Sécurité des JAR et du Bytecode

Programmation avancée – Java

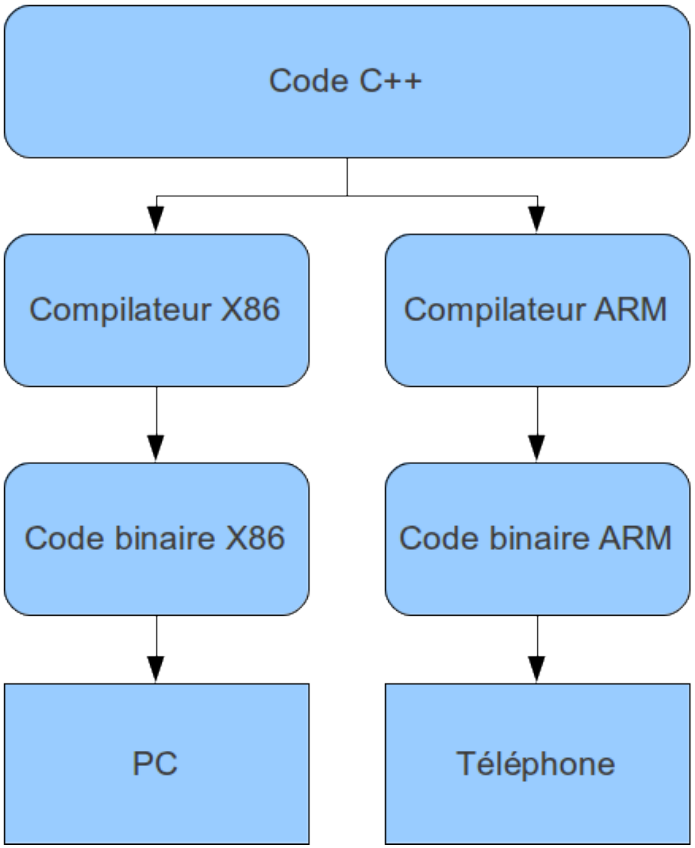
F5 – ISIMA 2020/2021

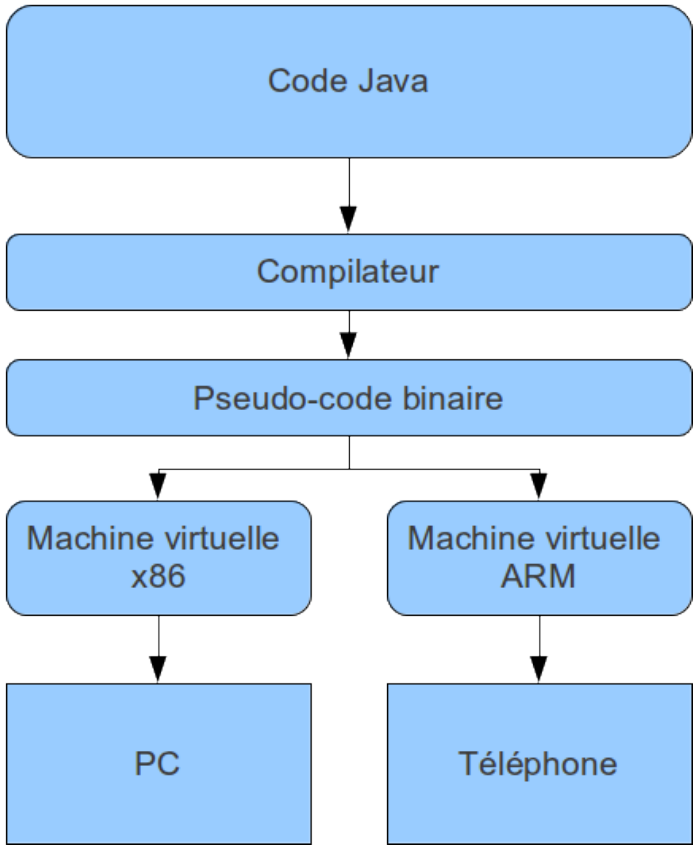
Olivier Goutet
o.goutet@openium.fr

24 novembre 2020

Plan

Compilation





Byte code

- Le code compilé Java est donc du Bytecode
- Java est un langage interprété
- Ce n'est pas la même chose que du code machine issue d'une compilation

Exemple de Bytecode

```
1   for (int i = 2; i < 1000; i++) {
2     for (int j = 2; j < i; j++) {
3         if (i % j == 0)
4             continue outer;
5     }
6     System.out.println (i);
7   }
```

