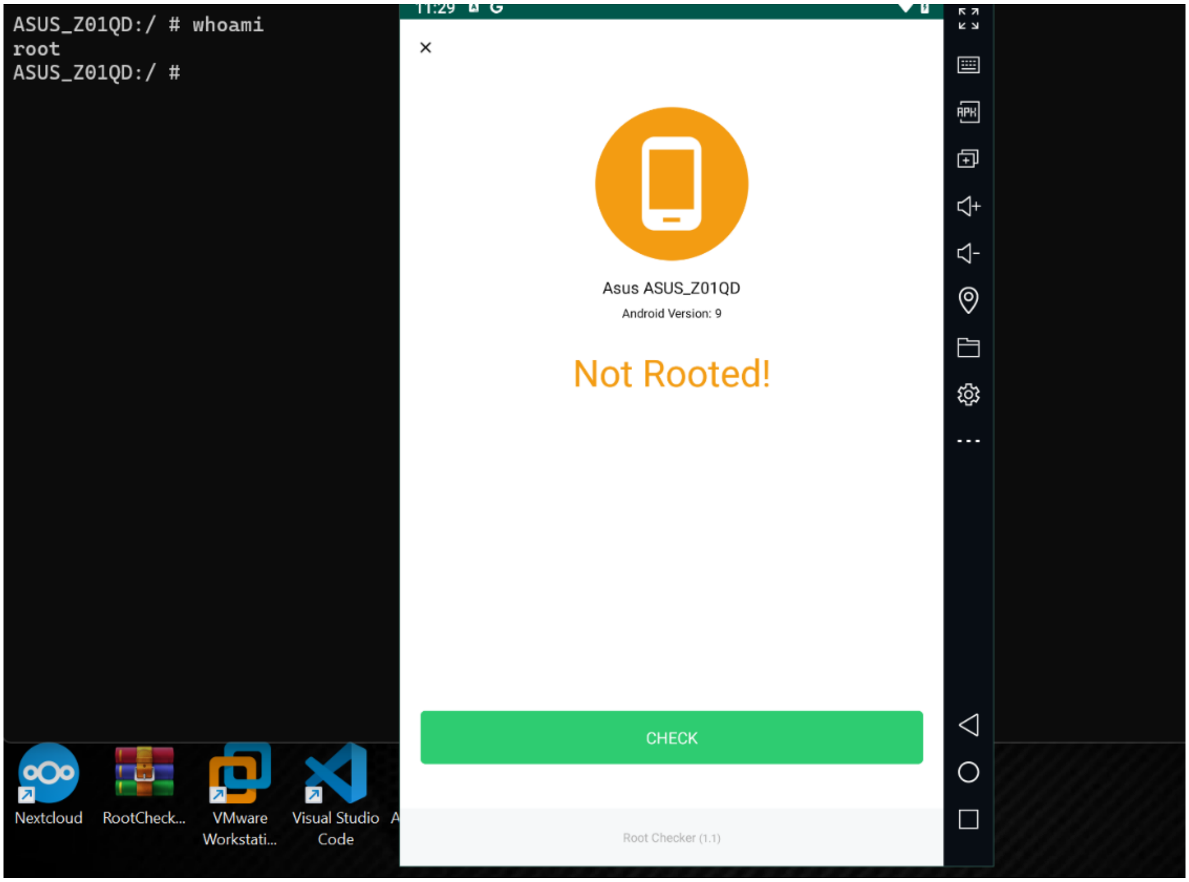
Android Technical Assessment

Before we dive in the Assessment I want to share with you the challenge that I faced , The app main functionality is Root Detecting as we all know, the problem that I faced was the app is not doing his job

I was Root and I didn't get detected as Shown here :



```
ASUS_Z01QD:/ # whoami
root
ASUS_Z01QD:/ #
```

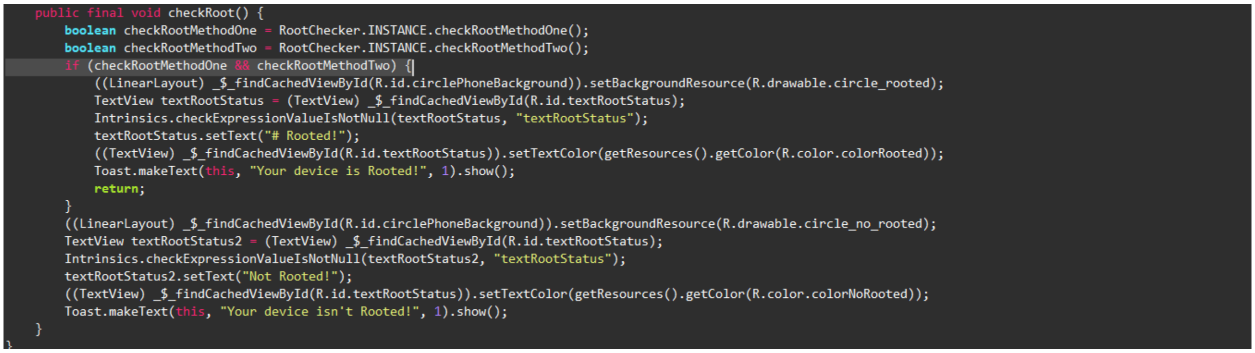
Asus ASUS_Z01QD

Android Version: 9

Not Rooted!

