

Modification LK.bin to erease Orange state and confirmation boot

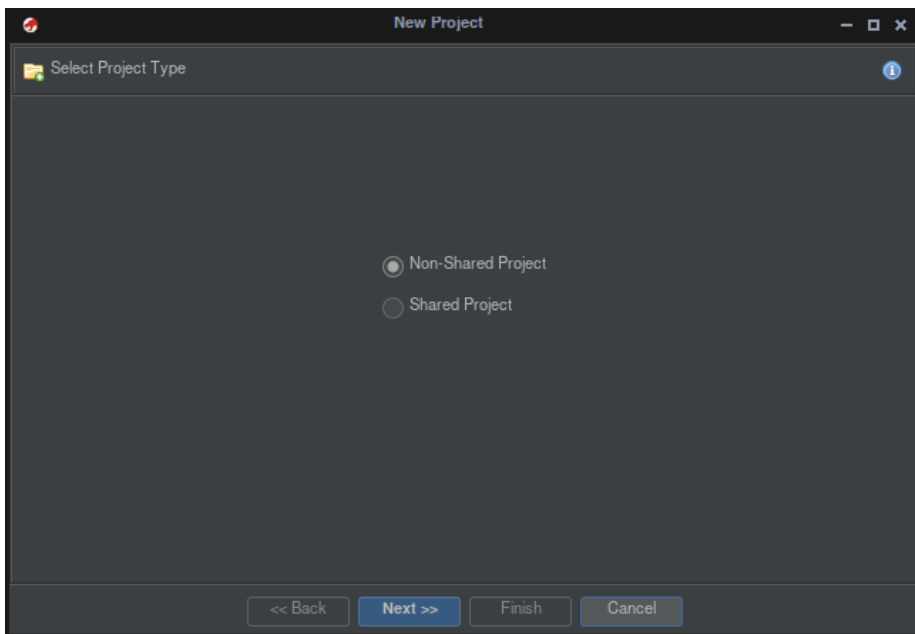
Bypass Confirmation Boot

Tool necessary : Ghidra

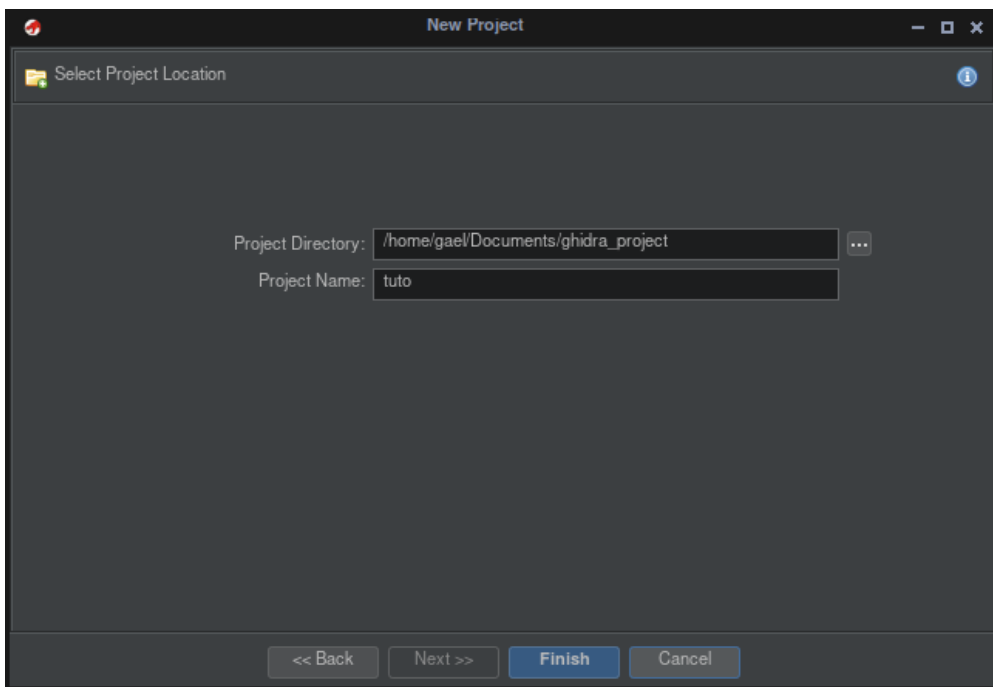
Download and install here : <https://github.com/NationalSecurityAgency/ghidra>

