SAP Exam – July 2024

Java

Evaluation:

- **The solution is obtained by writing the source code in a single .java file, *YourNameExam.java*. You can use as many static methods as you wish.**
- Each solution must provide a package name in this format ***ism.ase.ro.sap.exam.lastName.firstName***
- ***The solution must generate each result on its own. Using hardcoded values (obtained from other students) is not allowed (the solution will be evaluated with 0)***
- To get each requirement points you need to provide an error free (0 compiler errors) program that generates/displays correct results. Incorrect or partially correct solutions are not evaluated
- All solutions will be cross-checked with MOSS. Solutions that share more than 30% (except the given code) will be evaluated with 0.

You are a forensic cybersecurity specialist employed to recover data from a ransomware attack. This is what you know:

- The ransomware attack has been conducted by a parasitic virus that encrypted most available files.
- The encryption was based on AES, with a 128 bit key, randomly secure generated (you can't brute force it)
- The attack was stopped before the virus had a chance to send the encryption key to the C&C (Command & Control) center.
- The key is stored locally in a random file. From previous investigations you know that is in one of the files from System32 (see the system32.zip given archive).
- Fortunately, you have the SHA2 fingerprint for all those files, computed 1 month ago (before the attack). They are given in the *fingerprints.txt* file. The values are stored in Base64 encoding.

(10 p) Use the *fingerprints.txt* content to identify the file from system32.zip which has been changed.

(10 p) Using the random password, extracted from the file identified at the previous step, decrypt the "*financialdata.enc*" file into "*financialdata.txt*". The virus has encrypted it using AES in CBC mode, with PKCS5Padding. Reverse engineering the virus you find out that that the IV had 1st byte (**from right to left**) equal with 23, 2nd byte equal with 20, 3rd byte equal with 2 and 4th byte equal with 3. The rest of them are all 0s.

(5 p) To confirm your success and get your bounty, write the value of the 1st IBAN into *myresponse.txt* and digital sign this file with your private key (you need to generate a private – public key using keytool). The signature is an RSA with SHA256 digital signature. Don't forget to send the *"financialdata.txt", "myresponse.txt"* and your signature stored in a file called ***DataSignature*.ds**

Upload

- **the .java file with your solution**
- ***financialdata.txt***
- ***myresponse.txt***
- ***DataSignature.ds***
- ***Your X509 certificate file***