TÉCNICO LISBOA

RC 19/20	LAB ASSIGNMENT	Guide.Part #:	6.0
From Application Layer to Network Layer		**Issue Date:**	9 Dec 2019
WiFi (IEEE 802.11) and SSL Protocol Analysis		**Due Date:**	
Author: Prof. Rui Cruz		**Revision:**	1.0

1 Introduction

The objective of the experiments in this Lab assignment is twofold.

Firstly, to investigate the Secure Sockets Layer (SSL) protocol, focusing on the SSL records sent over a TCP connection. You will do that by analyzing a trace of the SSL records sent between a host and an e-commerce server on the Internet. You will investigate the various SSL record types as well as the fields in the SSL messages.

Secondly, to analyse the IEEE 802.11 wireless network protocol, a protocol that you use everyday and almost everywhere (in your smartphone, your laptop, etc.).

In this lab we will use trace files that were previously captured.

Recommendation: review sections 7.3 (WiFi: 802.11 Wireless LANs), and section 8.6 (SSL) of the Book. Additionally, you might want to check the following documents:

- "A Technical Tutorial on the 802.11Protocol" by Pablo Brenner (Breezecom Communications), `http://www.sss-mag.com/pdf/802_11tut.pdf`;
- "ANSI/IEEE Std 802.11, 1999 Edition (R2003)" (in this Lab Support Files). In particular, you may find Table 1 on page 36 of the standard particularly useful when looking through the wireless trace.

Preliminary Notes

The instructions in this document are applicable to Computers at the IST Labs.

Nevertheless, one nice feature of the software stack we are going to use is that it is portable to many platforms including **YOUR OWN** personal computers, running the following Operating Systems:

- Microsoft Windows from version 10 up
- Apple macOS from versions 10.13 'High Sierra' up
- Debian-based Linux, such as Ubuntu (recommended) from versions 12.04 'Precise' up.

It is not recommended to apply this setup to a virtual machine (nested virtualization), although possible, as the configuration requires access to a hypervisor environment (recommended Virtualbox) in the host system.

Before proceeding you should verify if you have a "clean" environment, i.e., no Virtual Machine "instances" running (using precious resources in your system), or inconsistent instances in Vagrant and Virtualbox. For that purpose run the `vagrant global-status` command and observe the results (as in the following example):

```
:~$ vagrant global-status
id       name       provider   state    directory
-------------------------------------------------------------------
28fb48a  mininet    virtualbox poweroff /Users/x/Projects/mininet
f0ccec2  web1       virtualbox running  /Users/x/Projects/multinode
f09c279  web2       virtualbox running  /Users/x/Projects/multinode
```

In the above example, you can observe that there are three Virtual Machine instances, being the first “mininet”, which is powered off, but two “web” servers still running. It is **advisable to halt VMs** if they are running, and then **clean and destroy VMs from previous Lab experiments** that are not related with this Lab, or that are not anymore needed.

Note: Avoid copying text strings from the command line examples or configurations in this document, as pasting them into your system or files may introduce/modify some characters, leading to errors or inadequate results.

2 Setting up the Experimental Environment

For this Lab experiment you should use the same environment created for previous LAB5.

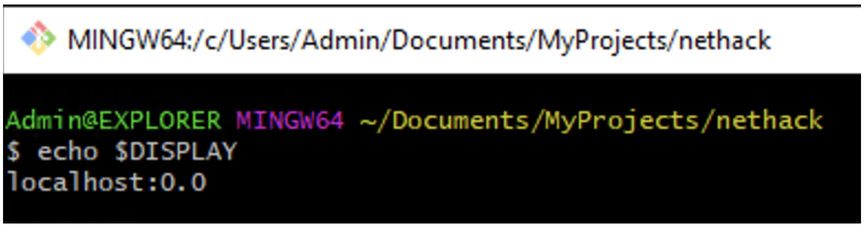
If you destroyed it, then you need to create a project directory, named, for example, **nethack**. In that case, download the file `RC_19_20_LAB_05.6_support_files.zip` from the course website and uncompress the content to the **nethack** folder.

The **nethack** project folder should have a structure similar to the following:

```
.
|__ Vagrantfile
|__ bootstrap-nethack.sh
|__ data
    |__ NAT-ISP-side.pcap
    |__ NAT-home-side.pcap
    |__ dhcp-trace.pcap
    |__ ethernet-ethereal-trace.pcap
    |__ ssl-trace.pcap
    |__ wifi-802-11.pcap
```

For Microsoft Windows hosts, you need to start “**VcXsrv**” before spin off the lab environment. The “**VcXsrv**” application will start a **X-Window**s server that will forward application GUIs running inside the VM to the host system.

