Supercharge Your AppSec Program with OWASP | Appdome Consumer Mobile Security Report and OWASP MASVS

Brian Reed | SVP Mobile Defense

brianr@Appdome.com

Who Works on Mobile App Security?

whoami

Brian Reed
SVP Mobile Defense
Appdome

brianr@Appdome.com

Connect with me on Linkedin in and DM me for this deck and more resources

~20 Years in Mobile

Remember when BlackBerry ruled the world? Now I live on iOS, Droid, Apple Watch, Oura Ring….

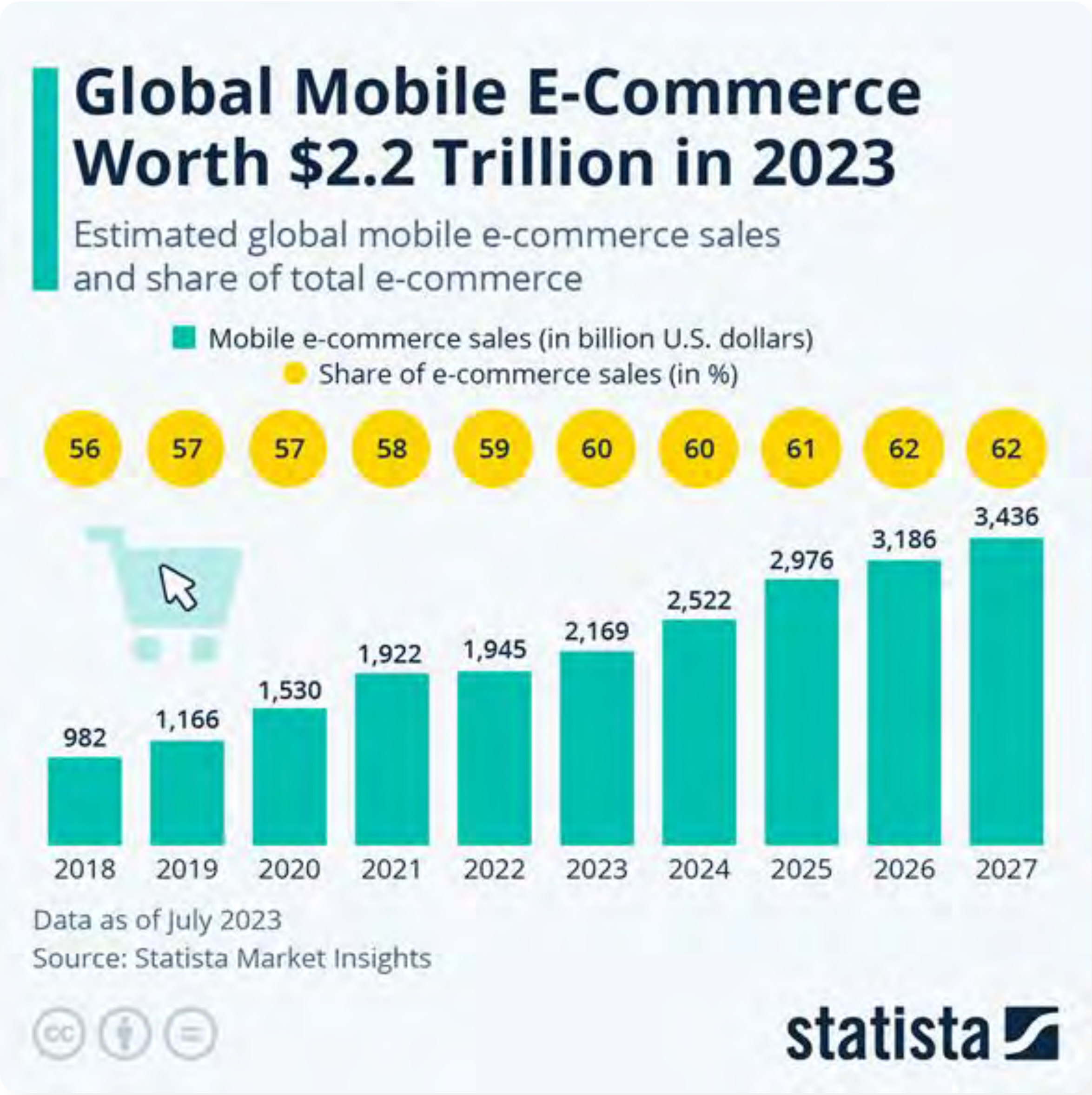
Appdome, NowSecure, Good Technology, BlackBerry, ZeroFOX, BoxTone, and MicroFocus/Intersolv

