Descrierea proiectului

Capitolul 1.1 – Descrierea proiectului

Pentru a putea incepe cu analiza protocolului TLS am decis sa incep proiectul mai intai cu o comunicatie simpla de client-server in java. Am ales sa incep proiectul astfel deoarece am dorit sa arat de ce avem nevoie protocolul de TLS si cum putem sa il aplicam intr-o aplicatie deja existenta dar cu o comunicatie nesigura.

Pentru crearea aplicatiei client-server cu comunicatie nesigura am create 2 clase : EchoServer si EchoClient din pachetele: com.tls.analysis.echo.server si respectiv com.tls.analysis.echo.client.

Capitolul 1.1.1.1 – Descrierea aplicatie EchoServer:

Incepem aplicatia prin a ne importa clasele necesare:

Text

Description automatically generated

In interiorul clasei EchoServer avem definit portul la care dorim sa ascultam pentru conexiuni ca si variabila statica finala de tip int.

In interiorul functiei main incercam sa creem un ServerSocket denumit sugestiv serverSocket pentru a asculta conexiuni dar si o variabila de tip Socket pe care am denumit-o client. Aceasta variabila am asignat-o prin apelul functiei serverSocket.accept() care intoarce un socket nou cu conexiunea creata.

Dupa ce conexiunea a fost creata cu succes, incepem sa citim ceea ce client-ul doreste sa ne comunice cat timp sesiunea este activa.



Mai multe functii din main arunca exceptii de tip IO, aceastea sunt subliniate cu rosu, astfel ca am introdus totul intr-un try catch. :

Text

Description automatically generated

Capitolul 1.1.1.2 – Rularea aplicatiei echoServer:

Pentru rularea aplicatiei in InteliJ, apasam cu click stanga pe iconita verde din stanga clasei si apoi run “Numele\_aplicatiei”.

Graphical user interface, text, application

Description automatically generated

Daca totul a decurs cum trebuie acesta este mesajul pe care am trebuie sa il vedem in consola:

Text

Description automatically generated

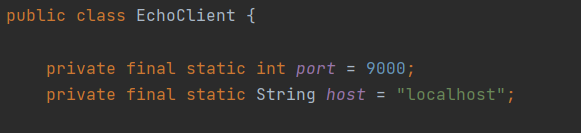
Capitolul 1.1.2.1 – Descrierea aplicatiei EchoClient

Ca si in cazul aplicatiei EchoServer incepem prin a ne importa clasele necesare:

Text

Description automatically generated

In interiorul clasei Echo client avem de aceasta data 2 variabile definite port-ul si adresa la care dorim sa ne conectam:



In interiorul functiei main incercam sa cream un socket cu conexiunea de la adresa si port-ul definite anterior si daca reuseste, acesta trimite un mesaj catre server pana cand este introdus mesajul “exit”. De asemenea, multe dintre functiile apelate in interiorul functiei main arunca exceptii de tip IO asa ca le-am introdus si pe acestea intr-un try-catch:

Text

Description automatically generated

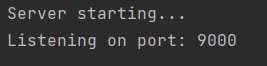
Capitolul 1.1.2.2 – Rularea aplicatiei EchoClient

Pentru a rula aplicatia EchoClient trebuie mai intai rulata aplicatia EchoServer.

Daca totul decurge bine in consola ar trebui sa vedem ca acum avem 2 tab-uri, fiecare consola cu aplicatia sa.



In consola EchoServer ar trebui sa vedem mesasjul acesta:



In consola EchoClient ar trebui sa vedem mesajul acesta:

Text

Description automatically generated

Acest mesaj practice ne confirma ca ne-am conectat la server si ca putem sa incepem sa comunicam cu server-ul.

O data trimis un mesaj, putem verifica pe server daca acesta a fost primit.

Text

Description automatically generated

Text

Description automatically generated

Capitolul 1.1.3 – De ce avem nevoie de protocolul TLS?

Cand initiem o astfel de comunicatie date noastre sunt trimise in “plain-text”, adica datele noastre sunt expuse catre oricine ar dori sa vada comunicatia.

Pentru a demonstra acest lucru am dorit sa interceptez traficul de date folosind aplicatia “WireShark” si ruland aplictia client-server pe o masina virtuala de linux.

Am inceput prin a rula cele 2 aplicatii server si client si apoi am inceput sa interceptez traficul de pe localhost cu WireShark cu un filtru pe portul pe care comunica cele 2 aplicatii.

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

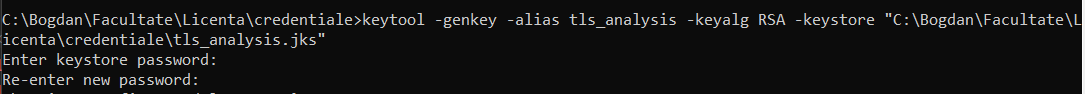
Un atacator poate vedea astfel ce mesaje trimite clar client-ul catre server.

Pentru a crea un KEYSTORE:

Rulam comanda

Keytool -genkey -alias tls\_analysis -keyalg RSA -keystore “C:\Bogdan\Facultate\Licenta\credentiale\tls\_analysis.jks”

Ne va cere sa ne alegem o parola:



Apoi ni se va cere sa introducem datele de identificare:

Text

Description automatically generated

Apoi ni se va cere sa confirmam corectitudinea datelor introduse si odata confirmata aceasta, cheia va fi generata.

A picture containing text

Description automatically generated