



WT BEARING

LT



13 cm

# MOBILE BANKING APPS ARE NOT CREATED EQUAL



# Agenda



Scott King

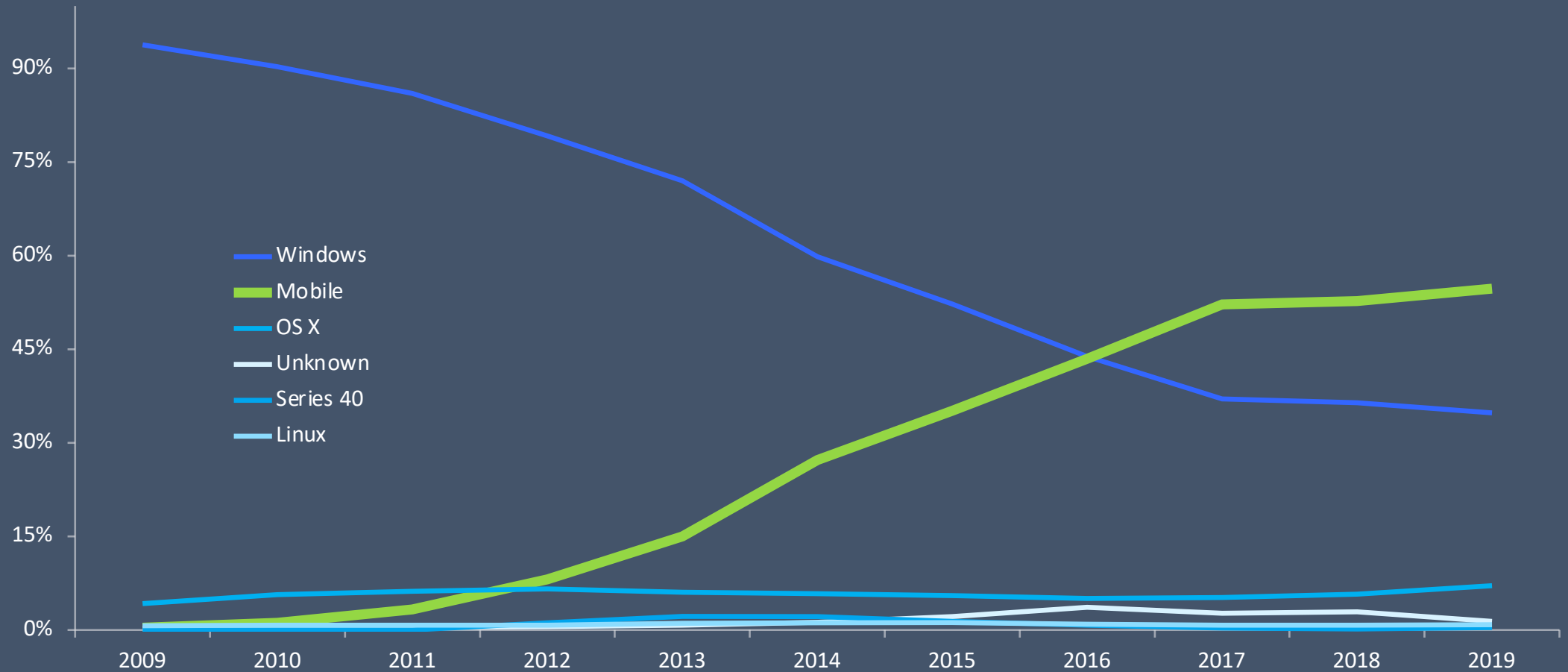
Director Embedded Security

Scott has over 20 years experience providing customized software solutions to enterprise customers in mobile, supply chain and DevOps. Scott invests his time researching mobile app security and worldwide mobile threat events.

1. State of mobile banking
2. OWASP Mobile Top 10 results
3. How mobile banking apps compare for privacy and security risks
4. Common security issues
5. Common privacy and data leakage issues
6. What you can do



