

**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

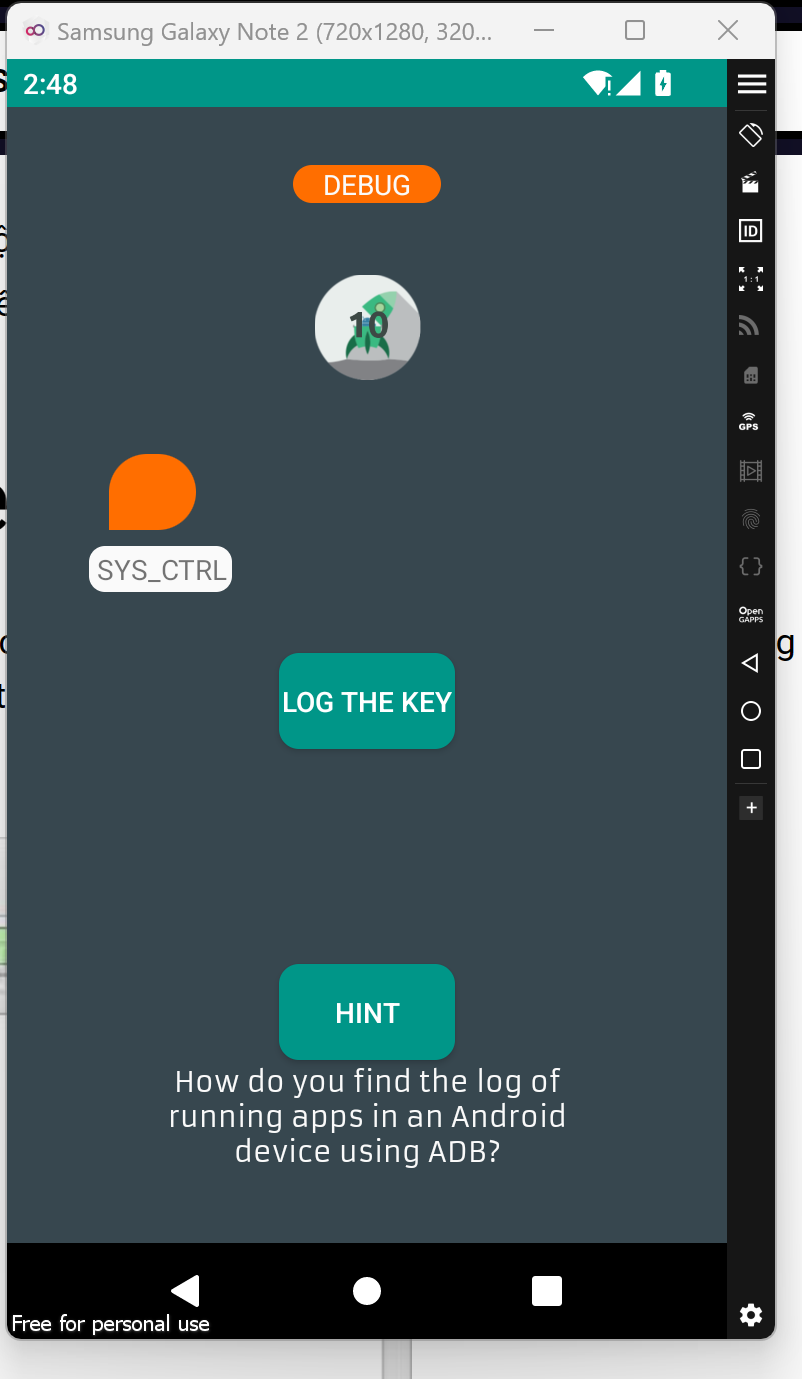
**D.1 EVABS**

A close-up of a computer screen

Description automatically generated

**Level 1:**

Hint ở đây nói rằng liệu chúng ta có thể xem log của những app đang chạy bằng adb tool hay không.

