Android Security Development

PART 1 – App Development

SEAN

Sean

- Developer
- erinus.startup@gmail.com
- https://www.facebook.com/erinus

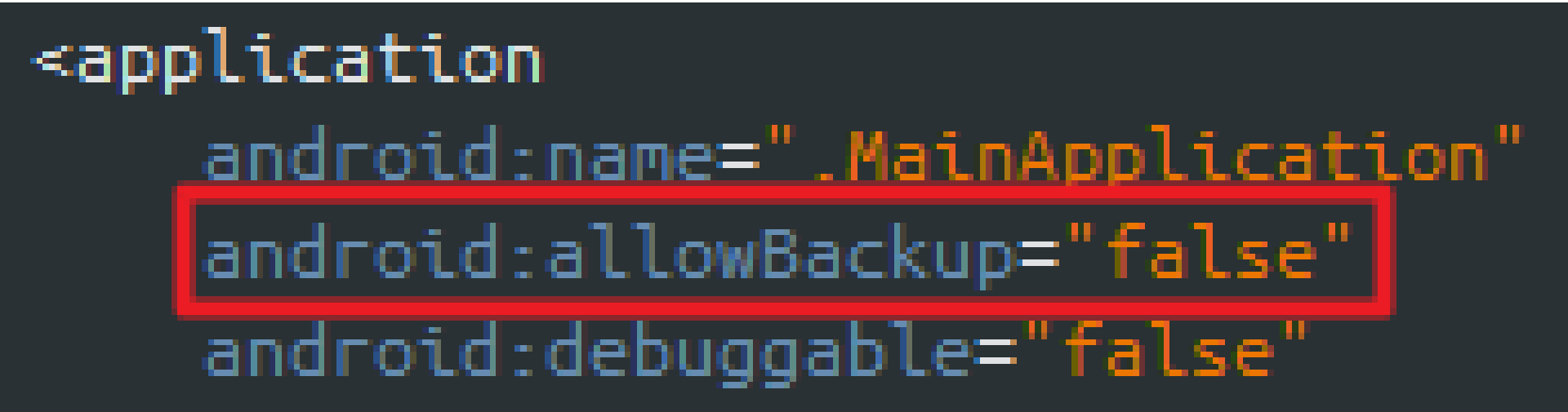
Something you need to know

- USB
- Screen
- Clipboard
- Permission
- Database
- Network
- Cryptography
- API Management

Security about USB

SAFE

ANDROID:ALLOWBACKUP = "FALSE"



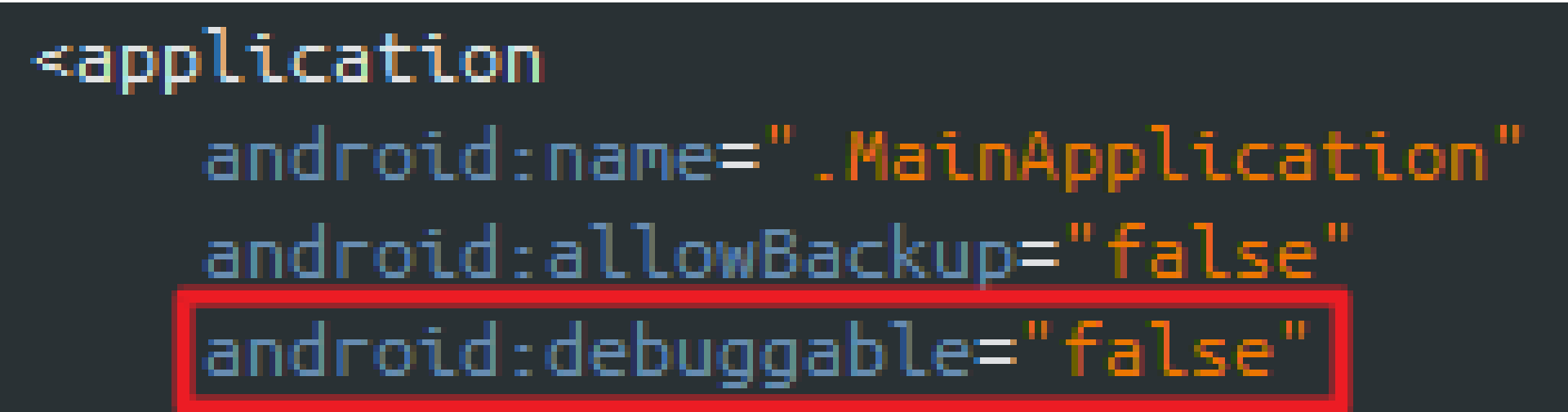
DANGEROUS

ANDROID:ALLOWBACKUP = "TRUE"

It will allow someone can backup databases and preferences.

