PROGRAMMING AND HACKING ANDROID

Fabrizio Cornelli
HT

Programming & Hacking Android

First rule of HT
- You do not talk about HT

Programming
- Software Engineer
- Constructive
- Good practices
- Team work
- Read the manual
- Frameworks and libraries
- Make the rules

Hacking
- Reverse Engineer
- Deconstructive
- Get into the details
- Subvert the manual
- Shortcut to the goal
- Small languages
- Break the rules

APT concept
- installation
- privilege escalation
- persistence
-]stealth[
- configuration

Life Cycle for an APT
- configuration and build
- installation
- persistence
- execution on event
- data gathering
- data exfiltration
- uninstall

Approaching a new scenario
Android
- new os, based on linux
- free but maintained by a big player
- bright future

Writing a new app on Android
- eclipse / android studio
- start from something simple
- build: ant vs gradle
- team working: svn, git or mercurial…
- signature
  - self signed signature
  - playstore
- manifest and permissions

Learn from Malware
- VT
- reverersing
- static / dynamic

What can we do?
- poc for build
- poc for installation
  - local
  - remote
- poc for persistence
- poc for invisibility

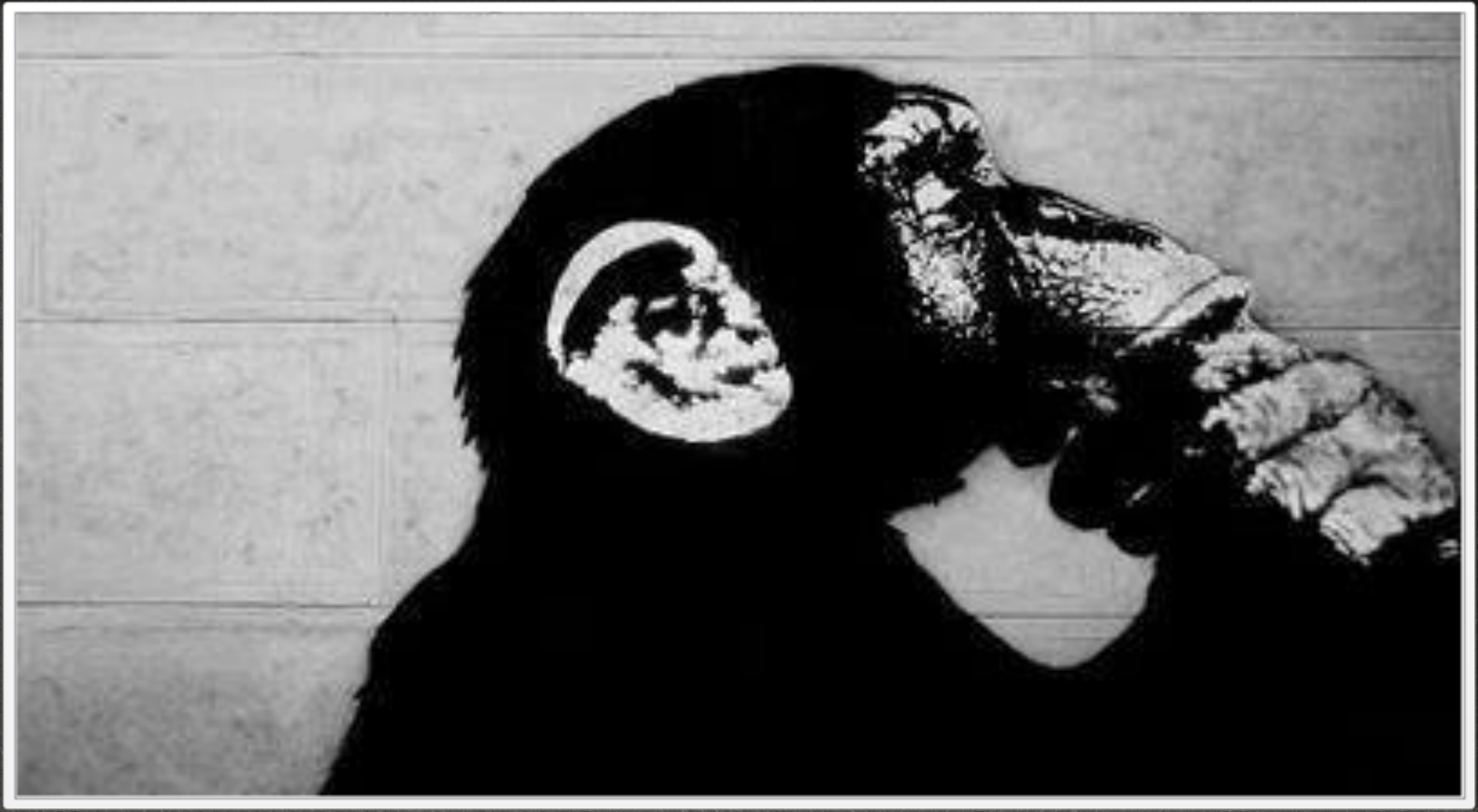
Hacking Android
Subverting Android permissions
- exploit
  - zero day / unpatched
  - get the system
  - get the root
- social engineering