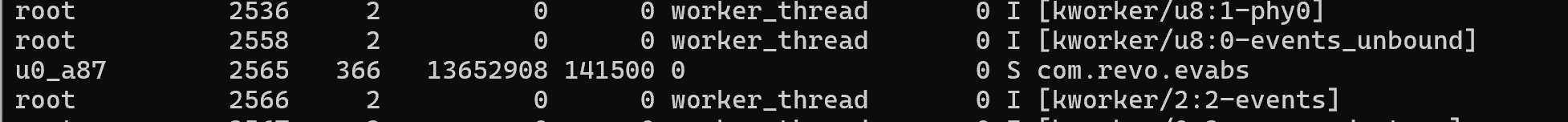


Vậy ta thử list ra các app đang chạy bằng lệnh **adb shell ps**

A screenshot of a computer screen

Description automatically generated

Ta thấy ứng dụng evabs đang được chạy với tên là **com.revo.evabs** và pid là 2565

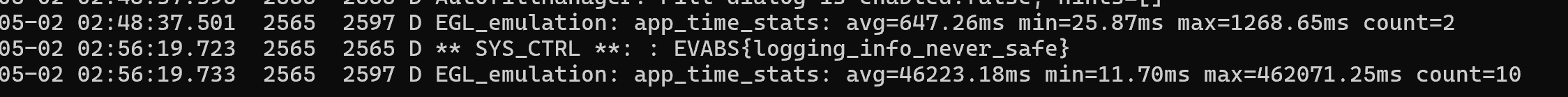


Ta tiến hành đọc log của pid này bằng lệnh **adb logcat --pid=2565**

A screenshot of a computer screen

Description automatically generated

Vẫn chưa thấy flag, ta bấm nút “Log the key” trong app để xem flag xuất hiện trong log



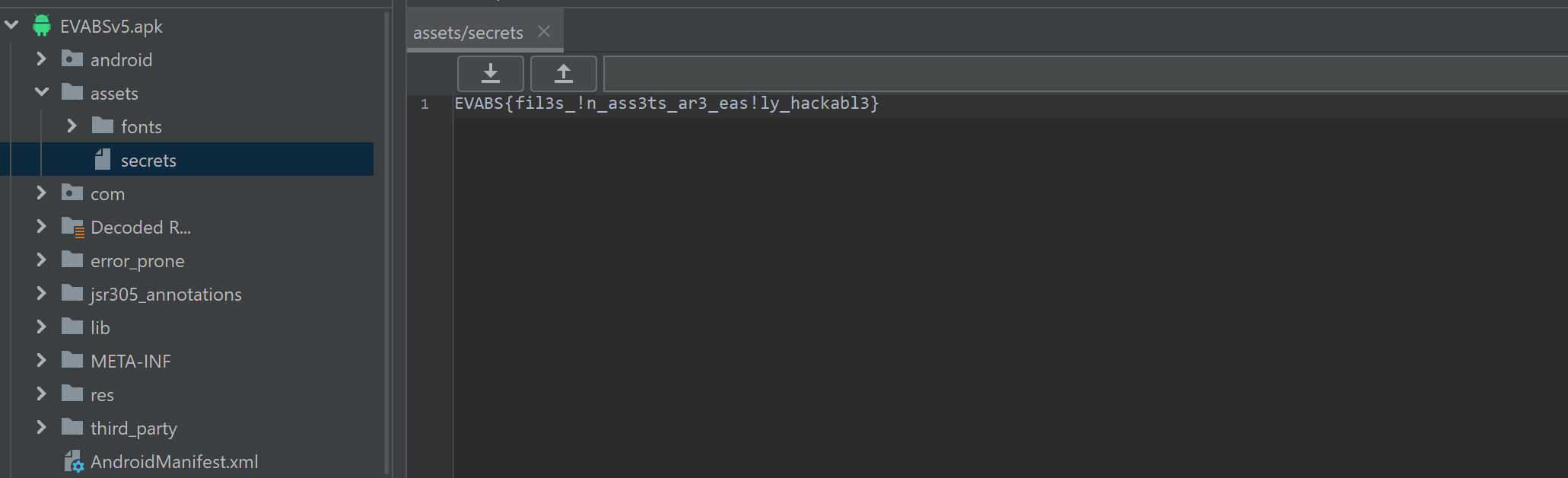
Vậy flag tìm được là: **EVABS{logging\_info\_never\_safe}**

