Bezpečnost OS Android

Bezpečnost instalačních APK balíčků OS Android

Zdeněk Říha zriha@fi.muni.cz
Fakulta informatiky, Masarykova univerzita

CROCS
Centre for Research on Cryptography and Security

Od jednoduchých zařízení ke smartphonům

MOBILNÍ TELEFONY

Mobilní telefony kdysi

Mobilní telefony o něco později

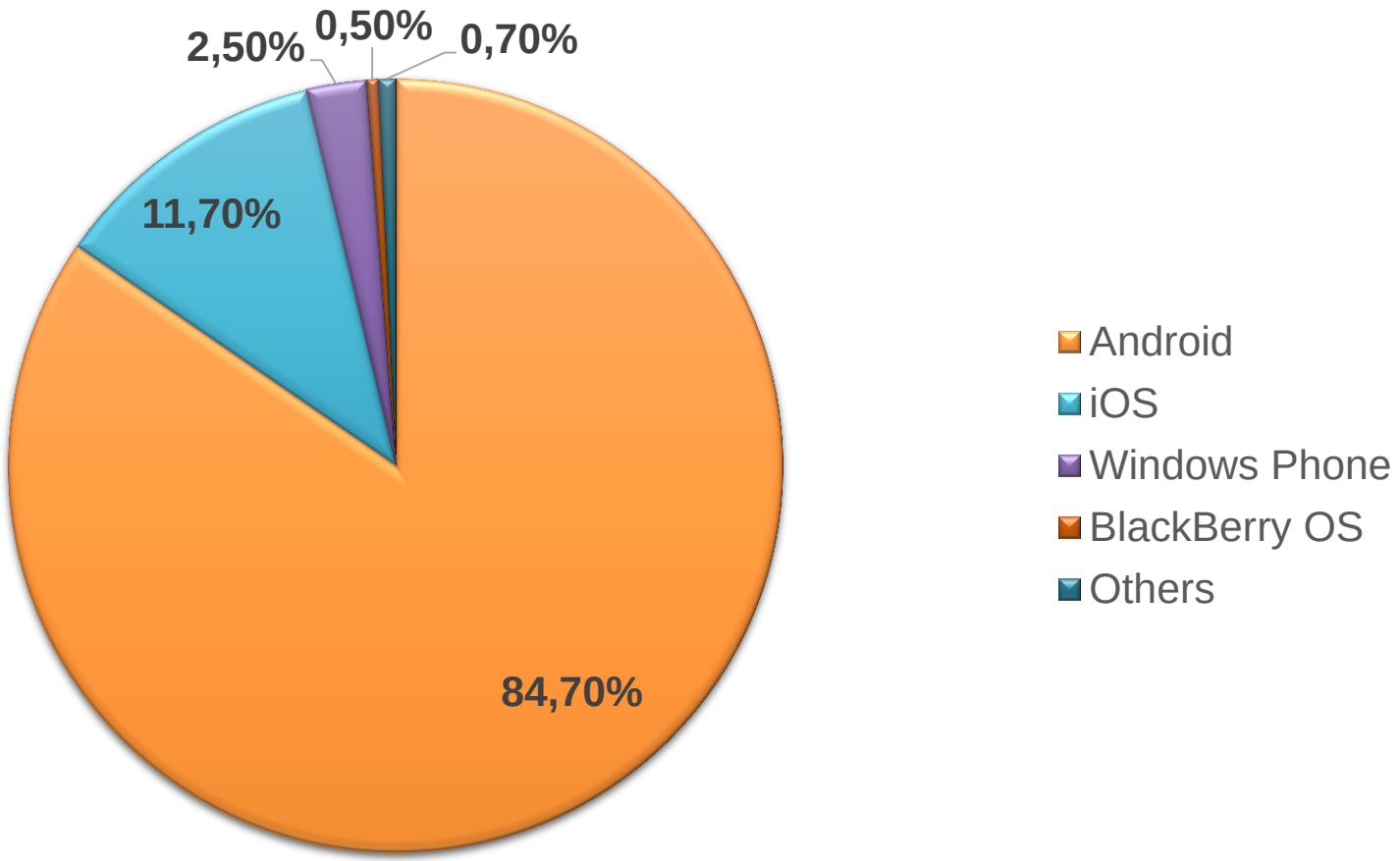
Zdroj: http://outloud93.files.wordpress.com/2010/02/dkmb86g_490ft6648cr_b.jpg

Mobilní telefony ještě o něco později

Zdroj: http://m.phonegg.com/8/859b.jpg

Mobilní telefony dnes

Podíl OS na trhu smartphonů (Q2 2014)



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

Nejrozšířenější mobilní OS

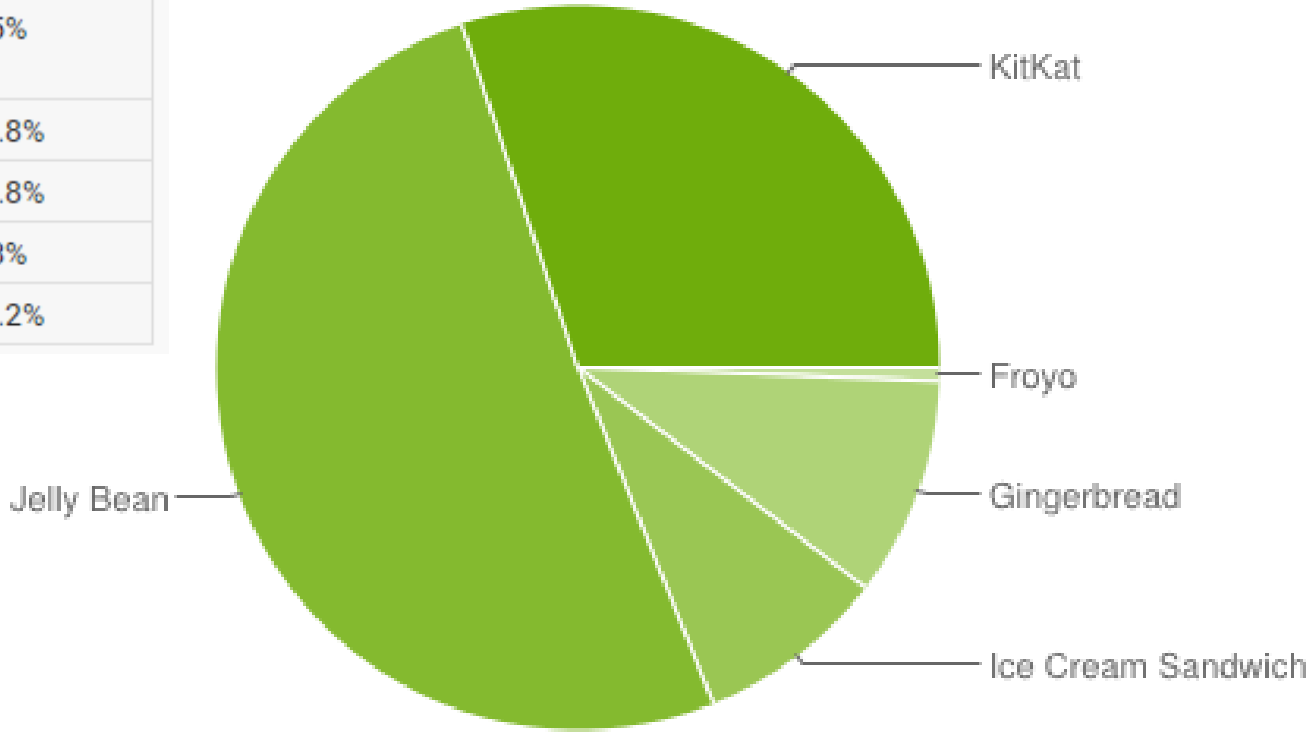
ANDROID

Stručná historie Androidu

- V roce 2003 založeno Android Inc. v Palo Alto
  - Snaha o OS pro digitální foťáky
  - Konkurence pro Symbian a Windows Mobile
- V roce 2005 Android, Inc. kupuje Google
- V roce 2007 Open Handset Alliance (Google + výrobci mobilů jako HTC, Sony, Samsung)
- První verze OS Android k dispozici v roce 2007
- První mobil s OS Android (HTC Dream) k dispozici v roce 2008
- Android 5.0 (Lollipop) vydán v listopadu 2014