~8 Years in OWASP

OWASP MAS Advocate

appdome

OWASP



Global Mobile E-Commerce Worth $2.2 Trillion in 2023

Estimated global mobile e-commerce sales and share of total e-commerce

Data as of July 2023

Source: Statista Market Insights

Mobile Attacks are Growing, Powered by AI

Black Mamba

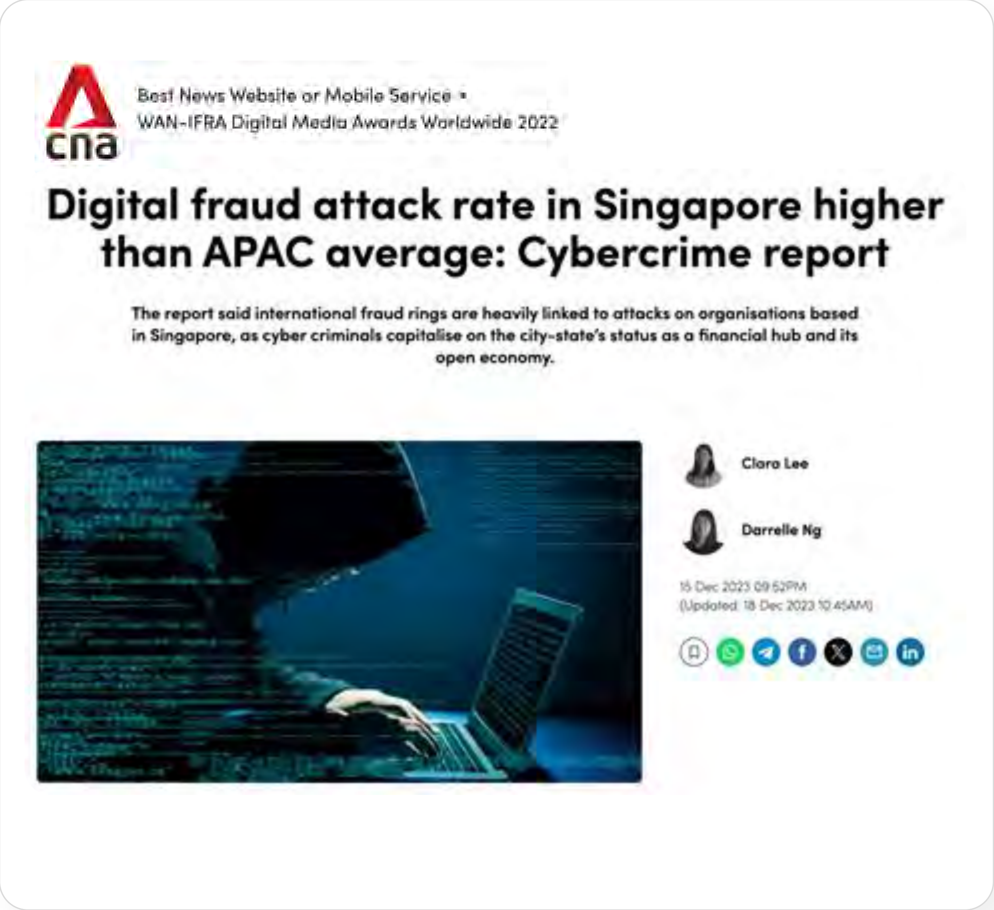
Polymorphic Keylogging C2C Malware capable of bypassing EDR systems. *Used in mobile endpoints and applications for some time.*

Deep Fakes & Face ID Bypass

Deepfake face attacks on ID verification systems up 704% in 2023. *Holy grail of authentication is now, with SwapCam, weakest link.*

Vishing & Voice Cloning

In fourth quarter 2023, vishing attacks rose by 260 percent .
Fake mobile calls using cloned voices make cyber awareness training near obsolete.



