모바일 앱

취약점 분석

기초

Name : 조민혁

Class Number : 5

Phone Number : 010-4923-2198

목 차

1. SolidApp.apk

1-1. 풀이 과정

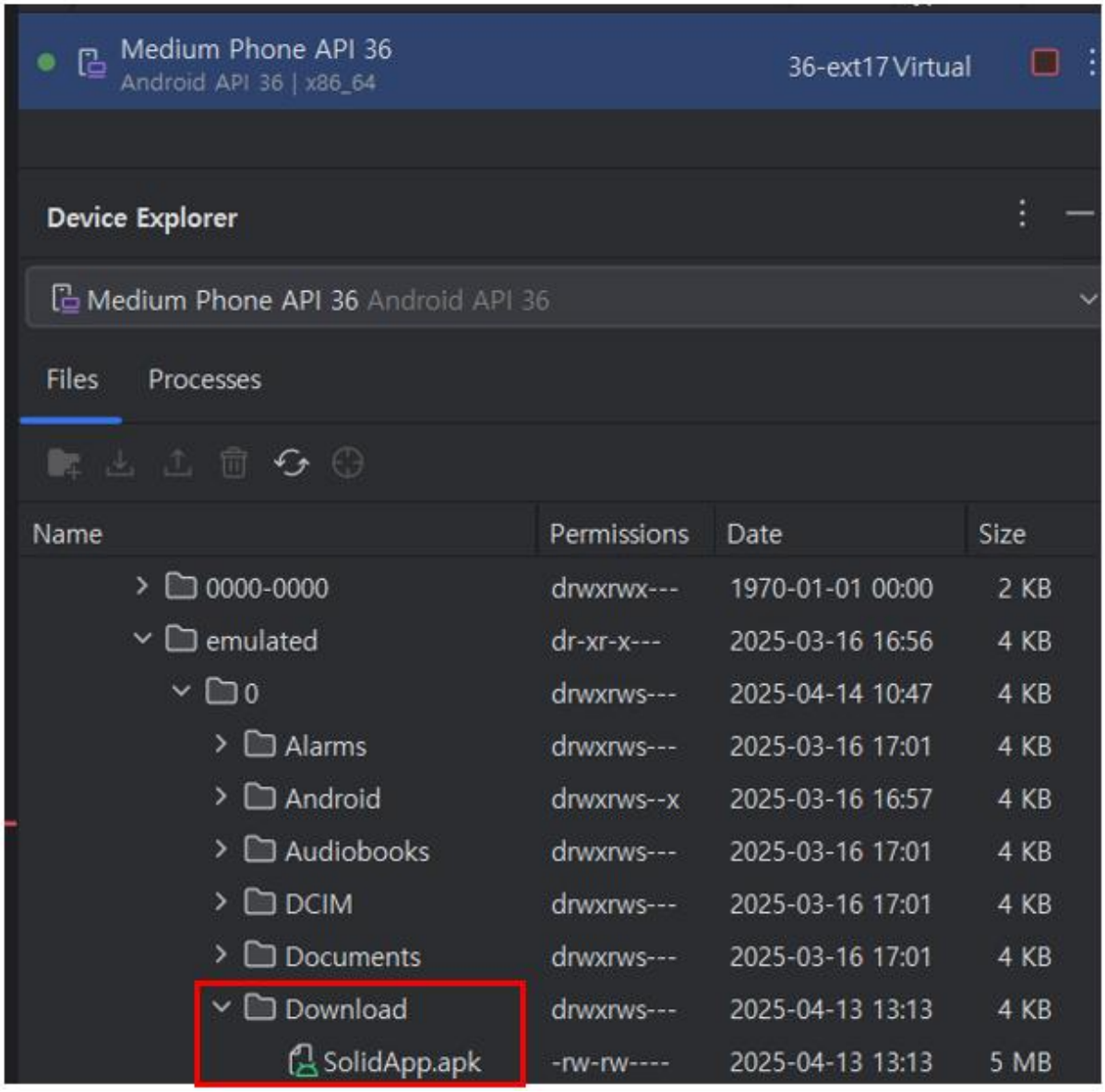

그림 1. Emulator에 SolidApp.apk 업로드

SolidApp.apk 을 파악하기 위해 Android Studio 에서 그림 1 과 같이 Emulator 에 SolidApp.apk 를 업로드하였다.

그림 2. SolidApp.apk 실행 화면

이후, 그림 2 와 같이 Emulator 를 작동하여 SolidApp.apk 를 실행하였다. 실행 시 Username 과 Password 를 입력하는 것과 로그인에 실패를 하게 된다면 앱이 종료되며 이외에는 알 수 있는 정보가 없었다. 이에 따라 SolidApp.apk 의 소스 코드를 분석하는 과정이 필요했다.

