

**Phòng thí nghiệm An toàn Thông tin**

**Trường Đại học Công nghệ Thông tin (UIT)**

# BÁO CÁO THỰC HÀNH

## Môn học: Bảo mật Web và Ứng dụng Kỳ báo cáo: Buổi 04 (Session 04)

**Tên chủ đề: Pentesting Android Application**

*GVHD: Ngô Đức Hoàng Sơn Ngày báo cáo: 3/5/2024* **Nhóm: 09**

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT213.O22.ATCL.2

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Hồ Ngọc Thiện | 21522620 | [21522620@gm.uit.edu.vn](mailto:21522620@gm.uit.edu.vn) |

## NỘI DUNG THỰC HIỆN:1

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Kịch bản 01 | 100% |
| 2 | Kịch bản 02 | 100% |
| 3 | Kịch bản 03 | 100% |
| 4 | Kịch bản 04 | 100% |
| 5 | Kịch bản 05 | 100% |
| 6 | Kịch bản 06 | 100% |
| 7 | Kịch bản 07 | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

1 Ghi nội dung công việc, các kịch bản trong bài Thực hành

**2**

# BÁO CÁO CHI TIẾT

**D.2 Droid**

A close-up of a text

Description automatically generated

**Challenge 1:**

Sau khi ta nhập 1 chuỗi bất kì, chỉ hiện ra dòng “Not today…”.

A screenshot of a computer

Description automatically generated

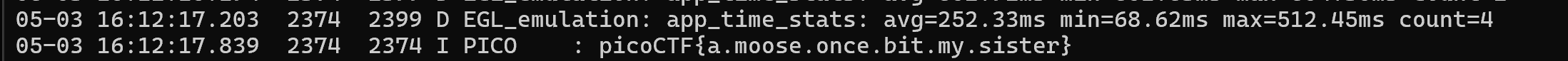
Có vẻ không có ý nghĩa gì, giờ thử dịch ngược file apk này và đọc mã nguồn của nó để phân tích. A screen shot of a computer program

Description automatically generated

Trong file **com.hellocmu.picoctf.FlagstaffHill.class,** ta thấy sau khi nhấn nút “Hello, I am a button” thì flag sẽ được in ra ở log. Vậy ta chỉ cần đi tìm pid của ứng dụng và đọc flag từ log là hoàn thành challenge này.



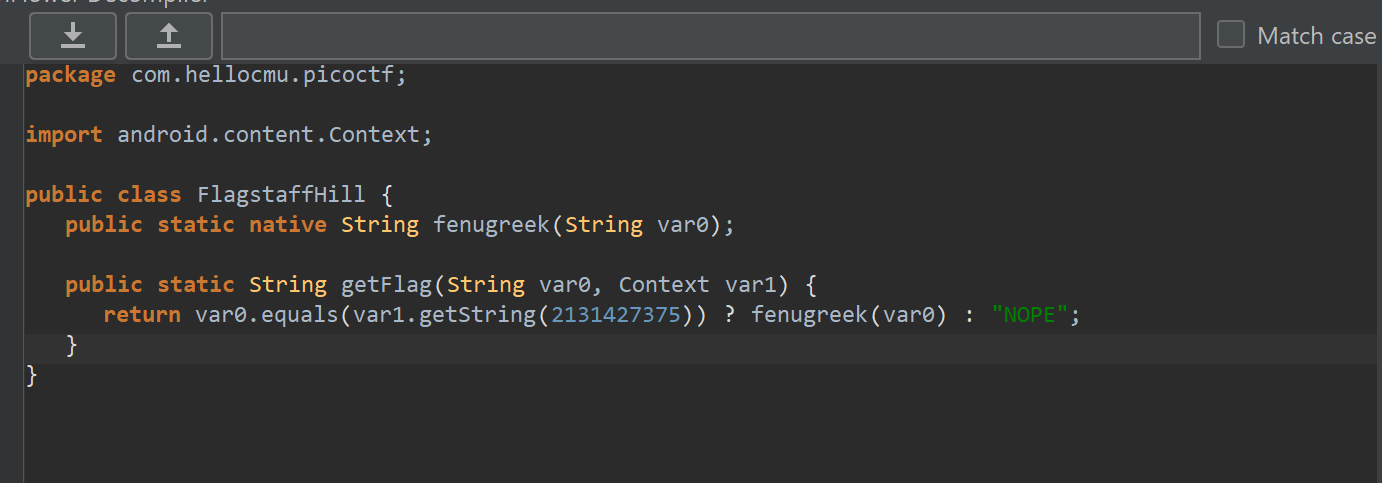
Ta thấy pid của app là 2374, đọc log nữa thôi.



