RSA®CONFERENCE CHINA 2012

RSA信息安全大会2012

THE GREAT CIPHER

MIGHTIER THAN THE SWORD

伟大的密码胜于利剑

OPEN PLATFORM SECURITY FOR MOBILE INTERNET

Tieyan Li

Irdeto (Online)

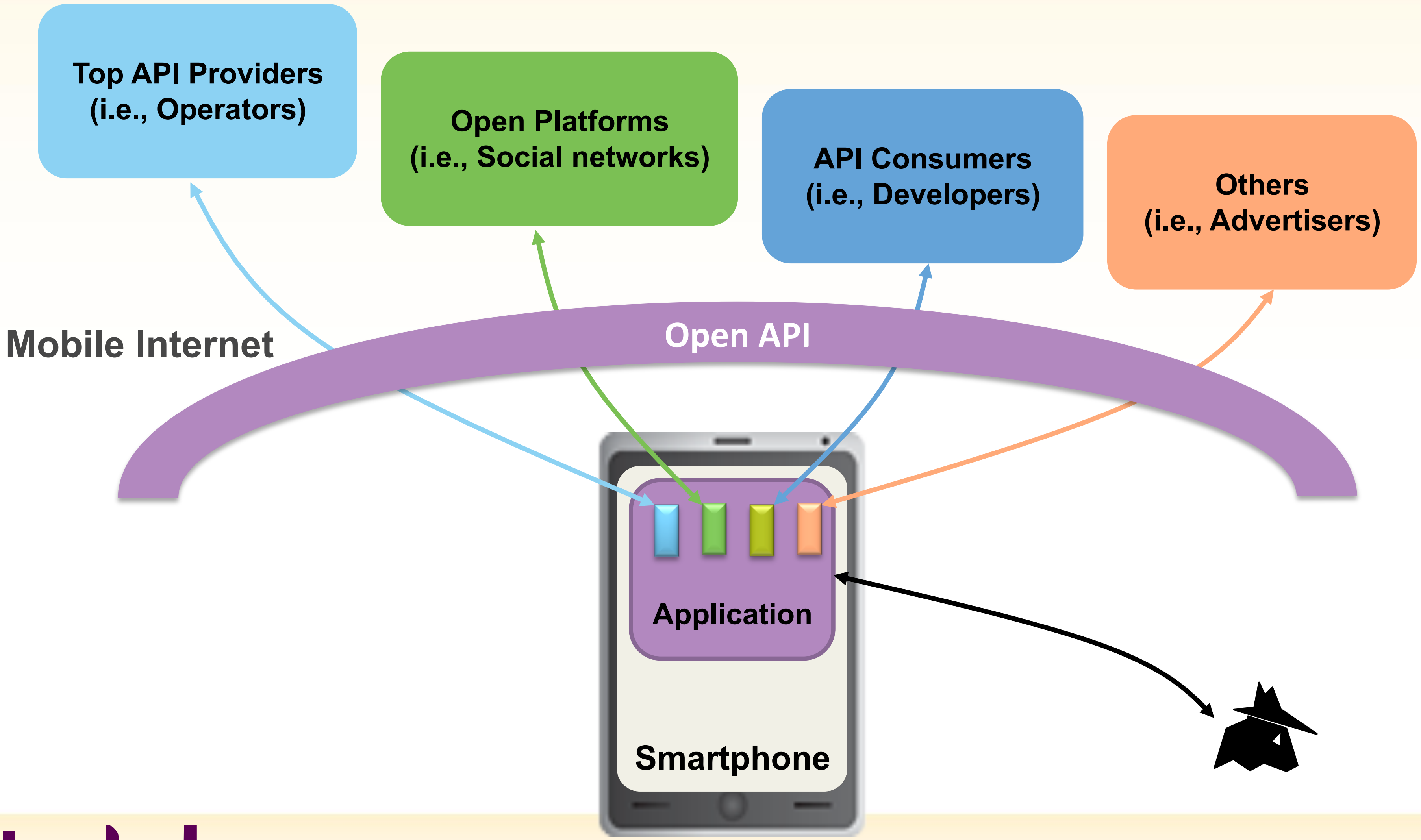
RSACONFERENCE
CHINA 2012
RSA信息安全大会2012

Agenda

- Open Mobile Platform: Risks and Mitigations
- Mobile Application Protection
- Android Platform Security
- Dynamic & Full Lifecycle Security

Open Mobile Platform

Open Platform Ecosystem



Top API Providers (i.e., Operators)

Open Platforms (i.e., Social networks)

API Consumers (i.e., Developers)

Others (i.e., Advertisers)

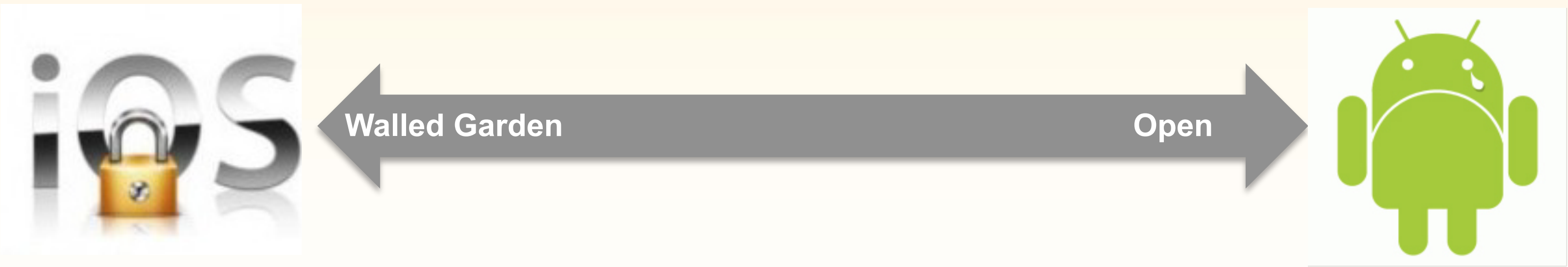
Mobile Internet

Open API

Application

Smartphone

Open has a price …..



Walled Garden

Open

- Apple controls what runs on the device via platform security
- Apple limits OS services that are available to third party apps
- Apple certifies/screens all apps
- Apple interested in the health of the eco-system (e.g. App developers, network traffic)

Pros

- Security is generally good (absent a jailbreak)
- Great consumer experience

Cons

- Mobile operators are marginalized in the iPhone eco-system
- High cost of devices

- Open source
- Apps have access to low level OS services
- Multiple app stores with no certification or screening process
- Relies on users to make informed decisions about security
- Supply chain is much more complex than IOS

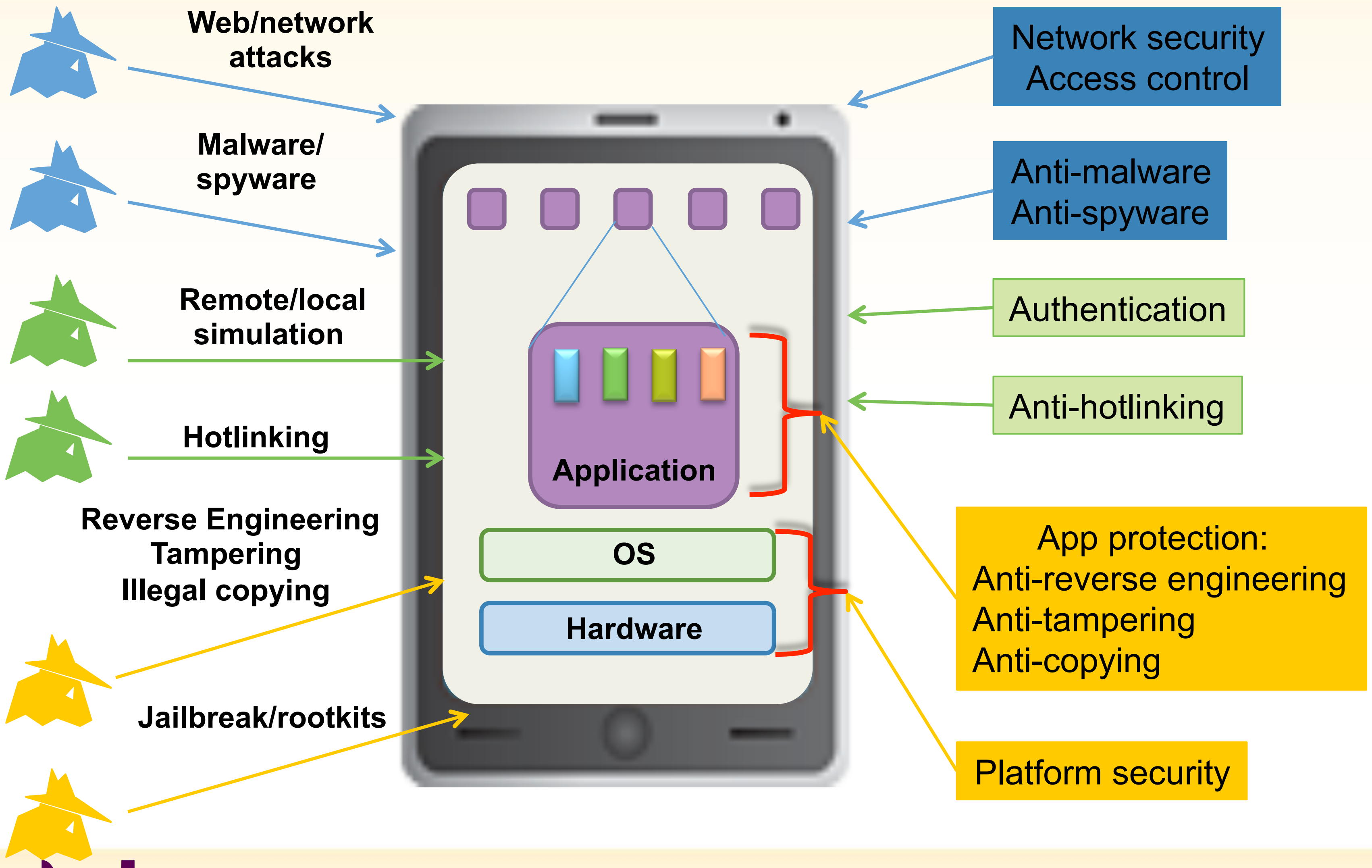
Pros

- Android provides greater opportunity for mobile operators
- Lower cost smartphones