SAFE

ANDROID:DEBUGGABLE = "FALSE"



DANGEROUS

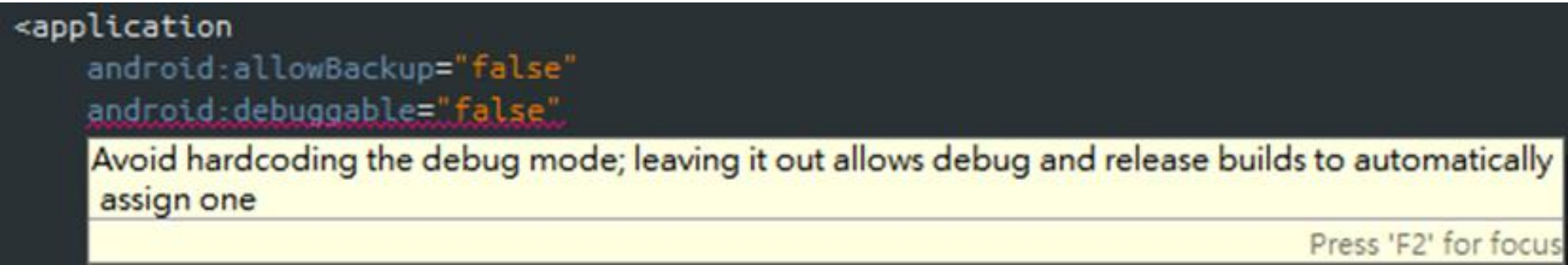
ANDROID:DEBUGGABLE = "TRUE"

It will let someone can see log message and do something more …

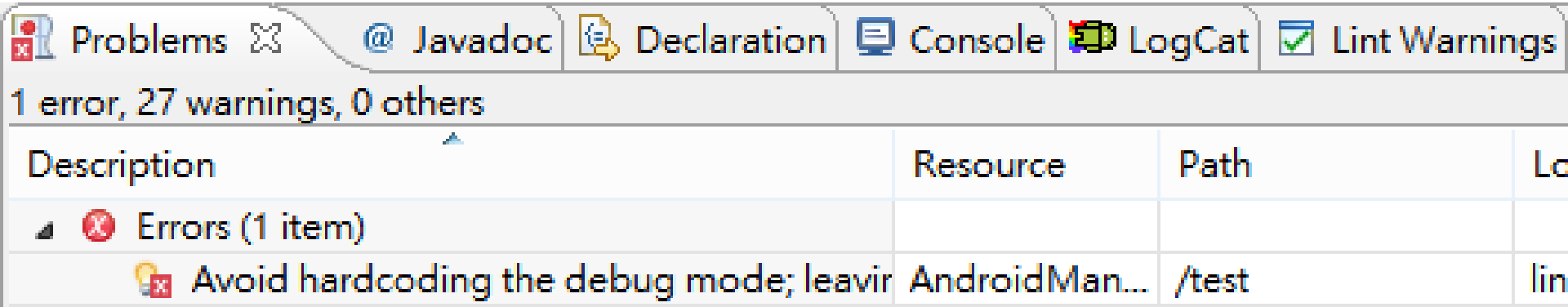
WHY ?

If you do not set android:debuggable="false", debug mode will depend on system setting.