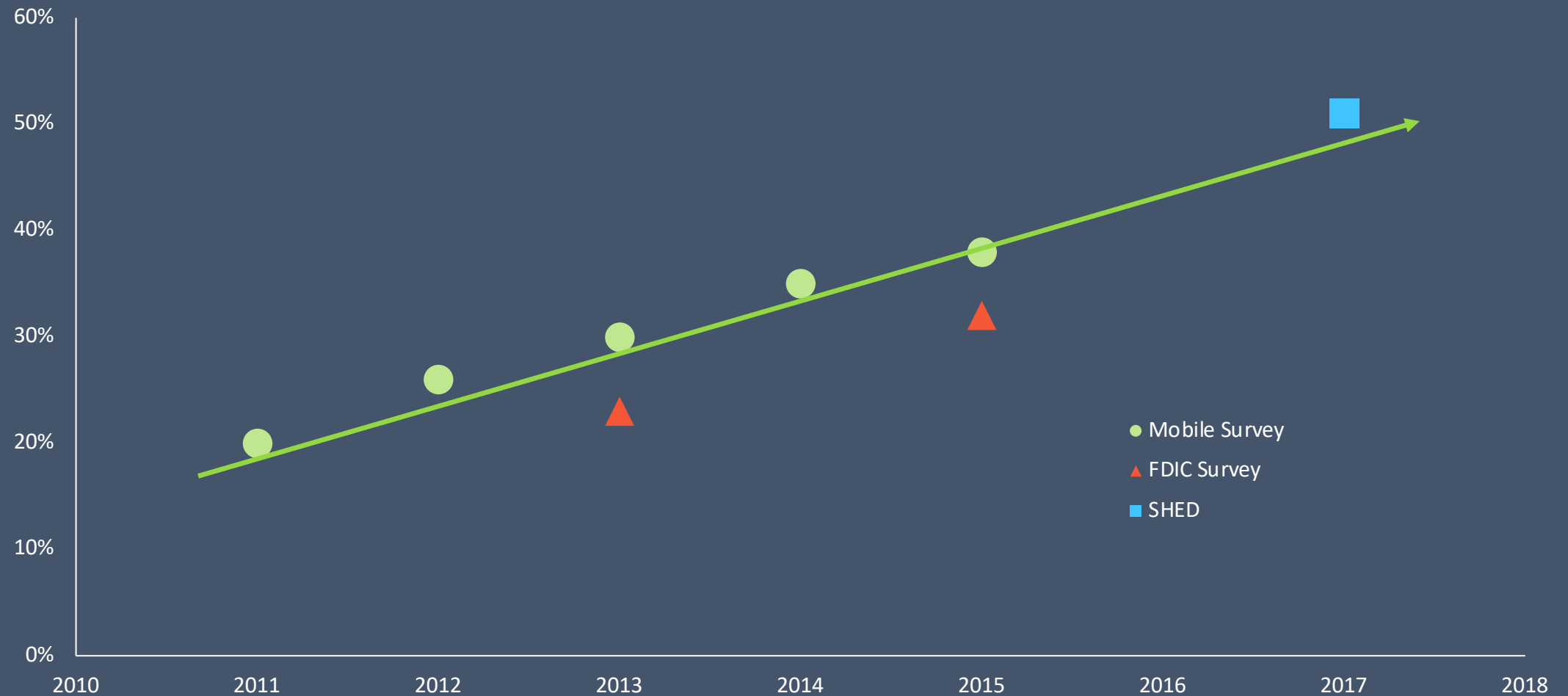
# Operating System Market Share



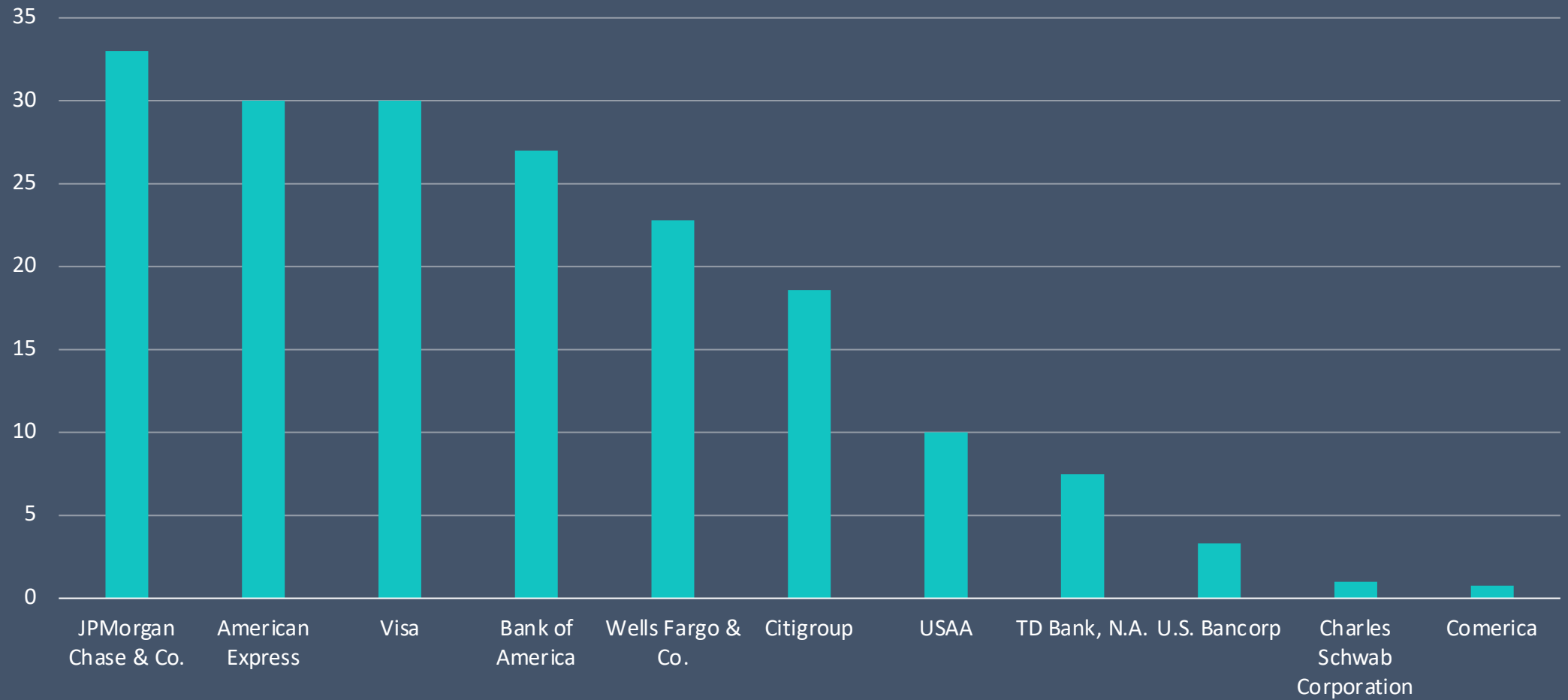


# Mobile Banking

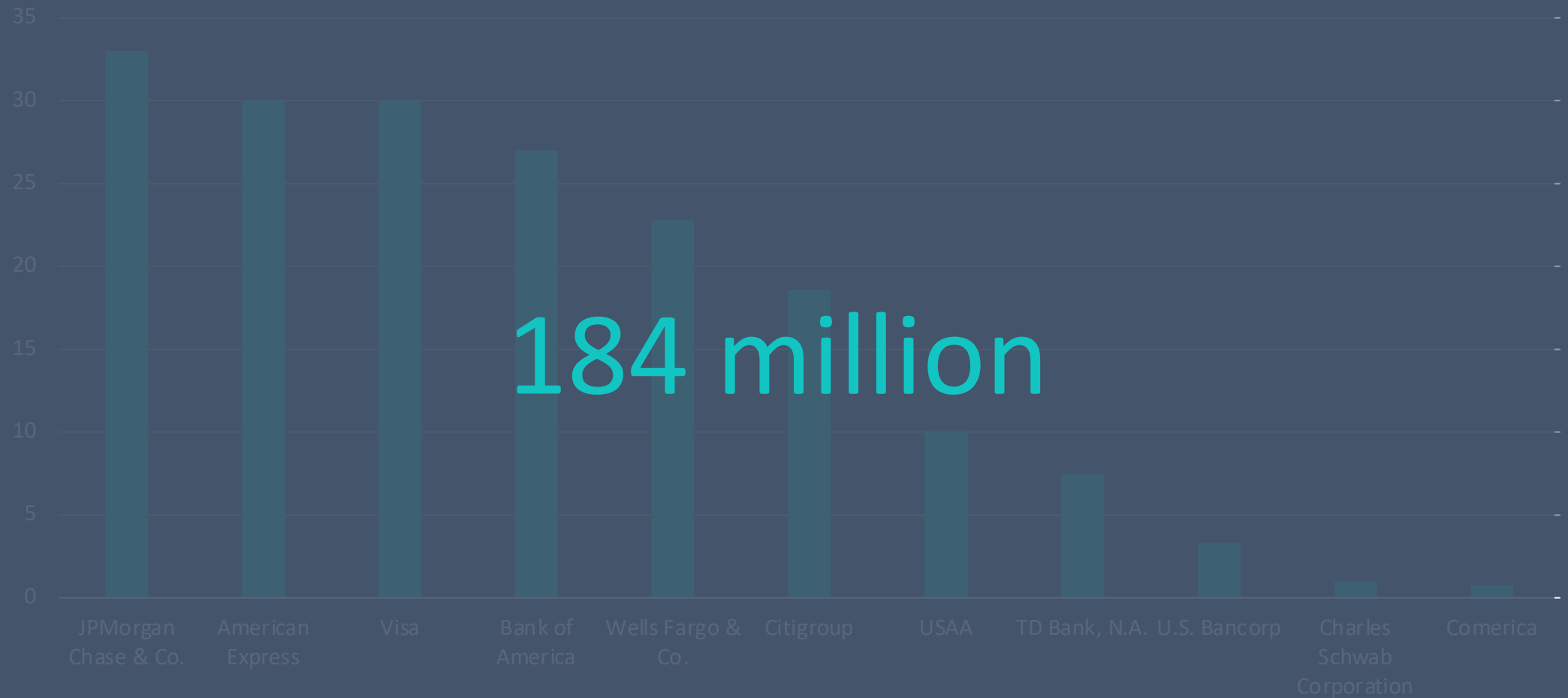
% of US account holders accessing via mobile



# Mobile banking users (millions)



# Mobile banking users (millions)





Mobile banking may help to address some challenges consumers face...