AV Evasion

Repackage / partitions

Bring together, engineer a product

We are hiring



FABRIZIO CORNELLI

zeno@hackingteam.com

CV

- Filibusta LUG
- Laurea a Crema, nel 2012
- CTO, Enterprise srl
- QA Manager, HT

FIRST RULE OF HT

YOU DO NO TALK ABOUT HT

]HackingTeam[

- Fighting crime since 2003
- Internet is wrong

Summary

- Developer
- Hacker
- Android
- How to write an APT on Android

```
                                        business
#include "Investment.h"
#include "MyProjects/Startup/Success.h"
#include "MyProjects/Startup/Business.h"

template< typename BusinessStrategy, typename Investment >
class Business: public BusinessStrategy
{
    Business( Investment& MyInvestment );
```

DEVELOPER

"If it ain't broke, don't fix it"

Software engineer

- Constructive
- Programming skills
- Good Practices
- Design then code (and test)
- RTFM
- Frameworks and Libraries
- Don’t be the first
- High level languages

Software Engineering Proverbs

- The ends does not justify the mean
- Choose two: good, fast, cheap
- Any fool can write code that a computer can understand. Good programmers write code that humans can understand. [M. Fowler]

HACKER

"shit happens"

Hacker

- Deconstructive
- Reverse Engineer
- Lateral Thinking
- Lazy
- Subvert the manual
- Shortcut
- Must be the first
- Low level languages (C, asm)

Hacking Proverbs

- the ends justify the means
- a clever person solves a problem, a wise person avoids it.

