

**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% |

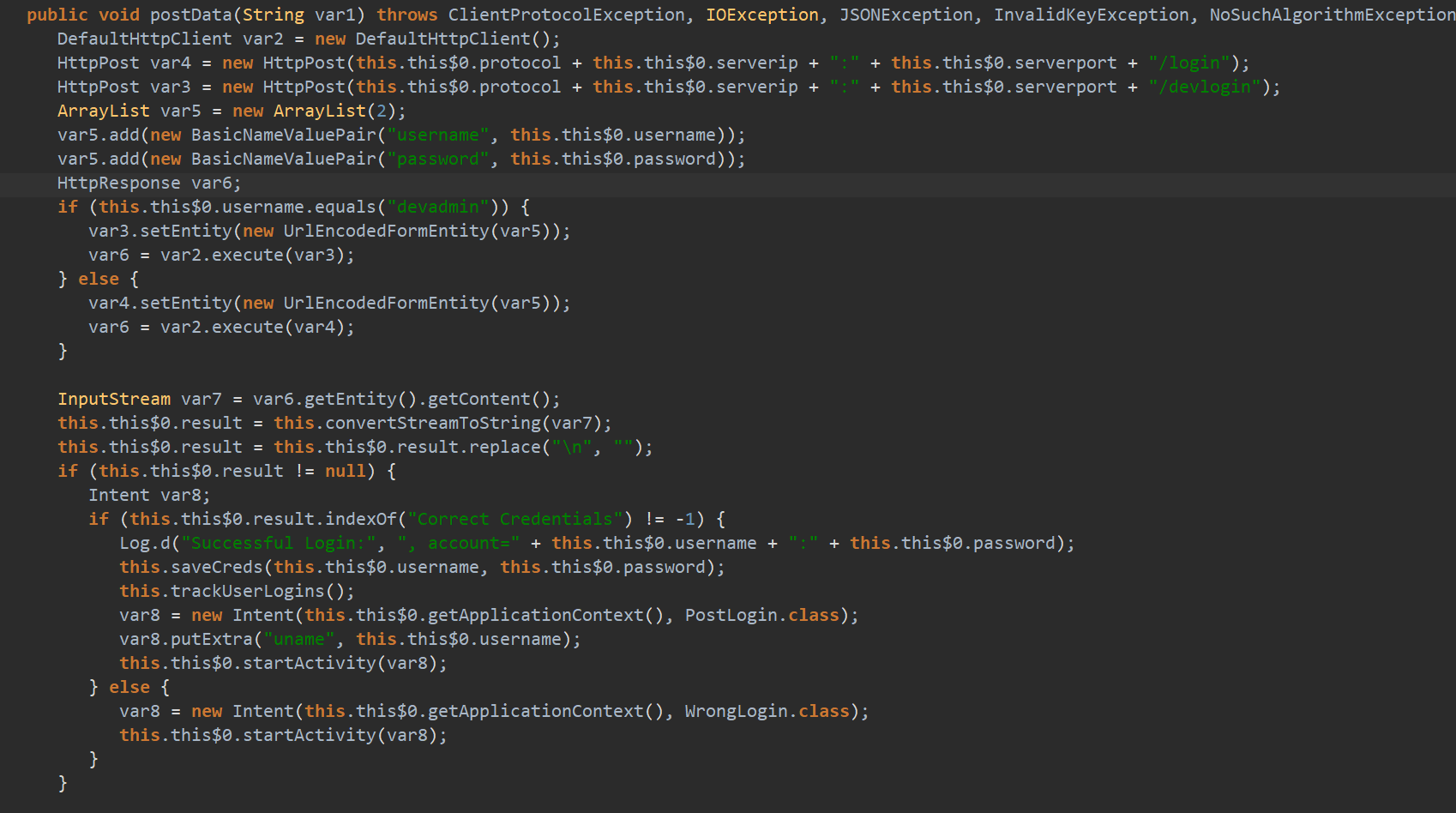
**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

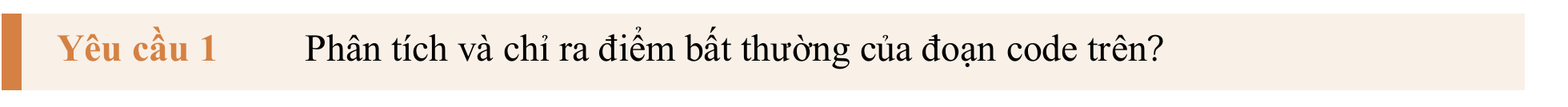
1 Ghi nội dung công việc, các kịch bản trong bài Thực hành

**2**

# BÁO CÁO CHI TIẾT

Sau khi tiến hành phân tích file InsecureBankv2.apk bằng ByteCode Viewer và scan với plugin “Malicious Code Scanner”, ta nhận thấy đoạn code bất thường ở file “**com/android/insecurebankv2/DoLogin$RequestTask.class”**





Điểm bất thường đầu tiên ta tìm được là đoạn code trên sử dụng giao thức Http, điều đó khiến cho các dữ liệu trao đổi giữa các bên có thể bị attacker chiếm đoạt bất cứ lúc nào nhằm mục đích sửa chữa và khai thác. Hơn nữa, cả trường password và username đều không được mã hóa mà truyền thẳng vào đối tượng HttpPost (**var3** và **var4**), sau đó gửi HttpPost. Việc không mã hóa này gây ra lỗi bảo mật nghiêm trọng, attacker có thể đứng giữa, bắt gói tin và chỉnh sửa gói tin, khiến cho các dữ liệu được lưu ở server có thể bị leak ra một cách dễ dàng.



