Gov-X Innovation Challenge 2021

Mobile Security

Niel van Rooyen

Head: Information Security(CISO)

Niel van Rooyen

Mobile Security

Background:
With 15 years experience in ICT and Cyber Security space, within the private sector ranging from mining, retail, manufacturing and telecommunication industries, I believe better collaboration between all of these industries and governments specifically around Cyber Security, we will start gaining the required knowledge an have the necessary edge against the ever evolving requirements and threat actors in the "Cyberspace".

Mobile is Everywhere

1. **Mobile is primary**
   91% of mobile users keep their device within arm's reach 100% of the time

2. **Insights from mobile data provide new opportunities**
   75% of mobile shoppers take action after receiving a location based messages

3. **Mobile is about transacting**
   96% year to year increase in mobile cyber Monday sales
   Source: IBM Coremetrics Retail Data –

4. **Mobile must create a continuous brand experience**
   90% of users use multiple screens as channels come together to create integrated experiences
   Source: Time, Inc.

5. **Mobile enables the Internet of Things**
   Global Machine-to-machine connections will increase from 2 billion in 2011 to **18 billion** at the end of 2022
   Source: GSMA, Machina Research

Uniqueness of Mobile…

Mobile devices are shared more often

- Personal phones and tablets shared with family
- Enterprise tablet shared with co-workers
- Social norms of mobile apps vs. file systems

Mobile devices have multiple personas

- Work tool
- Entertainment device
- Personal organization
- Security profile per persona?

Mobile devices are diverse

- OS immaturity for enterprise mgmt
- BYOD dictates multiple OSs
- Vendor / carrier control dictates multiple OS versions

Mobile devices are used in more locations

- A single location could offer public, private, and cell connections
- Anywhere, anytime
- Increasing reliance on enterprise WiFi

Mobile devices prioritize the user

- Conflicts with user experience not tolerated
- OS architecture puts the user in control
- Difficult to enforce policy, app lists

"Why would anyone want to limit the iPhone?"

Mobile Presents Management and Security Challenges

5 in 20 Mobile devices stolen in 2020

155% by which mobile malware increased

70% of Mobile device spam is fraudulent financial services

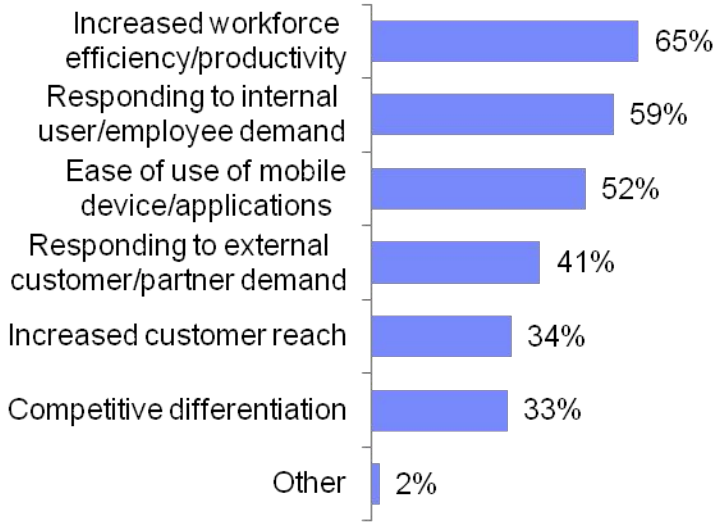
77% growth in Google Android malware

350% by which WiFi hotspots are set to increase by 2020, providing more opportunities for “man-in-the middle” attacks

Billions Android app downloads reached – over 90% of the top 100 have been hacked

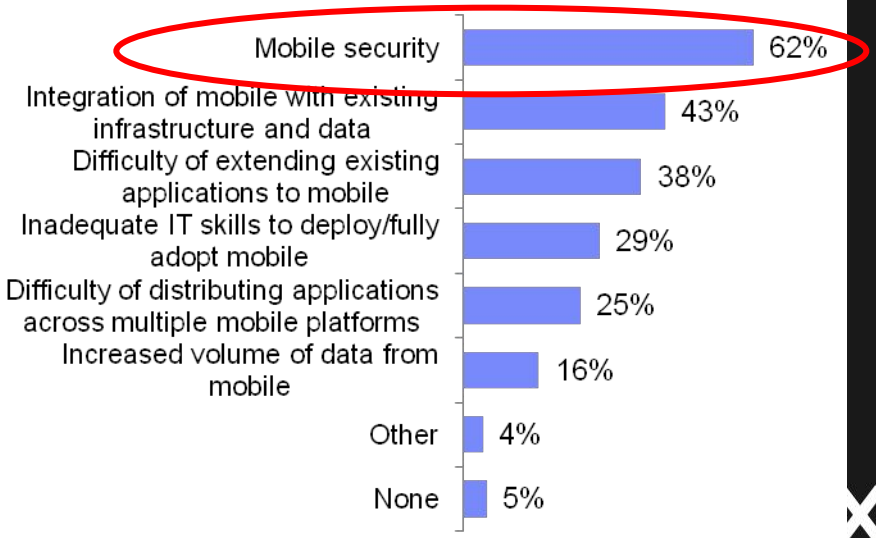
Security is the leading barrier to mobile adoption