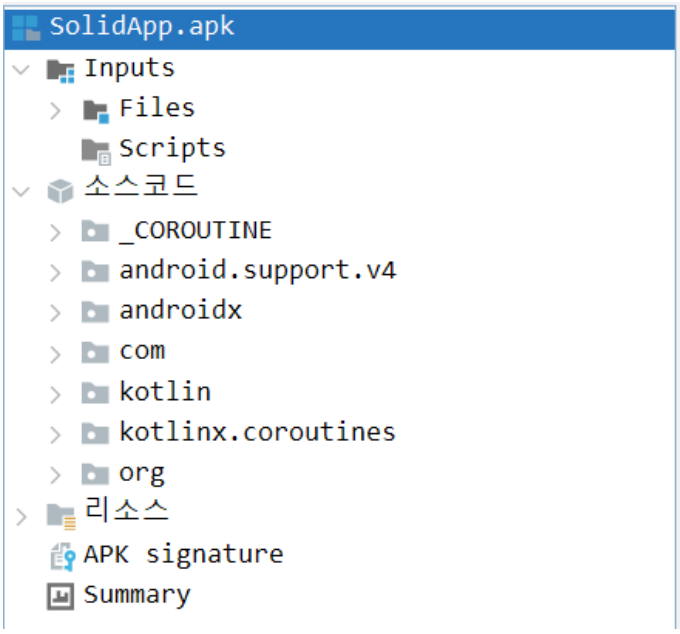

그림 3. jadx-gui를 통한 SolidApp.apk 패키지 화면

'apktool -d SolidApp.apk -o SolidApp' 명령을 실행 후에 jadx-gui 를 통해 그림 3 과 같은 화면을 확인할 수 있었다.

```
public final class AuthLogic {
    private final String decodedUsername = decodeBase64("cmVkdGVhbQ==");
    private final String staticString = "gttcurct";
    private final int key = 6;

    private final String decodeBase64(String encoded) {
        byte[] decodedBytes = Base64.decode(encoded, 0);
        Intrinsics.checkNotNullExpressionValue(decodedBytes, "decodedBytes");
        return new String(decodedBytes, Charsets.UTF_8);
    }

    private final String xorString(String input, int key) {
        StringBuilder sb = new StringBuilder();
        int length = input.length();
        for (int i = 0; i < length; i++) {
            sb.append((char) (input.charAt(i) ^ key));
        }
        String sb2 = sb.toString();
        Intrinsics.checkNotNullExpressionValue(sb2, "xorResult.toString()");
        return sb2;
    }
```

그림 4. AuthLogic.class 코드

먼저, 로그인 화면이 존재했기에 로그인 로직을 어떻게 처리하는 지 확인하는 과정이 필요했다. 그림 4 에 나타나있듯 해당 코드에서 "cmVkdGVhbQ=="를 Base64 로 Decode 한 것이 decodedUsername 이란 것을 알 수 있었다. 또한 xorString 메소드에서 key 와 input 을 택한 값이 StringBuilder 에 담기는 것을 통해 비밀번호라는 의심을 할 수 있었다. 이에 따라 xorString 메소드를 어디서 호출하는 지 확인이 필요했다.

```
public final boolean isPasswordValid(String inputPassword) {
    Intrinsics.checkNotNullParameter(inputPassword, "inputPassword");
    return Intrinsics.areEqual(inputPassword, xorString(this.staticString, this.key));
}
```

그림 5. isPasswordValid 코드

그림 5에 나타나있듯 staticString과 key를 매개변수로 하여 xorString 메소드를 호출하는 것을 통해 staticString과 key 값을 xorString 메소드를 처리한 결과가 패스워드란 것을 알 수 있었다. 이에 따라 해당 코드와 동일 c언어 스크립트를 작성하여 실행한 결과 그림 6과 같은 결과를 알 수 있었다.