Best News Website or Mobile Service •
WAN-IFRA Digital Media Awards Worldwide 2022

Digital fraud attack rate in Singapore higher than APAC average: Cybercrime report

The report said international fraud rings are heavily linked to attacks on organisations based in Singapore, as cyber criminals capitalise on the city-state's status as a financial hub and its open economy.

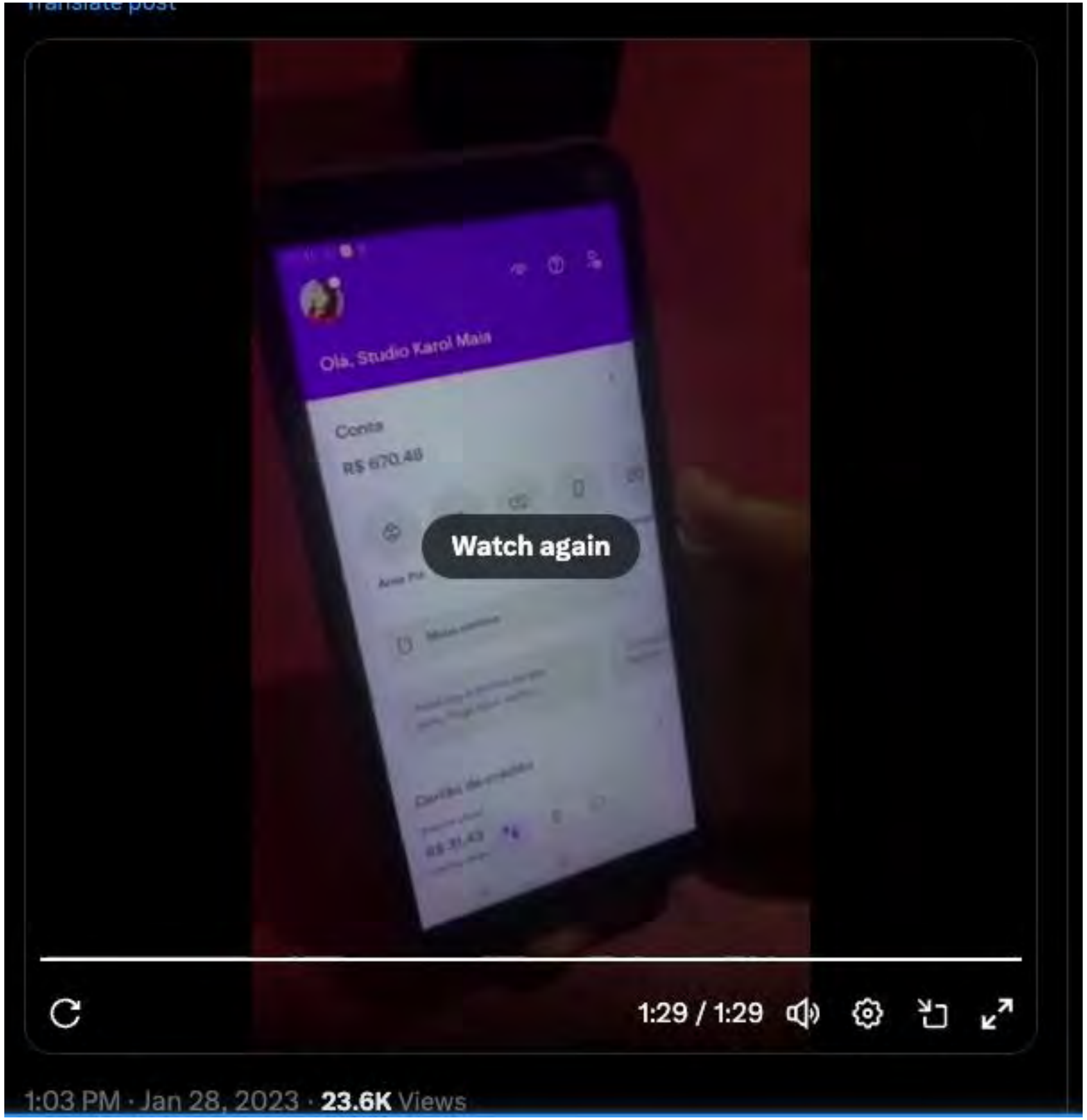
Clara Lee

Darrelle Ng

15 Dec 2023 09:52PM
(Updated: 18 Dec 2023 10:45AM)

Mobile App Overlay Attack

Exploiting Android Accessibility Services to Steal Money



1:03 PM · Jan 28, 2023 · **23.6K** Views

Who has to battle with dev & the business to prioritize AppSec?