Open file lk.bin in Ghidra

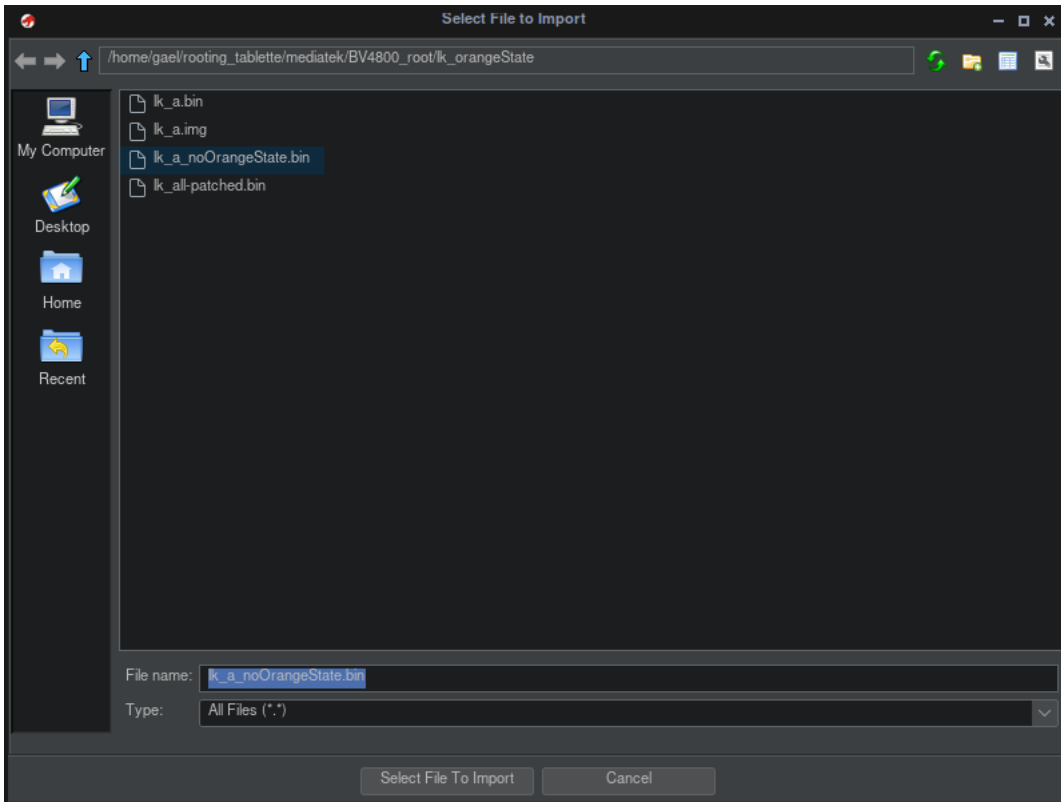
- File --> new project
- Non-Shared Project --> Next