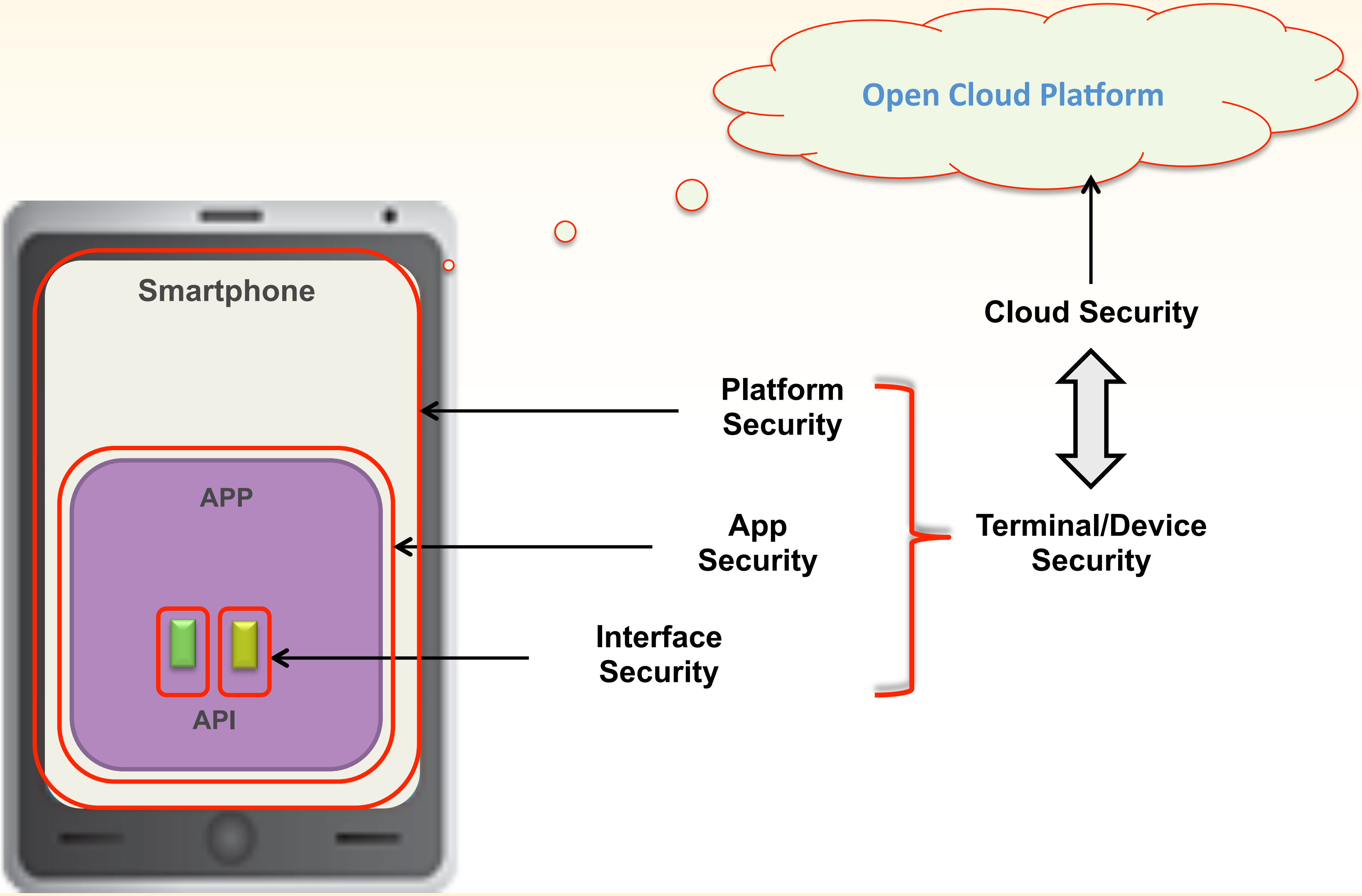
Cons

- Security and operational issues impact mobile operators and consumers

Risks and Mitigations

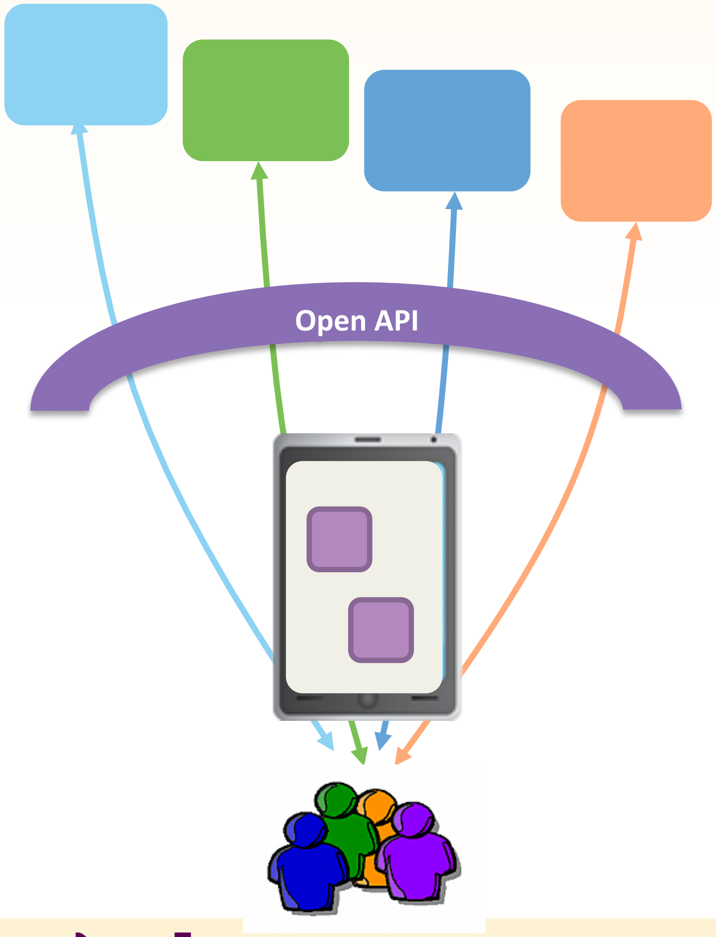


Building Trust Boundary

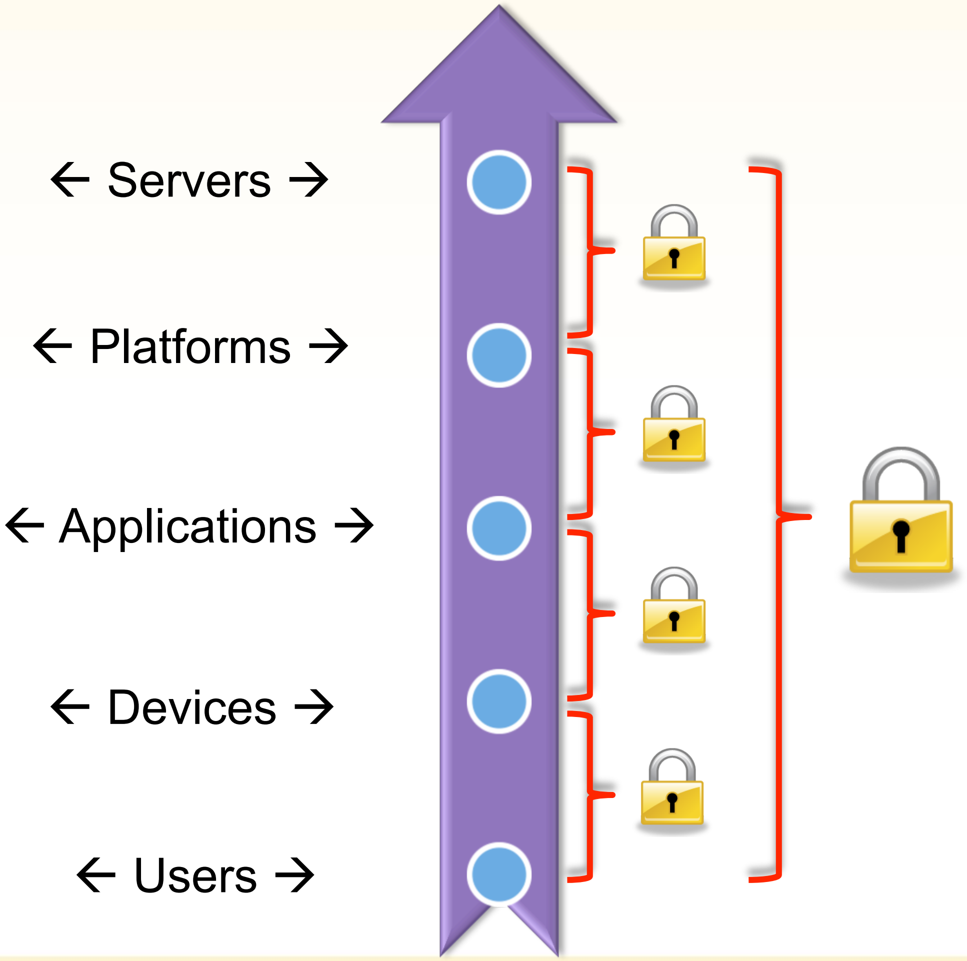


Simpler, Closer, Vertical Security

Mobile Internet



Vertical Security



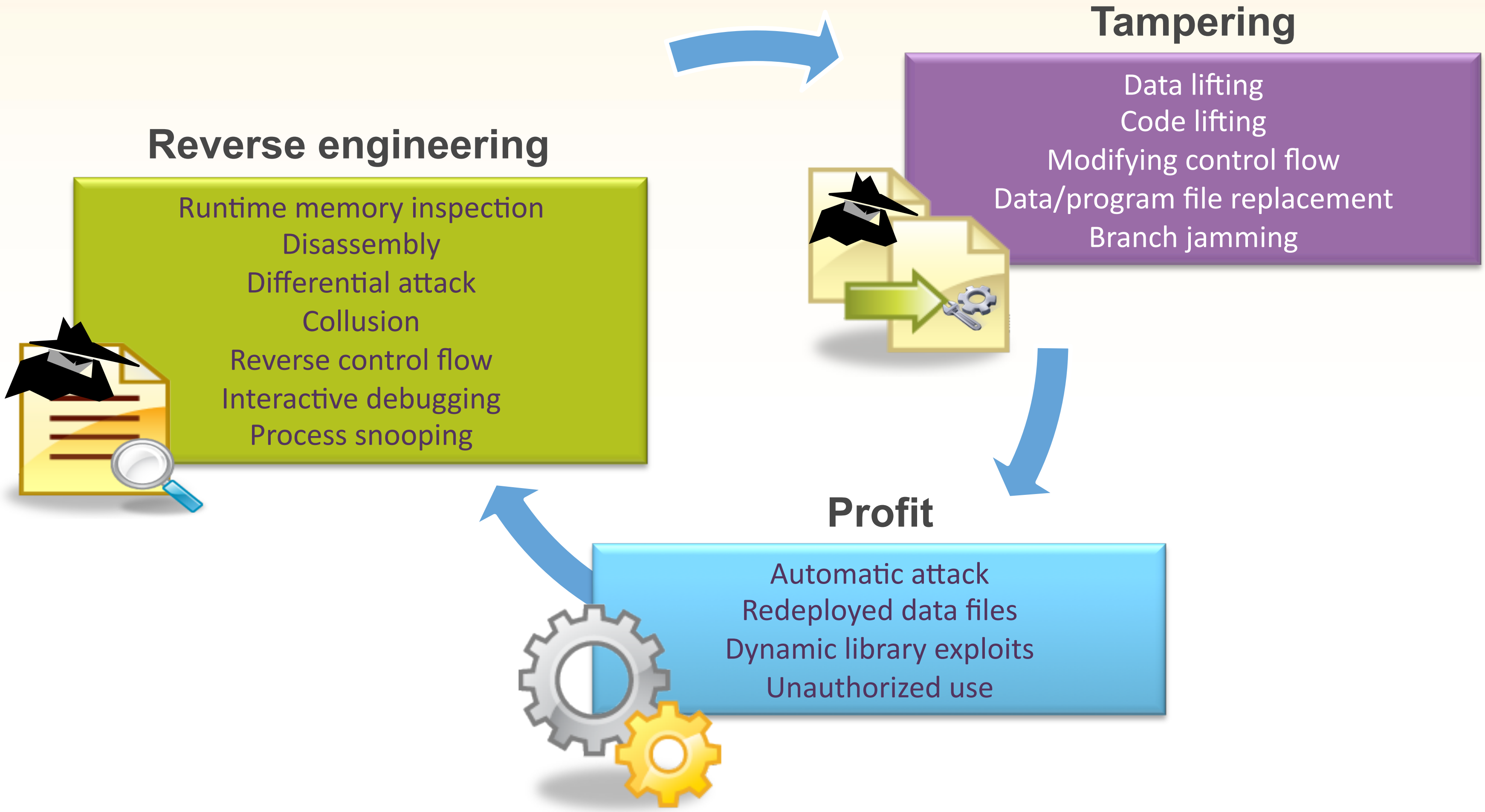
Mobile Application Protection

Why App Protection?