Verify also that you have an environment Variable with name **DISPLAY** and with **Variable value** of `localhost:0.0`, as illustrated in Figure 1.



MINGW64:/c/Users/Admin/Documents/MyProjects/nethack

```
Admin@EXPLORER MINGW64 ~/Documents/MyProjects/nethack
$ echo $DISPLAY
localhost:0.0
```

Figure 1: Verify DISPLAY Variable

For the experiments in this lab you will use **Wireshark** (A network Sniffer and Protocol dissector tool, provisioned and configured in the VM) to capture the packets in the network interface in order for you to analyse them.

For that purpose, you will need two Terminal (Bash) windows opened. On the first window start Wireshark with the following command:

```
$ vagrant ssh -c wireshark -- -X
```

In the second Terminal window start the ssh connection with the VM:

```
$ vagrant ssh
```

2.1 Using Shared Folders in Vagrant

One simple way to share information between one virtual machine, designated as ***guest***, and the machine that hosts it, known as the ***host***, is by mapping one folder or directory of the ***host***'s file system to one folder on the ***guest***'s file system.

For the VM in this Lab you have a shared folder, as you can observe in the Vagrantfile in the lines that define them, where the first path is related to the ***host*** and the second path refers to the ***guest***.

```
xxxxx_config.vm.synced_folder "data", "/home/vagrant/data"
```

Having this type of share, you ensure that whatever you have/put on the “**data**” folder in your ***host*** will appear and be available inside the VM in the “data” ***home*** folder. Similarly, whatever is produced as result of commands or programs will be available on the “**data**” folder in your ***host***.

RC 19/20	**LAB ASSIGNMENT**	**Guide.Part #:**	6.0
From Application Layer to Network Layer		**Issue Date:**	9 Dec 2019
WiFi (IEEE 802.11) and SSL Protocol Analysis		**Due Date:**	
Author: Prof. Rui Cruz		**Revision:**	1.0

3 Analysis of an SSL Session

For this experiment load into Wireshark the trace file **ssl-trace.pcap** using the **File** pull down menu, choosing **Open**, and then selecting that trace file from folder **data** (because Wireshark is running in the VM).

The trace file was captured during the process of purchasing an item on a e-commerce site.

3.1 EXP1.1: SSL session

Your Wireshark GUI should be displaying only the Ethernet frames that have SSL records. It is important to keep in mind that **an Ethernet frame may contain one or more SSL records**. Note that this is very different from HTTP, for which each frame contains either one complete HTTP message or a portion of a HTTP message. Also, an SSL record may not completely fit into an Ethernet frame, in which case multiple frames will be needed to carry the record.

When answering the following questions (and those in the next subsections), you should hand in a dissection of the packet(s) within the trace that you used to answer the question asked. To export a packet summary/data, use Use **File** → **Export Packet Dissections** → **As Plain text**, choose **Selected packet only**, choose **Packet summary line**, and select the minimum amount of packet detail that you need to answer the question.

1. For each of the first 8 Ethernet frames, specify the source of the frame (client or server), determine the number of SSL records that are included in the frame, and list the SSL record types that are included in the frame.

2. Each of the SSL records begins with the same three fields (with possibly different values). One of these fields is “content type” and has length of one byte. List all three fields and their lengths.

3.2 EXP1.2: ClientHello Record

4. Expand the **ClientHell**o record. (If your trace contains multiple **ClientHello** records, expand the frame that contains the first one.) What is the value of the content type?

5. Does the **ClientHello** record contain a **nonce** (also known as a “challenge”)? If so, what is the value of the challenge in hexadecimal notation?

6. Does the **ClientHello** record advertise the cypher suites it supports? If so, in the first listed suite, what are the public-key algorithm, the symmetric-key algorithm, and the hash algorithm?

3.3 EXP1.3: ServerHello Record

7. Locate the **ServerHello** SSL record. Does this record specify a chosen cipher suite? What are the algorithms in the chosen cipher suite?

8. Does this record include a **nonce**? If so, how long is it? What is the purpose of the client and server **nonces** in SSL?

9. Does this record include a session ID? What is the purpose of the session ID?

10. Does this record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame?

3.4 EXP1.4: Client Key Exchange Record

11. Locate the client key exchange record. Does this record contain a pre-master secret? What is this secret used for? Is the secret encrypted? If so, how? How long is the encrypted secret?

3.5 EXP1.5: Change Cipher Spec Record (sent by client) and Encrypted Handshake Record

12. What is the purpose of the **Change Cipher Spec** record? How many bytes is the record in your trace?

13. In the encrypted handshake record, what is being encrypted? How?

14. Does the server also send a change cipher record and an encrypted handshake record to the client? How are those records different from those sent by the client?

