Android AIDS In CSharp: Automatic Intelligence De-advertisement Scheme
Hello!

aaaddress1(ADR)

1. Reverse Engineering Skill
   a. Windows
   b. Android
2. TDoHacker Core Member
3. Hack Bot
   a. CrackShield / MapleHack
   b. Tower Of Savior
   c. FaceBook
4. CSharp, C/C++, Assembly x86
   Python, Smali, Pascal, VB.NET
Outline

➢ Why Do We Need AIDS?
➢ How Ads Were Displayed?
➢ How Android Works.
➢ Android Basic Cracking.
➢ Live Demo Of Patching Ads In APK.
➢ Ads In Game Engine: Unity.
➢ ProGuard Obfuscation.
➢ How To Build This Tool In CSharp.
➢ Live Demo.
Why Do We Need AIDS?
To Surf The Net.
To Surf The Net
結果還是GG，上面那指令還真的有想過（抹臉）

尤理衡 嘿嘿

留言......

韋倪豪和柯炯勝都分享了 7Lions 的影片。

7Lions

When you really gotta get away from the photogs!

參考翻譯

Ur ANDROID App Ranks 1st?
www.strategy.co
Get your ANDROID App Ranked 1st with our FREE optimization tool!

Capcom卡普空
一起來狩獵吧！
G5大型改版 全面進化 全新古龍種魔物「雅翁龍」勢如破竹來襲！一起來狩獵吧！
身體部位
原來Umbilicus是肚臍！
教你腋下、鬢角等奇怪部位的英文
How To Remove Ads On Web Browser: Clearly
How To Remove Ads On WebBrowser: AdBlock
SAVING YELLO

Save Yello from his out-of-bowl terror!

FREE download!
But...

How To Remove Ads On Android?
Remove Ads On Android: AdAway

AdAway lets you select your own sources of hosts files. You can add exceptions to your Whitelist if an app is not working when specific hosts are blocked.

You can define extra hostnames in your own Blacklist. You can add your own (hostname, IP)-pairs in the Redirection List.

Requirements:
- Android version: 2.1
- ROOT required
- Read/Write access on system-Partition

Problems can occur on some HTC devices when system-Partition is not writable!

Need Root And might not work on HTC phones.
Remove Ads On Android:
AdBlock

若裝置已取得root權限，則Adblock Plus會自動過濾所有網路流量。

On non-rooted devices running Android 4.1.2, 4.2.1 and higher, Adblock Plus will filter all WiFi traffic, but it needs to be configured as a proxy server manually. Detailed instructions are provided by Adblock Plus for not experienced users.

對於未取得root權限及執行Android 3.1以上版本的裝置，Adblock Plus僅會過濾透過WiFi連線的網路流量。對於未取得root權限及執行Android 3.0以下版本的裝置，則需要手動將Adblock Plus設定為一個代理伺服器。部分裝置不支援代理伺服器設定，Adblock Plus自然也就無法在這些裝置上運作。
If I Have An Android Phone
But ... I Don’t Have Root...
Could We Remove Those Ads On Android?
未知的來源
允許從來源安裝的應用程式，而非 Play Store
If we can...

Patched Ads
APK

APK
How Ads Were Displayed
Android M
Developer Preview

Get your apps ready for the next version of Android. Test on Nexus 5, 6, 9, and Player.

› Get started
public final class **AdView** extends **ViewGroup**

java.lang.Object

| ← | android.view.View |
| ← | android.view.ViewGroup |
| ← | com.google.android.gms.ads.AdView |

The **View** to display banner ads. The ad size and ad unit ID must be set prior to calling **loadAd(AdRequest)**.

Publishers using DoubleClick for Publishers or Search Ads for Apps should use **PublisherAdView** or **SearchAdView**, respectively.
AdView Type Ads

// Create a banner ad. The ad size and ad unit ID must be set before calling loadAd.

mAdView = new AdView(this);
mAdView.setAdSize(AdSize.SMART_BANNER);
mAdView.setAdUnitId("myAdUnitId");

// Create an ad request.