Flag tìm được: **picoCTF{a.moose.once.bit.my.sister}**

**Challenge 2:**

Ở challenge này ta cũng tiến hành dùng Bytecode Viewer để đọc mã nguồn của chương trình. Ta đọc file **com.hellocmu.picoctf.FlagstaffHill.class**



A screen shot of a computer code

Description automatically generated

Ta thấy hàm **getFlag()** nhận 2 đầu vào là chuỗi input ta nhập vào (var0) và một đối tượng context (var1), sau đó so sánh giá trị của var0 và giá trị chuỗi của Context var1 sau khi thực hiện hàm **getString(2131427375)**. Nếu bằng nhau thì thực hiện hàm **fenugreek()** để in ra flag.

Hmm, nhưng làm sao ta tìm được chuỗi giá trị của biến var1. Ta có thể sử dụng hook script để thử in ra nó.

A screen shot of a computer

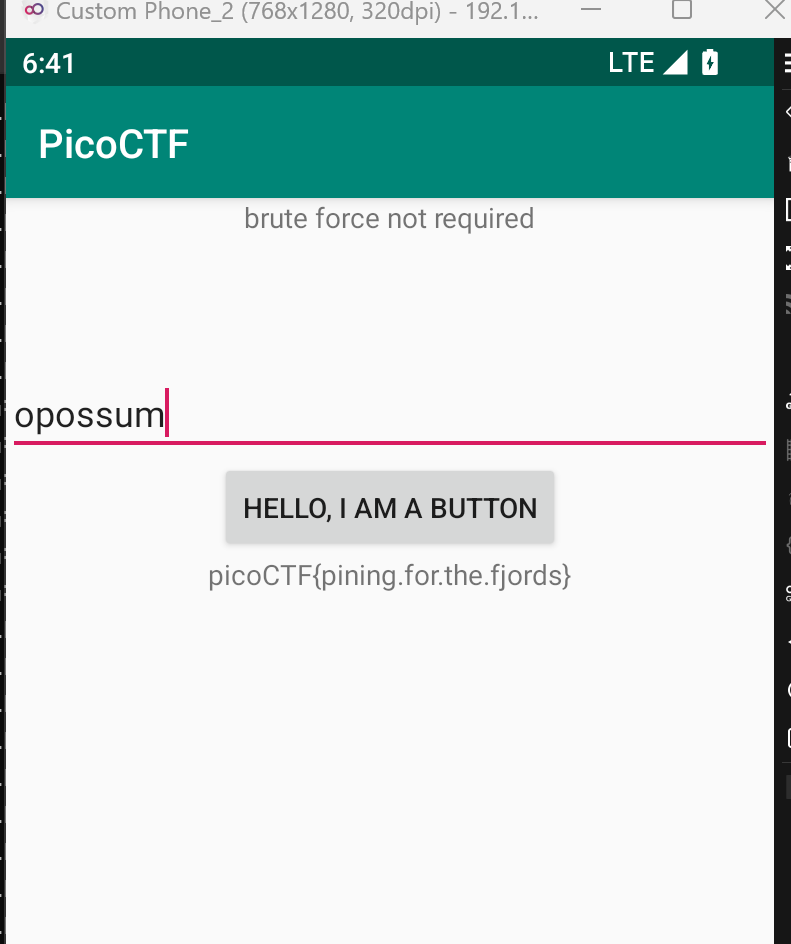
Description automatically generated

Chạy code trên và nhập 1 giá trị bất kì và submit. Ta thấy được chuỗi “opossum”