Sau khi gõ lệnh adb shell để vào được máy ảo, ta tiến hành vào đường dẫn **“/data/data/com.android.insecurebankv2/databases”** và truy cập vào database **“mydb”**

A screen shot of a black screen

Description automatically generated

Sau đó ta thực hiện truy xuất tất cả các bảng trong mydb bằng lệnh **SELECT name FROM sqlite\_master WHERE type='table';.** Sau đó truy xuất tất cả dữ liệu của tất cả các bảng.

A screen shot of a computer

Description automatically generated

Ta thấy thông tin về names đã bị lộ, điều đó chứng tỏ dữ liệu lưu trữ ở đây không an toàn khi người ngoài có thể truy xuất 1 cách dễ dàng.

A close up of a text

Description automatically generated

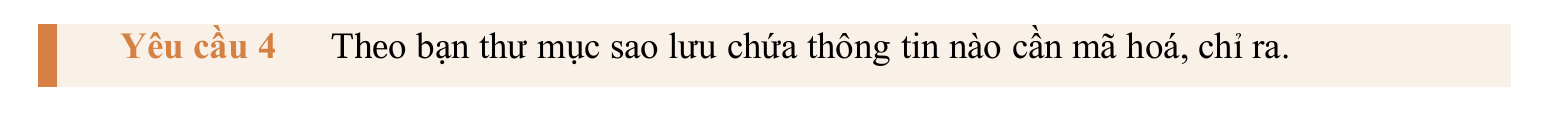
Sau khi thực hiện tìm kiếm các từ khóa nhạy cảm thì ta không phát hiện được bát kì thông tin nào hữu ích.

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated



Sau khi giải nén ta thu được các tập tin như sau:

A screenshot of a computer program

Description automatically generated

Các tập tin chúng ta cần mã hóa là:

**- apps/com.android.insecurebankv2/sp.** Tập tin bao gồm 2 file quan trọng sau:

A screen shot of a computer

Description automatically generated

- **apps/com.android.insecurebankv2/db**

A screenshot of a computer

Description automatically generated



Như ta có thể thấy, 2 chuỗi cần giải mã đang được mã hóa ở định dạng base64, ta thử decode base64 để xem có ra được chuỗi cần tìm hay không.

A screenshot of a computer screen

Description automatically generated

Ta sử dụng đoạn code này để decode

A screen shot of a computer program

Description automatically generated

Kết quả thu được như sau:

A black background with white text

Description automatically generated

Ta đã tìm được username, tuy nhiên password vẫn chưa tìm được. Ta thử giải mã password bằng AES256. Update đoạn code như sau:

A screen shot of a computer program

Description automatically generated

Kết quả thu được:

A computer screen shot of a computer code

Description automatically generated

A close up of a text

Description automatically generated

Sử dụng Bytecode Viewer để tiến hành xem đoạn code ở file **/com/android/insecurebankv2/PostLogin.class.**  Ở file này, ta thấy được hàm **showRootStatus()** hiển thị “RootDevice” hoặc “Device not Rooted!!”

A screen shot of a computer program

Description automatically generated

Theo yêu cầu đề bài, ta cần khiến cho điện thoại luôn ở trạng thái “Root Device!!”, vậy ta phải thỏa điều kiện của 2 hàm **doesSuperuserApkExist()** và **doesSUexist().**  Cùng xem code của 2 hàm này.

A screen shot of a computer code

Description automatically generated

Ta thấy hàm **doesSUexist()** không thể compile thành java được, tuy nhiên ta có thể xem code của hàm này dưới dạng ngôn ngữ Smalli. Còn hàm **doesSuperuserApkExist()** thì kiểm tra xem có tồn tại file trong đường dẫn **“/system/app/Superuser.apk”** không. Ta connect vào shell và thử tìm file này.



Ta thấy file Superuser.apk thực chất chứa trong đường dẫn **“system/app/Superuser/Superuser.apk”**, vậy hàm **doesSuperuserApkExist()** đang kiểm tra sai đường dẫn, ta có thể sửa lại đường dẫn này trong tập tin Smalli đã được decompile trước đó.

A screen shot of a computer

Description automatically generated

Ta tiếp tục xem hàm doesSUexist() xem nó hoạt động như thế nào

A computer screen shot of a black screen

Description automatically generated

Ở hàm này hệ thống sẽ thực thi lệnh **/system/xbin/which su** để tìm kiếm file nhị phân su, nếu có thì trả về đường dẫn của nó, nếu tồn tại đường dẫn thì trả về true, còn không thì trả về false. Vậy ở hàm này ta không cần phải chỉnh sửa thêm gì cả.

Sau khi đã chỉnh sửa, ta tiến hành build lại file .apk và kí lên nó.

A screenshot of a computer

Description automatically generated

Cài đặt lên máy ảo và xem kết quả:

A screenshot of a phone

Description automatically generated



A computer screen shot of a code

Description automatically generated

Ta nhận thấy ta có thể ghi đè hàm **doesSuperuserApkExist()** dựa vào đoạn code trên, vậy phần code ta cần hoàn thành phải luôn luôn đưa hàm này trả về giá trị true. Vậy ta thử ghi đè hàm bằng dòng “return true” xem điều gì xảy ra.

A screen shot of a computer program

Description automatically generated

Kết quả ta thu được:

A black background with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

---

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

## thường gặp Nhóm 07

**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**