What if we brought consumer voice into the conversation?

appdome

2024 Singapore Consumer Expectations of Mobile App Security

© 2024 – Appdome Inc., All Rights Reserved

appdome | OWASP

2024 Philippines Consumer Expectations of Mobile App Security

© 2024 – Appdome Inc., All Rights Reserved

appdome | OWASP

2024 Australia Consumer Expectations of Mobile App Security

© 2024 – Appdome Inc., All Rights Reserved

Scan to Download

Appdome + OWASP Partnership for Consumer Voice

Scan to Download

Introduction

The Open Worldwide Application Security Project (OWASP) is the world's largest nonprofit foundation and global community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain secure applications that can be trusted.

Thousands of teams around the world build software applications of all types including web, cloud, mobile, IoT and more across a myriad of platforms. The OWASP Mobile App Security (MAS) flagship project provides a proven, community-driven security standard for mobile teams to deliver high quality, secure mobile apps.

In 2024, OWASP is proud to join Appdome to bring the voice of Consumer Expectations of Mobile App Security to the OWASP community and raise awareness among cyber practitioners at software organizations to the cybersecurity and anti-fraud demands of everyday mobile end users. The report provides a powerful voice of consumer expectations about mobile app security and privacy protection.

The report highlights three critical areas relevant to all mobile brands and enterprises:

1. People depend extensively on mobile apps in life and work.
2. People have strong and detailed expectations of security, anti-fraud and data privacy in their mobile applications.
3. People will reward brands and enterprises that protect their mobile use against these threats.

These consumer perspectives on fraud, security, privacy and trust are highly valuable for every organization. Cybersecurity leaders can leverage this consumer and employee feedback into the conversation about proper defense models, requirements and practices. This same voice can be used with business leaders, development teams, and other stakeholders to show the OWASP mobile standard matters to app users more than ever.

The 2024 Appdome Consumer Survey confirms that the OWASP mobile standard is more than the best practice guideline for mobile app security. The OWASP mobile standard is the baseline expectation of global consumers. This validation reinforces the critical value of OWASP mobile standard and the importance of implementing comprehensive security, anti-fraud, anti-malware and other defensive measures in mobile applications. Mobile software teams should leverage the OWASP Mobile App Security project and resources as a baseline for their mobile security and privacy programs. Mobile software teams should also leverage the data in the 2024 Global Consumer Survey to deploy the security, anti-fraud, anti-malware and data privacy protections demanded by all users, globally.

Andrew van der Stock

Executive Director, OWASP Foundation

© 2024 – Appdome Inc. | Singapore Consumer Expectations of Mobile App Security

appdome | OWASP

Mobile Fraud Is Top of Mind for the Consumer

Mobile fraud became the top concern for Singaporeans as AI-based attacks grew in both intensity and diversity. Consumers are surprisingly aware of the many ways mobile fraud occurs.

Many Singaporeans Are Victims

One of the most startling data points from the 2024 survey is the significant percentage of consumers in Singapore who have been affected by mobile fraud and social engineering scams. 32.7% of Singaporeans responded that they or someone close to them has experienced a social engineering scam, and 38.4% said that they or someone close to them has experienced cyber-attacks, mobile malware, or mobile fraud. The actual impact of these attacks is likely to be much higher, as artificial intelligence, deep fakes, voice cloning and the use of other sophisticated malware and other methods make it almost impossible for mobile users to detect attacks until it's too late.

Have you, a family member, or a friend experienced a cyber-attack, malware, or mobile fraud?

Have you, or anyone you know, experienced a social engineering scam?