A screenshot of a computer

Description automatically generated

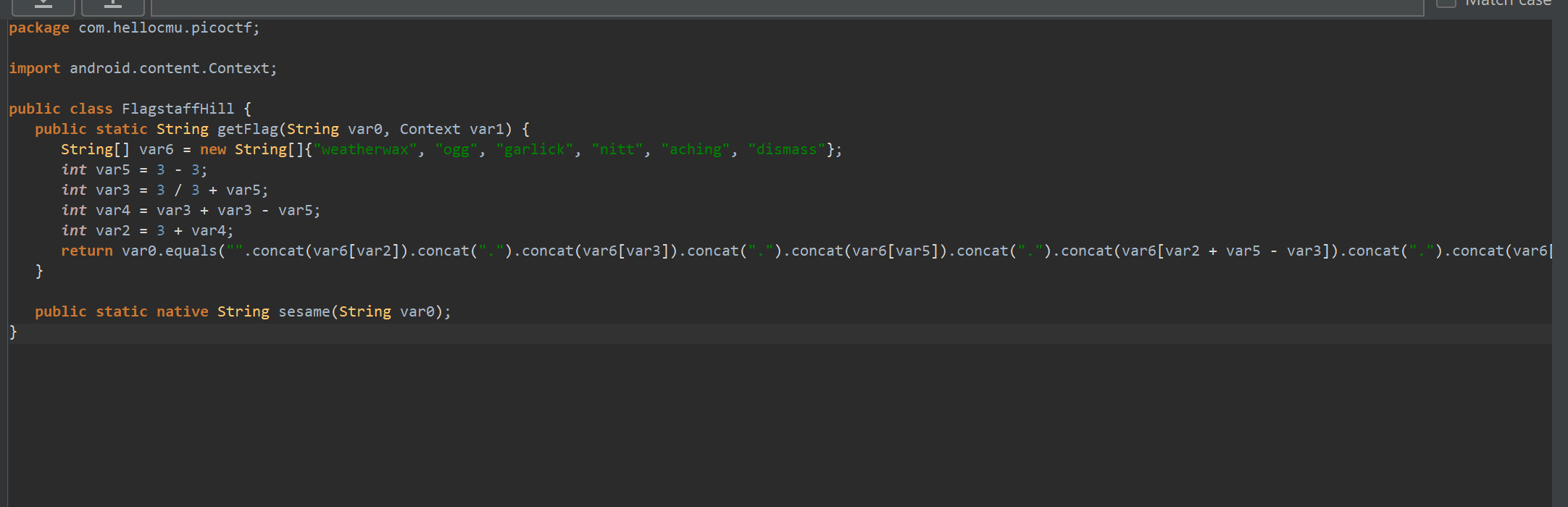
Tiến hành chạy lại ứng dụng và nhập chuỗi “opossum”. Ta thu được flag



Flag tìm thấy: **picoCTF{pining.for.the.fjords}**

**Challenge 3:**

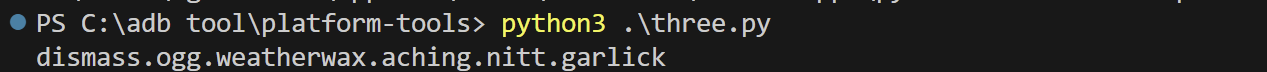
Tiếp tục với Bytecode Viewer như mọi khi thôi 😊



Ta thấy nó so sánh chuỗi ta nhập vào và chuỗi được nối từ các phần tử trong mảng var6, nếu bằng nhau thì gọi hàm **sesame** để in ra flag. Ta có thể tìm chuỗi cần nhập bằng code python sau

A black rectangular object with white text

Description automatically generated



Nhập chuỗi vào ứng dụng và lấy flag thôi.