Drivers for Adopting Mobile



Base: Those who deployed/piloted/plan to adopt mobile, excluding don't know (n=1117)

Barriers to Adopting Mobile



Base: Those who deployed/piloted/plan to adopt mobile, excluding don't know (n=1115)

Mobile Security Challenges Faced By Enterprises

Achieving Data Separation & Providing Data Protection

- Personal vs corporate
- Data leakage into and out of the enterprise
- Partial wipe vs. device wipe vs legally defensible wipe
- Data policies

Adapting to the BYOD/ Consumerization of IT Trend

- Multiple device platforms and variants
- Multiple providers
- Managed devices (B2E)
- Unmanaged devices (B2B,B2E, B2C)
- Endpoint policies
- Threat protection

Providing secure access to enterprise applications & data

- Identity of user and devices
- Authentication, Authorization and Federation
- User policies
- Secure Connectivity

Developing Secure Applications

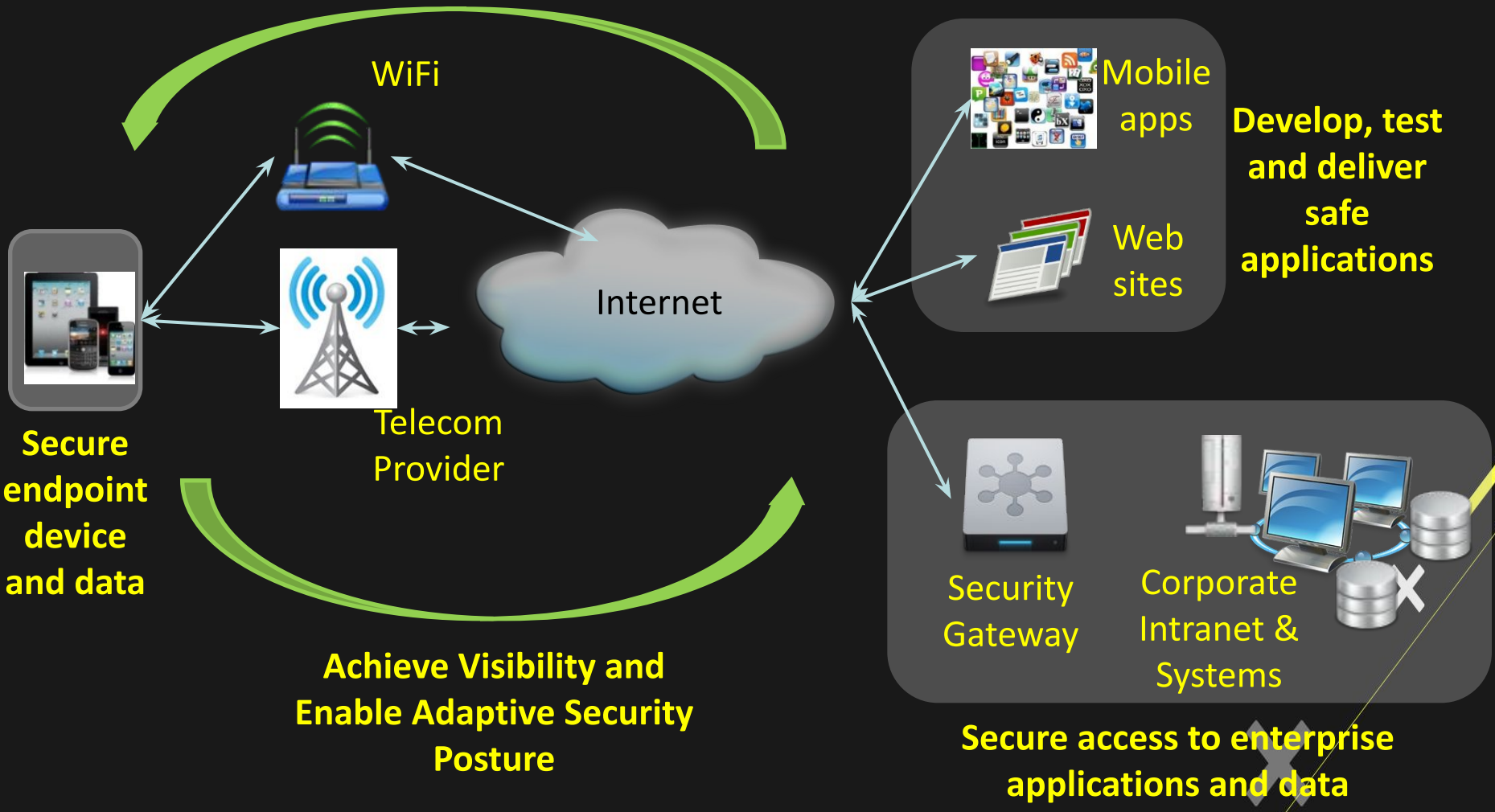
- Application life-cycle
- Static & Dynamic analysis
- **Call and data flow analysis**
- Application policies

Designing & Instituting an Adaptive Security Posture

- Policy Management: Location, Geo, Roles, Response, Time policies
- Security Intelligence
- Reporting

Interrelated

Visualizing Mobile Security



Addressing Security Imperatives and Challenges?

Device Management and Security

How do I handle BYOD and ensure compliance for new devices?

