Network Security

- DNS Cache Poisoning -

GROUP 14
Bruno Boscia
Davide Todeschi
Giacomo Filippetto

Lab outline

- **Objective**
- **Environment setup**
- **Step 1:** remote attack (with DoS)
- **Step 2:** defense strategy through port and transaction ID randomization
- **Step 3:** attacker has access to the DNS's network (or evantually has perfomed a MITM Attack)

Objective

The objective is to show:

- How to poison a record in a recursive DNS's cache
- How defense strategies can prevent a DNS cache poisoning attack

Environment setup

- Start **all** the virtual machines
- The following username and password are valid on each Virtual Machine

  username : **user**

  password : **netsec**

- Enter the ‘’super user’’ mode in the server machines typing ‘’su’’ and entering the password
- Let’s start!

Environment setup

UBUNTU VM

192.168.56.104

- Attacker

- Victim user

DEBIAN VM

192.168.56.103

- DNS Nameserver

DEBIAN VM

192.168.56.101

- DNS Authoritative server
- Real website server

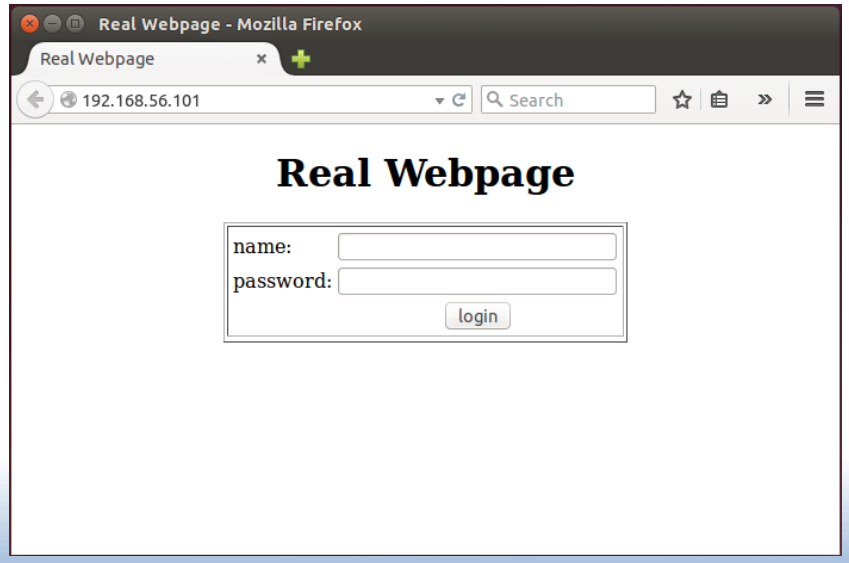
192.168.56.102

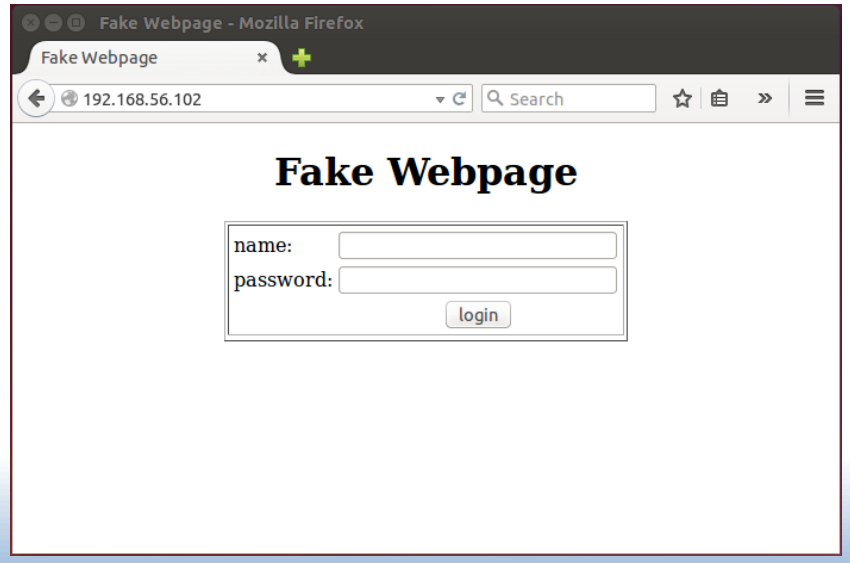
- Fake website server

Environment setup

On the Ubuntu machine, verify you can reach the two Web Servers:

- 192.168.56.101 ‘Real Website’
- 192.168.56.102 ‘Fake Website’

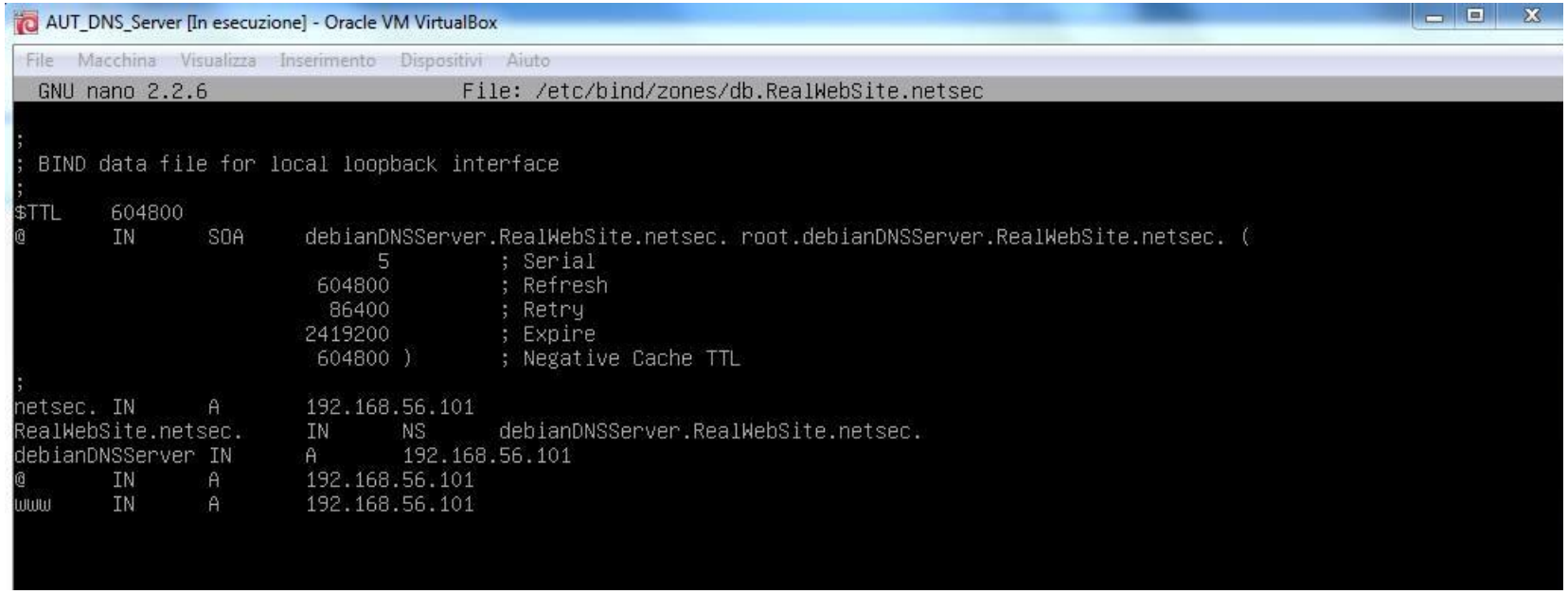




Environment setup

On the AUT_DNS_Server machine type:

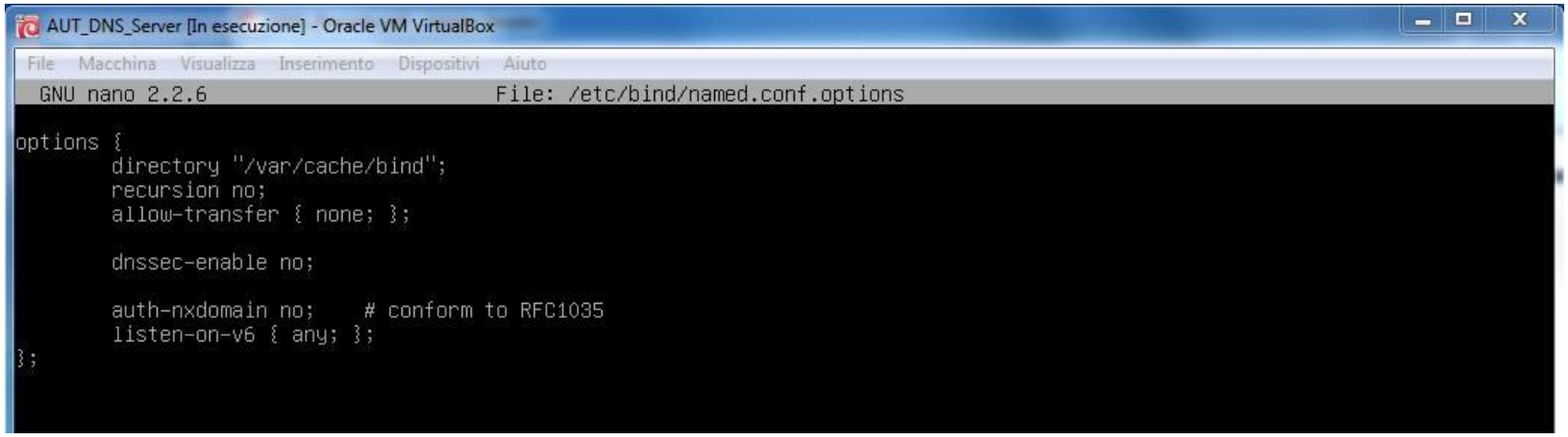
> nano /etc/bind/zones/db.RealWebSite.netsec



