

Forum:[Hack Forums](#)

Thread: [Hex values doubled in IDA/Ghidra decompiled code](#)

samuwhale - September 2, 2022, 8:53 pm

I have the following piece of dart code, which check if the input is "ABCDE" and prints correct if the input matches, and it works fine.

```
Code import 'dart:io';
final List<int> flag = [65, 66, 67, 68, 69];
void main() {
  print("\ngive flag: ");
  List<int> input = stdin.readLineSync(encoding: latin1)!.codeUnits;
  if (input.length != flag.length) {
    print("\nnot flag");
    return;
  }
  for (int i = 0; i < flag.length; i++) {
    if (input[i] != flag[i]) {
      print("\nnot flag");
      return;
    }
  }
  StringBuffer newFlag = new StringBuffer();
  input.forEach((e) {
    newFlag.writeCharCode(e);
  });
  print("correct");
}
```

I compiled the code with dart compile aot-snapshot exmaple.dart and then tried to decompile it with IDA and Ghidra. However, the original decimal values seem to be doubled in the decompiled code.

Ghidra decompiled code:

Code void flag(void)

```
{
  long IVar1;
  long extraout_RAX;
  long unaff_R14;

  if (&stack0xffffffffffff8 < *(undefined **)(ulong *)(unaff_R14 + 0x40) ||
      &stack0xffffffffffff8 == *(undefined **)(ulong *)(unaff_R14 + 0x40)) {
    **(code **)(unaff_R14 + 0x238)();
  }
  IVar1 = Precompiled_Stub__iso_stub_AllocateArrayStub();
  *(undefined8 *)(IVar1 + 0x17) = 0x82;
  *(undefined8 *)(IVar1 + 0x1f) = 0x84;
  *(undefined8 *)(IVar1 + 0x27) = 0x86;
  *(undefined8 *)(IVar1 + 0x2f) = 0x88;
```

```
*(undefined8 *)(&IVar1 + 0x37) = 0x8a;  
new __GrowableView._withData();  
*(undefined8 *)(&extraout_RAX + 0xf) = 10;  
return;  
}
```

As you can see: 0x82, 0x84, 0x86, 0x88, 0x8a in decimal are 130, 132, 134, 136, 138.

The dec values are double of what was originally in the code (65, 66, 67, 68, 69). I am not sure why this is happening... I suspect there some carry over or something.

My aim is to be able to tell that "ABCDE" is the required input just by the .aot file. Thanks :)

AbakBarama - September 3, 2022, 12:54 am

I didn't work with dart, but can suppose...

You've created numeric array of type "int". Each number occupies 8 bytes (see decompiled code). We don't know about internal representation of "int". Least significant bit can have non-standard meaning. For example, it can store sign of number, or some flag/tag for service purposes.

You can try to use negative numbers, and you'll see sign location.