- Multiple device platforms and variants
- Managed devices (B2E)
- Data separation and protection
- Threat protection

Network and Data Management and Security

How do I protect the corporation from data leakage and intrusions?

- Identity management and mobile entitlements
- Policy management and enforcement
- Secure connectivity
- Security intelligence and reporting

Application Management and Security

How do I secure, control and service applications?

- Application lifecycle and performance
- Vulnerability and penetration testing
- Policy management: location, geo, roles, response, time policies

Thinking Through Mobile Management and Security

IBM Mobile Management and Security Strategy

- Management and safe use of smartphones and tablets in the enterprise
- Secure access to corporate data and supporting privacy
- Visibility and security of enterprise mobile platform

At the Device

Enroll
Register owner and services

Configure
Set appropriate security policies

Monitor and Manage
Ensure device compliance and mange Telecom expenses

Reconfigure
Add new policies over-the-air

De-provision
Remove services and wipe

On the Network

Authenticate
Properly identify mobile users

Encrypt
Secure network connectivity

Monitor and Manage
Log network access and events manage network performance

Control
Allow or deny access to apps

Block
Identify and stop mobile threats

For the Mobile App

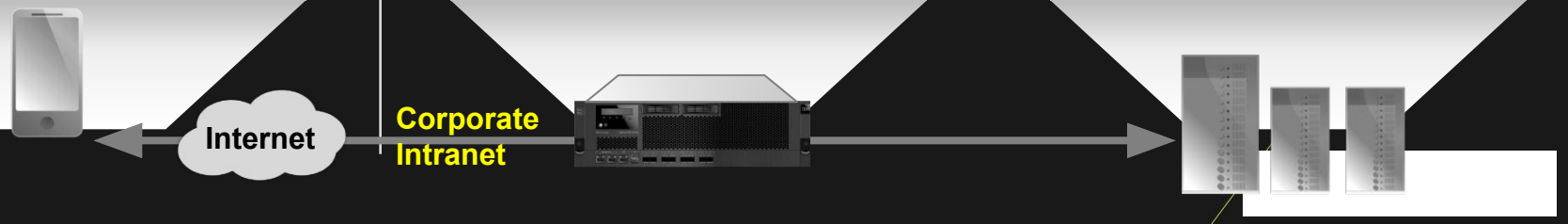
Develop
Utilize secure coding practices

Test
Identify application vulnerabilities

Monitor and Manage
Correlate unauthorized activity and Manage app performance

Protect
Defend against application attacks

Update
Patch old or vulnerable apps



Getting Started with Mobile Security Solutions…

What are your operational priorities?

Business Need:

Protect Data & Applications on the Device

- Prevent Loss or Leakage of Enterprise Data
  - Wipe
  - Local Data Encryption
- Protect Access to the Device
  - Device lock
- Mitigate exposure to vulnerabilities
  - Anti-malware
  - Push updates
  - Detect jailbreak
  - Detect non-compliance
- Protect Access to Apps
  - App disable
  - User authentication
- Enforce Corporate Policies

Business Need:

Protect Enterprise Systems & Deliver Secure Access

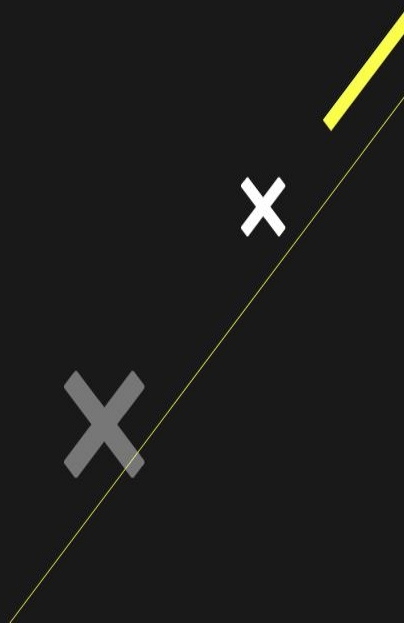
- Provide secure access to enterprise systems
  - VPN
- Prevent unauthorized access to enterprise systems
  - Identity
  - Certificate management
  - Authentication
  - Authorization
  - Audit
- Protect users from Internet borne threats
  - Threat protection
- Enforce Corporate Policies
  - Anomaly Detection
  - Security challenges for access to sensitive data

Business Need:

Build, Test and Run Secure Mobile Apps

- Enforce Corporate Development Best Practices
  - Development tools enforcing security policies
- Testing mobile apps for exposure to threats
  - Penetration Testing
  - Vulnerability Testing
- Provide Offline Access
  - Encrypted Local Storage of Credentials
- Deliver mobile apps securely
  - Enterprise App Store
- Prevent usage of compromised apps
  - Detect and disable compromised apps

