

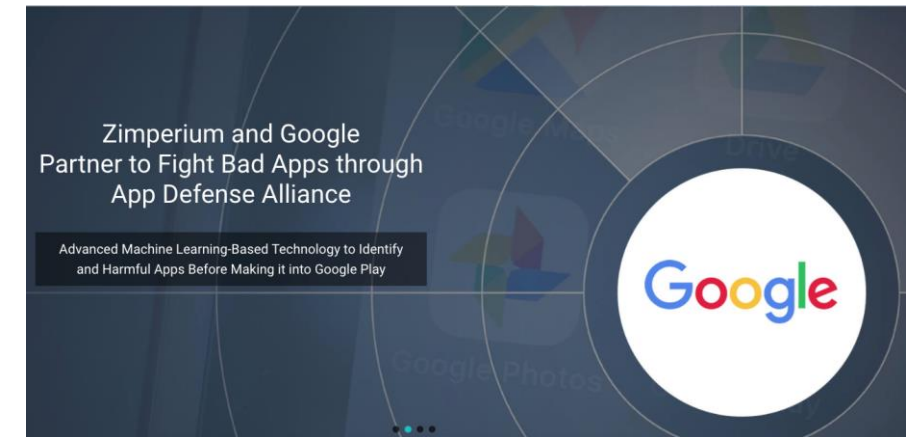


# Mobile App Protection



- **Founded:** 2010 (Private)
- **Headquarters:** Dallas, TX
- **Employees:** 160
- **Funded:** Warburg Pincus, SoftBank, Samsung, Sierra Ventures, Telstra
- **Market Positioning:**
  - Gartner: **Leader, Mobile Threat Defense (MTD)**
  - Frost & Sullivan: **Best Practices Award**
  - IDC: **Leader, Mobile Threat Management (MTM)**
  - Forrester: **Leader, Mobile Security**
  - Google: **App Defense Alliance**
- **Chosen by top global enterprises, over 80M+ Mobile Endpoints**