- The Evil-Twin
  - Piracy: Violate the intellectual property of the original Apps.
  - Malware injection: Re-packaged Apps may contain malwares, botnets, trojans, etc.
  - A recent study disclosed that nearly 86% of all malware payloads are found in re-packaged versions of legitimate applications.
- Major task is to protect an App from:
  - Illegal copying
    - I.e., paid assets, virtual goods
  - Reverse-engineering
    - I.e., leading to loss of IP, re-packaging
  - Tampering
    - I.e., game cheating, bypass billing point, or piggybacking malicious code

Attacks on software

Software is susceptible to different attacks



Tampering

- Data lifting
- Code lifting
- Modifying control flow
- Data/program file replacement
- Branch jamming

Profit

- Automatic attack
- Redeployed data files
- Dynamic library exploits
- Unauthorized use

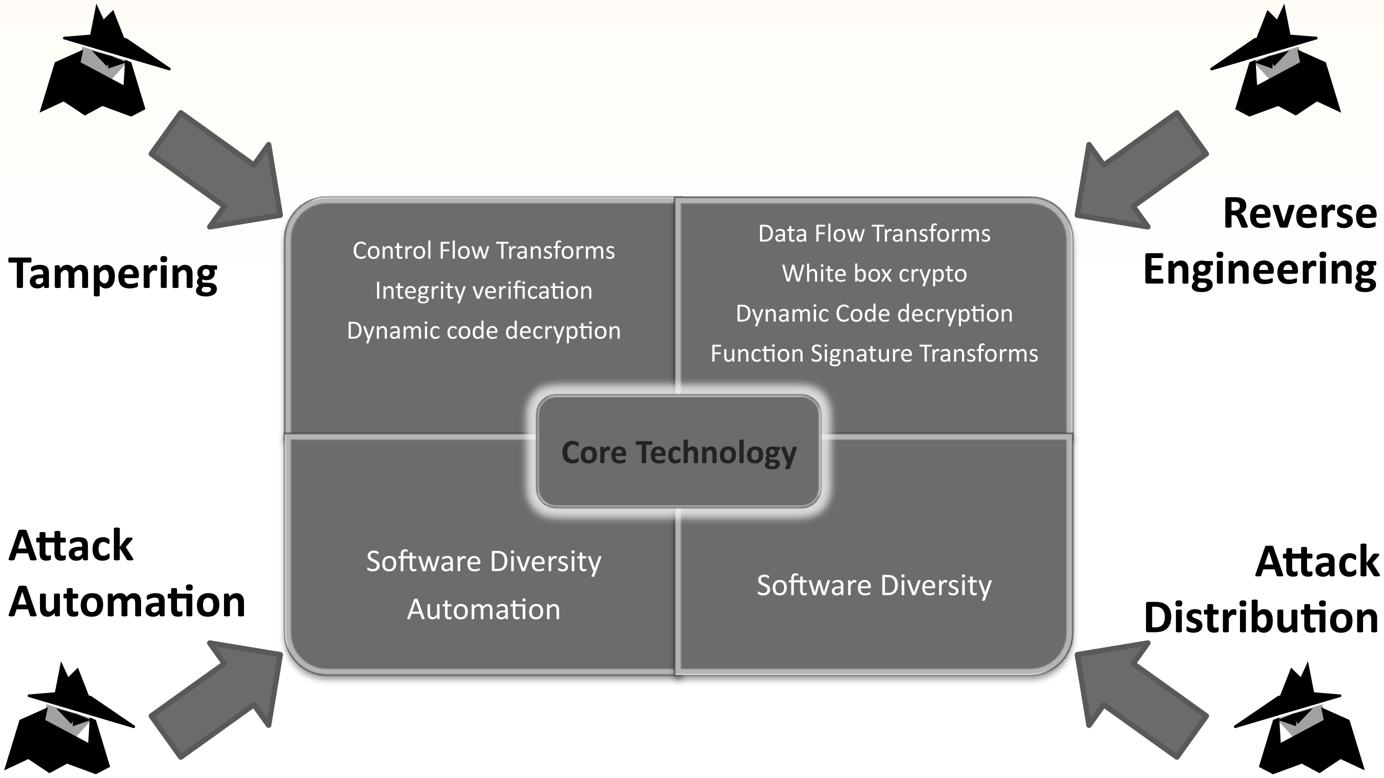
Reverse engineering

- Runtime memory inspection
- Disassembly
- Differential attack
- Collusion
- Reverse control flow
- Interactive debugging
- Process snooping

Different attacks need different protection

Software Protection

- World leading technology to protect software against reverse engineering, and automated attacks.



Tampering

Control Flow Transforms
Integrity verification
Dynamic code decryption

Reverse Engineering

Data Flow Transforms
White box crypto
Dynamic Code decryption
Function Signature Transforms

Core Technology

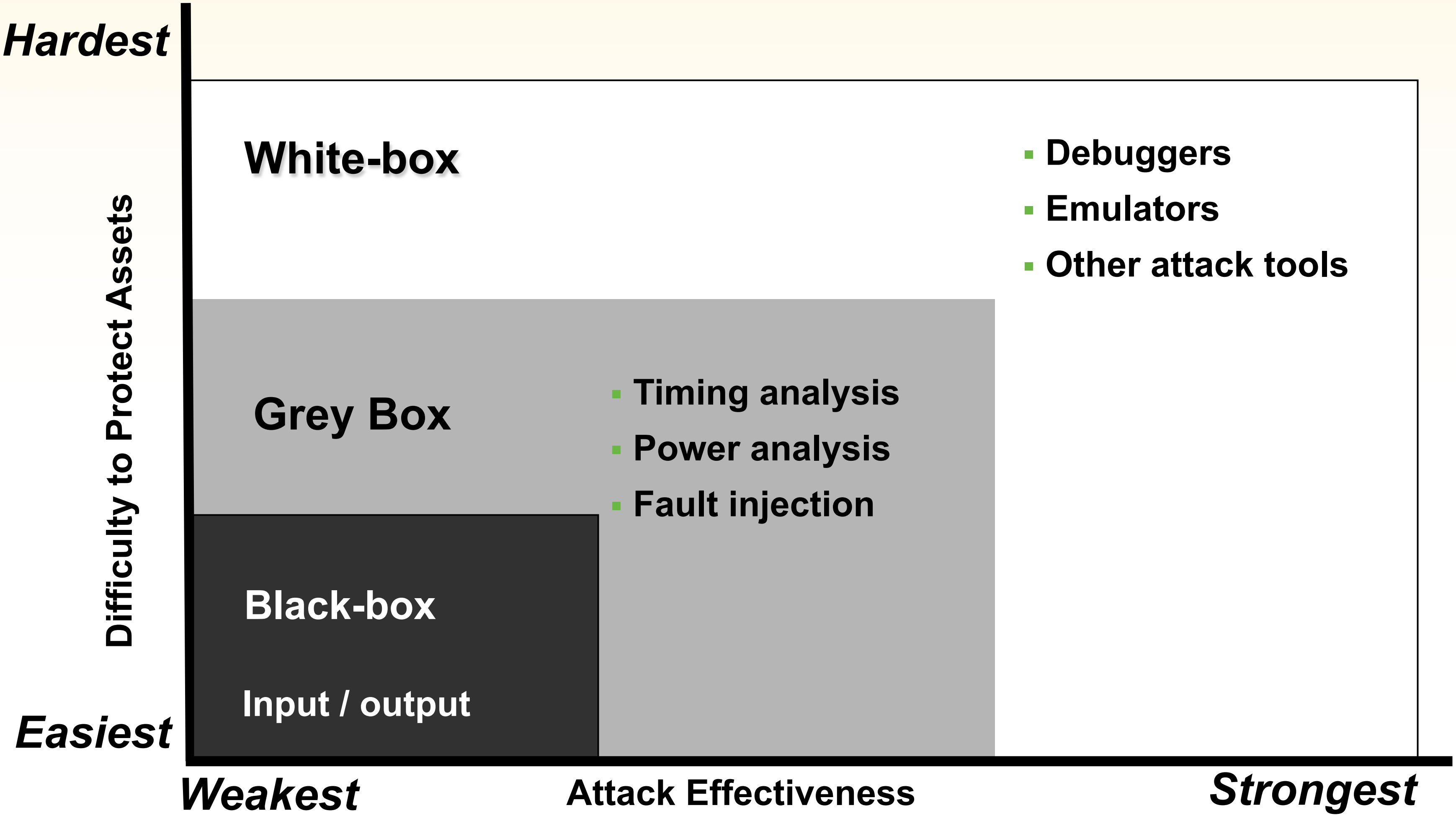
Attack Automation

Software Diversity
Automation

Attack Distribution

Software Diversity

Security Models



White-box

- Debuggers
- Emulators
- Other attack tools

Grey Box

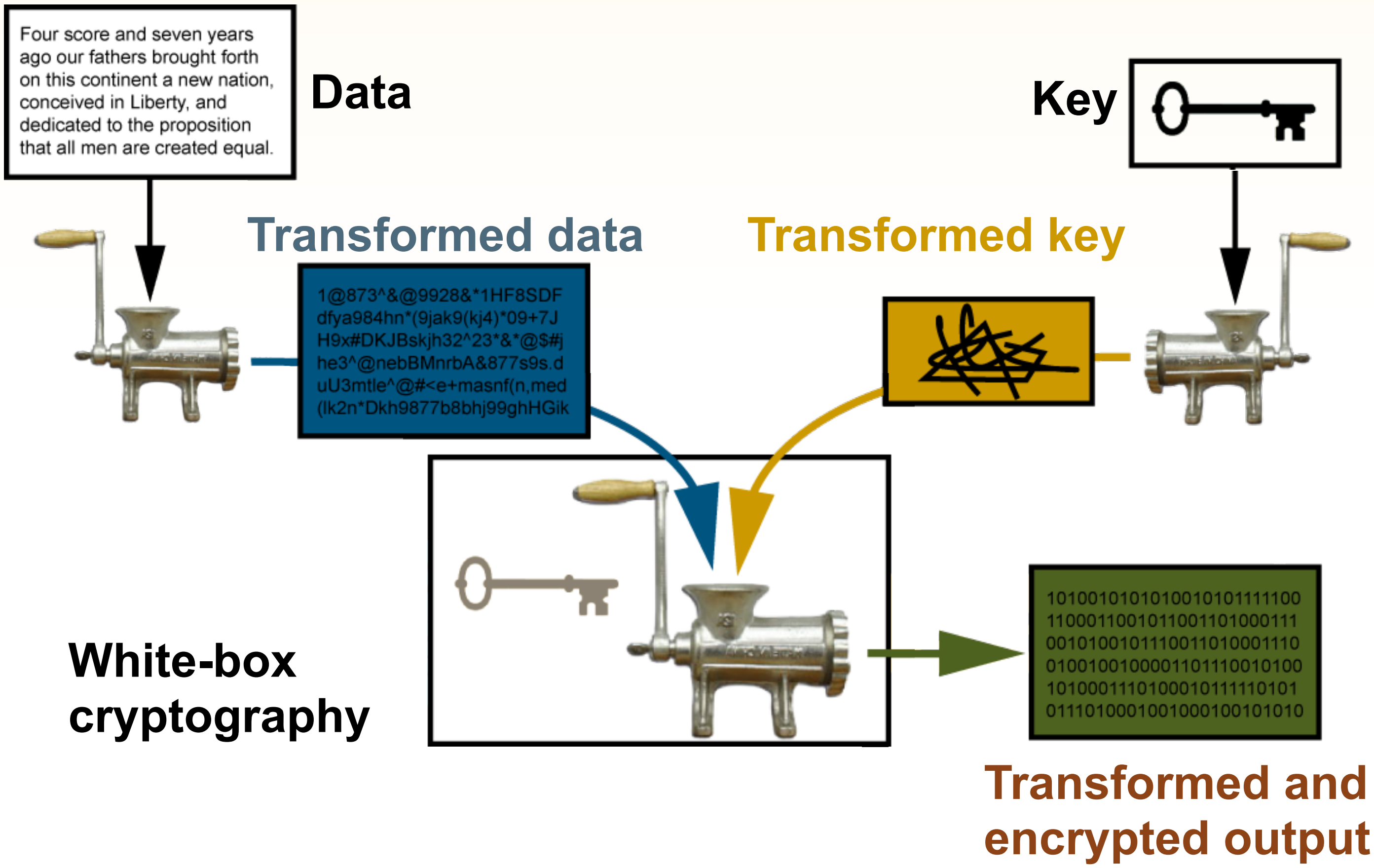
- Timing analysis
- Power analysis
- Fault injection

Black-box

Input / output

White-Box Cryptography

White-box cryptography ensures the input data, keys and resulting output data are protected at all times



Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Data

Key

Transformed data

```
1@873^&@9928&*1HF8SDF
dfya984hn*(9jak9(kj4)*09+7J
H9x#DKJBskjh32^23*&*@$#j
he3^@nebBMnrbA&877s9s.d
uU3mtle^@#<e+masnf(n,med
(lk2n*Dkh9877b8bhj99ghHGik
```