Není Android jako Android

Version	Codename	API	Distribution
2.2	Froyo	8	0.6%
2.3.3 - 2.3.7	Gingerbread	10	9.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	8.5%
4.1.x	Jelly Bean	16	22.8%
4.2.x		17	20.8%
4.3		18	7.3%
4.4	KitKat	19	30.2%



Základní vlastnosti OS Android

- Založeno na jádře OS Linux
- Co aplikace to jiné UID (user ID)

- Aplikace psané v jazyce Java, (Java bytecode)
- JVM nazývaná Dalvik s JIT kompilací

- Android Native Development Kit (NDK)
- Aplikace v C/C++

Licence OS Android

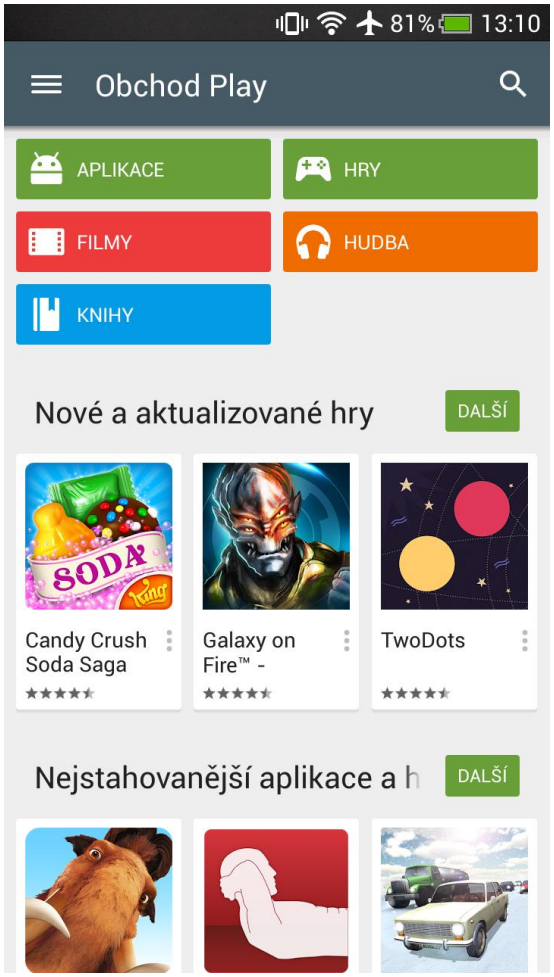
- OS Android je vyvíjen společností Google
- Zdrojový kód zveřejněn až uvolněním nové verze
- Většina kódu má licenci Apache Licence version 2.0
- Jádro si drží licenci GNU GPL v2

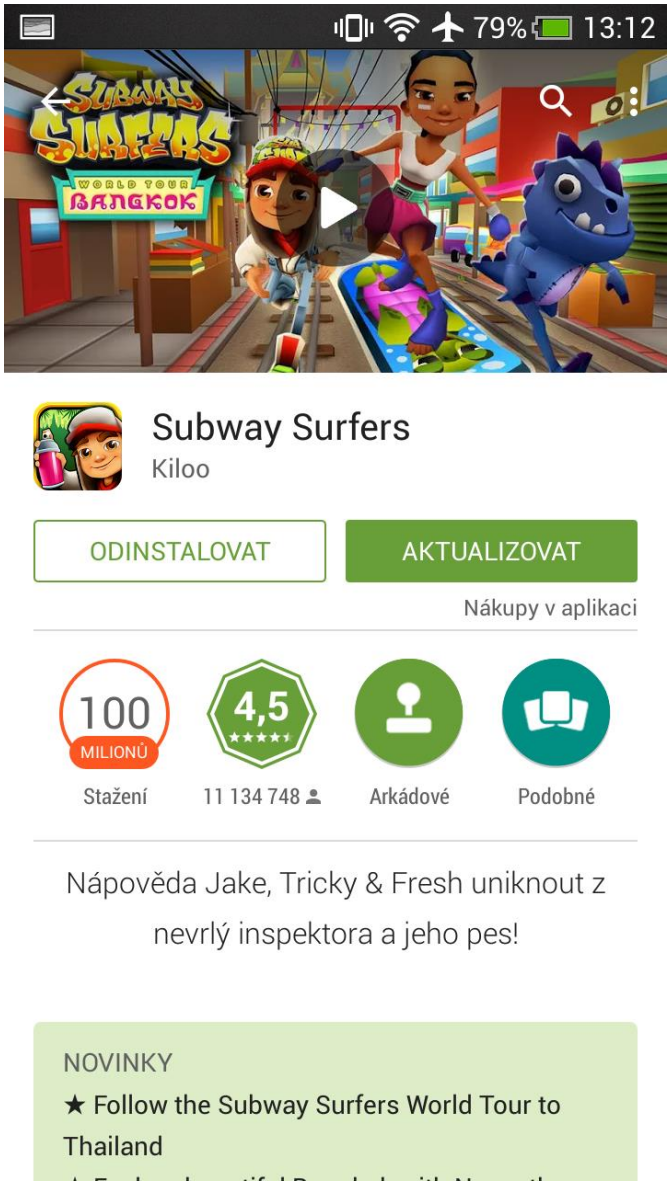
- Logo “Android” je licencováno Googlem
- Google Mobile Services jsou proprietární SW
  - Včetně Google Play Store
- Hromada dodatečných podmínek (HW kompatibilita)

Kde získat aplikace pro OS Android

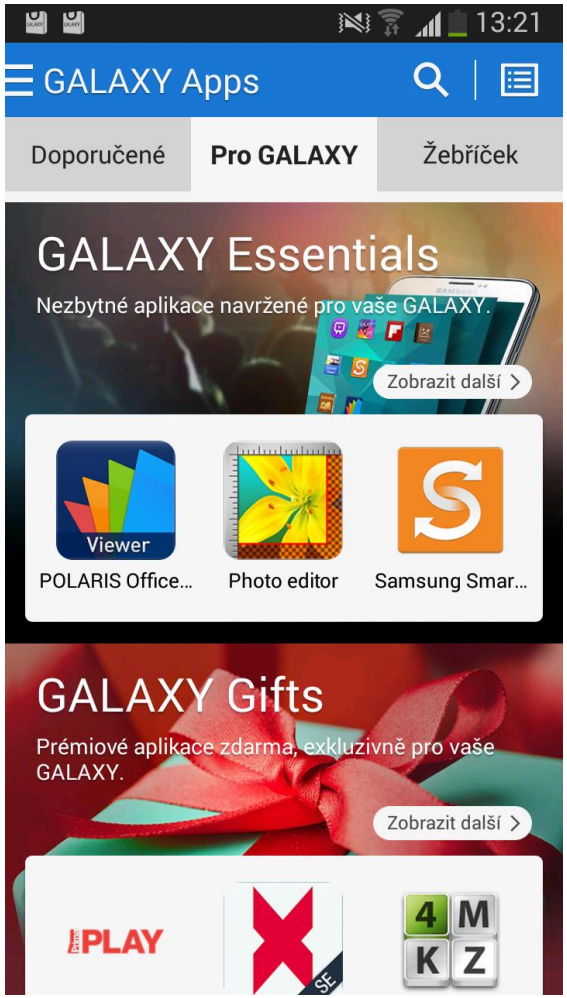
- Google Play (Store, Music, Movies & TV, Books)
  - Vznikla v roce 2012 spojením Google Music a Android Market
- Alternativní obchody
  - Amazon Appstore
  - SlideME
  - 1Mobile Market
  - Samsung Galaxy Apps
  - Mobile9
  - …