## Next Gen on-device AI Mobile Security



# Why Mobile App Protection?

*Your app running on mobile devices you have no control*



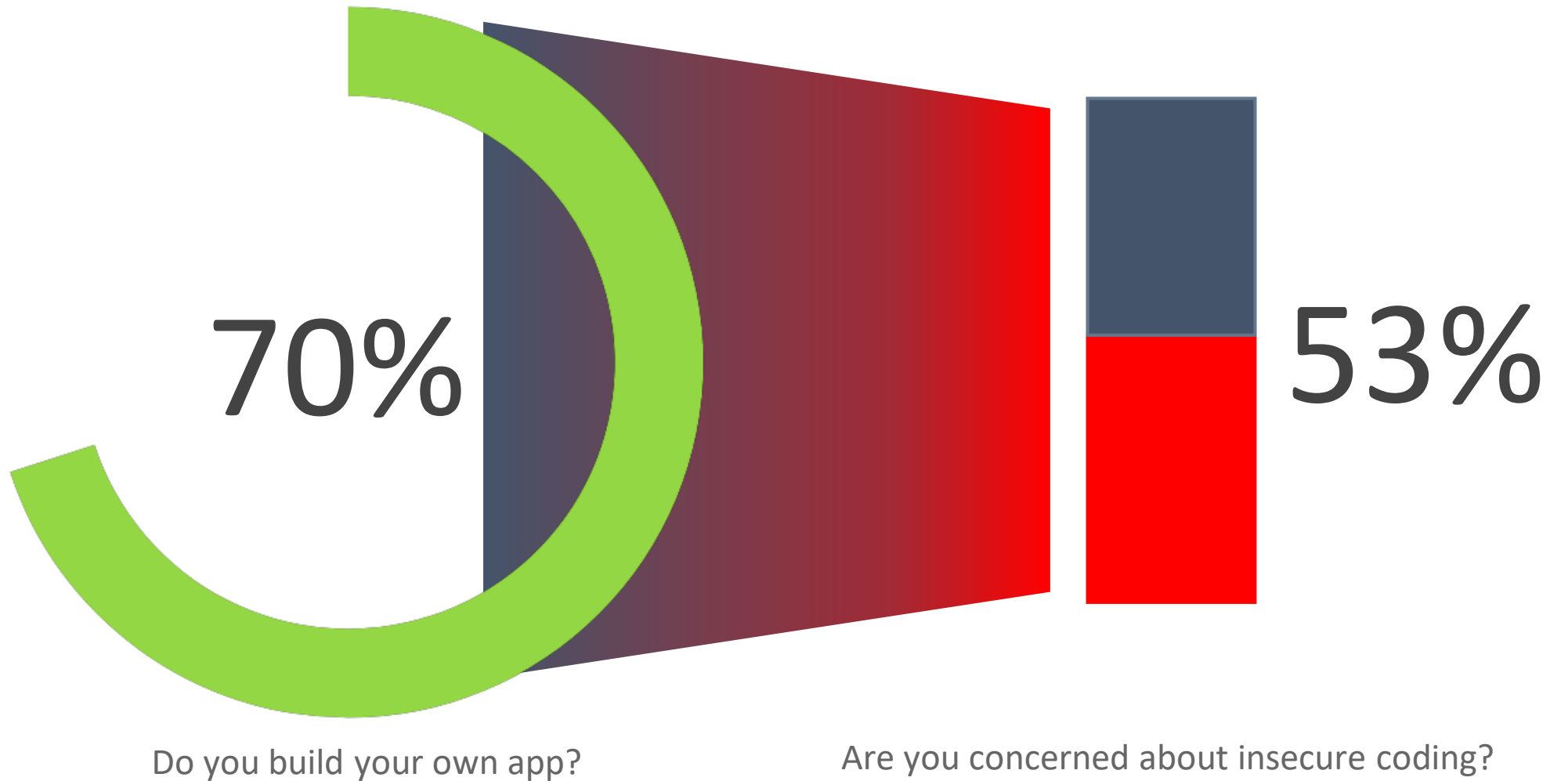
Your  
App



Your App running  
on compromised device



# Are the apps you build secure?



# How are Apps Targeted in the Wild?

## Piracy



## API key extraction



## Cloning & IP theft



## Financial fraud



## Malware insertion



## Credential harvesting

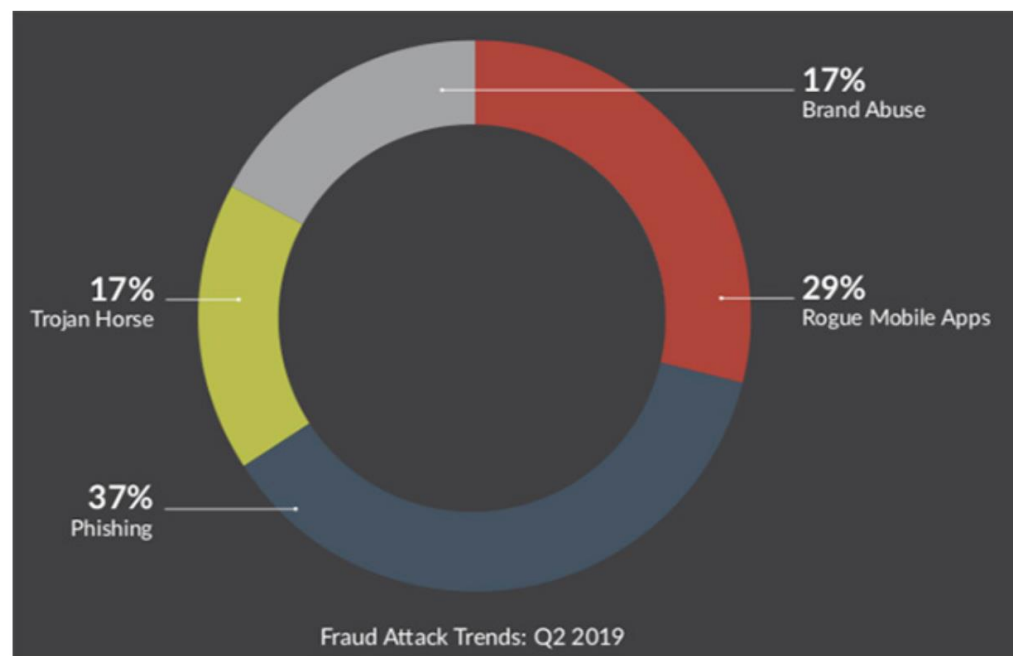




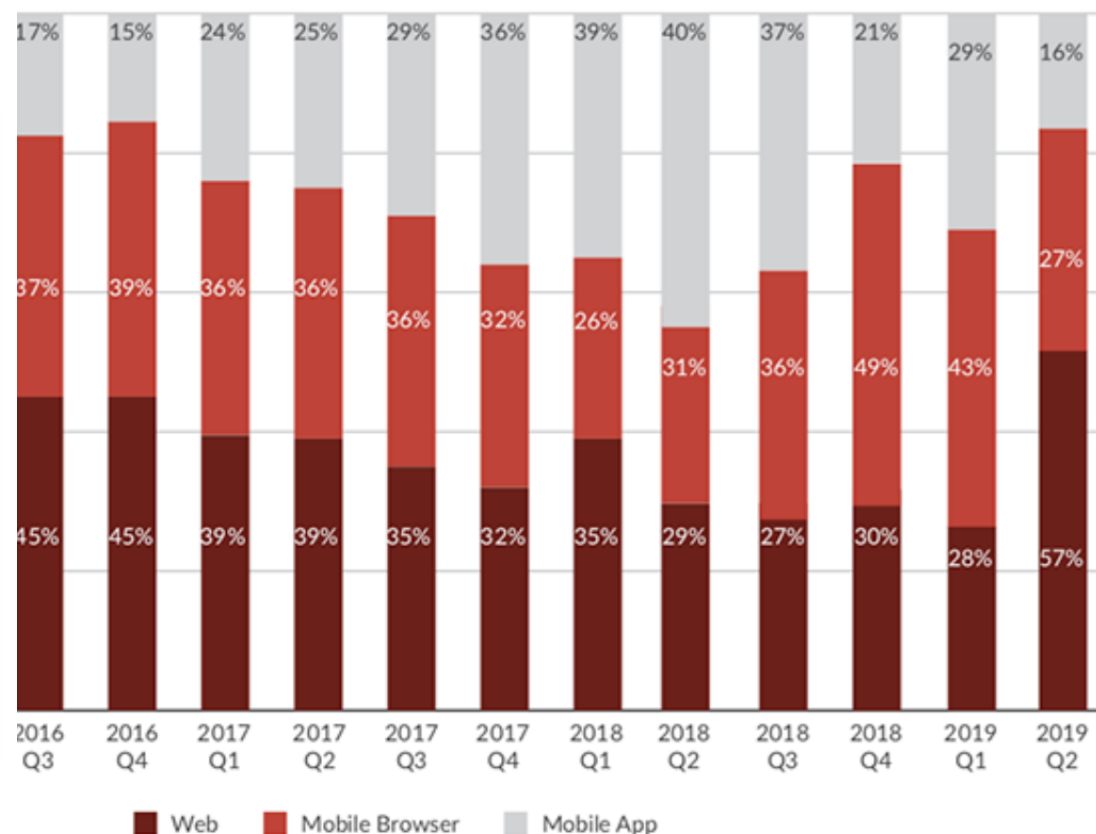
# Your App is a new target – more than 50% fraud

## Fake mobile app fraud tripled in first half of 2019

In Q2 2019, RSA Security identified 57,406 total fraud attacks worldwide. Of these, **phishing attacks** were the most prevalent (37%), followed by fake mobile apps (usually apps posing as those of popular brands).



## Fraud Transaction Distribution by Channel



# Hackers Steal 2.5M Overnight From Tesco Bank by Reverse Engineering Mobile App

**“It has been revealed weaknesses in the bank’s mobile applications left the door open for cybercriminals to brute force their way in and take more than £2.5 million of customers’ money...”**

**The Financial Conduct Authority (FCA) fined Tesco Bank £16.4m for “failing to exercise due skill, care and diligence in protecting” its current account holders.**

The image shows a large blue sign with the Tesco Bank logo. The word "TESCO" is in red, bold, sans-serif capital letters, and "Bank" is in blue, bold, sans-serif capital letters. Below "TESCO" are five blue diagonal stripes. The sign is mounted on a blue structure, and a building is visible in the background.