Environment setup

On the AUT_DNS_Server machine type:

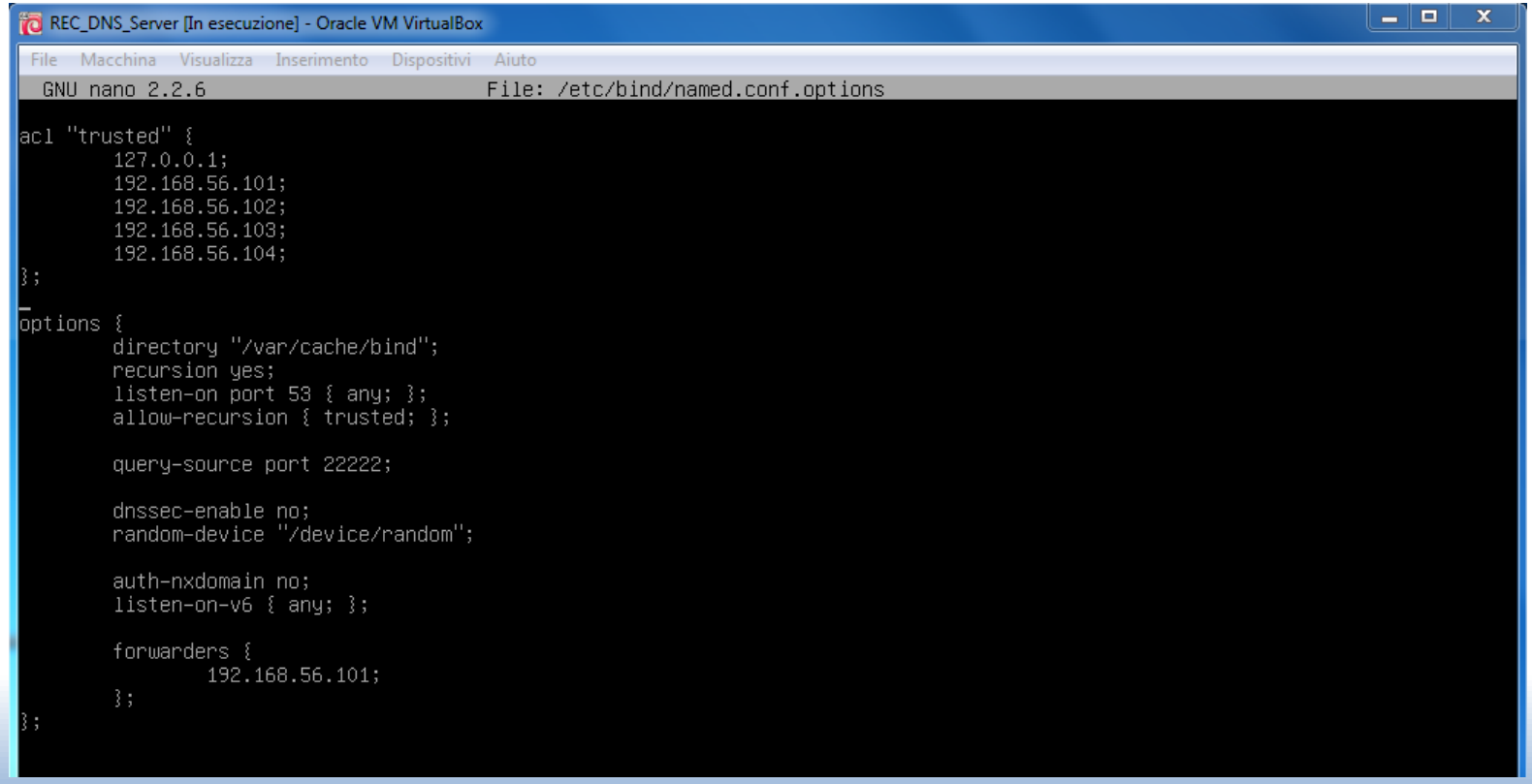
> nano /etc/bind/named.conf.options



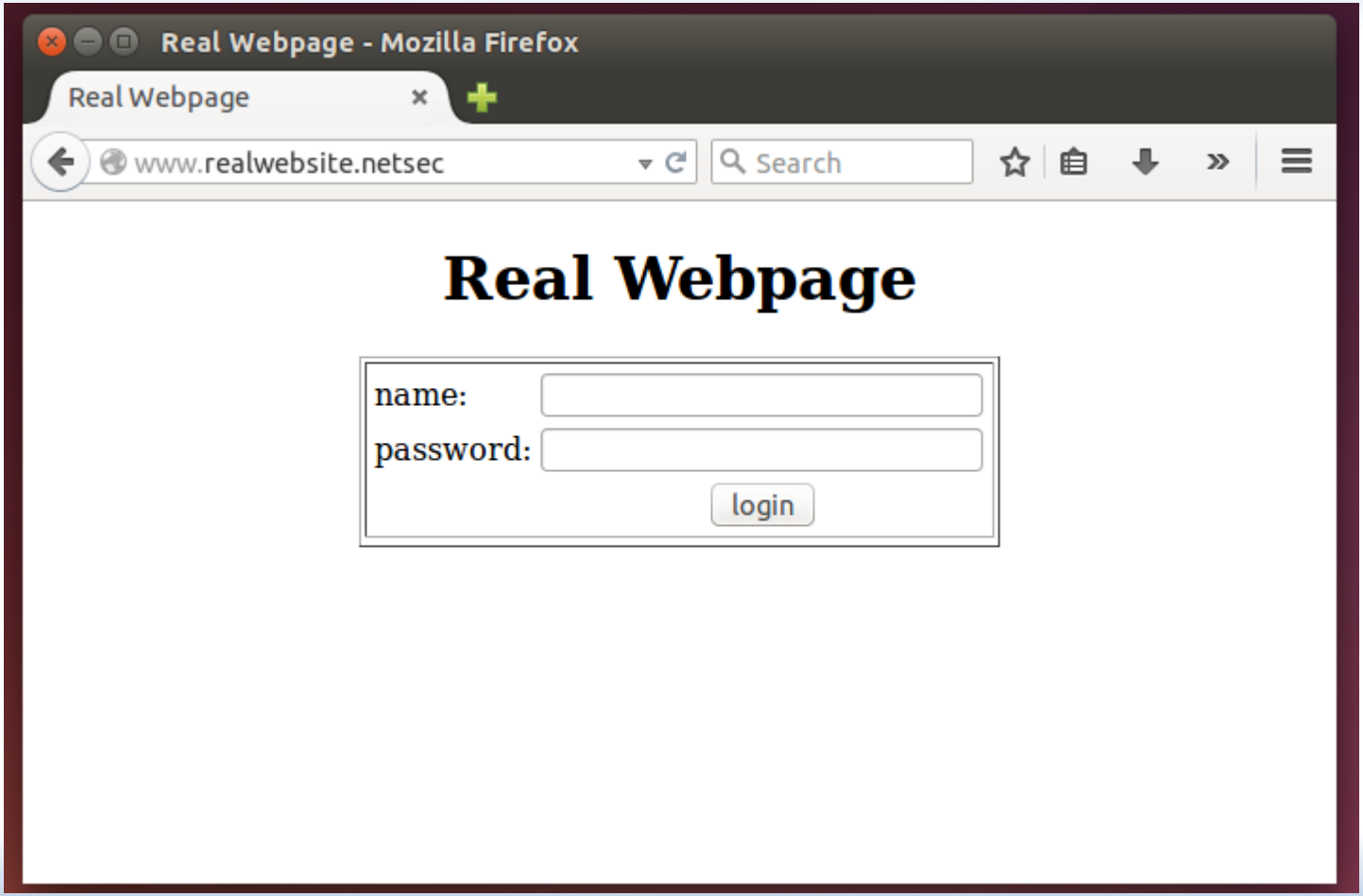
Environment setup

On the REC_DNS_Server machine type:

> nano /etc/bind/named.conf.options



Environment setup



About the DNS Cache

on the REC_DNS_Server:

- In order to visualize the actual cache,

  > rndc dumpdb -cache

  > nano /var/cache/bind/named_dump.db

- In order to flush the cache:

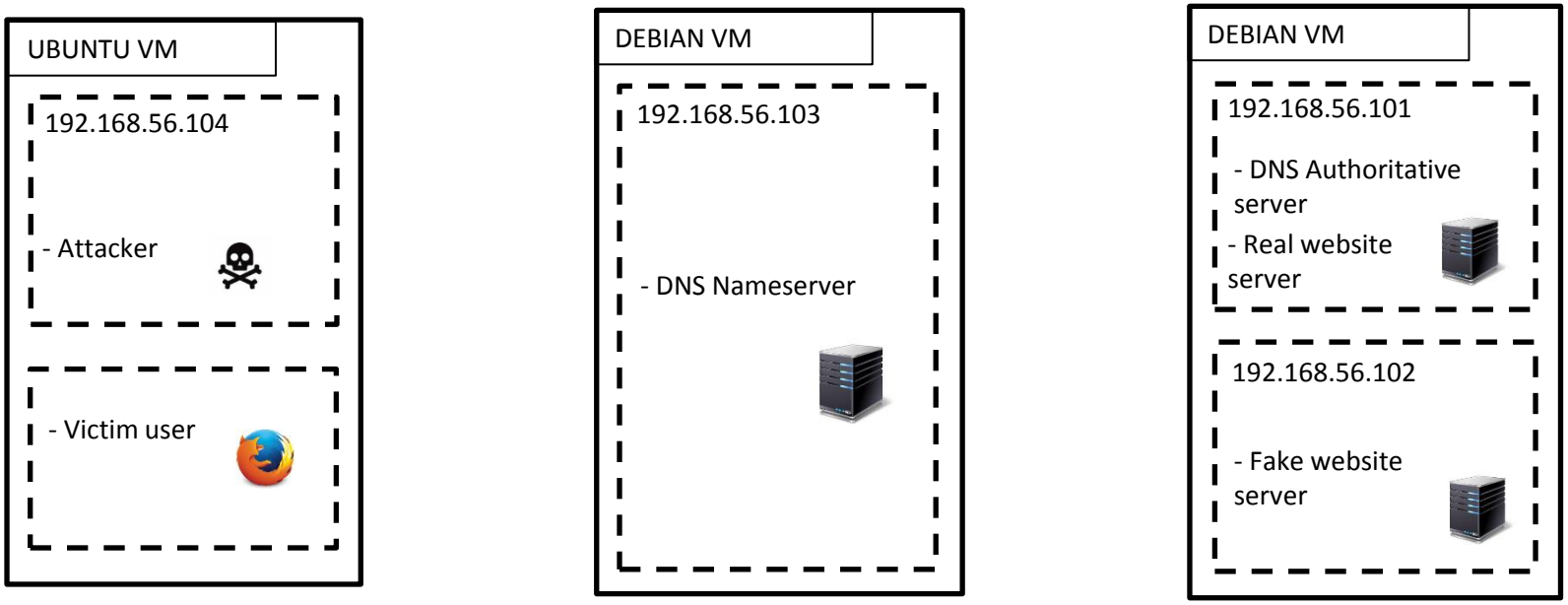
  > rndc flush

Step 1 - Attack

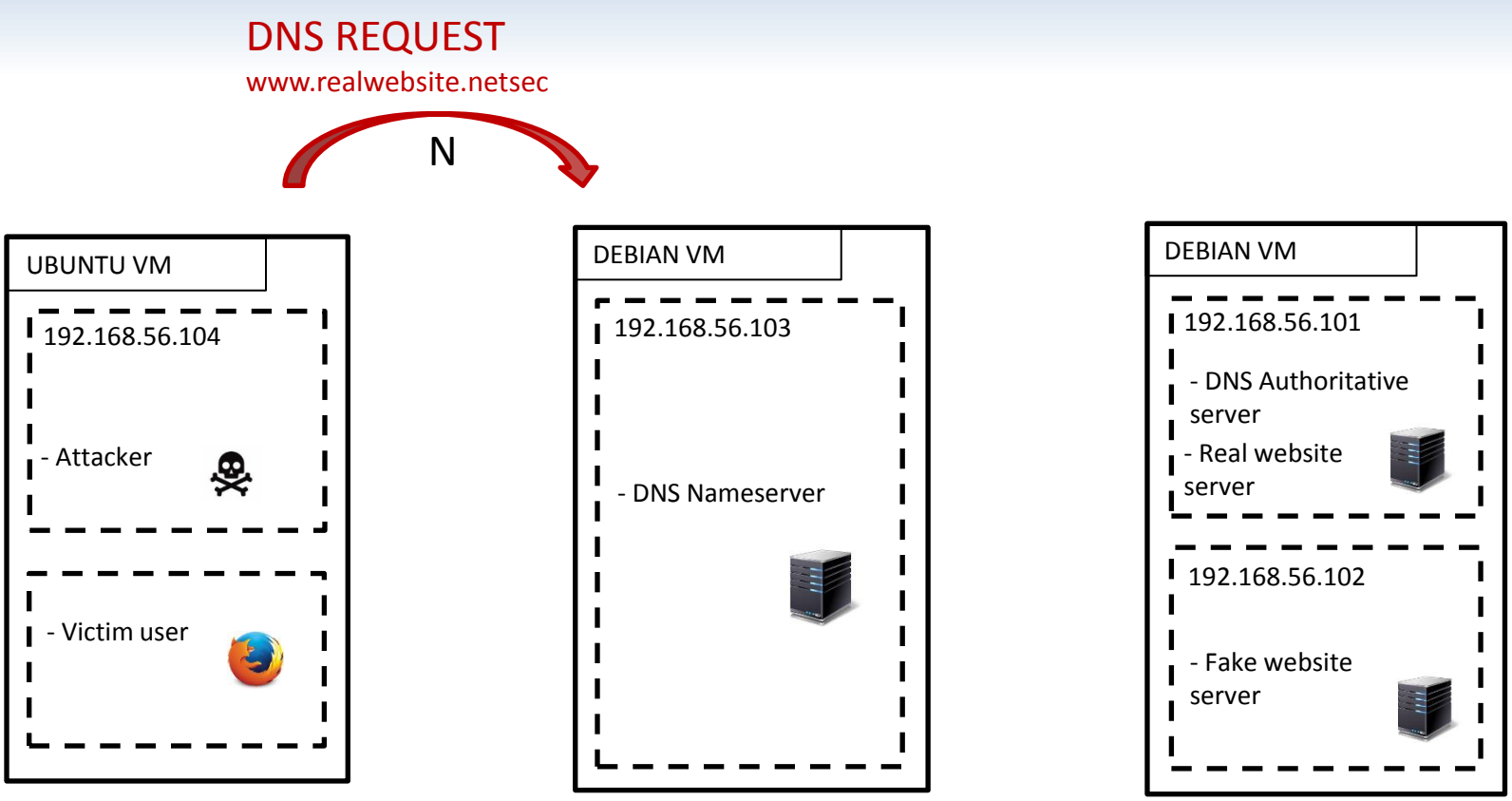
Birthday Attack:

- Continuously generate DNS requests and DNS answers with random Transaction IDs
- The attack exploits the mathematics behind the ‘birthday paradox’