However, a well-designed and secure mobile platform, as well as consumer access to and facility with mobile technology and the Internet, are likely needed in order for mobile banking to be a reliable banking channel.





Mobile banking may help to address some challenges consumers face...

However, a well-designed and secure mobile platform, as well as consumer access to and facility with mobile technology and the Internet, are likely needed in order for mobile banking to be a reliable banking channel.



<http://ir.usbank.com/sec-filings/sec-filing/10-k/0001193125190047201> Mobile Banking Apps Are Not Created Equal - All Rights Reserved © 2019




“...attacks may include computer viruses, malicious or destructive code, denial-of-service attacks, ransomware or ransom demands to not expose security vulnerabilities in the Company’s systems or the systems of third parties. These risks may increase in the future as the Company continues to increase its mobile and internet-based product offerings.... In addition, the Company’s customers often use their own devices, such as computers, smart phones and tablet computers, to make payments and manage their accounts. The Company has limited ability to assure the safety and security of its customers’ transactions with the Company to the extent they are using their own devices, which could be subject to similar threats.





Mobile malware will constitute  
30% of all malware in 2019.



A portrait of Jamie Dimon, Chairman and Chief Executive Officer of JP Morgan Chase. He is a middle-aged man with grey hair, wearing a dark suit jacket over a light blue shirt. He is smiling slightly and looking towards the camera. The background is a blurred office setting with other people and computer monitors.

**“The threat of  
cyber security may  
very well be the  
biggest threat to  
the U.S. financial  
system.”**

Jamie Dimon

Chairman and Chief Executive Officer

JP Morgan Chase



TESCO Bank

TESCO Bank

Sources:

<https://www.theguardian.com/business/2016/nov/08/tesco-bank-cyber-thieves-25m>

<https://www.fca.org.uk/news/press-releases/fca-fines-tesco-bank-failures-2016-cyber-attack>



Mobile Banking Apps Are Not Created Equal - All Rights Reserved © 2019

# Hackers Steal £2.5M Overnight from Tesco 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) later fined Tesco Bank £16.4m for “failing to exercise due skill, care and diligence in protecting” its current account holders.

Sources:

<https://www.theguardian.com/business/2016/nov/08/tesco-bank-cyber-thieves-25m>

<https://www.fca.org.uk/news/press-releases/fca-fines-tesco-bank-failures-2016-cyber-attack>

Mobile Banking Apps Are Not Created Equal - All Rights Reserved © 2019





# Advanced Application Analysis



## Technical Summary



40/100



**MED**

Privacy Risk

77/100



**HIGH**

Security Risk

This Technical Summary contains a mid-level summary and score information for an app's identified risk conditions. This digest is intended for a technical audience and provides a listing of items we identified. Findings are separated into analysis areas, followed by categories and additional support details when available. Each finding is represented by a Red, Orange, Yellow or Green colored square.

- Red indicates a high level of risk and used to indicate when a test has failed.
- Orange indicates a moderate level of risk
- Yellow indicates a low risk or informational finding
- Green indicates that no risk conditions were identified and used to indicate when a test has passed.

### Index

[Privacy Summary](#)

[Security Summary](#)