SE vs Hacker

- permanent settlements and government

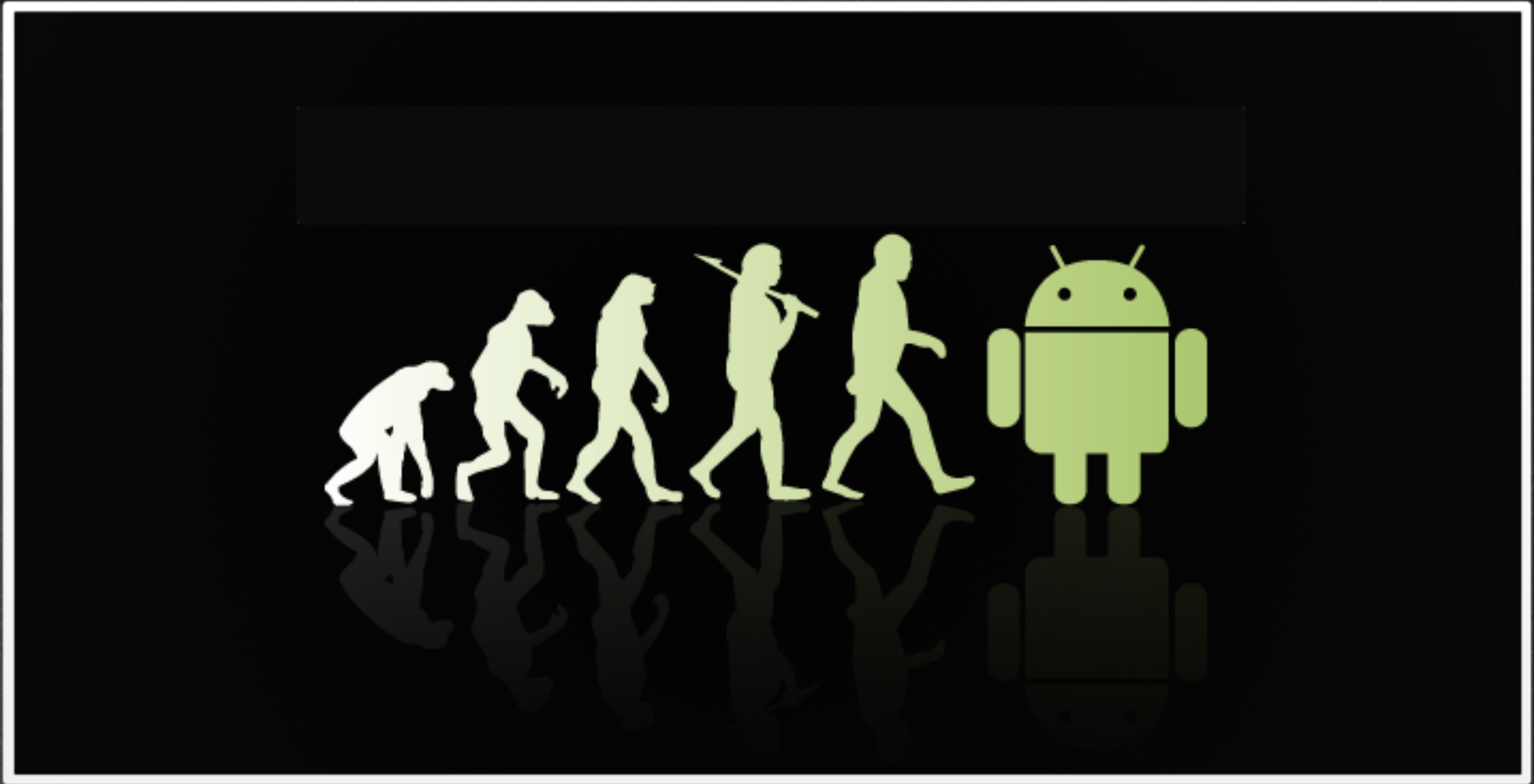
- hunter-gatherers

SE & Hacking

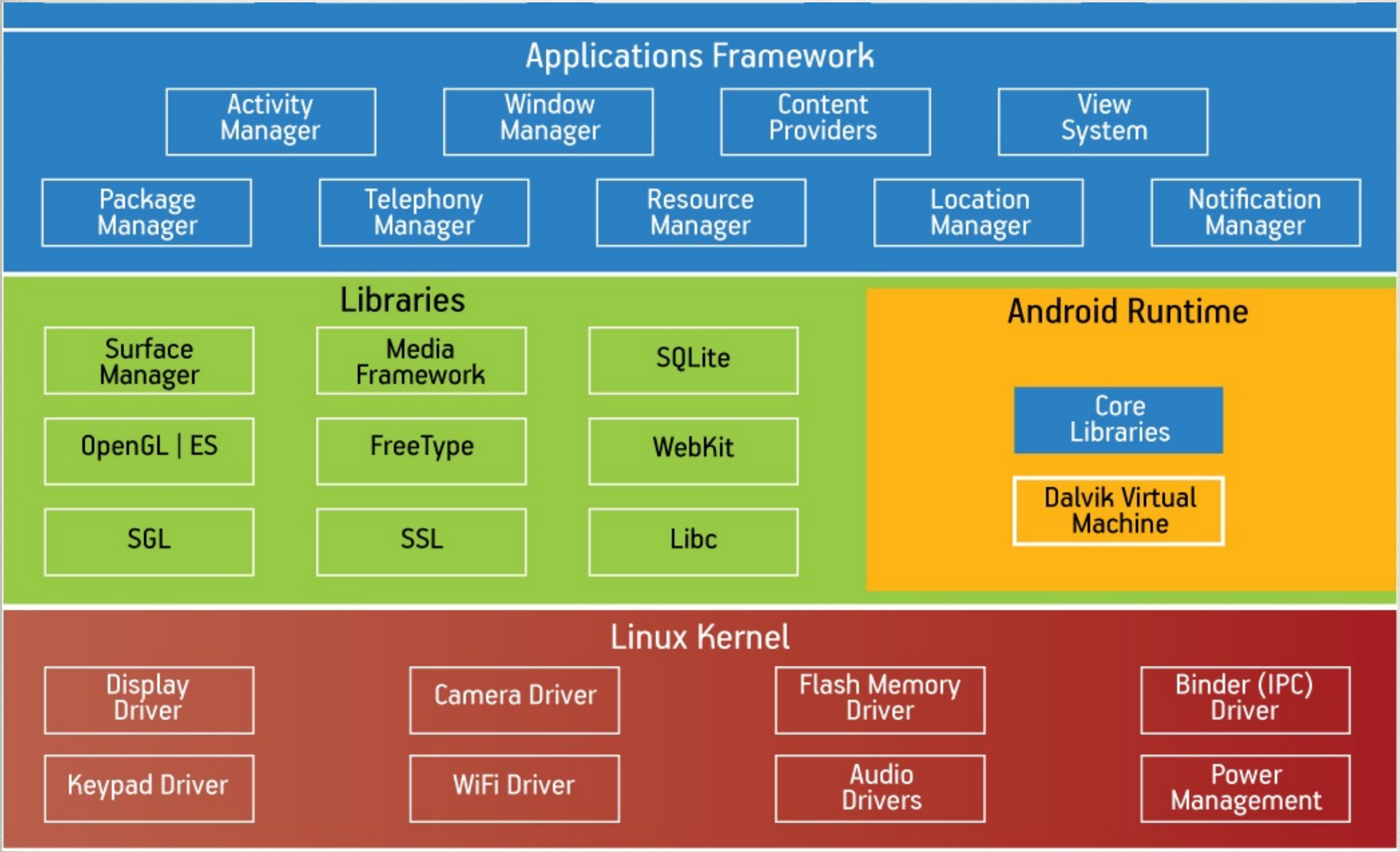
- Nerd / Geek
- Discipline
- Require both a lot of study
- Imagination
- Security