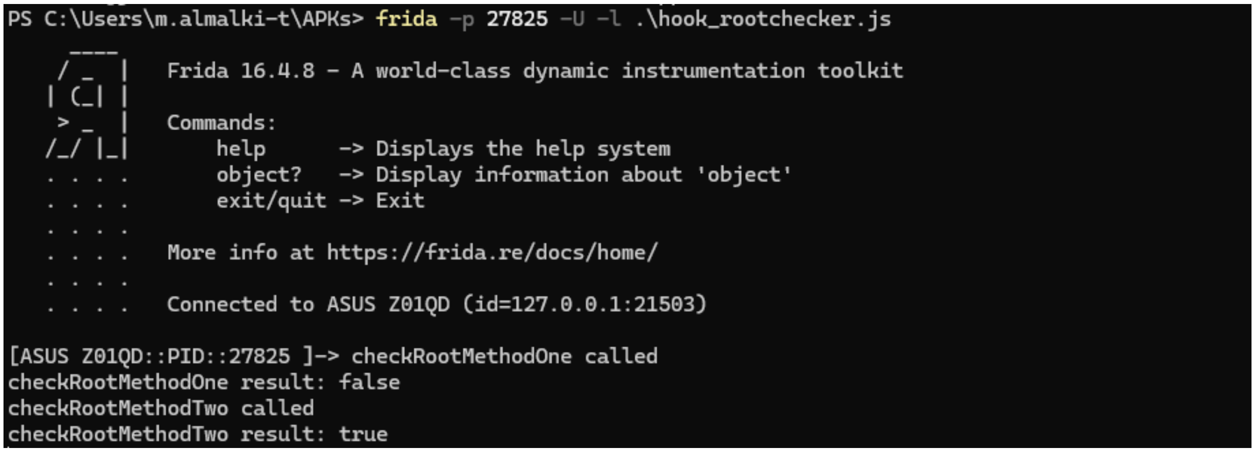
I started debugging to try to fix the issue, I looked at the logs using logcat but nothing seems to be related to the problem, I opened the app in jadx to analyses the code and I found the logical error.

The **if statement** in **checkroot()** needs **checkRootMethodOne** and **checkRootMethodTwo** to be both true to return true (Logical AND).



```
    public final void checkRoot() {
        boolean checkRootMethodOne = RootChecker.INSTANCE.checkRootMethodOne();
        boolean checkRootMethodTwo = RootChecker.INSTANCE.checkRootMethodTwo();
        if (checkRootMethodOne && checkRootMethodTwo) {
            ((LinearLayout) _$_findCachedViewById(R.id.circlePhoneBackground)).setBackgroundResource(R.drawable.circle_rooted);
            TextView textRootStatus = (TextView) _$_findCachedViewById(R.id.textRootStatus);
            Intrinsics.checkExpressionValueIsNotNull(textRootStatus, "textRootStatus");
            textRootStatus.setText("# Rooted!");
            ((TextView) _$_findCachedViewById(R.id.textRootStatus)).setTextColor(getResources().getColor(R.color.colorRooted));
            Toast.makeText(this, "Your device is Rooted!", 1).show();
            return;
        }
        ((LinearLayout) _$_findCachedViewById(R.id.circlePhoneBackground)).setBackgroundResource(R.drawable.circle_no_rooted);
        TextView textRootStatus2 = (TextView) _$_findCachedViewById(R.id.textRootStatus);
        Intrinsics.checkExpressionValueIsNotNull(textRootStatus2, "textRootStatus");
        textRootStatus2.setText("Not Rooted!");
        ((TextView) _$_findCachedViewById(R.id.textRootStatus)).setTextColor(getResources().getColor(R.color.colorNoRooted));
        Toast.makeText(this, "Your device isn't Rooted!", 1).show();
    }
}
```

I made a script to check the return value of these two methods and my suspicion was right :

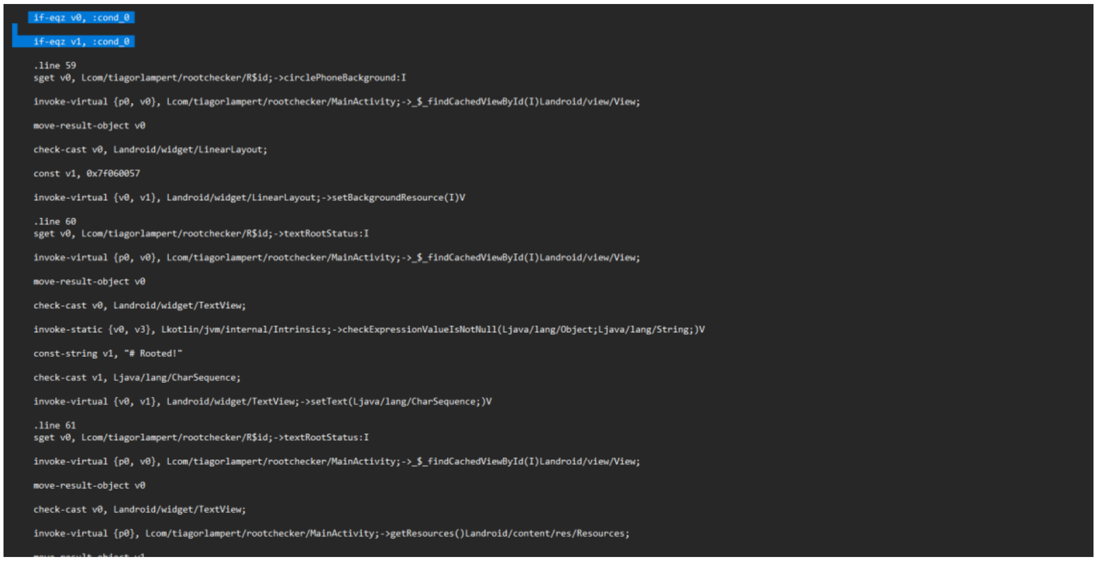


```
PS C:\Users\m.almalki-t\APKs> frida -p 27825 -U -l .\hook_rootchecker.js
     ____
    / _  |   Frida 16.4.8 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to ASUS Z01QD (id=127.0.0.1:21503)

[ASUS Z01QD::PID::27825 ]-> checkRootMethodOne called
checkRootMethodOne result: false
checkRootMethodTwo called
checkRootMethodTwo result: true
```

checkRootMethodOne is returning false that means what ever the return value of **checkRootMethodTwo** it will never detect The Rooted device.