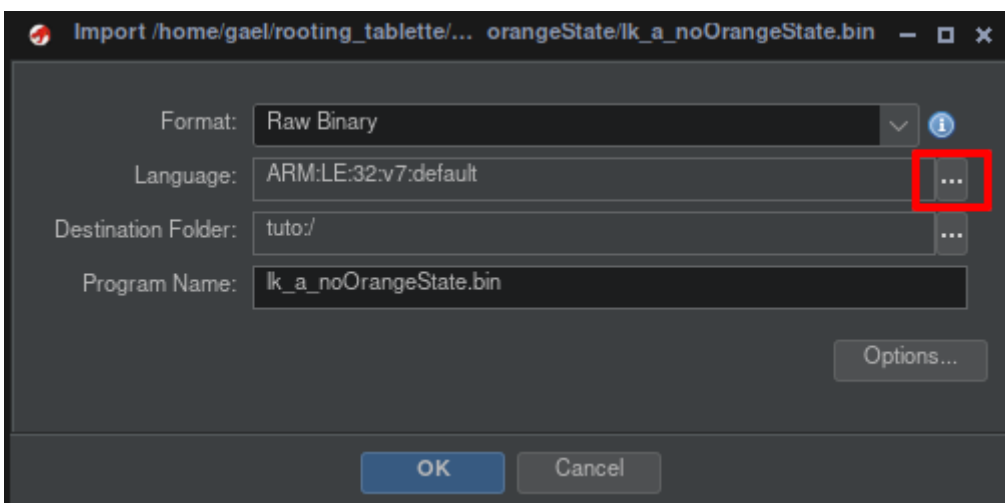
- Give a name to your project --> Next

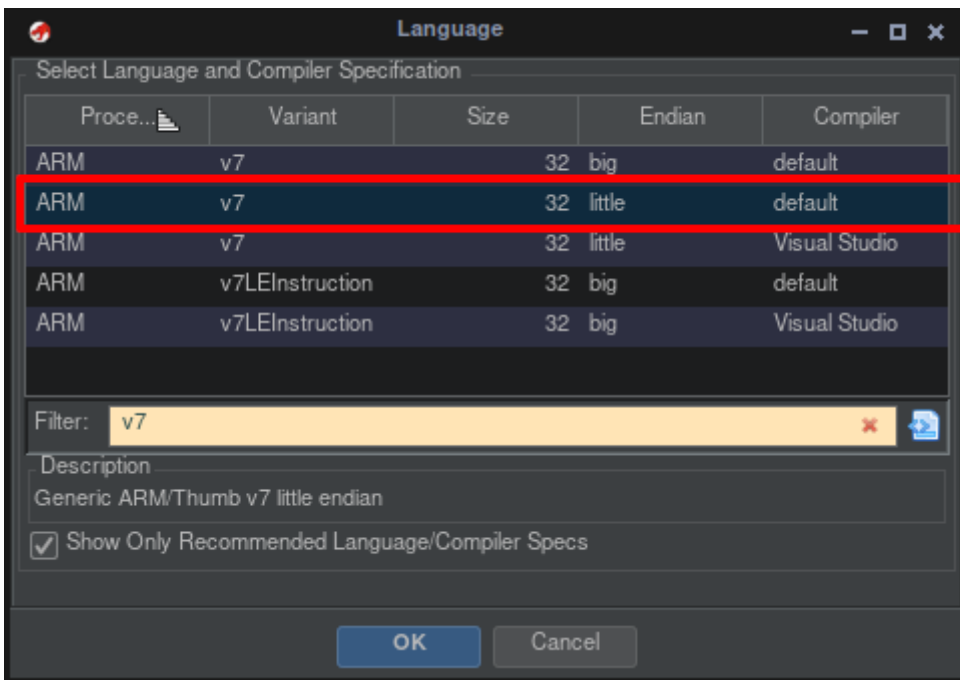


- File --> import File
- Import File lk.bin you want patch

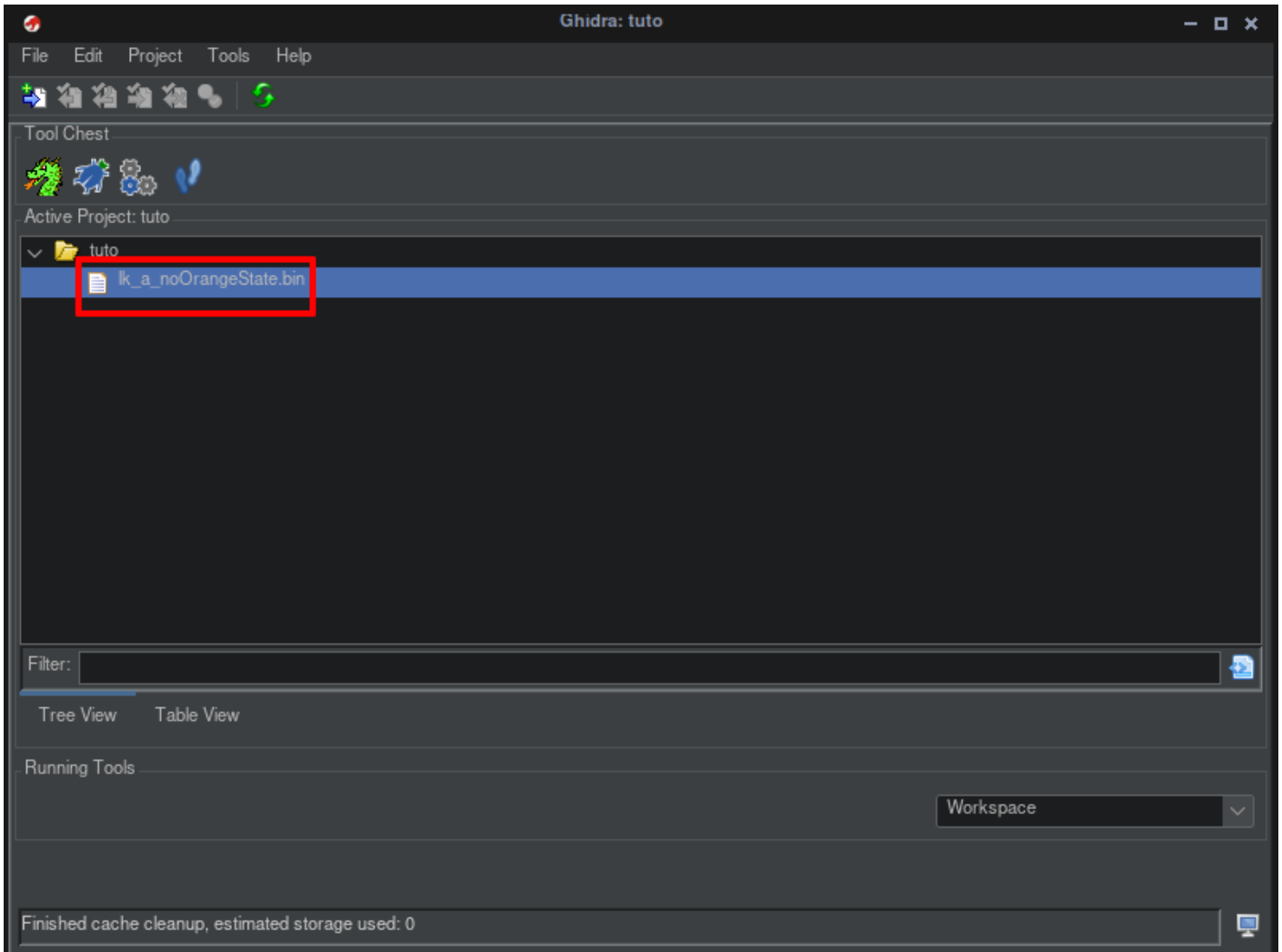


