IMT Atlantique
Département Informatique
Technopôle de Brest-Iroise - CS 83818
29238 Brest Cedex 3
URL: **www.imt-atlantique.fr**

UE PRIP
2022

Lab DNS: Domain Name System

Edited: September 30, 2022
Version: 1.3-2020-09

Report filled-in by:

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1. Objectives

This lab aims at helping you to:

- Understand how the domain names system of the Internet works
- Get familiarised with the packet analyser software Wireshark initial objective

2. Pre-LAB

- Study the linux commands `ifconfig, ip address, ip route, dig, host, ping`. You can, for instance, read the man page of these commands by typing on a UNIX terminal `man <command>` (For the `ip` tools commands, you must replace the space with a dash, e.g. `man ip-address`).
- Investigate the different types of resource records existing for the Internet class in the DNS, and their purpose. In particular, complete Table 1.
- Investigate what are iterative DNS queries and recursive DNS queries.
- Readings:
  - The introduction of this lab
  - Serveur DNS faisant autorité : définition, by Stéphane Bortzmeyer: https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html
  - Résolveur DNS : définition https://www.bortzmeyer.org/resolveur-dns.html
  - On the usage of DNS by CDNs: https://labs.ripe.net/Members/emileaben/how-ripe-atlas-help
  - On censorship applied through DNS queries: https://labs.ripe.net/Members/stephane_bortzmeyer/dns-censorship-dns-lies-seen-by-atlas-probes (optional)
  - Nouvelles attaques facilitant l'empoisonnement DNS: https://www.bortzmeyer.org/dns-attaques-shulman.html

3. Introduction

3.1. The Domain Names System

We, as humans, are used to refer to Web pages, mailboxes, and other network resources by using a readable, easy to remember name, like for instance www.imt-atlantique.fr. However, network equipments understand and use numerical addresses (e.g. IP addresses). Having only IP address to refer to network resources would mean, for instance, accessing the Web page hosted at 2001:660:7302:2::21 (or 192.108.117.237), which is not only hard to remember, but supposes that if the Web page changes its location, then we, users, should be aware of this change. As a consequence, a high-level readable names system is used in the Internet in order to allow to decouple machine *names* from machine *addresses*.

How to achieve this decoupling and the mapping between names and addresses at the Internet scale? How to have unique names all across the Internet? How to avoid having a central entity managing the mapping? How to accomplish a system at the Internet-scale avoiding huge files/data bases? The way the Internet community has solved this is called Domain Name System (DNS), and has become a key component of the Internet.

The essence of DNS is a hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme. It is primarily used for mapping host names (often referred to as

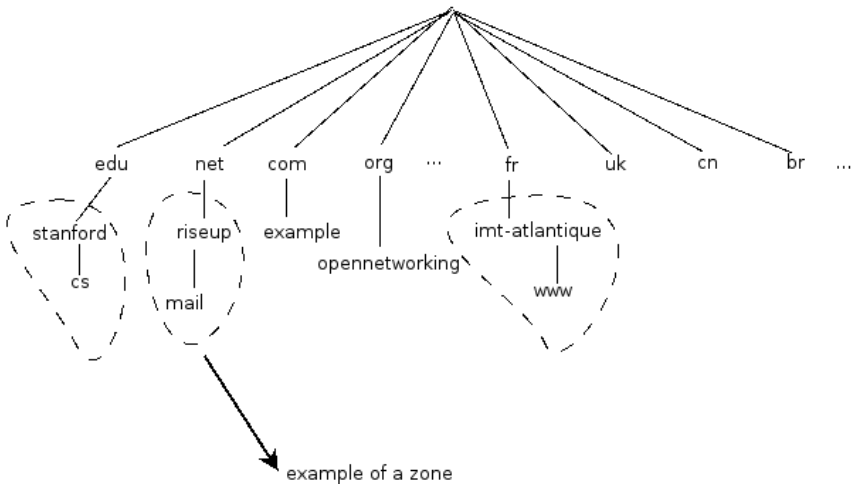

Figure 1: Example of the DNS Name space

Fully Qualified Domain Names (FQDN)) to IP addresses but can also be used for other purposes [8]. DNS is defined in multiple RFCs [1], [2], [3] and more than 20 others. RFCs (Request for Comments) are documents published by the Internet Engineering Task Force [7], Internet community defining Internet standards.

