Assignment 2

08.10.2014

RSA

Objective:

Do a secure TCP bridge between two computers (server & client). The server will be configured to send traffic received on port Y (encrypted), to a target computer (deciphered) (target computer and port are set on the server, can be hard-coded). When the client computer receives a connection on port X, it will then open a ciphered TCP connection to the server on port Y and forward data. Once the connections are established, traffic will be asymmetric like in any classical TCP connections.

RSA will provide a secure connection between the client and the server. You can encrypt all the traffic with RSA or use RSA only to share an initial symmetrical session key (more speed). Client and Server will know the public key of each other.

Application Layout:

A: A client application (e.g.: browser)
C: The RSA client (listening on port X, e.g.: 8080)
S: The RSA server (listening on port Y, e.g.: 20000)
T: Target of the TCP connection (e.g.: website, google.com)

A → C:8080 [encrypt] → (traffic is encrypted here) → S:20000 [decrypt] → google.com:80

You can do the demonstration using a single machine by running both C and S on the localhost, and then having the application connect to localhost:X (e.g.: http://localhost:8080/).

You can also demonstrate the bridge using two computers, and/or any other TCP protocol (e.g.: https, ftp, etc).

Hand-in:

A demonstration of the software.
The source files and project files should be compressed and submitted to the Ilias page of the course.

Notes:

The RSA algorithm must be implemented manually, and the RSA server application must be able to generate the RSA key-pair.

References:

http://en.wikipedia.org/wiki/Rsa
http://www.labri.fr/perso/betrema/deug/poly/crypto.html

Deadline:

22 October 2014 (2 weeks)