**Level 2:**

**A screenshot of a chat

Description automatically generated**

Hint bảo rằng ta có thể tìm được flag ở file assets trong file apk. Sử dụng bytecode viewer để tiến hành xem file assets.



Flag được tìm thấy: **EVABS{fil3s\_!n\_ass3ts\_ar3\_eas!ly\_hackabl3}**

**Level 3:**

A screenshot of a computer

Description automatically generated

Đầu tiên, ta thử decompile file EVABSv5.apk bằng apktool.

A screenshot of a computer

Description automatically generated

Thư mục được decompile như sau.

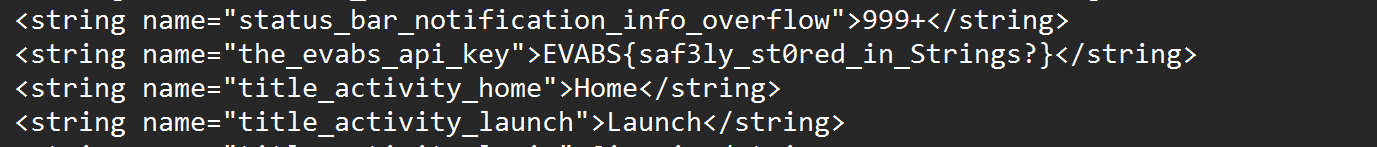
A screenshot of a computer

Description automatically generated

Thông thường, các file .xml sẽ được lưu trong tập tin res. Ta vào đường dẫn res/values/strings.xml để lấy flag.

A screenshot of a computer

Description automatically generated



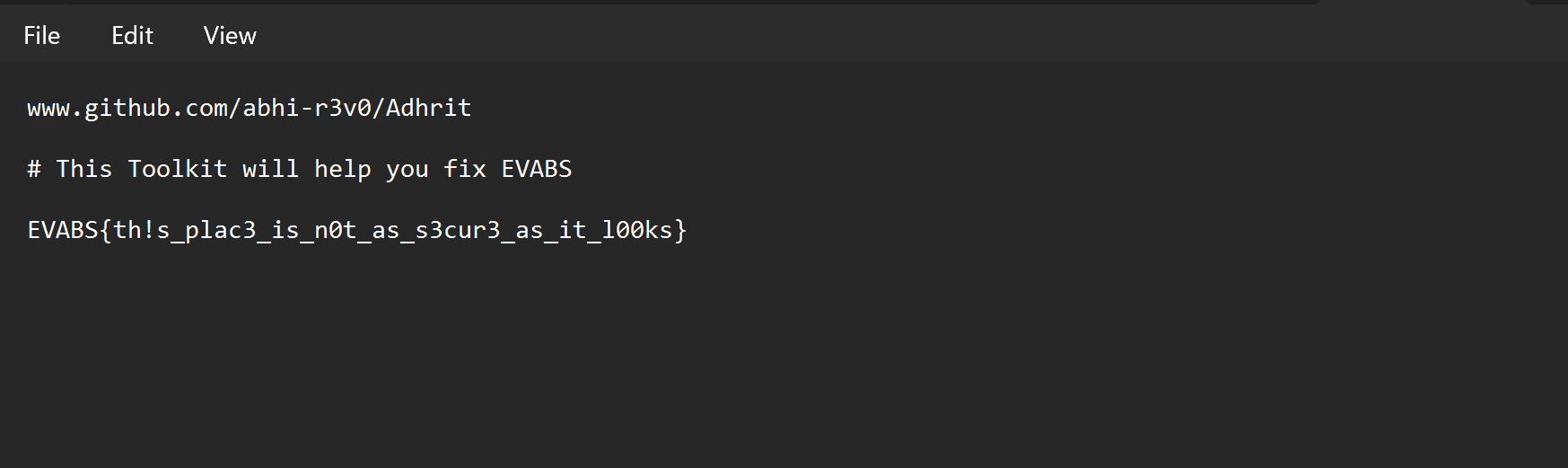
Flag được tìm thấy: **EVABS{saf3ly\_st0red\_in\_Strings?}**

