Assignment 1
8-04-2019

Programmazione di Reti

Ing. Chiara Contoli, PhD
chiara.contoli@unibo.it

Corso di Laurea Triennale in
Ingegneria e Scienze Informatiche

Homework

Goal. The goal of this assignment is twofold:

1. show ability in using Wireshark to understand *utilities* and *protocols* behavior;
2. investigate carefully an FTP session

The **FTP** protocol is used to transfer files, both in download and upload, between (typically) remote hosts. To achieve this goal, it leverages to distinct connections:

- Command connection
- Data connection

both provided over TCP protocol.

Your assignment

The assignment contains two parts: practice with Wireshark (*Task1*) and questions and answers session (*Task2*).

Task1: In this task, you are asked to use Wireshark to carefully analyze an FTP session considering that FTP can open the data connection in two different ways:

- ACTIVE mode
- PASSIVE mode

In this task, you will:

1. **use Wireshark to capture** traffic generated by *FTP* utility running on **your personal pc**. To do this, run Wireshark and start listening on the *proper* interface;
2. **start an FTP session** from the Terminal on your personal pc to ftp.ubuntu.com server and perform the sequence of command described in next slide

Your assignment

Phase-0: Before starting the FTP session, open Wireshark and capture on the desired network interface (i.e., beware of the interface type, depending if you are using a WiFi connection or an Ethernet connection). In order to better analyze the FTP session, insert proper filter in order to display **only** those packets belonging to the FTP session.

Phase-1: Once Wireshark is started, open your Terminal and start an FTP session to *ftp.ubuntu.com*; insert username (*anonymous*) and password (can be anything, including no password).

Phase-2: Once you are connected to the FTP server (command line is as follow: *ftp>*) enter, exactly in this order, the following:

1. ***?***: the *question mark* allows to display which command, i.e., actions, can be performed on the FTP server
2. ***ls***: this command allows to display the content of the current folder
3. ***cd <folder>***: adopt this command to enter the folder displayed at step 2
4. ***ls***: enter again this command
5. ***get _ls-lR.gz***: this command allows the download of ls-lR.gz
6. ***exit (or bye)***: this command allows terminate FTP session

Your assignment

Task2: in this task, you are asked to answer questions related to Task1. In order to answer the questions properly, you might have to carry out those Phase multiple times

NOTE. FTP is subject to TCP timeout connection; therefore, when taking your time to analyze and reason about each Phase, this might result in a timeout connection expiration, that can be verified both on Wireshark and on Terminal. On the Terminal, you will face the following output

421 Timeout.

In such situation, you will have to:

1. Exit FTP session from the Terminal (command exit | bye)
2. Stop Wireshark capture and start it again
3. Perform the Phases again

Your assignment

Questions about Phase-0:

1. Which is the most appropriate filter that allows you to display *only* traffic belonging to the FTP session? To answer this question, consider that there are multiple filters that you can adopt to achieve such goal, but one of those filters is the most suitable one.

Questions about Phase-1:

1. Which is the IP address of the Ubuntu FTP server?
2. Which is the TCP port adopted by your FTP client in order to start the TCP three-way-handshake to the Ubuntu FTP server?
3. Which are the messages, Request and Response, exchanged during this Phase? Describe who sends this message to who (i.e., from client to server, or vice-versa). Also, indicate: i) if those messages belongs to the command or to the data connection; ii) which TCP ports are used during this exchange.
4. Is there any message exchanged on the data connection? Hint: to answer this question, consider to change your filter in *ip.addr == <ubuntu_ftp_server_IP_Addr>*

Your assignment

Questions about Phase-2: to answer the following questions, consider to filter the capture on Ubuntu FTP server IP address. For each step of this Phase (see slide 4), answer the following question:

1. Which messages, if any, are exchanged between client and server?
2. On which connection, command or data, are those message exchanged?
3. In case messages are exchanged on data connection:
    a. Is data connection opened in ACTIVE or PASSIVE mode? Please, motivate your answer by telling your consideration
    b. Whether connection is ACTIVE or PASSIVE, which TCP port is used to establish the connection?
4. For each connection, command and data, list all messages (request and response) being exchanged, describing the meaning of each message.

What to submit (and how)

Whoever fails in following these simple rules will incur a penalty in the grading process.

Submit by your institutional e-mail a zip file of a directory containing the **whole** assignment. Mail sent from others e-mail will not be considered. The *object* of the e-mail *must* indicate *which* assignment you are submitting and *all* group members full name. Also, whoever of the group member submit the assignment *must* put in CC *all* the other member of the group.

The directory **must** contain:

1. Two saved Wireshark capture (one capture per Task);
2. A report of the **whole** assignment in *.pdf* format(i.e., both Task1 and Task2); report has to be written in *neat* and *clear* Italian.

What to write in the Report (1/2)

Length is not always a synonymous of quality: keep your report relevant to the topic you are required to write about, and try to be clear and effective in your explanation.

For **Task1**, report **must** contain:

- A description of how you carried out each phase.
- Screenshots of:
  - Wireshark interface that show the traffic of interest properly filtered and packets FTP header content (consider to screenshot the first 2 main Wireshark windows).
  - Terminal: showing each Phase.

For **Task2**, report **must** contain:

- For each Phase, answers to each question, preferably supported by Wireshark and Terminal screenshots