Android FakeID

Android Fake ID

Android Fake ID is a vulnerability that allows malicious applications to impersonate specially trusted applications without any user notification.

The main issue

Android Package Installer doesn’t verify authenticity of certificate claims so....

A certificate can masquerade as any other certificate.

Why is this a security threat?

- It allows programs to escape their sandbox.
- It can tap into NFC hardware
- It can access the data and web traffic of other apps
- Ultimately, the whole system could be compromised

Timeline

February 2010 - All Android Users were at risk

April 2014 - Found by BlueBox and disclosed to Google and released for patching

April 2014 - Up to present day: Patched for certain models

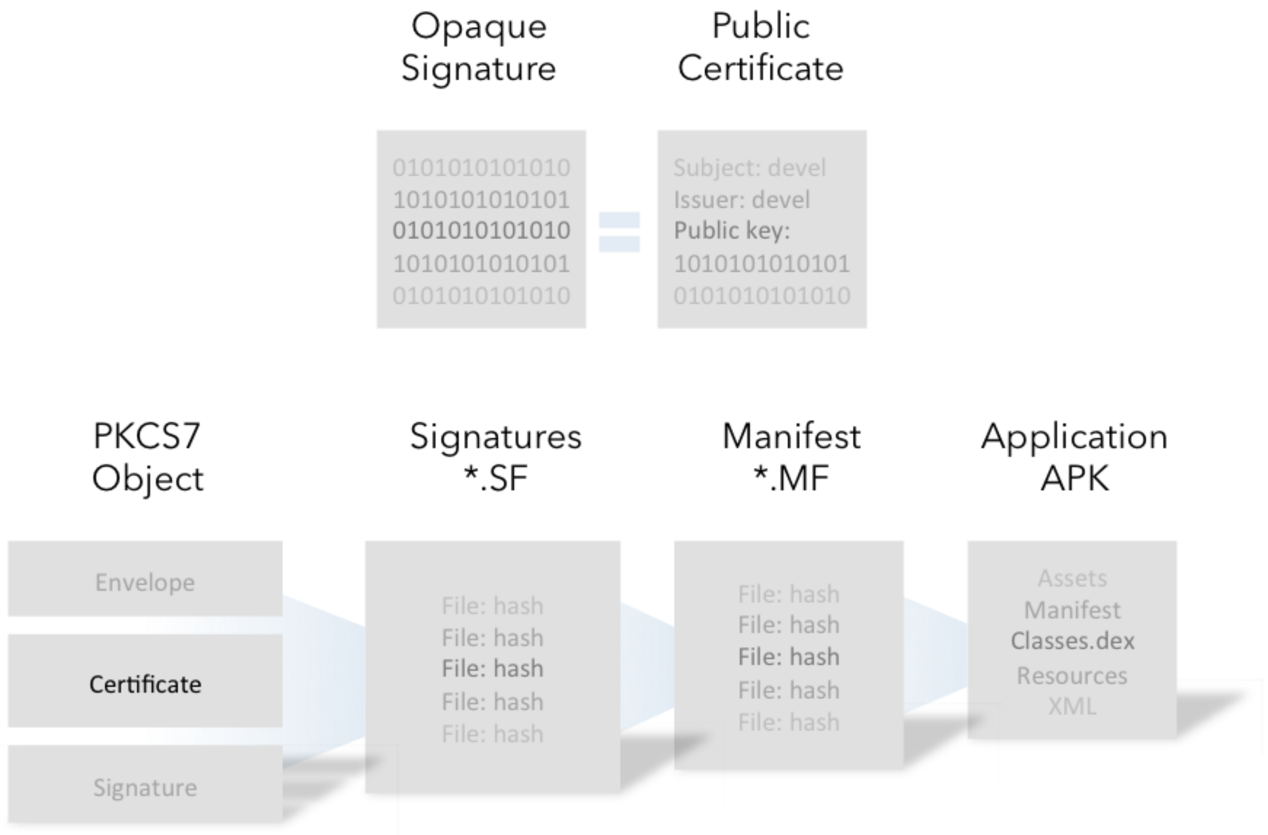
July 2014 - Publicly disclosed

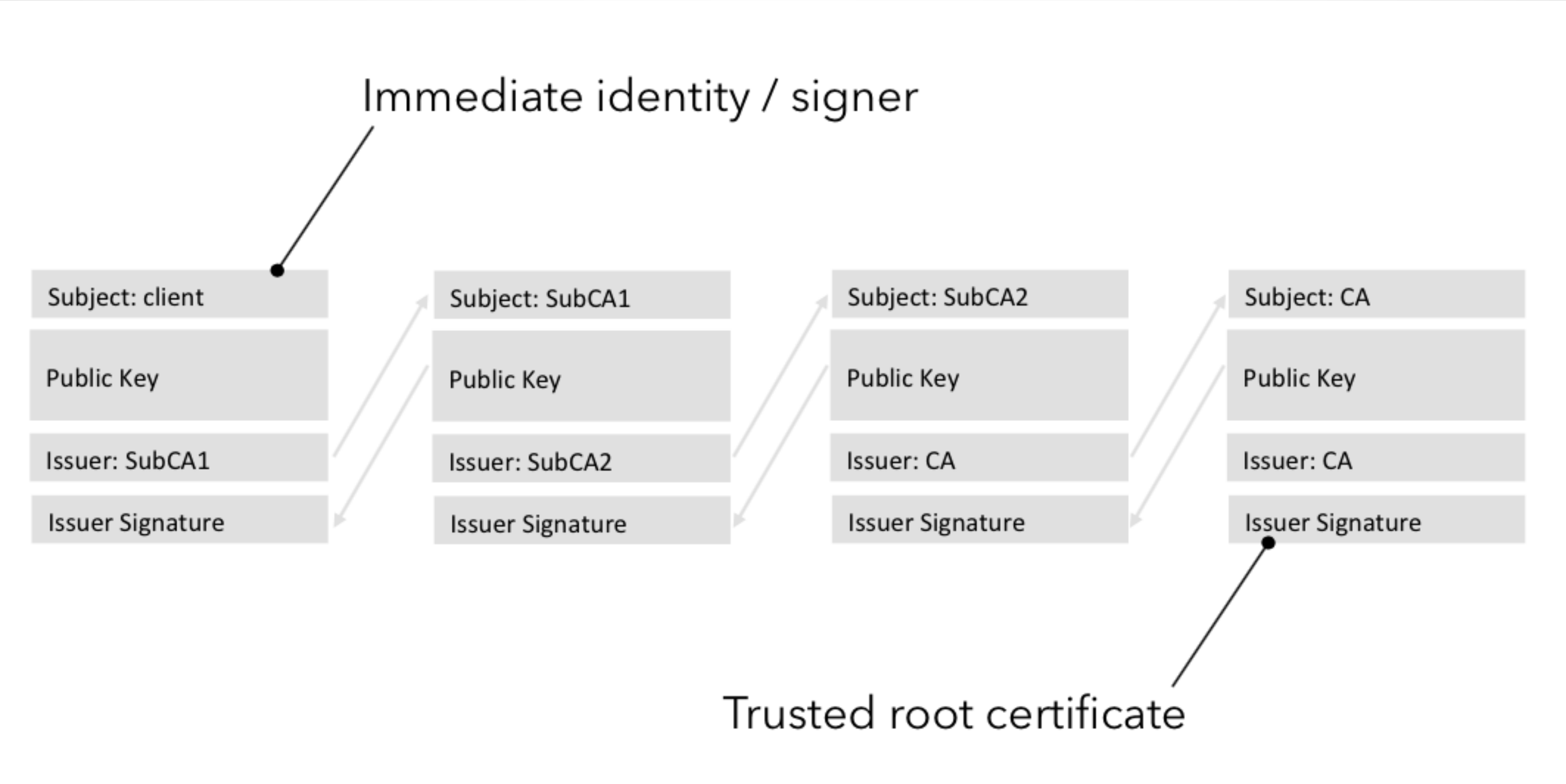
Where the bug originated

The faulty code originated in the Apache Harmony an open source alternative to Oracle’s Java.