**Level 4:**

A screenshot of a chat

Description automatically generated

Hint của level này nói rằng có thể có flag nằm trong thư mục res, hãy thử tìm nó.



Ta tìm thấy flag ở file res/raw/link.txt

Flag: **EVABS{th!s\_plac3\_is\_n0t\_as\_s3cur3\_as\_it\_l00ks}**

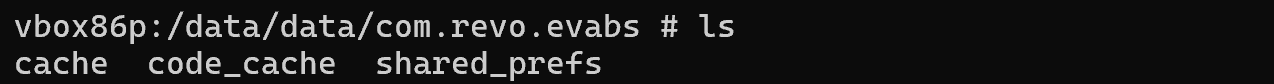
**Level 5:**

Hint ở level này gợi ý flag nằm ở nơi ta lưu cặp key-value trong app.

A screenshot of a computer

Description automatically generated

Thông thường, key-value được lưu ở thư mục data/data/<Tên app>/… Ta thử dùng **adb shell** để thử tìm kiếm.



Có thư mục **shared\_prefs,** truy cập vào và tìm kiếm flag. Tuy nhiên, ta lại không thấy flag xuất hiện. Liệu flag đã bị xóa???

A screen shot of a computer screen

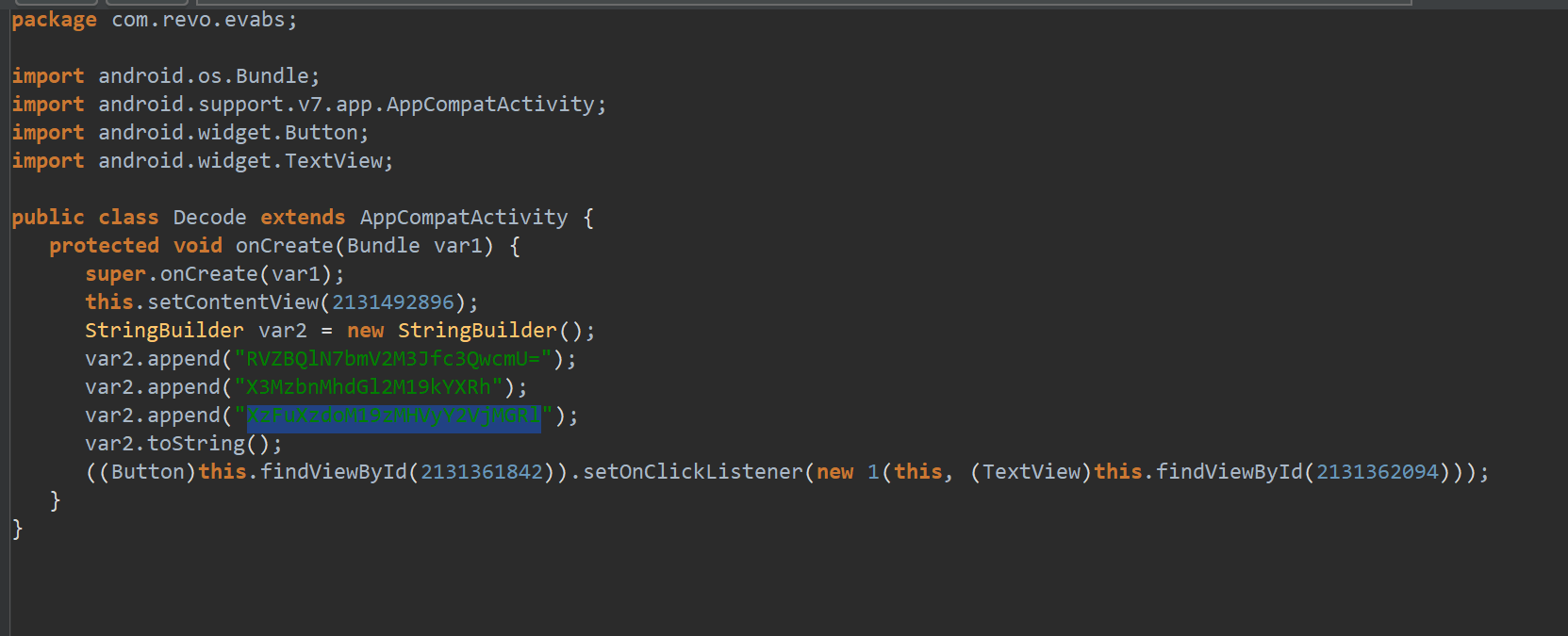
Description automatically generated

**Level 6: DB Leak lỗi không thể truy cập được**

**Level 7: Không thể start ExportedActivity**

**Level 8:**

Ta có thể truy cập vào file **com/revo/evabs/Decode.class** trong Bytecode Viewer để tiến hành đọc và phân tích code.