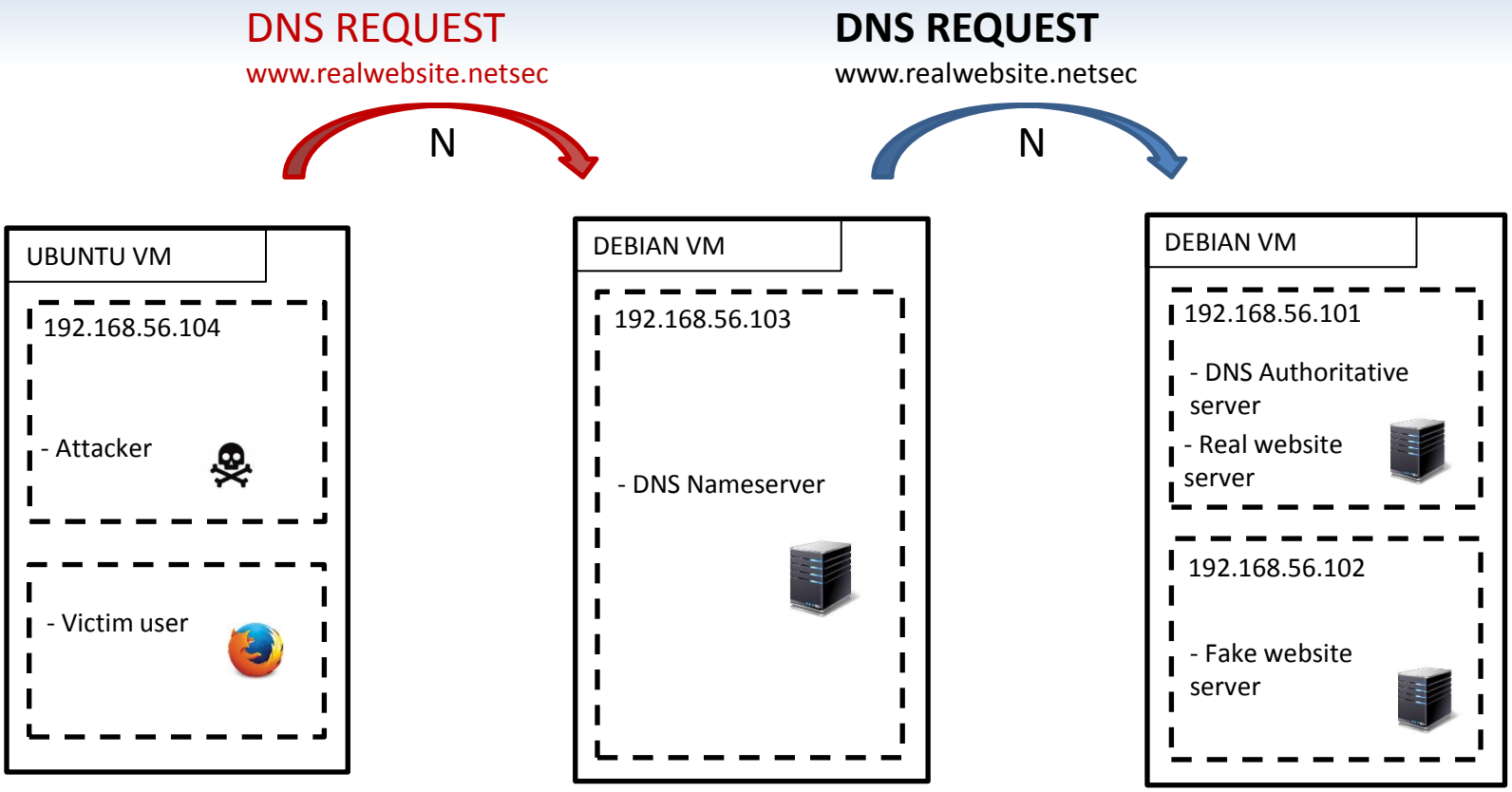
Step 1 - Attack



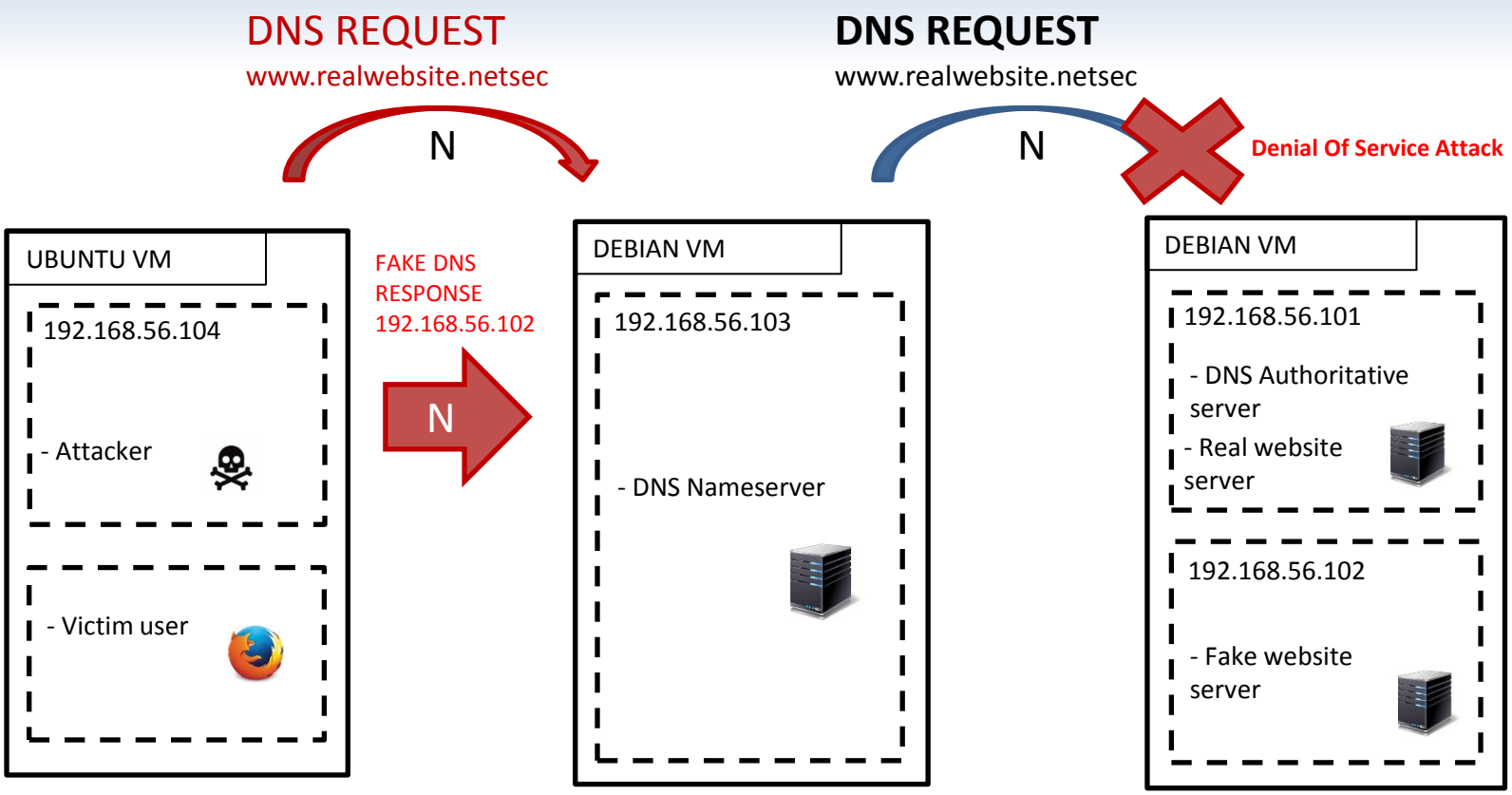
Step 1 - Attack



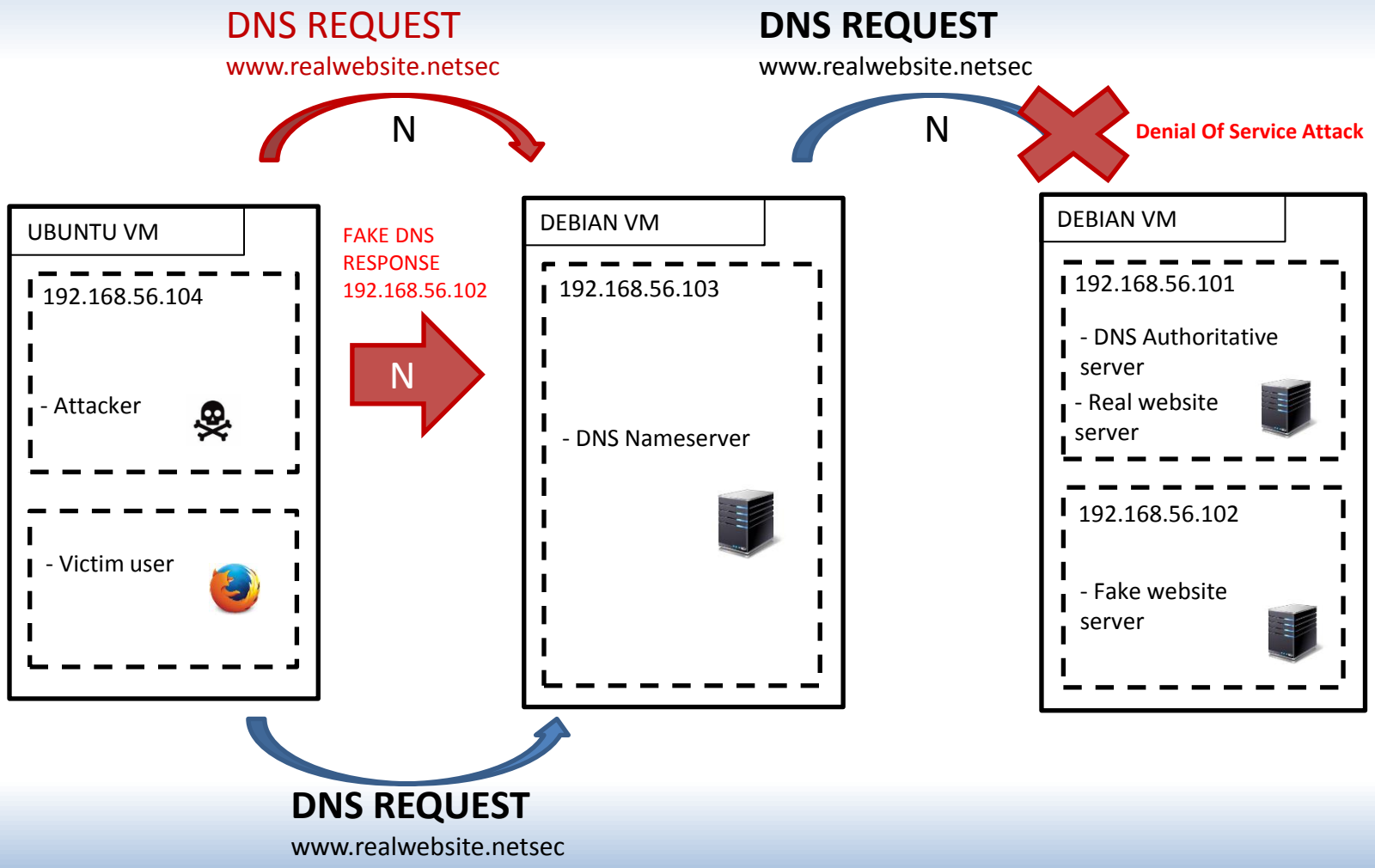
Step 1 - Attack



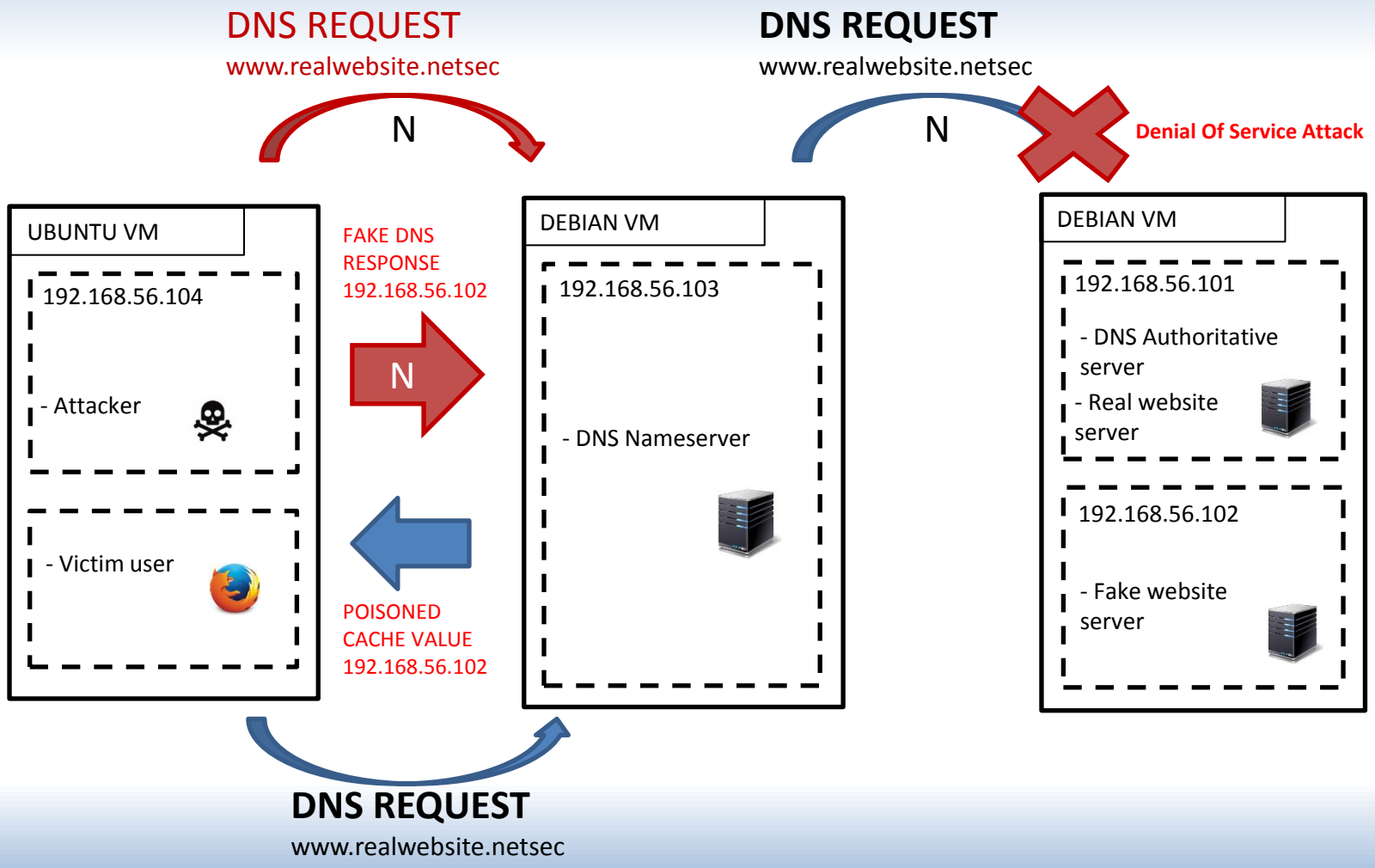
Step 1 - Attack



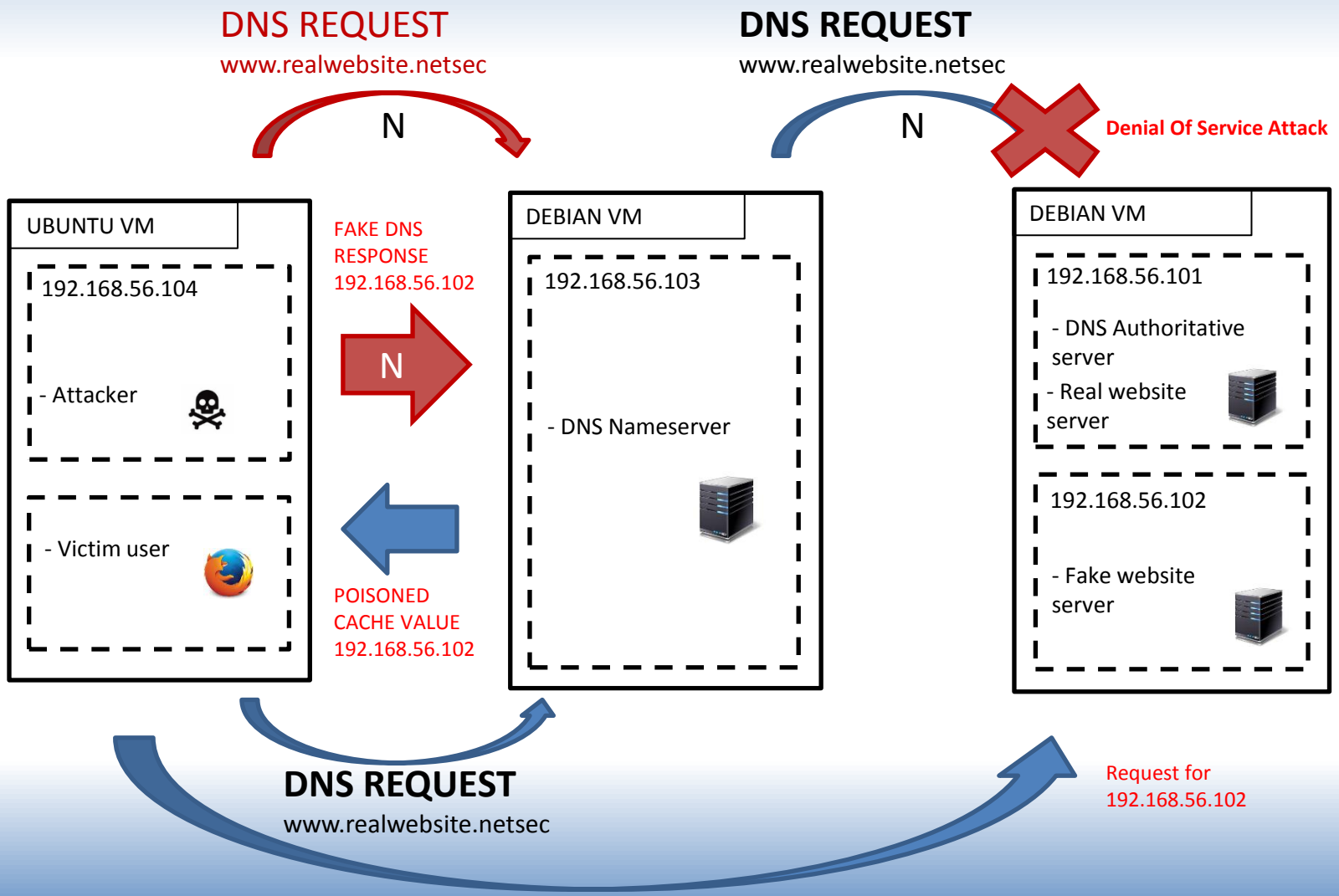
Step 1 - Attack



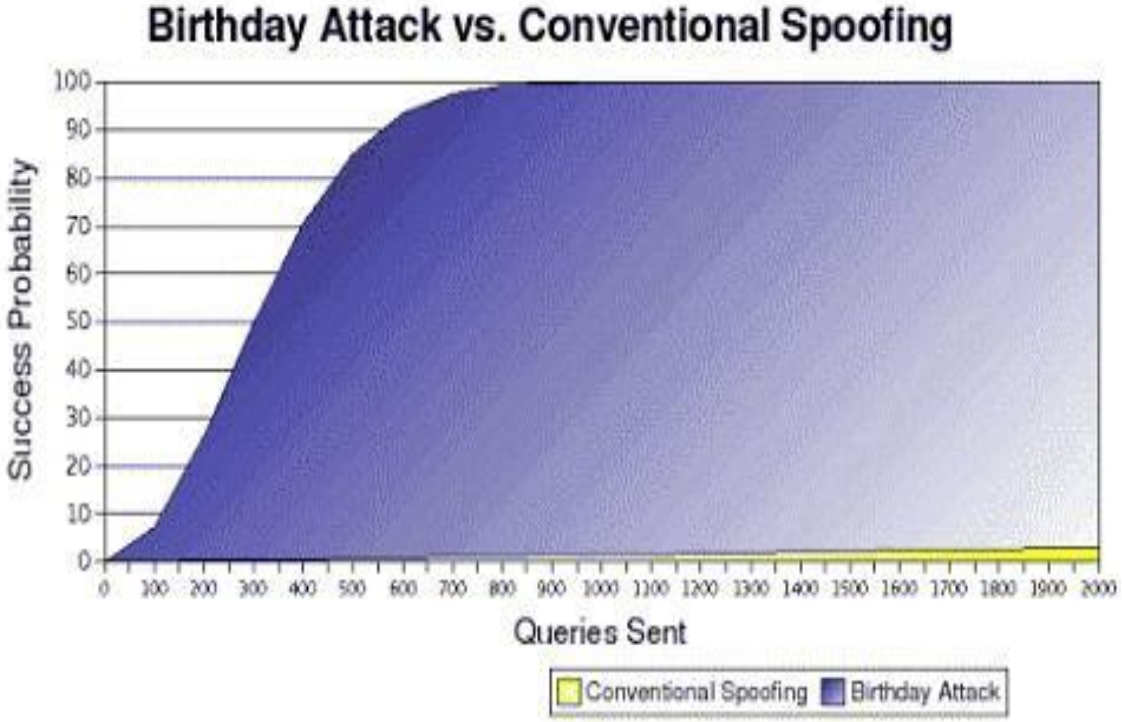
Step 1 - Attack



Step 1 - Attack



Step 1 - Attack



Step 1 - Attack

Now we have to perform a DoS attack,

but for this time we will cheat a bit…

Firewall

Block a specific IP address:

> iptables –A INPUT –s 192.168.56.103 –j DROP

Remove the block on that address:

> iptables –D INPUT –s 192.168.56.103 –j DROP

Show actual rules:

> iptables –L

Firewall

Now, if you send a ping from the REC_DNS_Server to the AUT_DNS_Server, you should receive no answer…

On the REC_DNS_Server type:

> ping 192.168.56.101

And check that nothing is diplayed after the following

PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.