Transformed key

White-box cryptography

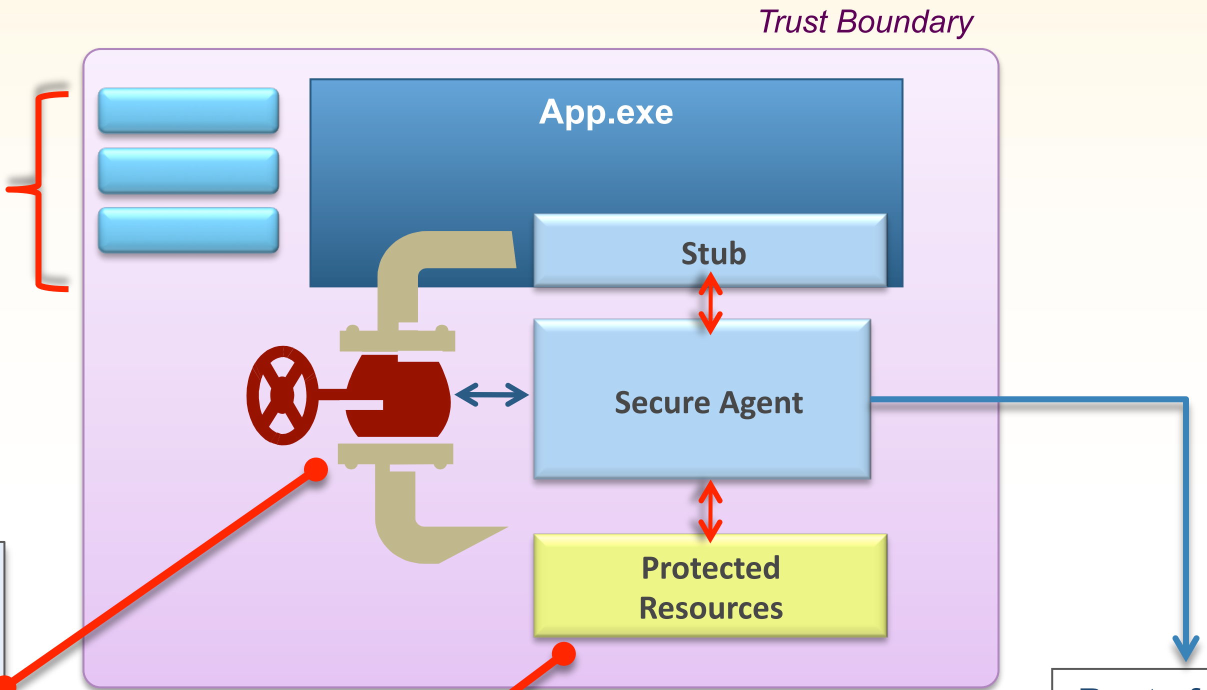
```
10100101010100101011111100
11000110010110011010000111
00101001011100110100011110
01001001000011011100101000
10100011101000101111110101
01110100010010001001010101
```

Transformed and encrypted output

Trust Boundary

Platform Components

- Program launcher
- Digital certificates
- Boot loader
- Hardware anchors
- Drivers
- Firmware



On/Off Decisions :

- Platform Integrity
- Node-lock
- Application integrity
- Resource integrity
- Debugger checks

Controls execution of the application:

- Resource files
- Access to online server

Root of Trust

Options

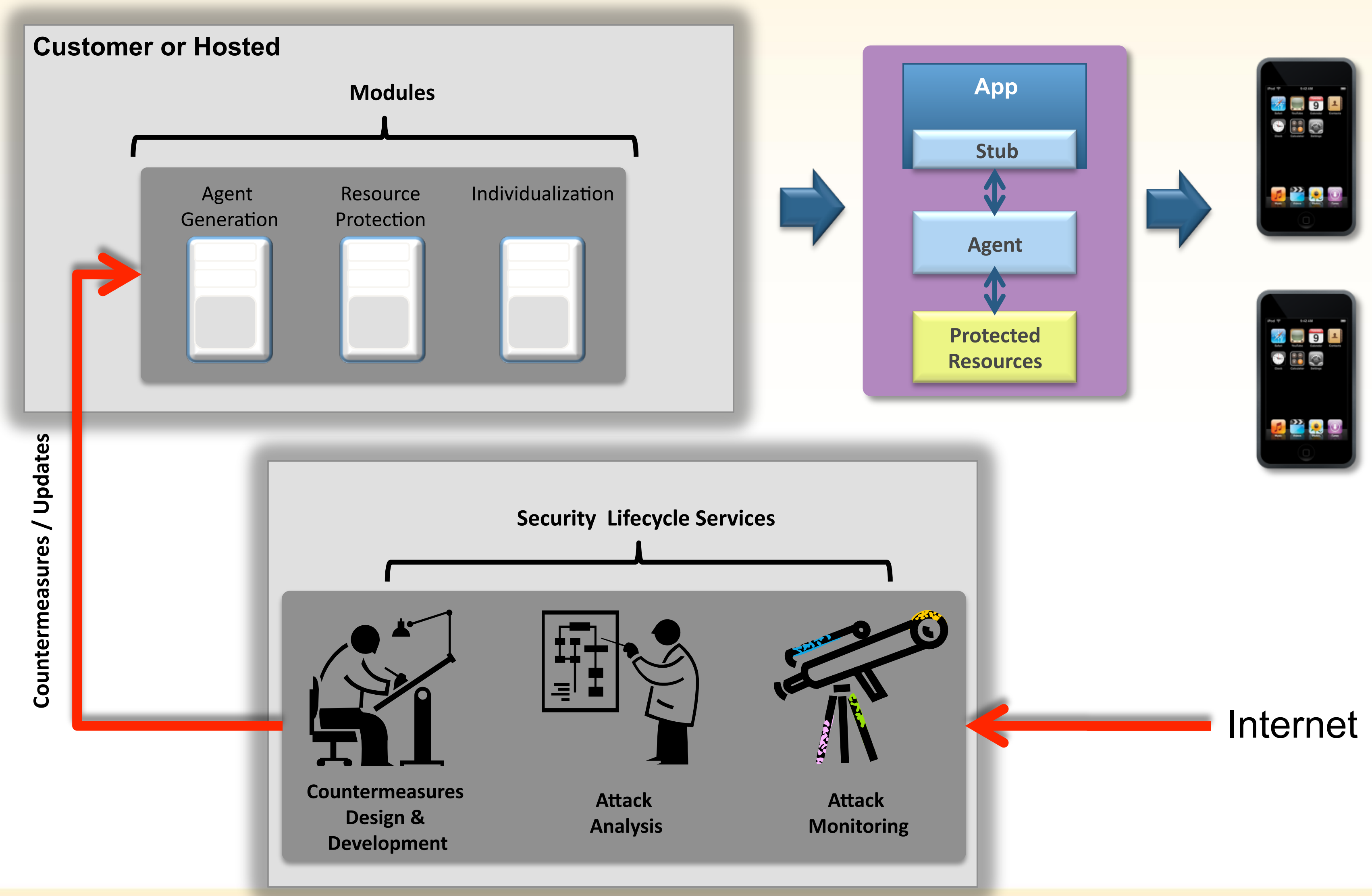
Secure Hardware

Device Fingerprint

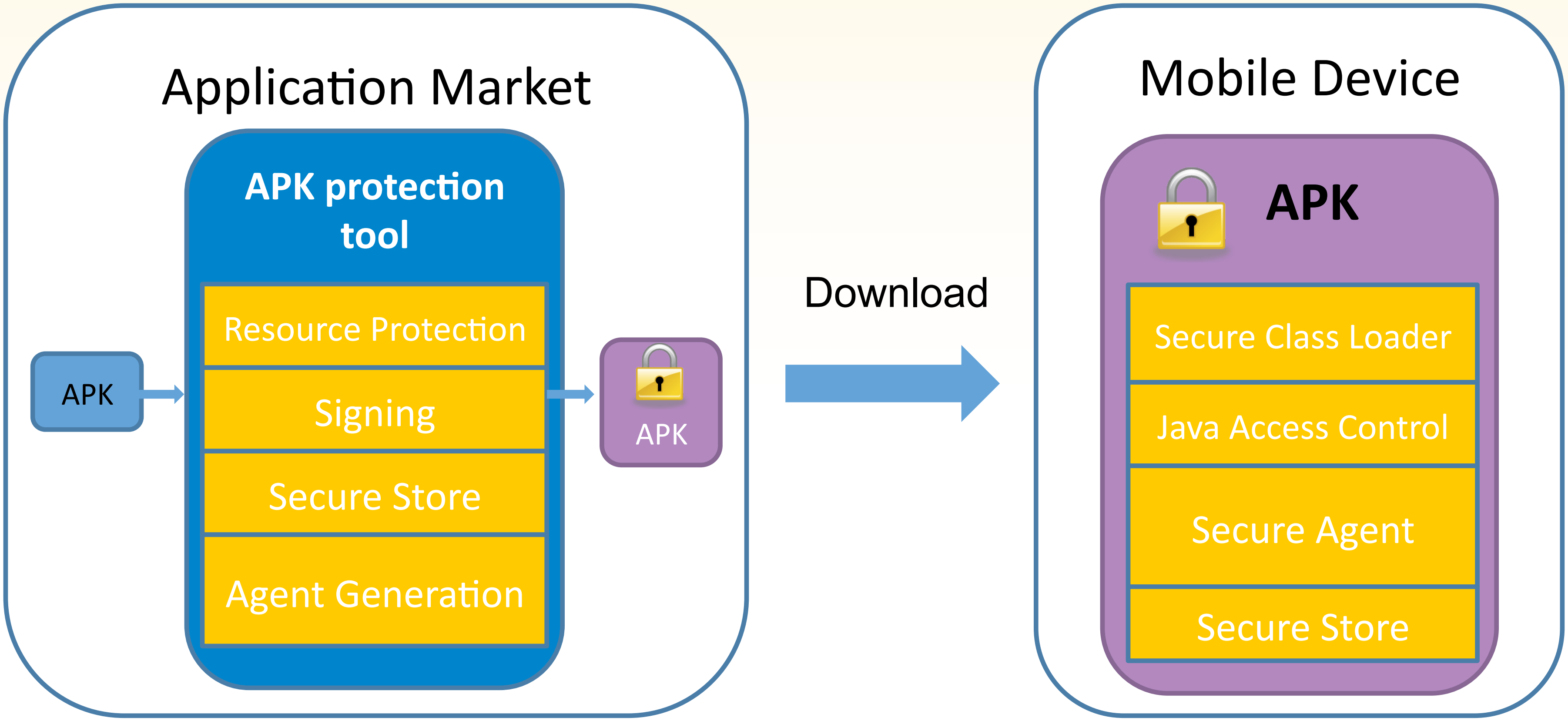
Secure MicroSD flash

Varying degrees of trust are supplemented by 2 way communications with the “head-end”

Deployment Model



Run-time Application Protection

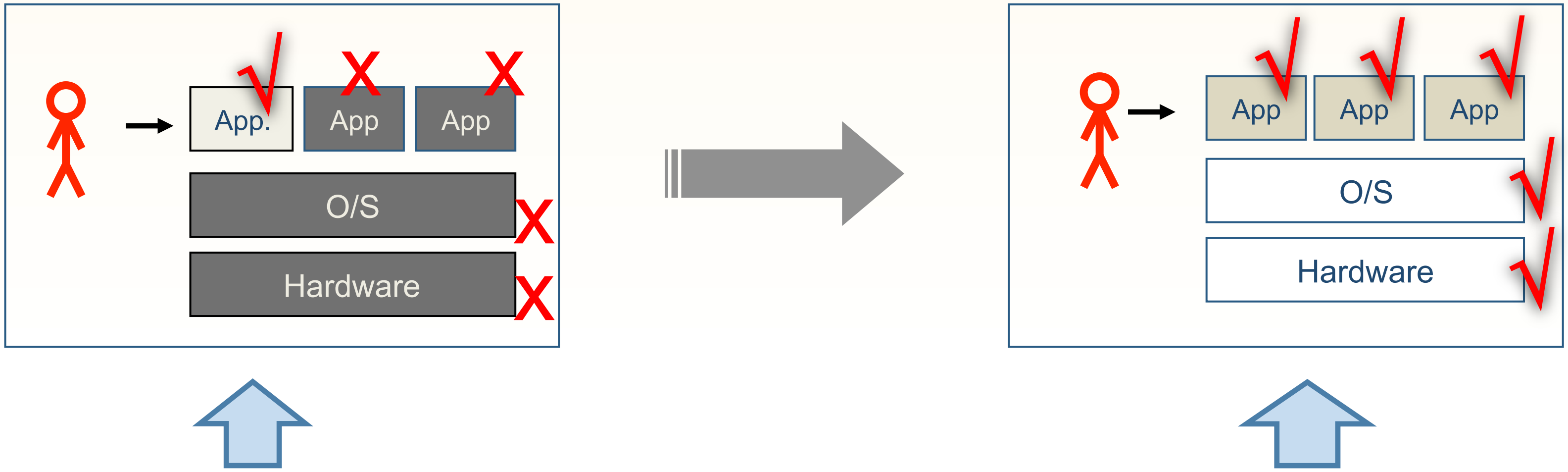


- Google Play's **App Encryption** mechanism on Jelly Bean (Android 4.1) doesn't provide **"Post-download" App Protection**!