Android Security Basics



Android Security Architecture

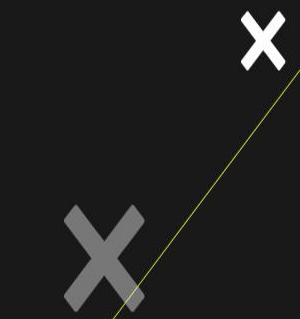
Security goals

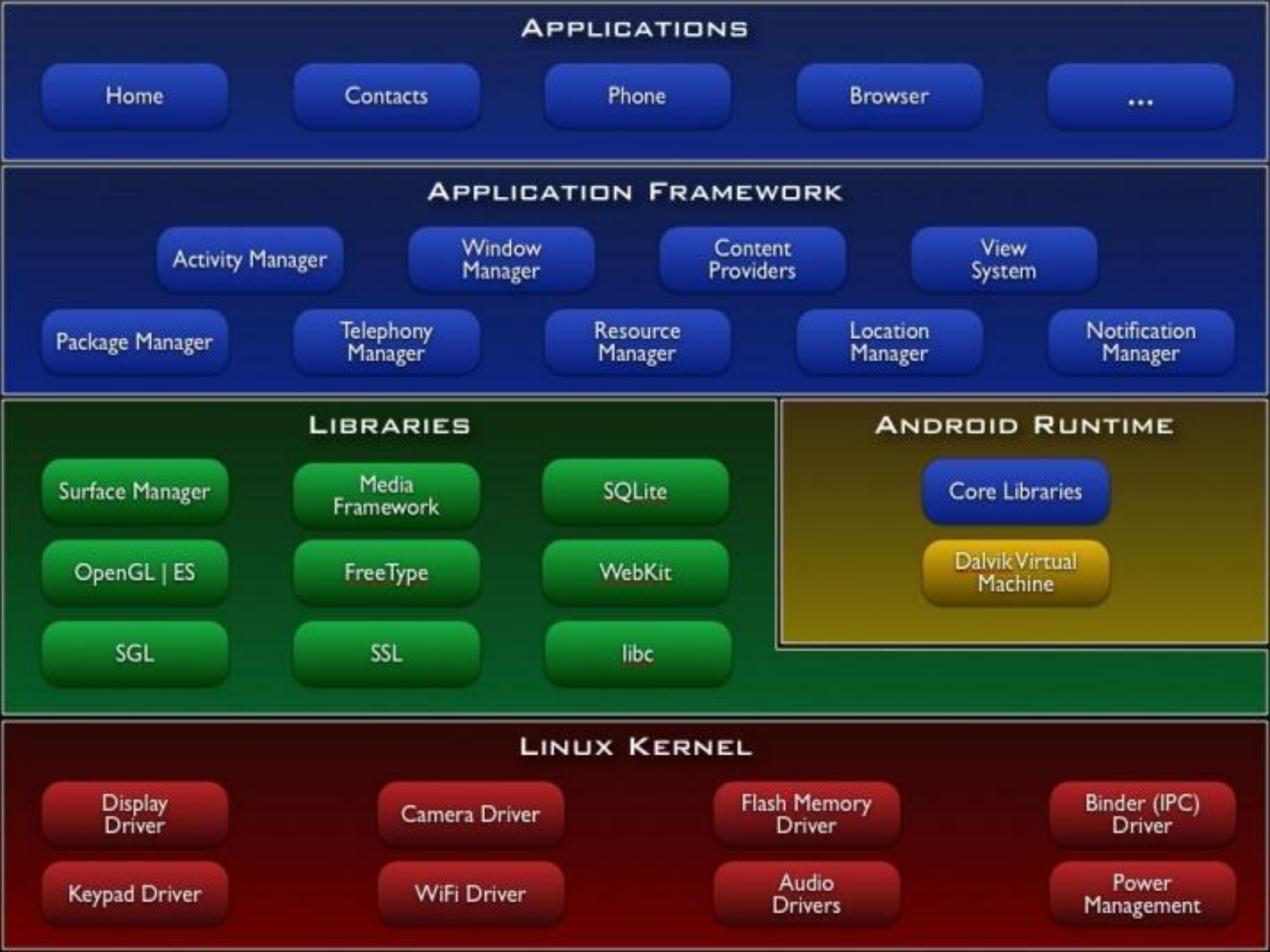
- Protect user data
- Protect system resources (hardware, software)
- Provide application isolation

Foundations of Android Security

Application Isolation and Permission Requirement

- Mandatory application sandbox for all applications
- Secure inter-process communication
- System-built and user-defined permissions
- Application signing




Applications
Home
Contacts
Phone
Browser
...
Application 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 software stack

- Each component assumes that the components below are properly secured.
- All code above the Linux Kernel is restricted by the Application Sandbox
- Linux kernel is responsible sandboxing application
  - “mutually distrusting principals”
  - Default access to only its own data
- The app Sandbox apps can talk to other apps only via Intents (message) , IPC, and ContentProviders
- To escape sandbox, permissions is needed

Security at the Linux kernel

- A user-based permissions model
- Process isolation: Each application has its sandbox based on separation of processes: to protect user resources from each another; each runs in its own Linux process to secure Inter-Process communication (IPC)

Ex:

- Prevents user A from reading user B's files
- Ensures that user A does not access user B's CPU, memory resources
- Ensures that user A does not access user B's devices (e.g. telephony, GPS, Bluetooth)

Application Sandbox

- The Android system assigns a unique user ID (UID) to each Android application and runs it as that user in a separate process.
- When launching a new *Activity*, the new process isn't going to run as the launcher but with its own identity with the permission specified by the developer.
- The developer of that application has ensured that it will not do anything the phone's user didn't intend. Any program can ask Activity Manager to launch almost any other application, which runs with that application's UID.
- Ex. application A is not allowed to do something malicious like to read application B's data or dial the phone without permission.
- All libraries, application runtime, and all applications run within the Application Sandbox in the kernel.