Programming

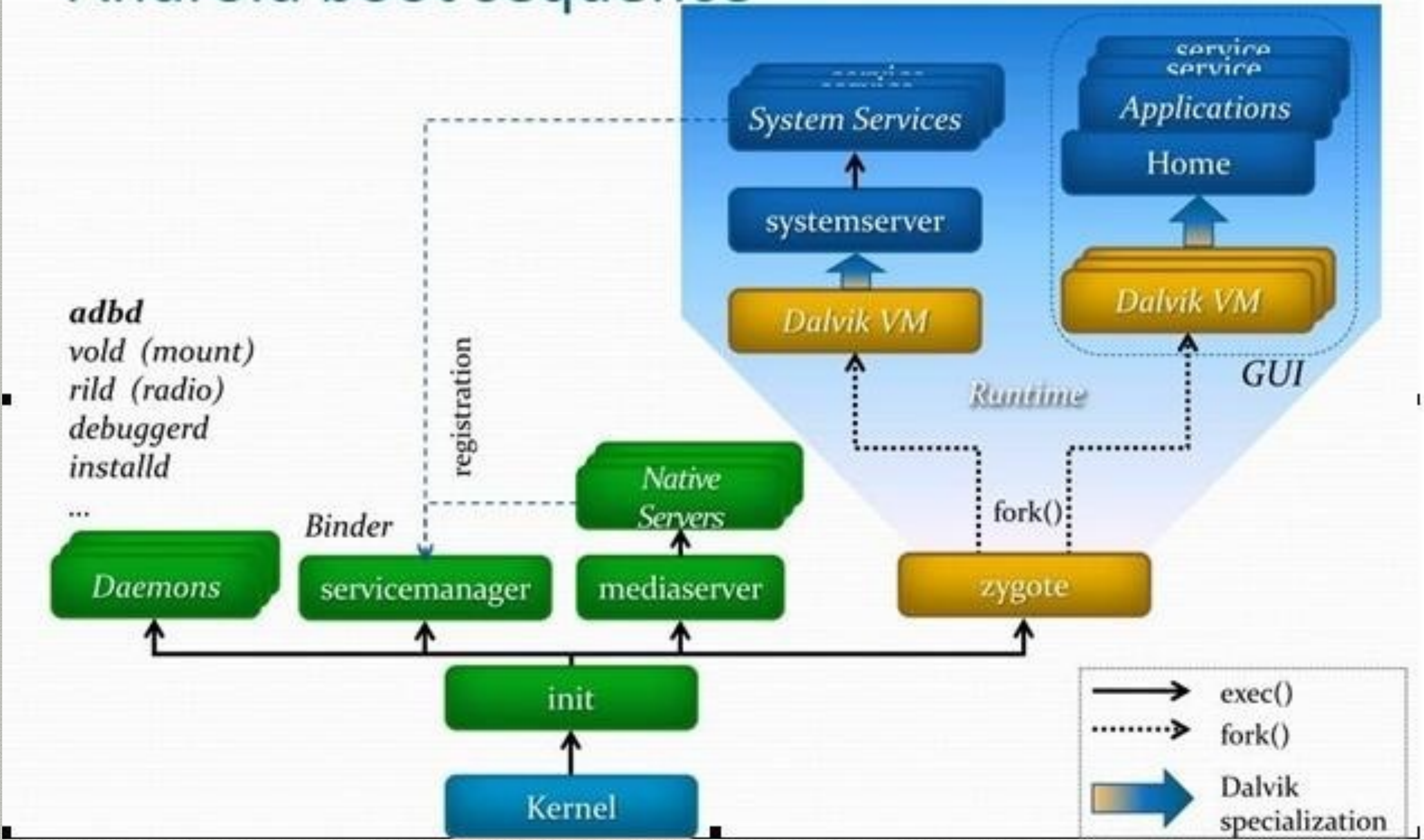
- Be a hacker: get you POC
- Be a developer: evolve it to a product