Google couldn’t strike a deal with Oracle, so they used Harmony to support Java on its OS.

How the Attack Works


Opaque Signature
0101010101010
1010101010101
0101010101010
1010101010101
0101010101010
Public Certificate
Subject: devel
Issuer: devel
Public key:
1010101010101
0101010101010
PKCS7 Object
Envelope
Certificate
Signature
Signatures *.SF
File: hash
File: hash
File: hash
File: hash
File: hash
Manifest *.MF
File: hash
File: hash
File: hash
File: hash
File: hash
Application APK
Assets
Manifest
Classes.dex
Resources
XML


Immediate identity / signer
Subject: client
Public Key
Issuer: SubCA1
Issuer Signature
Subject: SubCA1
Public Key
Issuer: SubCA2
Issuer Signature
Subject: SubCA2
Public Key
Issuer: CA
Issuer Signature
Subject: CA
Public Key
Issuer: CA
Issuer Signature
Trusted root certificate

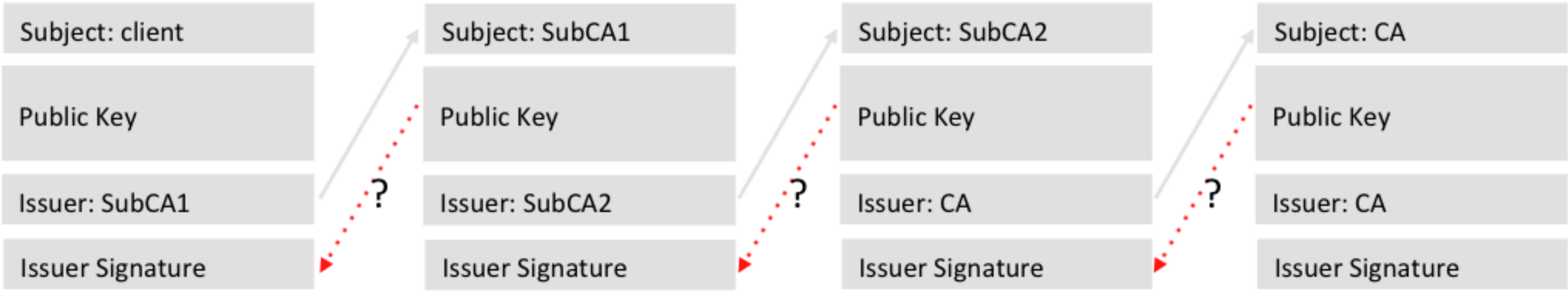
AndroidXRef Jelly Bean 4.3

xref: /libcore/luni/src/main/java/org/apache/harmony/security/utils/JarUtils.java

Home | History | Annotate | Line# | Navigate | Download Search only in **JarUtils.java**

```java
        if (!sig.verify(sigInfo.getEncryptedDigest())) {
            throw new SecurityException("Incorrect signature");
        }

        return createChain(certs[issuerSertIndex], certs);
    }

    private static X509Certificate[] createChain(X509Certificate  signer, X509Certificate[] candidates) {
        LinkedList chain = new LinkedList();
        chain.add(0, signer);

        // Signer is self-signed
        if (signer.getSubjectDN().equals(signer.getIssuerDN())){
            return (X509Certificate[])chain.toArray(new X509Certificate[1]);
        }

        Principal issuer = signer.getIssuerDN();
        X509Certificate issuerCert;
        int count = 1;
        while (true) {
            issuerCert = findCert(issuer, candidates);
            if( issuerCert == null) {
                break;
            }
            chain.add(issuerCert);
            count++;
            if (issuerCert.getSubjectDN().equals(issuerCert.getIssuerDN())) {
                break;
            }
            issuer = issuerCert.getIssuerDN();
        }
        return (X509Certificate[])chain.toArray(new X509Certificate[count]);
    }

    private static X509Certificate findCert(Principal issuer, X509Certificate[] candidates) {
        for (int i = 0; i < candidates.length; i++) {
            if (issuer.equals(candidates[i].getSubjectDN())) {
                return candidates[i];
            }
        }
        return null;
    }
```