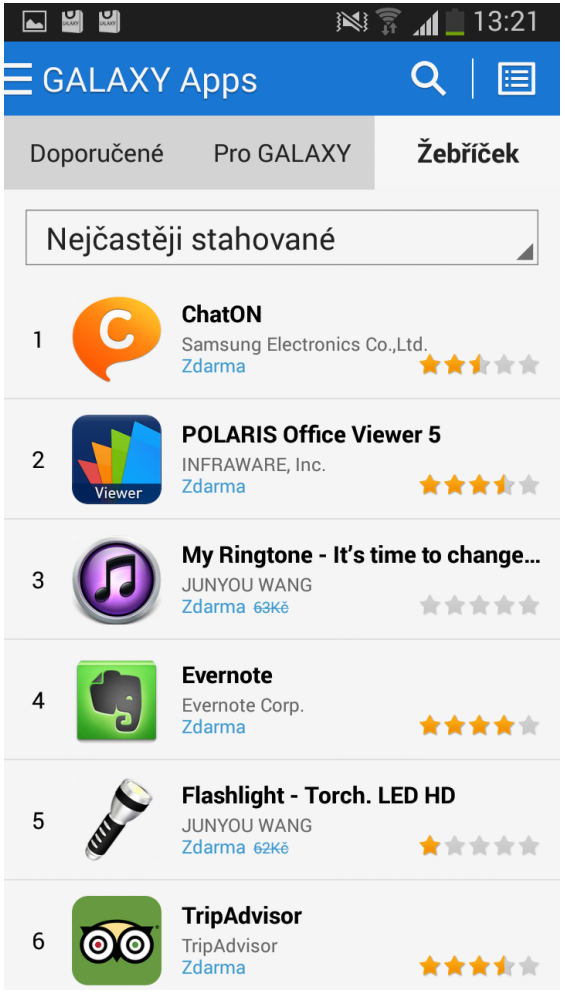
Google Play Store

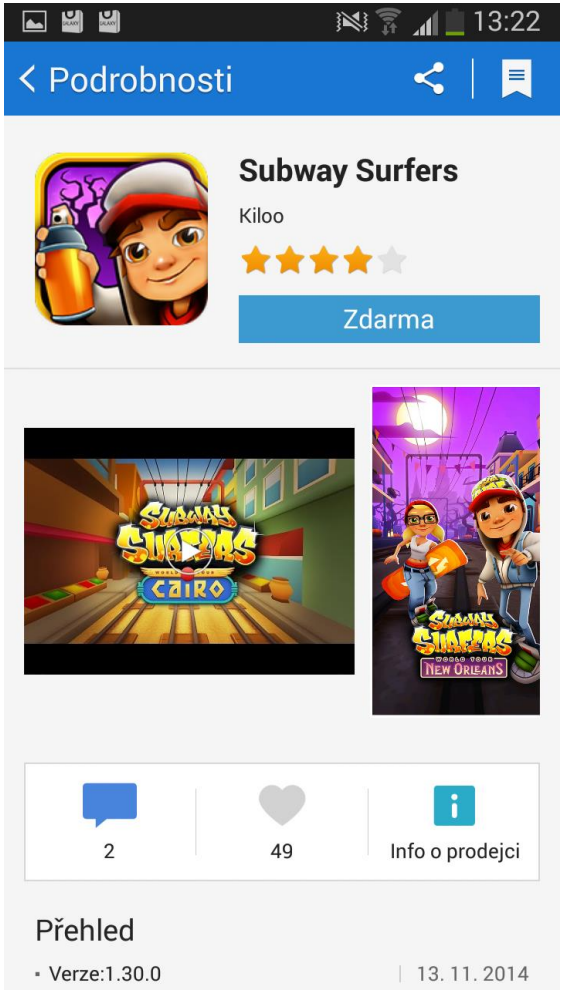




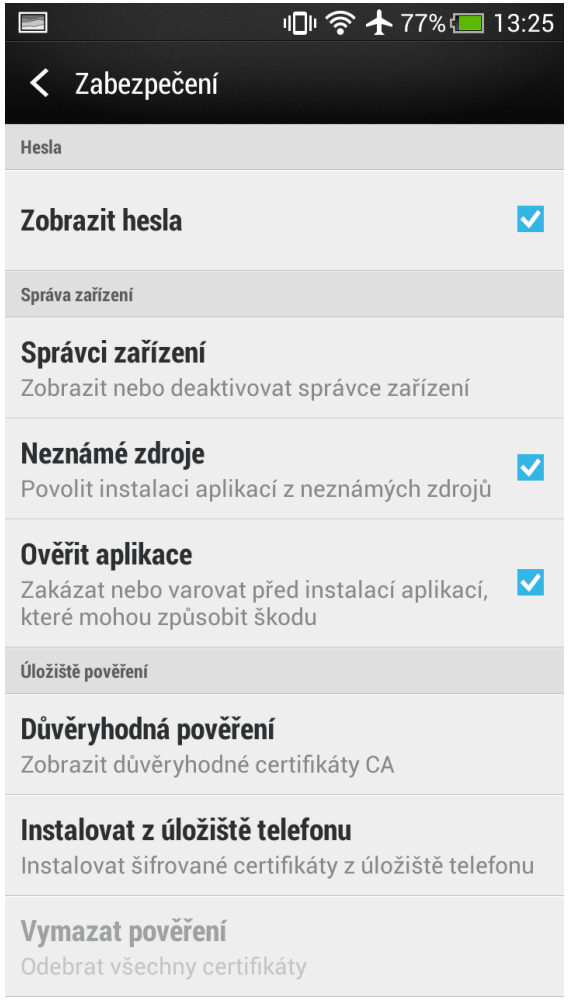
Samsung Galaxy App

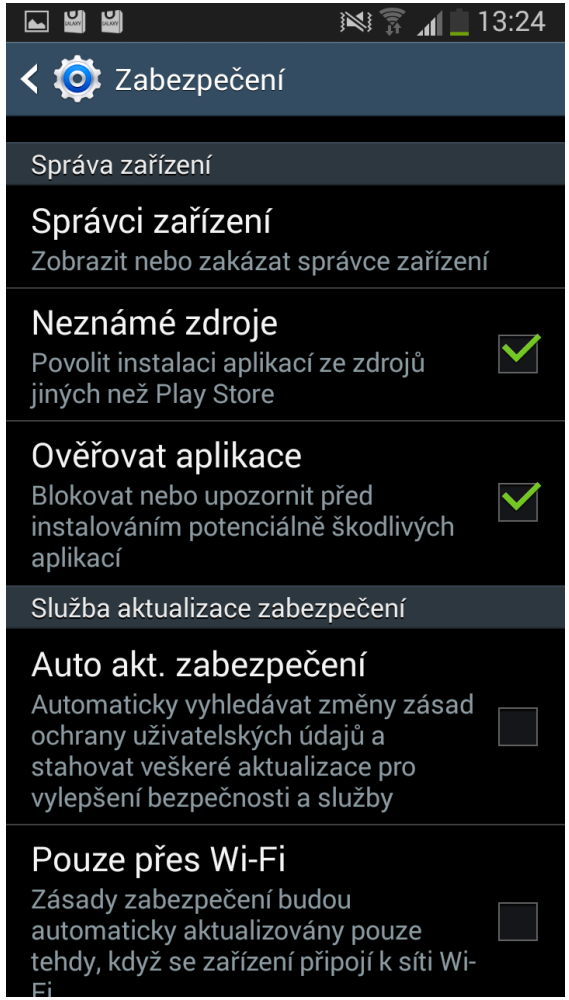




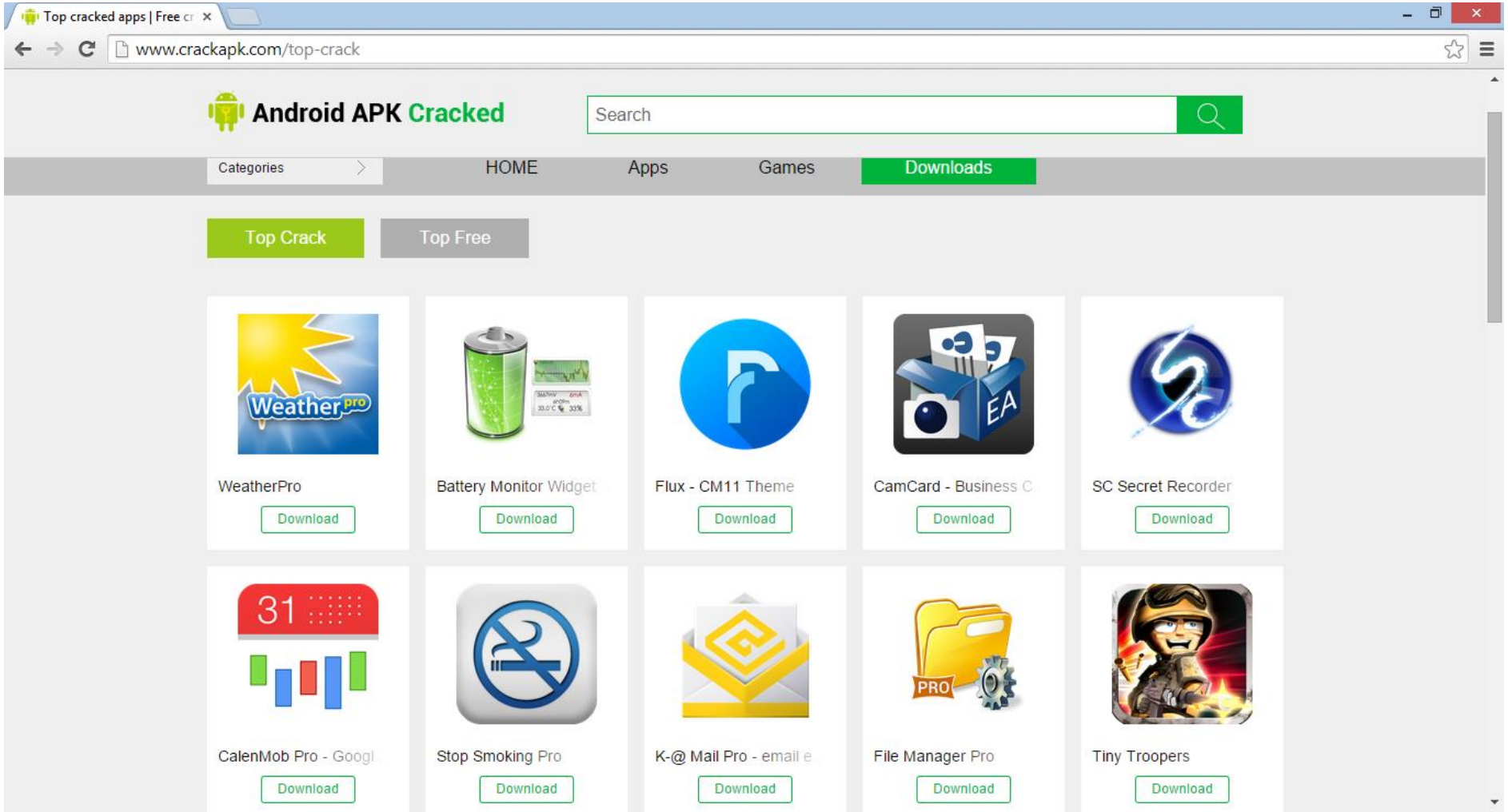


Instalace aplikací z alternativních zdrojů





Jiné alternativní zdroje



Proč alternativní zdroje

- Proč někdy nestačí Google Play Store?
- Google omezuje a kontroluje obsah obchodu:
  - Malware (kdo by jej chtěl ☺)
  - Nelegální obsah