- Select Language --> ARM v7 | 32 | little | default

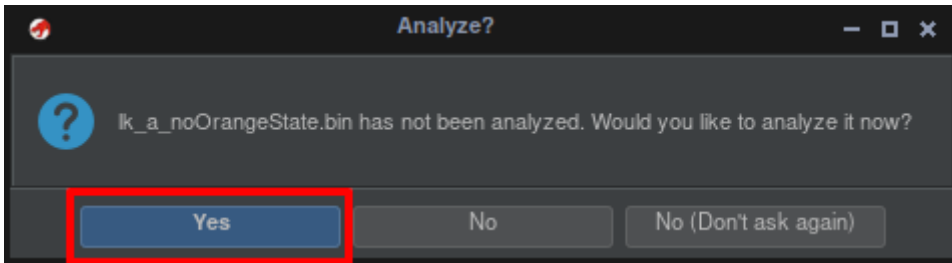




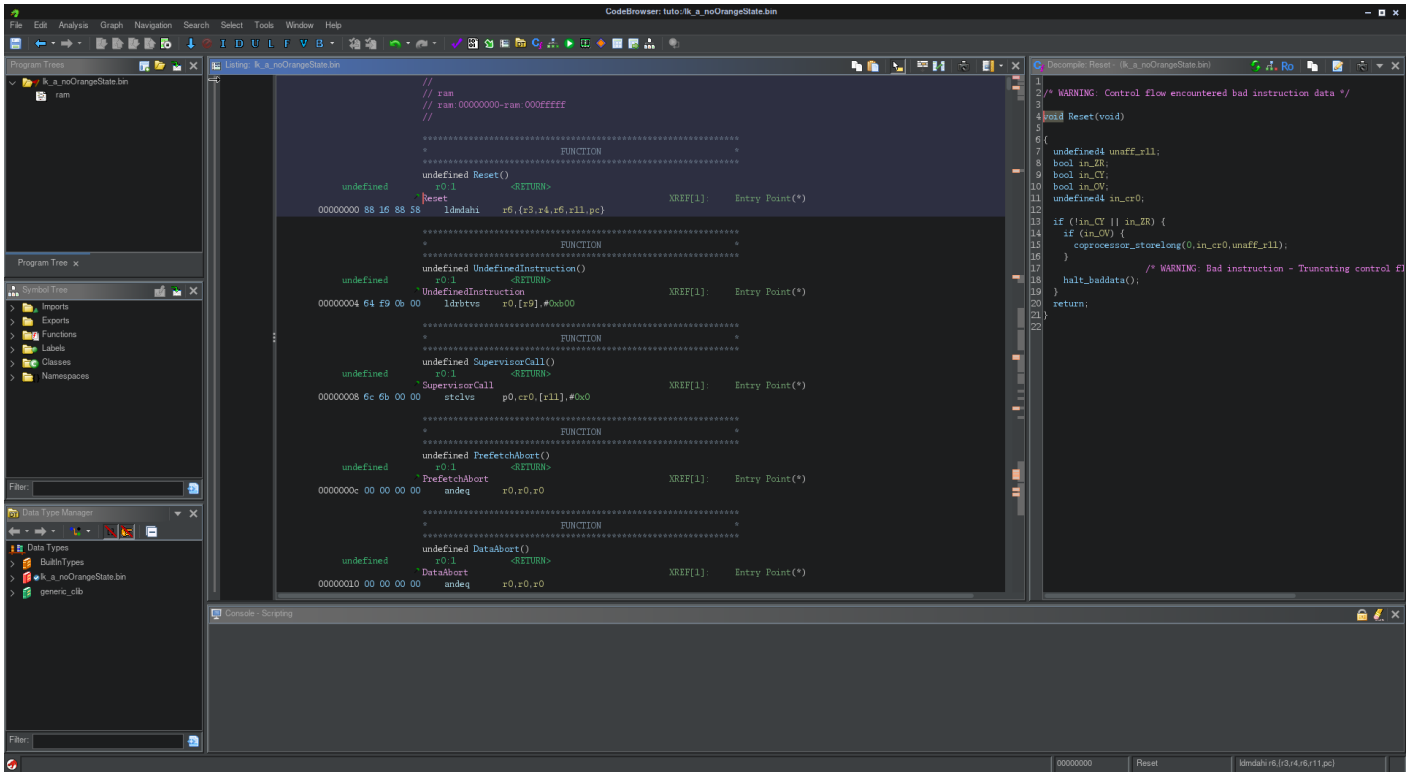
- Double click on file



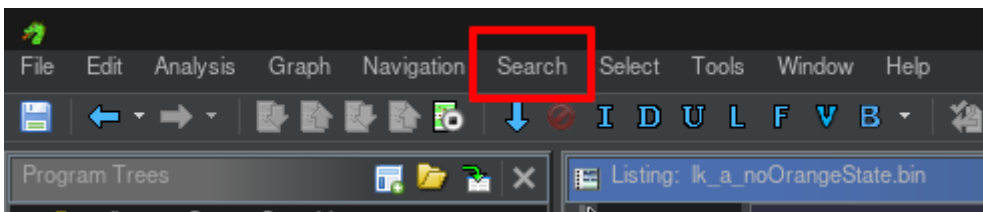
