Mobile app vulnerability report for net.roamler

Platform: Android
Package Name: net.roamler
Package Version Name: 3.0.0
Package Version Code: 300114
Min Sdk: 14
Target Sdk: 21
MD5   : 07f4d4ff8b8a52de68497226b3f945d5
SHA1 : 19bc18871d78e82bf240beae6d8237e76abb5fcb
SHA256: a89dc010e031dc997746d622c5b0c488bb0f9f3f2133ab7b010c2d4452f3325b
SHA512: 69cbcc9e9d5ce7a933948a222761c8c96b2128e6e822c3d45fe3ba400897514e14969ec30ac3afe21f1731c0efa2fe79b542ce9c1fdce734bdb48786bfef6ac2

Found 8 potential issues:

ISSUE 1 : <#BID 64208, CVE-2013-6271#> Fragment Vulnerability Checking:
'Fragment' or 'Fragment for ActionbarSherlock' has a severe vulnerability prior to Android 4.4 (API 19).
Please check:
(1)http://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT
(2)http://developer.android.com/reference/android/preference/PreferenceActivity.html#isValidFragment(java.lang.String)
(3)http://stackoverflow.com/questions/19973034/isvalidfragment-android-api-19
(4)http://securityintelligence.com/new-vulnerability-android-framework-fragment-injection/
(5)http://securityintelligence.com/wp-content/uploads/2013/12/android-collapses-into-fragments.pdf
(6)https://cureblog.de/2013/11/cve-2013-6271-remove-device-locks-from-android-phone/
You MUST override 'isValidFragment' method in every "PreferenceActivity" class to avoid Exception throwing in Android 4.4:
Lti/modules/titanium/ui/android/TiPreferencesActivity;
All of the potential vulnerable "fragment":
Lcom/facebook/login/LoginFragment;
Lcom/google/android/gms/common/api/zzm;
Lcom/google/android/gms/common/api/zzn;
Lcom/google/android/gms/maps/MapFragment;
Lcom/google/android/gms/maps/StreetViewPanoramaFragment;
Lcom/google/android/gms/maps/SupportMapFragment;
Lcom/google/android/gms/maps/SupportStreetViewPanoramaFragment;
Lti/modules/titanium/ui/widget/tabgroup/TiUIActionBarTab$TabFragment;

ISSUE 2 : <KeyStore><Hacker> KeyStore Protection Checking:
The Keystores below seem using "byte array" or "hard-coded cert info" to do SSL pinning (Total: 2). Please manually check:
=> Lcom/facebook/ads/internal/http/a;->a(Lorg/apache/http/params/BasicHttpParams;
Lorg/apache/http/conn/scheme/SchemeRegistry;)V (0x14) ---> Ljava/security/KeyStore;->load(Ljava/io/InputStream; [C)V
=> Lcom/facebook/ads/internal/util/g;->b()Lorg/apache/http/client/HttpClient; (0x42) --->
Ljava/security/KeyStore;->load(Ljava/io/InputStream; [C)V

ISSUE 3 : <SSL_Security> SSL Implementation Checking (Verifying Host Name in Custom Classes):

This app allows Self-defined HOSTNAME VERIFIER to accept all Common Names(CN).
This is a critical vulnerability and allows attackers to do MITM attacks with hi
s valid certificate without your knowledge.
Case example:
(1)http://osvdb.org/96411
(2)http://www.wooyun.org/bugs/wooyun-2010-042710
(3)http://www.wooyun.org/bugs/wooyun-2010-052339
Also check Google doc: http://developer.android.com/training/articles/security-s
sl.html (Caution: Replacing HostnameVerifier can
be very dangerous).
OWASP Mobile Top 10 doc: https://www.owasp.org/index.php/Mobile_Top_10_2014-M3
Check this book to see how to solve this issue: http://goo.gl/BFb65r

To see what’s the importance of Common Name(CN) verification.
Use Google Chrome to navigate:
- https://www.google.com => SSL certificate is valid
- https://60.199.175.158/ => This is the IP address of google.com, but the CN
is not match, making the certificate invalid. You
still can go Google.com but now you cannot distinguish attackers from normal use
rs