Dashboard

In-Apps

Threats

Devices

Users

Policies

Manage

Downloads

Docs

Policy

Terms

Version 5.1.108

# Dashboard

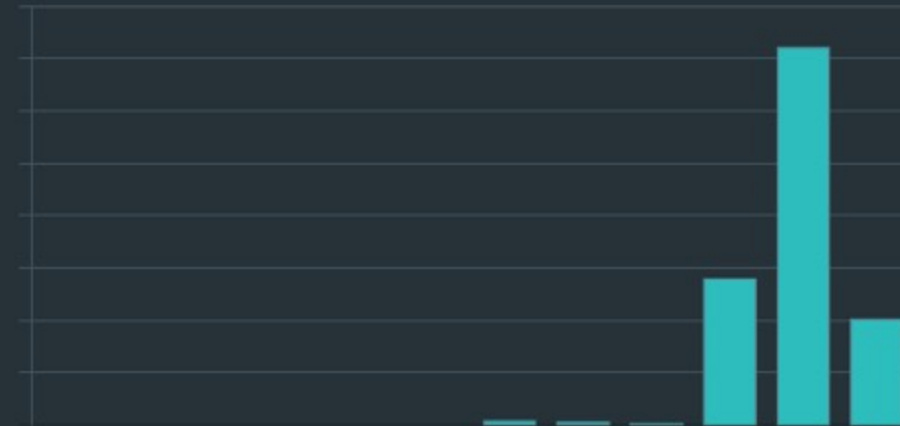
Showing Last 30 Days

## Threats

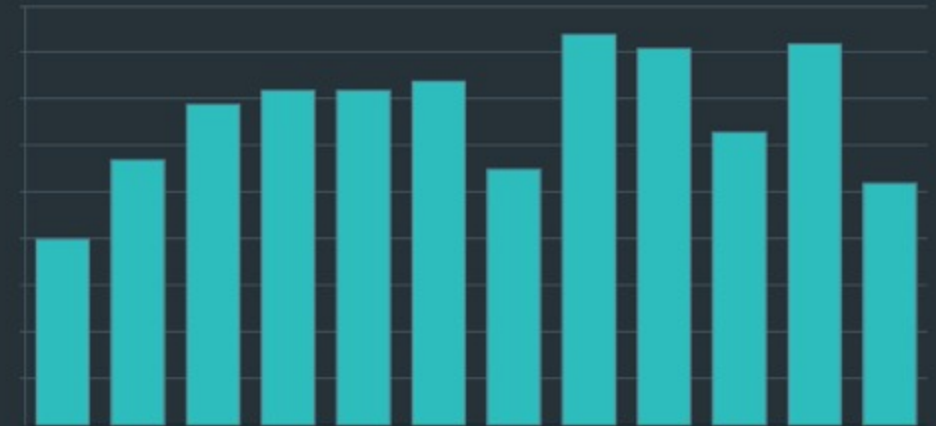
Timeline



## Monthly Active Devices (MAD)



## Daily Active Devices (DAD)



- Dashboard
- In-Apps
- Threats
- Devices
- Users
- Policies
- Manage

Devices

All

Last 30 Days

2K

Total Devices

2K

iOS

0

Android

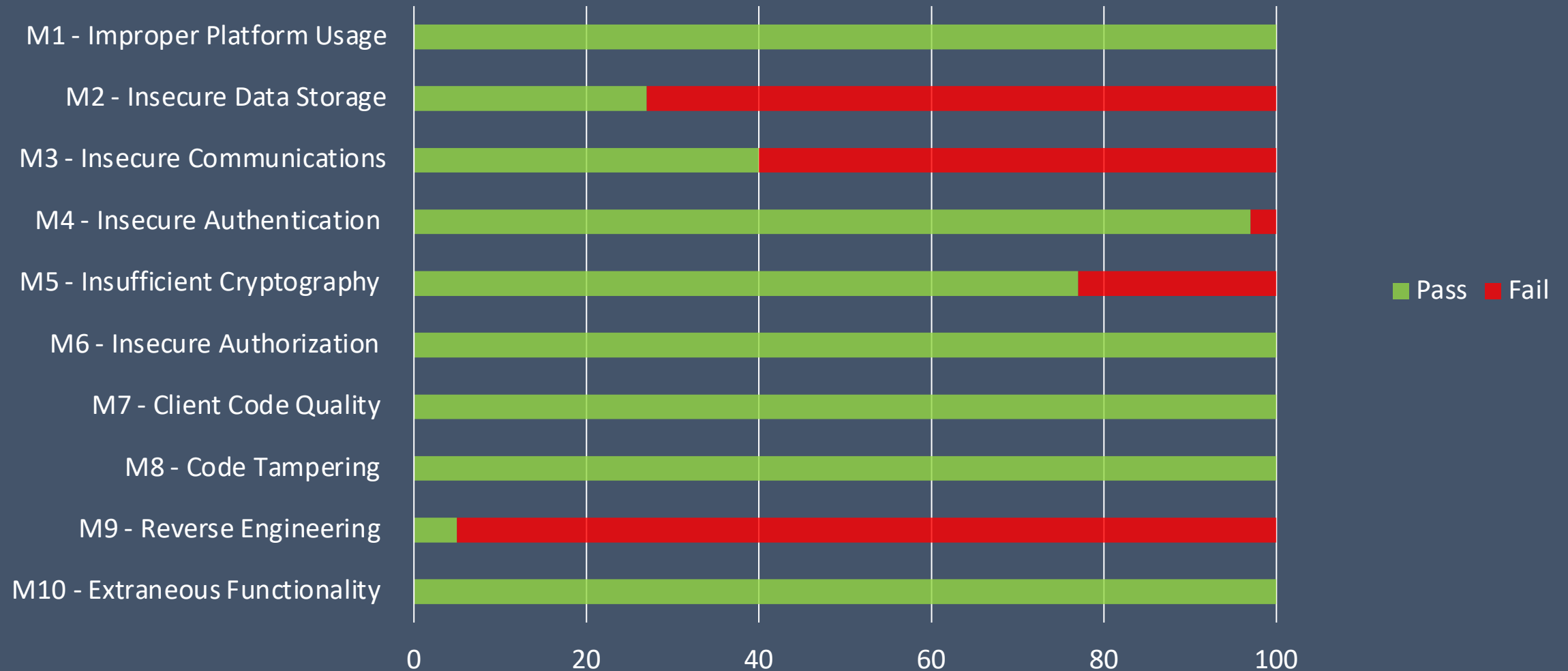


Search

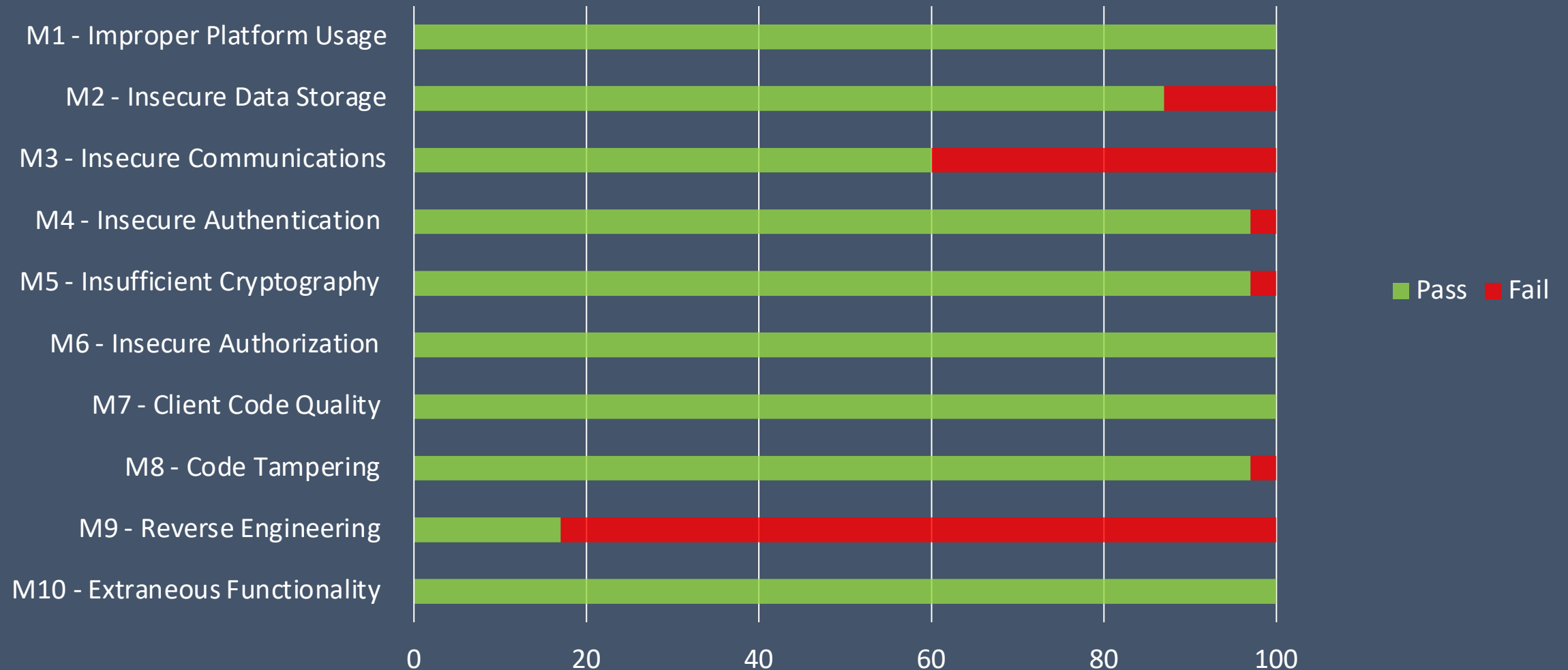
Applied Filters

Model	OS	OS Version	App Name	Version	Build	Bundle Id	Group Name	Created	Last Seen		
iPhone	<div></div> iOS	12.3.1					Default Group	38 minutes ago	38 minutes ago		
iPhone	<div></div> iOS	12.3.1					Default Group	an hour ago	an hour ago		