**TESCO Bank**



# Fake App - Repackage and Redistribute

## Fake Covid-19 apps fish in the troubled waters

*Some map-based applications that trace the path of the virus across the globe could end up infecting a user's phone with a virus, the digital kind that is. Spam documents that offer information about the virus through emails and message attachments are also increasing, cyber security firms said.*

By Priyanka Sangani, Anandi Chandrashekhar, ET Bureau | Last Updated: Mar 23, 2020, 07:39 AM IST



Save



*What such apps do instead is lock out the user and ask for ransoms to unlock their device.*

Mumbai | Pune: Web and mobile applications that track the spread of the **Covid-19** virus outbreak are also loading **ransomware** trojans and trackers to snoop on users, according to cyber security firms. For instance, some map-based applications that trace the path of the virus across the globe could end up infecting a user's phone with a virus, the digital kind that is. Spam documents that offer information about the virus through emails and message attachments are also increasing, the firms said.

COVID-19 CASES

Confirmed

Deaths





# Popular Consumer App Hacked

Private-hire drivers caught hacking Grab, Gojek apps to bypass system and raise earnings



1 of 2 Modified versions of ride-hailing apps such as Grab and Gojek are being hawked online and through messaging apps. PHOTO: BOON TAT TAN/FACEBOOK

## Attacker can

- Reverse Engineer your app - (IP theft, API usage)
- Modify / Repackage
- Inject fake transactions (fraud)
- Attack the backend via app

# Mobile App Attack

- **Static Attack (offline)**



*Hackers download and transform the code into human readable format,*

- Code Logic / IP
- API keys / endpoints
- Remove/Inject malicious code
- Repackage the app

- **Dynamic Attack (online)**

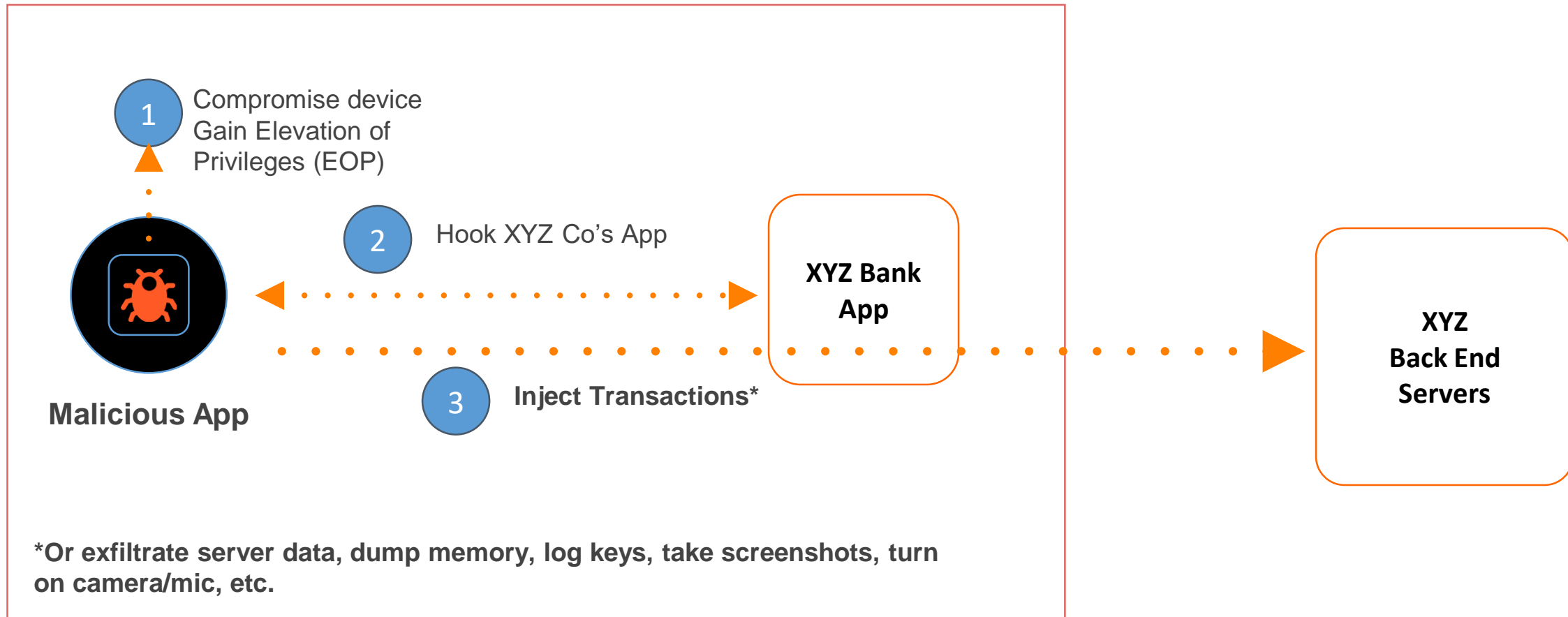


*Hacker “owns” the device gathers app’s behavior,*

- Root / Jailbreak / Compromise
- Network/MITM
- Privilege escalation
- Hooking / Debugging
- Code injection



# Online - Anatomy of An Attack – XYZ Bank



# Regulations





# RMIT - Mobile App and Devices

## Appendix 4 Control Measures on Mobile Application and Devices

1. A financial institution should ensure digital payment, banking and insurance services involving sensitive customer and counterparty information offered via mobile devices are adequately secured. This includes the following:
  - (a) ensure mobile applications run only on the supported version of operating systems and enforce the application to only operate on a secure version of operating systems which have not been **compromised**, jailbroken or rooted i.e. the security patches are up-to-date;
  - (b) design the mobile application to operate in a secure and tamper-proof environment within the mobile devices. The mobile application shall be prohibited from storing customer and counterparty information used for authentication with the application server such as PIN and passwords. Authentication and verification of unique key and PIN shall be centralised at the host;
  - (c) undertake proper due diligence processes to ensure the application distribution platforms used to distribute the mobile application are reputable;
  - (d) ensure proper controls are in place to access, maintain and upload the mobile application on application distribution platforms;
  - (e) activation of the mobile application must be subject to authentication by the financial institution;
  - (f) ensure secure provisioning process of mobile application in the customer's device is in place by binding the mobile application to the customer's profile such as device ID and account number; and
  - (g) monitor the application distribution platforms to identify and address the distribution of fake applications in a timely manner.