Please check the code inside these methods:
Lti/modules/titanium/network/httpurlconnection/NullHostNameVerifier;->verify
(Ljava/lang/String; Ljavax/net/ssl/SSLSession;)Z

ISSUE 4 : <SSL_Security> SSL Implementation Checking (Verifying Host Name in Fiel
ds):
This app does not check the validation of the CN(Common Name) of the SSL certifi
cate ("ALLOW_ALL_HOSTNAME_VERIFIER" field or
"AllowAllHostnameVerifier" class).
This is a critical vulnerability and allows attackers to do MITM attacks with hi
s valid certificate without your knowledge.
Case example:
(1)http://osvdb.org/96411
(2)http://www.wooyun.org/bugs/wooyun-2010-042710
(3)http://www.wooyun.org/bugs/wooyun-2010-052339
Also check Google doc: http://developer.android.com/training/articles/security-s
sl.html (Caution: Replacing HostnameVerifier can
be very dangerous).
OWASP Mobile Top 10 doc: https://www.owasp.org/index.php/Mobile_Top_10_2014-M3
Check this book to see how to solve this issue: http://goo.gl/BFb65r

To see what’s the importance of Common Name(CN) verification.
Use Google Chrome to navigate:
- https://www.google.com => SSL certificate is valid
- https://60.199.175.158/ => This is the IP address of google.com, but the CN
is not match, making the certificate invalid. You
still can go Google.com but now you cannot distinguish attackers from normal use
rs

Please check the code inside these methods:
=> Lcom/facebook/ads/internal/util/g;->b ()Lorg/apache/http/client/HttpClien
t;
=> Lcom/facebook/ads/internal/http/a;->a (Lorg/apache/http/params/BasicHttpP
arams;
Lorg/apache/http/conn/scheme/SchemeRegistry;)V

ISSUE 5 : <SSL_Security> SSL Connection Checking:
URLs that are NOT under SSL (Total:4):
http://api.appcelerator.com/p/v1/geo?
=> Lti/modules/titanium/geolocation/TiLocation;->buildGeocoderURL(Ljava/
lang/String; Ljava/lang/String;
Ljava/lang/String; Ljava/lang/String; Ljava/lang/String; Ljava/lang/Str
ing;)Ljava/lang/String;
http://play.google.com/store/apps/details?id=com.facebook.orca

=> Lcom/facebook/messenger/MessengerUtils;->openMessengerInPlayStore(Landroid/content/Context;)V
http://plus.google.com/
=> Lcom/google/android/gms/common/internal/h;-><clinit>()V
=> Lcom/google/android/gms/common/internal/zzm;-><clinit>()V
http://www.roamler.co.uk/
=> Lnet/roamler/RoamlerAppInfo;->getUrl()Ljava/lang/String;

ISSUE 6 : <SSL_Security> SSL Implementation Checking (WebViewClient for WebView):
DO NOT use "handler.proceed();" inside those methods in extended "WebViewClient", which allows the connection even if the SSL
Certificate is invalid (MITM Vulnerability).
References:
(1)A View To A Kill: WebView Exploitation: https://www.iseclab.org/papers/webview_leet13.pdf
(2)OWASP Mobile Top 10 doc: https://www.owasp.org/index.php/Mobile_Top_10_2014-M3
(3)https://jira.appcelerator.org/browse/TIMOB-4488
Vulnerable codes:
Lcom/facebook/ads/internal/view/a$b;->onReceivedSslError(Landroid/webkit/WebView; Landroid/webkit/SslErrorHandler;
Landroid/net/http/SslError;)V
Lcom/facebook/ads/internal/view/c$a;->onReceivedSslError(Landroid/webkit/WebView; Landroid/webkit/SslErrorHandler;
Landroid/net/http/SslError;)V
Lti/modules/titanium/ui/widget/webview/TiWebViewClient;->onReceivedSslError(Landroid/webkit/WebView;
Landroid/webkit/SslErrorHandler; Landroid/net/http/SslError;)V