AdRequest.Builder adRequestBuilder = new AdRequest.Builder();

// Add the AdView to the view hierarchy.

layout.addView(mAdView);

// Start loading the ad.

mAdView.loadAd(adRequestBuilder.build());
SAVING YELLO

Save Yello from his out-of-bowl terror!

FREE download!
public final class InterstitialAd extends Object

java.lang.Object
→ com.google.android.gms.adsInterstitialAd

Full-screen interstitial ads. The ad unit ID must be set prior to calling loadAd(AdRequest).

Publishers using DoubleClick for Publishers should use PublisherInterstitialAd.
InterstitialAd Type Ads

// If the interstitial is finished loading,
// the user will view the interstitial before proceeding.

mInterstitialAd = new InterstitialAd(this);
mInterstitialAd.setAdUnitId("myAdUnitId");

// Create an ad request.
AdRequest.Builder adRequestBuilder = new AdRequest.Builder();

// Optionally populate the ad request builder.
adRequestBuilder.addTestDevice(AdRequest.DEVICE_ID_EMULATOR);

mInterstitialAd.loadAd(adRequestBuilder.build());
new ADView

setAdSize

loadAd

newInterstitialAd

loadAd
How Android Works
APK Runs → Send Message By Socket → System_Server

- ForkSystemServer()
- Send Execute Request By Socket

Zygote Process → Fork → Zygote Process

ForkAndSpecialize() → Process
Process

Classes.dex

loadClassFromDex()

ClassObject

gDvm.ladedClasses()
dvmVerifyCodeFlow()

Main

FindClass()

All Classes

dvmInterPret()

Working
How APP Works On Android

1. Process Created By Zygote Process
2. APP Code Load By loadClassFromDex() To Load Classes.dex From APK
3. Get All Classes By gDvm.ladedClasses()
4. APK Base On Class Unit
Android Basic Cracking
We know that APK runs on Android this way

APK Package → Check Sign & Extract → Classes.dex
We know that APK runs on Android this way

```
Classes.dex

DexFile.loadDex()

Dalvik VM
```
How We Do Code Review of Classes.dex?
About

smali/baksmali is an assembler/disassembler for the dex format used by dalvik, Android's Java VM implementation. The syntax is loosely based on Jasmin's/dedexer's syntax, and supports the full functionality of the dex format (annotations, debug info, line info, etc.)

The names "smali" and "baksmali" are the Icelandic equivalents of "assembler" and "disassembler" respectively. Why Icelandic you ask? Because dalvik was named for an Icelandic fishing village.

Curious what the smali format looks like? Here's a quick HelloWorld example to whet your appetite.

Got questions/comments? Need help? Come hang out in #smali on freenode.

News

2015-04-30 v2.0.6 is out! Bugfixes, etc.

- Fixes a bug with parameter annotations (credit: Jiri Hrushka)
- Improves the build experience when dx is not on path
- baksmali should be working correctly on windows 8.1 now
- Fixes for the deodexerant makefile (credit: Victor Kaiser-Pendergrast)
- Various other bugfixes, etc. (thanks to Rover12421 for a handful of small fixes)

2015-01-20 v2.0.5 is out! It fixes a multi-threading issue in baksmali in 2.0.4, and it switches the default for implicit references to no-implicit-references, for better backwards compatibility.
Smali Assembly Byte Code