# MAS - Technology Risk Mgmt Guidelines -

Mar 2019

## Mobile Applications

C.1 Security measures that should be considered for securing mobile applications are as follow:

- (a) avoid storing or caching data in the mobile application to mitigate compromise of the data on the device;
- (b) implement anti-hooking or anti-tampering mechanisms to prevent injection of malicious code that could alter or monitor the behaviour of the application at runtime;
- (c) implement appropriate application integrity check (e.g. using checksum and digital signature) to verify the authenticity and integrity of the application and code obfuscation techniques to prevent reverse engineering of the mobile application;
- (d) implement certificate or public key pinning to protect against MITMA;
- (e) implement a secure in-app keypad to mitigate against malware that captures keystrokes; and
- (f) implement device binding to protect the software token from being cloned.

## Mobile Applications Scanning

A.2 Common testing methods for security vulnerabilities in software applications include:

### (a) Static Application Security Testing

Static Application Security Testing (SAST) involves a set of tools or technologies designed to scan and analyse static source codes, byte codes and binaries for coding and design flaws indicative of security vulnerabilities. The tester will have full internal knowledge of the system including architecture and design specifications, source codes or configuration files to guide the testing.

### (b) Dynamic Application Security Testing

Dynamic Application Security Testing (DAST) involves a set of tools or technologies designed to detect conditions indicative of exploitable vulnerabilities in a system in its run-time state. The tester has no prior knowledge of the system when the test is performed.

### (c) Interactive Application Security Testing

Interactive Application Security Testing (IAST) involves a combination of SAST and DAST techniques to analyse application codes, run-time controls libraries, requests and responses, as well as data and control flows and identify vulnerabilities in a system.

What security feedback do you get today?





**BUILD COMPLIANT**

What issues should be fixed before releasing our app?

**BUILD SECURE**

How can we harden our app against reverse engineering or code tampering?

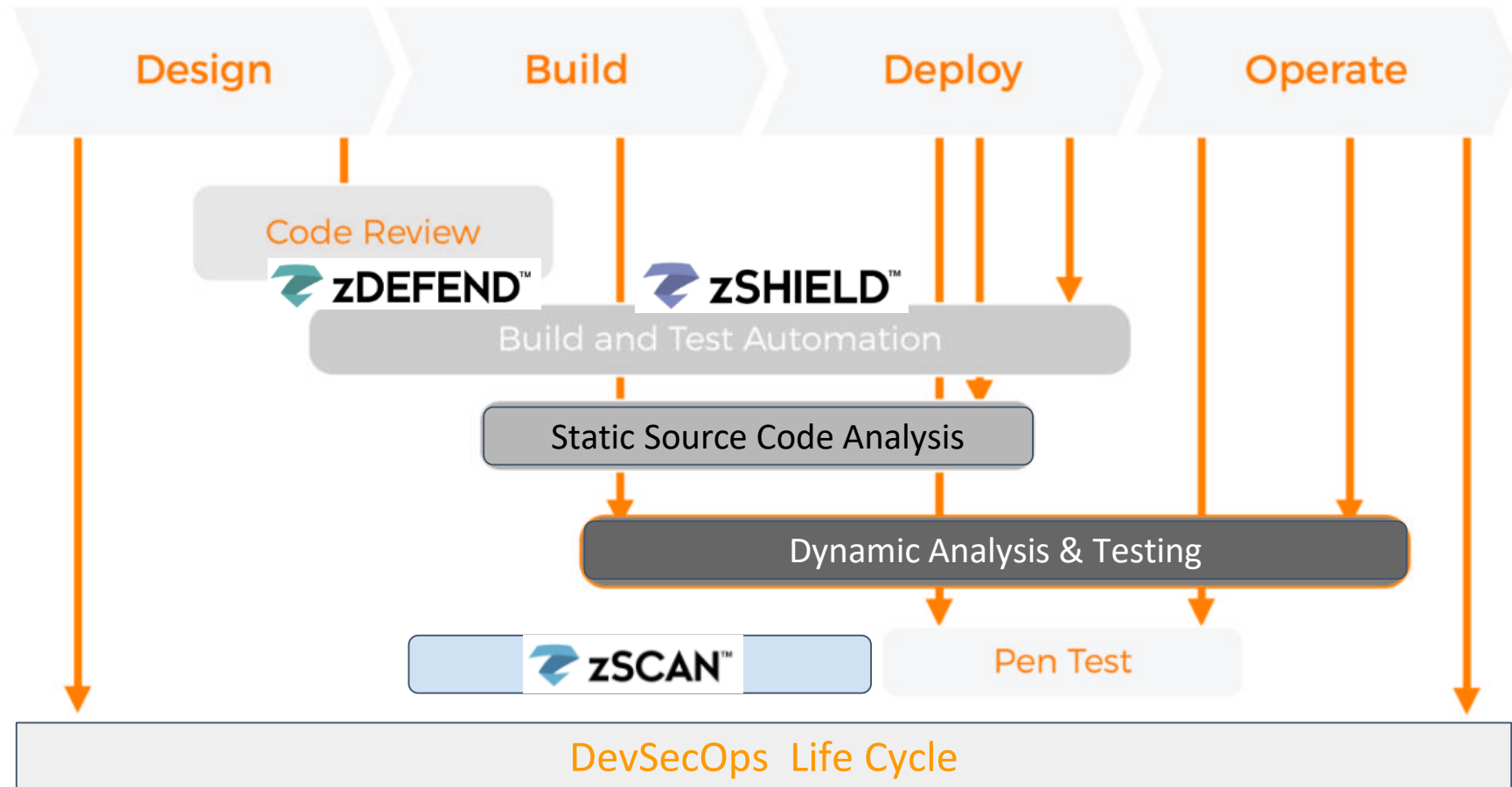
**RUN SECURE**

How can we protect our app from advanced attacks on end user devices?