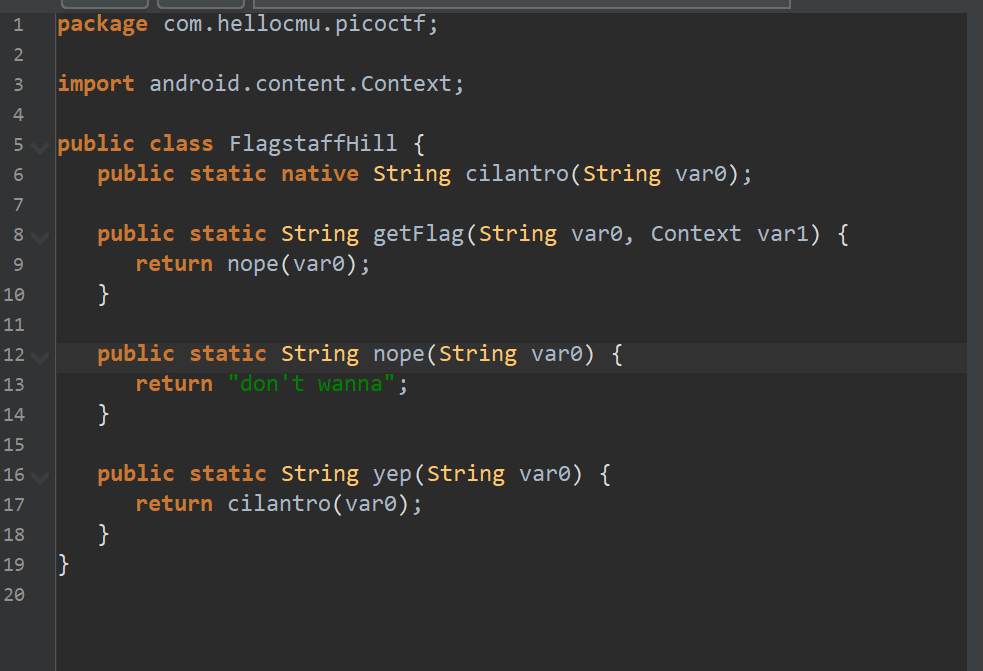
A screenshot of a phone

Description automatically generated

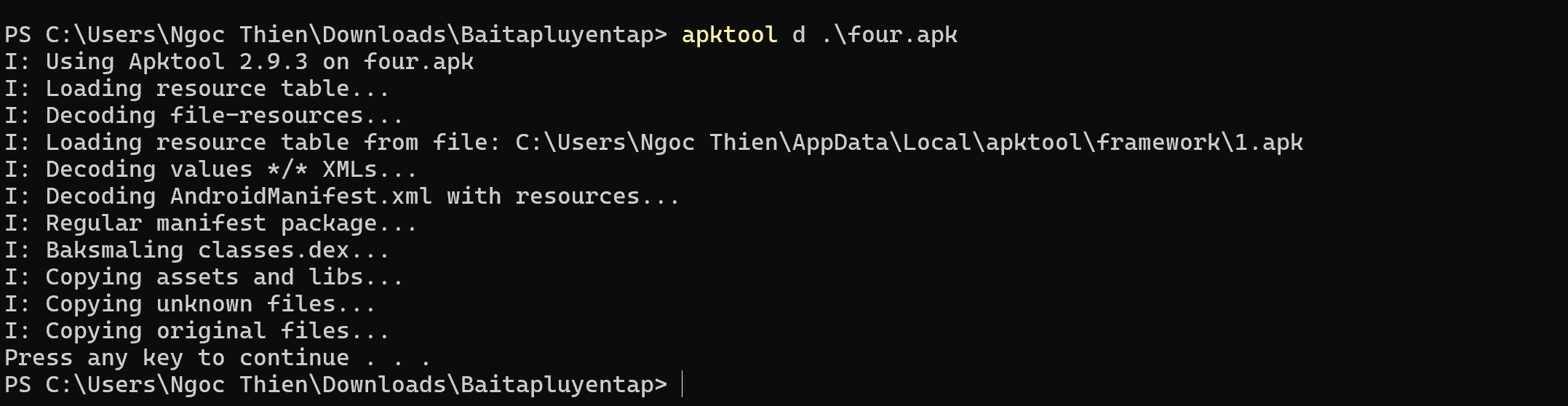
Flag tìm được: **picoCTF{what.is.your.favorite.colour}**

**Challenge 4:**

Tiếp tục với Bytecode Viewer, ta kiểm tra mã nguồn của chương trình



Ta thấy hàm **getFlag()** đang trả về giá trị của hàm **nope(),** 1 giá trị mà ta không cần. Kết quả chúng ta cần là giá trị trả về của hàm **cilantro()**. Có vẻ có thể patch trong file smali được decompile. Ta tiến hành decompile file apk.



