TÉCNICO LISBOA

Computer Networks — 2023/24, Q2		**Assignment:**	Lab 4
Domain Name System		**Version:**	1.2
Issued:	2023-11-20	**Submission Due:**	2023-11-26

Submission note. You will need to submit all the answers to this lab's questions in Moodle, under "Lab 4: DNS".

Important. Using the provided virtual machine execute the command `git pull origin master` in `/home/rc/lab-files`. Next, execute `/home/rc/lab-files/04-lab-dns/setup.sh`.

1 Goals of the laboratory

- Explore different types of DNS (resource) records (SOA, A, AAAA, NS, CNAME, and MX).
- Learn the differences between an iterative and a recursive DNS query.

2 Exercises

2.1 Types of DNS records

In this part of the assignment, you will be using `dig`, a tool to perform DNS lookups, to learn more about the most important types of DNS records: SOA, A, AAAA, NS, CNAME, and MX. The tool supports multiple flags to control the DNS query, like `-t`, which lets you request a specific type of DNS record (possible values are `any`, `soa`, `a`, `aaaa`, `ns`, `cname`, `mx`, etc.).

Hint. To answer the questions at the bottom of the section, analyze the output of `dig`.

2.1.1 DNS record syntax

DNS records follow a simple syntax, as shown below (this is a simplified version).

```
<name> <ttl> <class> <type> ...
```

Fields are space-separated and some are optional (some types of DNS records may even have additional fields on top of optional ones):

- `<name>` is the domain;
- `<ttl>` stands for time-to-live and dictates the time, in seconds, that a record may be temporarily stored in cache;
- `<class>` refers to the class of DNS record (although, in practice, records always refer to the Internet, `IN`);
- `<type>` refers to the type of DNS record, as described in the following subsections.

2.1.2 Type SOA (start of authority)

A type SOA record stores important information about a domain or zone, such as the master authoritative DNS server for the domain, the email address of the administrator, and a serial number that can be used to detect changes in the zone. Read more about how this data is structured here.

Get the SOA record for the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t soa tecnico.ulisboa.pt
```

2.1.3 Type A (address)

A type A DNS record is a translator that maps a (fully qualified, or canonical) domain to an IPv4 address.

Get the IPv4 address of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t a tecnico.ulisboa.pt
```

It is important to mention that a DNS client may receive multiple records of the same type in the same DNS query response.

Get the IPv4 address of the domain `rtp.pt`.

```
:~$ dig -t a rtp.pt
```

This is useful for client-side load-balancing: the DNS client randomly picks one record, distributing network traffic across multiple servers.

2.1.4 Type AAAA (quad A)

A type AAAA DNS record is similar to a type A DNS record, but maps to an IPv6 address.

Get the IPv6 address of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t aaaa tecnico.ulisboa.pt
```

2.1.5 Type NS (name server)

A type NS DNS record indicates which DNS servers are authoritative for a given domain.

Get the list of authoritative servers of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t ns tecnico.ulisboa.pt
```

2.1.6 Type CNAME (canonical name)

A type CNAME DNS record maps an alias name to a canonical domain, but never to an IP address.

Get the canonical name of the domain `www.ulisboa.pt`.

```
:~$ dig -t cname www.ulisboa.pt
```

2.1.7 Type MX (mail exchange)

A type MX DNS record specifies the mail server responsible for handling email messages of a given domain.

Get the mail servers of the domain `tecnico.ulisboa.pt`.

```
:~$ dig -t mx tecnico.ulisboa.pt
```

2.1.8 Questions

Q1a What are the names of the DNS servers that are authoritative for the domain `tecnico.ulisboa.pt`?

Q1b What are the IPv4 addresses of these authoritative servers?

Q2a What is the canonical name of `www.ulisboa.pt`?

Q2b What is the respective IPv4 address?

Q3 What is the email address of the administrator of the domain `tecnico.ulisboa.pt`?

Q4a In the scope of the domain `tecnico.ulisboa.pt`, what server IPv4 addresses will be contacted for HTTP(S)?

Q4b And what server names will be contacted for SMTP?

2.2 Iterative vs. recursive DNS queries

In this part of the assignment, you will learn the differences between iterative and recursive queries, two strategies used to resolve a domain. You will mainly be looking at the number and content of messages exchanged between the DNS client and the DNS server(s) involved.

> **Hint.** To capture DNS traffic, use Wireshark with an adequate filter (e.g. `dns and udp`). To answer the questions at the bottom of the section, inspect the most relevant packets that you captured (do not overlook the «Additional records» section of some responses).

Before proceeding, try to answer this question: in the previous exercises, how did `dig` know which DNS server(s) to contact if you didn't specify any? The truth is that your operating system keeps a list of DNS servers to reach out to when resolving a domain (on Linux, you can find the list at `/etc/resolv.conf`.) Each server on that list is known as a *resolver*, and it is the first stop in the DNS lookup, responsible for handling the client that made the initial query. In other words, for instance, when you `ping google.com`, `ping` asks the first resolver on the list (the resolver is the DNS client) to resolve the domain `google.com`.

2.2.1 Iterative query

Start a new packet capture and perform an iterative DNS query to resolve the domain `tecnico.ulisboa.pt`.

> **Hint.** To force `dig` to communicate directly with each DNS server in an iterative fashion, use `+trace` (in this particular case, `dig` becomes the DNS client). Moreover, consider using `+nodnssec` (to prevent the tool from requesting DNSSEC records) and `-4` (to use IPv4 query transport only) to improve the readibility of your packet captures.

```
:~$ dig +trace tecnico.ulisboa.pt +nodnssec -4
```

2.2.1.1 Questions

Q5 How many DNS requests does `dig` make to resolve the domain? *Hint: Wireshark may be your friend.*

Q6 Build a table with the name of the DNS servers that respond to the requests, and their DNS types, sorting them by when they were contacted (from first to last).

Q7 DNS query responses sometimes include additional records that aim to reduce the number of additional queries and speed up name resolution. How? (Select all that apply.) *Hint: analyse the responses from a TLD server, for instance*

2.2.2 Recursive query

Start a new packet capture and perform a recursive DNS query to resolve the domain `tecnico.ulisboa.pt`.

1. First flush the DNS cache:

```
:~$ sudo systemd-resolve --flush-caches
```

2. Perform the recursive DNS query

```
:~$ dig tecnico.ulisboa.pt -4
```

2.2.2.1 Questions

Q8 This time, how many servers does `dig` contact? *Hint: Wireshark may be your friend.*

Q9 Who is now performing the DNS queries?