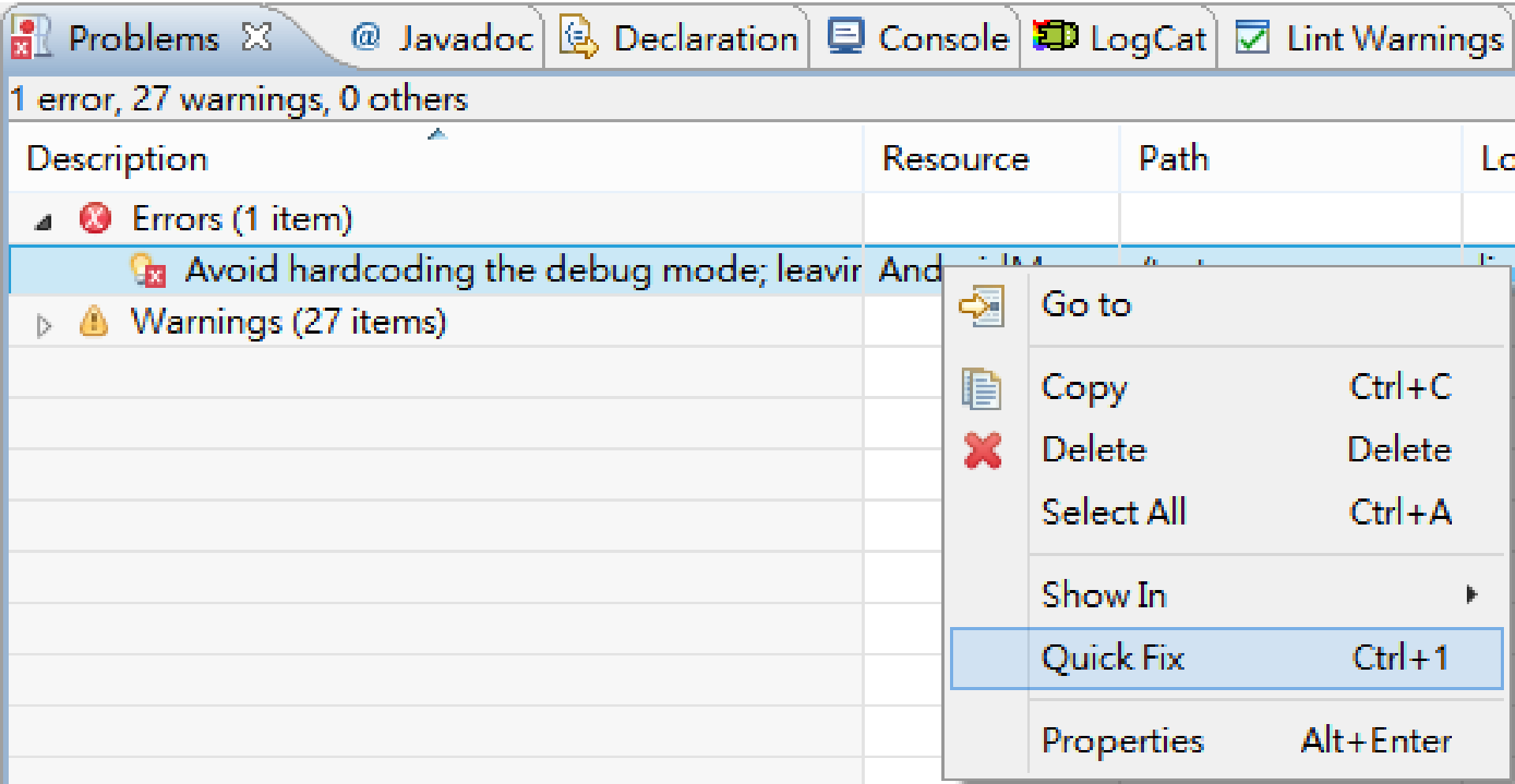
IF ERROR NOTIFICATION SHOWS IN ECLIPSE WHEN SET ANDROID:DEBUGGABLE, IT IS ALL ABOUT ADT LINT.