    - Pirátské kopie aplikací a jiných typů dat
    - Špionážní aplikace (spying SW)
  - Obsah nevhodný pro Google
    - Např. Google blokuje vše, co odstraňuje reklamu

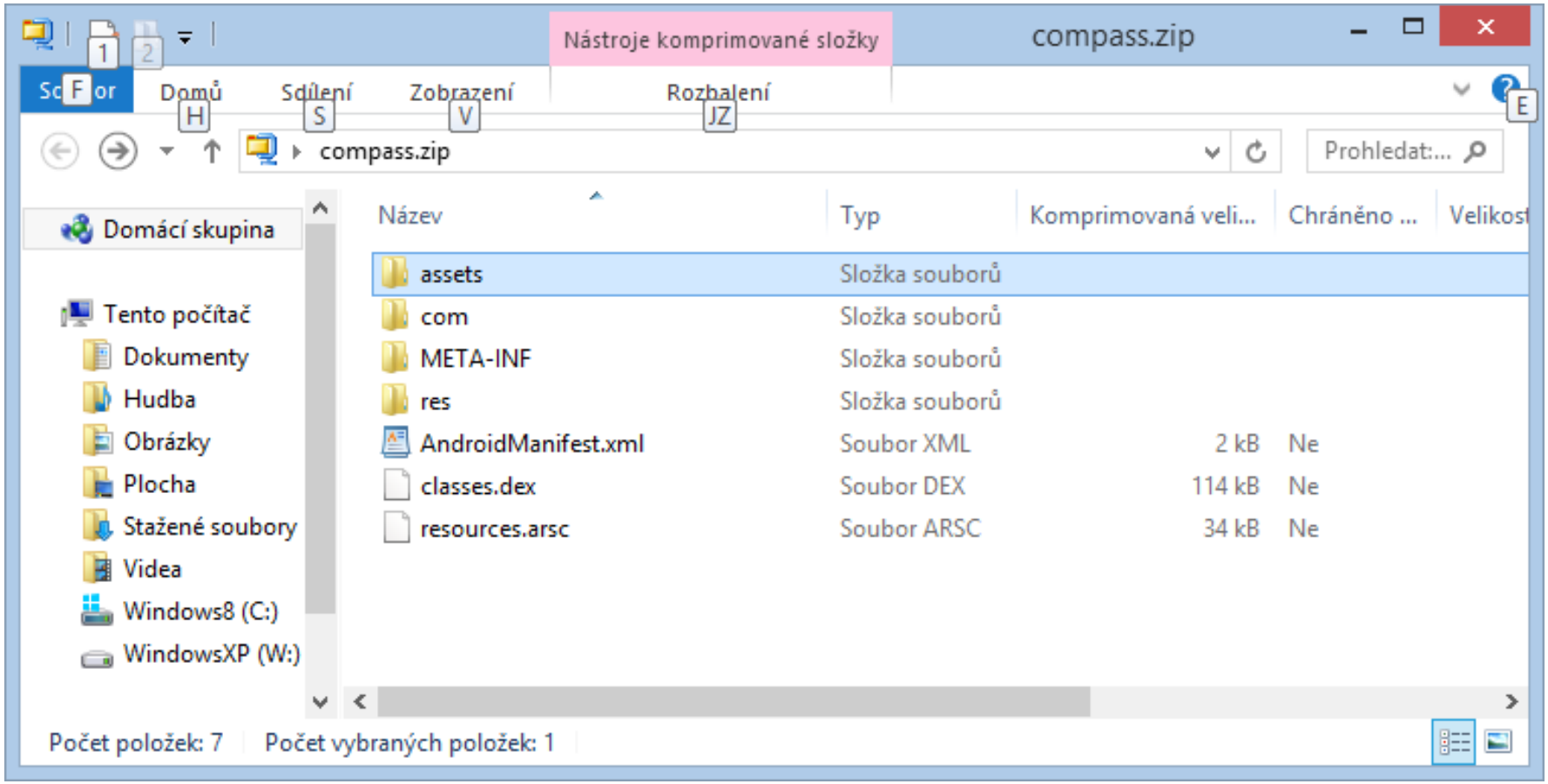
Modifikace instalačních balíků

APK SOUBORY

Instalační APK soubory

- Umožní nainstalovat aplikaci na OS Android
- Něco jako:
  - setup.exe nebo *.msi soubory v MS Windows
  - *.rpm soubory v Fedora Linuxu
- Obsahují programové instrukce a další data
  - ikony
  - obrázky
  - texty
  - zvuky