.method protected onCreate(Landroid/os/Activity;)V
REGISTER 8

const/4 v3, 0x0

const/4 v2, 0x1

invoke-super {p0, p1}, Landroid/app/Activity

const v0, 0x7f04001b

invoke-virtual {p0, v0}, Lcom/shen/app/TaskActivity

invoke-direct {p0}, Lcom/shen/app/TaskActivity;->Constructor()V

move-result v0

if-nez v0, :cond_12

:goto_11
return-void

:cond_12
invoke-virtual {p0}, Lcom/shen/app/TaskActivity;->Constructor()V

move-result-object v0

invoke-virtual {v0}, Landroid/content/Activity;->finish()V

:goto_11
return-void

Classes.dex  ->  smali.jar
.Smali Assembly Byte Code

.method protected onCreate(Landroid/c
    .registers 8

    const/4 v3, 0x0

    const/4 v2, 0x1

    invoke-super {p0, p1}, Landroid/c

    const v0, 0x7f04001b

    invoke-virtual {p0, v0}, Lcom/shen/a

    invoke-direct {p0}, Lcom/shen/a

    move-result v0

    if-nez v0, :cond_12

    :goto_11
    return-void

    :cond_12

    invoke-virtual {p0}, Lcom/shen/a

    move-result-object v0

    invoke-virtual {v0}, Landroid/co
But...Smali Bytecode Is Too many Much, Could We Find Important Info In Short Time?
@clkao @ETBlue @evenwu 講個秘訣：講「講個秘訣前，先講講個秘訣，再講秘訣」前，先講「講個秘訣：講『講個秘訣前，先講講個秘訣，再講秘訣』前，先講『講個秘訣』，再講『講個秘訣前，先講講個秘訣，再講秘訣』」，再講「講個秘訣前，先講講個秘訣，再講秘訣」。
Tools to work with android .dex and java .class files

<table>
<thead>
<tr>
<th>Branch</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.x</td>
<td>shutup compiler</td>
<td>a month ago</td>
</tr>
<tr>
<td></td>
<td>support git/hg revision in meta-info</td>
<td>2 months ago</td>
</tr>
<tr>
<td></td>
<td>shutup compiler</td>
<td>a month ago</td>
</tr>
<tr>
<td></td>
<td>fix compiler error by perv commit</td>
<td>15 days ago</td>
</tr>
<tr>
<td></td>
<td>fix</td>
<td>a month ago</td>
</tr>
<tr>
<td></td>
<td>Goodbye googlecode, thanks for the hosting</td>
<td>3 months ago</td>
</tr>
<tr>
<td></td>
<td>fix compiler error in generating asmifier of dex file</td>
<td>a month ago</td>
</tr>
<tr>
<td></td>
<td>fix compiler error by perv commit</td>
<td>15 days ago</td>
</tr>
<tr>
<td></td>
<td>remove println</td>
<td>15 days ago</td>
</tr>
<tr>
<td></td>
<td>use antlr4 to parse smali files</td>
<td>15 days ago</td>
</tr>
</tbody>
</table>
```java
private boolean b(Context paramContext, AdSize paramAdSize, AttributeSet paramAttributeSet, String paramString)
```

```java
private Set<String> c(String paramString, Context paramContext, AttributeSet paramAttributeSet, String paramString1)
```

```
j = (str1.equals("null"))
```
new ADView

loadAd

newInterstitialAd

loadAd

Ads Display
public void loadAd(AdRequest paramAdRequest) {
    if (this.a == null)
        return;
    if (isRefreshing())
        this.a.e();
    this.a.a(paramAdRequest);
}
Prevent loadAd()

1. Find Out All Codes Calling loadAd(), And Patching Those.

2. Clear All Code Of loadAd(), Let It Be An Empty Function.

So.. We Can... do it this way
public void loadAd(AdRequest paramAdRequest)
{
}

.method public FuncName( ... path... )V/Z/B/S/I/C/J/F/D
    .registers VariableCount
    .param p1, "adRequest"  #Lcom/google/ads/AdRequest;
    .prologue

        return-void
    .end method
.method public loadAd(Lcom/google/ads/AdRequest;)V
    .registers 3
    .param pl, "adRequest"  # Lcom/google/ads/AdRequest;
    
    return-void
.end method
Demo : Patch Ads
Ads In Unity
Unity Works On Android

APK → Class.dex → Smali ByteCode → Dalvik VM

- Assembly-CSharp.dll
- Assembly-CSharp-FirstPass.dll
- UnityEngine.dll
- mscorelib.dll

API In Classes.dex
理想

現実
ProGuard Obfuscation
Android Basic Cracking Is Easy And Fun, Until...