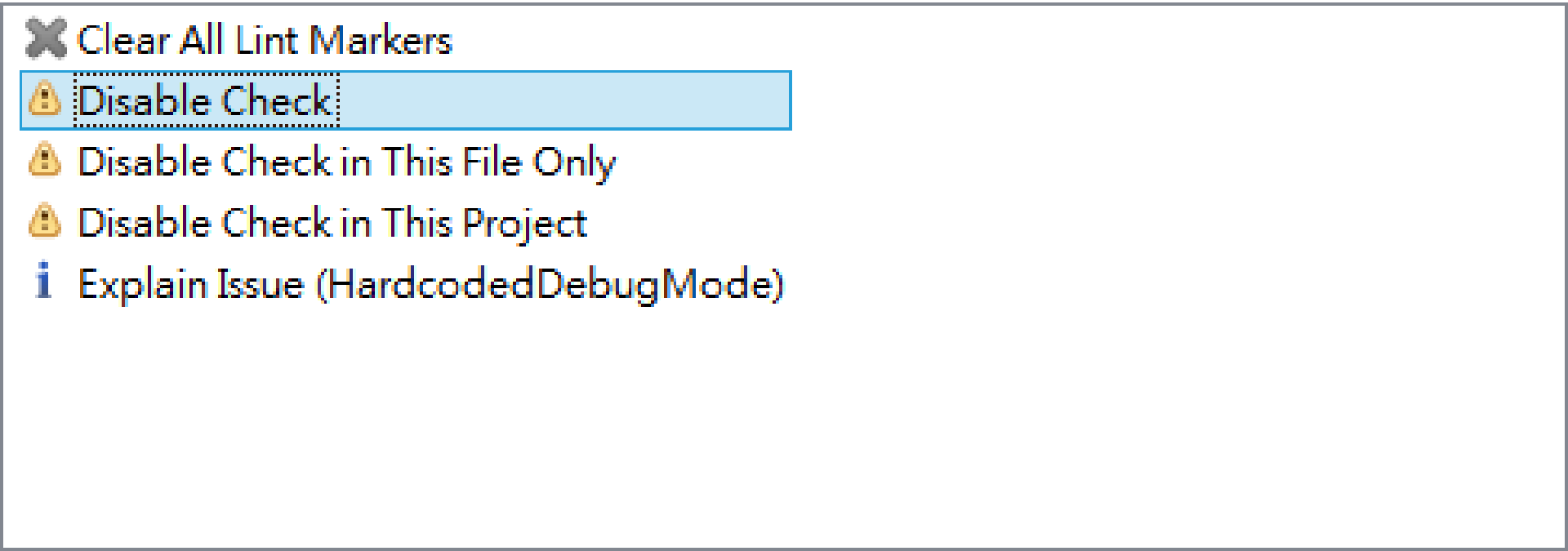
CLICK ON "PROBLEMS" TAB



RIGHT CLICK ON ITEM
AND CHOOSE "QUICK FIX"



CHOOSE "DISABLE CHECK"



Security about SCREEN

GETWINDOW().SETFLAGS(LAYOUTPARAMS.FLAG_SECURE, LAYOUTPARAMS.FLAG_SECURE);

It disable all screen capture (except rooted device)

- [POWER] + [VOL-DWN]
- OEM feature like SAMSUNG / HTC

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    BaseActivity.this.getWindow().setFlags(LayoutParams.FLAG_SECURE,
            LayoutParams.FLAG_SECURE);
}
```

Security about CLIPBOARD

WHEN USER LEAVE APP

You want to clear clipboard

YOU WANT TO ALLOW

User use something copied from other apps in your app

ALSO WANT TO REJECT

User can not use something copied from your app in other apps

FIRST

SAVE THE STATE OF APPLICATION

onResume => FOREGROUND
onPause => BACKGROUND

SECOND

USE RUNNABLE AND POSTDELAYED 500 MS

When onPause is triggered, you can detect the state of application after 500ms.

LAST

DETECT STATE AND SETPRIMARYCLIP

If STATE equals BACKGROUND, execute

BaseActivity.this.mClipboardManager

.setPrimaryClip(ClipData.newPlainText("", ""));

THE TOP ITEM WILL BE EMPTY
IN CLIPBOARD STACK

Android only let app access the top item in clipboard stack on non-rooted device.

Security on **PERMISSION**

ONLY USE NECESSARY PERMISSIONS

IT IS COMMON SENSE

BUT SOMETHING MORE

GOOGLE CLOUD MESSAGING
NEEDS
ANDROID.PERMISSION.GET_ACCOUNTS

BUT

GOOGLE CLOUD MESSAGING
NEEDS
ANDROID.PERMISSION.GET_ACCOUNTS

- The `com.google.android.c2dm.permission.RECEIVE` permission so the Android application can register and receive messages.
- The `android.permission.INTERNET` permission so the Android application can send the registration ID to the 3rd party server.
- The `android.permission.GET_ACCOUNTS` permission as GCM requires a Google account (necessary only if if the device is running a version lower than Android 4.0.4)
- The `android.permission.WAKE_LOCK` permission so the application can keep the processor from sleeping when a message is received. Optional—use only if the app wants to keep the device from sleeping.

ONE YEAR LATER

YOU SHOULD REMOVE "GET_ACCOUNTS"

When you do not support
Android 4.0.3 and older version

Security on Database

SQLITE

RECOMMENDED

SQLCipher

Support iOS / Android

https://www.zetetic.net/sqlcipher/open-source

SQLite Encryption Extension

http://www.sqlite.org/see/

Security on NETWORK

USE HTTPS WITH SELF-SIGNED CERTIFICATE

BUT

SOMETHING IGNORED ?

DO YOU CHECK
HOSTNAME IS VALID ?

VERIFY HOSTNAME

```
HttpsURLConnection
        .setDefaultHostnameVerifier(new HostnameVerifier() {
            @Override
            public boolean verify(String hostname,
                    SSLSession session) {
                return hostname.equals(host);
            }
        });
```

DO YOU AVOID
IMPORTING MALICIOUS CERT ?

CREATE BRAND NEW KEYSTORE AND IMPORT SERVER CERT