Permissions and Encryption

- **Permissions**

  In Android, each application runs as its own user. Unless the developer explicitly exposes files to other applications, files created by one application cannot be read or altered by another application.

- **Password Protection**

  Android can require a user-supplied password prior to providing access to a device. In addition to preventing unauthorized use of the device, this password protects the cryptographic key for full file system encryption.

Encryption

- Encryption

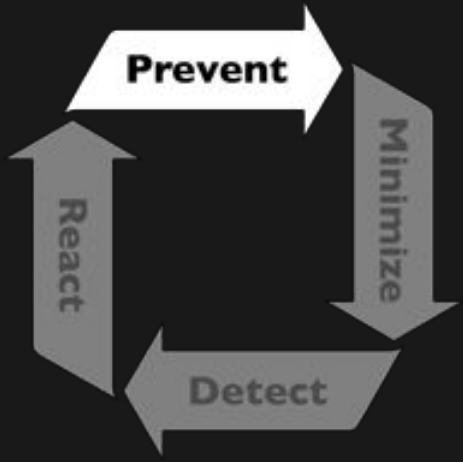
  Android 3.0+ provides full filesystem encryption, so all user data can be encrypted in the kernel

- For a lost or stolen device, full filesystem encryption on Android devices uses the device password to protect the encryption key, so modifying the bootloader or operating system is not sufficient to access user data without the user's device password or BioMetrics.