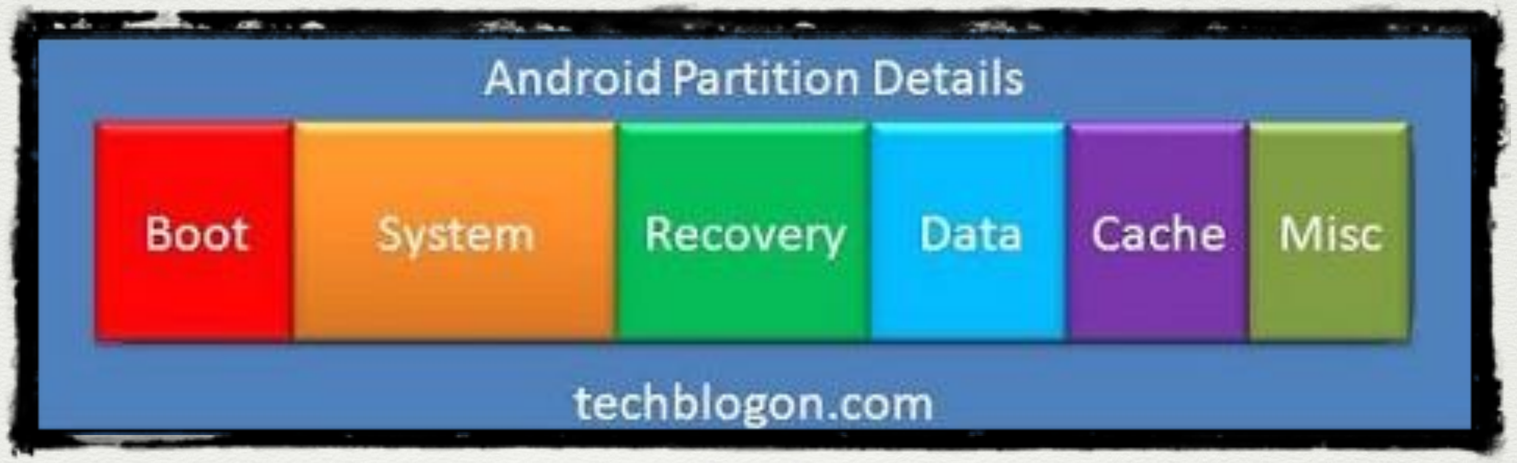


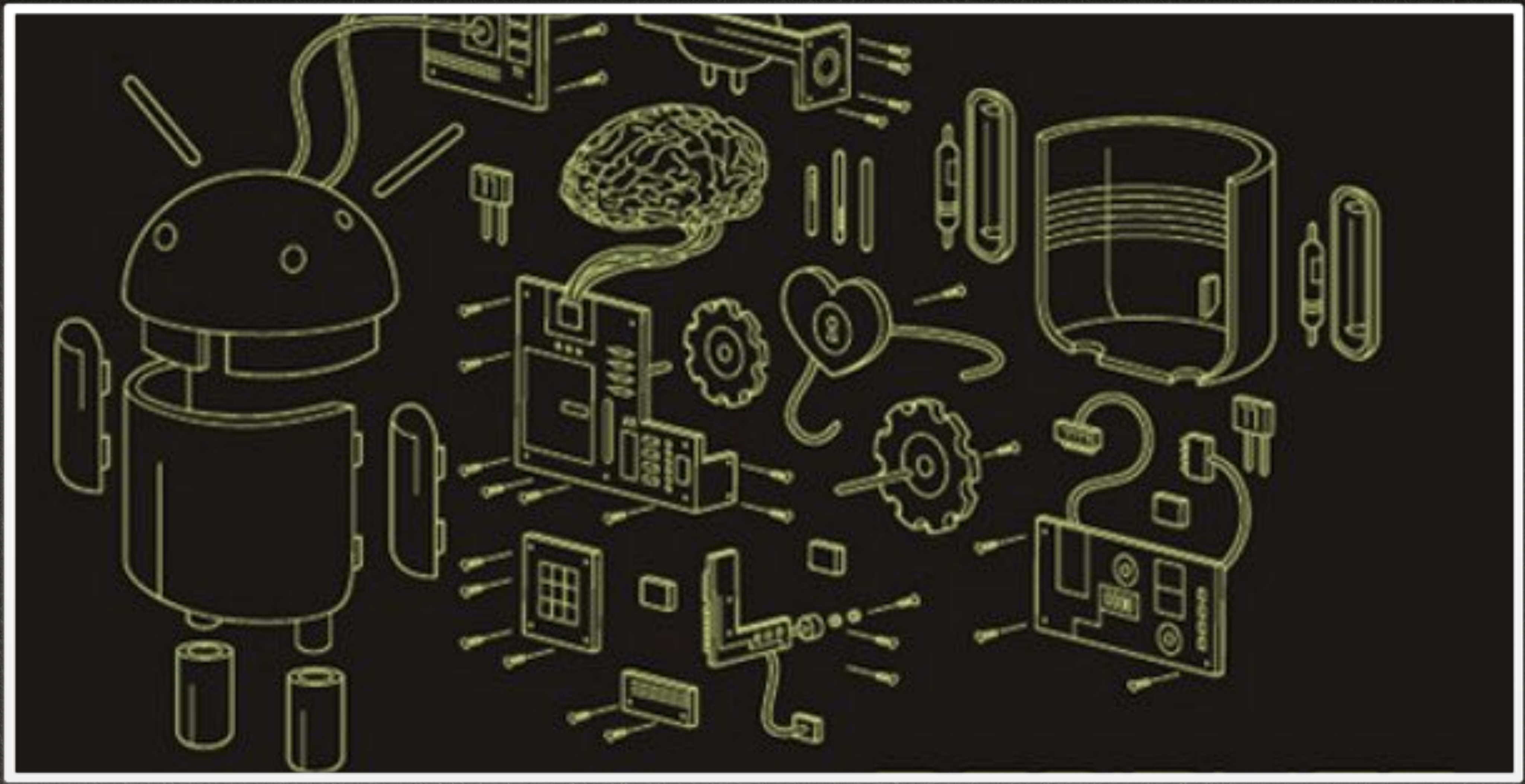
ANDROID


Applications Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
Libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Android boot sequence




Android Partition Details
Boot
System
Recovery
Data
Cache
Misc
techblogon.com



ANDROID DEVELOPMENT

Android Studio

- IntelliJ Platform
- IDE
- gradle
- adb
- emulator

APK

- classes.dex : Code
- Manifest
- Resources / Assets /Licence
- Libs