ISSUE 7 : <SSL_Security> SSL Certificate Verification Checking:
This app DOES NOT check the validation of SSL Certificate. It allows self-signed, expired or mismatch CN certificates for SSL
connection.
This is a critical vulnerability and allows attackers to do MITM attacks without your knowledge.
If you are transmitting users’ username or password, these sensitive information may be leaking.
Reference:
(1)OWASP Mobile Top 10 doc: https://www.owasp.org/index.php/Mobile_Top_10_2014-M3
(2)Android Security book: http://goo.gl/BFb65r
(3)https://www.securecoding.cert.org/confluence/pages/viewpage.action?pageId=134807561
This vulnerability is much more severe than Apple’s "goto fail" vulnerability: http://goo.gl/eFlovw
Please do not try to create a "X509Certificate" and override "checkClientTrusted", "checkServerTrusted", and "getAcceptedIssuers"
functions with blank implementation.
We strongly suggest you use the existing API instead of creating your own X509Certificate class.
Please modify or remove these vulnerable code:
[Confirm Vulnerable]
=> Lcom/facebook/ads/internal/util/q$1;
-> used by: Lcom/facebook/ads/internal/util/q;-><init>(Ljava/security/KeyStore;)V
=> Lti/modules/titanium/network/NonValidatingTrustManager;
-> used by: Lti/modules/titanium/network/NonValidatingSSLSocketFactory;-><init>()V
-> used by: Lti/modules/titanium/network/TiHTTPClient;->setUpSSL(Z Ljavax/net/ssl/HttpsURLConnection;)V

ISSUE 8 : <WebView><Remote Code Execution><#CVE-2013-4710#> WebView RCE Vulnerability Checking:

Found a critical WebView "addJavascriptInterface" vulnerability. This method can be used to allow JavaScript to control the host
application.
This is a powerful feature, but also presents a security risk for applications targeted to API level JELLY_BEAN(4.2) or below,
because JavaScript could use reflection to access an injected object’s public fields. Use of this method in a WebView containing
untrusted content could allow an attacker to manipulate the host application in unintended ways, executing Java code with the
permissions of the host application.
Reference:
1."http://developer.android.com/reference/android/webkit/WebView.html#addJavascriptInterface(java.lang.Object,
java.lang.String) "
2.https://labs.mwrinfosecurity.com/blog/2013/09/24/webview-addjavascriptinterface-remote-code-execution/
3.http://50.56.33.56/blog/?p=314
4.http://blog.trustlook.com/2013/09/04/alert-android-webview-addjavascriptinterface-code-execution-vulnerability/
Please modify the below code:
=> Lcom/facebook/ads/internal/view/a;-><init>(Landroid/content/Context; Lcom/facebook/ads/internal/view/a$a; I)V (0x68) --->
Lcom/facebook/ads/internal/view/a;->addJavascriptInterface(Ljava/lang/Object; Ljava/lang/String;)V
=> Lcom/facebook/ads/internal/view/c;->c()V (0x5e) --->
Lcom/facebook/ads/internal/view/c;->addJavascriptInterface(Ljava/lang/Object; Ljava/lang/String;)V
=> Lbolts/WebViewAppLinkResolver$2;->then(Lbolts/Task;)Lbolts/Task; (0x56) --->
Landroid/webkit/WebView;->addJavascriptInterface(Ljava/lang/Object; Ljava/lang/String;)V
=> Lti/modules/titanium/ui/widget/webview/TiWebViewBinding;->addJavascriptInterfaces()V (0x1c) --->
Landroid/webkit/WebView;->addJavascriptInterface(Ljava/lang/Object; Ljava/lang/String;)V
=> Lti/modules/titanium/ui/widget/webview/TiWebViewBinding;->addJavascriptInterfaces()V (0x2e) --->
Landroid/webkit/WebView;->addJavascriptInterface(Ljava/lang/Object; Ljava/lang/String;)V
=> Lti/modules/titanium/ui/widget/webview/TiWebViewBinding;->addJavascriptInterfaces()V (0x40) --->
Landroid/webkit/WebView;->addJavascriptInterface(Ljava/lang/Object; Ljava/lang/String;)V