Cornerstones of Android security

- Prevention
- Minimization
- Detection
- Reaction

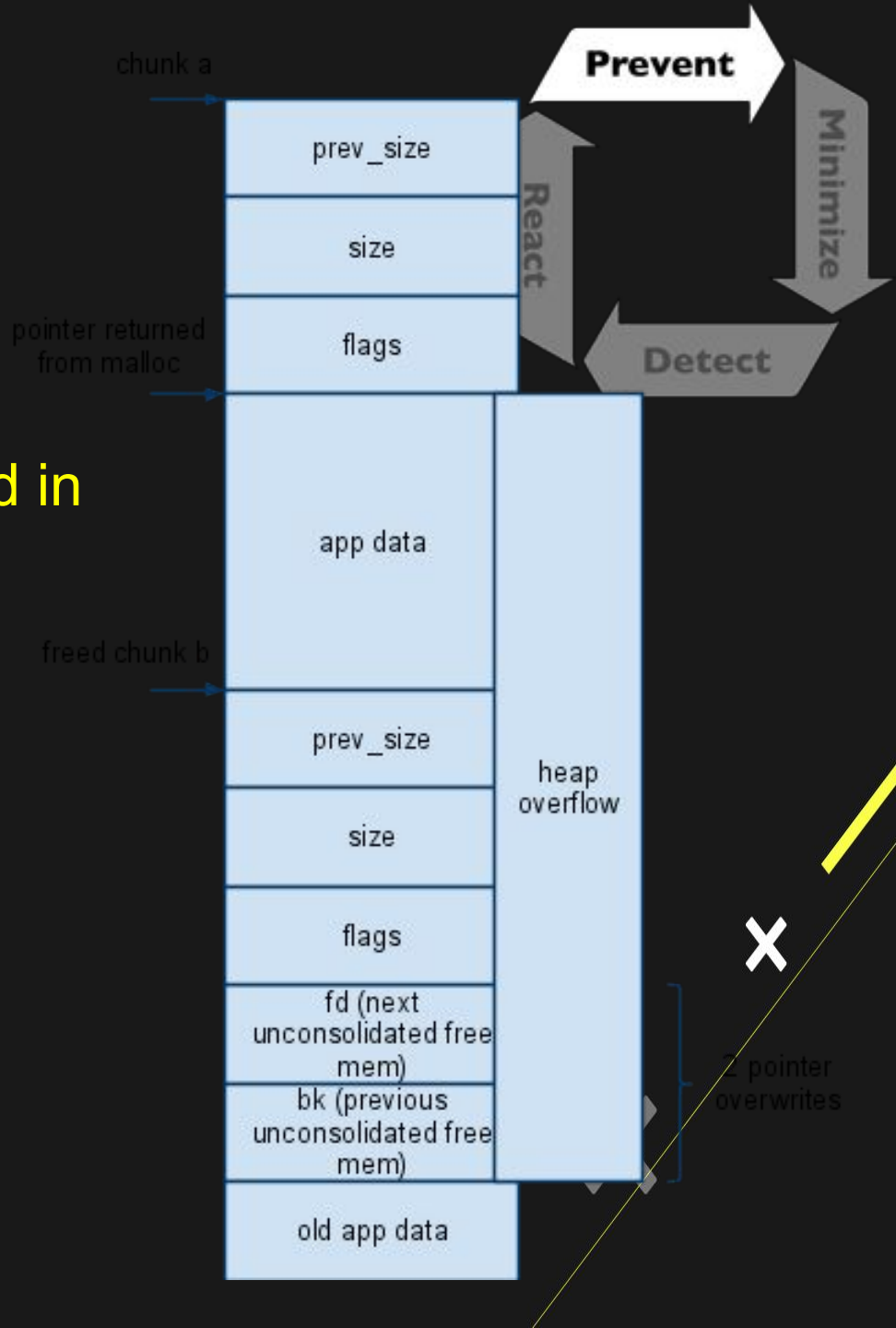
Prevent



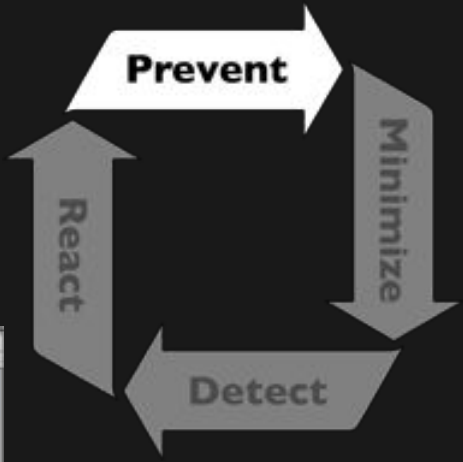
- 5 million new lines of code
- Uses almost 100 open source libraries
- Android is open source ⇒ can't rely on obscurity
- Teamed up with security experts from
  - Google Security Team
  - iSEC Partners
  - n.runs
- Concentrated on high risk areas
  - Remote attacks
  - Media codecs
  - New/custom security features
- Low-effort/high-benefit features
  - ProPolice stack overflow protection
  - Heap protection in dlmalloc

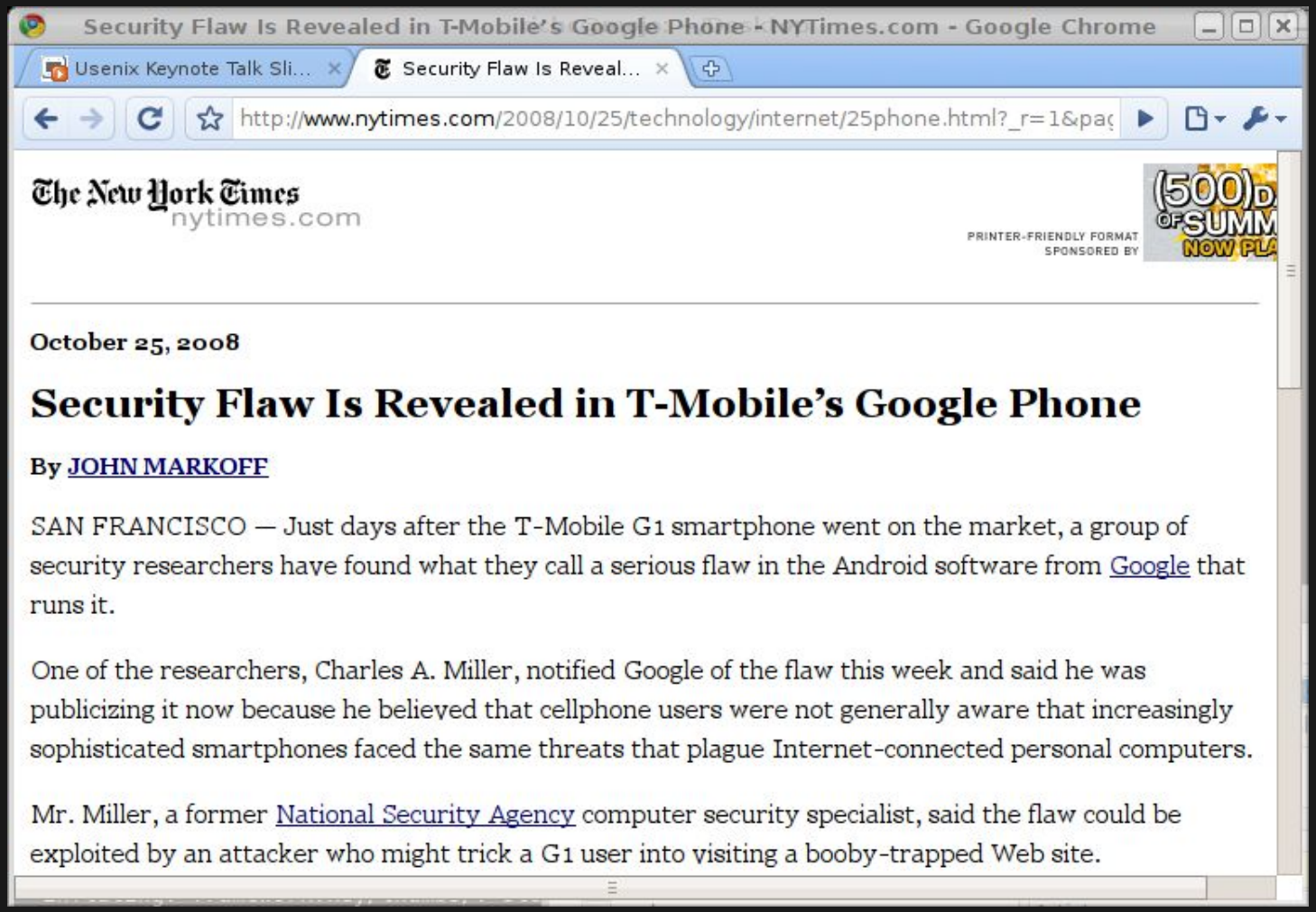
dlmalloc

- Heap consolidation attack
- Allocation meta-data is stored in band
- Heap overflow can perform 2 arbitrary pointer overwrites
- To fix, check:
  - b->fd->bk == b
  - b->bk->fd == b