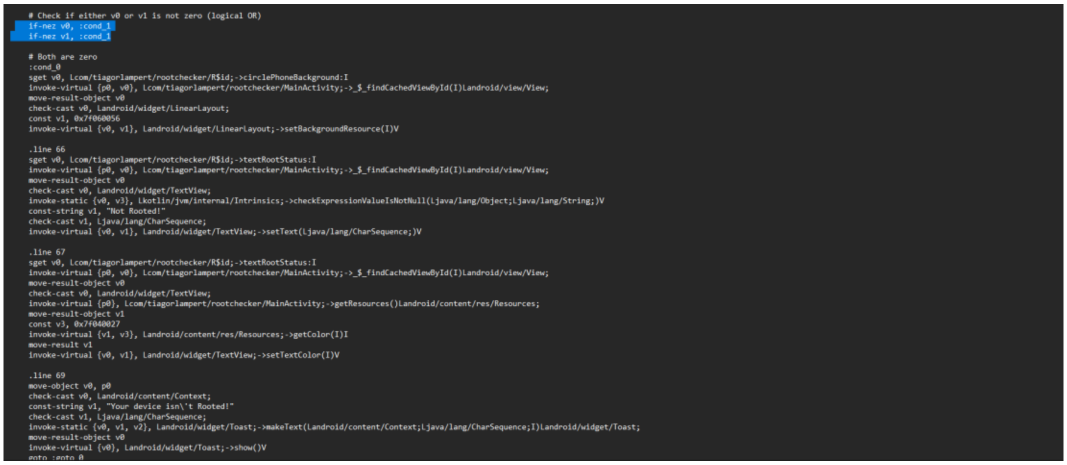
Now I need to fix the Logical error in the **if statement**, I decompiled the app using **apktool** , opend **MainActivity.smali**, an changed the code

from this :



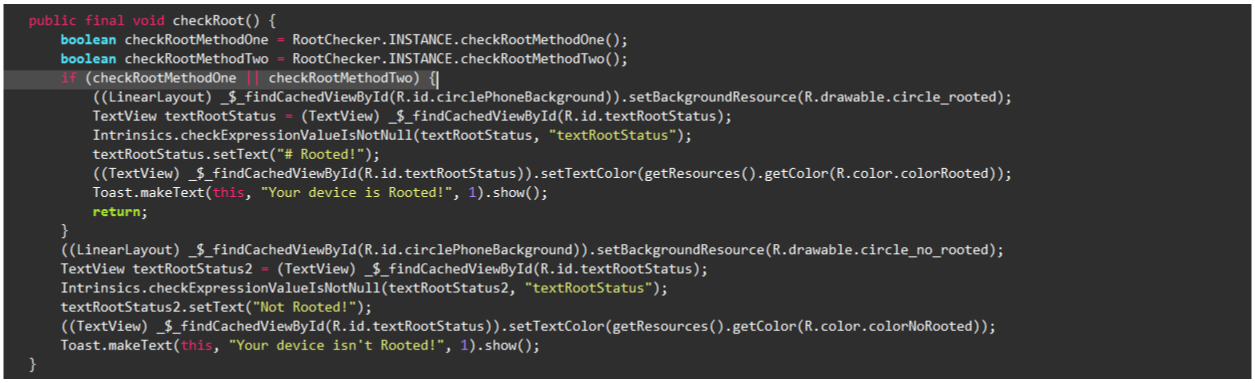
```
    if-eqz v0, :cond_0

    if-eqz v1, :cond_0

    .line 59
    sget v0, Lcom/tiagorlampert/rootchecker/R$id;->circlePhoneBackground:I

    invoke-virtual {p0, v0}, Lcom/tiagorlampert/rootchecker/MainActivity;->_$_findCachedViewById(I)Landroid/view/View;

    move-result-object v0

    check-cast v0, Landroid/widget/LinearLayout;

    const v1, 0x7f060057

    invoke-virtual {v0, v1}, Landroid/widget/LinearLayout;->setBackgroundResource(I)V

    .line 60
    sget v0, Lcom/tiagorlampert/rootchecker/R$id;->textRootStatus:I

    invoke-virtual {p0, v0}, Lcom/tiagorlampert/rootchecker/MainActivity;->_$_findCachedViewById(I)Landroid/view/View;

    move-result-object v0

    check-cast v0, Landroid/widget/TextView;

    invoke-static {v0, v3}, Lkotlin/jvm/internal/Intrinsics;->checkExpressionValueIsNotNull(Ljava/lang/Object;Ljava/lang/String;)V

    const-string v1, "# Rooted!"

    check-cast v1, Ljava/lang/CharSequence;

    invoke-virtual {v0, v1}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V

    .line 61
    sget v0, Lcom/tiagorlampert/rootchecker/R$id;->textRootStatus:I

    invoke-virtual {p0, v0}, Lcom/tiagorlampert/rootchecker/MainActivity;->_$_findCachedViewById(I)Landroid/view/View;

    move-result-object v0

    check-cast v0, Landroid/widget/TextView;

    invoke-virtual {p0}, Lcom/tiagorlampert/rootchecker/MainActivity;->getResources()Landroid/content/res/Resources;

    move-result-object v1
```

To this :