3.6 EXP1.6: Application Data

15. How is the application data being encrypted? Do the records containing application data include a MAC? Does Wireshark distinguish between the encrypted application data and the MAC?

16. Comment on and explain anything else that you found interesting in the trace.

4 WiFi Protocol Experiment

For this experiment load into Wireshark the trace file **wifi-802-11.pcap** using the **File** pull down menu, choosing **Open**, and then selecting that trace file from folder **data** (because Wireshark is running in the VM).

The trace file was captured on a network composed of a Linksys 802.11g combined access-point (AP)/router, with two wired PCs and one wireless host PC attached to the AP/router. In the trace file, we will see frames captured on channel 6. Since the host and the AP that we are interested in are not the only devices using channel 6, we will see a lot of frames that we are not interested in for this lab, such as beacon frames advertised by a neighbor's AP also operating on channel 6.

The wireless host activities taken in the trace file are:

- The host is already associated with the *30 Munroe St* AP when the trace begins.
- At t = 24.82, the host makes an HTTP request to `http://gaia.cs.umass.edu/wireshark-labs/alice.txt`.
  The IP address of *gaia.cs.umass.edu* is 128.119.245.12.
- At t=32.82, the host makes an HTTP request to `http://www.cs.umass.edu`, whose IP address is 128.119.240.19.
- At t = 49.58, the host disconnects from the *30 Munroe St* AP and attempts to connect to the `linksys_ses_24086`. This is not an open APt, and so the host is eventually unable to connect to this AP.

4.1 EXP2.1: Beacon Frames

Recall that **beacon** frames are used by an IEEE 802.11 AP to advertise its existence. To answer some of the questions below, you'll want to look at the details of the "IEEE 802.11" frame and subfields in the middle Wireshark window.

When answering a question below, you should hand in a dissection of the packet(s) within the trace that you used to answer the question asked. To export a packet summary/data, use Use **File** → **Export Packet Dissections** → **As Plain text**, choose **Selected packet only**, choose **Packet summary line**, and select the minimum amount of packet detail that you need to answer the question.

1. What are the SSIDs of the two access points that are issuing most of the beacon frames in this trace?
2. What are the intervals of time between the transmissions of the beacon frames of the the `linksys_ses_24086` access point? And from the *30 Munroe St.* access point? (Hint: this interval of time is contained in the beacon frame itself).
3. What (in hexadecimal notation) is the source MAC address on the beacon frame from *30 Munroe St*? Recall from Figure 7.13 in the Book (reproduced here in Figure 2) that the source, destination, and BSS are three addresses used in an IEEE 802.11 frame. For a detailed discussion of the 802.11 frame structure, see section 7 in the IEEE 802.11 standards document.

4. What (in hexadecimal notation) is the destination MAC address on the beacon frame from *30 Munroe St*?

5. What (in hexadecimal notation) is the MAC BSS ID on the beacon frame from *30 Munroe St*?

6. The beacon frames from the *30 Munroe St* access point advertise that the access point can support four data rates and eight additional "extended supported rates". What are these rates?

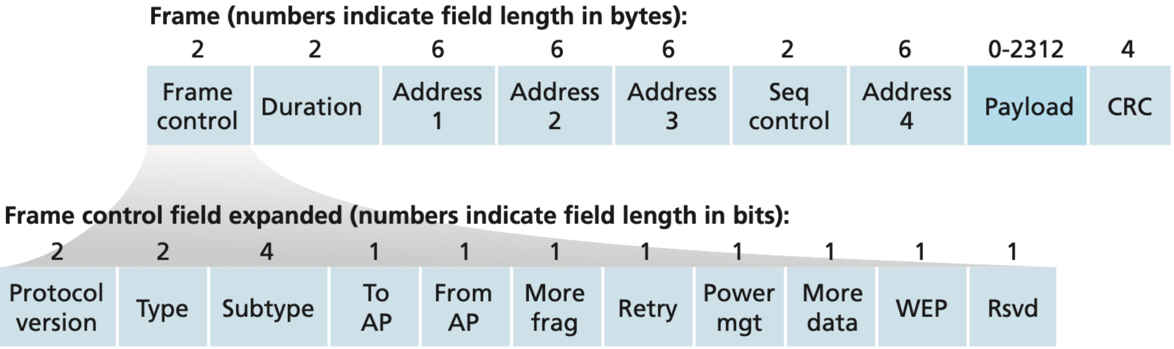


Figure 2: The IEEE 802.11 frame

4.2 EXP2.2: Data Transfer

Since the trace starts with the host already associated with the AP, let's first look at data transfer over an IEEE 802.11 association before looking at AP association/disassociation. Recall that in this trace, at t = 24.82, the host makes an HTTP request to `http://gaia.cs.umass.edu/wireshark-labs/alice.txt`. The IP address of *gaia.cs.umass.edu* is 128.119.245.12. Then, at t=32.82, the host makes an HTTP request to `http://www.cs.umass.edu`.

