

# bugKuCTF第四道reverse题目Timer(阿里CTF)writeup

转载

iqiqiya 于 2018-05-13 17:41:46 发布 4351 收藏  
分类专栏: [-----bugkuCTF 我的逆向之路 我的CTF进阶之路](#) 文章标签: [alictfbugkuCTF Timer](#)



[-----bugkuCTF 同时被 3 个专栏收录](#)

9 篇文章 0 订阅  
订阅专栏



[我的逆向之路](#)

108 篇文章 10 订阅  
订阅专栏

[我的CTF进阶之路](#)

108 篇文章 18 订阅  
订阅专栏

题目截图:

## Timer(阿里CTF) 60

flag格式

flag{}

链接: <https://pan.baidu.com/s/1vIU3YKmxo0fnBm4rSciT8g> 密码: i1py

Flag

Submit

<https://blog.esdn.net/xiangshangbashaonian>

题目链接: <https://pan.baidu.com/s/1vIU3YKmxo0fnBm4rSciT8g> 密码: i1py

### 解析:

apk安装之后出现一个读秒的, 20万秒, 大于3600, 所以大于一个小时, 所以太慢了, 不等了。

jeb反编译, MainActivity代码:

```
package net.bluelotus.tomorrow.easyandroid;

import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;

public class MainActivity extends AppCompatActivity {
    int beg;
    int k;
```

```

int now;
long t;

static {
    System.loadLibrary("lhm");
}

public MainActivity() {
    super();
    this.beg = (((int)(System.currentTimeMillis() / 1000)) + 200000);
    this.k = 0;
    this.t = 0;
}

public static boolean is2(int n) {
    boolean v1 = true;
    if(n > 3) {
        if(n % 2 != 0 && n % 3 != 0) {
            int v0 = 5;
            while(true) {
                if(v0 * v0 <= n) {
                    if(n % v0 != 0 && n % (v0 + 2) != 0) {
                        v0 += 6;
                        continue;
                    }

                    return false;
                }
                else {
                    return v1;
                }
            }

            return false;
        }

        v1 = false;
    }
    else if(n <= 1) {
        v1 = false;
    }

    return v1;
}

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    this setContentView(2130968600);
    View v2 = this.findViewById(2131492944);
    View v3 = this.findViewById(2131492945);
    Handler v0 = new Handler();
    v0.postDelayed(new Runnable() {
        public void run() {
            MainActivity.this.t = System.currentTimeMillis();
            MainActivity.this.now = ((int)(MainActivity.this.t / 1000));
            MainActivity.this.t = 1500 - MainActivity.this.t % 1000;
            this.val$tv2.setText("AliCTF");
            if(MainActivity.this.beg - MainActivity.this.now <= 0) {
                this.val$tv1.setText("The flag is:");
                this.val$tv2.setText("alictf/" + MainActivity.this.stringFromJNI2(MainActivity.this.now));
            }
        }
    }, 1500);
}

```

```

        this.val$tv2.setText("alict{" + MainActivity.this.stringFromJNI2(MainActivity.this.k) + "}");
    }

    if(MainActivity.is2(MainActivity.this.beg - MainActivity.this.now)) {
        MainActivity.this.k += 100;
    }
    else {
        --MainActivity.this.k;
    }

    this.val$tv1.setText("Time Remaining(s):" + (MainActivity.this.beg - MainActivity.this.now));
    this.val$handler.postDelayed(((Runnable)this), MainActivity.this.t);
}
}, 0);
}

public boolean onCreateOptionsMenu(Menu menu) {
    this.getMenuInflater().inflate(2131558400, menu);
    return 1;
}

public boolean onOptionsItemSelected(MenuItem item) {
    boolean v1 = item.getItemId() == 2131492959 ? true : super.onOptionsItemSelected(item);
    return v1;
}

public native String stringFromJNI2(int arg1) {
}
}

```

关键点:

```

if(MainActivity.this.beg - MainActivity.this.now <= 0) {
    this.val$tv1.setText("The flag is:");
    this.val$tv2.setText("alict{" + MainActivity.this.stringFromJNI2(MainActivity.this.k) + "}");
}

```

找到beg和now的点:

```

public MainActivity() {
    super();
    this.beg = (((int)(System.currentTimeMillis() / 1000)) + 200000); //打开activity时候的时间 (除以
    this.k = 0;
    this.t = 0;
}

```

以及`MainActivity.this.now = ((int)(MainActivity.this.t / 1000));`即判断的时候的秒  
那么`beg - now`就是200000+开始的时间-判断的时间，也就是200000-已经过去的时间，所以就是如果过去了200000秒，就可以出现flag了。

呈现flag是通过native层的`this.val$tv2.setText("alictf{" + MainActivity.this.stringFromJNI2(MainActivity.this.k) + "})`；也就是`stringFromJNI2`来呈现的，我们不必逆向so，这个参数是一个k，如果真的去等200000秒（虽然不可能），那么这里的k应该是多少呢？

跟k相关的两个关键点如下：

```
public static boolean is2(int n) {
    boolean v1 = true;
    if(n > 3) {
        if(n % 2 != 0 && n % 3 != 0) {
            int v0 = 5;
            while(true) {
                if(v0 * v0 <= n) {
                    if(n % v0 != 0 && n % (v0 + 2) != 0) {
                        v0 += 6;
                        continue;
                    }

                    return false;
                }
                else {
                    return v1;
                }
            }

            return false;
        }

        v1 = false;
    }
    else if(n <= 1) {
        v1 = false;
    }

    return v1;
}

if(MainActivity.is2(MainActivity.this.beg - MainActivity.this.now)) {
    MainActivity.this.k += 100;
}
else {
    --MainActivity.this.k;
}
```

总结思路：通过`beg - now`代入`is2`函数对k进行操作，200000，需要一秒一秒的操作，而我们，直接通过写代码模拟出200000的结果，找到k，然后改动k值，直接调用，就可以得到flag了。

代码

is2是用java写的，懒得看逻辑，直接复制到java里边：

```
package test;

public class Main {

    public static boolean is2(int n) {
        boolean v1 = true;
        if(n > 3) {
            if(n % 2 != 0 && n % 3 != 0) {
                int v0 = 5;
                while(true) {
                    if(v0 * v0 <= n) {
                        if(n % v0 != 0 && n % (v0 + 2) != 0) {
                            v0 += 6;
                            continue;
                        }
                    }
                    return false;
                }
            }
            else {
                return v1;
            }
        }
        v1 = false;
    }
    else if(n <= 1) {
        v1 = false;
    }

    return v1;
}

public static void main(String args[]) {
    int time = 200000;
    int k = 0;
    while (time > 0) {
        if (is2(time)) {
            k += 100;
        }
        else {
            k--;
        }
        time--;
    }
    System.out.println(k);
}
}
```

运行得到k值，然后apktool 反编译原apk,更改smali，两个点：

- 将输出flag的条件反过来，即MainActivity\$1.smali中的 if-gtz v0, :cond\_0这句话（后面是输出The Flag Is那里的跳转）改为 if-ltz v0, :cond\_0
- 将k值改为常量，即上文提到的smali文件中的 iget v3, v3 ....(省略);->k:I之后添加 const v3, 1616384

重新打包，签名，安装，得到flag为alictf{Y0vAr3TimerMa3te7}

转载自：[https://blog.csdn.net/qq\\_29343201/article/details/51607527](https://blog.csdn.net/qq_29343201/article/details/51607527)