Towards Platform Security

Application Protection

Platform Security



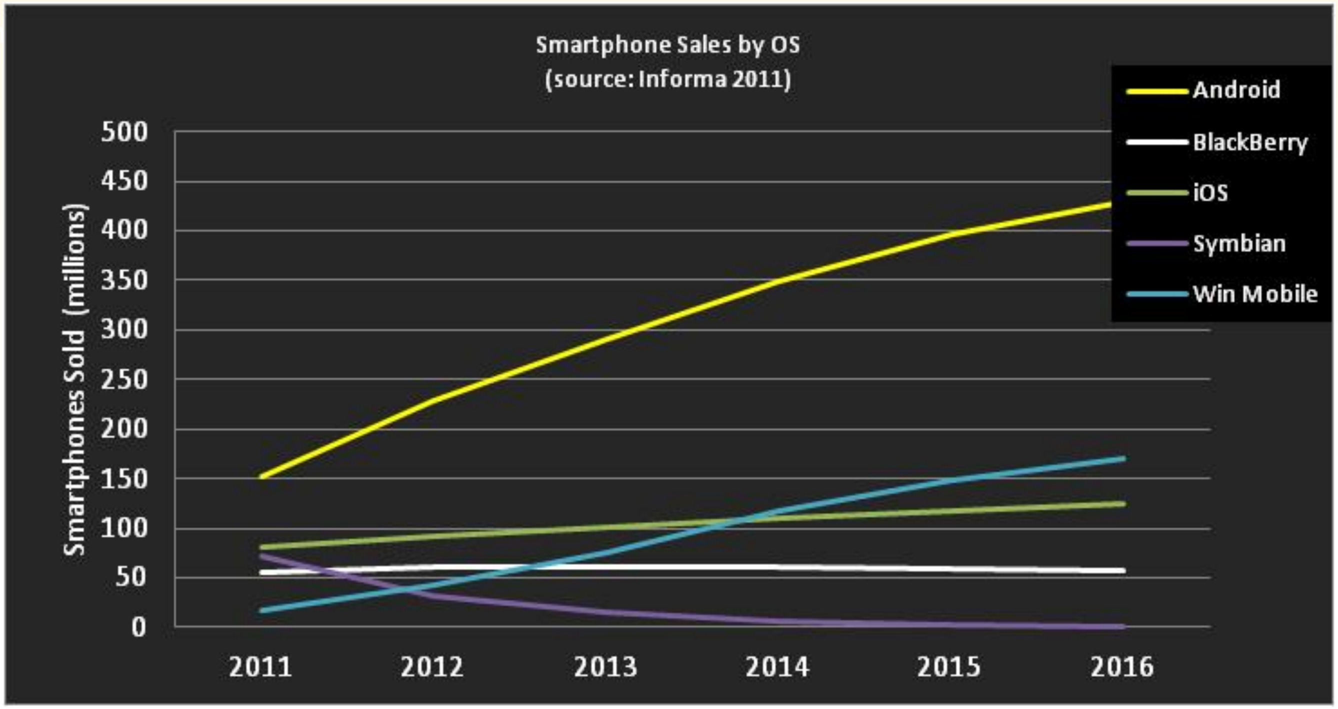
- Applications are protected one by one
- Protected applications are trusted;
- Unprotected applications are not trusted;
- Whole platform is not trusted.

- An alternative approach is to create a "trusted platform", often a competing approach to Trusted applications
- "Platform Security" indirectly enables Applications security.

Android Platform Security

Android Market

Source: Infor



- Android accounts for 50% of smartphones sold worldwide
- 300 million cumulative units activated including 12 million tablets
- Current activation rate is 850 thousand units/day (310 million annualized)
- Large deployment of Android makes it a target for hackers

Android Security Challenges

Malicious Software

- Consumer malware
- Advanced threats
  - Rootkits
  - Botnets
  - Spyware
- Compromised Apps

Grayware

- Apps that abuse privacy
- Apps that abuse the network
- Adware
- Tethering Apps
- Hijacked Apps
- Attack tools
- Censored Apps

Apps

Distributed by:

- Mobile operators
- Android market
- Third party app stores
- Other

Trusted Apps / Enterprise Apps

- Security apps
- Payment / mobile commerce apps
- Customer support apps
- Email apps
- Salesforce apps

Malicious Software / Grayware

Affects: MNO’s, Consumers, Enterprises

Impact: Network outages, customer support costs, consumer privacy, service fraud, brand

Security Challenges;

- Detect Malicious Apps and Grayware
- Preventing Malicious Apps and Grayware
- Protecting the security functions that prevent these threats

Apps/ Trusted Apps/ Enterprise Apps

Affects: MNO’s, Enterprises

Impact: ability to deploy new apps/services, consumer privacy, brand

Security Challenges:

- Prevent Trusted Apps/Enterprise Apps and data from being compromised
- Prevent piracy / hijacking of Apps
- Protecting the security functions that enable the above

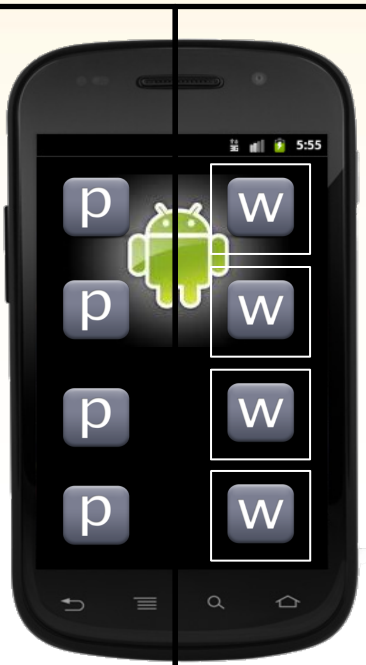
Android Security Approaches

- Google “Bouncer”
  - *Once an application is uploaded, the service immediately starts* ***analyzing it for known malware, spyware and trojans****. It also looks for* ***behaviors*** *that indicate an application might be misbehaving, and compares it against previously analyzed apps to detect possible red flags. We actually* ***run every application*** *on Google’s cloud infrastructure and* ***simulate how it will run on an Android device*** *to look for hidden, malicious behavior.*
- Dissected by Jon Oberheide and Charlie Miller, on SummerCon’12.
  - It uses Linux + Cloud + Simulation (QEMU)
  - It will catch crappy malware, it won't catch sophisticated malware
- Many other security approaches:
  - App censorship tools: RiskRanker, jointly by NCSU and NQ Mobile.
  - Mobile AV, security management solutions from security vendors.
  - Research works on Android permission models in Academia.

BYOD

Personal Domain

- User can download/use apps from anywhere
- Corporate IT cannot access apps and data in the Personal domain via the management interface
- If the device is lost or stolen
    - User can locate / lock / wipe personal data



Work Domain

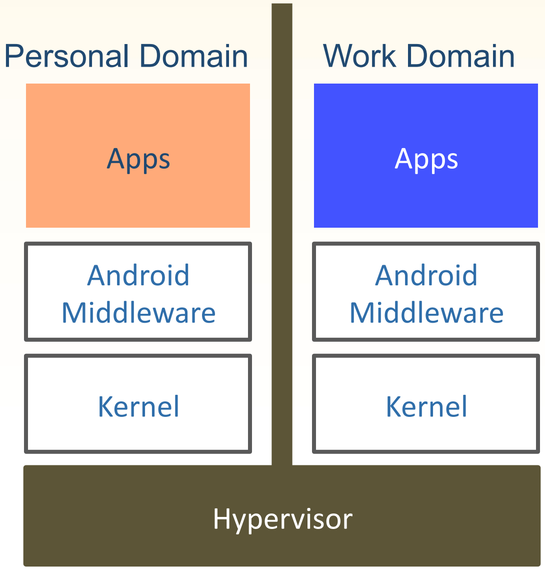
- Work apps authorized by corporate IT
- Corp. IT set policy for apps/data
- Prevent loss of data (emails, contacts, SMS, other) via
    - Malware that infects the Personal Domain
    - Lost/stolen device
    - Removal or loss of SD card containing confidential data
- If the phone is lost or stolen:
    - Work data is encrypted
    - Work domain can be remotely wiped by corporate IT

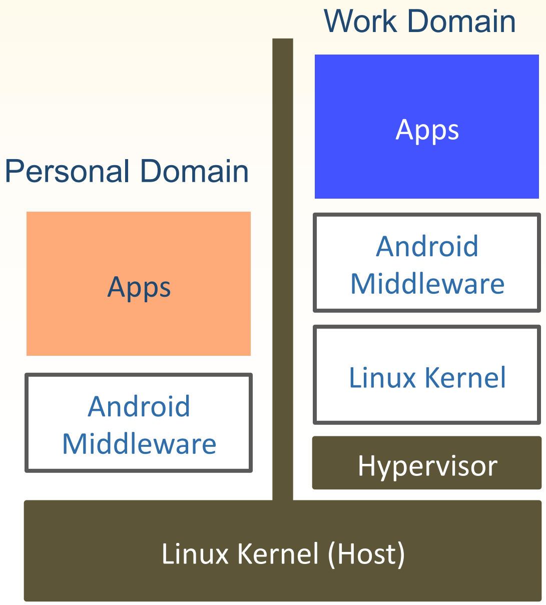
Seamless transition between Work and Personal apps/domains

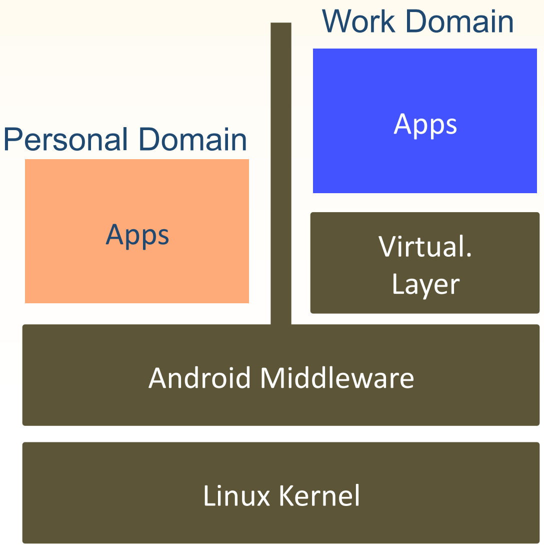
Low device overhead

Ease of integration / deployment

Domain Isolation Approaches







Type 1 Hypervisor

- Not currently supported by the ARM instruction set
- OEM Integration is an issue
- e.g. Redbend

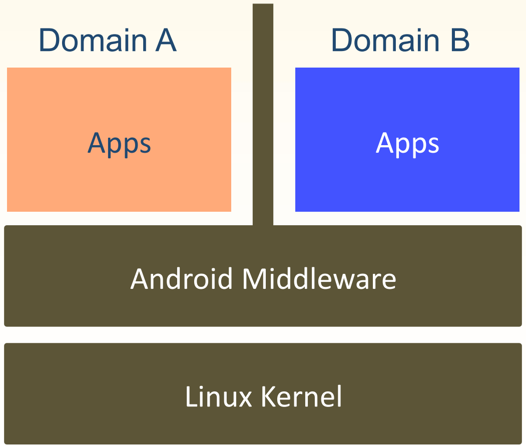
Type 2 Hypervisor

- Exposed to kernel layer attacks
- Less integration required with OEM
  - Integration can occur later in the development cycle
- e.g. VMware Mobile Horizons

OS- Level VM

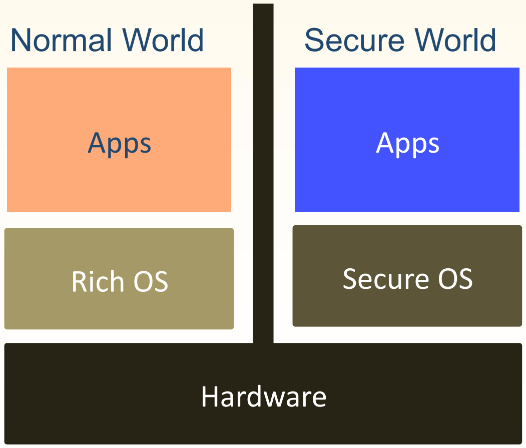
- Exposed to user and kernel layer malware
- Heavy performance penalty for work apps
- Requires work apps to be ported to the VM
- Work app IPC handled by the VM
- e.g. Enterproid Divide