3.1.1. The DNS Name Space

In order to avoid name collisions, the DNS name space is organized in a hierarchical way. The top of the naming hierarchy for the Internet is managed by the ICANN (Internet Corporation for Assigned Names and Numbers), an organization created in 1988 for this purpose. Figure 1 shows some of the so-called top-level domains and several sub-domains, in a tree representation. The leaves of the tree represent domains that have no sub-domains (but can of course contain one or thousands of *records*). Each domain is named by the path upward from it to the (unnamed) root. The components (*labels*) are separated by a dot.

Each domain, sub-domain or *zone* of the tree can be managed by a different authority. In this way, the database containing the name system information is hierarchically divided into non-overlapping zones. Divisions are made by *delegation* of a zone by the managing authority of the immediately upwards zone. The data base corresponding to a zone is managed by the zone's *authoritative name servers*.

The data base of the DNS is formed by the so called *resource records*. These records can be of different types, according to the information they contain. For instance, a record of class internet and type AAAA maps a name into an IPv6 address. In general, a resource record contains 5 fields, namely *owner (domain name), Time to live, Class, Type, rdata (value)*.

Question 3.1.

Within the Internet class, there are several types of records. Investigate the purpose of each type of record and complete Table 1. Write down below the name of further records, and their purpose, if you know more of them.

We have provided here a very brief description of the DNS. Students should refer to other sources such as [8, 9] to further understand this key component of the Internet.

Type	Purpose	Example domain name	Example value
A	Maps a domain name to an IPv4	www.imt-atlantique.fr	192.108.117.237
AAAA			
NS			
MX			
CNAME			
PTR			
SOA			
RRSIG			

Table 1: Common DNS resource records in the Internet class.

3.1.2. The DNS Main Actors

In order to better understand the functioning of the DNS is important to understand -and distinguish- the following terms:

- **Authoritative name server** a server managing a zone, containing the records corresponding to that specific zone.
- **Resolver** server that is queried by a client (for instance your host) and which performs the DNS queries in order to resolve a mapping. DNS queries are addressed to authoritative name servers. Resolvers usually store records into their cache.
- **Resource Record** constitute the DNS database, see description in the previous subsection.
- **Root servers** authoritative name servers maintaining records of the top-level domains.

3.1.3. DNS Security Extensions (DNSSEC)

DNSSEC is a set of extensions that makes it possible for resolvers to validate the **authenticity** and **integrity** of DNS data. The original design of DNS did not consider security aspects, and DNSSEC has been an important effort aiming to include data authentication and integrity. This set of extensions do not provide confidentiality neither availability, though. DNSSEC prevents users to get fake or manipulated data, that could be created by attacks against DNS resolvers such as cache poisoning (See readings above). DNSSEC is mainly defined by three RFCs: [4], [5], and [6].

3.2. The Packet Analyser Wireshark

Wireshark [10] is an opensource network packet analyser. It allows you to capture network packets and to display the packets data in a detailed user friendly way. “You could think of a network packet analyzer as a

