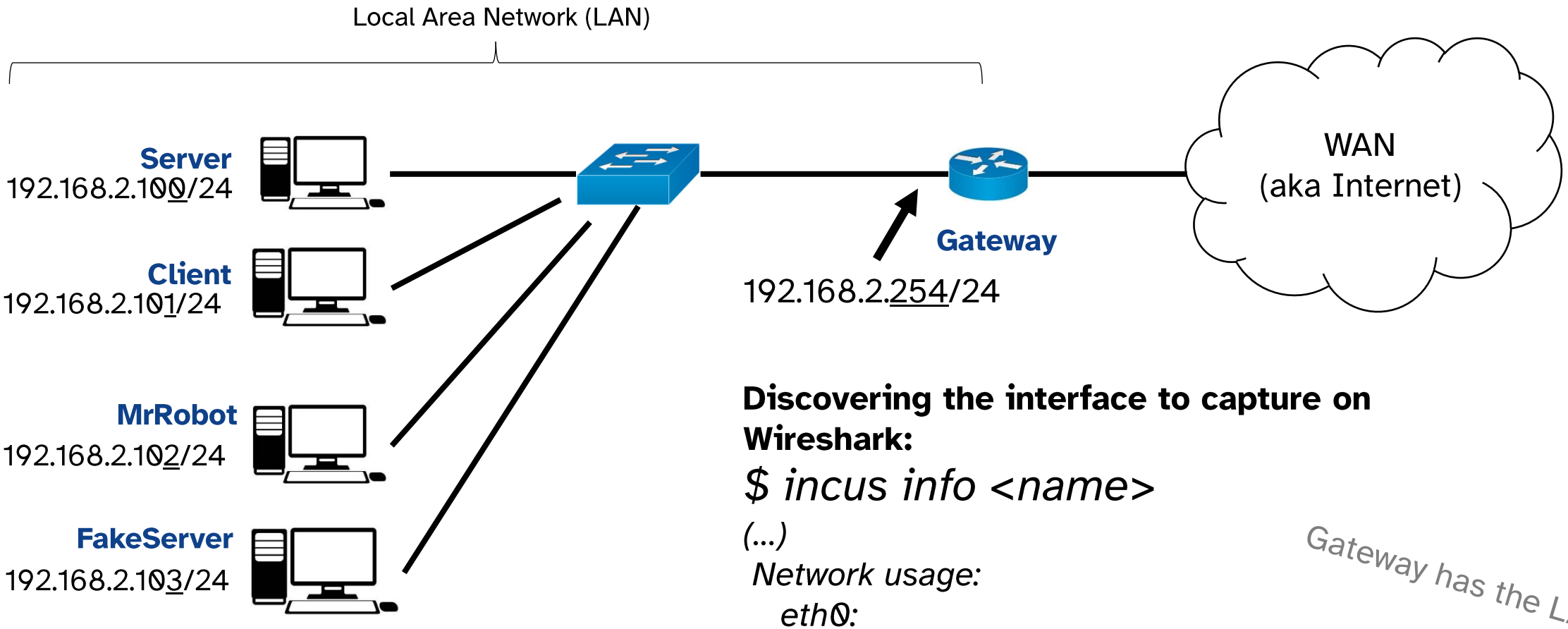
We can change the destination address that should be contacted for that domain, causing the client to contact the wrong server

Bonus Information:

If we know the resolution a client is requesting, we can forge a DNS reply that will change the resolved address even if we don't control the Local DNS

This is a race condition, you will need to respond faster than the Local DNS to succeed!

Guide Topology



Discovering the interface to capture on Wireshark:

$ incus info <name>

(...)

Network usage:

eth0:

Type: broadcast

State: UP

Host interface: ***veth45969bf9***

Gateway has the LAN on eth1