- respond yes to analyzed



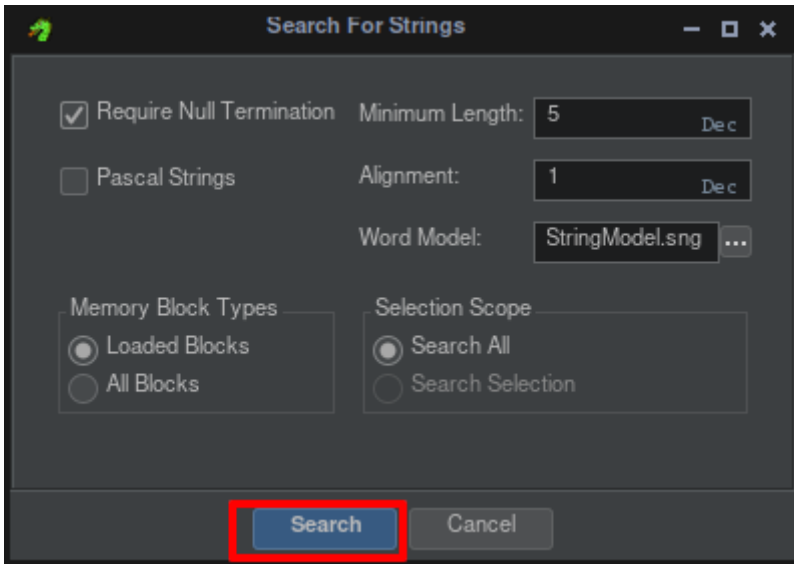
- We will get



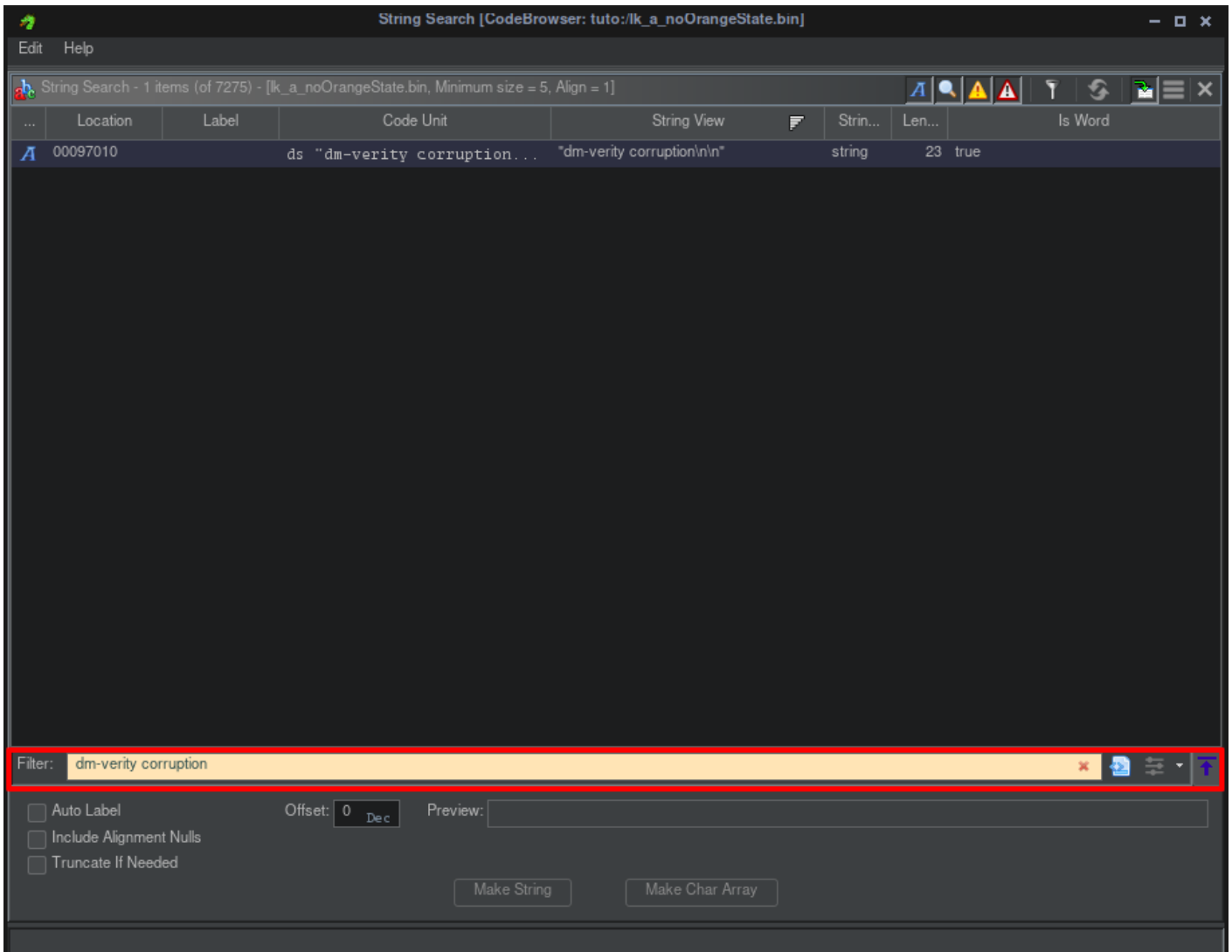
- Search --> for String



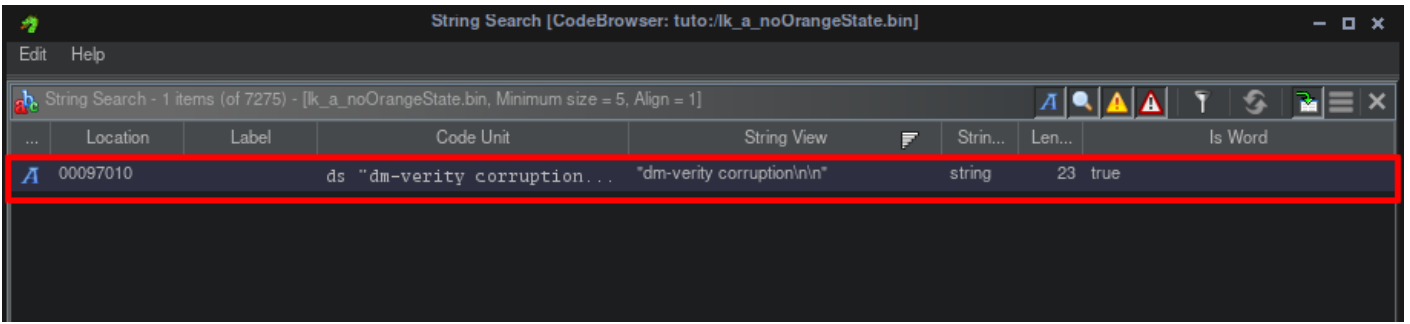
