DSP-LAB 2016-17 Secure Java Secure Native Interface using Intel SGX

Clindo Devassy K, Subhadeep Manna

Instructions for running the Java-Server on Ubuntu 16.xx vanilla

Prerequisites

1. Installation of **Oracle Java Development Kit 1.8** on vanilla Ubuntu.

2. Installation of **Intel SGX_SDK** for Linux.

   Preferred path is: **'/opt/intel/sgxsdk/'**

3. Add the environment variable for SGX simulation as follows:
   $export LD_LIBRARY_PATH=**'/opt/intel/sgxsdk/sdk_libs'**

4. **Optional**: Change the JNI path setting in the make file. If JDK is installed on custom folder.

Setting Up and Running the Java SGX Server

1. Copy the project "**Server**" to a directory in the Machine
   From
   \\DSP_LAB_2016_17\Final_Project\JavaSGX_Client_Server\Server\
   present in the repository.

2. Go to the following directory:

   \\DSP_LAB_2016_17\Final_Project\JavaSGX_Client_Server\Server\src\Java_Src_Server

   To Start the SGX Server the following steps would be needed: -

   a. Give permissions to execute the script "**./Server_Script**".
      $ chmod 777 Server_Script

b. Run the Script using

```
$ ./Server_Script.sh
```

3. After the server starts it waits for the Clients to connect on port 9999.

Instructions for running the Client on Ubuntu 16.xx vanilla

Prerequisites

1. Installation of **Oracle Java Development Kit 1.8** on vanilla Ubuntu.

Setting Up and Running the Java SGX Client

2. Copy the project “**Client**” to a directory in the Machine
   From
   \\DSP_LAB_2016_17\Final_Project\JavaSGX_Client_Server\Server\src\Client
   present in the repository.

   To Start the Client the following steps would be needed: -

   c. Give permissions to execute the script “**./Client_Script**”.

   ```
   $ chmod 777 Client_Script
   ```

   d. Run the Script using

   ```
   $ ./ Client_Script.sh
   ```

3. The Client connects to a SGX Java Server running on port 9999.

Additional Info: Running bash script in some Linux systems may need installing dos2unix

```
$sudo apt-get install dos2unix
```

```
$dos2unix <Script_Name.sh>
```

For Tunnelling

Before each session do the tunnelling

```
1. ssh -L 9000:localhost:9000  <username>@128.211.1.26
2. logout
3. ssh -L 9000:128.10.130.58:9000 <username>@128.211.1.26
```

Note: Here <username> is username and password for Zed.cs.purdue.edu