```
CertificateFactory certificateFactory = CertificateFactory
        .getInstance("X.509");
Certificate certificate = certificateFactory
        .generateCertificate(cert);
KeyStore keyStore = KeyStore.getInstance(KeyStore.getDefaultType());
keyStore.load(null, null);
keyStore.setCertificateEntry(host, certificate);
TrustManagerFactory trustManagerFactory = TrustManagerFactory
        .getInstance(TrustManagerFactory.getDefaultAlgorithm());
trustManagerFactory.init(keyStore);
SSLContext sslContext = SSLContext.getInstance("TLS");
sslContext.init(null, trustManagerFactory.getTrustManagers(), null);
HttpsURLConnection.setDefaultSSLSocketFactory(sslContext
        .getSocketFactory());
```

DOUBLE CHECK
THE BINARY CONTENT IFCERT ?

VERIFY BINARY CONTENT OF SERVER CERT

Avoid Man-in-the-Middle attack

```
for (Certificate cert : conn.getServerCertificates()) {
    if (cert instanceof X509Certificate) {
        MessageDigest messageDigest = MessageDigest.getInstance("SHA-1");
        messageDigest.update(cert.getEncoded());
        if (Arrays.equals(messageDigest.digest(),
                Hex.decodeHex(hash.toCharArray()))) {
            result = true;
            break;
        } else {
            result = false;
        }
    }
}
```

WHY ?

SSL MECHANISM IN OS MAY BE WRONG

APPLE SSL / TLS Bug (CVE-2014-1266)

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer
signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
    OSStatus        err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

Chinese MITM Attack on iCloud

Experts in network security monitoring and network forensics

NETRESEC

NETRESEC | Products | Resources | Blog | About Netresec |

NETRESEC > Blog