# OWASP Mobile Top 10 - Banking

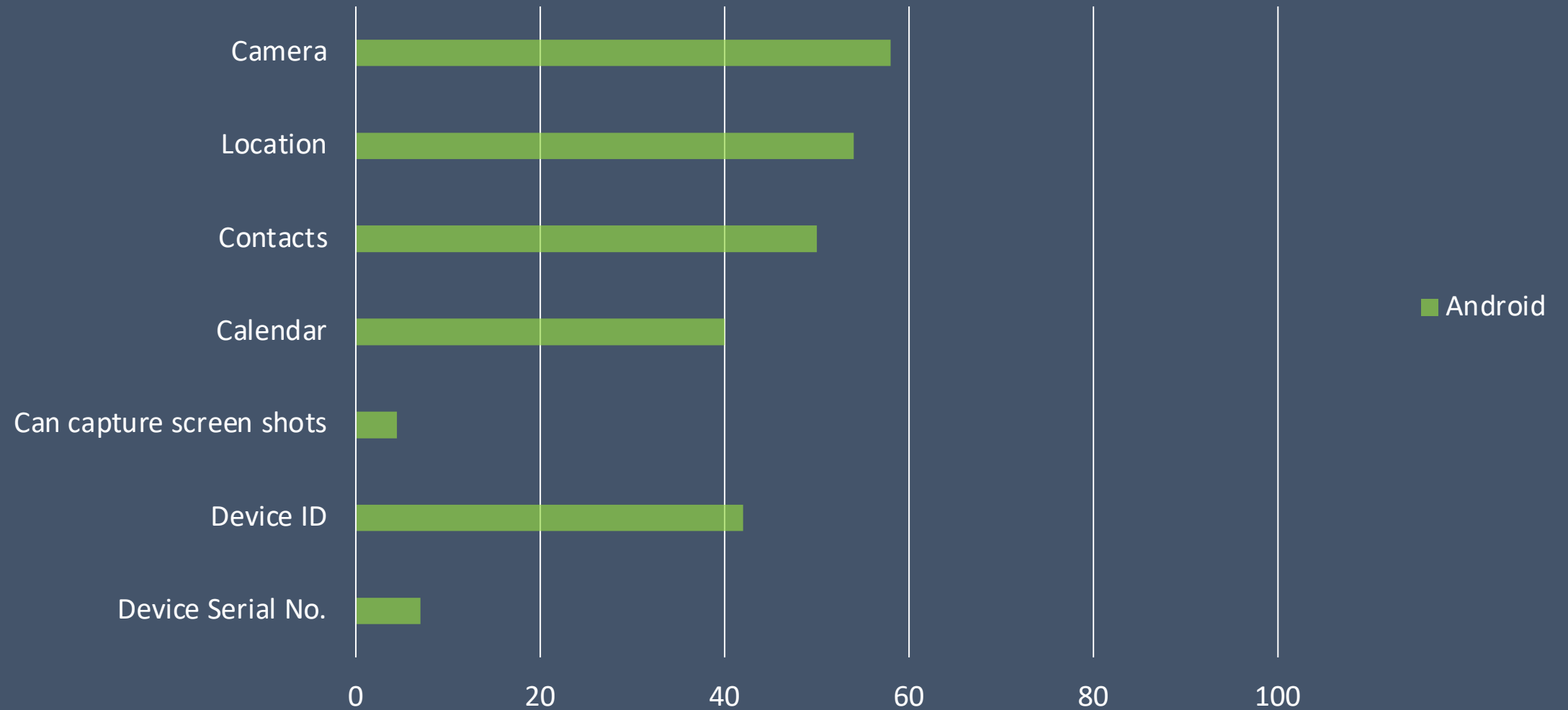


# OWASP Mobile Top 10 - Travel

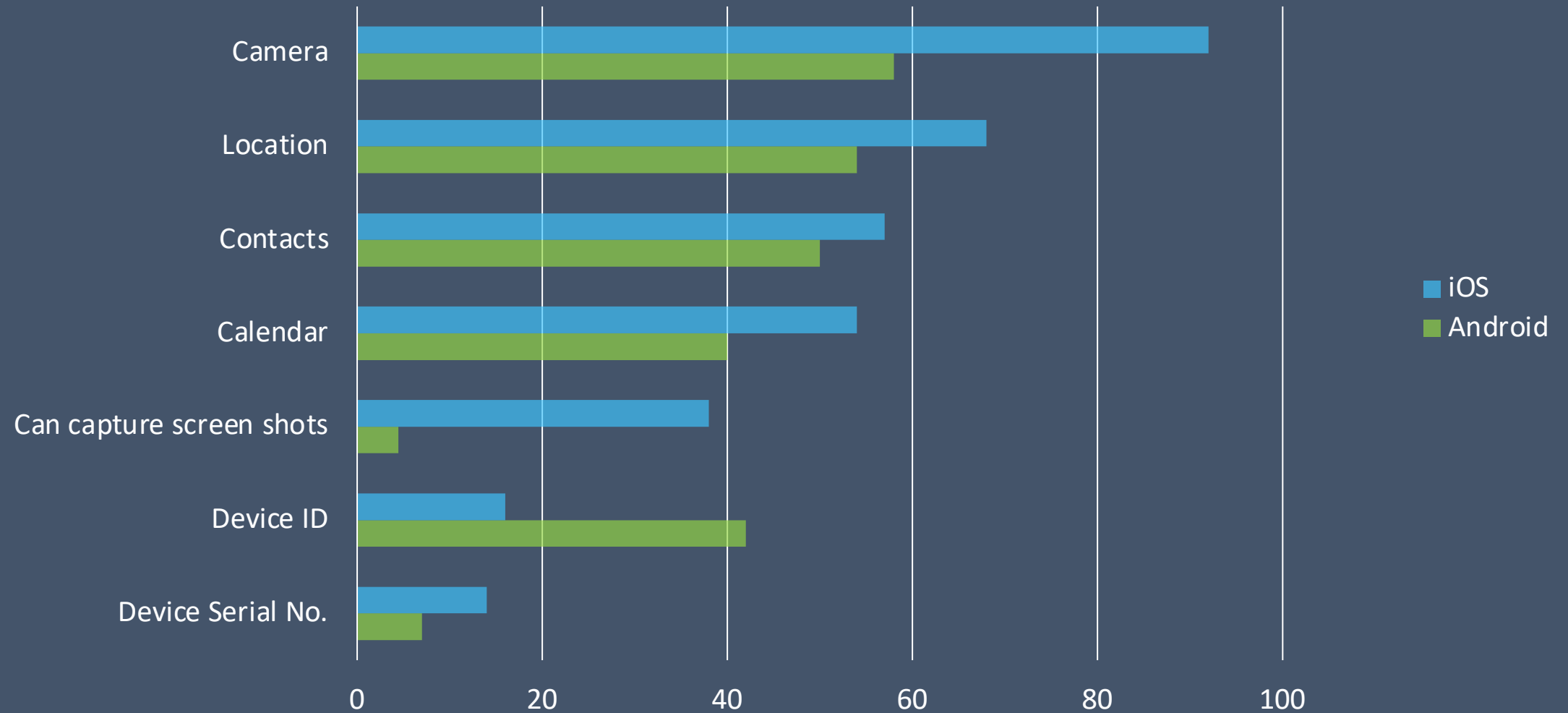




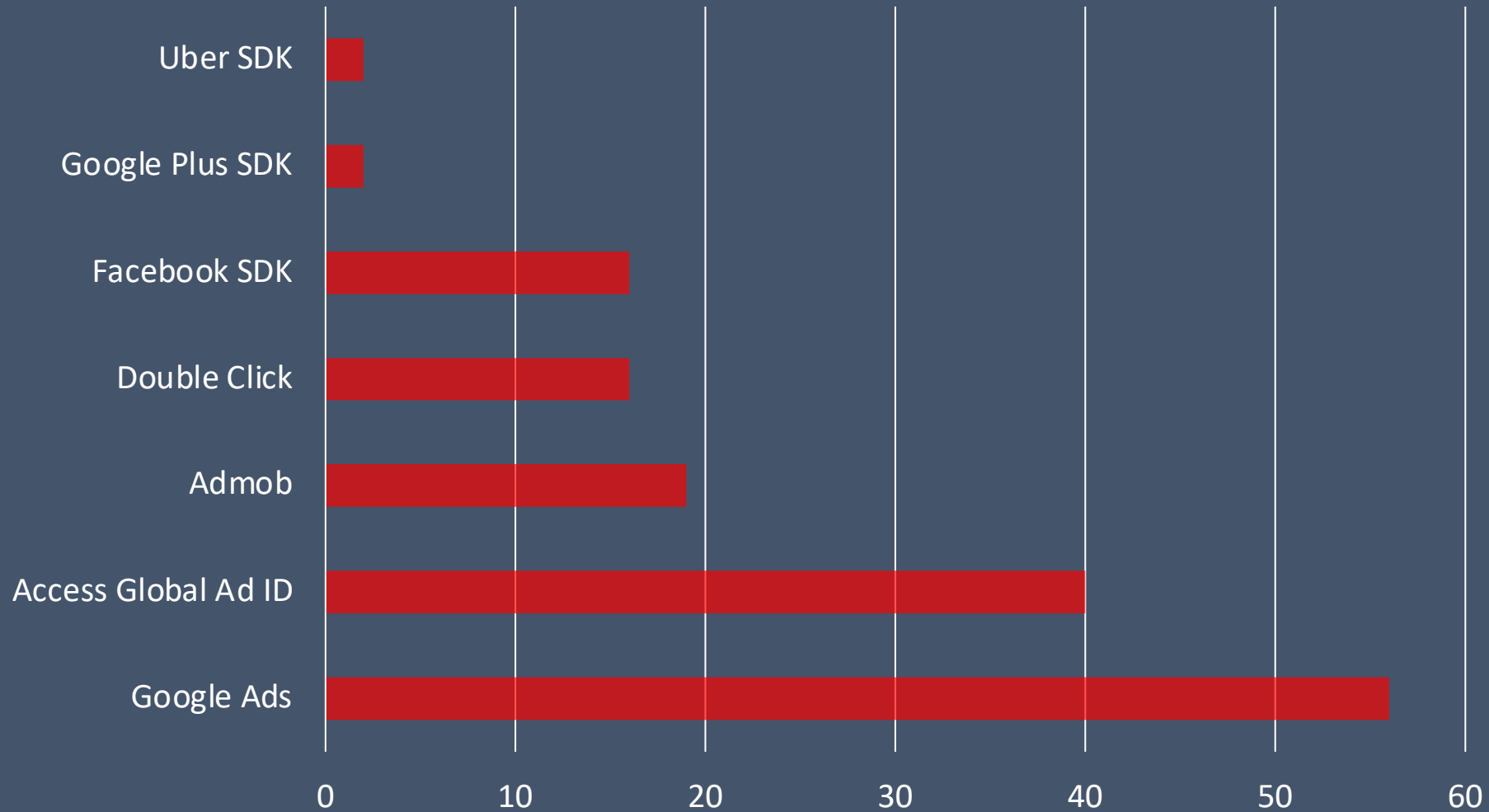
# Permissions



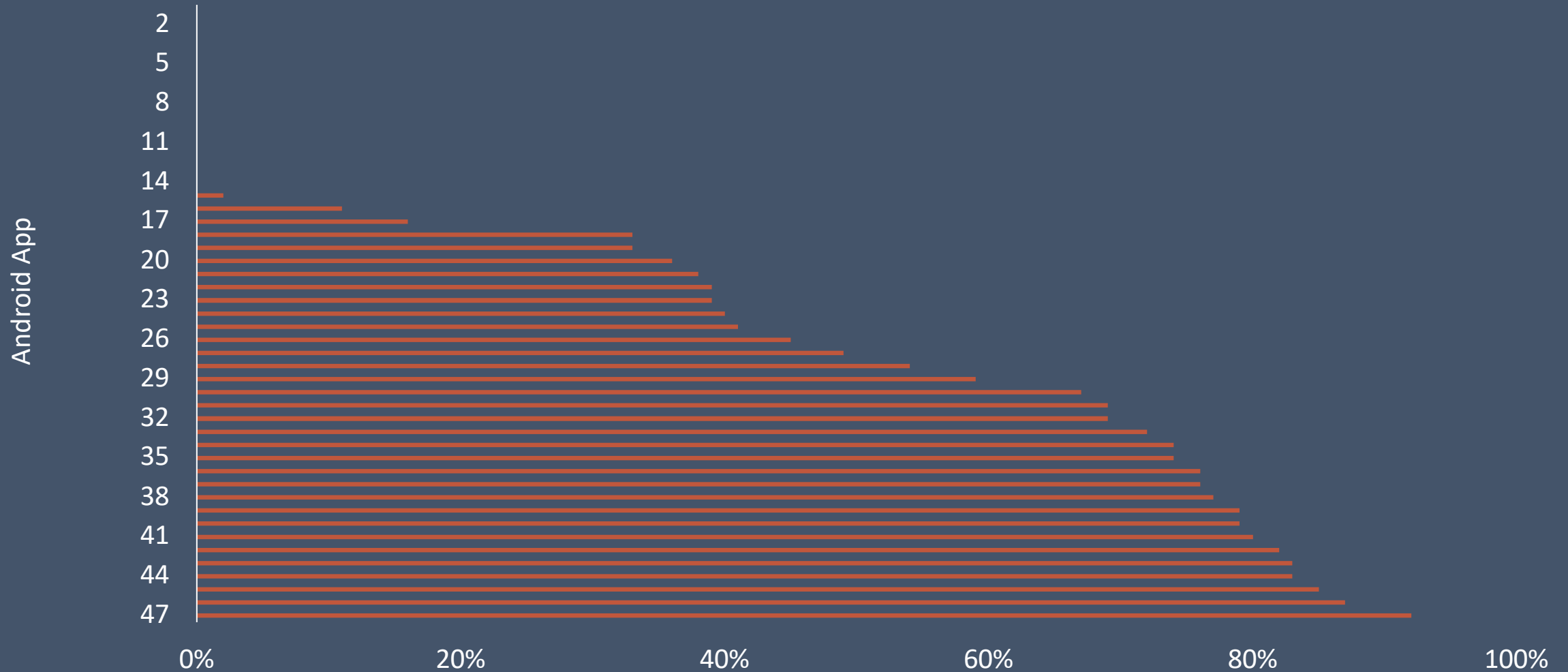
# Permissions



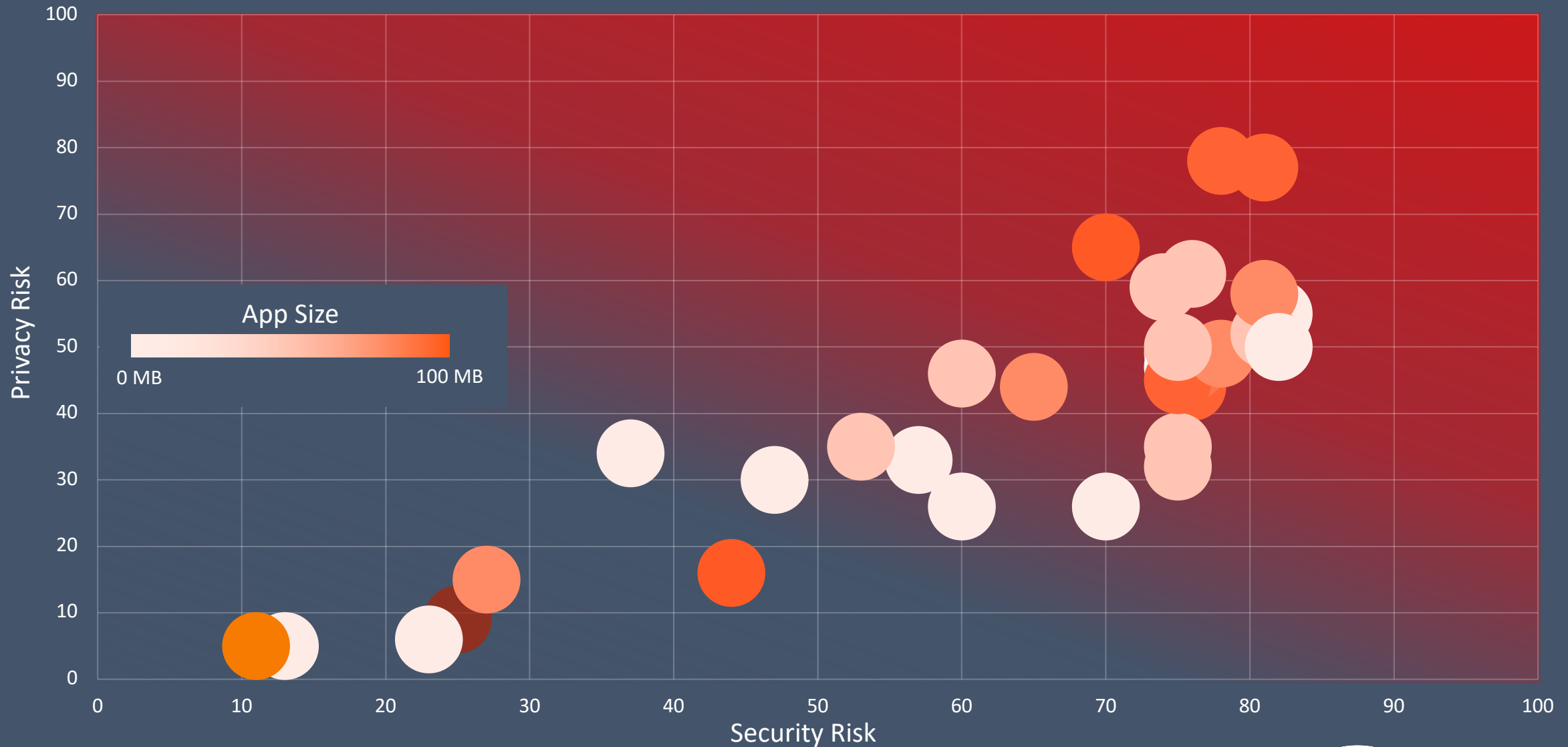
# Advertising



# Android Shared Code



# Privacy and Security Risk Scores - Android



# Bank - US 36x

The iOS app implements a low-level IOKit device enumeration API calls. Implementing a low-level IOKit device enumeration API call could indicate a malicious intent to obtain data normally not available. The developer should consider using an approved Apple method to avoid this risk condition. The Android app uses the 'proceed()' method during SSL error handling. This can allow the connection to be established even if the SSL Certificate is invalid. This is not a