MWS Cup2013
事前課題 3「Android アプリ脆弱性解析」解答例

1. 出題の意図

2012年以降 Android アプリの脆弱性報告が急激に増加しており，Androidアプリ開発におけるセキュリティへの関心が高まりを見せている．また，一般社団法人 日本スマートフォンセキュリティ協会(以下,JSSEC)から Android アプリのセキュア設計・セキュアコーディングガイド [1]が公開されるなど，安全な設計・実装に関する情報が提供され、多くの開発者に参照されるようになってきた．本課題を通じて、ガイドの内容が具体的な脆弱性の発見および対策にどの程度効果を持つか評価し，JSSECにフィードバックを行う．なお課題で用意した Android アプリは，IPAに報告された実際の Android アプリ脆弱性の内容を意図的に含ませたものであり、脆弱性対策の実践的な演習としての効果を期待したものである．

2. 出題内容

サンプル Android アプリ「Secret Memo」(MWS-Sample) は,以下のようなアプリである。
・ユーザーが入力したメモ(文字)を暗号化して保存・読み込むアプリ
・以下の 2 つの Activity から構成されている
1. MainActivity
- 他アプリ等から起動され,メモ入力を制御する.ユーザーが入力したメモを保存するため、PasswordActivity を呼び出す.
2. PasswordActivity
- パスワード入力画面を提供し,入力されたパスワードを呼び出し元に返す.本アプリ内でのみ利用することを意図している.
このアプリ関する以下の問いに答えよ.

[設問1] MWS-Sample のソースコードを調査し、このアプリに含まれる実装不備(脆弱性)をついた攻撃を受けた場合、どのような被害が発生し得るか、説明しなさい。(2 点)
本課題での実装不備とは、JSSEC セキュアコーディングガイド[1]に記載されるルールから逸脱している項目とする。調査には脆弱性検査サービスの Secure Coding Checker[2]を利用して良い。

[設問2] 設問 1 で説明した被害の原因となったアプリの実装不備（脆弱性）を説明しなさい。ファイルおよびコード行数を含めた上で説明すること。なお、実装不備は 2 か所ある。（各 2 点）

[設問3] 設問 2 で説明した実装不備（脆弱性）を悪用して設問 1 で説明した被害を発生させる攻撃プログラムを作成し、攻撃方法を説明しなさい。実装不備（脆弱性）を攻撃するソースコード断片を示して説明すること。なお、実装不備（脆弱性）が 2 つあるので、攻撃方法も 2 つ説明すること。また作成した攻撃プログラムの提出は不要である。（各 1 点）

[設問4] 設問 2 で見つけた実装不備（脆弱性）を修正した MWS-Sample プログラムを作成し、修正方法を説明しなさい。ソースコードのどの部分をどのように修正するかソースコード断片を示して説明すること。なお、実装不備（脆弱性）が 2 つあるので、修正方法も 2 つ説明すること。
脆弱性が修正されたことの確認に Secure Coding Checker[2] を利用してよい。また修正した MWS-Sample プログラムの提出は不要である。（各 1 点）

3. 解答例

MWS Cup 2013 参加チームから提出された事前課題レポートより, 解答内容を抜粋・要約する. 文末の () 内は参加チーム名を表す.

3.1. 設問1

- パスワードとユーザが入力したメモを資源と捉え、これらがユーザの意図と無関係に書き換えられたり、外部に流出したりする (Alkaneters)
- 暗黙的 Intent を送信して同一アプリ内の PasswordActivity を呼び出すため, 他アプリに対してセンシティブな情報を送信したり, 意図せぬ戻り値を受け取った場合に意図しない動作が起きる. (セキュリティ讃歌)
- パスワードが漏洩すると、保存されたメモの内容が第三者に知られる。パスワードは他のサービス・アプリケーション等で「使いまわし」をしているユーザが多いため、サンプルアプリ以外での被害 も発生する(GOTO Love and

初代森研)