- Signature

ADB

- Android Debug Bridge
- Device in Debug Mode
- subcommands:
    - shell
    - pull
    - push
    - install
    - kill-server
    - reboot

REVERSING

Tools

- Decompilers
  - jd-gui
  - dad
  - jeb
- Apk dissectors
  - androguard
  - apktool
- Reversing Frameworks
  - IDA
  - Radare
- Network analyzer
  - Wireshark / tcpdump
  - burp

apktool

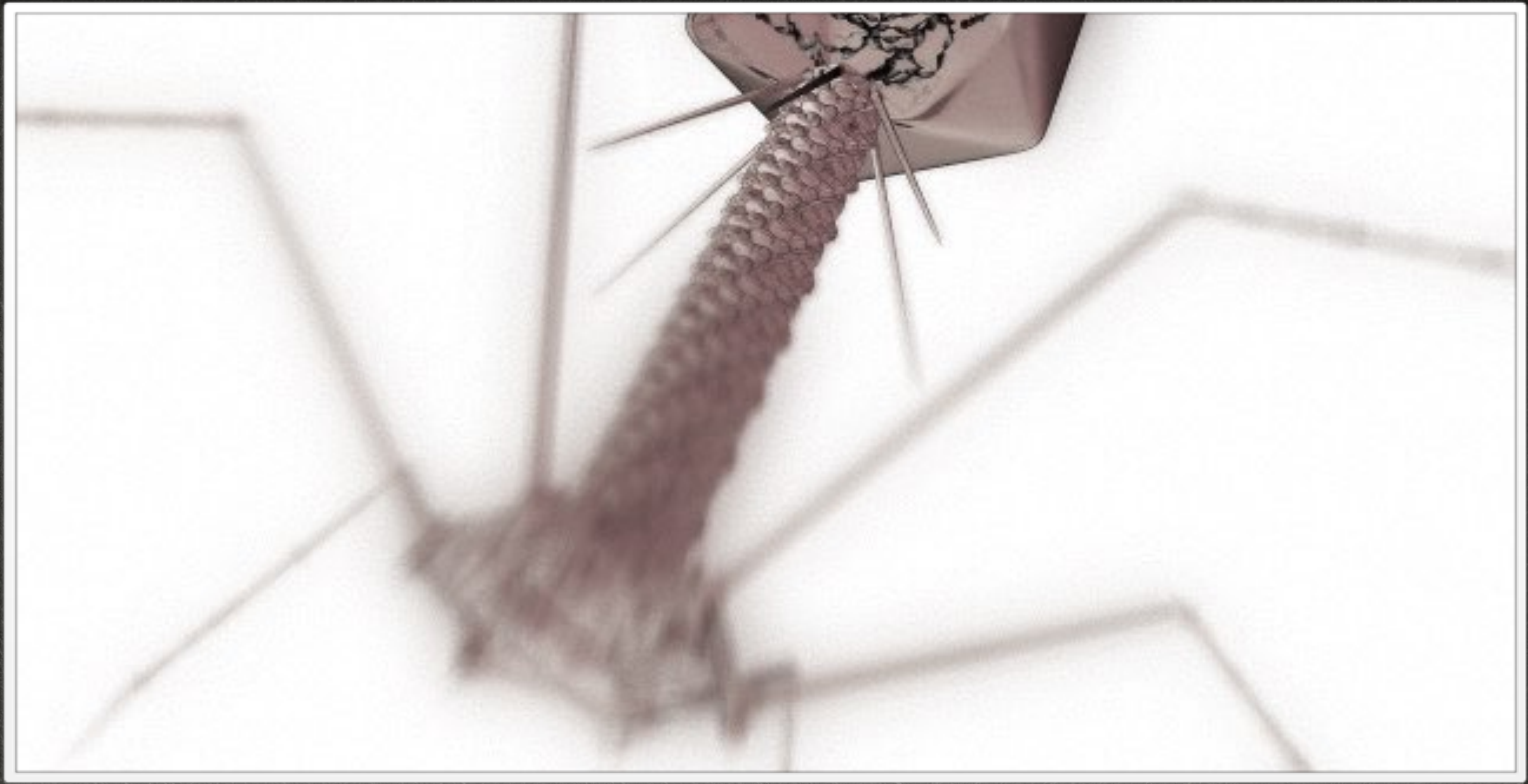
- decode apk
- build apk
- internal use of smali/baksmali
- needs jarsigner

