

Android应用防止so注入防止动态调试参考代码

2022-03-19 09:04:20 By 小峰

由于公司应用需要过安全测试，测试那边说有so注入漏洞，所以找到了这份代码，并成功通过测试。

废话不多说。直接上代码

```
//非Debug 编译，反调试检测
if(!BuildConfig.DEBUG) {
    if(isDebuggable()) {
        exit(0);
    }

    Thread t = new Thread(new Runnable() {
        @Override
        public void run() {
            while(true) {
                try {
                    sleep(100);
                    if(Debug.isDebuggerConnected()) {
                        exit(0);
                    }

                    if(isUnderTraced()) {
                        exit(0);
                    }
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }, "SafeGuardThread");
    t.start();
}

if(isUnderTraced()) {
    System.exit(0);
}

CrashReport.initCrashReport(getApplicationContext(), "4abcbaaf23", BuildConfig.DEBUG);
}

private boolean isUnderTraced() {
    String processStatusFilePath = String.format(Locale.US, "/proc/%d/status", android.os.Process.m
yPid());
    File procInfoFile = new File(processStatusFilePath);
    try {
        BufferedReader b = new BufferedReader(new FileReader(procInfoFile));
        String readLine;
```

```
while ((readLine = b.readLine()) != null) {
    if(readLine.contains("TracerPid")) {
        String[] arrays = readLine.split(":");
        if(arrays.length == 2) {
            int tracerPid = Integer.parseInt(arrays[1].trim());
            if(tracerPid != 0) {
                return true;
            }
        }
    }
}

b.close();
} catch (Exception e) {
    e.printStackTrace();
}

return false;
}

public boolean isDebuggable() {
    return 0 != ( getApplicationInfo().flags & ApplicationInfo.FLAG_DEBUGGABLE );
}
```

版权声明：

本站遵循 [署名-非商业性使用-相同方式共享 2.5](#) 共享协议。

转载请注明转自[疯人峰语](#)并标明URL。

本文链接：<https://www.iiii.name/?p=2562>