Diversity of Consumer Fears Is Growing

At 49.9% and 49.1%, fraud and hacking top the list of Singaporean consumer concerns, which is not surprising given the epidemic nature of scams and mobile fraud as well as the stringent government response to it. Other top fears expressed by consumers are mobile malware and Trojans, which have increased to 40.0% and 20.5%, respectively. The data also reveals a slight uptick in consumer fears over network-based attacks, up about 6.42% from 2022.

What on-device threats most concern you as a user of mobile apps?

How have cyber security threats vs fraud evolved over time?

How Important are Network Level Threats such as Data Breaches & MiTM Attacks?

Social Engineering Emerges as a Top Concern

In 2024, we asked Singaporeans which social engineering scams pose the biggest risk. 71.7% of Singaporeans reported SMS phishing ("smishing") as the biggest risk, followed by 66.1% for email phishing, 64.5% for voice phishing ("vishing"), and 56.4% for brand impersonation scams. Vishing and smishing now rival traditional email phishing as top threats to mobile users and apps. This rise underscores the evolving threat landscape and the need for strong anti-fraud measures. Also noteworthy, Singaporeans are 28.6% more concerned about love scams in mobile apps than their global peers, likely due to frequent public service campaigns in Singapore warning against such scams.

Which social engineering scams pose the biggest risk (select 3)?

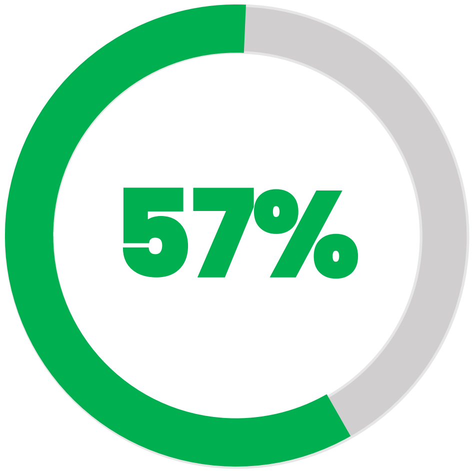
© 2024 – Appdome Inc. | Singapore Consumer Expectations of Mobile App Security

appdome | OWASP

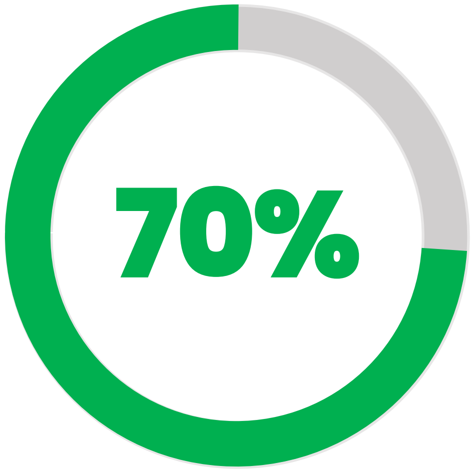
appdome

OWASP

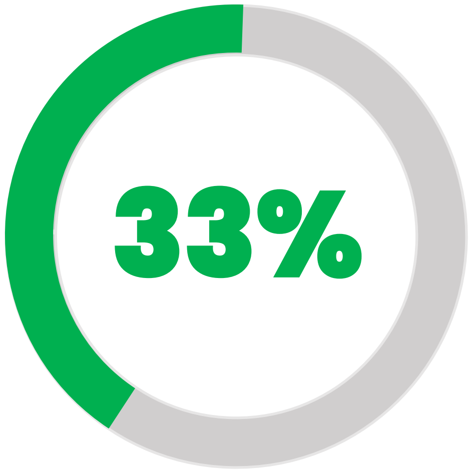
Singaporeans Primarily Live and Work on Mobile Apps, Powering Communication & mCommerce