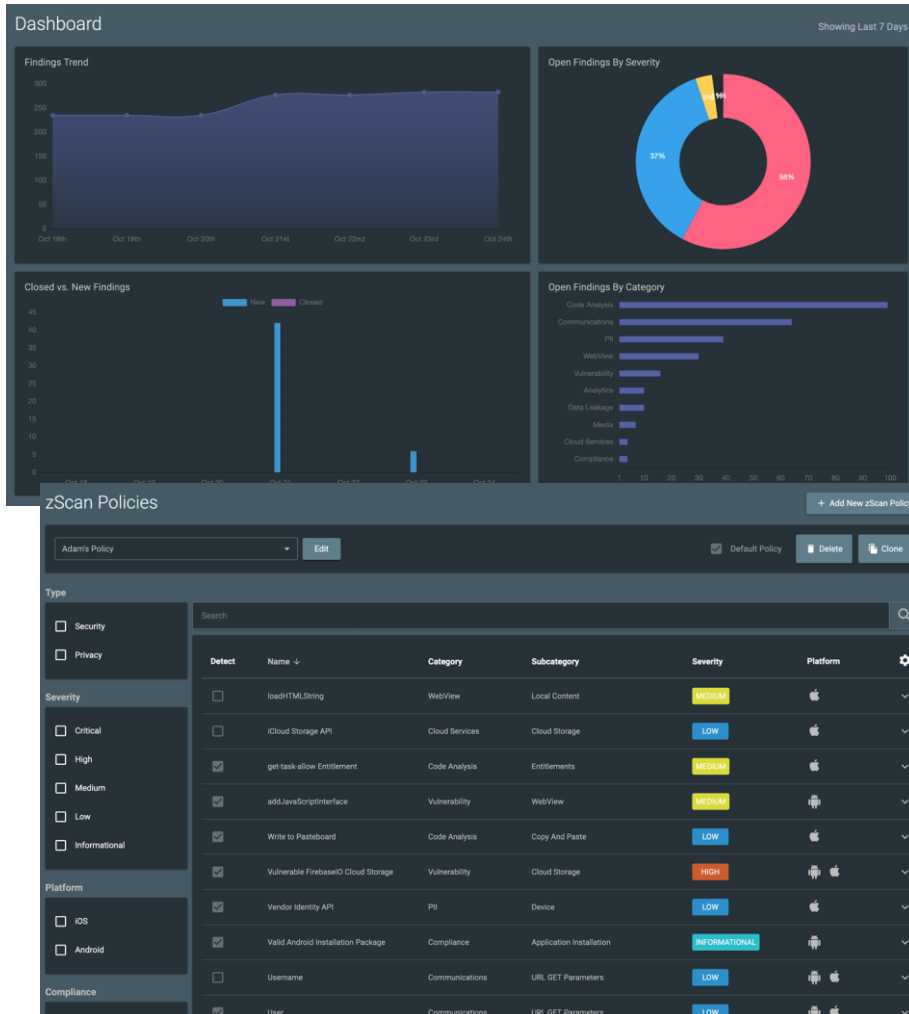




## Software Development Life Cycle



# zScan - Continuous App Security Assessment

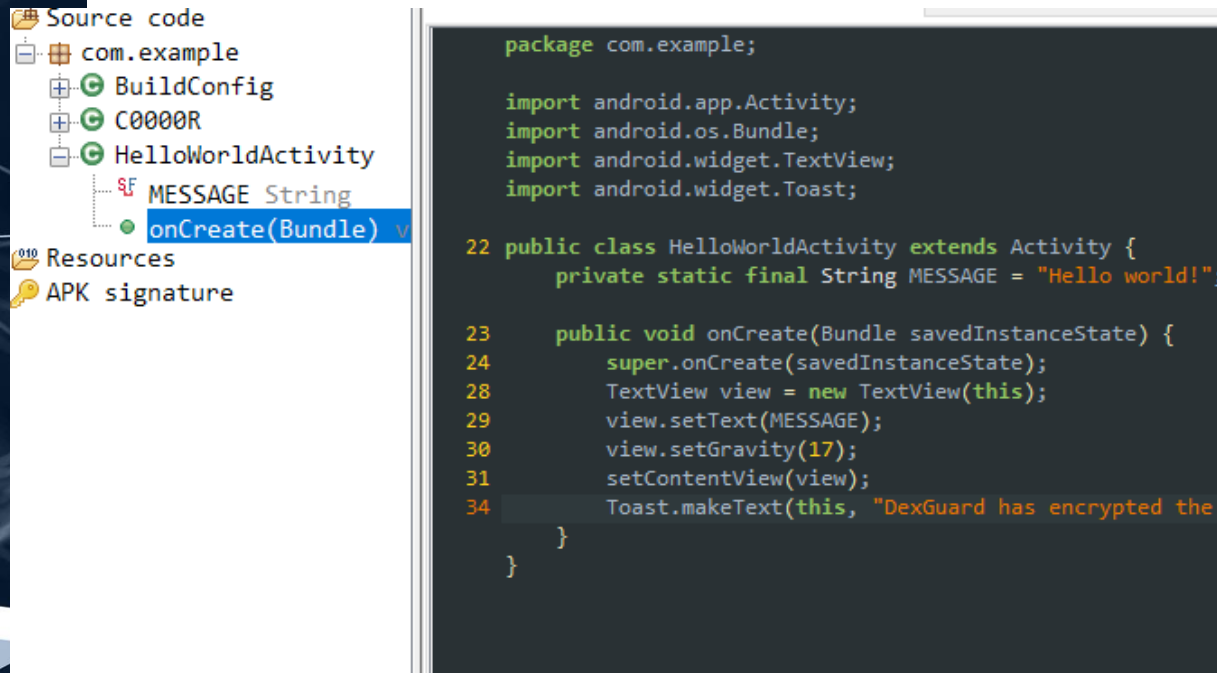


- **Continuous and Automated** assessment of mobile app during development lifecycle
- **Early** identification of security and privacy risks
- **Developer/Security** role persona
- Customizable assessment and policy such as **NIAP, OWASP...etc**
- **Compare** app risk builds
- **Integrated** into **CI/CD pipelines** and ticketing system (JIRA) to assign security findings