APK = ZIP soubor

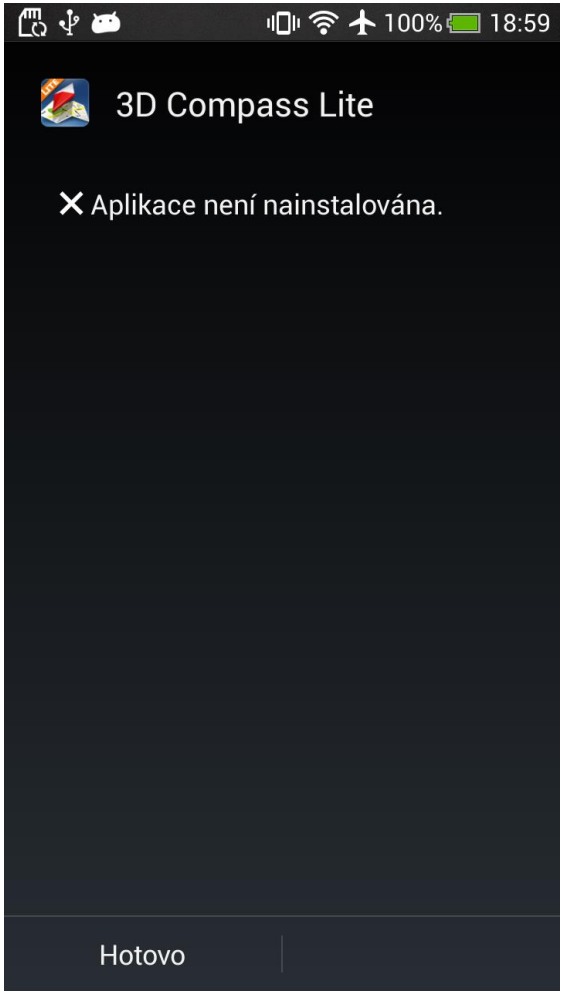


Struktura APK souboru

- **classes.dex** – zkompilovaný zdrojový kód (bytecode pro Dalvik Virtual Machine)
- **resources.arsc** – předkompilované zdroje (např. XML soubory)
- **AndroidManifest.xml** – přehledový metadata soubor celého balíku (verze, práva, vazby)
- **res** – adresář s nekompilovanými zdroji (obrázky)
- **assets** – další zdroje
- **lib** – kompilovaný platformě závislý kód (nativní)
- **META-INF** – digitální podpis balíku (manifest, certifikát, přehled hashů souborů)

Digitální podpis APK souboru

- Zajišťuje integritu souborů v balíku
- Při změně obsahu některého souboru digitální podpis nesedí
- OS Android toto pozná a instalaci neprovede
- APK soubor však můžeme podepsat klíčem libovolným!!!
- A toho lze využít k útokům



Modifikace APK souboru – Proč?

- Protože u digitálního podpisu se nekontroluje, kdo APK soubor podepsal, ale pouze, zda popis, je formálně v pořádku, lze APK soubory modifikovat a výsledek někomu podstrčit
- Proč?
- Malware
  - Spyware (umístění, SMS)
  - Botnet klient
  - Jiné ekonomické výhody (např. bitcoin mining)

Modifikace APK souboru – Jak?

- ZIP soubor dekomprimujeme
- Dekompilujeme classes.dex
- Získáme tak zdrojový kód ve formátu SMALI
  – ne program v Javě!
- Zdrojový program ve SMALI modifikujeme
- Můžeme modifikovat i jiné zdroje
  – Ikonky, texty, atd.
- Opět assemblujeme SMALI do classes.dex
- Digitálně podepíšeme a zabalíme ZIPem

Modifikace APK souboru – Detaily

- Nástroj apktool
- Parametr „d“ (decode)
  - Rozbalí ZIP, dekóduje binární XML, dekóduje classes.dex

```
[root@test resources]# java -jar apktool.jar d /tmp/z/compass.apk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/apktool/framework/1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...
```

Modifikace APK souboru – Detaily

- Modifikace smali zdrojových souborů

root@test:/tmp/z/hello_world/src/main/resources/compass/smali

```
# instance fields
.field private a:Ljava/lang/ref/WeakReference;
    .annotation system Ldalvik/annotation/Signature;
        value = {
            "Ljava/lang/ref/WeakReference",
            "<",
            "Ld;",
            ">;"
        }
    .end annotation
.end field


# direct methods
.method public constructor <init>(Ld;)V
    .locals 1

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    new-instance v0, Ljava/lang/ref/WeakReference;
                                                  8,1           16%
```

Modifikace APK souboru – Detaily

- Modifikujeme AndroidManifest.xml
- Přidáme odkazy na náš vložený kód
  - Automatické spuštění (neviditelné) služby
  - Nové aktivity (uživatelské rozhraní)
  - Vazba na určité události
    - Např. přijetí SMS zprávy
- Přidáme oprávnění, je-li třeba
  - Tj. pokud je aplikace již nepoužívá/neuvádí
  - Např. pro komunikaci s Internetem, čtení SMS zpráv