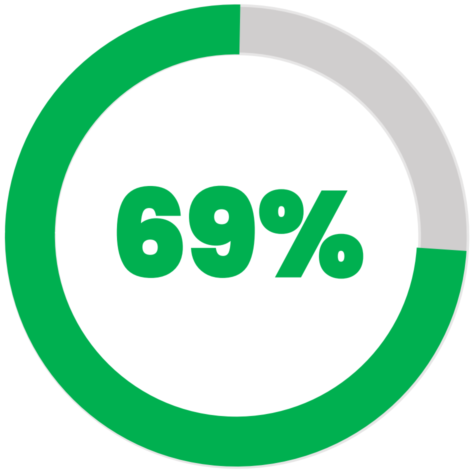
prefer mobile apps over other channels to buy goods and services



use more than 5 apps on average per week

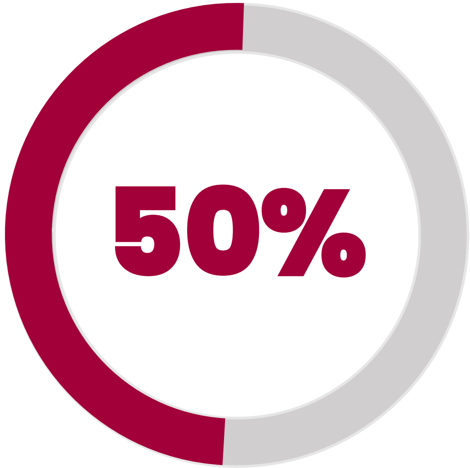


use more than 10 apps on average per week

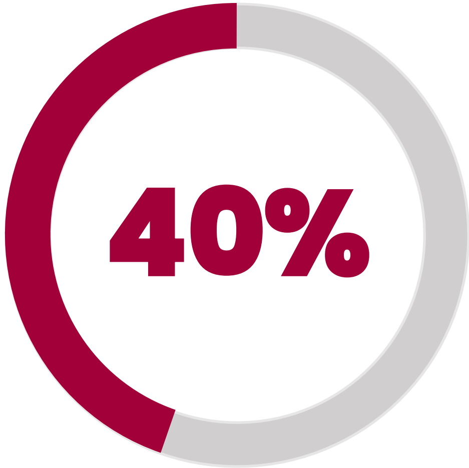


say that their use of mobile apps has increased over the last 12 months

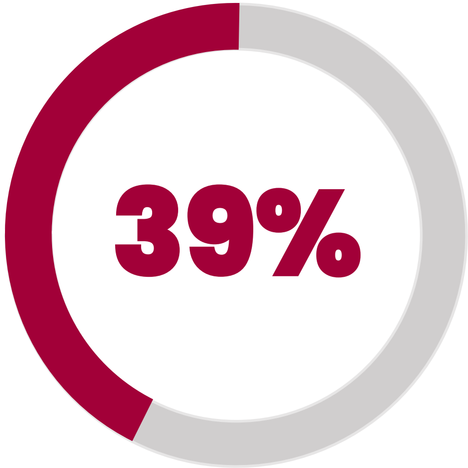
Singaporeans Are Concerned About Mobile Fraud, Social Engineering and On-device Threats