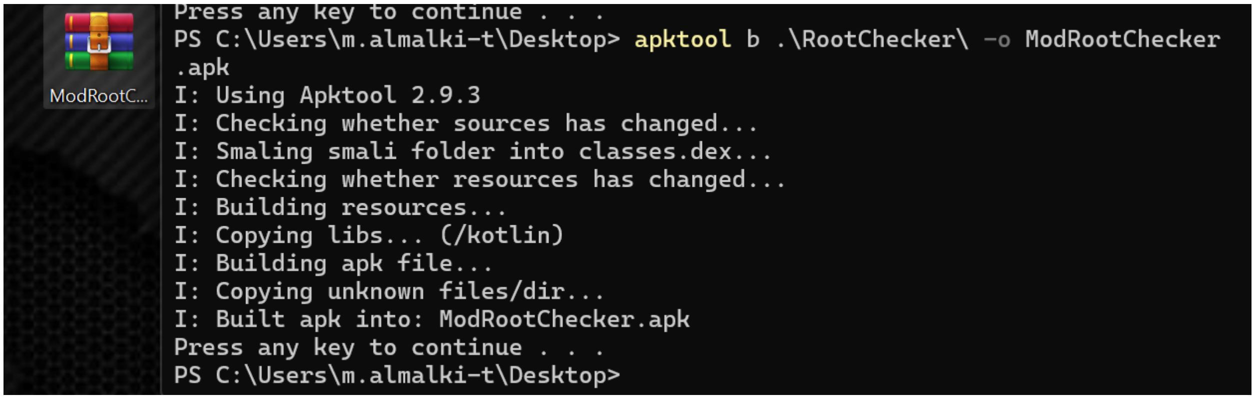
```
    # Check if either v0 or v1 is not zero (logical OR)
    if-nez v0, :cond_1
    if-nez v1, :cond_1

    # Both are zero
    :cond_0
    sget v0, Lcom/tiagorlampert/rootchecker/R$id;->circlePhoneBackground:I
    invoke-virtual {p0, v0}, Lcom/tiagorlampert/rootchecker/MainActivity;->_$_findCachedViewById(I)Landroid/view/View;
    move-result-object v0
    check-cast v0, Landroid/widget/LinearLayout;
    const v1, 0x7f060056
    invoke-virtual {v0, v1}, Landroid/widget/LinearLayout;->setBackgroundResource(I)V

    .line 66
    sget v0, Lcom/tiagorlampert/rootchecker/R$id;->textRootStatus:I
    invoke-virtual {p0, v0}, Lcom/tiagorlampert/rootchecker/MainActivity;->_$_findCachedViewById(I)Landroid/view/View;
    move-result-object v0
    check-cast v0, Landroid/widget/TextView;
    invoke-static {v0, v3}, Lkotlin/jvm/internal/Intrinsics;->checkExpressionValueIsNotNull(Ljava/lang/Object;Ljava/lang/String;)V
    const-string v1, "Not Rooted!"
    check-cast v1, Ljava/lang/CharSequence;
    invoke-virtual {v0, v1}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V

    .line 67
    sget v0, Lcom/tiagorlampert/rootchecker/R$id;->textRootStatus:I
    invoke-virtual {p0, v0}, Lcom/tiagorlampert/rootchecker/MainActivity;->_$_findCachedViewById(I)Landroid/view/View;
    move-result-object v0
    check-cast v0, Landroid/widget/TextView;
    invoke-virtual {p0}, Lcom/tiagorlampert/rootchecker/MainActivity;->getResources()Landroid/content/res/Resources;
    move-result-object v1
    const v3, 0x7f040027
    invoke-virtual {v1, v3}, Landroid/content/res/Resources;->getColor(I)I
    move-result v1
    invoke-virtual {v0, v1}, Landroid/widget/TextView;->setTextColor(I)V

    .line 69
    move-object v0, p0
    check-cast v0, Landroid/content/Context;
    const-string v1, "Your device isn\'t Rooted!"
    check-cast v1, Ljava/lang/CharSequence;
    invoke-static {v0, v1, v2}, Landroid/widget/Toast;->makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/widget/Toast;
    move-result-object v0
    invoke-virtual {v0}, Landroid/widget/Toast;->show()V
    goto :goto_0
```

Now the code looks like this :

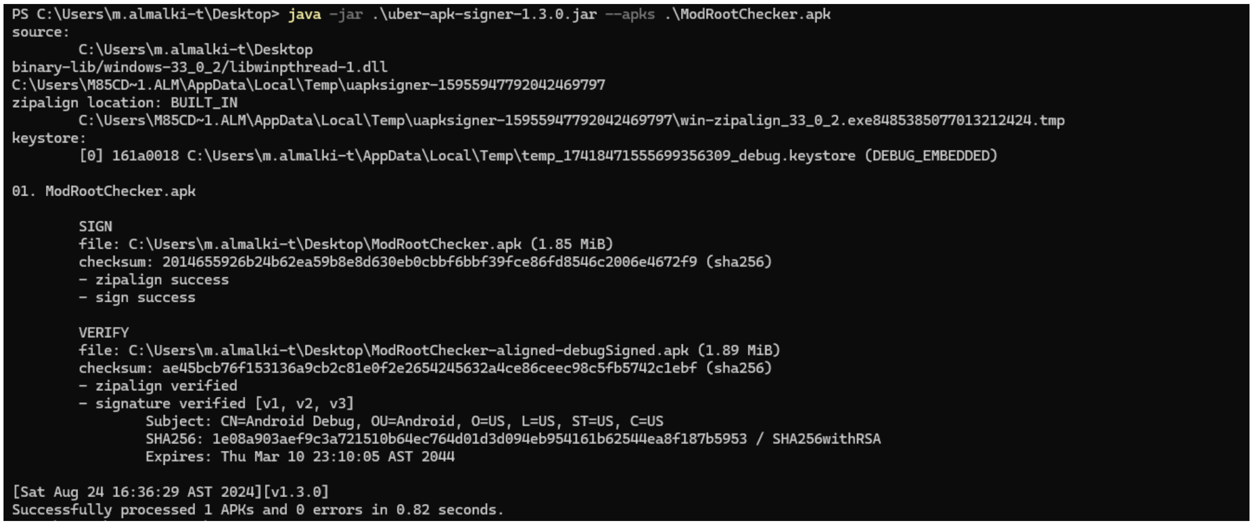


```
public final void checkRoot() {
    boolean checkRootMethodOne = RootChecker.INSTANCE.checkRootMethodOne();
    boolean checkRootMethodTwo = RootChecker.INSTANCE.checkRootMethodTwo();
    if (checkRootMethodOne || checkRootMethodTwo) {
        ((LinearLayout) _$_findCachedViewById(R.id.circlePhoneBackground)).setBackgroundResource(R.drawable.circle_rooted);
        TextView textRootStatus = (TextView) _$_findCachedViewById(R.id.textRootStatus);
        Intrinsics.checkExpressionValueIsNotNull(textRootStatus, "textRootStatus");
        textRootStatus.setText("# Rooted!");
        ((TextView) _$_findCachedViewById(R.id.textRootStatus)).setTextColor(getResources().getColor(R.color.colorRooted));
        Toast.makeText(this, "Your device is Rooted!", 1).show();
        return;
    }
    ((LinearLayout) _$_findCachedViewById(R.id.circlePhoneBackground)).setBackgroundResource(R.drawable.circle_no_rooted);
    TextView textRootStatus2 = (TextView) _$_findCachedViewById(R.id.textRootStatus);
    Intrinsics.checkExpressionValueIsNotNull(textRootStatus2, "textRootStatus");
    textRootStatus2.setText("Not Rooted!");
    ((TextView) _$_findCachedViewById(R.id.textRootStatus)).setTextColor(getResources().getColor(R.color.colorNoRooted));
    Toast.makeText(this, "Your device isn't Rooted!", 1).show();
}
```