Step 1 - Attack

Birthday Attack:

Open the Ubuntu Console and run the following script:

> gedit /home/user/Desktop/birthday_attack.py

Step 1 - Attack

Birthday Attack:

Open the Ubuntu Console and run the following script:

> sudo python /home/user/Desktop/birthday_attack.py

Step 1 - Attack

If it works, you should see something like this…

Step 1 - Attack

…well, actually… it doesn’t…

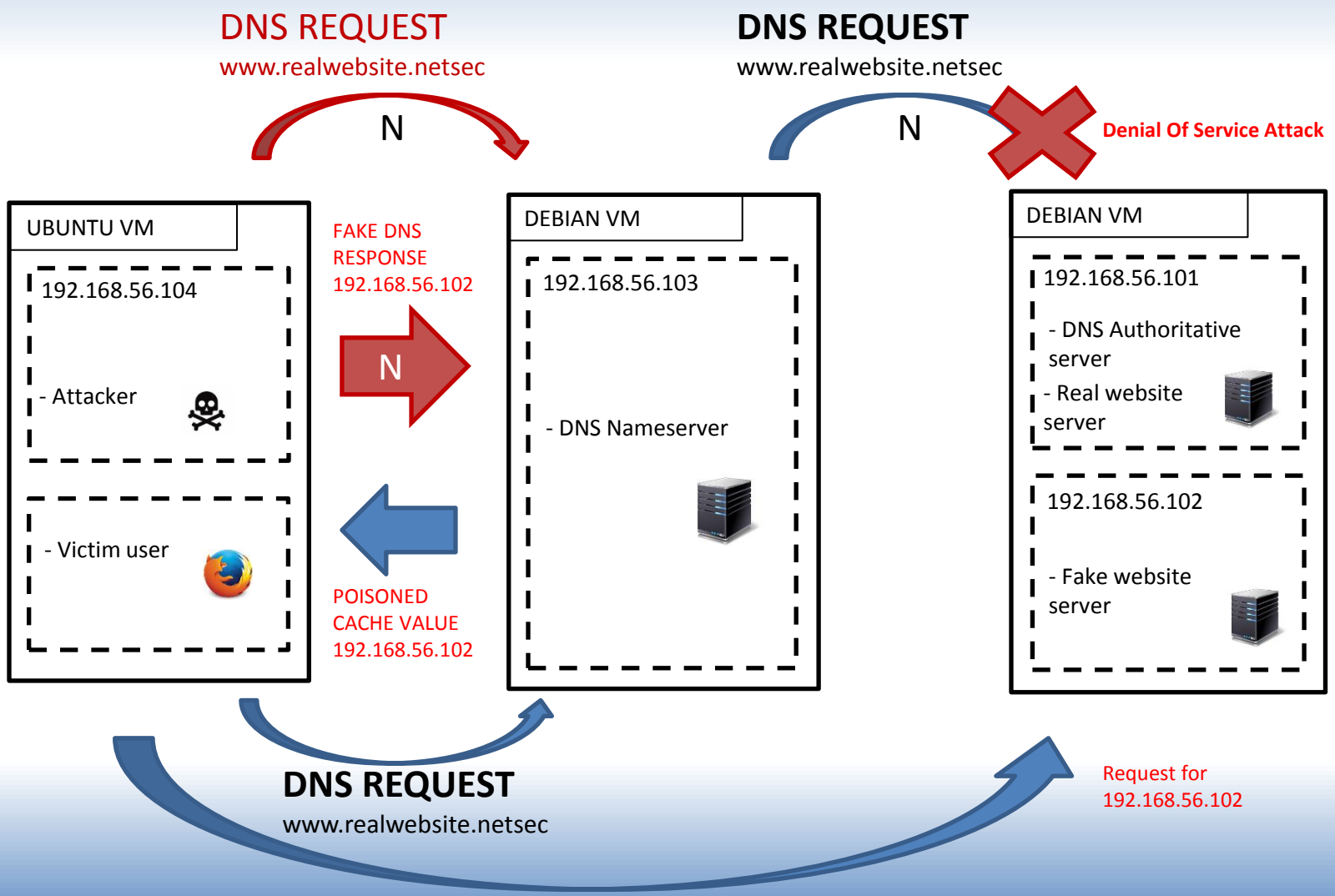
Step 2 - Defense

- The previous attack exploit a vulnerability that used to be present in bind4, but we are using bind9 instead…

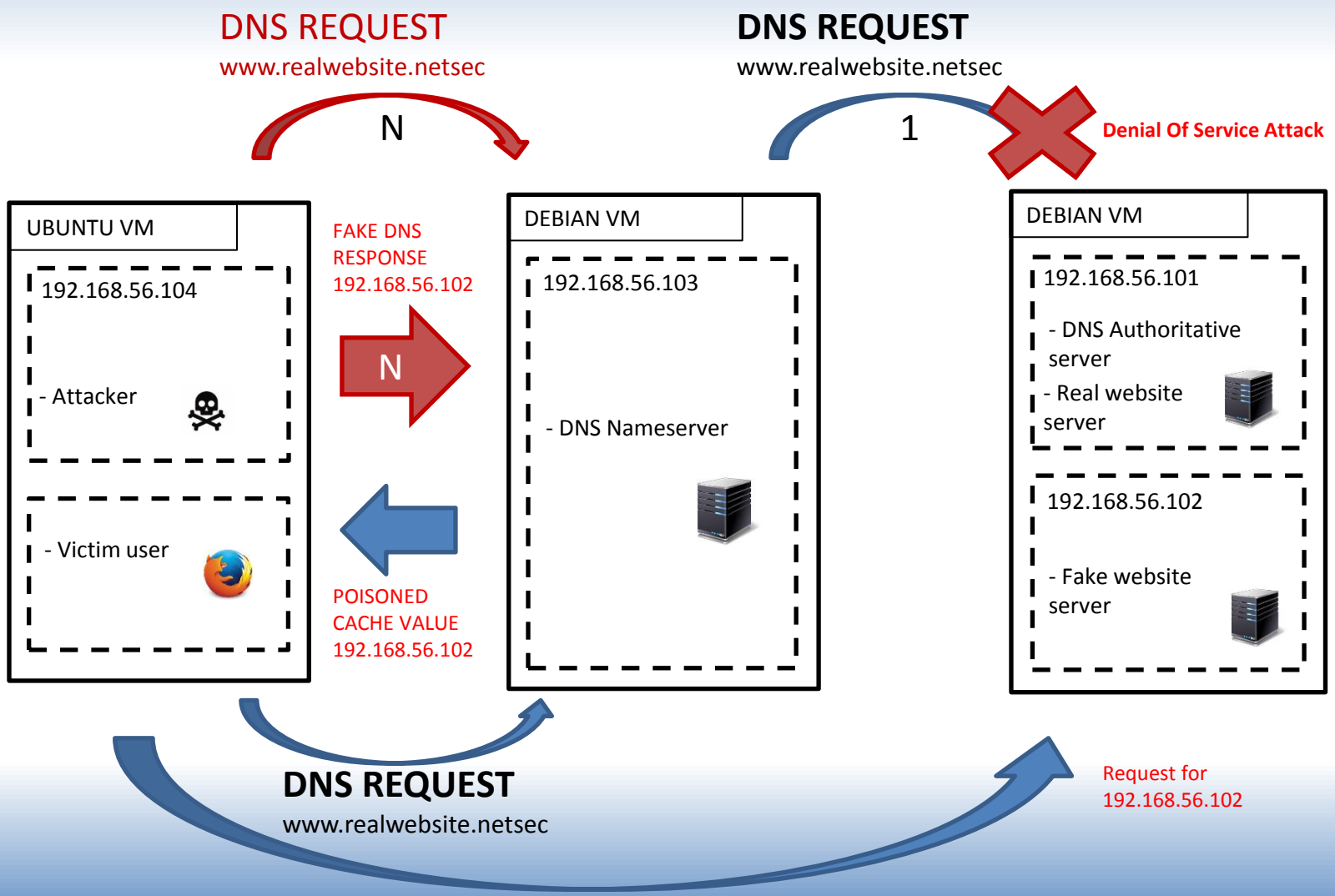
- The birthday attack has become useless since even if the DNS recursive receives more than one request, it forwards only one of them, which times out after few seconds.

- However, with a fast enough algorithm, this defense could be broken.

Step 2 – Defense



Step 2 – Defense



Step 2 - Defense

- A system's defense can easily be hardened by introducing Port randomization.