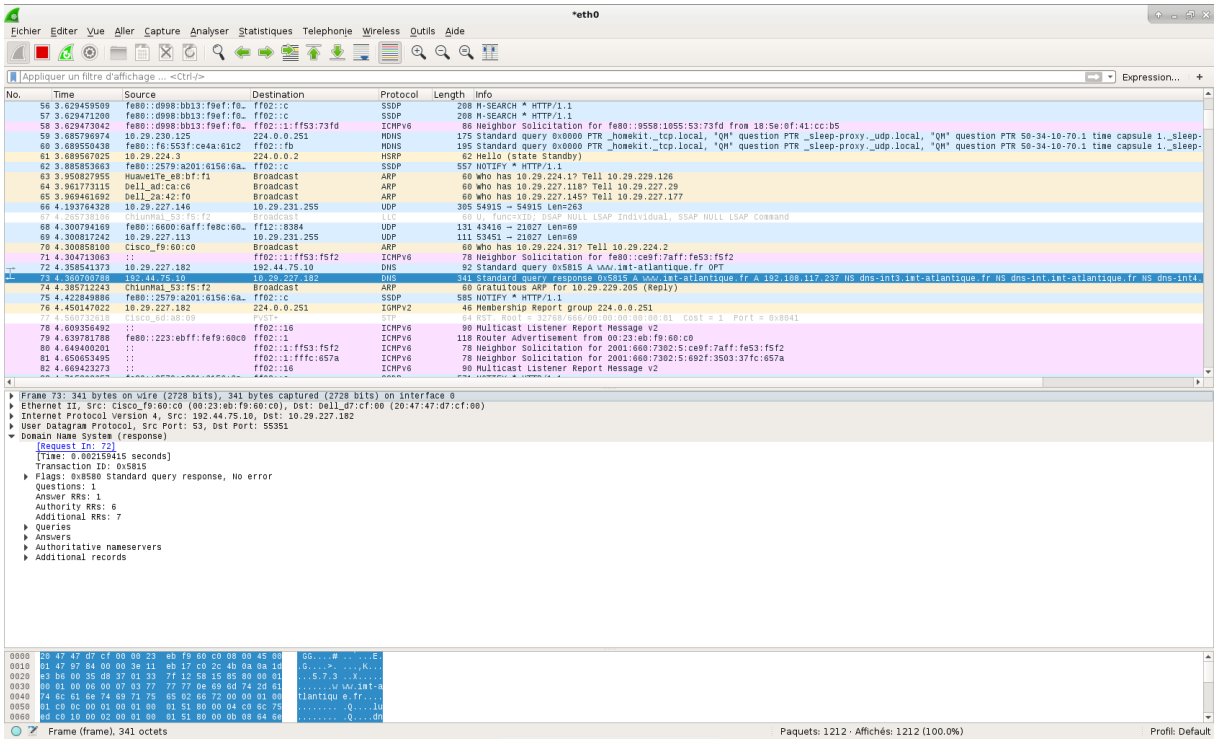

Figure 2: A wireshark packet capture example source [10]

measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course)." [10]

Figure 2 shows an example wireshark capture. We can distinguish the sequence of captured packets, followed by the detailed, human-readable content of the selected packet, finally followed by the packet data in hexadecimal.

3.3. Capturing traffic

In order to capture traffic in a wired interface in a unix system using wireshark make sure to have the adequate privileges (e.g. on Debian and based distributions, the user should be part of the *wireguard* group).

```
wireshark &
```

Once in wireshark, before starting a capture, you must select the interfaces where you want to capture traffic. To do so, click `Capture` → `Options` → select interface `start`.

At any moment you can save one capture to analyse it later on. You can also change the visualisation options, to get more or less columns, activate/desactivates colors, etc.

4. Hands On

4.1. Some useful unix commands.

1) Start the course's VM and open a new terminal in the VM

Question 4.1.

Use the command `ip a` (shortcut for `ip address`) to obtain the information of the network interfaces available in your working VM. Are they IP addresses v4 or v6? Write down the obtained address(es).

Question 4.2.
Which DNS resolver is your working VM using? For obtaining that information show the content of file /etc/resolv.conf. You can for example execute `cat /etc/resolv.conf`

Note that the information from the two previous questions will be useful when you will inspect network packets in Section 3.2.