7. Find the IEEE 802.11 frame containing the **SYN** TCP segment for this first TCP session (that downloads alice.txt). What are three MAC address fields in the 802.11 frame?

8. Which MAC address in this frame corresponds to the wireless host (give the hexadecimal representation of the MAC address for the host)? And the MAC address of the AP? And the MAC address the first-hop router?

9. What is the IP address of the wireless host sending this TCP segment? What is the destination IP address? Does this destination IP address correspond to the

host, AP, first-hop router, or some other network-attached device? Explain what you found.

10. Find the IEEE 802.11 frame containing the **SYNACK** segment for this TCP session. What are three MAC address fields in the IEEE 802.11 frame?
11. Which MAC address in this frame corresponds to the host? And which one is the MAC address of the AP? And which one is the MAC address of the first-hop router?
12. Does the sender MAC address in the frame correspond to the IP address of the device that sent the TCP segment encapsulated within this datagram? (Hint: review Figure 6.19 in the Book if you are unsure of how to answer this question, or the corresponding part of the previous question. It is particularly important that you understand this).

4.3 EXP2.3: Association/Disassociation

Recall from Section 7.3.1 in the Book that a host must first associate with an access point before sending data. Association in IEEE 802.11 is performed using the **ASSOCIATE REQUEST** frame (sent from host to AP, with a frame type **0** and subtype **0**, see Section 7.3.3 in the book) and the **ASSOCIATE RESPONSE** frame (sent by the AP to a host with a frame type 0 and subtype of **1**, in response to a received **ASSOCIATE REQUEST**). For a detailed explanation of each field in the IEEE 802.11 frame, see page 34 (Section 7) of the IEEE 802.11 specification.

13. What two actions are taken (i.e., frames are sent) by the host in the trace just after t=49, to end the association with the *30 Munroe St* AP that was initially in place when trace collection began? (Hint: one is an IP-layer action, and one is an IEEE 802.11-layer action).
14. Looking at the IEEE 802.11 specification, is there another frame that you might have expected to see, but do not see here?
15. Examine the trace file and look for **AUTHENTICATION** frames sent from the host to an AP and vice versa. How many **AUTHENTICATION** messages are sent from the wireless host to the `linksys_ses_24086` AP (which has a MAC address of `Cisco_Li_f5:ba:bb`) starting at around t=49?.
16. Does the host want the authentication to require a key or to be open?
17. Do you see a reply **AUTHENTICATION** from the `linksys_ses_24086` AP in the trace?

18. Now let's consider what happens as the host gives up trying to associate with the `linksys_ses_24086` AP and now tries to associate with the *30 Munroe St* AP. Look for **AUTHENTICATION** frames sent from the host to and AP and vice versa. At what times are there an **AUTHENTICATION** frame from the host to the *30 Munroe St.* AP, and when is there a reply **AUTHENTICATION** sent from that AP to the host in reply? (Note that you can use the filter expression:
    `wlan.fc.subtype==11 and wlan.fc.type==0 and wlan.addr==IntelCor_d1:b6:4f`
    to display only the **AUTHENTICATION** frames in this trace for this wireless host.)

19. An **ASSOCIATE REQUEST** from host to AP, and a corresponding **ASSOCIATE RESPONSE** frame from AP to host are used for the host to associated with an AP. At what time is there an **ASSOCIATE REQUEST** from host to the *30 Munroe St* AP? When is the corresponding **ASSOCIATE REPLY** sent? (Note that you can use the filter expression:
    `wlan.fc.subtype<2 and wlan.fc.type==0 and wlan.addr==IntelCor_d1:b6:4f`
    to display only the **ASSOCIATE REQUEST** and **ASSOCIATE RESPONSE** frames for this trace.)

20. What transmission rates is the host willing to use? And the AP? To answer this question, you will need to look into the parameters fields of the IEEE 802.11 wireless LAN management frame.

4.4 EXP2.4: Other Frame types

The trace contains a number of **PROBE REQUEST** and **PROBE RESPONSE** frames.

21. What are the sender, receiver and **BSS ID** MAC addresses in these frames? What is the purpose of these two types of frames? (To answer this last question, you will need to dig into the references cited earlier in this Lab).