- continue with clicking on Search



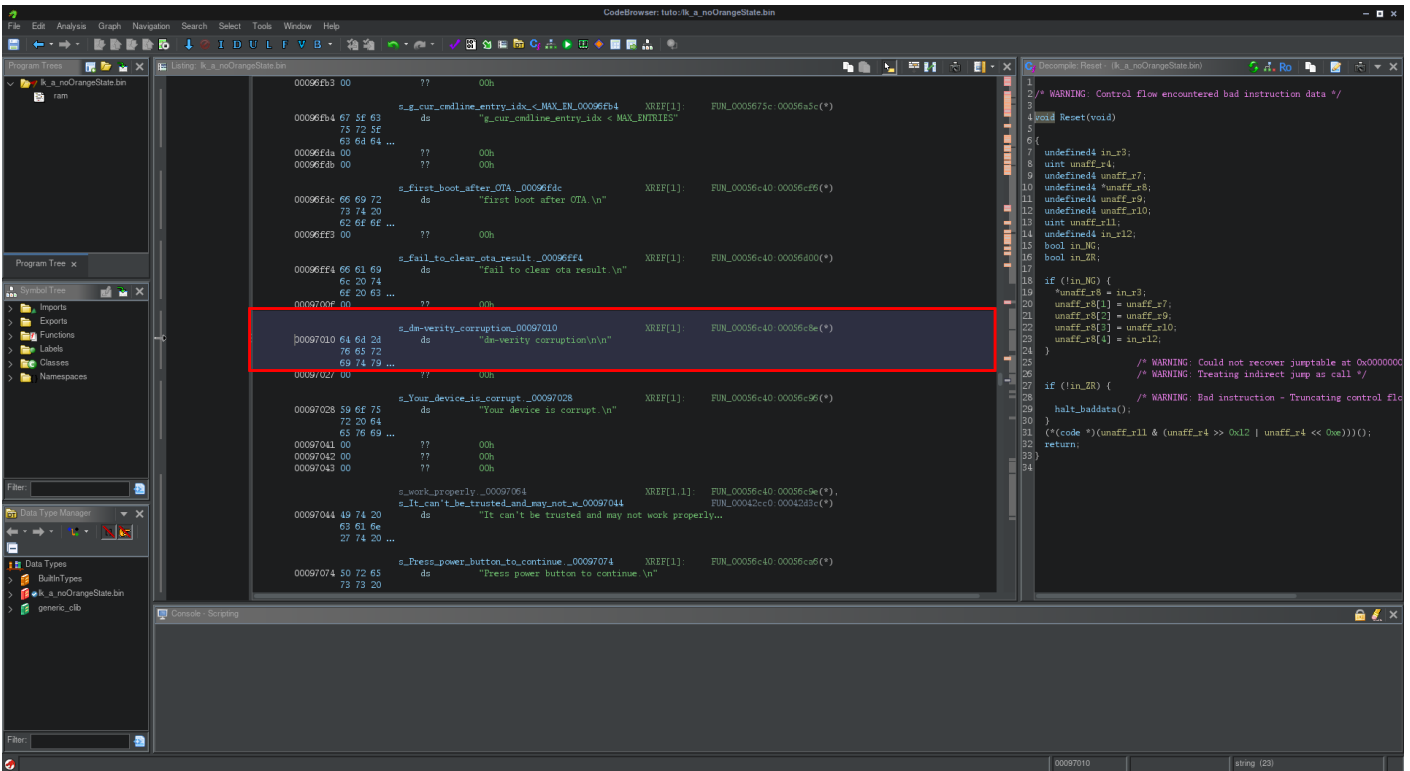
- type "dm-verity corruption" on filter



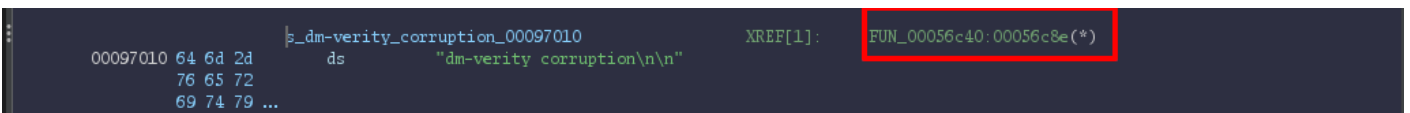
- One final will find --> click on line find