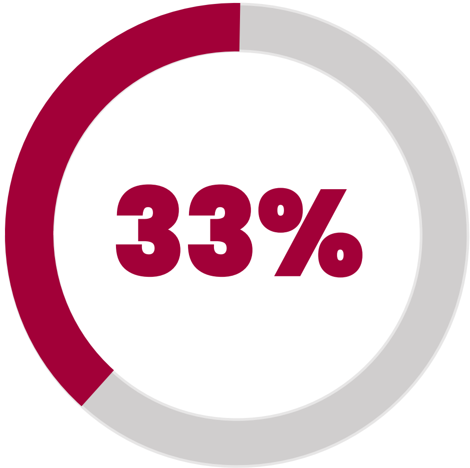
Say **mobile fraud** is their top fear



Fear on-device **malware**

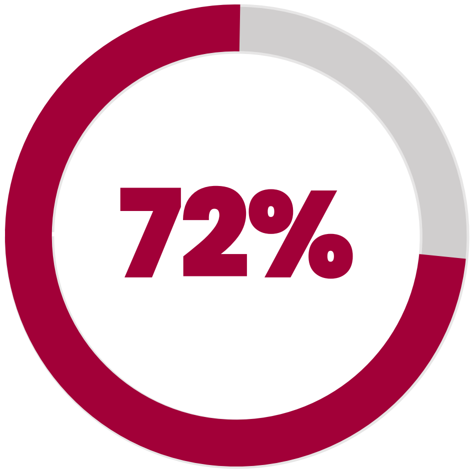


have been affected by **mobile fraud**

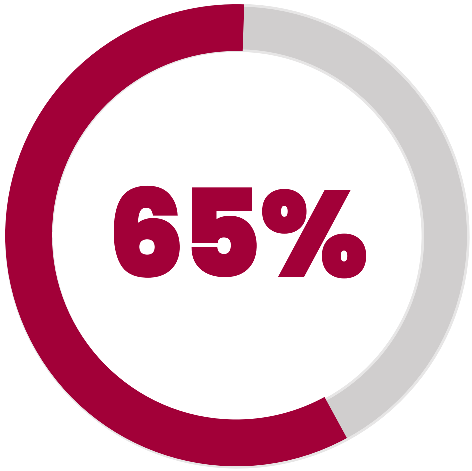


have encountered **social engineering scams**

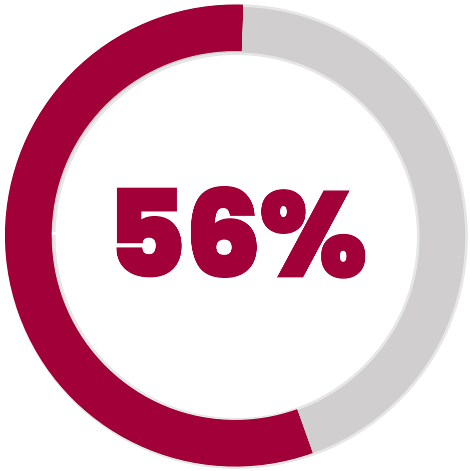
Singaporeans have Suffered from Social Engineering Attacks and Scams on Mobile



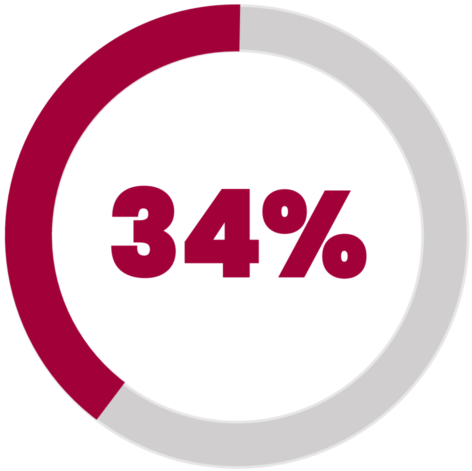
have experienced **smishing**



have experienced **vishing**

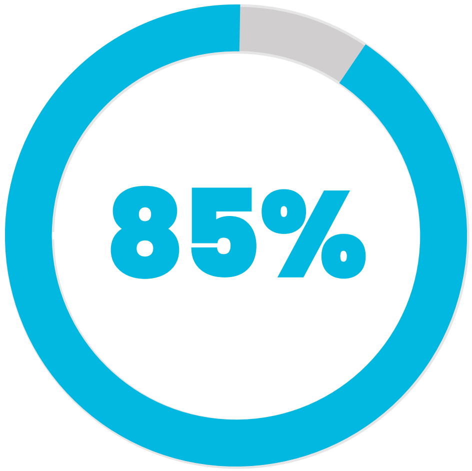


have experienced **brand impersonation scams**

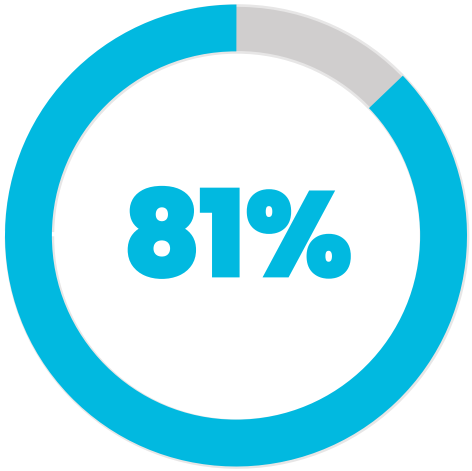


fear **love scams** (28.5% higher than global)