smali

```
.method public static main([Ljava/lang/String;)V
    .registers 2

    sget-object v0, Ljava/lang/System;->out:Ljava/io/PrintStream;

    const-string   v1, "Hello World!"

    invoke-virtual {v0, v1}, Ljava/io/PrintStream;->println(Ljava/lang/String;)V

    return-void
.end method
```

learn from malware

- http://contagiominidump.blogspot.it/
- virus total
- androguard DatabaseAndroidMalwares



APT CONCEPTS

Advanced Persistence Threats

Life cycle

- Configuration
- Build
- Installation
- Execution
- Persistence
- Data Exfiltration
- Uninstall

HACKING ANDROID

Get the root

- Flash the OS
- Use a local to root exploit
- Use a system exploit

Starting at the boot

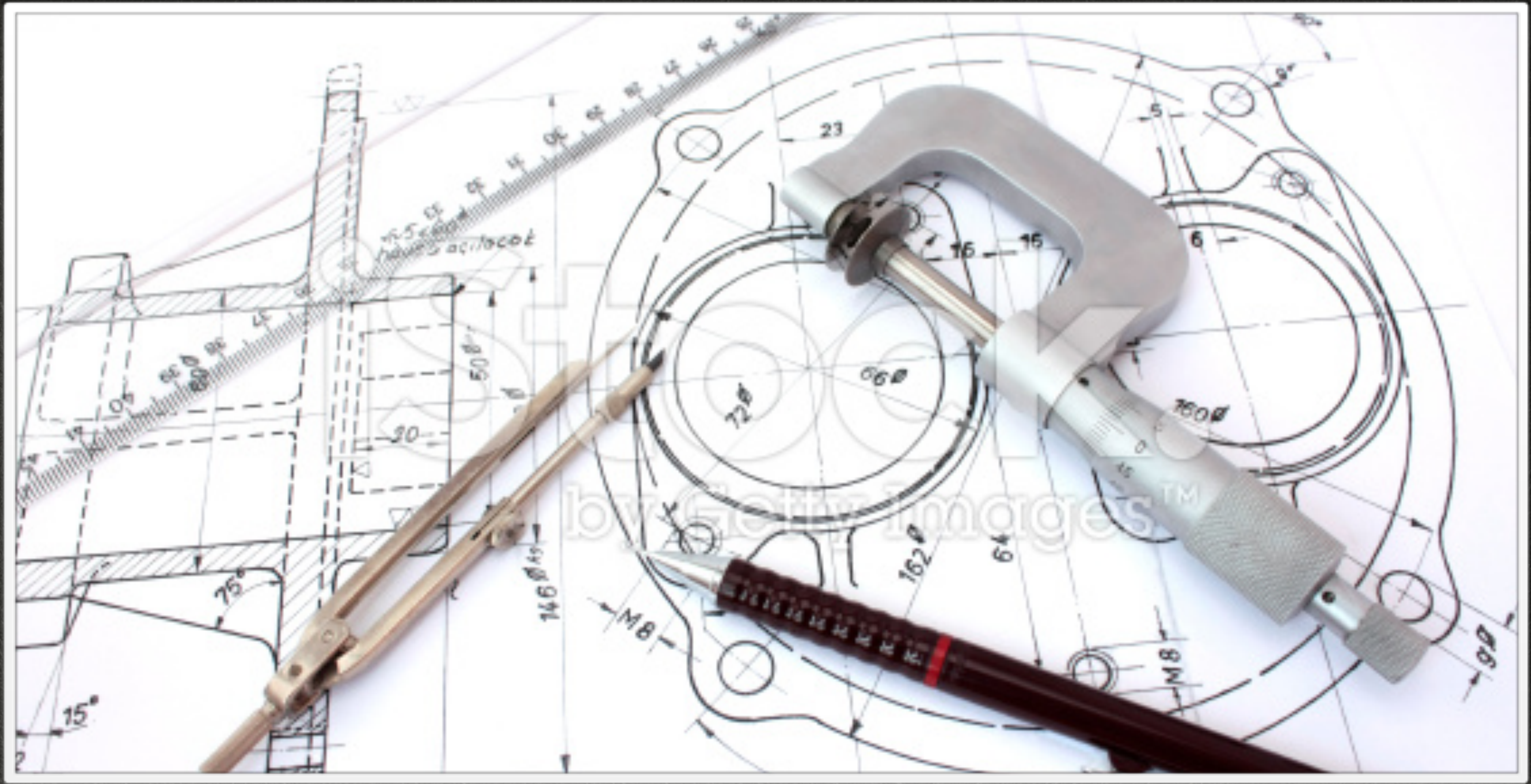
- Use the Manifest
- Use the OS (root)