I compiled the app and signed it :

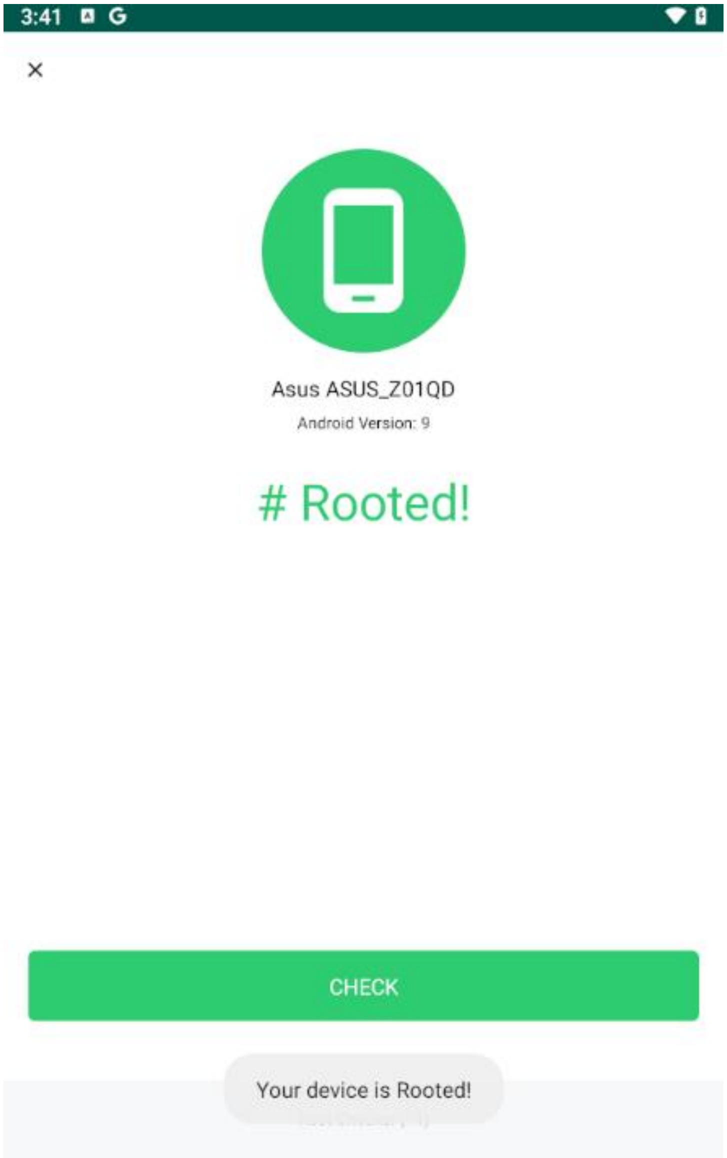


```
Press any key to continue . . .
PS C:\Users\m.almalki-t\Desktop> apktool b .\RootChecker\ -o ModRootChecker
.apk
I: Using Apktool 2.9.3
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/kotlin)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: ModRootChecker.apk
Press any key to continue . . .
PS C:\Users\m.almalki-t\Desktop>
```



```
PS C:\Users\m.almalki-t\Desktop> java -jar .\uber-apk-signer-1.3.0.jar --apks .\ModRootChecker.apk
source:
        C:\Users\m.almalki-t\Desktop
binary-lib/windows-33_0_2/libwinpthread-1.dll
C:\Users\M85CD~1.ALM\AppData\Local\Temp\uapksigner-1595594779204246979
zipalign location: BUILT_IN
        C:\Users\M85CD~1.ALM\AppData\Local\Temp\uapksigner-1595594779204246979\win-zipalign_33_0_2.exe8485385077013212424.tmp
keystore:
        [0] 161a0018 C:\Users\m.almalki-t\AppData\Local\Temp\temp_17418471555699356309_debug.keystore (DEBUG_EMBEDDED)

01. ModRootChecker.apk

        SIGN
        file: C:\Users\m.almalki-t\Desktop\ModRootChecker.apk (1.85 MiB)
        checksum: 2014655926b24b62ea59b8e8d630eb0cbbf6bbf39fce86fd8546c2006e4672f9 (sha256)
        - zipalign success
        - sign success

        VERIFY
        file: C:\Users\m.almalki-t\Desktop\ModRootChecker-aligned-debugSigned.apk (1.89 MiB)
        checksum: ae45bcb76f153136a9cb2c81e0f2e2654245632a4ce86ceec98c5fb5742c1ebf (sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                Subject: CN=Android Debug, OU=Android, O=US, L=US, ST=US, C=US
                SHA256: 1e08a903aef9c3a721510b64ec764d01d3d094eb954161b62544ea8f187b5953 / SHA256withRSA
                Expires: Thu Mar 10 23:10:05 AST 2044

[Sat Aug 24 16:36:29 AST 2024][v1.3.0]
Successfully processed 1 APKs and 0 errors in 0.82 seconds.
```