The logic accepts _*one*_ trusted certificate _*anywhere*_ in signature/certificate chain


Subject: client
Public Key
Issuer: SubCA1
Issuer Signature
?
Subject: SubCA1
Public Key
Issuer: SubCA2
Issuer Signature
?
Subject: SubCA2
Public Key
Issuer: CA
Issuer Signature
?
Subject: CA
Public Key
Issuer: CA
Issuer Signature

Example of faked certificate attack

The app pretends to be created by Adobe Systems - Adobe is granted the privilege of being able to add code to other apps in order to support their use of its Flash media-player plug-in. The malware can take advantage of this to install Trojan horse malware into otherwise authentic programs

Webview plugin manager

- Plugins signed by Adobe (Flash) reloaded into any/all apps using framework webview

NFC access.xml

- Match a package signature wildcard (Google Wallet), get access to NFC secure element

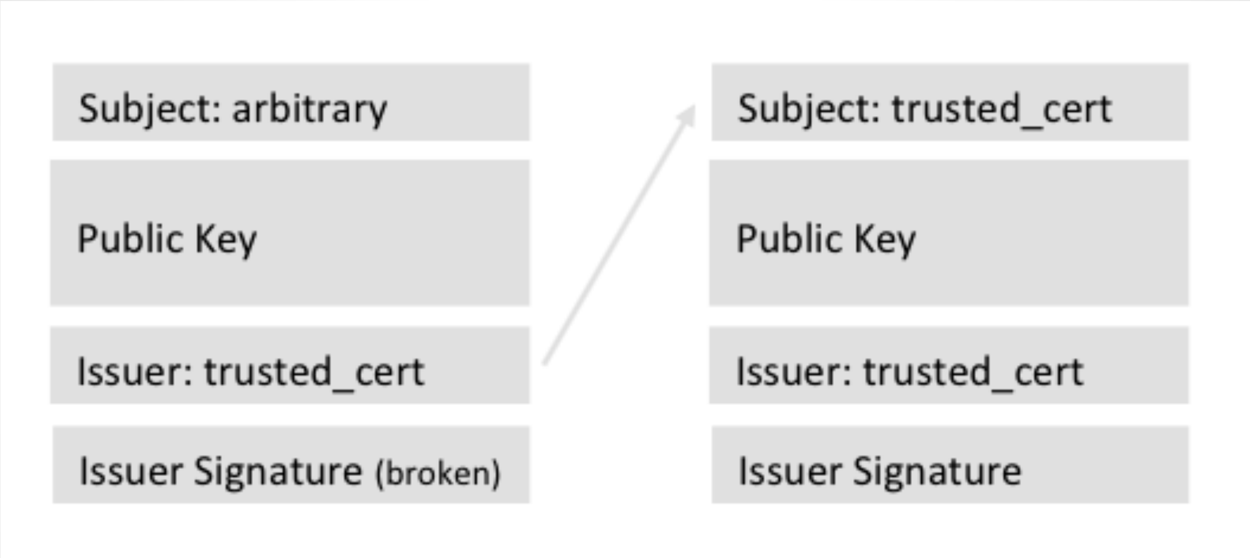
3LM device management extensions

- Former Google/Motorola technology, included with older devices LG

MDM device extensions

- System functions available to apps signed by LG platform signature

```
targetcert = OpenSSL.crypto.load_certificate( target )
pk = OpenSSL.crypto.PKey()
pk.generate_key( OpenSSL.crypto.TYPE_RSA, 1024)
newcert = OpenSSL.crypto.X509()
newcert.get_subject().CN = “arbitrary”
newcert.set_issuer( targetcert.get_subject() )
newcert.set_pubkey( pk )
newcert.sign( pk, “sha1” )
pkcs12 = OpenSSL.crypto.PKCS12()
pkcs12.set_privatekey( pk )
pkcs12.set_certificate( cert )
pkcs12.set_ca_certificates( [targetcert] )
finalPkcs12Data = pkcs12.export( passphrase=“1234” )
```