Monday, 20 October 2014 13:35:00 (UTC/GMT)

Chinese MITM Attack on iCloud

Users in China are reporting a MITM attacks on SSL connections to iCloud.

GreatFire.org, who monitor the Great Firewall of China (GFW), also published a blog post on their website earlier today saying:

> This is clearly a malicious attack on Apple in an effort to gain access to usernames and passwords and consequently all data stored on iCloud such as iMessages, photos, contacts, etc.

Fake SSL Certificate

In their blog post GreatFire also linked a packet capture file, which we have analyzed in order to verify the MITM attack. We loaded the PcapNG file into NetworkMiner Professional and extracted the X.509 SSL certificate.

Recent Blog Posts

» Chinese MITM Attack on iCloud

» Verifying Chinese MITM of Yahoo

» Analysis of Chinese MITM on Google

» Running NetworkMiner on Mac OS X

» NetworkMiner 1.6 Released

» PCAP or it didn't happen

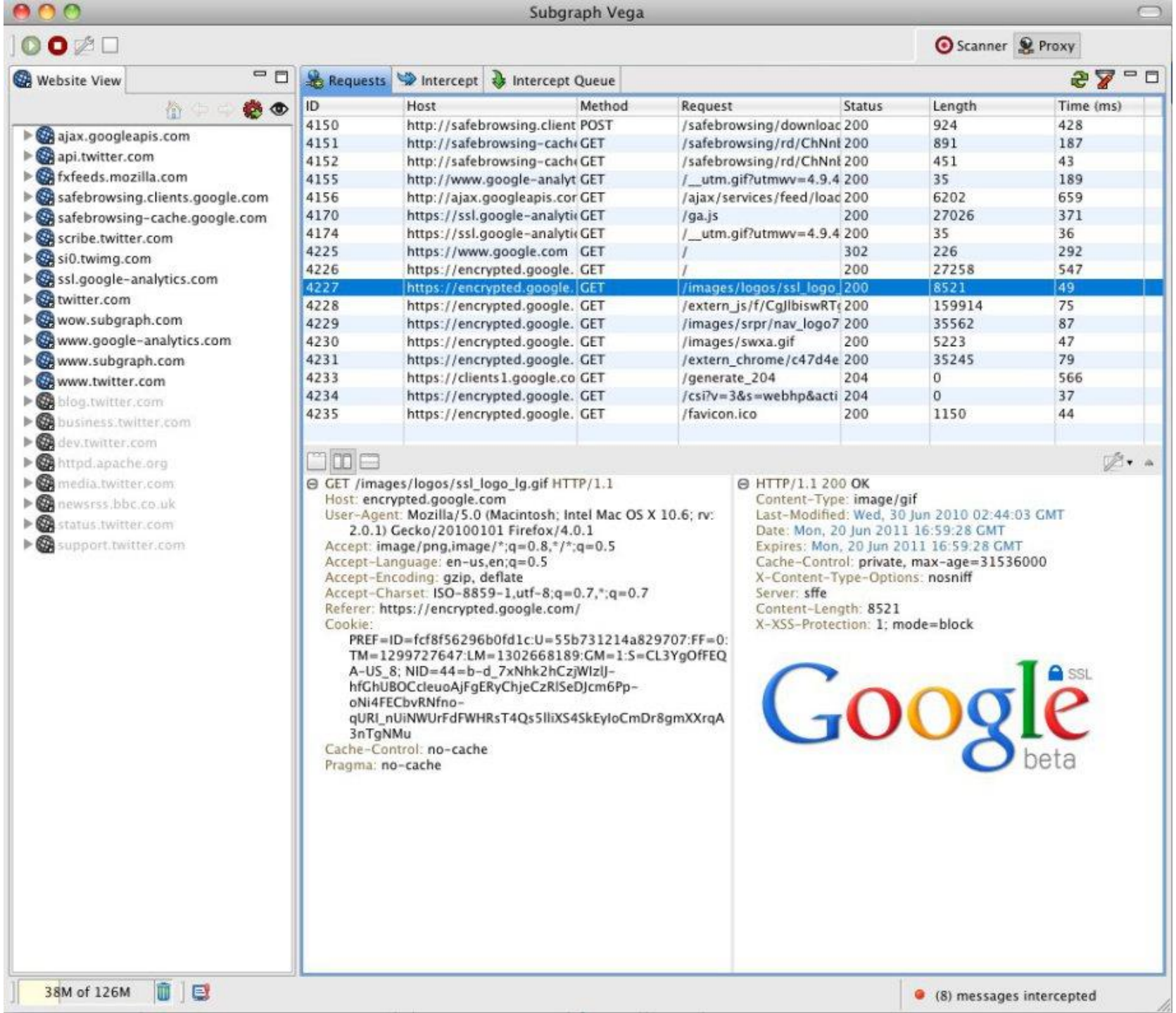
Blog Archive

» 2014 October
» 2014 September
» 2014 June
» 2014 May

SSL TUNNEL KEEP DATA SAFE ?

NO

YOU STILL NEED ENCRYPT DATA


Subgraph Vega
Scanner
Proxy
Website View
ajax.googleapis.com
api.twitter.com
fxfeeds.mozilla.com
safebrowsing.clients.google.com
safebrowsing-cache.google.com
scribe.twitter.com
si0.twimg.com
ssl.google-analytics.com
twitter.com
wow.subgraph.com
www.google-analytics.com
www.subgraph.com
www.twitter.com
blog.twitter.com
business.twitter.com
dev.twitter.com
httpd.apache.org
media.twitter.com
newsrss.bbc.co.uk
status.twitter.com
support.twitter.com
Requests
Intercept
Intercept Queue
ID Host Method Request Status Length Time (ms)
4150 http://safebrowsing.client POST /safebrowsing/downloac 200 924 428