4.2. DNS basic functioning and commands

Question 4.3.
Use the command `dig -x` to query the domain name associated to the 2620:0:2d0:200::7 address, and write it down. What kind of query (lookup) is it? (Clue: look at `man dig`).

Question 4.4.
Indicate the IPv4 address associated to www.brest.fr. You can again use command `dig` again.

Question 4.5.
Which type of DNS record allows you to obtain such information?

Question 4.6.
Use `dig` to obtain the IPv6 address associated to www.nic.cl. Write down the address and the full command you need to use.

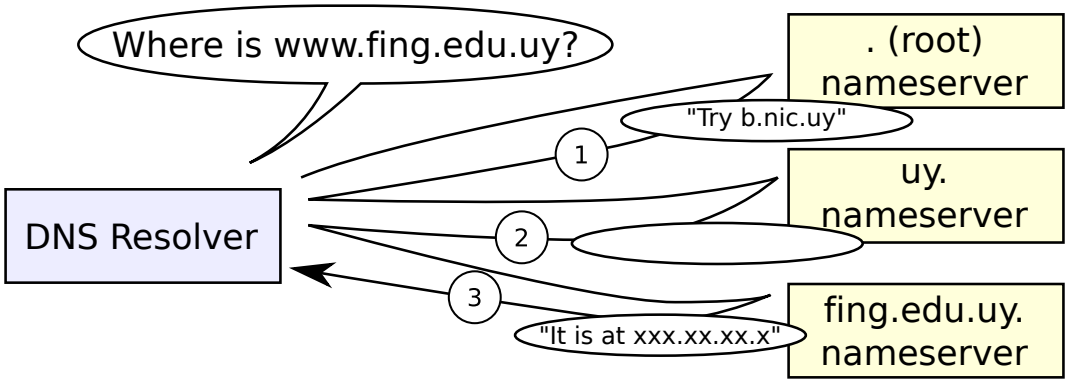

Figure 3: DNS resolution of www.fing.edu.uy. Image based on https://en.wikipedia.org/wiki/File:Example_of_an_iterative_DNS_resolver.svg

Question 4.7.

Query now the A record of www.fing.edu.uy showing all the hierarchy. You can use `dig` with the `+trace` option (along with `+multiline` for a more human-readable output). Who send the messages numbered *1*, *2* and *3* from Figure 3 (Look at the "Received . . . from" lines in the output)? Complete the answers from the `uy.` and `fing.edu.uy` nameservers. What can you conclude about the functioning of the domain service? Explain.

4.3. Analysing DNS traces

We are now going to use Wireshark to see what is going on down the wire.

1) Open a terminal on your virtual machine and type the command `wireshark` to open wireshark. In wireshark, start a new capture in the `eth0` interface.

2) In another terminal, issue a one-time ping (`ping -c 1`) to www.imt-atlantique.fr

3) In Wireshark examine the captured packets.

Question 4.8.

What protocols do you see? Why?

4) Use a visualisation filter to see only the DNS messages.

5) Locate the DNS query and response sent and received due to your previous command.

Question 4.9.
Are them sent over UDP or TCP? What is the destination port for the DNS query message? What is the source port of DNS response message?

Question 4.10.
To what IP address is the DNS query message sent? Compare this to the resolver used by your working VM (see your answers to part 4.1). Are these two IP addresses the same?

6) Now, we will find out the serves managing e-mail for imt-atlantique.fr (you can refer to Table 1). In a terminal use the `dig MX` program to find out the servers exchanging mails for imt-atlantique.fr

Question 4.11.
What information did you get?

7) Observe the captured DNS traces

Question 4.12.
Examine the DNS query message. What "Type" of DNS records does the query ask for? Does the query message contain any "answers"?

Question 4.13.
Examine the DNS response message. What name servers does the response message provide? Does this response message also provide the IP addresses of the name servers?

8) Find out the authoritative name server of the tools.ietf.org domain.