- On the REC_DNS_Server type:

  > nano /etc/bind/named.conf.options

- Then, remove or comment 'query-source port 22222;' and check that random-device is set to ''/device/random'';

Step 2 - Defense

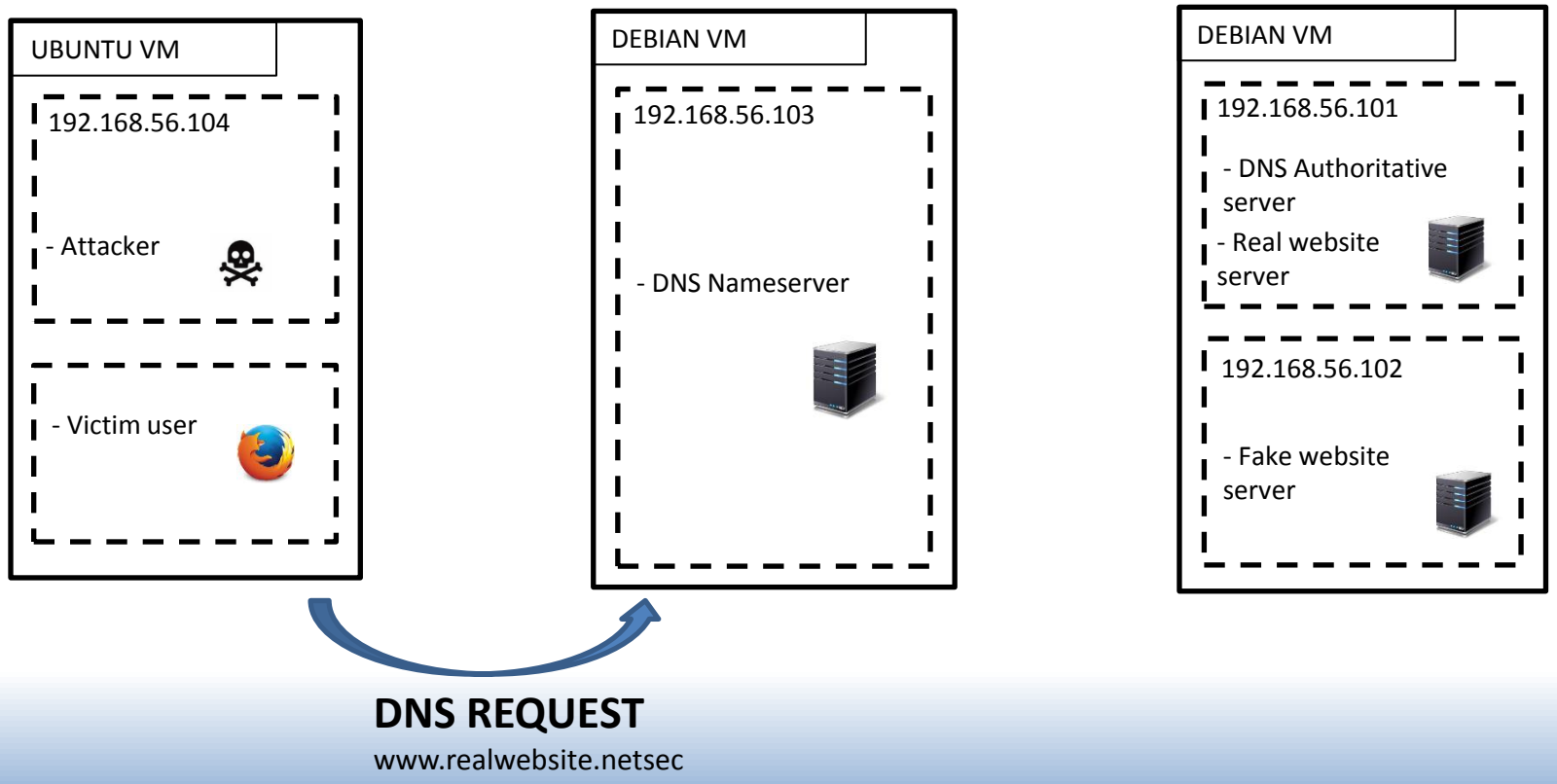
- Reproducing such an attack would be much harder, if not impossible, even with a very powerful algorithm.

Step 3 - Attack

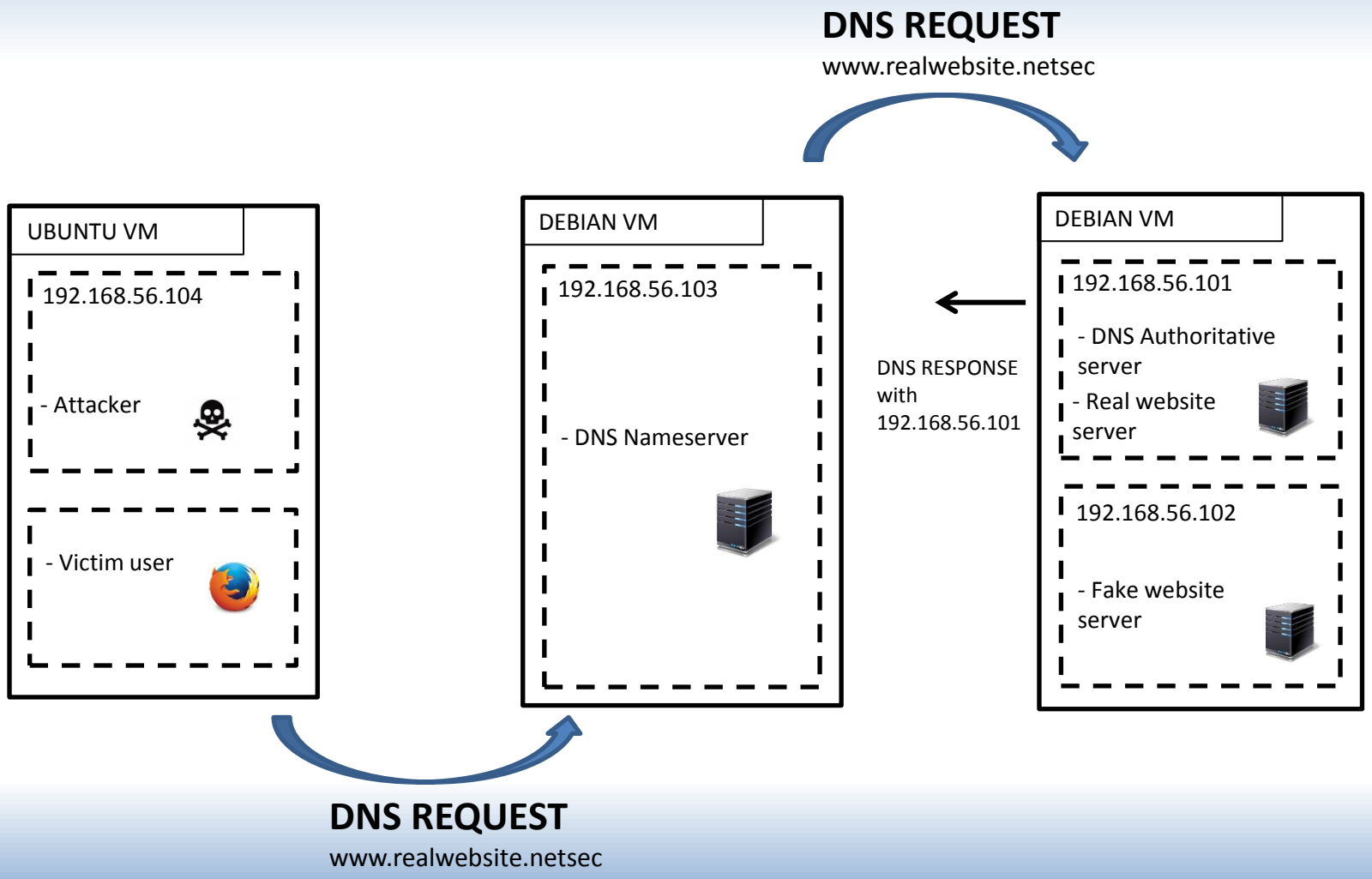
Packet Sniffing

In the case the attacker has access to the DNS's physical network (or he has peformed a Man In The Middle attack), he could be able to sniff the packets, answering to the DNS Queries while acting like the Authoritative DNS.