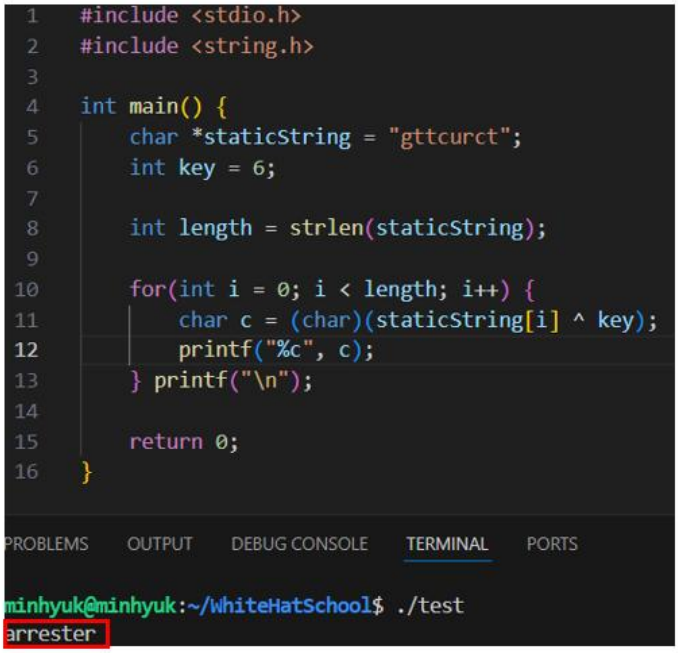

그림 6. c언어로 작성한 xorString 메소드

그림 6과 같이 해당 메소드를 실행하면 arrester가 출력값으로 나오는 것을 알 수 있었고 이것이 비밀번호라는 것을 확인할 수 있었다.

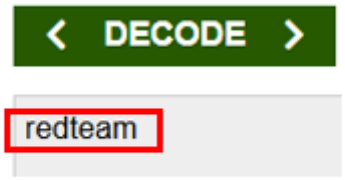

그림 7. Base64 디코딩 결과

아이디의 경우 앞서 언급한 decodedUsername을 확인하면 그림 7과 같이 redteam으로 나오는 것을 알 수 있다.

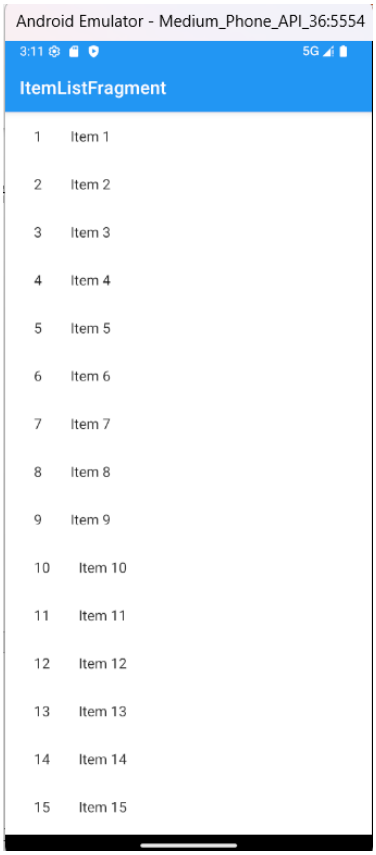

그림 8. 로그인 성공 화면

redteam을 아이디로 arrester를 비밀번호로 로그인 폼에 입력하면 그림 8과 같이 정상적으로 로그인에 성공하는 화면을 확인할 수 있다.

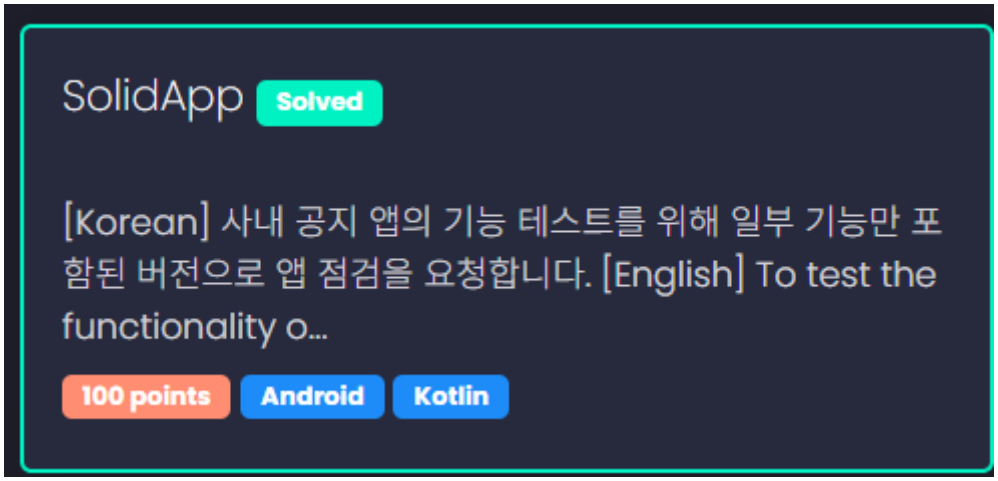

그림 9. SolidApp.apk 문제 풀이 성공 화면

FLAG 형식이 flag{username_password} 였기에 이를 mobilehacking.kr에 입력하면 그림 9와 같은 문제 풀이 성공 화면을 확인할 수 있다.