4151 http://safebrowsing-cach GET /safebrowsing/rd/ChNnl 200 891 187
4152 http://safebrowsing-cach GET /safebrowsing/rd/ChNnl 200 451 43
4155 http://www.google-analyt GET /__utm.gif?utmwv=4.9.4 200 35 189
4156 http://ajax.googleapis.cor GET /ajax/services/feed/loac 200 6202 659
4170 https://ssl.google-analyti GET /ga.js 200 27026 371
4174 https://ssl.google-analyti GET /__utm.gif?utmwv=4.9.4 200 35 36
4225 https://www.google.com GET / 302 226 292
4226 https://encrypted.google. GET / 200 27258 547
4227 https://encrypted.google. GET /images/logos/ssl_logo_ 200 8521 49
4228 https://encrypted.google. GET /extern_js/f/CgJlbiswRT 200 159914 75
4229 https://encrypted.google. GET /images/srpr/nav_logo7 200 35562 87
4230 https://encrypted.google. GET /images/swxa.gif 200 5223 47
4231 https://encrypted.google. GET /extern_chrome/c47d4e 200 35245 79
4233 https://clients1.google.co GET /generate_204 204 0 566
4234 https://encrypted.google. GET /csi?v=3&s=webhp&acti 204 0 37
4235 https://encrypted.google. GET /favicon.ico 200 1150 44
GET /images/logos/ssl_logo_lg.gif HTTP/1.1
Host: encrypted.google.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv: 2.0.1) Gecko/20100101 Firefox/4.0.1
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referer: https://encrypted.google.com/
Cookie:
PREF=ID=fcf8f56296b0fd1c:U=55b731214a829707:FF=0:TM=1299727647:LM=1302668189:GM=1:S=CL3YgOfFEQA-US_8; NID=44=b-d_7xNhk2hCzjWlzlJ-hfGhUBOCcleuoAjFgERyChjeCzRlSeDJcm6Pp-oNi4FECbvRNfno-qURI_nUiNWUrFdFWHRsT4Qs5lliXS4SkEyIoCmDr8gmXXrqA3nTgNMu
Cache-Control: no-cache
Pragma: no-cache
HTTP/1.1 200 OK
Content-Type: image/gif
Last-Modified: Wed, 30 Jun 2010 02:44:03 GMT
Date: Mon, 20 Jun 2011 16:59:28 GMT
Expires: Mon, 20 Jun 2011 16:59:28 GMT