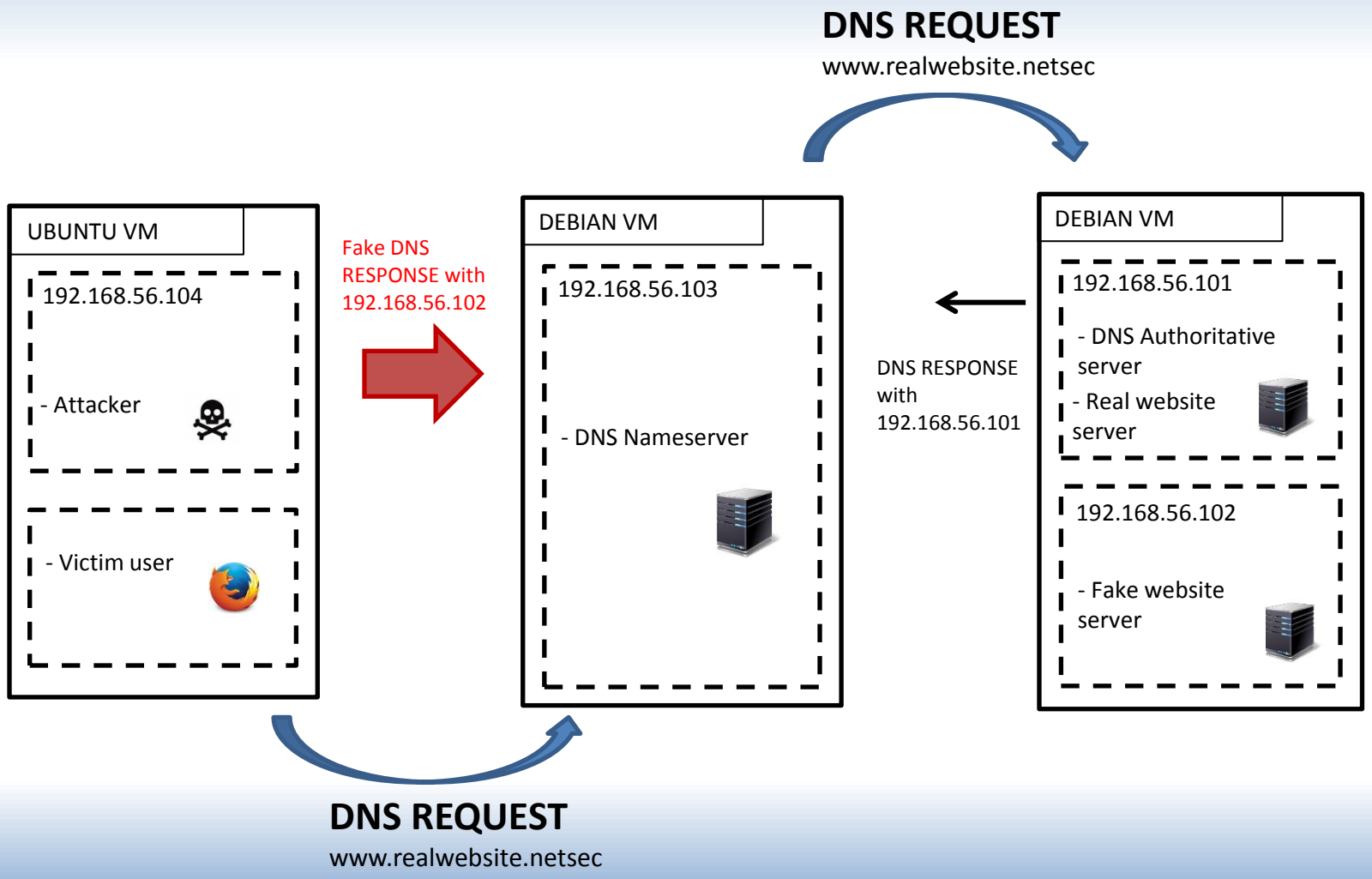
Step 3 - Attack



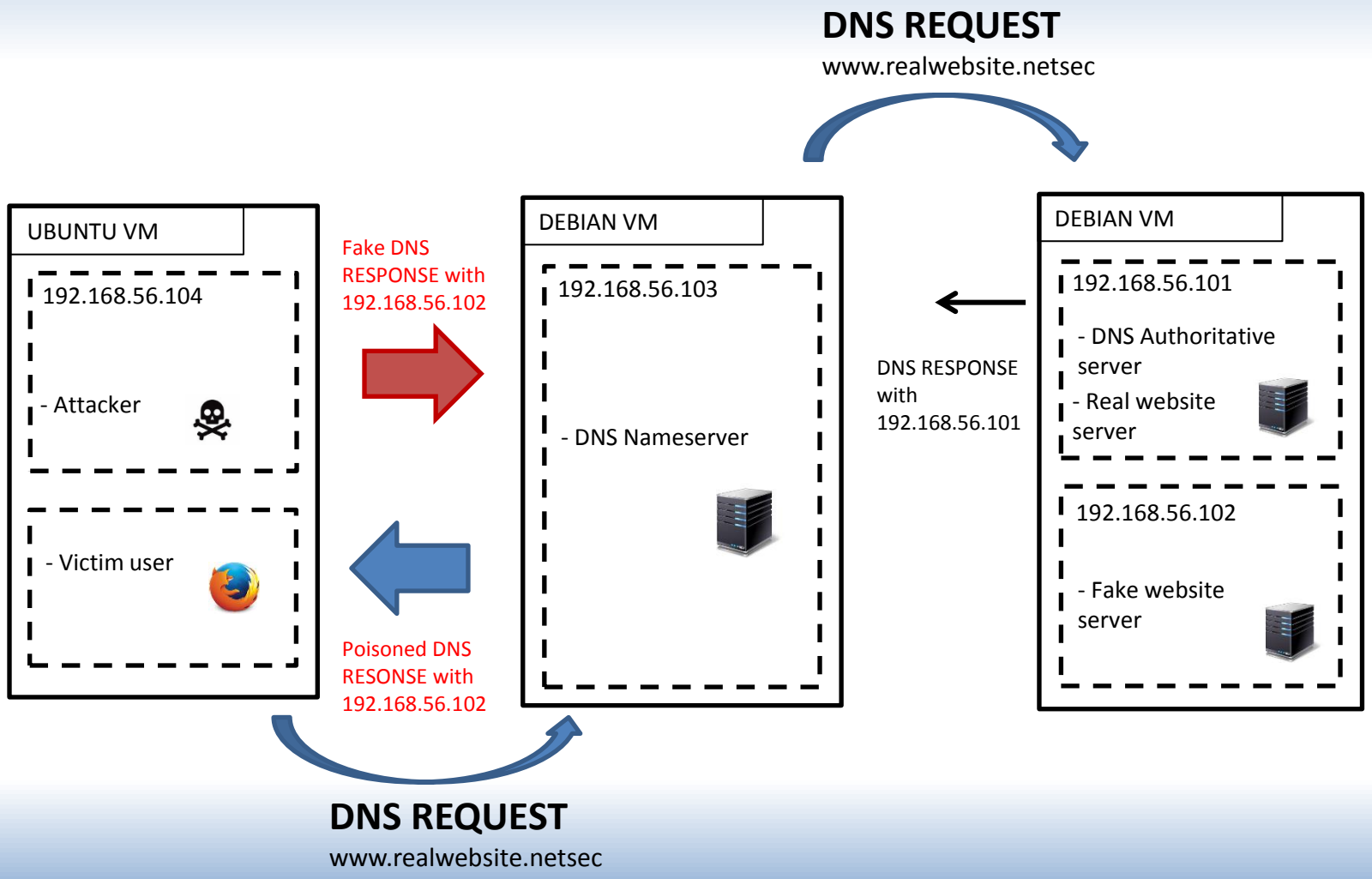
Step 3 - Attack



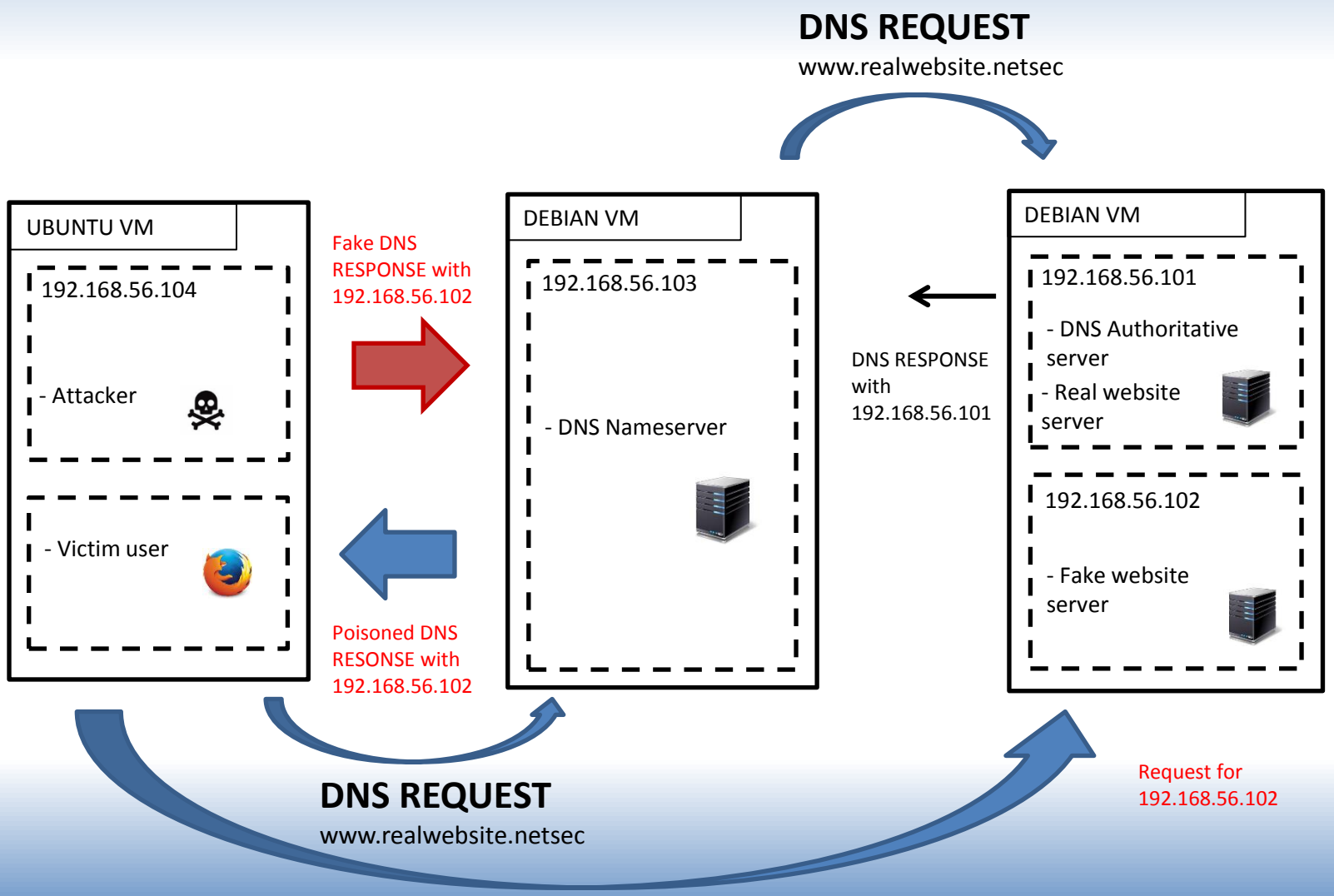
Step 3 - Attack



Step 3 - Attack



Step 3 - Attack



Step 3 - Attack

On the Ubuntu machine, open 2 terminals.

- In the first one, type

  > sudo python /home/user/Desktop/sniffing_attack.py

- In the second one, type

  > ping www.realwebsite.netsec

If the attack went well, when you open the browser and look for 'www.realwebsite.netsec' you should be redirected to the fake website.