Université Grenoble Alpes **Grenoble INP**
UFR IM2AG **ENSIMAG**

Master 2 CyberSecurity
Software Security and Secure Programming

Exercise on Access Control

Exercise 1

We consider a Java Class C1 with a public method m1() allowing to perform some computations on a **secret** resource *key* and returning some integer value. Clearly, this method should **not** be called by any **untrusted** caller. To ensure that, the caller should provide as a parameter to m1() some credential as a string s.
A check is performed within m1() to verify that the caller is legitimate. When it is the case, permission P, allowing to read key is granted. Later on this permission is disabled (when no longer required). The corresponding code (in pseudo Java) is given below.

```
import java.util.* ;

class C1 {

int key[N] ;  // secret resource of size N

public int m1 (String s, int length) {
 // s is used to authenticate the caller
 int i, sum, result ;
 b = checkAcess(s) ;
 if (b) enablePermission(P) ; // give read acces to buffer key
 try {
   if (b) {
         i=0 ;
         sum= 0 ;
         while (i<length) {
               sum = key[i] + sum ;
               i = i+1 ;
        } ;
          disablePermission(P) ; // disable acccess to buffer key
          if (sum>0)
                result = Hash(sum); // returns a positive hash value
          else
                result = -1 ;
          return result ;
   }
  } catch (IndexOutofBoudsException e) {
        // in case key is accessed out of bounds
       System.out.println("Error !") ;
 }
}
}
```

Q1. Why is it necessary/useful to explicitly enable permissions to read key inside m1()(since the caller credential is already explicitly checked beforehand) ? Indicate in which conditions enabling this permission is required or not required ...

Q2. The way permission P is enabled/disabled inside m1() is clearly **insecure**. Indicate why, and how to correct it.

Q3. If this code was written in C or C++, it would **not** be possible to enable/disable permission P like in Figure 2. Explain (in a few lines) which other solutions could be used in terms of access control (indicating their advantages and drawbacks).

Q4. If a trusted caller executes method m1(), which information could it get `about secret buffer key` ?Assuming that function call Hash(sum) returns no confidential information about sum, does m1() leak some confidential information about key ? If yes, which information, if not, why not ?