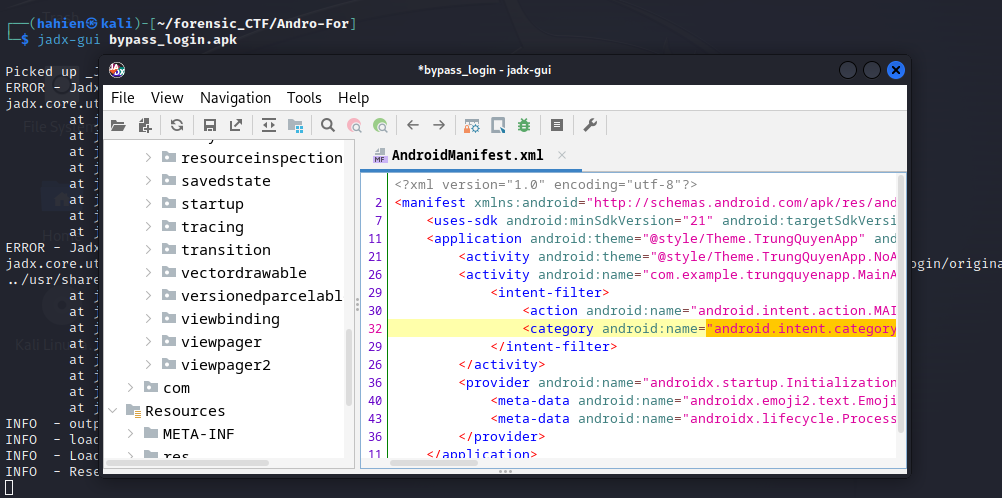
A screenshot of a computer

Description automatically generated



<?xml *version*="1.0" *encoding*="utf-8"?>

<manifest *xmlns:android*="http://schemas.android.com/apk/res/android" *android:versionCode*="1" *android:versionName*="1.0" *android:compileSdkVersion*="32" *android:compileSdkVersionCodename*="12" *package*="com.example.trungquyenapp" *platformBuildVersionCode*="32" *platformBuildVersionName*="12">

    <uses-sdk *android:minSdkVersion*="21" *android:targetSdkVersion*="32"/>

    <application *android:theme*="@style/Theme.TrungQuyenApp" *android:label*="@string/app\_name" *android:icon*="@mipmap/ic\_launcher" *android:debuggable*="true" *android:testOnly*="true" *android:allowBackup*="true" *android:supportsRtl*="true" *android:roundIcon*="@mipmap/ic\_launcher\_round" *android:appComponentFactory*="androidx.core.app.CoreComponentFactory">

        <activity *android:theme*="@style/Theme.TrungQuyenApp.NoActionBar" *android:label*="@string/title\_activity\_user" *android:name*="com.example.trungquyenapp.User" *android:exported*="false"/>

        <activity *android:name*="com.example.trungquyenapp.MainActivity" *android:exported*="true">

            <intent-filter>

                <action *android:name*="android.intent.action.MAIN"/>

                <category *android:name*="android.intent.category.LAUNCHER"/>

            </intent-filter>

        </activity>

        <provider *android:name*="androidx.startup.InitializationProvider" *android:exported*="false" *android:authorities*="com.example.trungquyenapp.androidx-startup">

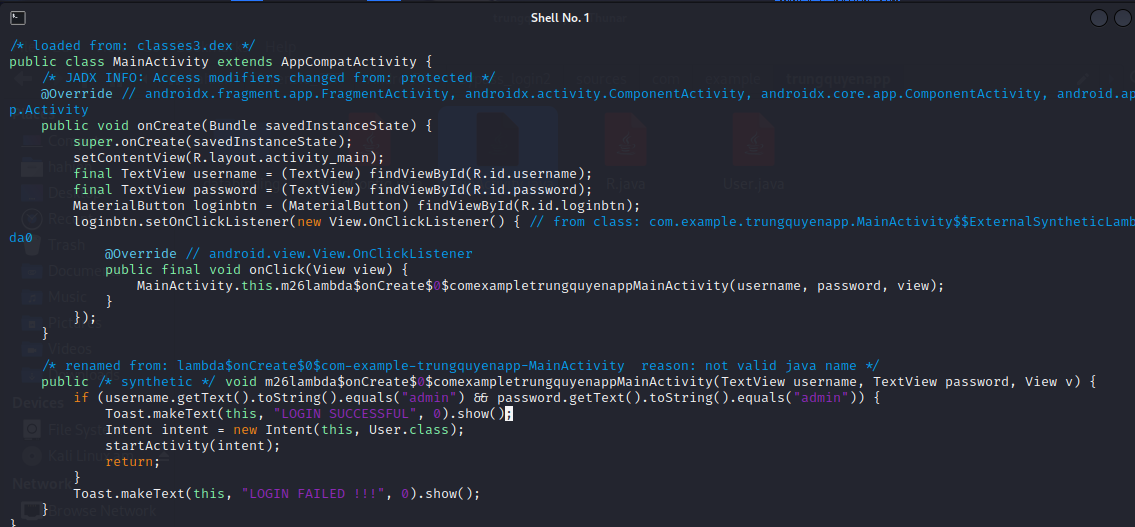
            <meta-data *android:name*="androidx.emoji2.text.EmojiCompatInitializer" *android:value*="androidx.startup"/>

