Kopi

Reversing - 50 points

# Soal

|  |
| --- |
| Lihat, seruput dan rasakan.  [kopi.zip](https://ctf.idsecconf.org/download?file_key=be7963b799741f2da3f54d1f808d429d598f4e66764a8bd3d20473317a54c9fd&team_key=2a3d00950bcec988c6f9a00242687028ade2e180c4a337d64a8761b253b2df59) - 8541262e67b3d55f5e71de07b5d259be |

# Solusi

Diberikan file Kopi.class, decompile file class Java tersebut dengan tools online, dapat source berikut:

|  |
| --- |
| **import** **java.io.PrintStream**; **import** **java.util.Stack**;  **class** **Kopi** {   Kopi()  {  flag = **new** StringBuilder();  }   **private** String getFlag()  {  **return** flag.toString();  }   **private** boolean checkPassword(String s)  **throws** Exception  {  String as[] = s.split("-");  Stack stack = **new** Stack();  Stack stack1 = **new** Stack();  String as1[] = as;  int j = as1.length;  **for**(int k = 0; k < j; k++)  {  String s1 = as1[k];  stack.push(Integer.valueOf(Integer.parseInt(s1)));  }   int i = ((Integer)stack.pop()).intValue();  stack1.push(Integer.valueOf(i));  j = ((Integer)stack.pop()).intValue();  stack1.push(Integer.valueOf(j));  **while**(!stack.empty())   {  int l = ((Integer)stack.pop()).intValue();  **if**(l != i - j)  **return** **false**;  stack1.push(Integer.valueOf(l));  i = j;  j = l;  }  **if**(i \* (j / i) != 1)  **return** **false**;  **for**(int i1 = 0; i1 < buff.length; i1++)  {  buff[i1] -= ((Integer)stack1.pop()).intValue();  flag.append((char)buff[i1]);  }   **return** **true**;  }   **public** **static** void main(String args[])  {  **if**(args.length != 1)  {  System.out.println("Usage: Kopi <password>");  **return**;  }  System.out.println((**new** StringBuilder()).append("Checking ... ").append(args[0]).toString());  Kopi kopi = **new** Kopi();  boolean flag1 = **false**;  **try**  {  flag1 = kopi.checkPassword(args[0]);  }  **catch**(Exception exception)  {  System.out.println("Invalid password");  **return**;  }  **if**(flag1)  System.out.println((**new** StringBuilder()).append("The flag is: ").append(kopi.getFlag()).toString());  **else**  System.out.println("Invalid password");  }   **private** StringBuilder flag;  **private** **static** int buff[] = {  103, 109, 99, 106, 128, 81, 89, 126, 141, 156,   163, 241, 351, 474, 715, 1097, 1664, 2668, 4251, 6890  };  } |

Jadi program meminta input yang nanti akan digunakan untuk mengurangi variabel “buff” yang nanti akan jadi flagnya, dengan menggunakan struktur data stack. Masalahnya, input apa yang valid?

### Leap of Faith

Tanpa memikirkan lebih jauh tentang algoritma pengecekannya, saya langsung olah variabel “buff” tersebut, karena sepertinya mirip kode ASCII, baru kemudian memikirkan dikurangi berapa supaya jadi flag yang kira-kira valid (103, 109, 99, 106, 128, 81, 89, 126, 141, 156, 163, 241, 351, 474, 715, 1097, 1664, 2668, 4251, 6890).

Asumsi pertama adalah 4 huruf awalnya “flag” (ascii: 102, 108, 97, 103). Bandingkan dengan 4 angka pertama buff: (103, 109, 99, 106). Kalau kita kurangkan hasilnya adalah: (1, 1, 2, 3). Kelihatannya seperti deret Fibonacci, sepertinya angka-angka dari program harus dikurangi dengan deret Fibonacci sepanjang itu.

Coba secara singkat di Python

|  |
| --- |
| In [1]: buff = [103, 109, 99, 106, 128, 81, 89, 126, 141, 156, 163,  241, 351, 474, 715, 1097, 1664, 2668, 4251, 6890]  In [2]: fibo = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233,  377, 610, 987, 1597, 2584, 4181, 6765, 10946,   17711, 28657, 46368, 75025, 121393, 196418, 317811]  In [3]: key = [buff[i] - fibo[i] for i in range(len(buff))]  In [4]: ''.join([chr(x) for x in key])  Out[4]: 'flag{ILikeJavainCTF}' |

Langsung dapat flagnya:

**flag{ILikeJavainCTF}**