recommended practice. The app can send SMS messages and can capture the receipt of an SMS message. This app is actively targeted by the BankBot malware campaign.

## OWASP Mobile Top 10



Improper Platform Usage	●	●
Insecure Data Storage	●	●
Insecure Communications	●	●
Insecure Authentication	●	●
Insufficient Cryptography	●	●
Insecure Authorization	●	●
Client Code Quality	●	●
Code Tampering	●	●
Reverse Engineering	●	●
Extraneous Functionality	●	●



## Security

## Privacy



37

69



81

77

# Bank - US 36x

## OWASP Mobile Top 10

The Android app uses the 'proceed()' method during SSL error handling. This can allow the connection to be established even if the SSL Certificate is invalid. This is not a recommended practice. The app can send SMS messages and can capture the receipt of an SMS message. This app is actively targeted by the BankBot malware campaign.

Improper Platform Usage  
Insecure Data Storage  
Insecure Communication  
Insecure Authentication  
Insufficient Cryptography  
Insecure Authorization  
Client Code Quality  
Code Tampering  
Reverse Engineering  
Extraneous Functionality



## Security

## Privacy

81

77



# Bank - US 56p

The iOS app implements a low-level IOKit device enumeration API call. Because enumeration is currently prohibited by Apple it could indicate trojan or malware activity. The developer should consider using an approved Apple method to avoid this risk condition. The app may be vulnerable to local or remote SQL injection attack. The Android app stores inline API keys and values. The app is not implementing Certificate Pinning. Both apps fail the Static Data Exposure and Source Code Reverse Engineering Exposure test and expose source level metadata symbols as outlined by OWASP Mobile Top 10.