5 Finishing your Experiments

In order to stop the Virtual Machine and to verify the global state of all active Vagrant environments on the system, we can issue the following commands:

```
:~$ vagrant halt
==> nethack: Attempting graceful shutdown of VM...
:~$ vagrant global-status
```

Confirm that the status of the VM is 'powered off'. In order to prevent the machine to keep using resources.

RC 19/20	LAB ASSIGNMENT	Guide.Part #:	6.0
From Application Layer to Network Layer		Issue Date:	9 Dec 2019
WiFi (IEEE 802.11) and SSL Protocol Analysis		Due Date:	
Author: Prof. Rui Cruz		Revision:	1.0

6 Submitting the Results of the Experiments

The experiments that you execute in this LAB will either produce results that you need to report or from which you will be asked questions about the execution. In order to report the results you achieved, proceed as follows:

6.1 General Procedure

The procedure for submission is quite simple:

1. In your Google Classroom for this course, you will find an Assignment for reporting this specific Lab experiment (with a Due Date);
2. The Assignment provides a Link to the TSUGI CLOUD Web Form with Exercise Questions;
3. When you are prepared with the requested materials (screen-shots, command line outputs, developed code, etc.) you will submit the items into the respective Exercise Form questions of the Assignment;
4. In the same Exercise Form of the Assignment you may be asked to comment your submissions;
5. Please note that in some types of Exercises you may also be asked to evaluate (anonymously) and provide feedback to the answers from some of your classmates (this peer-grading feedback counts for your grading).
6. When finished answering the Exercise Form, click the button **Done** in the top left of the Form; You will be returned to the Google Classroom Assignment;
7. In the Assignment, do not forget to confirm (click the button) **MARK AS DONE** or **TURN IN** when the assignment is completed;

6.2 Specific Procedure

For this LAB Assignment you will provide answers to the questions related to to the questions related to the analysis of SSL Protocol and the analysis of WiFi Protocol, as follows:

6.3 EXP1.1: SSL session

1. Report (content in plain text) your analysis with the answers to Questions 1 and 2, to submit in the TSUGI Form where asked;

RC 19/20	**LAB ASSIGNMENT**	**Guide.Part #:**	6.0
From Application Layer to Network Layer		**Issue Date:**	9 Dec 2019
WiFi (IEEE 802.11) and SSL Protocol Analysis		**Due Date:**	
Author: Prof. Rui Cruz		**Revision:**	1.0

6.4 EXP1.2: ClientHello Record

1. Report (content in plain text) the analysis of the ClientHello records, with the answer to Questions 4 to 6, to submit in the TSUGI Form where asked;

6.5 EXP1.3: ServerHello Record

1. Report (content in plain text) the analysis of the ServerHello records, with the answer to Questions 7 to 10, to submit in the TSUGI Form where asked;

6.6 EXP1.4: Client Key Exchange Record

1. Report (content in plain text) your analysis of the Key Exchange Record, with the answers to Question 11, to submit in the TSUGI Form where asked;

6.7 EXP1.5: Change Cipher Spec Record and Encrypted Handshake Record

1. Report (content in plain text) your analysis of the Encrypted Handshake Record, with the answers to Questions 12 to 14, to submit in the TSUGI Form where asked;

6.8 EXP1.6: Application Data

1. Report (content in plain text) your analysis of the Application Data, with the answers to Questions 15 to 16, to submit in the TSUGI Form where asked;

6.9 EXP2.1: Beacon Frames

1. Report (content in plain text) your analysis of the Beacon frames, with the answers to Questions 1 to 6, to submit in the TSUGI Form where asked;

6.10 EXP2.2: Data Transfer

1. Report (content in plain text) your analysis of the Data Transfer, with the answers to Questions 7 to 12, to submit in the TSUGI Form where asked;

6.11 EXP2.3: Association/Disassociation

1. Report (content in plain text) the Association/Dissociation, with the answer to Question 13 to 20, to submit in the TSUGI Form where asked;

RC 19/20	**LAB ASSIGNMENT**	**Guide.Part #:**	6.0
From Application Layer to Network Layer		**Issue Date:**	9 Dec 2019
WiFi (IEEE 802.11) and SSL Protocol Analysis		**Due Date:**	
Author: Prof. Rui Cruz		**Revision:**	1.0

6.12 EXP2.4: Other Frame types

1. Report (content in plain text) your analysis of Other Frame types, with the answers to Question 21, to submit in the TSUGI Form where asked;

WARNING. Submissions MUST BE MADE in the TSUGI CLOUD Web Form through Classroom. No other type of submission will be considered for evaluation.