And installed it and it now works :



Asus ASUS_Z01QD

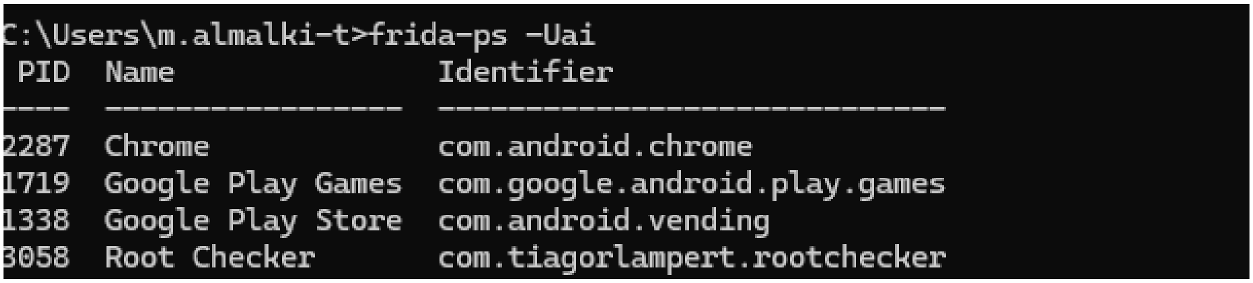
Android Version: 9

Rooted!

Your device is Rooted!

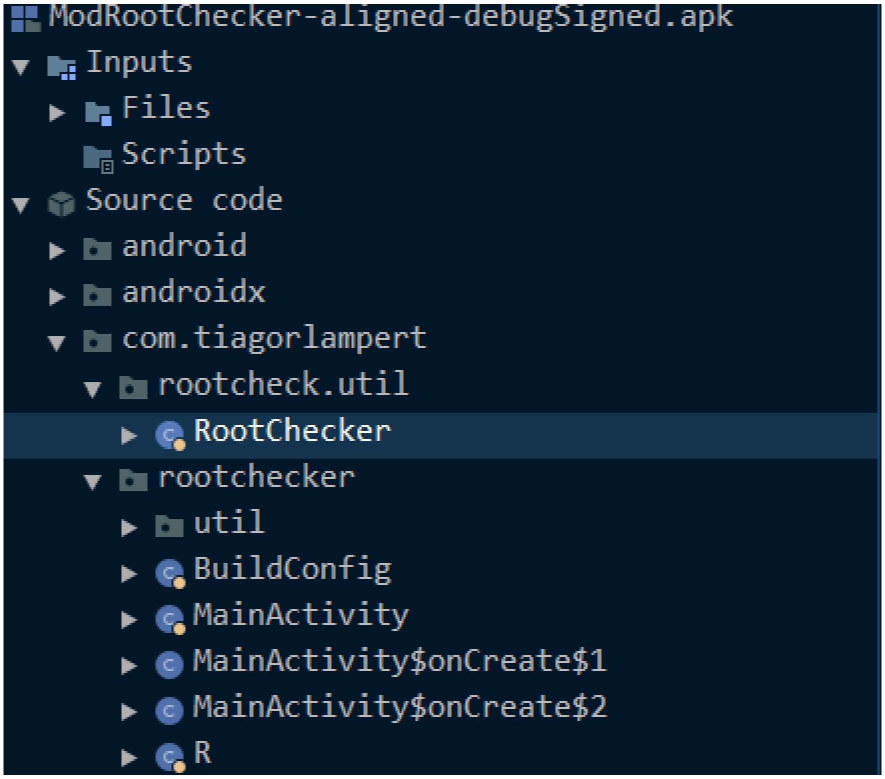
Now I can start the Assessment.

1. Find the process-id of the RootChecker application using Frida.



```
C:\Users\m.almalki-t>frida-ps -Uai
 PID  Name                Identifier
----  ------------------  ------------------------------
2287  Chrome              com.android.chrome
1719  Google Play Games   com.google.android.play.games
1338  Google Play Store   com.android.vending
3058  Root Checker        com.tiagorlampert.rootchecker
```