            <meta-data *android:name*="androidx.lifecycle.ProcessLifecycleInitializer" *android:value*="androidx.startup"/>

        </provider>

    </application>

</manifest>



Username: admin

Password: admin

Lỗ hổng bảo mật lớn: thông tin đăng nhập được kiểm tra trực tiếp trong mã nguồn và không có bất kỳ biện pháp bảo vệ nào khác. Điều này có nghĩa là bất kỳ ai cũng có thể dễ dàng đọc mã nguồn và biết được tên người dùng và mật khẩu đúng là "admin".

Để khắc phục vấn đề này, bạn nên sử dụng các biện pháp bảo mật như mã hóa mật khẩu, xác thực máy chủ, hoặc sử dụng các dịch vụ quản lý danh sách người dùng như Firebase Authentication. Điều này sẽ làm cho ứng dụng của bạn an toàn hơn và tránh được các cuộc tấn công thông tin đăng nhập.

A screenshot of a computer screen

Description automatically generated

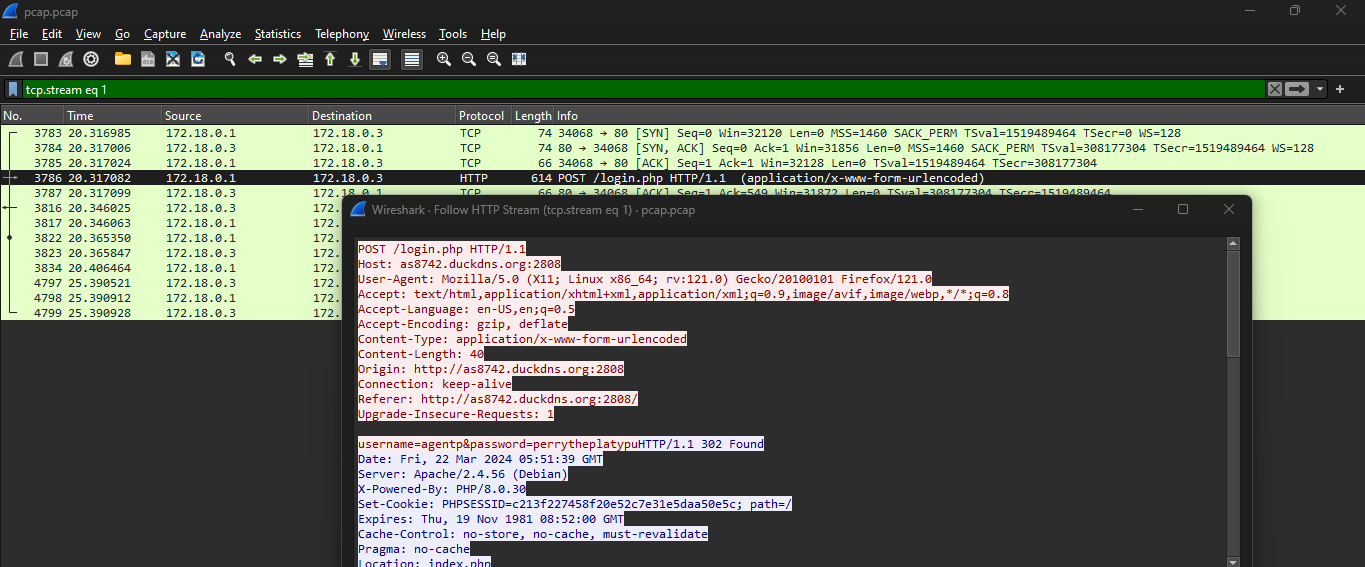
1. **Tìm IP của Web Server:**
   * Mở tập tin pcap.pcap bằng Wireshark.
   * Sử dụng tính năng tìm kiếm trong Wireshark để tìm các gói tin HTTP hoặc HTTPS.