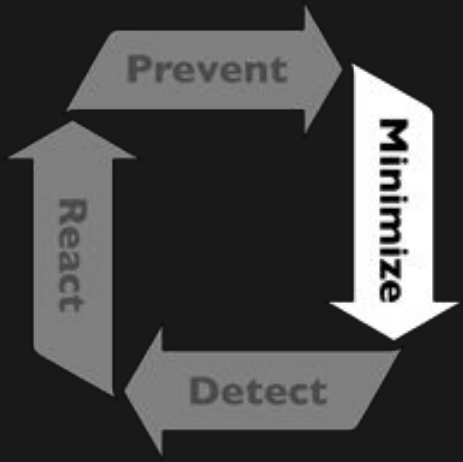


WebKit Heap Overflow



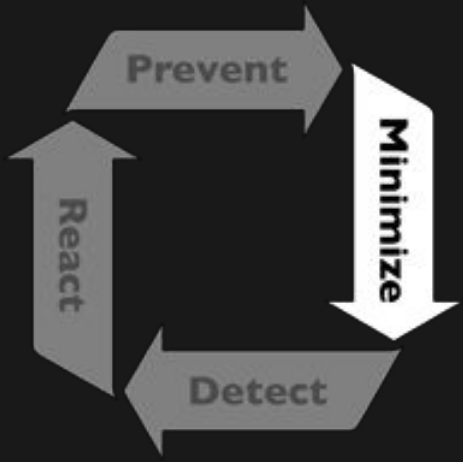


Minimize



- We cannot rely on prevention alone
  - Vulnerabilities happen
- Users will install malware
- Code will be buggy
- How can we minimize the impact of a security issue?
- My webmail cannot access my banking web app
  - Same origin policy
- Why can malware access my browser? my banking info?
- Extend the web security model to the OS

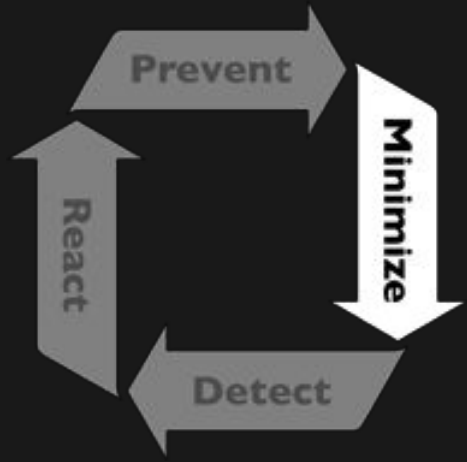
Minimize

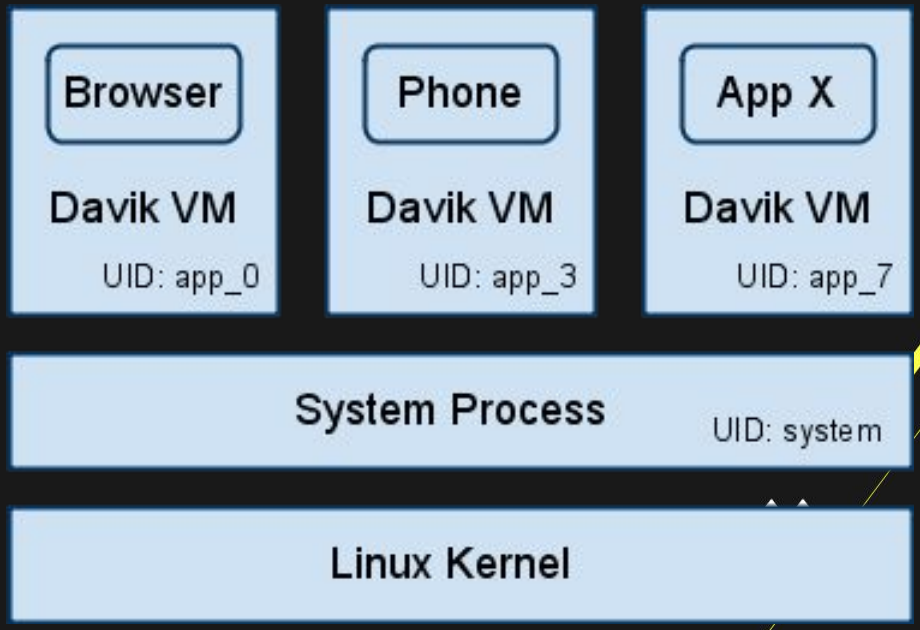


- Traditional operating system security
  - Host based
  - User separation
- Mobile OSes are for single users
- User separation is like a "same user policy"
- Run each application in its own UID is like a "same application policy"
  - Privilege separation
- Make privilege separation relatively transparent to the developer