Domain Isolation Approaches



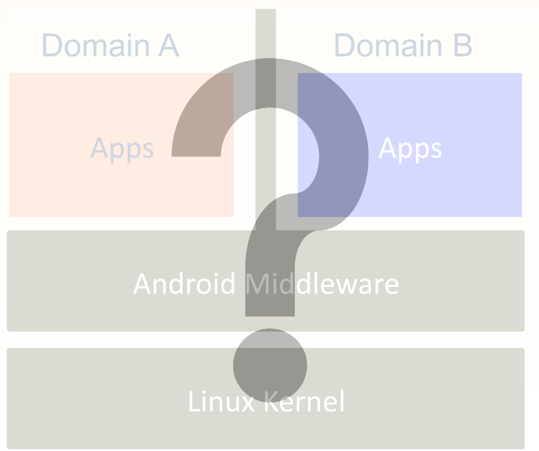
TrustDroid

- Need to modify the Android middleware
- Use Tomoyo Linux
- Exposed to kernel layer attacks
- Low CPU/memory/battery overhead
- Academic work
- Strong assumption on TCB



TrustZone

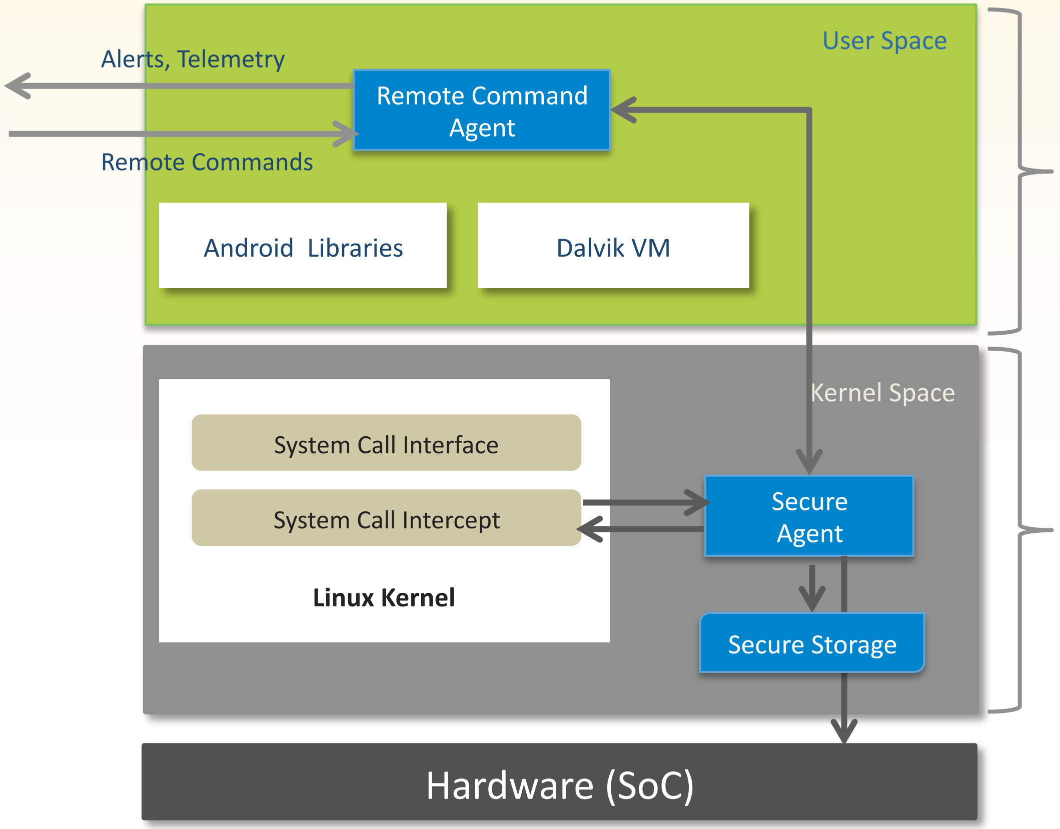
- Leverage hardware security
- Strong security for secure world apps
- Security APIs for apps in Rich OS
- Requires adoption by chip manufacturers, mobile device OEMs
- e.g. ARM TrustZone, TEE



An ideal solution?

- No reliance on hardware
- No duplication of software stack
- Prevent advanced malwares
- Easy integration and adoption
- Small TCB
- Maximize performance
- Minimize overhead
- Seamless switch between domains

Device View



Android

- Standard Android middleware

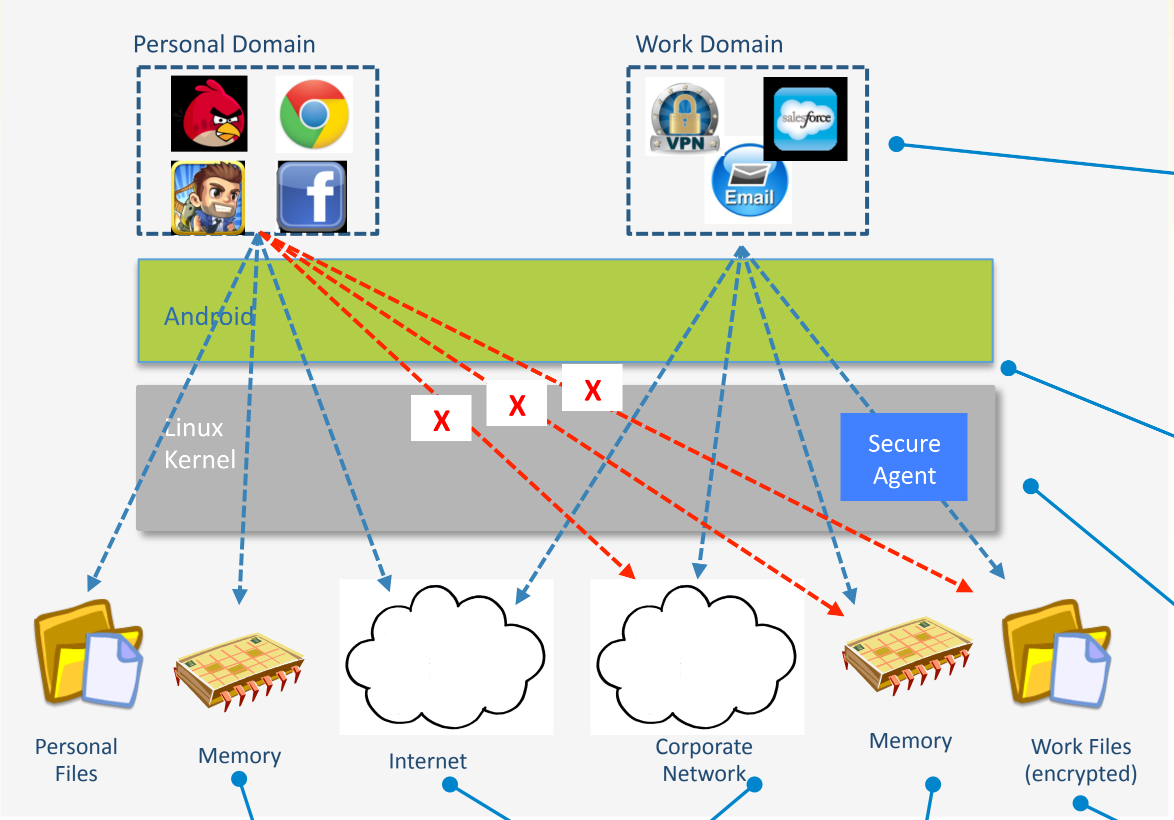
Secure Agent:

- Loadable kernel module
- Controls access to kernel objects and services in accordance with functionality and policies (e.g. app loading, memory access, etc.)
- Prevents rootkit installation
- Monitors kernel integrity
- Verifies integrity of apps and permissions
- Maintains a persistent secure store for policy, signature and telemetry data

Security

- Agents are protected against tampering and reverse engineering
- Communications via secure channels
- All Agents can be diverse to prevent automated attacks
- Built with small TCB
- Secure anchoring to hardware SoC

Domain Isolation



Work Domain white list of apps authorized by corp. IT.
Use of Work apps can be tied to:

- User session
- Presence of a SIM card
- Kernel integrity
- Other (e.g. location, integrity verification of apps)

IPC channels between domains are blocked

Domain isolation can be enforced if malware in the Personal Domain gains root access with Mandatory Access Control (MAC). Advanced threats such as rootkits are prevented.

Corporate Network are only accessible to apps in the Work Domain. Access to the Internet by Work Domain apps can also be controlled.

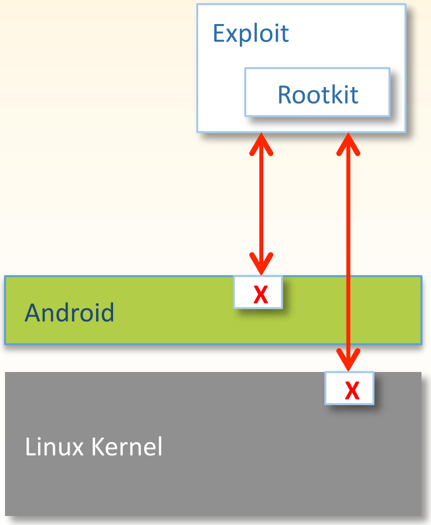
Work Files are encrypted and not visible to apps in other domains and vice versa.

Memory is protected against snooping (e.g. malware with root access, memory dumps, debuggers)

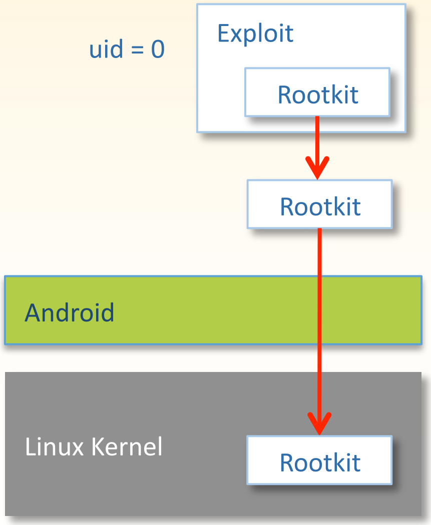
Kernel Malware and Rootkits

- It is relatively easy for Android malware to get root access by exploiting one of the OS layer vulnerabilities
  - A recent study, *"Dissecting Android Malware: Characterization and Evolution,“ (Oakland 2012),* found that 37% of Android malware samples studied contained root-level exploits and more than 90 percent of malware samples were botnet capable.
  - Root-level exploits are a particular concern because once a rootkit exploit has been installed in an Android device, detection and recovery are particularly difficult.
  - This addresses a fundamental security issue associated with Linux in that it does not enforce access control once a process has root access
- Platform security solution must address kernel layer malware (e.g. malware that gains root access) and kernel rootkits

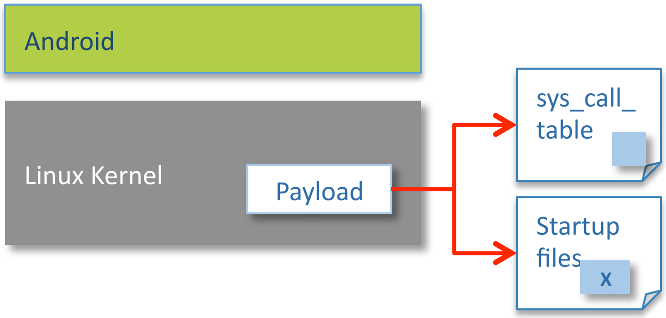
Rootkit Attacks



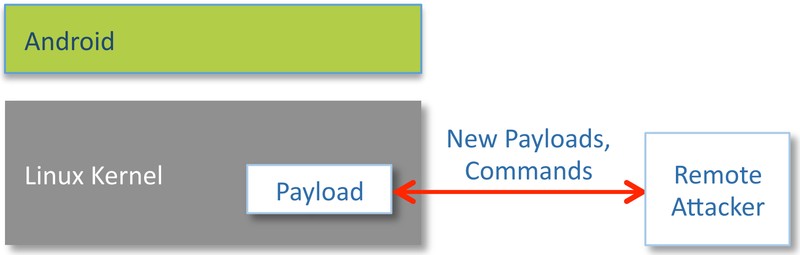
1. **Gain Root Access** – Leverage existing vulnerability to gain root access



2. **Unpack/Install the Payload** – Unpack payload and insert into kernel using LKM or /dev/kmem