A screenshot of a computer

Description automatically generated

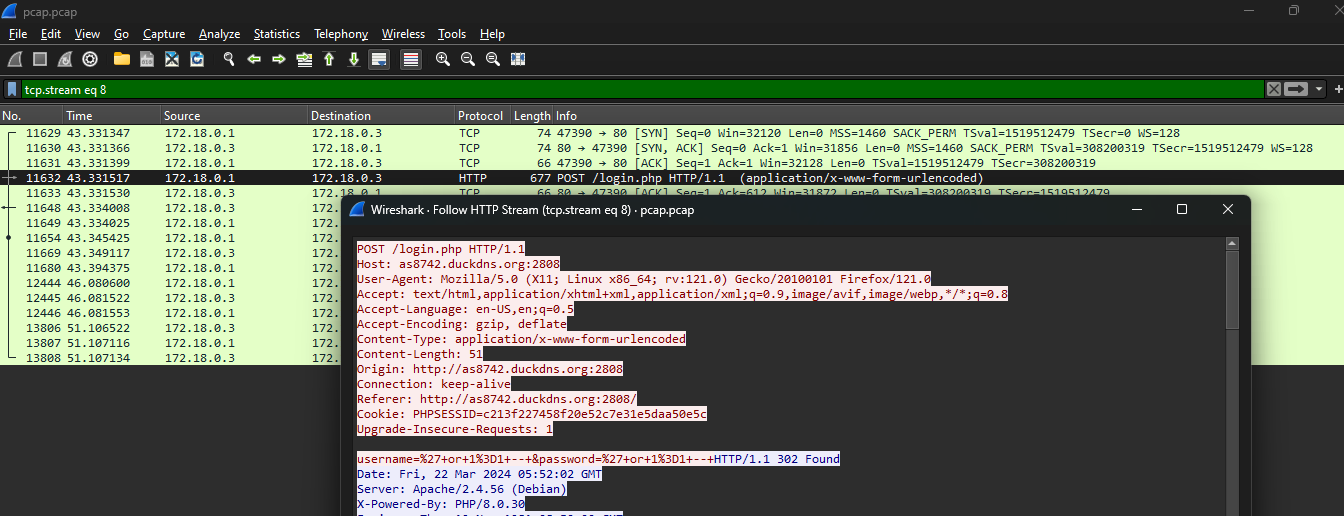
* + Xem địa chỉ IP đích của các gói tin này để xác định IP của web server: 151.101.78.132.

1. **Tìm Username và Password của một tài khoản sử dụng server:**
   * Tiếp tục sử dụng Wireshark.
   * Lọc các gói tin có chứa thông tin đăng nhập, chẳng hạn như giao thức HTTP POST hoặc FTP.



* + Xem nội dung của các gói tin để tìm thông tin đăng nhập.
* Username: agent
* Password: perrytheplatypu

1. **Xác định Hacker tấn công từ bên trong mạng hay bên ngoài:**
   * Kiểm tra địa chỉ IP nguồn của các gói tin tấn công.
   * So sánh địa chỉ IP với dải IP nội bộ để xác định liệu tấn công có từ bên trong mạng hay không.