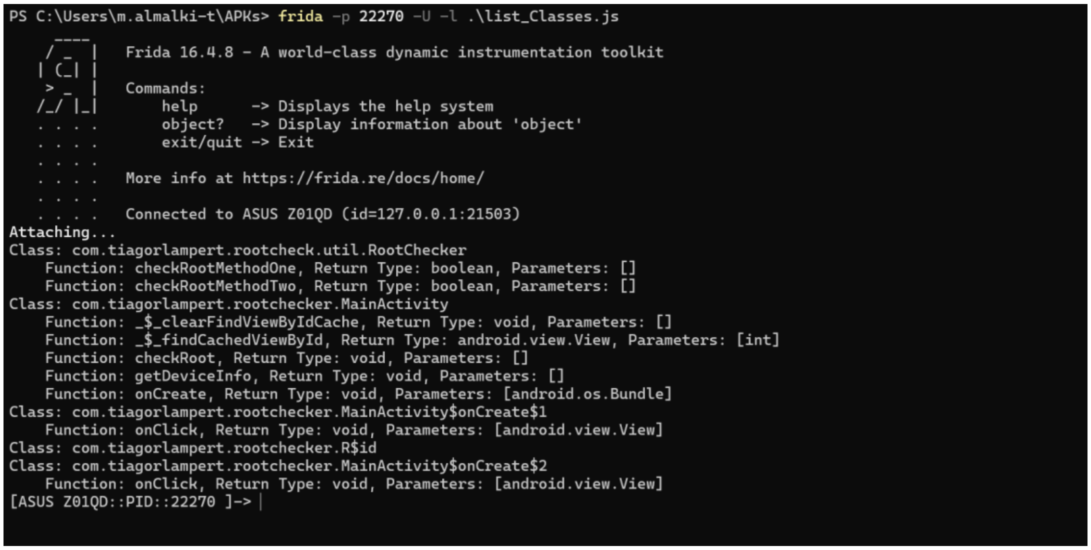
2. List the classes from the installed RootChecker application using Jadx-Gui.



```
ModRootChecker-aligned-debugSigned.apk
▼ Inputs
  ▶ Files
    Scripts
▼ Source code
  ▶ android
  ▶ androidx
  ▼ com.tiagorlampert
    ▼ rootcheck.util
      ▶ RootChecker
    ▼ rootchecker
      ▶ util
      ▶ BuildConfig
      ▶ MainActivity
      ▶ MainActivity$onCreate$1
      ▶ MainActivity$onCreate$2
      ▶ R
```

You can also search throw the classes

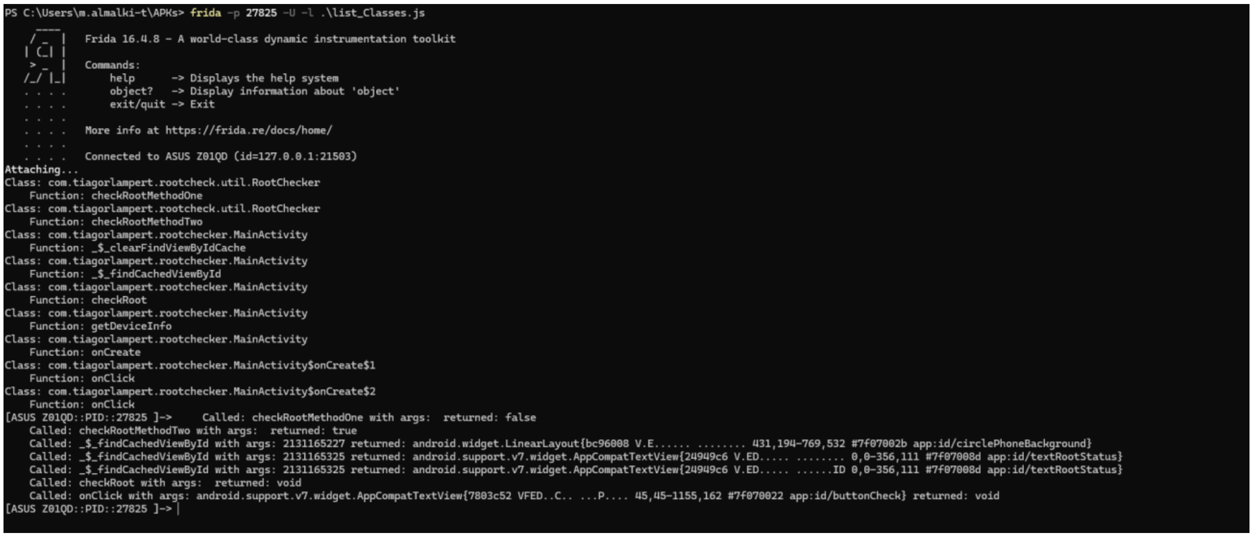
3. List functions from RootChecker application.



```
PS C:\Users\m.almalki-t\APKs> frida -p 22270 -U -l .\list_Classes.js
     ____
    / _  |   Frida 16.4.8 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to ASUS Z01QD (id=127.0.0.1:21503)
Attaching...
Class: com.tiagorlampert.rootcheck.util.RootChecker
    Function: checkRootMethodOne, Return Type: boolean, Parameters: []
    Function: checkRootMethodTwo, Return Type: boolean, Parameters: []
Class: com.tiagorlampert.rootchecker.MainActivity
    Function: _$_clearFindViewByIdCache, Return Type: void, Parameters: []
    Function: _$_findCachedViewById, Return Type: android.view.View, Parameters: [int]
    Function: checkRoot, Return Type: void, Parameters: []
    Function: getDeviceInfo, Return Type: void, Parameters: []
    Function: onCreate, Return Type: void, Parameters: [android.os.Bundle]
Class: com.tiagorlampert.rootchecker.MainActivity$onCreate$1
    Function: onClick, Return Type: void, Parameters: [android.view.View]
Class: com.tiagorlampert.rootchecker.R$id
Class: com.tiagorlampert.rootchecker.MainActivity$onCreate$2
    Function: onClick, Return Type: void, Parameters: [android.view.View]
[ASUS Z01QD::PID::22270 ]-> |
```

Execute a script to list all functions in all classes within the application, detailing arguments and return types for each function.