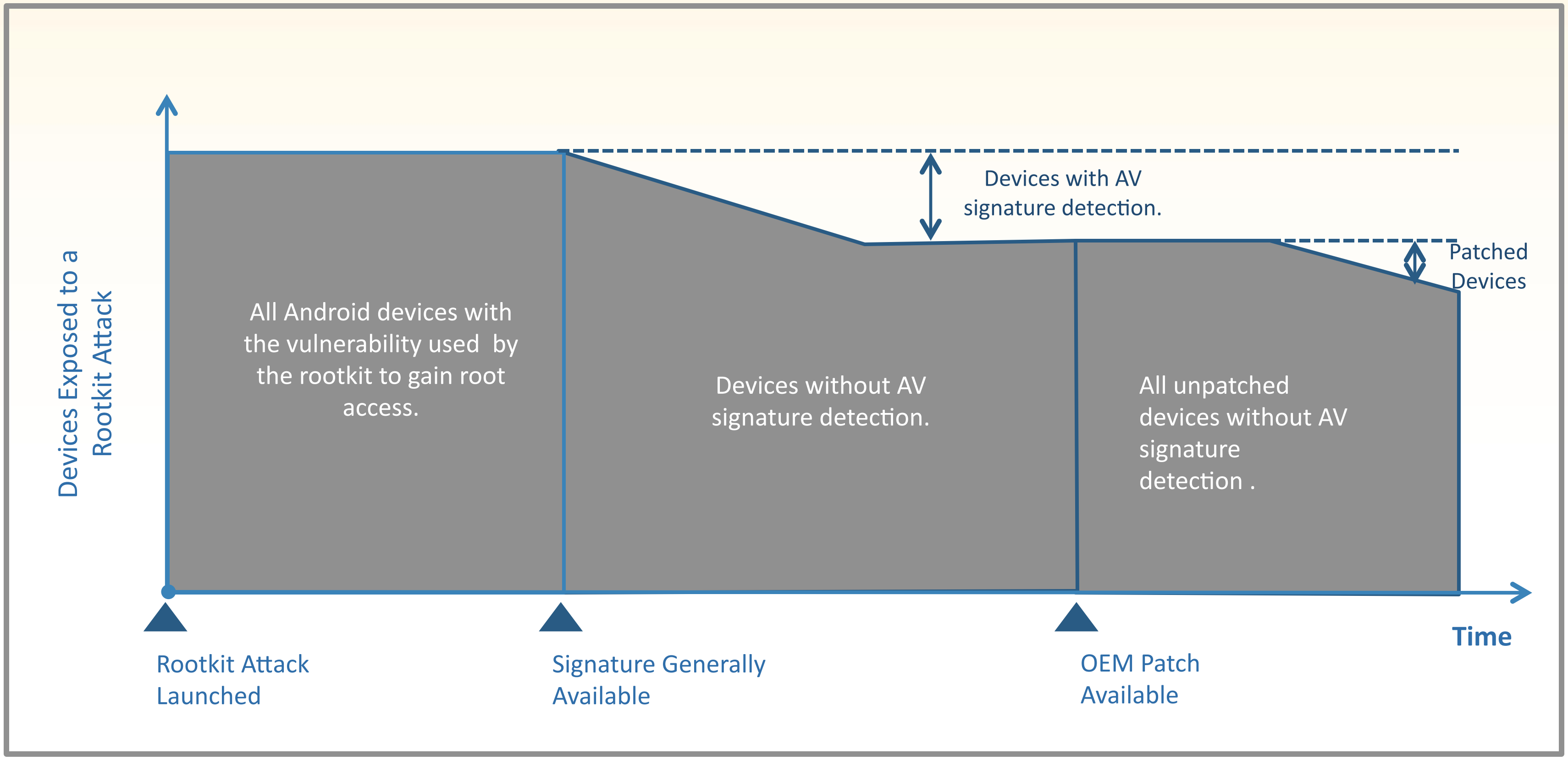


3. **Persistence/Concealment** -- Conceal rootkit and establish permanence by modifying sys_call_table and startup files.



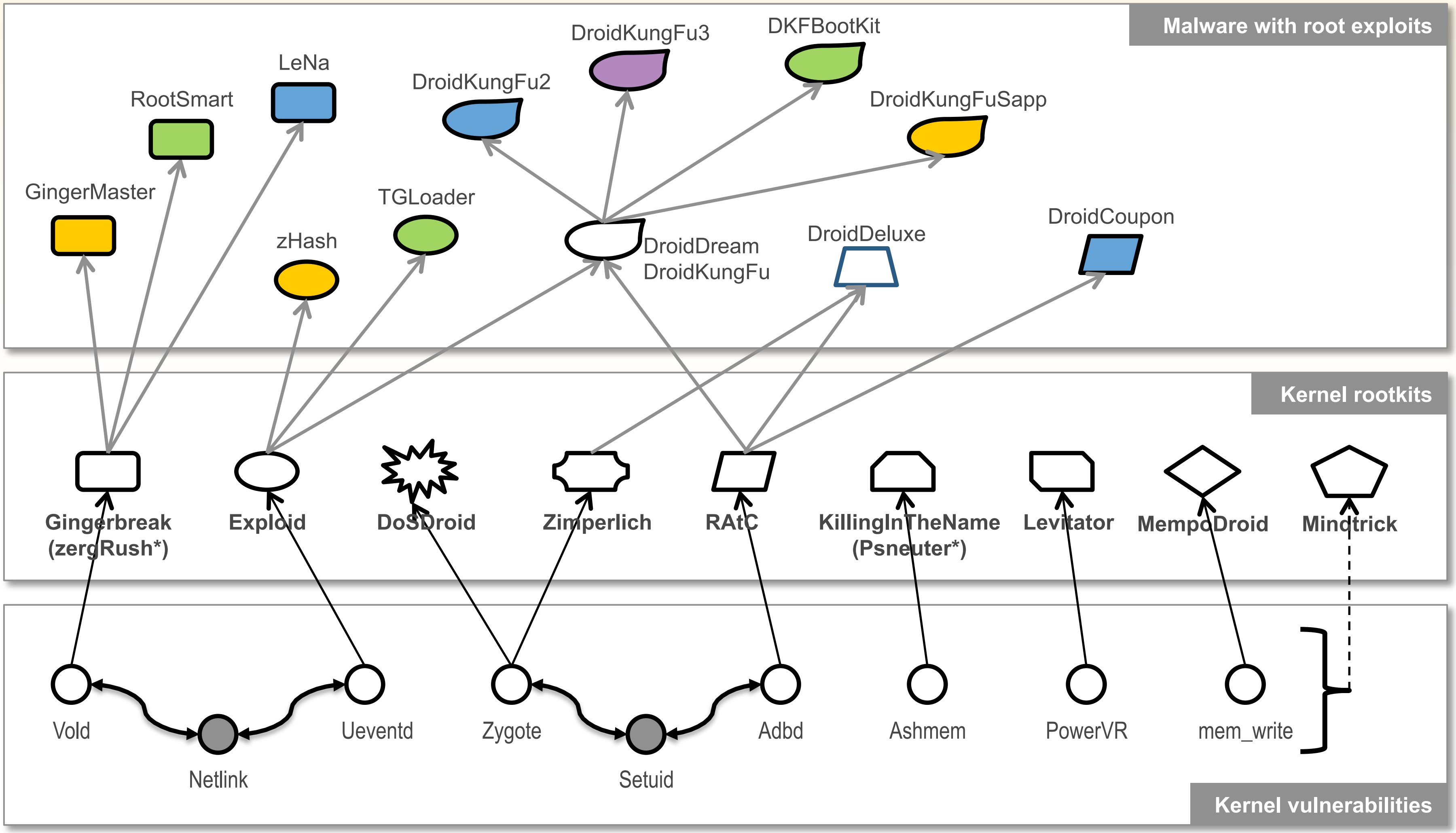
4. **Attack** – At this point the attacker owns the device and can do whatever he or she wishes to do remotely :
   - Install new software
   - Monitor all communications
   - Access the camera and microphone
   - Kill the device, etc.

Exposure to Rootkit Attacks


Devices Exposed to a Rootkit Attack
All Android devices with the vulnerability used by the rootkit to gain root access.
Devices without AV signature detection.
All unpatched devices without AV signature detection .
Devices with AV signature detection.
Patched Devices
Time
Rootkit Attack Launched
Signature Generally Available
OEM Patch Available

- Rootkit attacks often do the most damage when they are initially launched
- Most mobile devices don’t have AV scanners so the exposure window can be lengthy

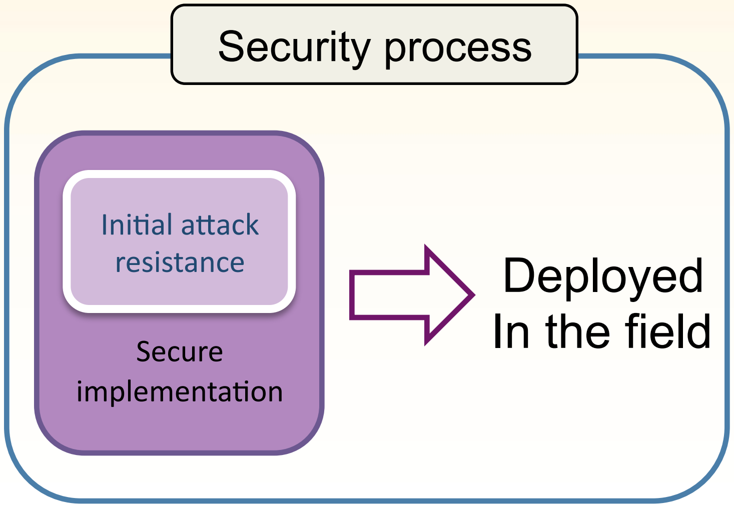
Android Rootkit Families

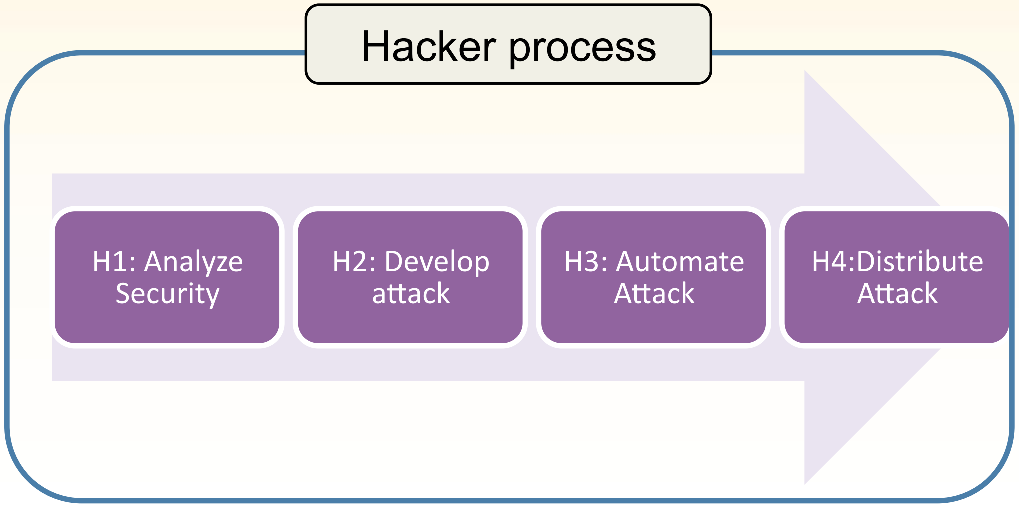


Note. Mindtrick assumes the device has been rooted so could leverage any of the known kernel vulnerabilities to gain root.