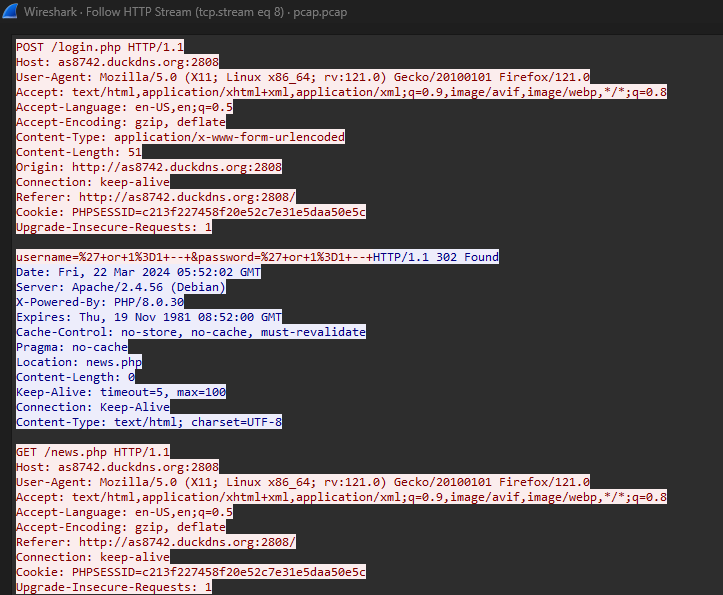
Dựa vào gói tin bạn đã phát hiện, có vẻ như có một cuộc tấn công SQL Injection đang diễn ra. Gói tin này có vẻ được sử dụng để kiểm tra tính hợp lệ của thông tin đăng nhập.

* Source IP là 172.18.0.1 và Destination IP là 172.18.0.3.
* Username được gửi đi là "%27+or+1%3D1+--+".
* Password được gửi đi là "%27+or+1%3D1+--+".

Trong cuộc tấn công SQL Injection, "%27+or+1%3D1+--+" là một phần của câu truy vấn SQL, trong đó "%27" là mã hóa của dấu nháy đơn (') và "1=1" là một điều kiện luôn đúng trong SQL.

Câu trả lời cho câu hỏi số 3: Hacker đang thực hiện cuộc tấn công từ bên trong mạng (source IP: 172.18.0.1) vào web server (destination IP: 172.18.0.3) bằng cách sử dụng kỹ thuật SQL Injection để thử đánh lừa hệ thống xác thực.

1. **Lỗ hổng được khai thác:**SQL injection
2. **Hacker đã login vào tài khoản nào: admin**