# zSHIELD - Code Obfuscation

## Before Protection

All the strings and code structure can see clearly in human readable form.



```
package com.example;

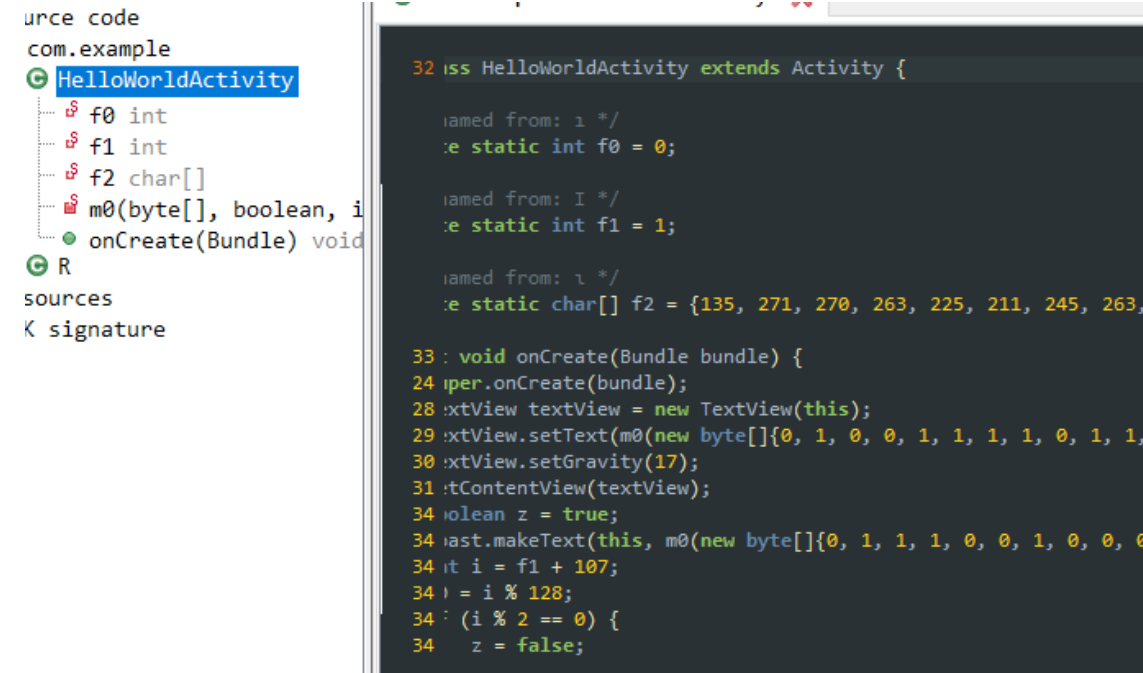
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;

22 public class HelloWorldActivity extends Activity {
    private static final String MESSAGE = "Hello world!";

23     public void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
28         TextView view = new TextView(this);
29         view.setText(MESSAGE);
30         view.setGravity(17);
31         setContentView(view);
34         Toast.makeText(this, "DexGuard has encrypted the"
    }
}
```

## After Protection

The code and string are not un-readable and difficult to understand



```
32 class HelloWorldActivity extends Activity {

    named from: 1 */
    static int f0 = 0;

    named from: 1 */
    static int f1 = 1;

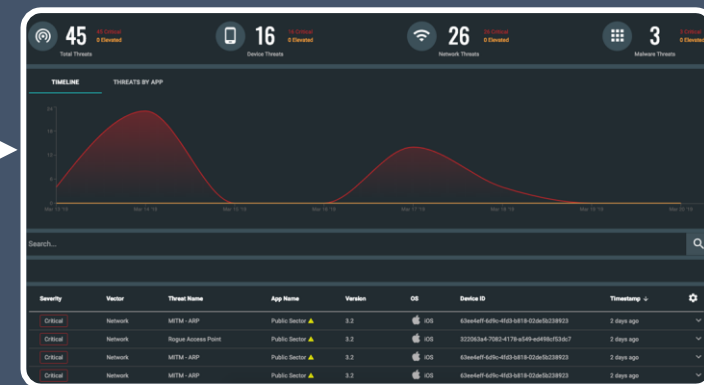
    named from: 1 */
    static char[] f2 = {135, 271, 270, 263, 225, 211, 245, 263,

33 : void onCreate(Bundle bundle) {
34 :per.onCreate(bundle);
28 :xtView textView = new TextView(this);
29 :xtView.setText(m0(new byte[]{0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1,
30 :xtView.setGravity(17);
31 :tContentView(textView);
34 :olean z = true;
34 :ast.makeText(this, m0(new byte[]{0, 1, 1, 1, 0, 0, 1, 0, 0, 0
34 :t i = f1 + 107;
34 : i = i % 128;
34 : (i % 2 == 0) {
34 : z = false;
}
```