Như ta có thể thấy 3 chuỗi trên được mã hóa base64. Vậy chỉ cần decode base64 là lấy được flag. Sử dụng đoạn code dưới đây để tiến hành decode.

A computer screen shot of a black screen

Description automatically generated

Kết quả thu được:

A screen shot of a computer

Description automatically generated

Flag tìm được: **EVABS{nev3r\_st0re\_s3ns!tiv3\_data\_1n\_7h3\_s0urcec0de}**

**Level 9:**

Ở level này, khi ta bấm nút Turn on thì hiện ra thông báo “SYS\_CTRL: ACCESS DENIED”.

**A screenshot of a computer

Description automatically generated**

Ta sử dụng Bytecode Viewer để tiến hành phân tích mã nguồn. Ở file **com/revo/evabs/SmalliInject.class,** ta thấy biến SIGNAL đang có giá trị là “LAB\_OFF”

A screen shot of a computer program

Description automatically generated

Tiếp tục qua đến file **com/revo/evabs/SmalliInject$2.class,** ta thấy rằng chỉ khi biến SIGNAL được gán giá trị là “LAB\_ON” thì mới có thể in ra flag.

A screenshot of a computer program

Description automatically generated

Vậy ta có thể sử dụng kĩ thuật patching để sửa đổi file apk. Truy cập vào file SmalliInject.smalli đã được decompile ở những level trước.