root@test:/tmp/z/hello_world/src/main/resources/compass

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:versionCode="352" android:versionName="3.52" android:installLocation="auto" package="com.a0soft.gphone.
aCompass"
  xmlns:android="http://schemas.android.com/apk/res/android">
    <supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScr
eens="true" android:xlargeScreens="true" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false" />
    <uses-feature android:name="android.hardware.location" />
    <uses-feature android:name="android.hardware.sensor.accelerometer" />
    <uses-feature android:name="android.hardware.sensor.compass" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <application android:theme="@*android:style/Theme.NoTitleBar.Fullscreen" android:label="@string/app_name" android:ico
n="@drawable/icon_free" android:name="com.a0soft.gphone.aCompass.MainApp" android:hardwareAccelerated="true">
        <uses-library android:name="com.google.android.maps" />
        <meta-data android:name="com.a0soft.gphone.aTrackDog.webURL" android:value="http://android.a0soft.com/?url=3DComp
ass.htm" />
        <activity android:name="com.a0soft.gphone.aCompass.MainWnd" android:screenOrientation="landscape" android:configC
hanges="keyboardHidden|orientation">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="com.a0soft.gphone.aCompass.ArWnd" android:screenOrientation="landscape" android:configCha
nges="keyboardHidden|orientation" />
        <activity android:name="com.a0soft.gphone.aCompass.screenshot.ShareScreenshotWnd" android:screenOrientation="land
scape" android:configChanges="keyboardHidden|orientation" />
        <activity android:name="com.a0soft.gphone.aCompass.PrefWnd" android:screenOrientation="landscape" android:configC
hanges="keyboardHidden|orientation" />
        <activity android:name="com.a0soft.gphone.aCompass.About" android:screenOrientation="landscape" android:configCha
nges="keyboardHidden|orientation" />
        <activity android:name="com.google.ads.AdActivity" android:configChanges="keyboard|keyboardHidden|orientation" />
    </application>
</manifest>
                                                                                1,1           All
```

Bezpečnostní prvky OS Android

- Linuxové jádro je multiuživatelské
  - V OS Android co aplikace to uživatel (UID)
- Linuxové jádro dobře izoluje jednotlivé procesy
  - Izoluje adresové prostory jednotlivých procesů a izoluje uživatelský proces od (struktur) jádra
- Další bezpečnostní prvky zajišťuje HW
  - Ochrana paměti
  - Režimy procesoru (CPU)

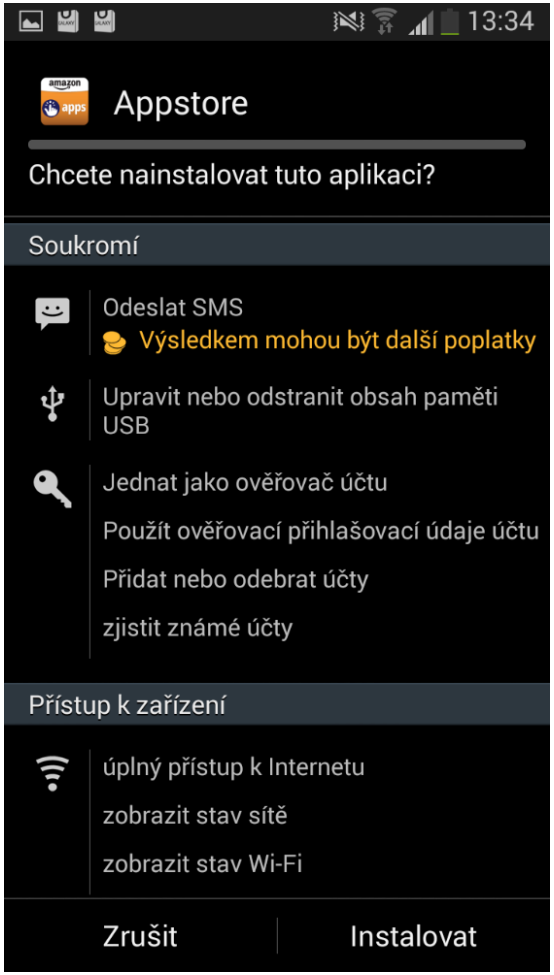
Bezpečnostní prvky OS Android

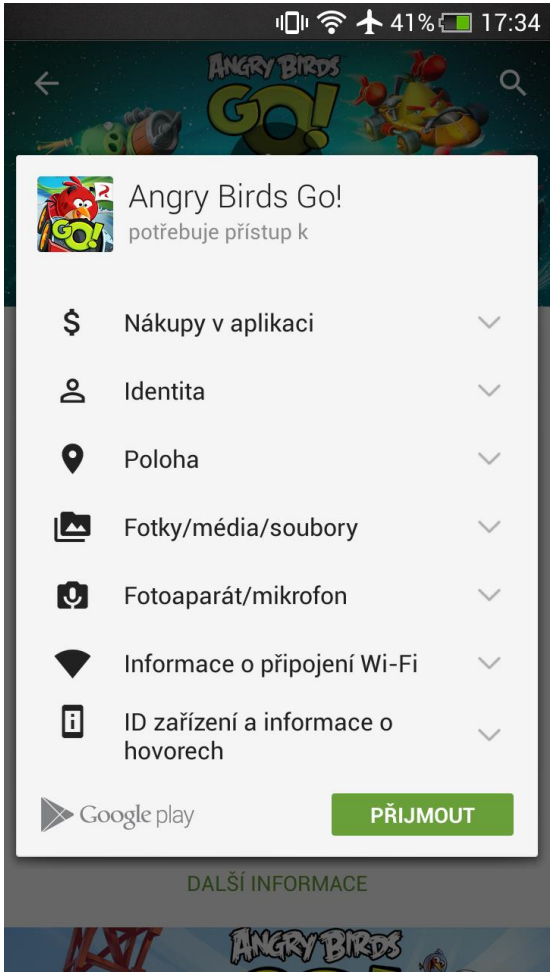
- Izolace procesů je důležitá
- Stejně důležité je však v určitých případech komunikaci/synchronizaci povolit
- Systémová volání (jádra)
- InterProcess Communication (IPC)
  - Semafory
  - Roury
  - …
- Systémové procesy poskytují užitečné služby
  - Pro možnost přístupu k určitým službám je třeba povolení

Oprávnění aplikací (Permissions)

- Oprávnění, která aplikace potřebuje jsou vypsána v AndroidManifest.xml
- Při instalaci aplikace se uživateli zobrazí seznam vyžadovaných oprávnění
  - Pokud uživatel souhlasí, pak se aplikace nainstaluje
  - Pokud uživatel nesouhlasí, aplikace se nenainstaluje
- Při použití povolených oprávnění se nic nezobrazuje
- Při pokusu o použití nepovoleného oprávnění dojde k chybě a vyvolá se výjimka SecurityException.

Oprávnění aplikací (Permissions)





Oprávnění aplikací (Permissions)

String	PROCESS_OUTGOING_CALLS	Allows an application to see the number being dialed during an outgoing call with the option to redirect the call to a different number or abort the call altogether.
String	READ_CALENDAR	Allows an application to read the user's calendar data.
String	READ_CALL_LOG	Allows an application to read the user's call log.
String	READ_CONTACTS	Allows an application to read the user's contacts data.
String	READ_EXTERNAL_STORAGE	Allows an application to read from external storage.
String	READ_FRAME_BUFFER	Allows an application to take screen shots and more generally get access to the frame buffer data.
String	READ_HISTORY_BOOKMARKS	Allows an application to read (but not write) the user's browsing history and bookmarks.
String	READ_INPUT_STATE	*This constant was deprecated in API level 16. The API that used this permission has been removed.*
String	READ_LOGS	Allows an application to read the low-level system log files.
String	READ_PHONE_STATE	Allows read only access to phone state.
String	READ_PROFILE	Allows an application to read the user's personal profile data.
String	READ_SMS	Allows an application to read SMS messages.
String	READ_SOCIAL_STREAM	*This constant was deprecated in API level 21. This functionality will be unsupported in the future; cursors returned will be empty. Please do not use.*

Modifikace APK souboru – Detaily (opak.)

- Modifikujeme AndroidManifest.xml
- Přidáme odkazy na náš vložený kód
  - Automatické spuštění (neviditelné) služby
  - Nové aktivity (uživatelské rozhraní)
  - Vazba na určité události
    - Např. přijetí SMS zprávy
- Přidáme oprávnění, je-li třeba
  - Tj. pokud je aplikace již nepoužívá/neuvádí
  - Např. pro komunikaci s Internetem, čtení SMS zpráv

Vytvoření nového APK souboru

- Nejprve nástroj apktool s parametrem “b”
- Vytvoříme nepodepsaný APK soubor
- Kompilujeme zdroje
- Kompilujeme smali kód

```
[root@test resources]# java -jar apktool.jar b compass c.apk
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
```

Podepsání APK souboru

- Nástroj jarsigner
- Podepíšeme našim klíčem, vložíme náš certifikát

```
[root@test resources]# jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -k
eystore my.keystore c.apk test
Enter Passphrase for keystore:
   adding: META-INF/MANIFEST.MF
   adding: META-INF/TEST.SF
   adding: META-INF/TEST.RSA
  signing: assets/zh/about.htm
  signing: assets/zh/desc.htm
  signing: assets/zh/whats_new.htm
  signing: assets/about.htm
  signing: assets/desc.htm
  signing: assets/test_ad.png
  signing: assets/whats_new.htm
  signing: res/drawable/bright_btn.xml
  signing: res/drawable/bright_off_normal.png
  signing: res/drawable/bright_off_pressed.png
  signing: res/drawable/bright_on_normal.png
  signing: res/drawable/bright_on_pressed.png
  signing: res/drawable/btn_zoom_in.xml
  signing: res/drawable/btn_zoom_in_disabled.png
```

zipalign

- Nepovinný krok
- Zarovnání ZIP souboru, aby data začínala na adresách dělitelných 4 (zarovnání na 32 bitů)
- Efektivita při instalaci