Cache-Control: private, max-age=31536000
X-Content-Type-Options: nosniff
Server: sffe
Content-Length: 8521
X-XSS-Protection: 1; mode=block
Google
SSL
beta
38M of 126M
(8) messages intercepted

DO NOT PUT KEY IN YOUR DATA

Security on CRYPTOGRAPHY

USE ANDROID SDK OR ANDROID NDK ?

ANDROID SDK: JAVA

DECOMPILE EASY
ANALYSIS EASY

ANDROID NDK: C AND C++

DISASSEMBLE EASY
ANALYSIS HARD

ANDROID NDK

OpenSSL Inside

ANDROID NDK

Can I customize ?

ANDROID NDK

PolarSSL

https://polarssl.org

PolarSSL

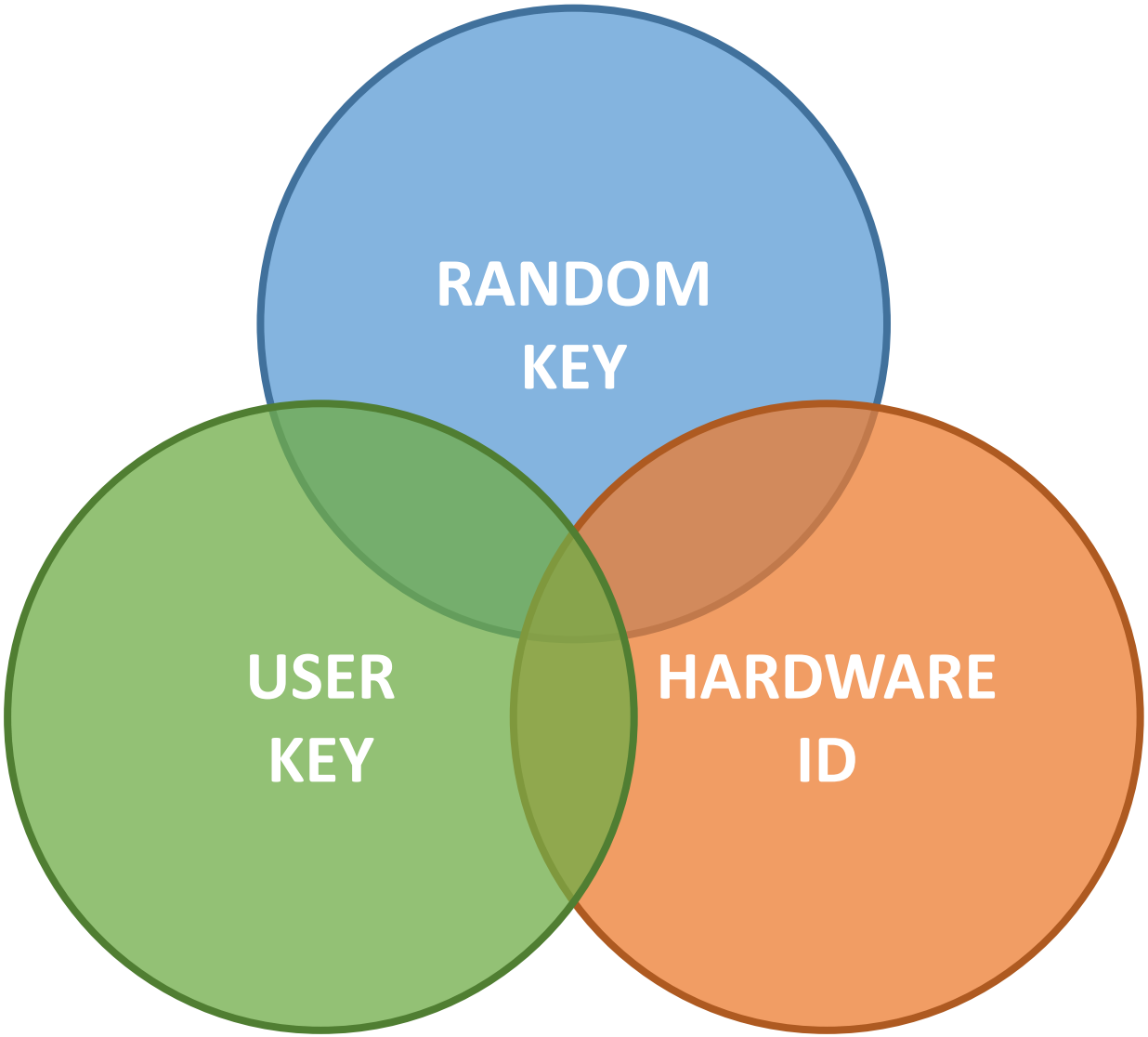
You can change SBOX of AES, ...

ALL KEY GENERATION AND ENCRYPTION MUST BE DONE IN ANDROID NDK

EVERYTHING IS DONE ?

NO

HOW TO GENERATE KEY ?


RANDOM
KEY
USER
KEY
HARDWARE
ID

RANDOM KEY

One Key – One Encryption

HARDWARE ID

IMEI / MEID

WIFI MAC Address

Bluetooth Address

IMEI / MEID

ANDROID.PERMISSION.READ_PHONE_STATE

WIFI MAC Address

ANDROID.PERMISSION.ACCESS_WIFI_STATE

Bluetooth Address

ANDROID.PERMISSION.BLUETOOTH

USER KEY

Input from user

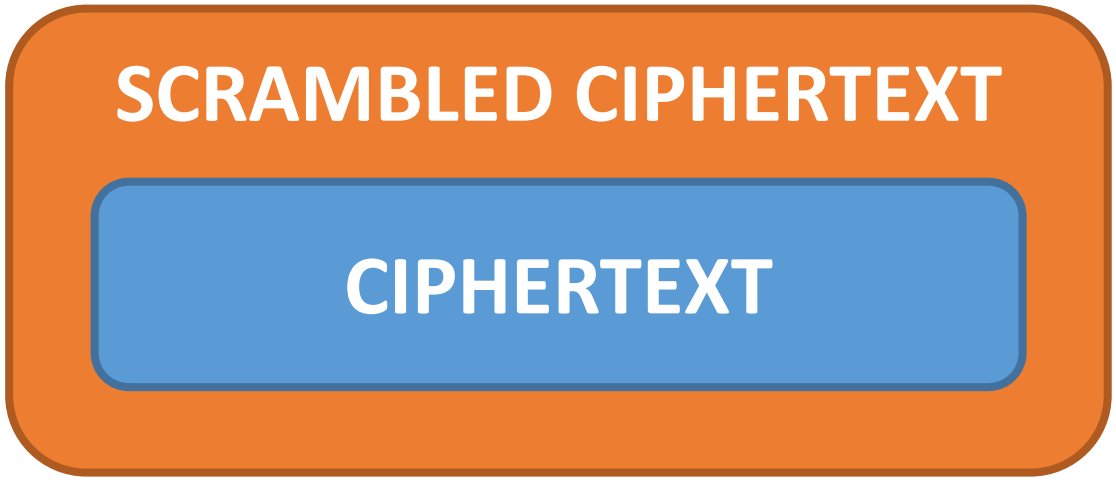
Only exist in memory

Just clear when exit

ONLY CIPHERTEXT ?

SCRAMBLE YOUR CIPHERTEXT

WEP can be cracked by collecting large amount packet and analyzing ciphertext.


SCRAMBLED CIPHERTEXT
CIPHERTEXT

HOW TO SCRAMBLE ?

MORE COMPLEX THAN BASE64

WIKI: Common Scrambling Algorithm
http://goo.gl/eP6lXj

IF ALL KEY LOST ?

SORRY

GOD BLESS YOU

API MANAGEMENT

ACCESS TOKEN

REFRESH PERIODICALLY

RANDOM GENERATE

HOW TO USE ACCESS TOKEN ?

ACCESS TOKEN

USER ID

ACCESS TOKEN

USER ID

HARDWARE ID

ACCESS TOKEN

USER ID

HARDWARE ID

ENCRYPT OR DECRYPT

ALL API ACCESS MUST USE ACCESS TOKEN

Next Part

Malicious Android App Dynamic Analyzing System