CSCI 445 - Homework #4

Matthew Berthoud

February 22 2024

1 Automated Security Analysis: Vulnerability Report

1. GCash overrides checkServerTrusted as shown in the output below:

```
GCash/smali/com/globe/gcash/android/activity/login/UrlConnectionUtil$1.smali
6: .implements Ljavax/net/ssl/X509TrustManager;
47: .method public checkServerTrusted
    ([Ljava/security/cert/X509Certificate;Ljava/lang/String;)V
```

   Upon manual review of the offending file, the equivalent java code has a checkServerTrusted method with an empty body. Therefore, this app is overriding default behavior to verify if a server is trusted, and instead proceed with the connection regardless. So, this is a true positive.

2. Zuum uses the AllowAllHostnameVerifier object, as identified in the output below:

```
Zuum/smali/com/m4u/vivozuum/comunicacao/https/ComunicacaoHttps.smali
280: .method private createTrustManager()[Ljavax/net/ssl/TrustManager;
291:     new-array v0, v4, [Ljavax/net/ssl/TrustManager;
302:     .local v0, "trustAllCerts":[Ljavax/net/ssl/TrustManager;
461:     invoke-direct {p0}, Lcom/m4u/vivozuum/comunicacao/https/
        ComunicacaoHttps;->createTrustManager()[Ljavax/net/ssl/TrustManager;
469:     invoke-virtual {v1, v2, v4, v5}, Ljavax/net/ssl/SSLContext;
        ->init([Ljavax/net/ssl/KeyManager;[
        Ljavax/net/ssl/TrustManager;Ljava/security/SecureRandom;)V
474:     new-instance v4, Lorg/apache/http/conn/ssl/AllowAllHostnameVerifier;
476:     invoke-direct {v4}, Lorg/apache/http/conn/ssl/AllowAllHostnameVerifier;
        -><init>()V
```

   This bypasses checking of connected hosts, which can expose the app to malicious traffic. The smali equates to setting the hostname verifier to the AllowAllHostnameVerifier, which is always insecure, so this is a true positive.

3. Oxigen "handles" SSL Errors by simply proceeding, as seen here:

```
OxigenWallet/smali/com/oxigen/oxigenwallet/hc.smali
46: .method public onReceivedSslError(
    Landroid/webkit/WebView;
```

```
    Landroid/webkit/SslErrorHandler;
    Landroid/net/http/SslError;)V
51:     invoke-virtual {p2}, Landroid/webkit/SslErrorHandler;->proceed()V
```

This equates to an empty body under the error handler, meaning if there is an SSL vulnerability/error, it will be caught by the handler, but no code will run, so it will slip through the system. Therefore, this is a true positive.

4. Money On Mobile (and every other app besides Zuum) uses a boatload of http links instead of https. This never acceptable, especially when dealing with finances, as this app's name suggests. There is no good reason to use http over https at this point in time, so no further checking is required to determine all http:// links are true positives of an SSL related vulnerability. I've provided one of many examples of the output from Money on Mobile below:

```
Money On Mobile/smali/com/mom/app/MSwipeAndroidSDKListActivity1.smali
120:     const-string v9,
        "http://msvc.money-on-mobile.net/WebServiceV3Client
        .asmx/getBalanceByCustomerId"
```

2 False Positives

1. In some cases, the SslErrorHandler was overridden by just referring back to the super class's handler, which is safe behavior, assuming the super class has a safely defined error handler. In the example below, the super class is WebViewClient, which is a class in android.webkit. Assuming this library has safe handling of SSL errors, we can call this a false positive.

```
OxigenWallet/smali/com/oxigen/oxigenwallet/c.smali
25: .method public onReceivedSslError(Landroid/webkit/WebView;
    Landroid/webkit/SslErrorHandler;Landroid/net/http/SslError;)V
30:     invoke-super {p0, p1, p2, p3}, Landroid/webkit/WebViewClient;
        ->onReceivedSslError(Landroid/webkit/WebView;
        Landroid/webkit/SslErrorHandler;Landroid/net/http/SslError;)V
```