```
[root@test resources]# ./zipalign 4 c.apk c_aligned.apk
[root@test resources]# ls -la *.apk
-rw-r--r--. 1 root root 305074 Nov 17 10:59 c_aligned.apk
-rw-r--r--. 1 root root 305023 Nov 17 10:48 c.apk
[root@test resources]#
```

Výsledek

- Jako výsledek máme správně podepsaný APK soubor
- Už stačí jen soubor někomu podstrčit
  - Přímo
  - Nahrajeme na nějaké úložiště (např. ulož.to)
    - Nejlépe udělat hromadně pro mnoho APK souborů
    - Automatizace vložení dodatečného kódu
  - Online modifikace při stahování libovolného APK souboru přes nezabezpečené spojení
    - Tak aby si toho uživatelé nevšimli

Automatizované vkládání kódu do libovolného APK souboru

AUTOMATIZACE

Automatizace celého procesu

- Získáme APK soubor
- Dekódujeme (apktool d)
- Modifikujeme
  - Přidáme vlastní zdrojové kódy
  - Modifikujeme existující zdrojové kódy resp. přímo AndroidManifest.xml aby spouštěl náš kód
  - Přidáme případně oprávnění do AndroidManifest.xml (nikdo nesleduje)
- Kompilujeme (apktool b)
- Podepisujeme
- Zarovnáme
- Máme APK soubor s modifikovanou funkčností

Automatizace modifikace

- Celý tento proces je relativně rychlý
- Na běžném počítači
  - při zcela automatizovaném provedení skriptem
  - asi 30s až 5 minut
- Nejprve potřebujeme celý originální APK soubor
  - Abychom jej mohli zpracovat apktoolem
- Pro scénář ukládání řady modifikovaných APK souborů na nějaké úložiště plně dostačující
  - Doba zpracování je víceméně nepodstatná
  - Důležitá je plná automatizace

Automatizace modifikace

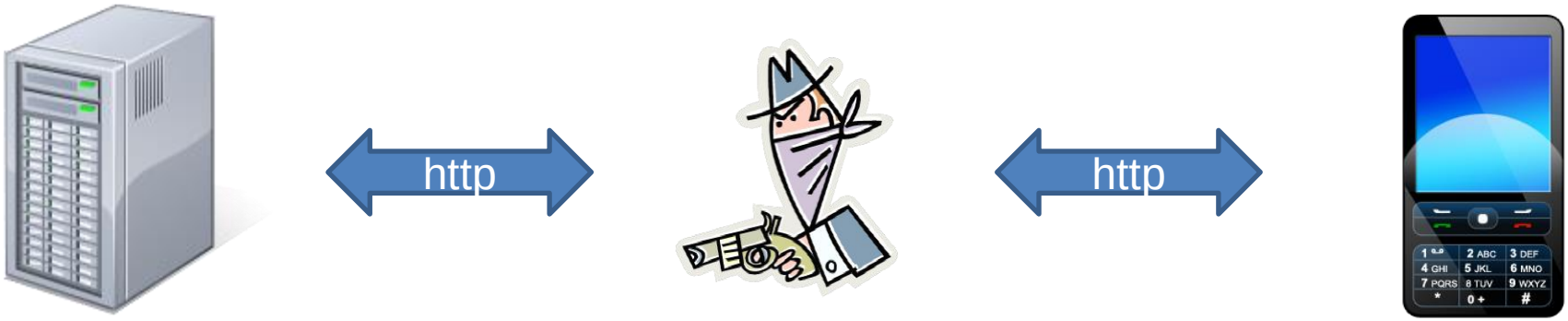
- Implementováno v Javě
- Využití běžných standardních nástrojů
  - Apktool, jarsigner, zipalign
  - Rychlý vývoj
  - Kompatibilita s většinou APK souborů
- Vkládaný kód ve smali
  - Není třeba umět programovat ve smali
  - Naprogramujeme v Javě, zkompilujeme a zpět převedeme do smali
  - Provedeme jen jednou na začátku, pak opakovaně vkládáme stejný kód do řady APK souborů

Modifikace APK souboru stahovaného přes nezabezpečené spojení

ONLINE MODIFIKACE

Scénář – online modifikace APK souboru

- Man in the middle



Reálný scénář

- Man in the middle – např. veřejný WiFi Access Point

Reálný scénář v praxi

- Transparentní http proxy server
  - Všechna spojení otevíraná na port 80 půjdou přes proxy
  - Zajímavé soubory zpracujeme sami zbytek neřešíme
    - Např. přípona .apk
    - Např. typ application/vnd.android.package-archive
  - Proxy získá originální soubor
  - Nechá jej modifikovat
  - Modifikovaný soubor pošle