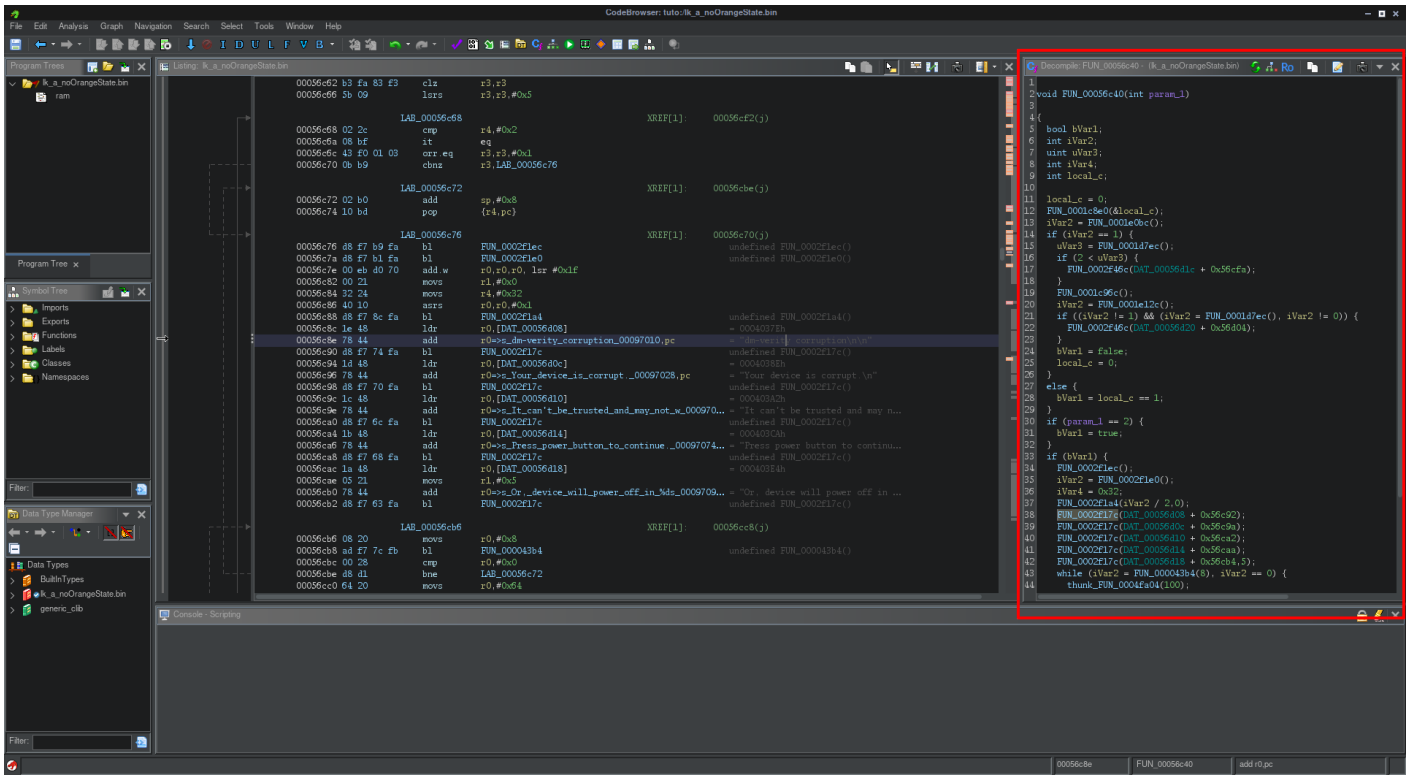
- this will give



- Go to the fonction who call the string "dm-verity corruption" with double click on hexa (FUN_00056c40:00056c8e)



- This will give



- We get the fonction we need on right

```
Decompile: FUN_00056c40 - (lk_a_noOrangeState.bin)
1
2 void FUN_00056c40(int param_1)
3
4 {
5     bool bVar1;
6     int iVar2;
7     uint uVar3;
8     int iVar4;
9     int local_c;
10
11     local_c = 0;
12     FUN_0001c8e0(&local_c);
13     iVar2 = FUN_0001e0bc();
14     if (iVar2 == 1) {
15         uVar3 = FUN_0001d7ec();
16         if (2 < uVar3) {
17             FUN_0002f46c(DAT_00056d1c + 0x56cfa);
18         }
19         FUN_0001c96c();
20         iVar2 = FUN_0001e12c();
21         if ((iVar2 != 1) && (iVar2 = FUN_0001d7ec(), iVar2 != 0)) {
22             FUN_0002f46c(DAT_00056d20 + 0x56d04);
23         }
24         bVar1 = false;
25         local_c = 0;
26     }
27     else {
28         bVar1 = local_c == 1;
29     }
30     if (param_1 == 2) {
31         bVar1 = true;
32     }
33     if (bVar1) {
34         FUN_0002f1ec();
35         iVar2 = FUN_0002fle0();
36         iVar4 = 0x32;
37         FUN_0002f1a4(iVar2 / 2, 0);
38         FUN_0002f17c(DAT_00056d08 + 0x56c92);
39         FUN_0002f17c(DAT_00056d0c + 0x56c9a);
40         FUN_0002f17c(DAT_00056d10 + 0x56ca2);
41         FUN_0002f17c(DAT_00056d14 + 0x56caa);
42         FUN_0002f17c(DAT_00056d18 + 0x56cb4, 5);
43         while (iVar2 = FUN_000043b4(8), iVar2 == 0) {
44             thunk_FUN_0004fa04(100);
45             iVar4 = iVar4 + -1;
46             if (iVar4 == 0) {
47                 FUN_00006670();
48                 return;
49             }
50         }
51     }
52     return;
53 }
54
```

- The function who call string dm-verity is highlighted

```
FUN_0002f17c(DAT_00056d08 + 0x56c92);
```