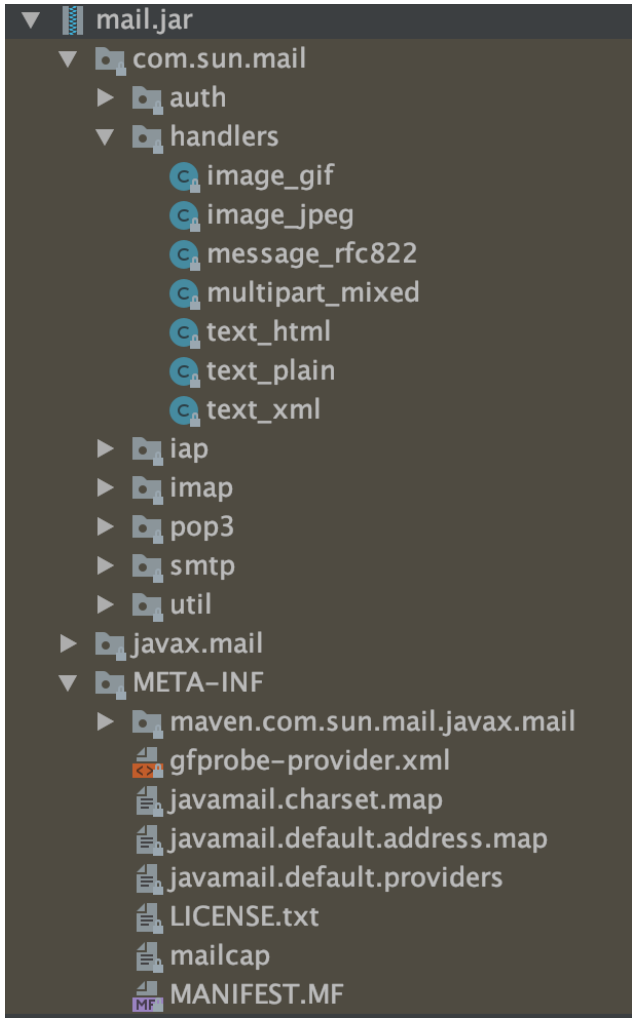
Exemple de Bytecode

```
 1 iconst_2
 2 istore_1
 3 iload_1
 4 sipush  1000
 5 if_icmpge       44
 6 iconst_2
 7 istore_2
 8 iload_2
 9 iload_1
10 if_icmpge       31
11 iload_1
12 iload_2
13 irem
14 ifne    25
15 goto    38
16 iinc    2, 1
17 goto    11
18 getstatic       #84; // Field java/lang/System.out:Ljava/io/PrintStream;
19 iload_1
20 invokevirtual   #85; // Method java/io/PrintStream.println:(I)V
21 iinc    1, 1
22 goto    2
23 return
```

Java ARchive

- Le JAR est le livrable habituel de JAVA
  - Pour une bibliothèque
  - Pour un programme
- `fichier.jar`
- C'est simplement un zip
  - Unzip

Java ARchive



MANIFEST.MF

- Métadonnée avec tout ce qui est nécessaire à l'archive JAVA
- Chemin META-INF/MANIFEST.MF
- Données
  - Clefs valeurs
  - Si application : point d'entrée du programme (Main)
  - `Main-Class: com.example.MyClassName`
  - versions
  - Classpath : `Class-Path: . pkg1.jar path/to/pkg2.jar`

Plan

Lets Unzip

- On retrouve plein de chose intéressantes.
- Demo

Décompilation

- Attention, vous n'avez pas le droit de décompiler du code sans l'autorisation de l'auteur. Vérifiez donc bien la licence avant de vous aventurer à cette opération !

Décompilation

- Peut-on partir d'un .class et arriver à un code compréhensible et analysable ?
- En Java très simple
- Plusieurs outils existent
  - http ://java-decompiler.github.io/
  - InteliJ
  - ...

Décompilation

- Demo

Comment protéger son programme

- Données ?
- Code ?

Comment protéger ses données

- Télécharger les données à l'exécution
- Chiffrer les données une fois reçue
- Constantes
  - Les chiffrer et les déchiffrer à l'utilisation
  - Méthode de chiffrement symétrique à cacher dans le code, sinon ça ne sert pas à grand chose...

Comment protéger son code

- Ne pas mettre de code sensible dans votre programme
  - Exécuter sur un serveur par exemple
- Java Native Interface (JNI)
  - Appel de bibliothèques .o depuis le programme JAVA
  - Appel de .so sous Android depuis le programme JAVA
- Obfuscation
  - Rendre le code plus difficile à lire
  - Remplace chaque attribut et chaque méthode
    - par un nom aléatoire...
    - Overload Induction par a, puis b, c ... aa, ab...
    - Invisibility : des caractères non autorisés
  - Supprime les commentaires JavaDoc

Obfuscation

```
1  // Orignal
2  public synchronized void put(int key, Employee value) {
3      Integer I = new Integer(key);
4      super.put(I, (Object) value);
5  }
6  //Aleatoire
7  public synchronized void yrwla35rn3z22jd0sci9(int sbhc8wduotn7gkbr8pq6, k0j9y980ekqci18t09
8      Integer f841593p5rh12zf88285 = new Integer(sbhc8wduotn7gkbr8pq6);
9      super.yrwla35rn3z22jd0sci9(f841593p5rh12zf88285, (Object) 78nrx59777f4io0qpg7t);
10 }
11 }
```

Obfuscation

```
1  // Orignal
2  public synchronized void put(int key, Employee value) {
3      Integer I = new Integer(key);
4      super.put(I, (Object) value);
5  }
6  // Overload Induction
7  public synchronized void a(int a, b c) {
8      Integer d = new Integer(a);
9      super.a(d, (Object) c);
10 }
11 }
```

Obfuscation

```
1  // Orignal
2  public synchronized void put(int key, Employee value) {
3      Integer I = new Integer(key);
4      super.put(I, (Object) value);
5  }
6  // Invisibility
7  public synchronized void #~a(int @b, f# a~) {
8      Integer #~b = new Integer(@b);
9      super.#~a(#~b, (Object) a~);
10 }
11 }
```

Comment protéger son code

- Complexifier son code
  - Mettre des classes inutiles dans l'arboressence
  - Créer du code inutilement complexe
  - on complexifie l'attaque... voilà tout
- Ne pas utiliser Java

Références

- https ://cyberzoide.developpez.com/securite/obfuscation/