A screen shot of a computer program

Description automatically generated

Ta cần đổi giá trị LAB\_OFF thành LAB\_ON

A screen shot of a computer code

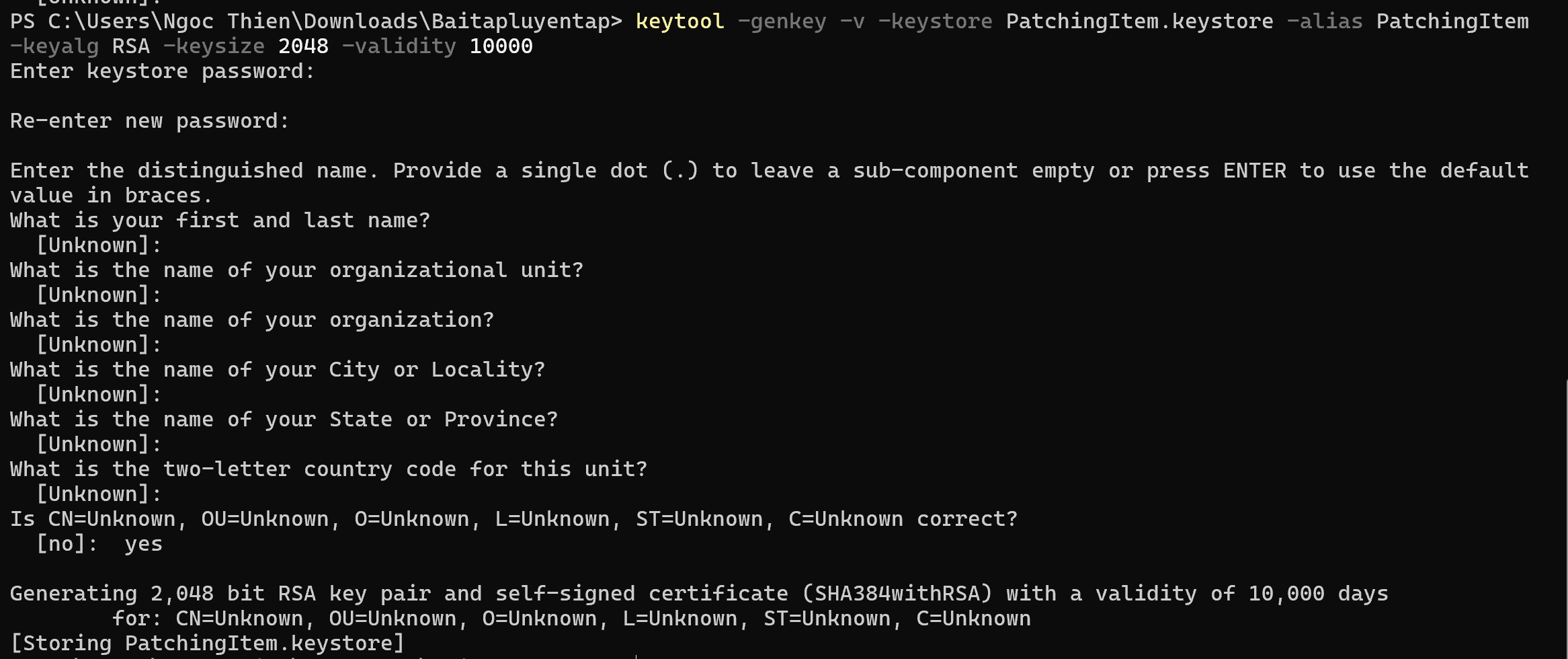
Description automatically generated

Sau đó, ta tiến hành build lại file đã patching

A screen shot of a computer

Description automatically generated

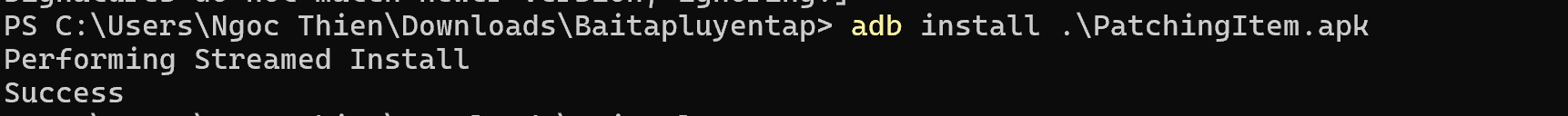
Sau đó là tạo keystore và kí vào file apk.



A white text on a black background

Description automatically generated

Sau đó ta cài đặt lại file apk lên emulator và lấy flag



A screenshot of a computer

Description automatically generated

Flag ta thu được: **EVABS{smali\_inj\_is\_l3thalr}**

**Level 10: Khi bấm nút “Receive” BurpSuite không thể intercept gói tin mặc dù khi truy cập vào trang web trên emulator thì vẫn bắt bình thường :((((**

**Level 11:**

Ở level này nó bắt chúng ta nhập 1 chuỗi gì đấy để có thể exploit.