- We can understand if the boot not start correctly

the var `param_1 == 2` and give `bVar1 = true;`

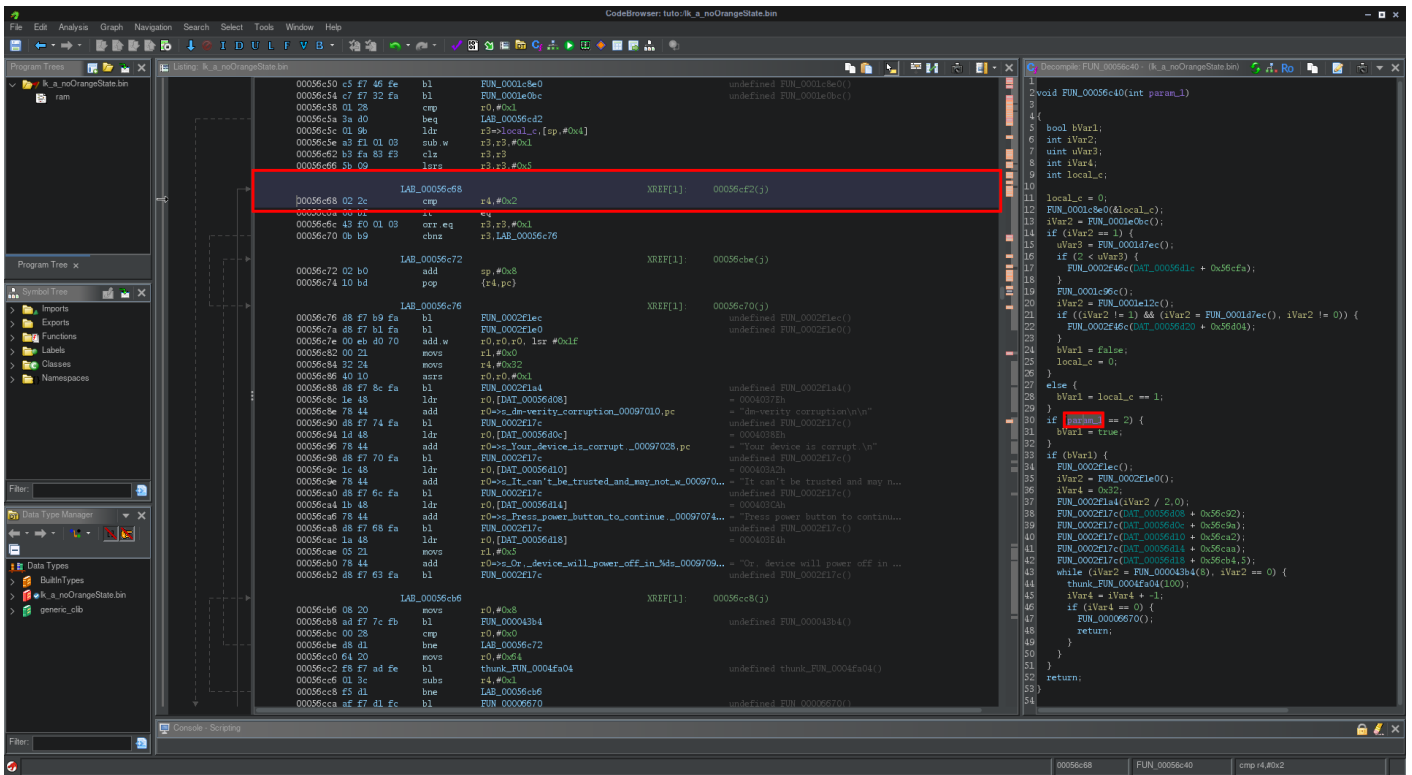
so the first an green is if boot is corectly and an red if we got an error

```
Decompile: FUN_00056c40 - (lk_a_noOrangeState.bin)
1
2 void FUN_00056c40(int param_1)
3
4
5 bool bVar1;
6 int iVar2;
7 uint uVar3;
8 int iVar4;
9 int local_c;
10
11 local_c = 0;
12 FUN_0001c8e0(&local_c);
13 iVar2 = FUN_0001e0bc();
14 if (iVar2 == 1) {
15     uVar3 = FUN_0001d7ec();
16     if (2 < uVar3) {
17         FUN_0002f46c(DAT_00056d1c + 0x56cfa);
18     }
19     FUN_0001c96c();
20     iVar2 = FUN_0001e12c();
21     if ((iVar2 != 1) && (iVar2 = FUN_0001d7ec(), iVar2 != 0)) {
22         FUN_0002f46c(DAT_00056d20 + 0x56d04);
23     }
24     bVar1 = false;
25     local_c = 0;
26 }
27 else {
28     bVar1 = local_c == 1;
29 }
30 if (param_1 == 2) {
31     bVar1 = true;
32 }
33 if (bVar1) {
34     FUN_0002f1ec();
35     iVar2 = FUN_0002f1e0();
36     iVar4 = 0x32;
37     FUN_0002f1a4(iVar2 / 2, 0);
38     FUN_0002f17c(DAT_00056d08 + 0x56c92);
39     FUN_0002f17c(DAT_00056d0c + 0x56c9a);
40     FUN_0002f17c(DAT_00056d10 + 0x56ca2);
41     FUN_0002f17c(DAT_00056d14 + 0x56caa);
42     FUN_0002f17c(DAT_00056d18 + 0x56cb4, 5);
43     while (iVar2 = FUN_000043b4(8), iVar2 == 0) {
44         thunk_FUN_0004fa04(100);
45         iVar4 = iVar4 + -1;
46         if (iVar4 == 0) {
47             FUN_00006670();
48             return;
49         }
50     }
51 }
52 return;
53 }
54
```

- We therefore need to modify one of these variables so that it is no longer used in the function.

```
param_1 == 2 or bVar1 = true;
```

- In the function, if we click on the desired variable, we move to the line where it is in the file.



- Right Click on line --> Patch instruction
- Change value of condition

```
LAB_00056c68 XREF[1]: 00056cf2(j)
00056c68 02 2c cmp r4, #0x2
```

- We can do like this

```
LAB_00056c68 XREF[1]: 00056cf2(j)
00056c68 05 2c cmp r4, #0x5
```

```
if (param_1 == 5) {
    bVar1 = true;
}
```

- After this we can save file quit

- For export with menu project
- Right click on file --> select export
- You can choose Format Original File

Find the warning string

<https://github.com/R0rt1z2/lkpatcher>

<https://lkpatcher.r0rt1z2.com/>

<https://blog.r0rt1z2.com/patch-mediatek-bootloader-images-lk.html>

Revision #5

Created 22 April 2024 08:36:27 by Foufure

Updated 23 May 2024 13:20:09 by Foufure