Subject: arbitrary
Public Key
Issuer: trusted_cert
Issuer Signature (broken)
Subject: trusted_cert
Public Key
Issuer: trusted_cert
Issuer Signature

How Google changed it

1) Google produced a code fix, provided it to Android manufacturers
2) Phone manufacturers must incorporate that fix into the firmware update for each phone
3) Carrier distributes final updates to phone manufacturers

Comparisons to iOS market

- Apple realized the importance of third party development in 2006
- Spent a year developing secure development kits
- It cited Flash's susceptibility to malware when the company refused to allow Flash into iOS.

Potential Harm

- Android FakeID has the ability to install viruses onto the phone
- No current exploits / damages are known or have been reported
- Vulnerability still exists for many android users

Public Response

When BlueBox reported the issue to Google, less than 4% of users were updated to KitKat (Android 4.4)

39.1% of all current Android users are running the latest version. That leaves 60 percent of current Android users are at risk.

"Fake ID unfortunately occurs in a manner that is hidden to the user - there's no prompts, no notifications, no need for special permissions. “

Mobile malware grew

155% in 2011

614% from March 2012 to March 2013

73% of all malware exploit holes in mobile payments by sending fraudulent premium SMS messages, each generating around $10 USD in immediate profit

Android is responsible for 92% of all known mobile malware. An increase from 47% in 2012...

...a significant threat given more than

1 BILLION

Android-based smart phones are estimated to be shipped in 2017

Source: Canalys Smart Phone Report, June 2013

There are more than

500

third-party app stores containing malicious apps

77% of Android threats could be largely eliminated today if all Android devices had the latest OS. Currently only 4% do

Things to do for Android Users

- Update to KitKat(Android 4.4)+
- Check the legitimacy of all downloaded apps
- Download Bluebox Security Scanner (on Google Play Store)

Android’s Vulnerability

“We do not guarantee that Android is designed to be safe; its format was designed to give more freedom.”

- Sundar Pichai (Vice President of Google)

Sources:

http://www.theguardian.com/technology/2014/jul/29/android-fake-id-flaw-google-patch

https://bluebox.com/technical/android-fake-id-vulnerability/

http://appleinsider.com/articles/14/07/29/new-android-fake-id-flaw-empowers-stealthy-new-class-of-super-malware-

http://www.androidcentral.com/fake-id-and-android-security-updated
http://www.computerworld.com/article/2476573/malware-vulnerabilities/your-android-has-a-fake-id-problem--allowing-malware-to-impersonate-trusted-.html

http://appleinsider.com/articles/14/07/29/new-android-fake-id-flaw-empowers-stealthy-new-class-of-super-malware-

http://www.bbc.com/news/technology-28544443

http://www.dailymail.co.uk/sciencetech/article-2709919/Massive-security-alert-Android-Fake-ID-bug-experts-warn-personal-financial-details-millions-users-risk-four-YEARS.html

http://www.theguardian.com/technology/2014/jul/29/android-fake-id-flaw-google-patch

http://www.pcworld.com/article/2459240/android-vulnerability-allows-malware-to-compromise-most-devices-and-apps.html

http://www.computerworld.com/article/2476573/malware-vulnerabilities/your-android-has-a-fake-id-problem--allowing-malware-to-impersonate-trusted-.html

http://news.slashdot.org/story/14/07/29/2230242/old-apache-code-at-root-of-android-fakeid-mess

https://securityledger.com/2014/07/old-apache-code-at-root-of-android-fakeid-mess/