package ro.ase.petricamariuscosmin.sap;

import java.io.BufferedInputStream;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import java.security.InvalidKeyException;

import java.security.NoSuchAlgorithmException;

import java.text.SimpleDateFormat;

import java.util.Base64;

import java.util.Calendar;

import java.util.Collection;

import java.util.HashMap;

import java.util.Map;

import java.util.Stack;

import javax.crypto.Mac;

import javax.crypto.spec.SecretKeySpec;

public class assignment {

public static byte[] getHMAC(File file, String key, String algorithm) throws IOException, NoSuchAlgorithmException, InvalidKeyException {

FileInputStream fis = new FileInputStream(file);

BufferedInputStream bis = new BufferedInputStream(fis);

byte[] buffer = new byte[10];

int noBytes = 0;

Mac mac = Mac.getInstance(algorithm);

mac.init(new SecretKeySpec(key.getBytes(), algorithm));

while(noBytes != -1) {

noBytes = bis.read(buffer);

if(noBytes != -1)

mac.update(buffer,0,noBytes);

}

fis.close();

return mac.doFinal();

}

public static void scan(String password, String mainDir, File HMACfile) throws InvalidKeyException, NoSuchAlgorithmException, IOException {

if(!HMACfile.exists()) {

HMACfile.createNewFile();

}

HashMap<String, String> map

= new HashMap<String, String>();

File file = new File(mainDir);

Stack<File> s = new Stack<>();

s.push(file);

// initially stack is not empty

System.out.println("Content of Directory " + mainDir

+ " is");

while (!s.empty()) {

File tmpF = s.pop();

// check if it is a file or not

if (tmpF.isFile()) {

System.out.println("File name is: " + tmpF.getName() + "and HMAC is: " + Base64.getEncoder().encodeToString(getHMAC(tmpF, password, "HmacSHA1")));

map.put(tmpF.getAbsolutePath(), Base64.getEncoder().encodeToString(getHMAC(tmpF, password, "HmacSHA1")));

BufferedWriter bf = null;

bf = new BufferedWriter(new FileWriter(HMACfile));

for (Map.Entry<String, String> entry :

map.entrySet()) {

// put key and value separated by -->

bf.write(entry.getKey() + "-->" + entry.getValue());

bf.newLine();

}

bf.flush();

bf.close();

}

else if (tmpF.isDirectory()) {

// It's an directory, list and push all

// files in stack

File[] f = tmpF.listFiles();

for (File fpp : f) {

s.push(fpp);

}

} // else if ends here

} // stack is not empty loop ends here

}

public static void check(String password, String mainDir, File HMACfile) throws InvalidKeyException, NoSuchAlgorithmException, IOException {

String timeStamp = new SimpleDateFormat("yyyyMMdd\_HHmmss").format(Calendar.getInstance().getTime());

File ReportFile = new File("Reportfile"+timeStamp+".txt");

PrintWriter pWriter = new PrintWriter(ReportFile);

if(!ReportFile.exists()) {

ReportFile.createNewFile();

}

if(!HMACfile.exists()) {

return;

}

File file = new File(mainDir);

Stack<File> s = new Stack<>();

s.push(file);

System.out.println("Content of Directory " + mainDir

+ " is");

String line = null;

HashMap<String, String> map

= new HashMap<String, String>();

while (!s.empty()) {

File tmpF = s.pop();

// check if it is a file or not

if (tmpF.isFile()) {

String HMACOfFile=Base64.getEncoder().encodeToString(getHMAC(tmpF, password, "HmacSHA1"));

FileReader reader = new FileReader(HMACfile);

BufferedReader bReader = new BufferedReader(reader);

while ((line = bReader.readLine()) != null) {

String[] keyValuePair = line.split("-->", 2);

if (keyValuePair.length > 1) {

String key = keyValuePair[0];

String value = keyValuePair[1];

map.put(key, value);

}

}

if(!map.containsKey(tmpF.getAbsolutePath())) {

System.out.println("NEW: " + tmpF.getAbsolutePath());

pWriter.println("NEW: " + tmpF.getAbsolutePath());

}

for (Map.Entry<String, String> entry :

map.entrySet()) {

if(tmpF.getAbsolutePath().equals(entry.getKey())){

if(HMACOfFile.equals(entry.getValue())){

System.out.println("OK: " + tmpF.getAbsolutePath());

pWriter.println("OK: " + tmpF.getAbsolutePath());

}

else {

System.out.println("CORRUPTED: " + tmpF.getAbsolutePath());

pWriter.println("CORRUPTED: " + tmpF.getAbsolutePath());

}

}

}

bReader.close();

}

else if (tmpF.isDirectory()) {

// It's an directory hence list and push all

// files in stack

File[] f = tmpF.listFiles();

for (File fpp : f) {

s.push(fpp);

}

}

}

pWriter.close();

}

public static void main(String[] args) throws InvalidKeyException, NoSuchAlgorithmException, IOException {

File HMACfile = new File(args[3]);

if(args[0].equals("scan")) {

scan(args[1], args[2], HMACfile);

}

if(args[0].equals("check")) {

check(args[1], args[2], HMACfile);

}

} // main function ends here

}