ProGuard
1+1 = 2

*Source Code Do = What Will Happen*
1 + 1 + 1 - 1 = 2
\[ 2 \times (1+1-7) \times (-1)/10 \times (4\gg 1) = 2 \]

*Source Code Do = What Will Happen (ProGuard Do Obfuscation)*
public void loadAd(AdRequest paramAdRequest)
{
    if (this.a == null)
        return;
    if (isRefreshing())
        this.a.e();
    this.a.a(paramAdRequest);
}
Fucking Code Review
public final void a(ad paramad)
{
    try
    {
        if (this.d == null)
        {
            if (((this.e == null) || (this.f == null)) && (this.d == null))
                throw new IllegalStateException("The ad size and ad unit ID must be set before loadAd is called.");
            Context localContext = this.g.getContext();
            this.d = M.a(localContext, new ay(localContext, this.e), this.f, this.a);
            if (this.c != null)
                this.d.a(new L(this.c));
            e();
        } else if (this.d.a(0.a(this.g.getContext(), paramad)))
            return;
            this.a.a(paramad.i());
            return;
    }
    catch (RemoteException localRemoteException)
    {
    }
}
public final void a(ad paramad)
{
    try
    {
        if (this.e == null)
        {
            if (this.f == null)
            { b("loadAd");
                this.e = M.a(this.b, new av(), this.f, this.a);
                if (this.d != null)
                { this.e.a(new l(this.d));
                }
            }
            if (this.e.a(0.a(this.b, paramad)))
            { this.a.a(paramad.i());
                return;
            }
        }
    }
    catch (RemoteException localRemoteException)
    {
    }
}
ProGuard Obfuscation Property

1. Type Name Will Keep Intact.
2. Only Confuse The Functions Used In App.
3. Add A New Type Named “Internal”.
4. Functions Will Be Translated Into Lots Of Pieces Put In Internal Type.
5. Would Not Confuse String In Function.
How To Analytics Code In ProGuard

How To Analytics Code In ProGuard

2. Review Obfuscation Codes.
How To Analytics Code In ProGuard

3. De-Obfuscation (If Possible?)
How To Build AIDS Tool In C#
How To Clear Ads In APK.
Protected By ProGuard Obfuscation Or Not

1. Extract Classes.dex From Ads
2. Translate Classes.dex To Smali Asm Code
3. Find The loadAd() And Empty It (If Protected By ProGuard, Need To Check All Functions In Intent Type)
4. Translate Smali Asm Code Back To .Dex
5. Package All Data To ZIP Package
6. Sign The ZIP And Name It As .APK
SharpCompress is a fully managed C# library to deal with many compression types and formats.
http://sharpcompress.codeplex.com
About

smali/baksmali is an assembler/disassembler for the dex format used by dalvik, Android's Java VM implementation. The syntax is loosely based on Jasmin's/dedexer's syntax, and supports the full functionality of the dex format (annotations, debug info, line info, etc.)

The names "smali" and "baksmali" are the Icelandic equivalents of "assembler" and "disassembler" respectively. Why Icelandic you ask? Because dalvik was named for an Icelandic fishing village.

Curious what the smali format looks like? Here's a quick HelloWorld example to whet your appetite.

Got questions/comments? Need help? Come hang out in #smali on freenode.

News

2015-04-30 v2.0.6 is out! Bugfixes, etc.

- Fixes a bug with parameter annotations (credit: Jiri Hrushka)
- Improves the build experience when dx is not on path
- baksmali should be working correctly on windows 8.1 now
- Fixes for the deodexerant makefile (credit: Victor Kaiser-Pendergrast)
- Various other bugfixes, etc. (thanks to Rover12421 for a handful of small fixes)