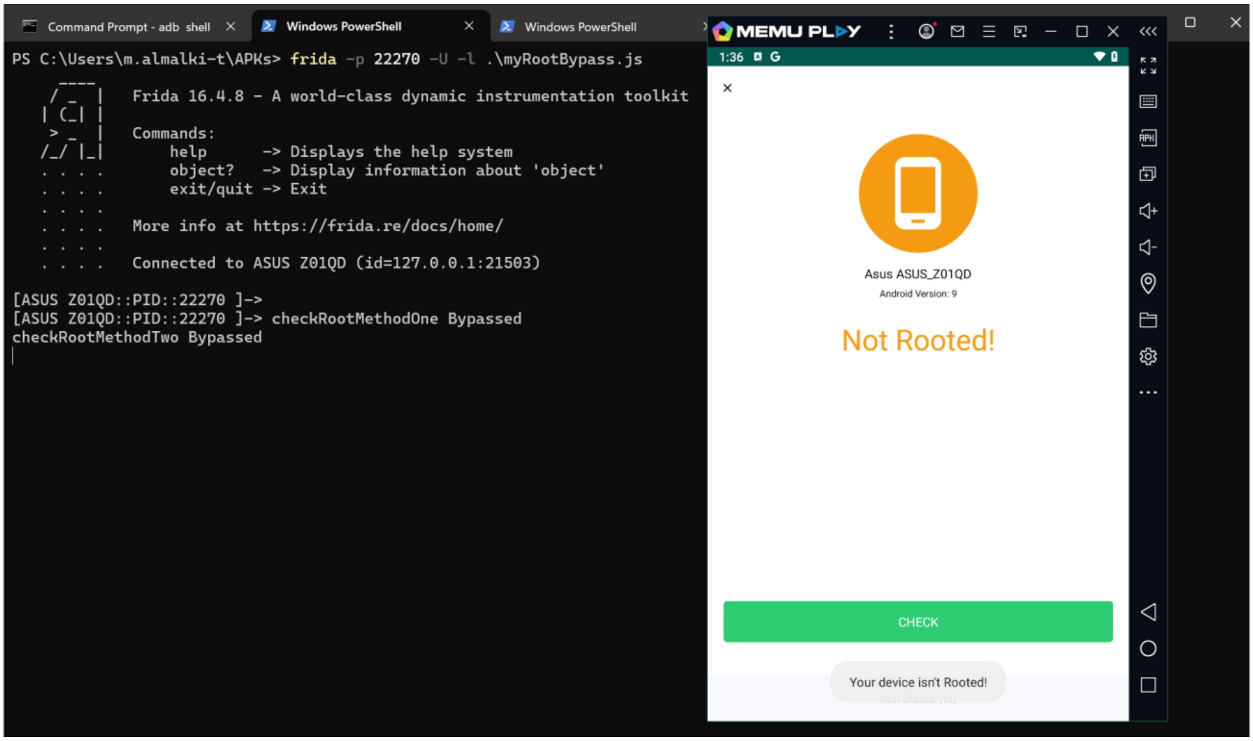
4. Display the return value of those functions.



```
PS C:\Users\m.almalki-t\APKs> frida -p 27825 -U -l .\list_Classes.js
     ____
    / _  |   Frida 16.4.8 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to ASUS Z01QD (id=127.0.0.1:21503)
Attaching...
Class: com.tiagorlampert.rootcheck.util.RootChecker
    Function: checkRootMethodOne
Class: com.tiagorlampert.rootcheck.util.RootChecker
    Function: checkRootMethodTwo
Class: com.tiagorlampert.rootchecker.MainActivity
    Function: _$_clearFindViewByIdCache
Class: com.tiagorlampert.rootchecker.MainActivity
    Function: _$_findCachedViewById
Class: com.tiagorlampert.rootchecker.MainActivity
    Function: checkRoot
Class: com.tiagorlampert.rootchecker.MainActivity
    Function: getDeviceInfo
Class: com.tiagorlampert.rootchecker.MainActivity
    Function: onCreate
Class: com.tiagorlampert.rootchecker.MainActivity$onCreate$1
    Function: onClick
Class: com.tiagorlampert.rootchecker.MainActivity$onCreate$2
    Function: onClick
[ASUS Z01QD::PID::27825 ]->     Called: checkRootMethodOne with args:  returned: false
    Called: checkRootMethodTwo with args:  returned: true
    Called: _$_findCachedViewById with args: 2131165227 returned: android.widget.LinearLayout{bc96008 V.E...... ........ 431,194-769,532 #7f07002b app:id/circlePhoneBackground}
    Called: _$_findCachedViewById with args: 2131165325 returned: android.support.v7.widget.AppCompatTextView{24949c6 V.ED..... ........ 0,0-356,111 #7f07008d app:id/textRootStatus}
    Called: _$_findCachedViewById with args: 2131165325 returned: android.support.v7.widget.AppCompatTextView{24949c6 V.ED..... ......ID 0,0-356,111 #7f07008d app:id/textRootStatus}
    Called: checkRoot with args:  returned: void
    Called: onClick with args: android.support.v7.widget.AppCompatTextView{7803c52 VFED..C.. ...P.... 45,45-1155,162 #7f070022 app:id/buttonCheck} returned: void
[ASUS Z01QD::PID::27825 ]-> |
```

Implement a script to hook into each function, displaying the return values when functions are called.

5. Ensure that the RootChecker applications always shows that the device **NOT** rooted.



```
PS C:\Users\m.almalki-t\APKs> frida -p 22270 -U -l .\myRootBypass.js
     ____
    / _  |   Frida 16.4.8 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to ASUS Z01QD (id=127.0.0.1:21503)

[ASUS Z01QD::PID::22270 ]->
[ASUS Z01QD::PID::22270 ]-> checkRootMethodOne Bypassed
checkRootMethodTwo Bypassed
```

Asus ASUS_Z01QD

Android Version: 9

Not Rooted!

Your device isn't Rooted!

Thank you for reviewing this summary of the issues I faced and how they were resolved. All scripts used for the assessment and the modified version of the app are available in the repository.

- Majed Almalki