9) Start a new capture, if you have stopped the previous one

10) In a terminal, run the command `dig tools.ietf.org @ip_authoritative server` where `ip_authoritative server` is one of the IPs found out in step **8**)

11) Repeat the command several times.

12) Now, type command `dig tools.ietf.org @1.1.1.1` Repeat the action several times.

13) You can now stop the capture

Question 4.14.
Observe the Wireshark capture. Compare the responses obtained from the authoritative and the recursive servers. In what do they differ? Look at the TTLs. Explain.

4.4. Validating DNS query answers with DNSSEC

For taking advantage of the security provided by DNSSEC, two conditions have to be fulfilled: (1) the domain zone that you want to query about has to be signed, and (2) your DNS resolver must be able to verify the record signatures.

Before answering the questions in this section, check first whether your DNS resolver validates or not the DNS answers. For that you can use two tools:

- The dnssec-failed.org domain, whose **signature is invalid**. Run `dig dnssec-failed.org`, and you should get a **SERVFAIL** message if the resolver validates the answer, or a **NOERROR** message with the full answer if it does not.
- The `http://dnssec.vs.uni-due.de/` website: provides an easy to use "DNSSEC resolver test". Simply click on the `Start test` button.

If you find out the resolver is not DNSSEC-able, compare the query for the *dnssec-failed.org* domain against a validating resolver:

14. Query a validating server, such as `1.1.1.1`, about the domain dnssec-failed.org: `dig dnssec-failed.org @1.1.1.1`.

Question 4.15.
What do you get? Explain.

15. Query now a non-signed domain and a signed domain:

```
dig +dnssec hola.com @1.1.1.1
```

```
dig +dnssec ripe.net @1.1.1.1
```

Question 4.16.
The difference is not very visible for the end user, but compare the **flags** in the **HEADERs** of both answers. What is (or should be) the difference between them (how the answer tells you the answer has **Authenticated Data (AD)** or not)?

16. Query again `ripe.net` using the `dig` command, including the +dnssec and +multiline options.

Question 4.17.
Explain what is the new record and its contents that you can read in the answer.

5. Conclusion

Question 5.1.
What is the purpose of the DNS protocol? Give an answer as complete as possible (which can be brief at the same time).

Question 5.2.
Why does the DNS protocol uses UDP as a transport protocol? What do you think are the advantages and disadvantages for this?

Question 5.3.
What are advantages and disadvantages of a distributed data base, as the one managing domain names?

Question 5.4.
Would you say that DNS is a secure protocol or not? Be as precise as you can in answer. What is DNSSEC for and what does it provide? Does DNSSEC provides confidentiality in the DNS messages exchange?

References

[1] RFC 1034 - DOMAIN NAMES - CONCEPTS AND FACILITIES, The Internet Engineering Task Force, [online] https://tools.ietf.org/html/rfc1034.

[2] RFC 1035 - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION, The Internet Engineering Task Force, [online] https://tools.ietf.org/html/rfc1035.

[3] RFC 2181 - Clarifications to the DNS Specification, The Internet Engineering Task Force, [online] https://tools.ietf.org/html/rfc2181.

[4] RFC 4033 - DNS Security Introduction and Requirements, The Internet Engineering Task Force, [online] https://tools.ietf.org/html/rfc4033.

[5] RFC 4034 - Resource Records for the DNS Security Extensions, The Internet Engineering Task Force, [online] https://tools.ietf.org/html/rfc4033.

[6] RFC 4035 - Protocol Modifications for the DNS Security Extensions, The Internet Engineering Task Force, [online] https://tools.ietf.org/html/rfc4033.

[7] The Internet Engineering Task Force [online] https://ietf.org/about/.

[8] Computer Networks, 5th edition, Andrew S. Tanenbaum, David J? Wetherall, PRENTICE HALL

[9] Les réseaux, 4th edition, Guy Pujolle, EYROLLES

[10] Wireshark network packet analyser [online] https://www.wireshark.org/