Đọc tập tin và tìm thấy method **getFlag()**

A computer screen shot of a computer code

Description automatically generated

Ta đổi hàm nope thành yep, sau đó kí và chạy lại chương trình.

A computer screen shot of a black screen

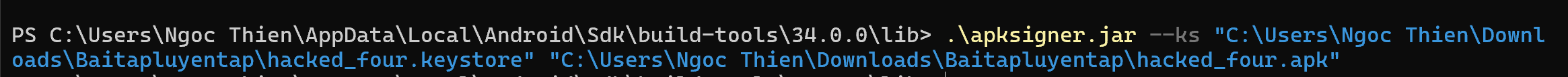
Description automatically generated

A screen shot of a computer

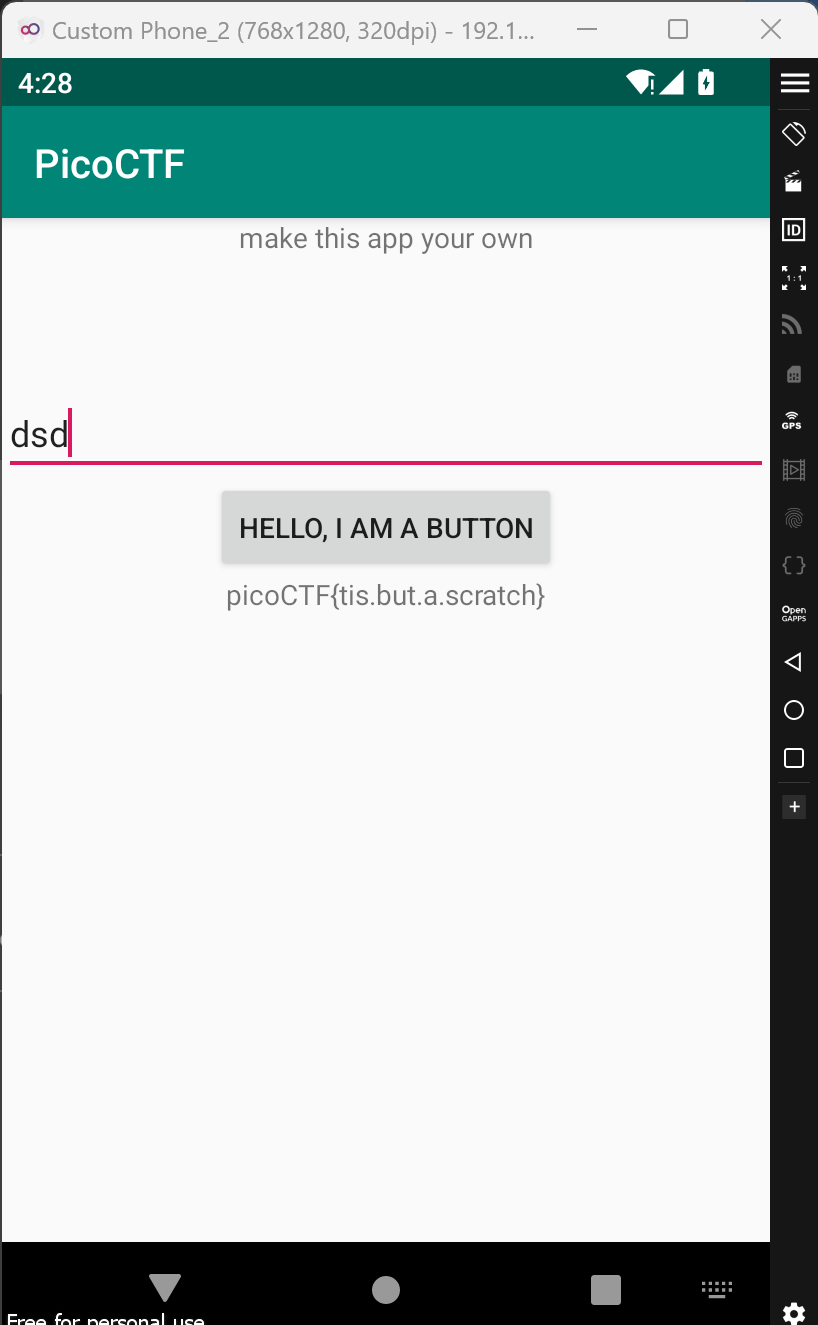
Description automatically generated

A screen shot of a computer

Description automatically generated



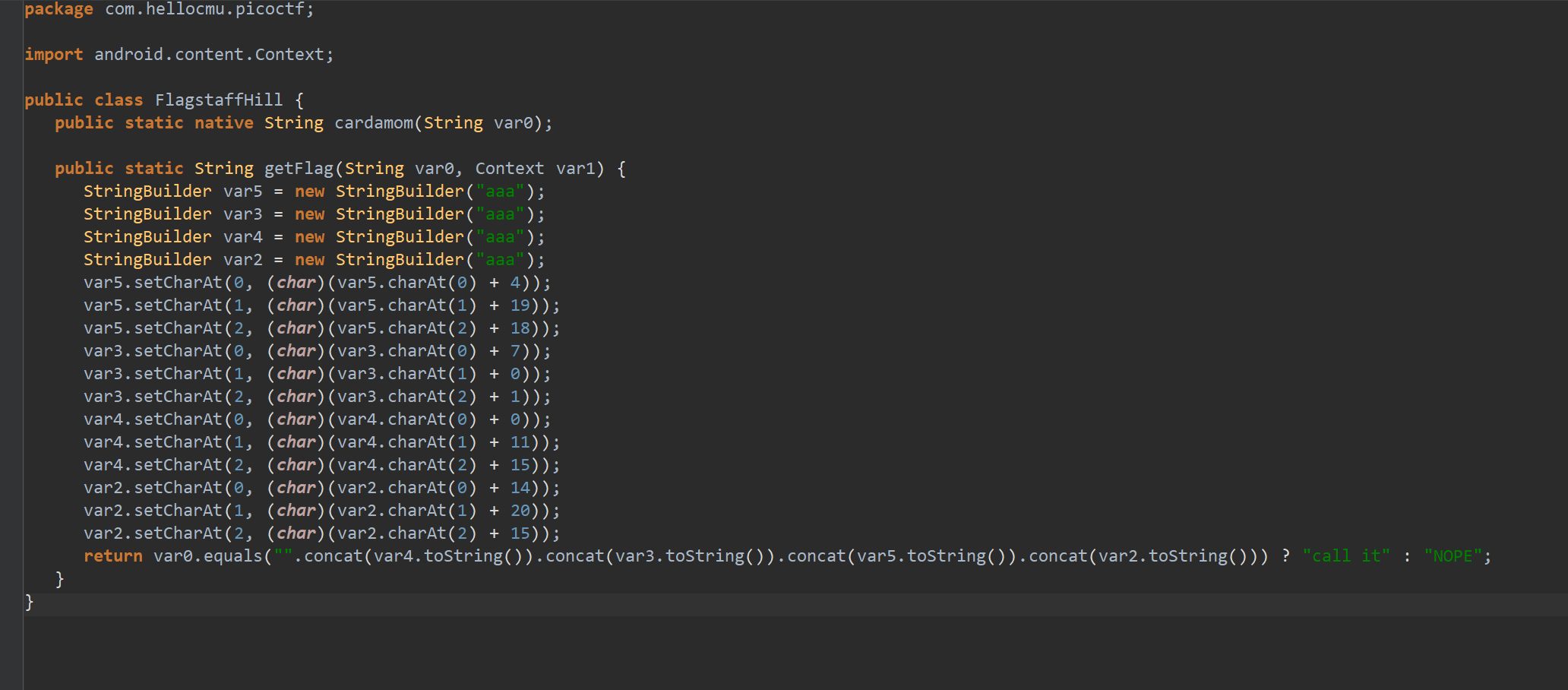
Sau đó ta tiến hành bật ứng dụng và thu flag