## OWASP Mobile Top 10



Improper Platform Usage	●	●
Insecure Data Storage	●	●
Insecure Communications	●	●
Insecure Authentication	●	●
Insufficient Cryptography	●	●
Insecure Authorization	●	●
Client Code Quality	●	●
Code Tampering	●	●
Reverse Engineering	●	●
Extraneous Functionality	●	●



## Security

## Privacy



46

60



25

9

# Bank - US 56p

## OWASP Mobile Top 10



## Security

## Privacy

The Android app stores inline API keys and values. The app is not implementing Certificate Pinning. Both apps fail the Static Data Exposure and Source Code Reverse Engineering Exposure test and expose source level metadata symbols as outlined by OWASP Mobile Top 10.

- Insufficient Cryptography
- Insecure Authorization
- Client Code Quality
- Code Tampering
- Reverse Engineering
- Extraneous Functionality



25

9

# Bank - US 49p

Bearer' related oauth tokens were found in the iOS App. An adversary could potentially gain access to these tokens. This app does not protect its jailbreak detection functionality from manipulation and fails the OWASP Top 10 Binary Protection testing. The Android app uses unsecure storage data mode (WORLD\_READABLE, WORLD\_WRITABLE). The app utilizes the Java.io.File delete() method to perform a file deletion. This method is not a secure file deletion method. This application uses synthetic methods to access private class entries, which are normally not accessible. This is a suspicious and unusual coding practice.

## OWASP Mobile Top 10



Improper Platform Usage	●	●
Insecure Data Storage	●	●
Insecure Communications	●	●
Insecure Authentication	●	●
Insufficient Cryptography	●	●
Insecure Authorization	●	●
Client Code Quality	●	●
Code Tampering	●	●
Reverse Engineering	●	●
Extraneous Functionality	●	●



## Security

## Privacy



35

43



82

50

# Bank - US 49p

## OWASP Mobile Top 10

The Android app uses unsecure storage data mode (WORLD\_READABLE, WORLD\_WRITABLE). The app utilizes the Java.io.File delete() method to perform a file deletion. This method is not a secure file deletion method. This application uses synthetic methods to access private class entries, which are normally not accessible. This is a suspicious and unusual coding practice.

Improper Platform Usage  
Insecure Data Storage  
Insecure Authentication  
Insufficient Cryptography  
Insecure Authorization  
Client Code Quality  
Code Tampering  
Reverse Engineering  
Extraneous Functionality



Security

Privacy

82

50



# Bank - US 18j

The iOS app is implementing the EventKit Framework which enables it to gain access to calendar data. 'Bearer' related oauth tokens were found. An adversary could potentially gain access to these tokens using the Juice Jacking approach if they are not encrypted. The app implements Swizzling API calls. This could indicate malware activity and may impact the app ability to trust security decisions. The Android app can execute commands at the OS level such as launching other applications and processes and can access system commands that require a rooted device. The app uses unsecure storage data mode (WORLD\_READABLE, WORLD\_WRITABLE).

## OWASP Mobile Top 10



Improper Platform Usage	●	●
Insecure Data Storage	●	●
Insecure Communications	●	●
Insecure Authentication	●	●
Insufficient Cryptography	●	●
Insecure Authorization	●	●
Client Code Quality	●	●
Code Tampering	●	●
Reverse Engineering	●	●
Extraneous Functionality	●	●



## Security

## Privacy



35

55



82

55

# Bank - US 18j

OWASP Mobile Top 10



Security

Privacy

The Android app can execute commands at the OS level such as launching other applications and processes and can access system commands that require a rooted device. The app uses unsecure storage data mode (WORLD\_READABLE, WORLD\_WRITABLE).

Insecure Data Storage

Insecure Communications

Insecure Authentication

Insufficient Cryptography

Insecure Authorization

Client Code Quality

Code Tampering

Reverse Engineering

Extraneous Functionality



82

55

# Bank - US 98y

The iOS app is actively monitoring the pasteboard and sends query parameters with private information such as the username. The app is compiled without implementing Stack smashing protection. This app implements AdMob advertising network. The Android app uses the WebKit to download a file from the Internet. The app enables WebView to execute JavaScript code. This could potentially allow an attacker to introduce arbitrary JavaScript code to perform malicious actions. This app is actively targeted by the BankBot malware campaign.

## OWASP Mobile Top 10



Improper Platform Usage	●	●
Insecure Data Storage	●	●
Insecure Communications	●	●
Insecure Authentication	●	●
Insufficient Cryptography	●	●
Insecure Authorization	●	●
Client Code Quality	●	●
Code Tampering	●	●
Reverse Engineering	●	●
Extraneous Functionality	●	●



## Security

## Privacy



47

82



76

47

# Bank - US 98y

OWASP Mobile Top 10



Security

Privacy

The Android app uses the WebKit to download a file from the Internet. The app enables WebView to execute JavaScript code. This could potentially allow an attacker to introduce arbitrary JavaScript code to perform malicious actions. This app is actively targeted by the BankBot malware campaign.

Insufficient Cryptography

Insecure Authorization

Client Code Quality

Code Tampering

Reverse Engineering

Extraneous Functionality



76

47



# Bank - US 51b

The iOS app sends query parameters with private information such as the password. The iOS app implements an over-the air app installation and links to several non-HTTPS link including a personal email address. The app was compiled without implementing Stack smashing protection. The Android app enables WebView to execute JavaScript code. The app grants one or more permissions to content provider's data. This could potentially lead to the disclosure of information. The Android app is actively targeted by the BankBot malware campaign.

## OWASP Mobile Top 10



	Apple	Android
Improper Platform Usage	●	●
Insecure Data Storage	●	●
Insecure Communications	●	●
Insecure Authentication	●	●
Insufficient Cryptography	●	●
Insecure Authorization	●	●
Client Code Quality	●	●
Code Tampering	●	●
Reverse Engineering	●	●
Extraneous Functionality	●	●



## Security

## Privacy



64

59



78

78

# Bank - US 51b

## OWASP Mobile Top 10



## Security

## Privacy

The Android app enables WebView to execute JavaScript code. The app grants one or more permissions to content provider's data. This could potentially lead to the disclosure of information. The Android app is actively targeted by the BankBot malware campaign.

Insufficient Cryptography

Insecure Authorization

Client Code Quality

Code Tampering

Reverse Engineering

Extraneous Functionality



78

78

# Bank - US 76r

The iOS app sends query parameters with private information such as the IP address. The app is using a non-encrypted HTTP connection. The app implements low-level API calls to retrieve the device global Ad identifier key and implements low-level telephony enumeration API calls. The Android app uses synthetic methods to access private class entries which are normally not accessible. This is a suspicious and unusual coding practice that should be reviewed. The permission to one or more content provider's data is granted. This could potentially lead to the disclosure of information. The app is actively targeted by the BankBot malware campaign.

## OWASP Mobile Top 10



	Apple	Android
Improper Platform Usage	●	●
Insecure Data Storage	●	●
Insecure Communications	●	●
Insecure Authentication	●	●
Insufficient Cryptography	●	●
Insecure Authorization	●	●
Client Code Quality	●	●
Code Tampering	●	●
Reverse Engineering	●	●
Extraneous Functionality	●	●



## Security

## Privacy



38

57



70

65