Network Security
Summer 2015
Exercise 3

Prof. Dr.-Ing. Matthias Hollick
Secure Mobile Networking Lab — SEEMOO
`https://www.seemoo.de`

Goal

This hands-on exercise aims to familiarize you with security protocols on the transport layer as well as on the network layer. To solve this exercise, you will dissect protocol traces using a network analyzer (Wireshark).

Deadline

The hard deadline for this exercise is **Monday 25th May, 2015, 23:00:00**. Late submissions are subject to the following penalty: (1) up to 1 day late: you will obtain 50 % of the achieved points; (2) up to 2 days late: you will obtain 25 % of the achieved points; (3) more than 2 days late: you will obtain zero points.

Bonus system

We decided to install a credit-based bonus system in our course. We will hand out credits in certain exercises if you (the students) deliver first-rate performance. Within the relevant exercises we will document the detailed requirements to obtain the bonus. Throughout the entire course 280 credits can be obtained in our bonus system. If you score at least 230 credits, you are eligible for a 0.7 grade bonus in the final exam. If you score between 190 and 229 credits, you are eligible for a 0.3 grade bonus. Below 190 credits we will not issue any bonus.

In this exercise, you can obtain up to *45 bonus credits* if you analyze the TLS trace we provide according to the tasks in Problem 3.1, *and* if you capture and annotate a novel, original, and security related Wireshark trace according to Problem 3.2.

How to solve this exercise

You will need to install Wireshark on your system (`https://www.wireshark.org/download.html`) to capture traces or to work on the provided traces.

Please hand in the solutions to the individual tasks in moodle as usual. This time, we accept plain ASCII text or PDF for the answers; if you want to upload a trace (Problem 3.2) please compress it as *.zip. The naming conventions are: ex03_lastname_firstname.{txt,pdf,zip}.

Acknowledgements

The idea and major structure for this exercise has been adapted from J. F. Kurose and K. W. Ross, “Computer Networking: A Top-down Approach”.

Problem 3.1 Analyzing a TLS trace (Bonus: 15 credits)

In this problem, we will investigate the Transport Layer Security (TLS) protocol, focusing on the TLS records sent over a TCP connection. We will do so by analyzing a trace of the TLS records sent between a host and a TLS-enabled server. We will investigate the various TLS record types as well as the fields in the TLS messages.

Getting the trace

For the first part of this exercise we provide a sample trace which will be the basis for the following tasks.
You can download it from `https://moodle.informatik.tu-darmstadt.de/mod/resource/view.php?id=26499`

Get an overview of the trace

Start Wireshark and load the sample trace. Use Wireshark to display only the Ethernet frames that have TLS records. It is important to keep in mind that an Ethernet frame may contain one or more TLS records (this is very different from HTTP, for which each frame contains either one complete HTTP message or a portion of a HTTP message). Also, a TLS record may not completely fit into an Ethernet frame, in which case multiple frames will be needed to carry the record.

Task 1.1: Which version of TLS is used in the trace? Who can decide which version should be used for the connection and in which record can it be announced?

Task 1.2: Draw a timing diagram which contains all major records used in the trace to setup the first TLS connection.

Client Hello

Task 2.1: Search for the first Client Hello record. Determine the destination ip address of the Client Hello record and try to find out to which company this ip address belongs.

Task 2.2: Does the ClientHello record contain a nonce? If so, what is the value of the nonce in hexadecimal notation?

Task 2.3: Does the ClientHello record advertise the cipher suites it supports? If so, list the public-key algorithm, the symmetric-key algorithm and the hash algorithm of the second listed suite.

ServerHello Record

Task 3.1: Search for the ServerHello TLS record. Does this record specify a chosen cipher suite? What are the algorithms in the chosen cipher suite?

Task 3.2: Does this record include a nonce? If so, how long is it? What is the purpose of the client and server nonces in TLS?

Task 3.3: Take a look at the Extensions of the ServerHello record. What is the purpose of the Next_Protocol_Negotiation Extension? Why is this extension field empty in the ClientHello record?

Task 3.4: Locate the Certificate record from the server. How many certificates are included in this record? Who are the subject and the issuer of the certificates? How are the certificates related to each other?

Client Key Exchange Record

Task 4.1: Locate the client key exchange record. Does this record contain a pre-master secret? If so, how long is this secret? If not, how does the key exchange work?

Session Ticket

Task 5.1: What is the purpose of having the server send a Session Ticket instead of a Session-ID, as it was the case in early SSL versions?

Change Cipher Spec Record

Task 6.1: What is the purpose of the Change Chiper Spec Record?

Task 6.2: Locate the Change Chipher Spec Record sent by the client. How many bytes are necessary for the Change Chipher Spec record in the trace?

Encrypted Handshake Record

Task 7.1: There are two encrypted handshake records. Do the client and server encrypted handshake records differ from each other? What is being encrypted in both of them and why?

Application Data

Task 8.1: Is the application data being encrypted? If so, how is it encrypted and do the records containing application data include any MAC (Message Authentication Code)? Can Wireshark distinguish between the encrypted data and such a MAC?

Problem 3.2 Capturing and annotating a security-related trace file (Bonus: 30 credits)

Task 9.1: You can obtain a bonus of 30 credits, if you capture and annotate a *novel*, *original*, and *security related* Wireshark trace. Please select a security protocol of your choice *other than TLS/SSL* (examples: DNSSEC, SSH, IPSec, your own VPN client, etc.).

Capture a trace that shows the operation of this protocol and explain the protocol interaction with bullets (what happens in which message). Each trace needs to be original, i.e., neither do we want you to download an existing trace from the web, nor to hand in group solutions. Everyone should capture an original trace. Hand in the trace together with the description as a *.zip file following our usual naming conventions and upload it to our course portal. *Please ensure that no personal passwords, etc. are part of the trace in plaintext.*