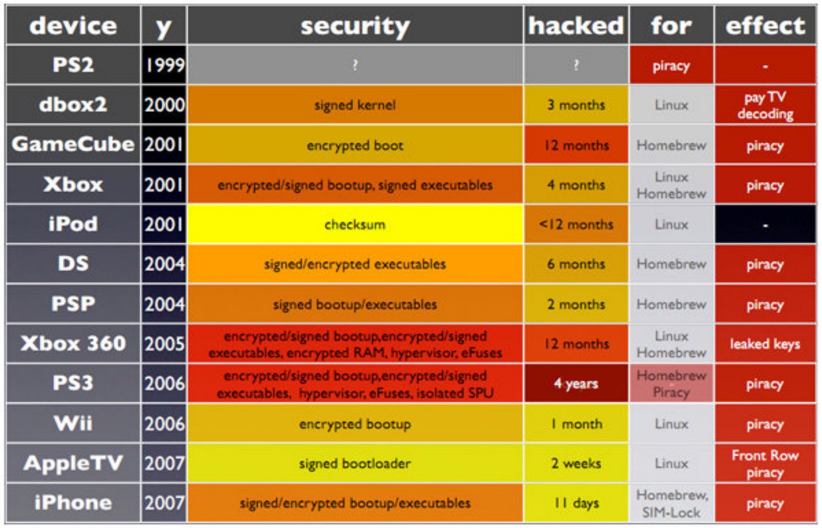
Dynamic & Full Lifecycle Security

Static Security



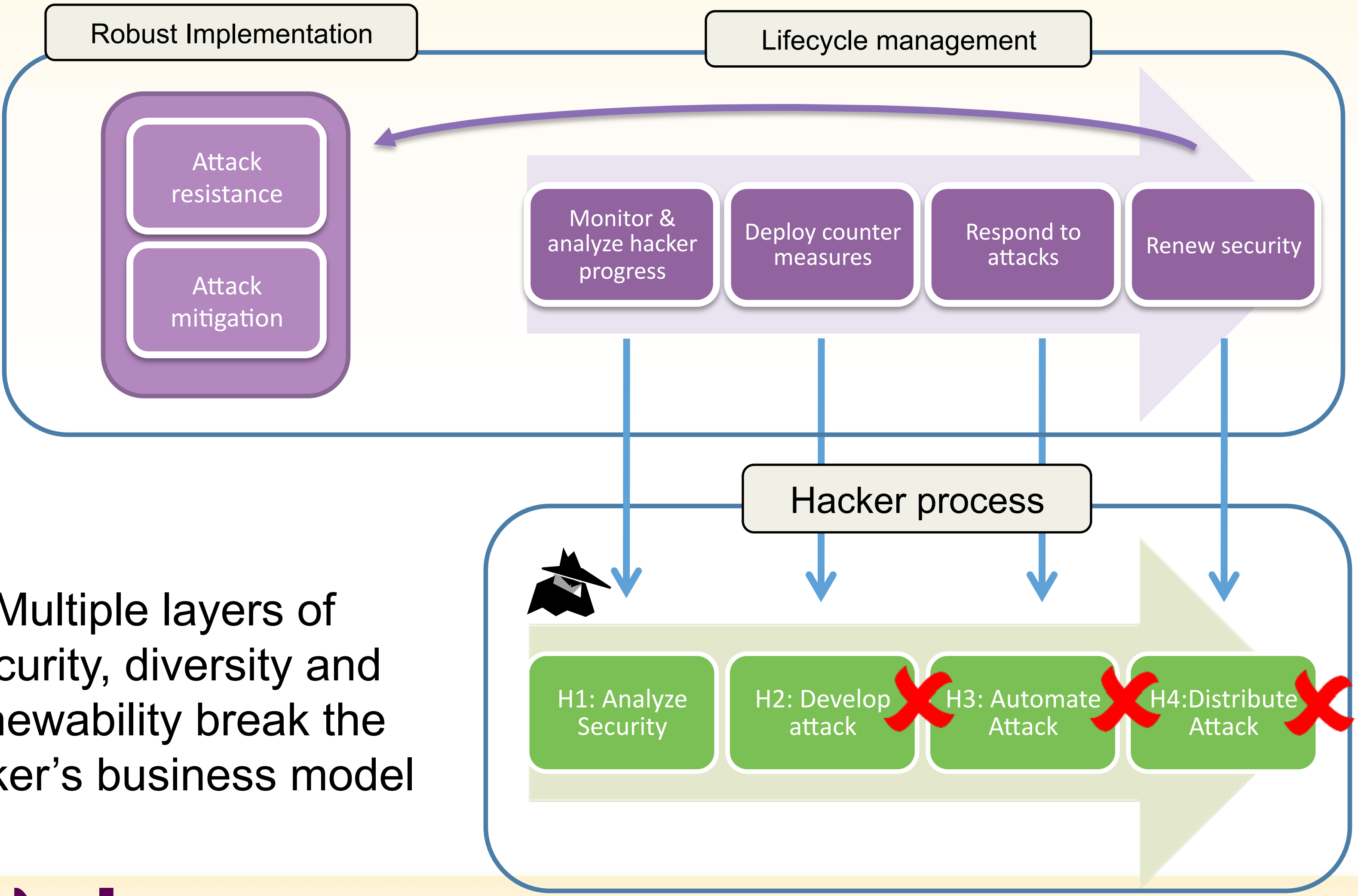


- Focus on initial resistance
- There will be a crack:
  - All static security solutions – even strong ones- in the market are compromised (see table)



device	y	security	hacked	for	effect
PS2	1999	?	?	piracy	-
dbox2	2000	signed kernel	3 months	Linux	pay TV decoding
GameCube	2001	encrypted boot	12 months	Homebrew	piracy
Xbox	2001	encrypted/signed bootup, signed executables	4 months	Linux Homebrew	piracy
iPod	2001	checksum	<12 months	Linux	-
DS	2004	signed/encrypted executables	6 months	Homebrew	piracy
PSP	2004	signed bootup/executables	2 months	Homebrew	piracy
Xbox 360	2005	encrypted/signed bootup,encrypted/signed executables, encrypted RAM, hypervisor, eFuses	12 months	Linux Homebrew	leaked keys
PS3	2006	encrypted/signed bootup,encrypted/signed executables, hypervisor, eFuses, isolated SPU	4 years	Homebrew Piracy	piracy
Wii	2006	encrypted bootup	1 month	Linux	piracy
AppleTV	2007	signed bootloader	2 weeks	Linux	Front Row piracy
iPhone	2007	signed/encrypted bootup/executables	11 days	Homebrew, SIM-Lock	piracy

Dynamic Security



Multiple layers of security, diversity and renewability break the hacker's business model

Security Lifecycle

Pre-Launch

Product Development

Security Design

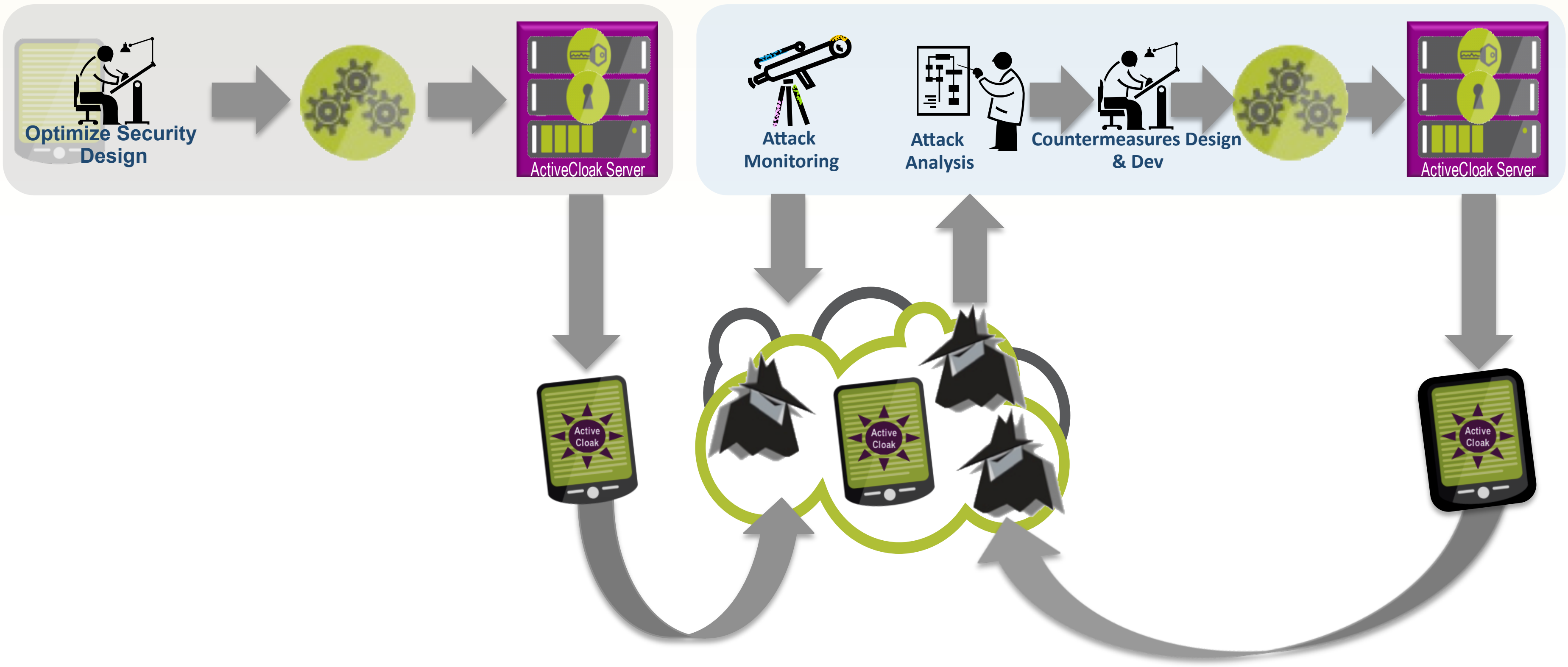
Initial Attack Resistance

Post-Launch

Watch And Defend

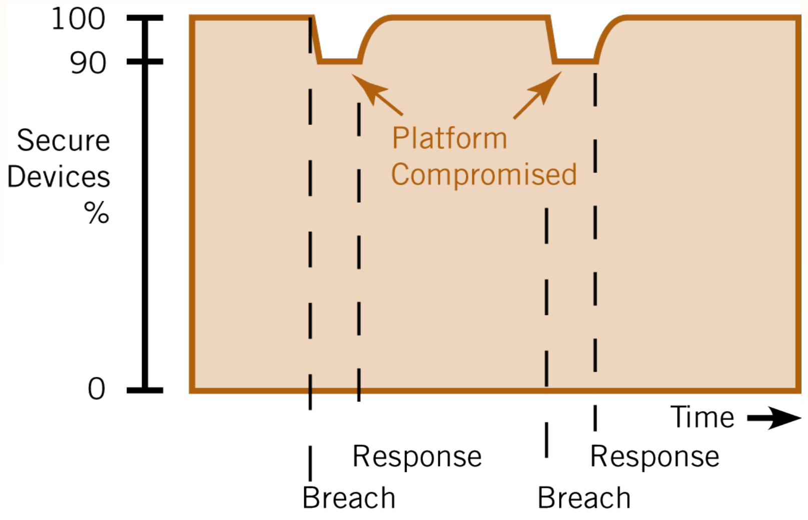
Mitigation Planning

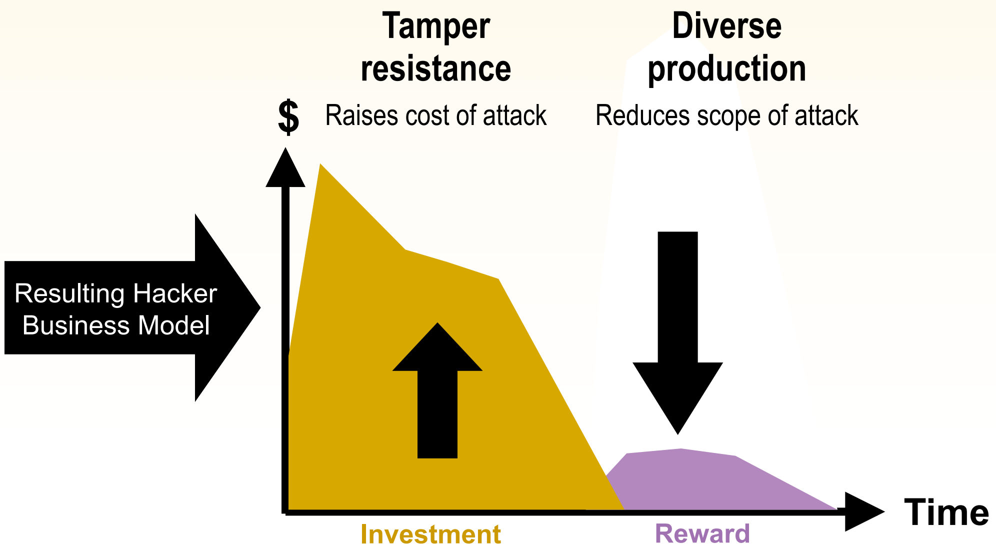
Renewed Attack Resistance



Attack Mitigation and Recovery

- Strong attack response
- Reduces duration of attack





Software Diversity Benefits

Minimize scope of attack -- Prevent automated attacks

Provide rapid recovery in the event of an attack

Make the business unattractive to the hacker

Summary

1, Openness VS. Security

Mobile Internet needs Open Platform

New malwares are coming Fast and Furious

Innovative Security is demanded

2, Security Strategy

App protection → platform security

Dynamic, Multi-layer, Full lifecycle

Simpler, Closer, Vertical

Thank You!

✉ li.tieyan@irdeto.com