Flag tìm thấy: **picoCTF{tis.but.a.scratch}**

**Challenge 5:**

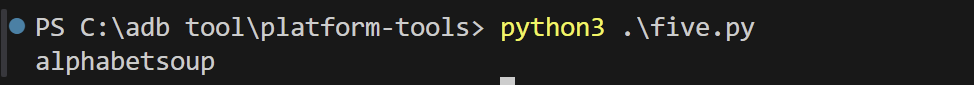
Với Bytecode Viewer, ta đọc mã nguồn của chương trình



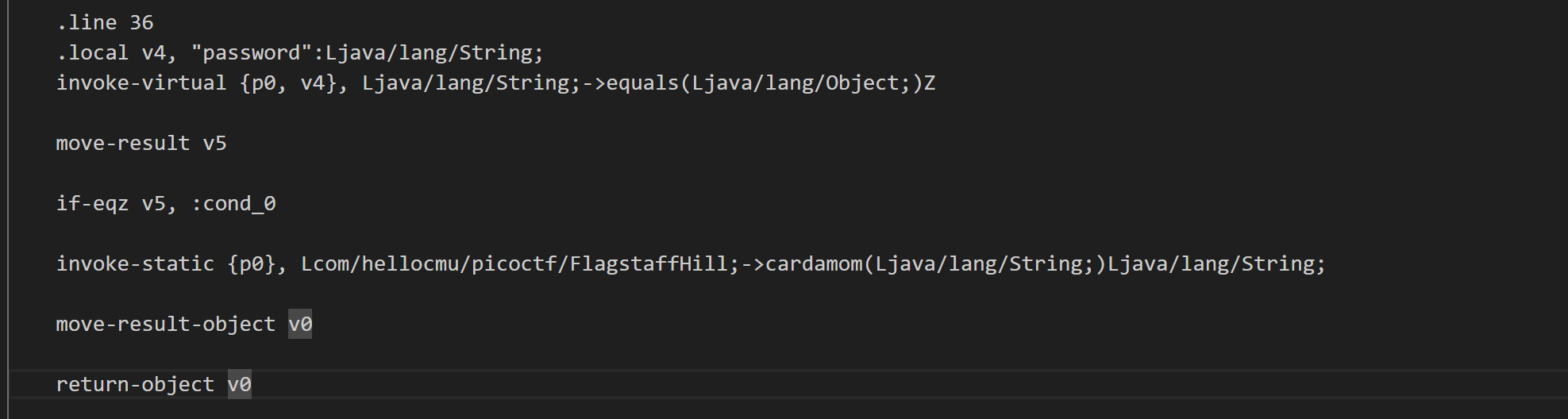
Nó bắt chúng ta nối chuỗi để rồi nếu nối đúng thì chỉ return được 1 dòng “call it” :))))). Vậy ta tiến hành patch lại app này để nó gọi được hàm **cardamom().** Decompile và đọc file smali. Đầu tiên ta nối chuỗi lại để thu được giá trị truyền vào hàm **cardamom()**

A screenshot of a computer program

Description automatically generated



Sau đó, vào file smali và sửa cho nó gọi hàm **cardamom()** khi đã nhập chuỗi đúng.



Build lại, ký và chạy lại app

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated



Cài đặt lại và bật ứng dụng lên để lấy flag

A screenshot of a computer

Description automatically generated

Flag tìm thấy: **picoCTF{not.particularly.silly}**

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

## thường gặp Nhóm 09

**19**

**YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hành theo hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

## Báo cáo:

* File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-SessionX\_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).

*Ví dụ: [*NT101.K11.ATCL*]-Session1\_Group3.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài Lab.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

* Chuẩn bị tốt.
* Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây

dựng bài thực hành.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**