# Reporting on All Application Threats



SDK + Post Build



## Implementation

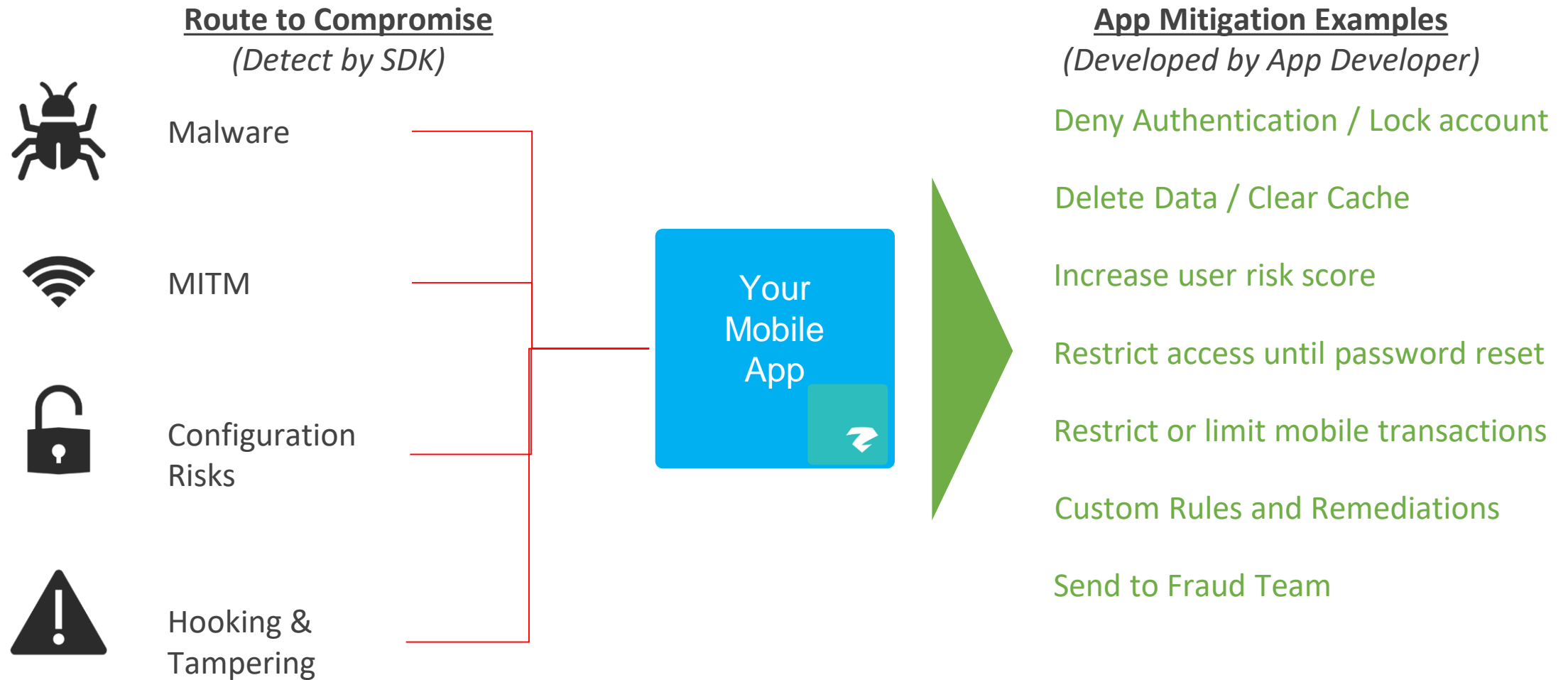
zShield enables your app to detect and report app tampering events into Zimperium's administration and reporting dashboard, zConsole. zDefend enables your app to report and remediate cyber threats to your app's code and its operating environment. This is the most comprehensive threat detection available to reduce the risk of third-party mobile devices causing fraudulent transactions.

## Threat Reporting

- App Tampering
- Malware
- Zero Day Attacks
- WiFi and Network threats
- Device Risks
  - Compromised
  - Vulnerable OS
  - Configuration Risk
  - Rooted/Jailbroken