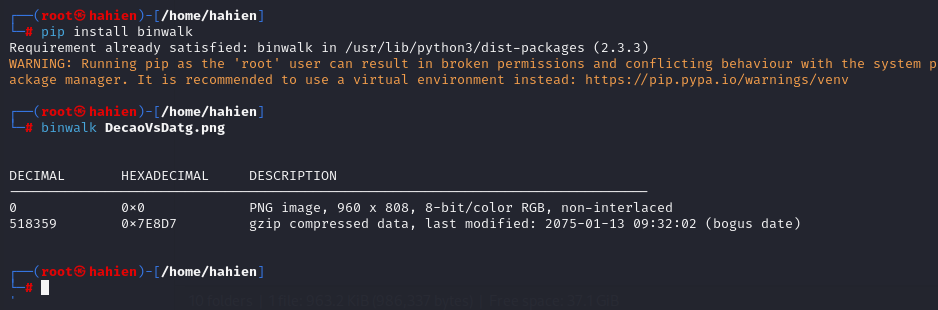


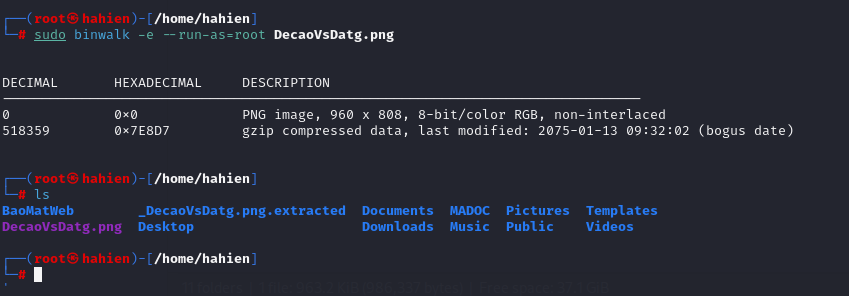
A screenshot of a computer

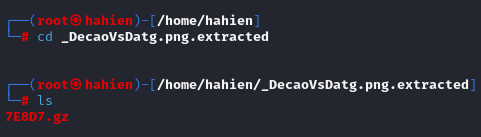
Description automatically generated

1. **Server mà hacker dùng để test là gì:**
   * Tìm các yêu cầu từ hacker đến một địa chỉ IP cụ thể hoặc các dịch vụ đặc biệt.
2. **Hacker đã lấy được mật khẩu của admin chưa:**
   * Tìm các gói tin chứa thông tin đăng nhập của admin.
   * Xem nội dung của các gói tin để kiểm tra liệu hacker đã lấy được mật khẩu hay không.
3. **Có nên tình nghi đặc vụ đó là người đã thực hiện cuộc tấn công không:**
   * Phân tích mối liên hệ giữa đặc vụ và các hoạt động đáng ngờ trong các gói tin.
   * Xem xét địa chỉ IP, hành vi truy cập và thời gian để xác định liệu đặc vụ có liên quan đến cuộc tấn công hay không.

* Đầu tiên, em thử dùng stegOnline, nhưng không thu được kết quả gì.
* Tiếp đến em dùng binwark để xem thông tin ảnh có chứa tệp ẩn không.







A screen shot of a computer

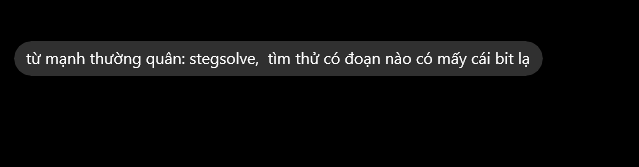
Description automatically generated

* Không giải nén được, nên em tìm cách khác thui.
* Hết cách, em hi vọng vào cái ảnh ban đầu xem kỹ có gì không.

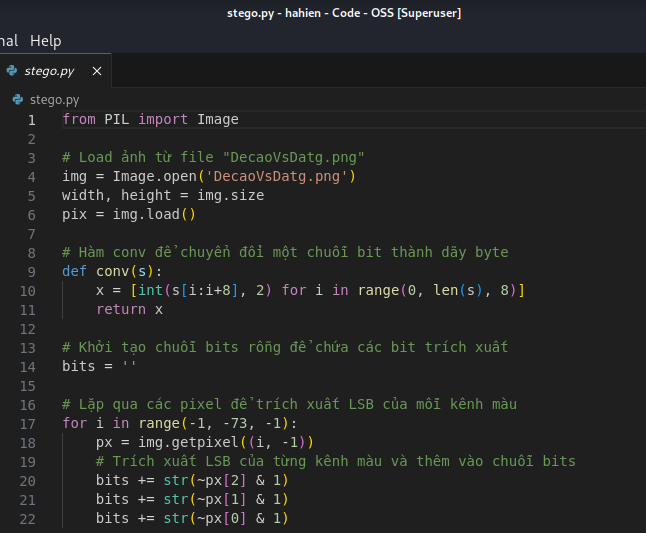
Two men in black and white

Description automatically generated

* TROLL Cả SHOWBIZ -> em nghĩ nó liên quan đến các bits cùng với hint đã tìm kiếm được.

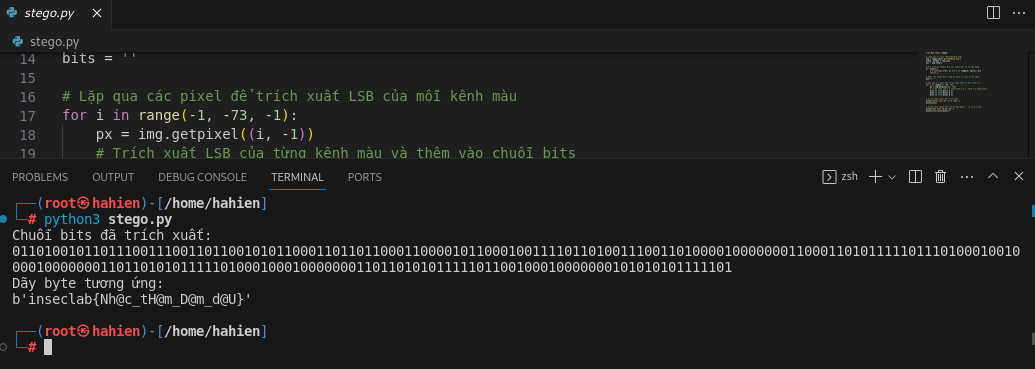


* Từ đó, có đoạn code sau:

 A screen shot of a computer code

Description automatically generated

* Và kết quả là:



* FLAG: inseclab{Nh@c\_tH@m\_D@m\_d@U}