3.2. 設問2

- 送信された Intent は、「com.example.mws_sample.ShowPassword」という名前のアクションが指定された Intent を受信する Intent-filter を記述することにより、他のアプリも受信することができる。(Alkaneters)
- PasswordActivity は非公開 Activity であり、MainActivity からのみ呼び出されることを想定しているが、マニフェストファイル (AndroidManifest.xml) の設定で外部アプリから参照可能であった。(GOTO Love and 初代森研)
- AndroidManifest.xml において、PasswordActivity は非公開 Activity であるにも関わらず、exported の属性が指定されておらず、かつ intent-filter が定義されている。この場合、あるアプリが別のアプリと同じ intent-filter を定義していると、同一アプリ内の非公開 Activity ではなく別のアプリの公開 Activity を呼び出してしまうことがある。(NU14)

3.3. 設問3

- 攻撃アプリ内の Activity から、PasswordActivity を呼び出すことで、正規のものと同一のパスワード画面が現れるため誤ってパスワードを入力してしまう可能性がある。(人海戦術チーム)
- PasswordActivity を模した不正な Activity を作成しサンプルプログラムの MainActivity から PasswordActivity を intent で呼び出す際に、模倣した Activity を選択肢に現れるよう intent-filter を調整した。不正なアプリを選択してしまった場合、外観は MWS-Sample プログラムの PasswordActivity と同様のものが表示され、誤ってパスワード入力してしまう(KIT-C-UT)
- MainActivity からの暗黙的 Intent が生成された際 に、ユーザに選択を促すダイアログが表示されるような Malicious アプリを作成。MaliciousApp を選択した場合に攻撃が成立し、Intent の横取り、パスワードの盗聴が可能になる(GOTO Love and 初代森研)

3.4. 設問4

- Intent を利用した呼び出しを明示的 Intent を使うように変更した。また PassowrdActicity の intent-filter を消去し、exported=false とした。これにより

同じ Action、Category を持つ Activity が他のアプリケーションに存在しても、もとのアプリケーション PasswordActivity しか実行できないようにした。(Team Enu)

- PasswordActivity に exported 要素を追加し、false を指定した。また、同じ intent-filter の action の値を持つ他のアプリが存在すると、意図せずにその他アプリの Activity を呼び出してしまうため Intent Filter を削除した(KIT-C-UT)
- Intent の盗聴による情報漏洩・改ざんに対しては、以下の 2 点を修正する
  1. PasswordActivity を非公開 Activity とする
  2. MainActivity からの呼出には明示的インテントを用いる
  (GOTO Love and 初代森研)
- PasswordActivity に関する記述箇所において、exported 属性を無属性から false に明示化した。また、export 属性直後の intent-filter を削除し、代わりに MainActivity.java では、暗黙的 intent から、クラス指定の明示的 intent で呼び出すように変更した(NU14)

4. 総評

全体として，脆弱性の内容とその対策を問うた設問2および設問4については正答率が高く，被害と攻撃方法を問うた設問1および設問3について正答率が低いという傾向が見られた．この結果から，参考文献 [1] [2]などを参照することで安全なAndroidアプリを開発する方法の学習および実践が十分に可能であり、現状でもアプリ開発者に対する情報提供が適切にされていることがわかる．一方で，Android アプリの脆弱性に対する攻撃方法やそれによって発生する被害の特定といった，リスク判断や疑似攻撃検査等のより高度でより実践経験を必要とする技術に関しては、参考文献 [1] [2]などでは詳しく扱っておらず、実践も重要なため、正解率の低下につながったものと考えられる．

今後は、参考文献 [1] [2]などの内容をより充実させるとともに、不足しているリスク判断や疑似攻撃検査といった分野の情報提供や専門家の育成が重要な施策となってくるであろう．

参考文献

[1] JSSEC「Android アプリのセキュア設計・セキュアコーディングガイド」
http://www.jssec.org/dl/android_securecoding.pdf

[2] Sony Digital Network Applications, Inc.「Secure Coding Checker」
http://www.sonydna.com/sdna/solution/scc.html