# In-App threat detection and remediation



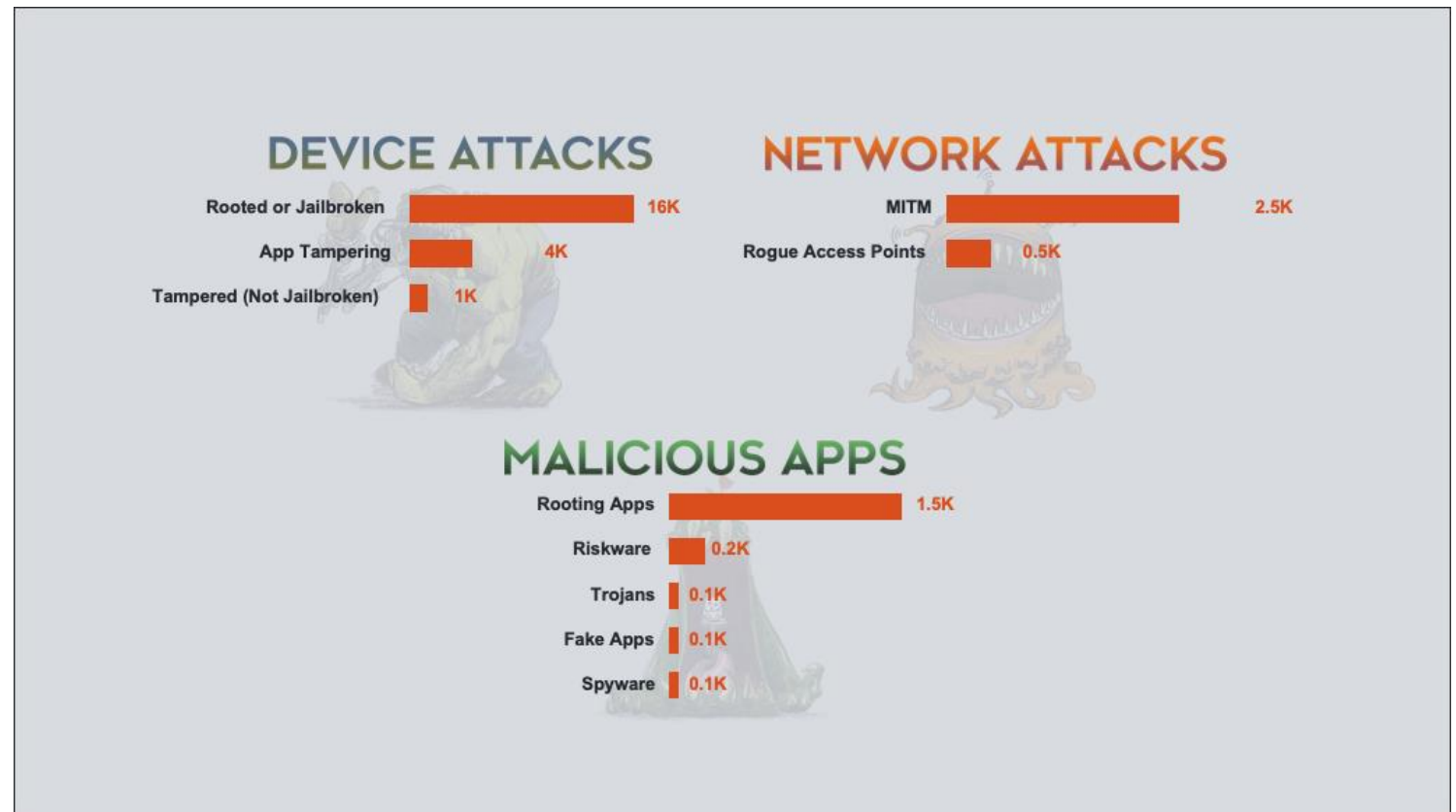
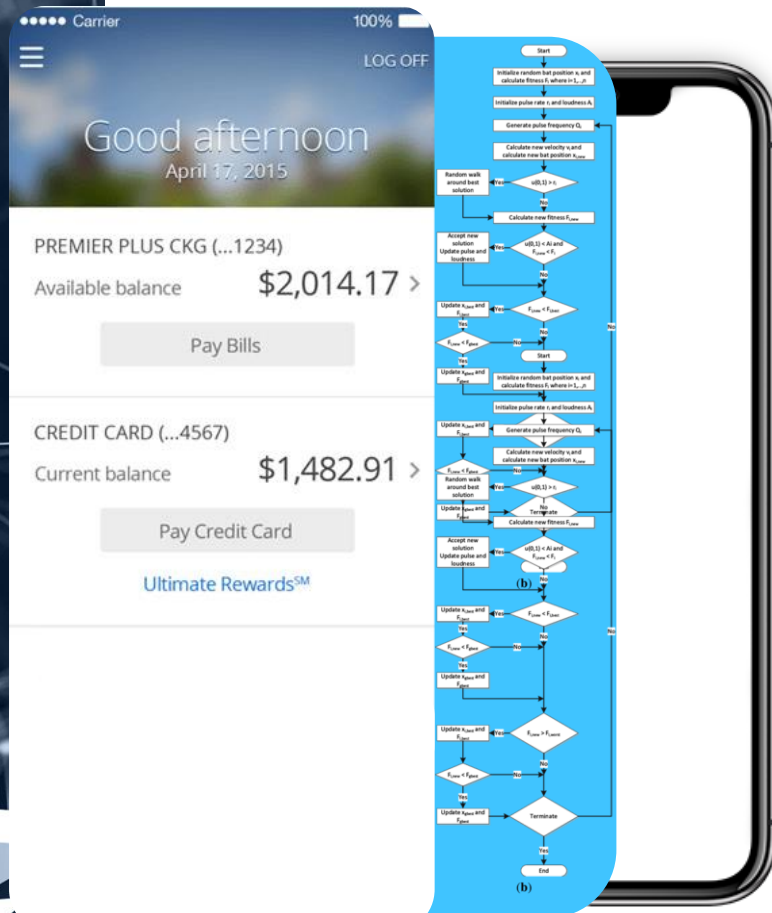
# Case Study



# Large US Bank - 60M users

## Deployed Zimperium to Reduce Over \$1B in Fraud

In the first **30 days**, they saw over **900,000 threats** with many connections to bad Wi-Fi or apps installed from 3rd party app stores.



# P2P Payment - Large Asian Banking Consortium

40+ Member Banks



Each bank in the consortium is responsible for developing their security features in the mobile app, not only its time consuming, but most of them lack the expertise and security are compromised with time to market pressure



In-App Protection (zIAP), was easy to implement to allow developer to focus on business function rather than security. Provide great visibility of the **device risk baseline** where application is running.



Launched an app security SDK (As Service) including **2FA, Biometrics and zIAP for real time threat detection**

Deployed across 4 member banks. Payment are processed but fraud alert are raised to trigger early investigation.


***Implement 2FA and Mobile Threat Defence - they were able to get approval for \$30,000 HKD transaction limit from \$10,000 HKD per day.***

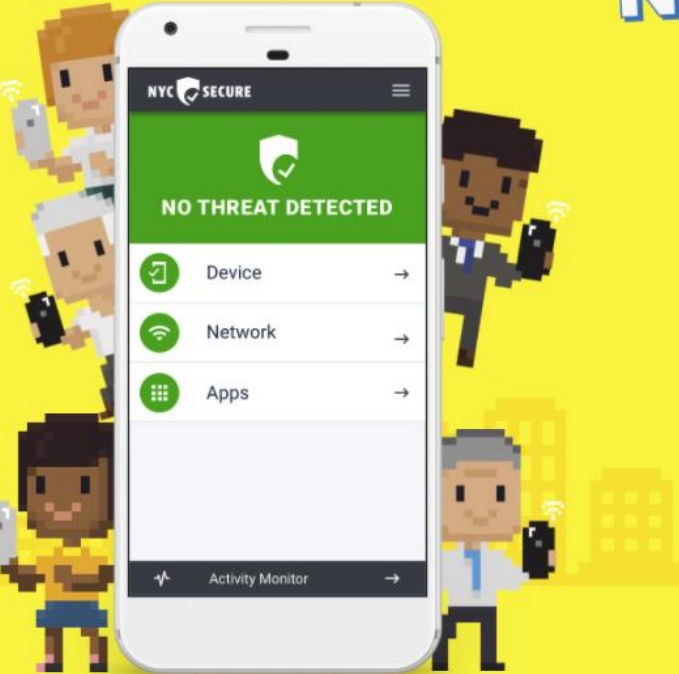




# NYC Secure






Translate ▼



## Now NYC can help protect your phone from cyber threats.

**Get the free NYC Secure app**

-  Alerts you to unsecure Wi-Fi networks, unsafe apps in Android, system tampering & more
-  Helps you protect your phone and your privacy
-  \$0 to download, \$0 to use, no in-app purchases, no ads

We intend to partner with a private company called Zimperium that has developed the most advanced mobile threat prevention app at the same time is incredibly protective of user privacy.

Geoff Brown  
CISO, New York City