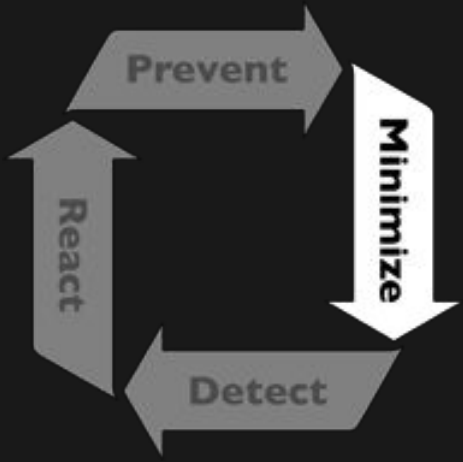
Application Sandbox

- Each application runs within its own UID and VM
- Default privilege separation model
- Instant security features
  - Resource sharing
    - CPU, Memory
  - Data protection
    - FS permissions
  - Authenticated IPC
    - Unix domain sockets
- Place access controls close to the resource, not in the VM



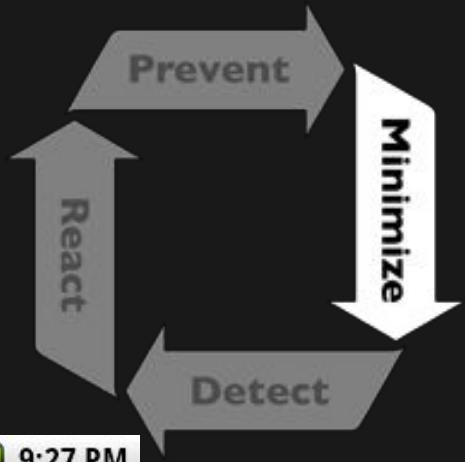


Application Sandbox



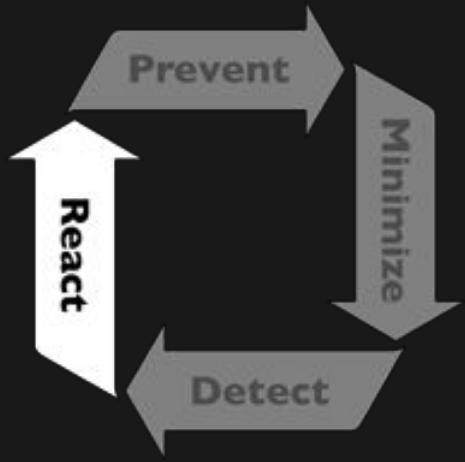
- Place access controls close to the resource
  - Smaller perimeter ⇒ easier to protect
- Default Linux applications have too much power
- Lock down user access for a "default" application
- Fully locked down applications limit innovation
- Relying on users making correct security decisions is tricky

Permissions



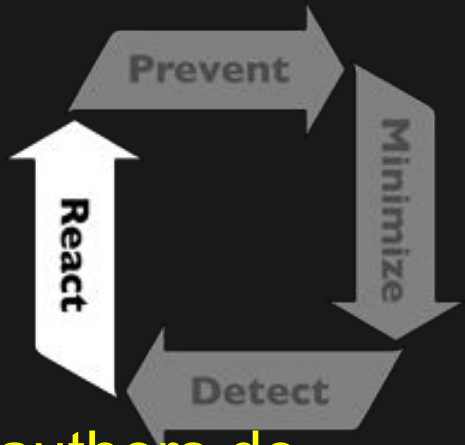
- Whitelist model
  1. Allow minimal access by default
  2. Allow for user accepted access to resources
- Ask users less questions
- Make questions more understandable
- 194 permissions
  - More ⇒ granularity
  - Less ⇒ understandability

React



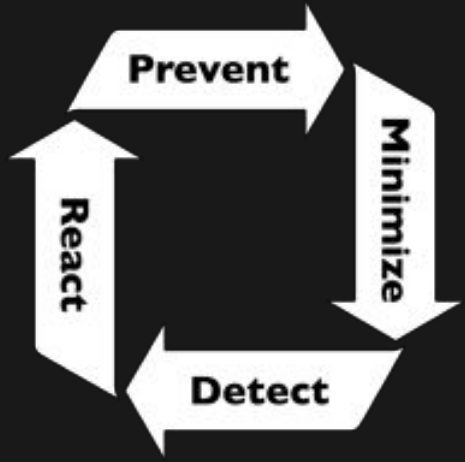
- Autoupdaters are the best security tool since Diffie-Hellman
- Every modern operating system should be responsible for:
  - Automatically updating itself
  - Providing a central update system for third-party applications
- Android's Over-The-Air update system (OTA)
  - User interaction is optional
  - No additional computer or cable is required
  - Very high update rate

Shared UID Regression



- Shared UID feature
  - Malware does not hurt computers, malware authors do
  - Two applications are signed ⇒ can share UIDs
  - More interactivity
- Panasonic reported that shared UID was broken
  - If the user installs malware, then the attacker could share UIDs with an existing installed app, like the browser
  - Breaks Application Sandbox

Security Philosophy



- Finite time and resources
- Humans have difficulty understanding risk
- Safer to assume that
  - Most developers do not understand security
  - Most users do not understand security

- Security philosophy cornerstones
  - Need to **prevent** security breaches from occurring
  - Need to **minimize** the impact of a security breach
  - Need to **detect** vulnerabilities and security breaches
  - Need to **react** to vulnerabilities and security breaches swiftly

Special Thanks

- SCRIBD
- Knowbe4
- BCX
- Nclose
- CDH
- UWC
- Samsung
- SA Government
- Bi Tech
- Trend Micro
- Novitas
- Moore
- WC CoLab
- Nectir
- VOX
- Future Innovation Lab