A screenshot of a computer

Description automatically generated

Ta đọc thử code để xem nó hoạt động như thế nào

A computer screen shot of a code

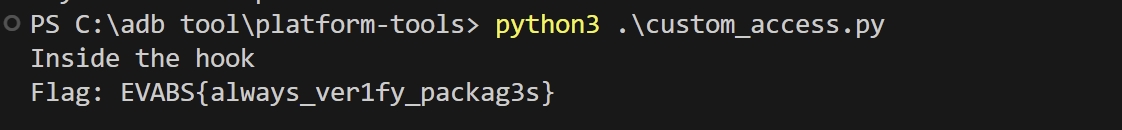
Description automatically generated

Chúng ta thấy sau khi nhập chuỗi “cust0m\_p3rm”, flag sẽ được truyền đến **com.revo.evabs.action.SENSOR\_KEY**. Hmm, có lẽ ta có thể tạo hookscript cho hàm **putExtra()** để hiển thị luôn flag mà không cần phải truyền đi đâu cả. Ta tạo như sau

A screenshot of a computer program

Description automatically generated

Tiến hành chạy script trên và thu được flag



Flag ta tìm được: **EVABS{always\_ver1fy\_packag3s}**

**Challenge 12:**

Ở level này, sau khi ta nhấn nút “Map Area”, ta thấy nó cung cấp 2 tọa độ x, y và tích của 2 tọa độ.

A screenshot of a computer

Description automatically generated

Sử dụng Bytecode Viewer để xem mã nguồn của nó

A computer screen shot of a program code

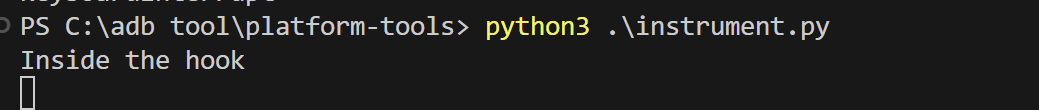
Description automatically generated

Ta thấy 2 biến a và b được gán là 25 và 2, chứng tỏ giá trị x luôn là 50. 1 biến var2 nằm trong khoảng từ 0 đến 69 bằng cách sử dụng hàm nextInt() của đối tượng lớp Random. Sau đó nó so sánh biến x với tổng của var2 và 150, nếu lớn hơn thì flag được tạo ra trong log. Ý tưởng ở đây là sửa đổi hàm nextInt() của lớp com.ulti.Random luôn trả về một số bé hơn 100 để có thể thỏa điều kiện tạo ra flag. Hook script được tạo như sau

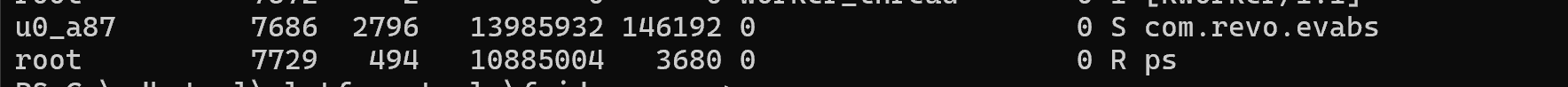
A screen shot of a computer program

Description automatically generated

Ta trả về cho hàm nextInt() là -101. Ta chạy script



Sau đó dùng lệnh **adb shell ps** để xem **com.evo.evabs** đang có pid là bao nhiêu



Tiếp tục dùng lệnh **adb logcat –pid=7686** để xem log của chương trình.

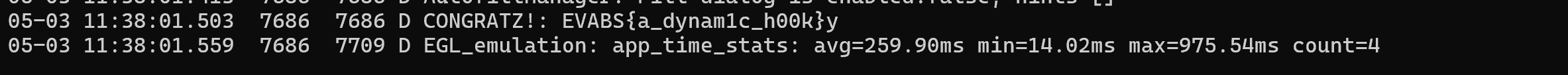
A screen shot of a computer

Description automatically generated

Cuối cùng ta bấm nút “Map Area” và xem flag xuất hiện ở log.

A screenshot of a video game

Description automatically generated



Flag tìm thấy: **EVABS{a\_dynam1c\_h00k}**

---

***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**