- Použili jsme „tinyproxy“ a trochu ji přeprogramovali

Automatizovaný skript

- Můžeme použít náš automatizovaný skript?
- Můžeme, ale problém je s časováním
- Pro běh skriptu potřebujeme kompletní stažený balík
  - Pro apktool
- Příklad
  - 2 minuty stahujeme
  - 1 minutu modifikujeme
  - Pak pošleme

Pohled uživatele

- Z hlediska uživatele se 3 minuty nic neděje
  - 0 procent staženo
- Pak se soubor stáhne vysokou rychlostí
- To je špatný výsledek
  - Je to podezřelé
  - Je velká šance, že uživatel během 3 minut čekání stahovaní zruší
- Potřebujeme nenápadnější řešení

Chytřejší modifikace na proxy

- Využijeme toho, že:
  - struktura APK je běžný ZIP
  - na pořadí souborů v ZIPu nezáleží
    - Digitální podpis neřeší pořadí souborů, typ komprese apod. Jde jen o haše jednotlivých souborů.
    - I kdyby to tak nebylo, uměl bych to řešit (podpis je můj), ale takto je to ještě snazší.
  - Ne všechny soubory v APK chci měnit
    - Typicky budu měnit classes.dex a AndroidMenifest.xml
    - Jestli soubor chci měnit nebo ne vím předem (teoreticky)

Proudový režim práce

- Již při stahování souboru začnu pracovat v proudovém režimu
- ZIP postupně dekomprimuji
- U každého souboru rozhodnu zda jej musím měnit
  - Pokud ano, tak jej zatím neposílám
  - Pokud ne, mohu jej rovnou poslat
- Až soubor stáhnu celý, provedu klasickou automatickou modifikaci
- Nakonec z modifikovaného souboru pošlu ty soubory, které jsem ještě neposlal

Uživatelský pohled

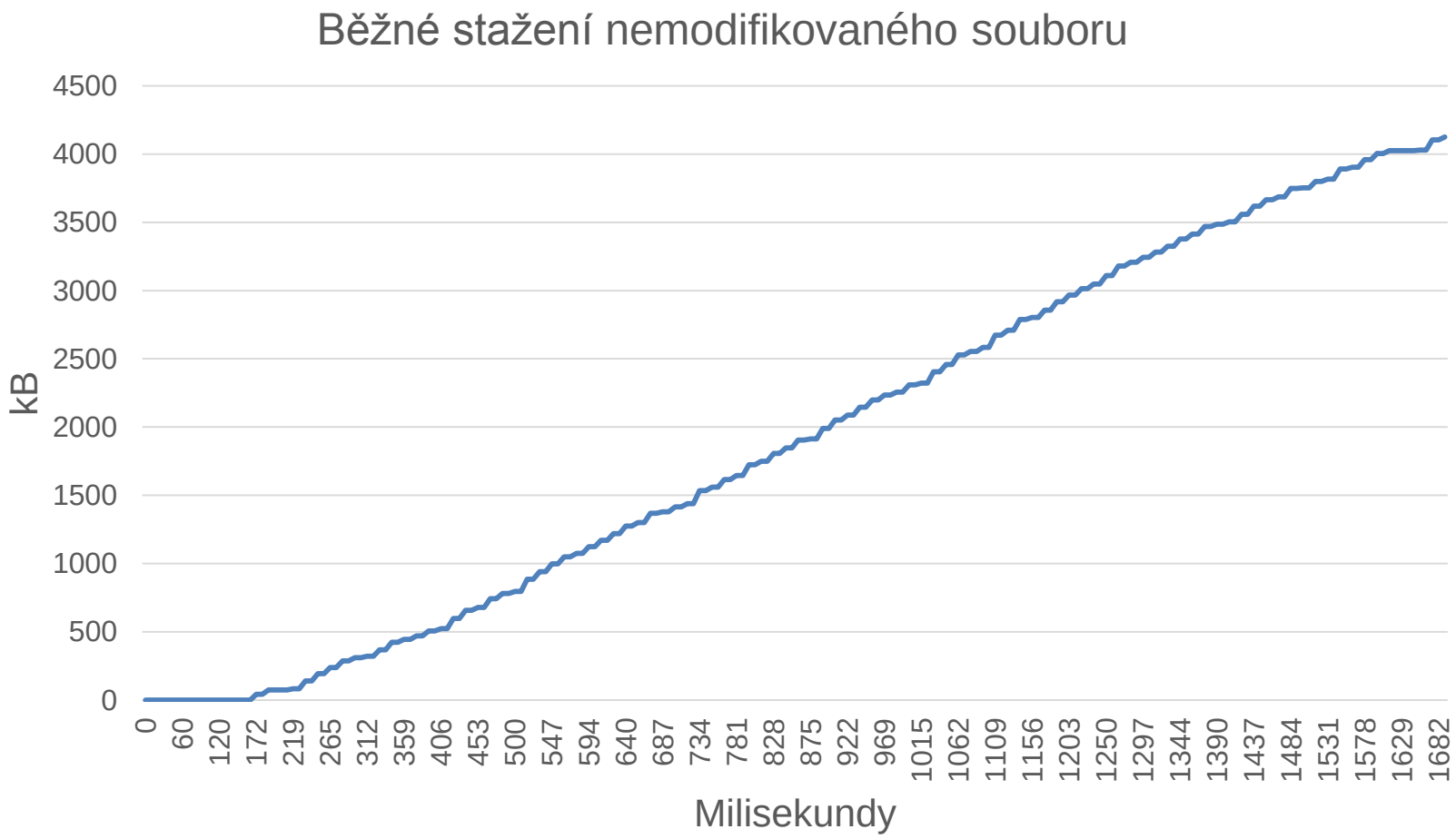
- Stahování začne (téměř) okamžitě
- Stahování pokračuje (přibližně) do poloviny
- Potom se stahování zastaví
  - Protože provádíme automatickou off-line modifikaci
- Potom stahování pokračuje

- Výsledek je výrazně lepší
  - Na začátku dostaneme nějaká data
- Ale pořad to ještě není ideální

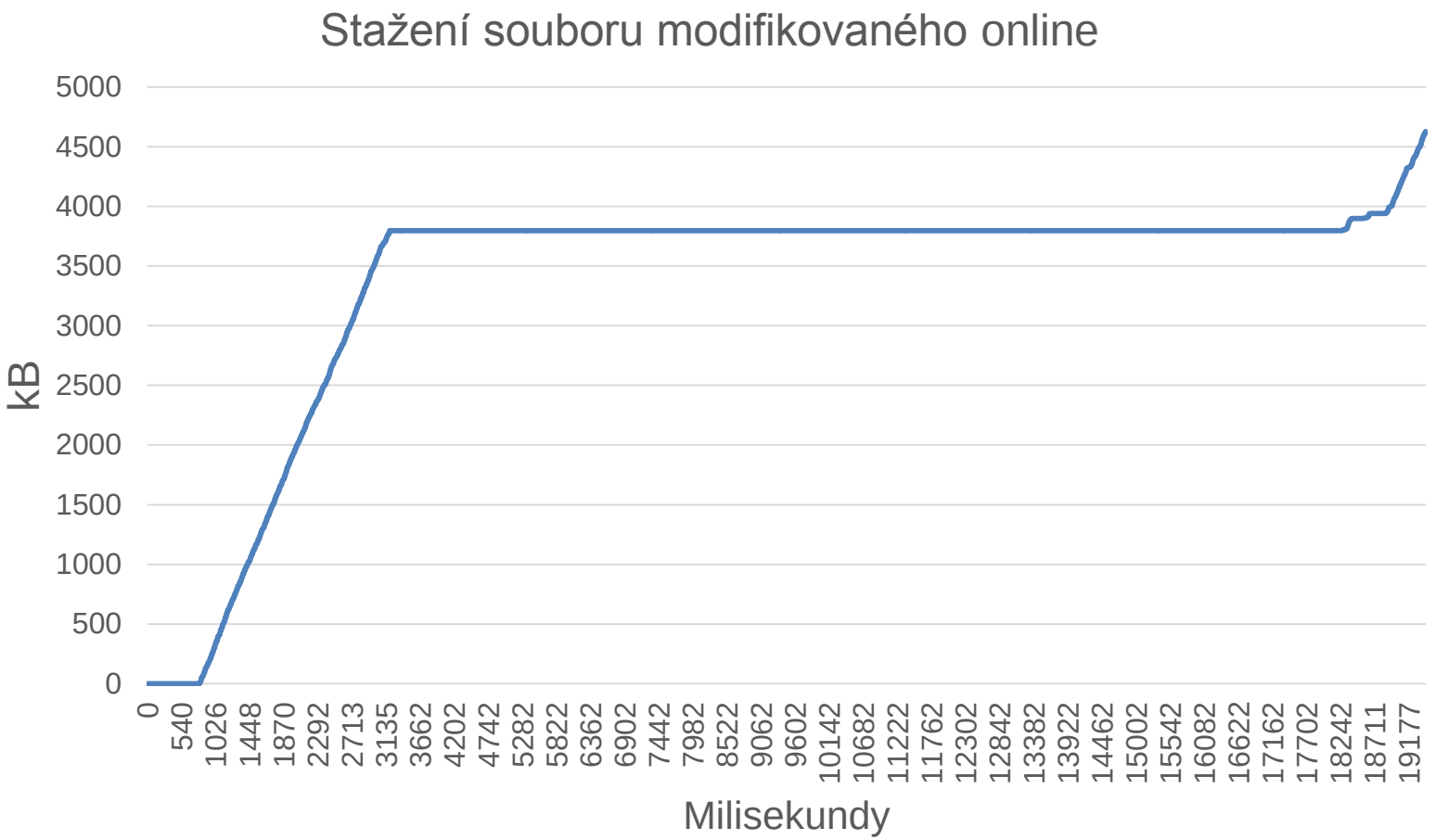
Stále lépe a lépe …

- Odstranění dlouhé prodlevy uprostřed
  - Snížíme rychlost hned od začátku ☺
- Stahování tak je pomalé, ale plynulé
  - A to není téměř nijak nápadné
- ALE abychom viděli prostup v procentech musíme znát přesnou velikost již na začátku stahování
  - Takže ji tipneme (raději více než méně…)
  - Tj. původní velikost + co vkládáme + něco navíc
  - A později, až známe skutečnou velikost, nacpeme nuly do ZIP poznámek

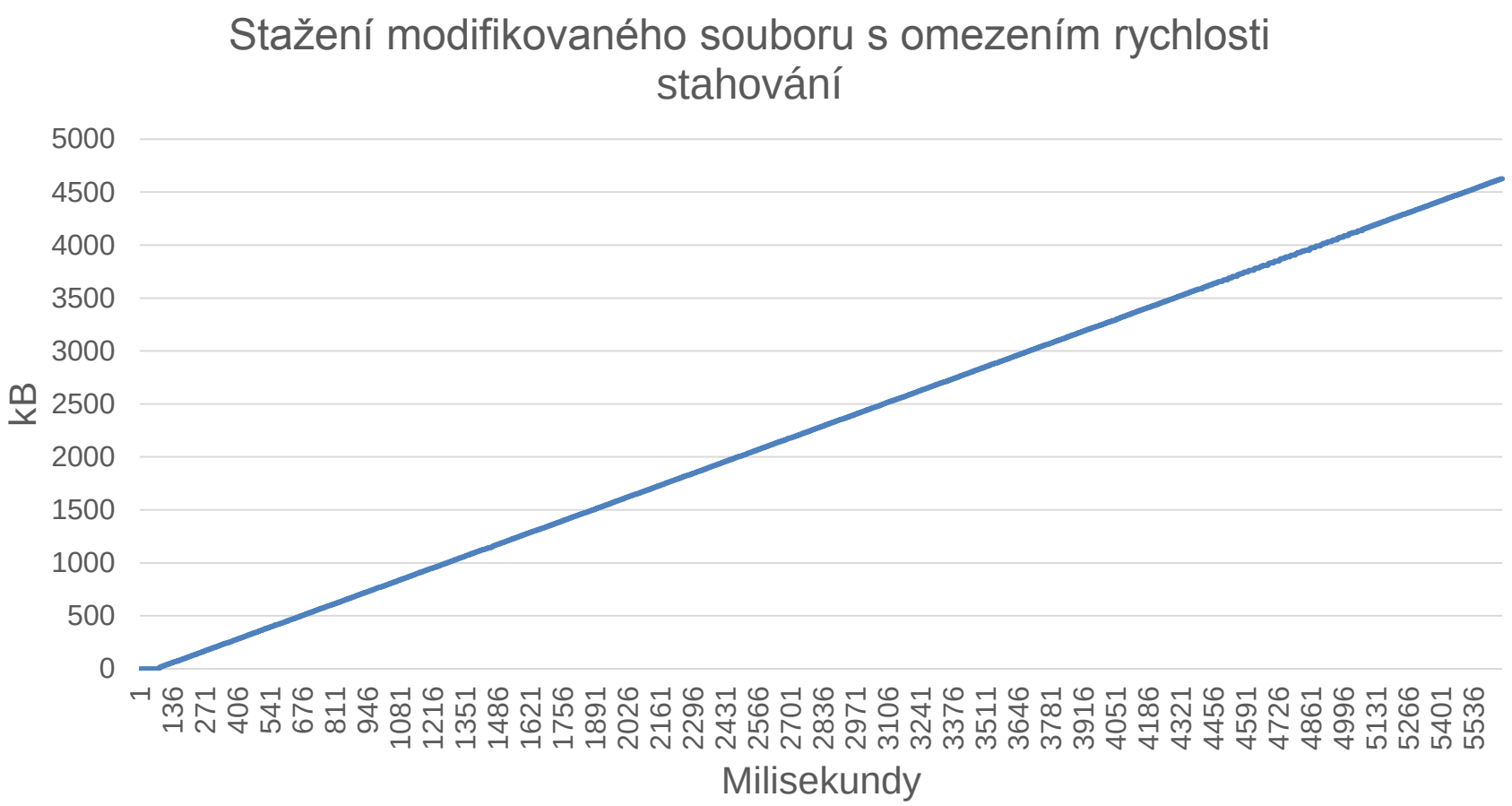
Ilustrace – stahování bez modifikace



Ilustrace – stažení modifikovaného souboru



Ilustrace – omezení rychlosti stahování



Závěr

Otázky?

zriha@fi.muni.cz