2015-01-20 v2.0.5 is out! It fixes a multi-threading issue in baksmali in 2.0.4, and it switches the default for implicit references to no-implicit-references, for better backwards compatibility.
Usage: signapk (options) [command] (files)

commands:
sign FILE  sign a .zip or .apk
sign FILE1 FILE2 create a signed copy of FILE1 as FILE2
cert FILE(s) print cert info on FILE's signer
certinfo FILE print detailed cert info on FILE's signer
cmp FILE compare signer of FILE with default/selected cert
cmp FILE1 FILE2 compare signer of FILE1 to signer of FILE2

options:
-k, --key FILE     key to sign with
-c, --cert FILE    to sign with
testkey or --c      if -c or -k are not files then they are considered
platform aliases to builtins (ie -k testkey or -c platform)
-f, --force        sign even if cert differs from original
-t, --tmp DIR      use DIR for tempdir instead of '/cache'
-d, --debug        output debugging
-V, --version      print 'signapk v0.3.1'

exit codes:
1: read error (file 1)  2: read error (file 2)
3: write error          4: ssl error
5: zip write error      9: key error
8: sign: cert mismatch  10: cmp: cert mismatch
128: script error       255: user error
private void GetFiles(DirectoryInfo di, string searchPattern, ref ArrayList MyFiles)
{
    try
    {
        foreach (FileInfo fi in di.GetFiles(searchPattern))
        {
            MyFiles.Add(fi.FullName);
        }
        foreach (DirectoryInfo d in di.GetDirectories())
        {
            GetFiles(d, searchPattern, ref MyFiles);
        }
    }
    catch (Exception)
    {
    }
}
int DeepFuckFuncContentText(string SmaliPath, string ContentText)
{
    string Source = (File.ReadAllText(SmaliPath));
    int FuckStrPos = Source.IndexOf(ContentText);
    if (FuckStrPos < 0) return (-1);

    int AdsFuncStartPos = Source.LastIndexOf(".method", FuckStrPos);
    int AdsFuncEndPos = Source.IndexOf(".end method", FuckStrPos);
    string AdsSource = (Source.Substring(AdsFuncStartPos, AdsFuncEndPos - AdsFuncStartPos));

    string PatchedBannerSource = "";
    foreach (var item in AdsSource.Split('\n'))
    {
        if (item.Length < 1) continue;
        if (new Regex("@"."\.*\".method\.*\""").IsMatch(item)) PatchedBannerSource += item + "\r\n";
        if (new Regex("@"."\.*\".registers\.*\""").IsMatch(item)) PatchedBannerSource += item + "\r\n";
        if (new Regex("@"."\.*\".prologue\.*\""").IsMatch(item)) PatchedBannerSource += item + "\r\n";
        if (new Regex("@"."\.*\"const\/?\.*\""").IsMatch(item)) PatchedBannerSource += item + "\r\n";
        if (new Regex("@"."\.*\"return\(-void\|\*\)\.*\""").IsMatch(item)) PatchedBannerSource += item + "\r\n";
    }
    Source = Source.Remove(AdsFuncStartPos, AdsFuncEndPos - AdsFuncStartPos);
    Source = Source.Insert(AdsFuncStartPos, PatchedBannerSource);
    File.WriteAllText(SmaliPath, Source);
    return AdsFuncStartPos;
}
一鍵全自動APK去廣告神器 By.aaaddress1.

Input APK: C:\Users\aaaddress1\Desktop\AutoPatchAds\RAR\RAR\bin\Debug\南臺學生助理1.0.apk
Output APK: C:\Users\aaaddress1\Desktop\AutoPatchAds\RAR\RAR\bin\Debug\南臺學生助理1.0_ADS_Fuck.apk

- 深度逆向去除ProGuard混淆爆破分析
- 生產去廣告版本

開始初始化工作.
開始解析APK.
開始逆向APK回Smali.
開始找尋所有可逆向分析之結構.
開始逆向分析...
- [受混淆保護廣告] outcom\google\android\gms\internal\af2.smali patched at 7733
- [受混淆保護廣告] outcom\google\android\gms\internal\ag2.smali patched at 4216
- [受混淆保護廣告] outcom\google\android\gms\internal\Id.E.smali patched at 1280
- [受混淆保護廣告] outcom\google\android\gms\internal\ftl.smali patched at 82585

一共發現了4處程式碼含有廣告(程式已修補完成).
正在將修改點包回程式核心.
正在將APK包回原始軟體壓縮包.
正在替APK重新簽章.
完成!
Demo AIDS
Hello!

aaaddress1(ADR)

Q&A

aaaddress1@gmail.com