Get data

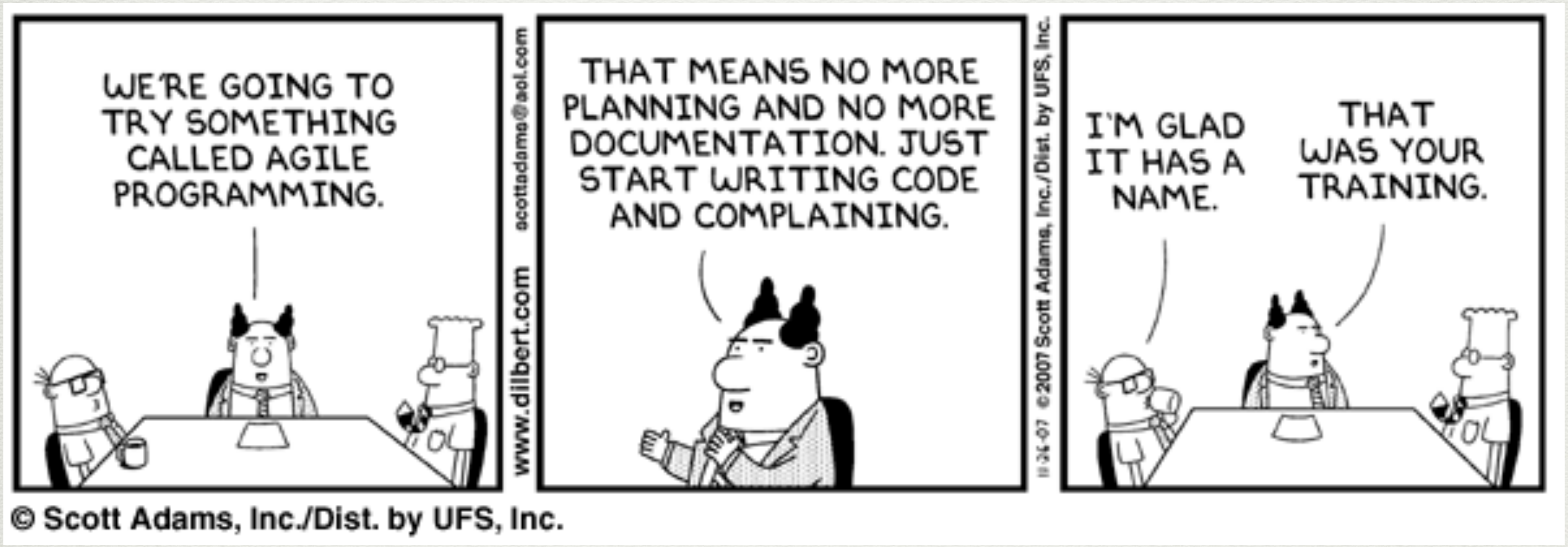
- Use Android API
- Use OS libraries
- Get data reading memory (root)
- Get data reading files (root)

Communication

- Covered link
- Use Android API



ENGINEER A PRODUCT


WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
www.dilbert.com
scottadams@aol.com
© 2007 Scott Adams, Inc./Dist. by UFS, Inc.
11-26-07
© Scott Adams, Inc./Dist. by UFS, Inc.

Agile programming

Protect from hackers

- anti reversing tricks
- polimorfic tools
- encryption and obfuscation
- anti virtualization tricks
- packing
- virtualize

Maintain the code

- Versioning
- Continuous integration
- Testing and code coverage
- Acceptance tests
- Automatic tests

Make a product

- Customer support
- Release policies
- Marketing

LINKS