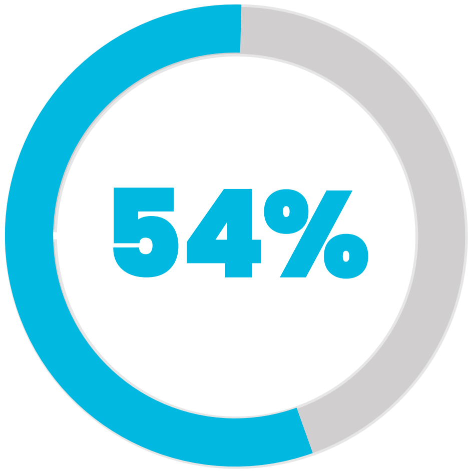
Responsibility for Mobile App Security Has Shifted to the Mobile App Makers



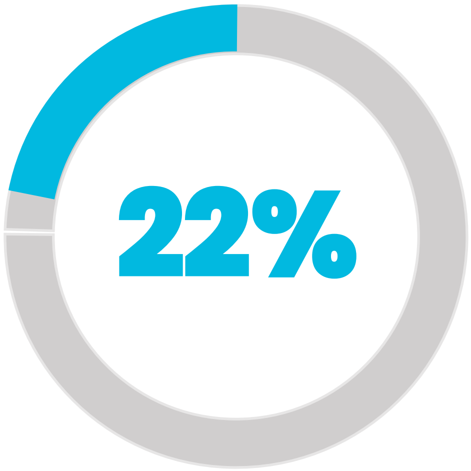
of Singaporeans say **security and privacy** are equal to more important than features



prefer **preemptive fraud protection** vs reimbursement after fraud happened

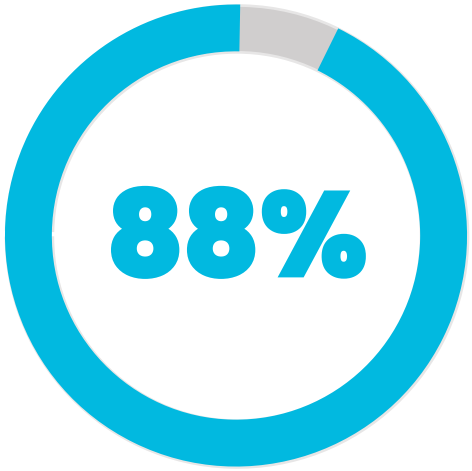


say it is the **developer or brand's job to protect them** against cyber threats, fraud, malware, privacy leaks

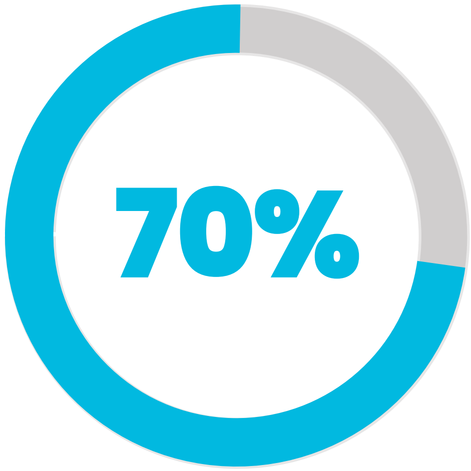


believe **app makers and developers don't care** about protecting them from fraud and security threats

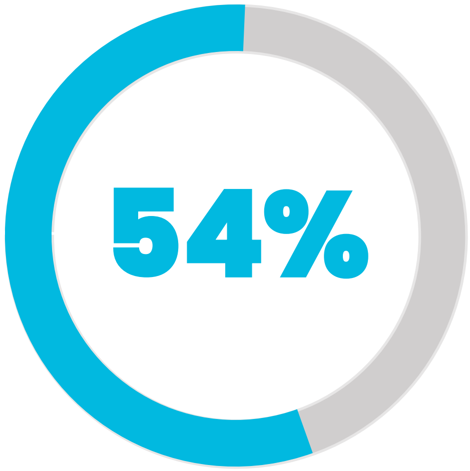
Singaporeans Seek Trust and Safety on Mobile, Will Abandon When They are Not Protected



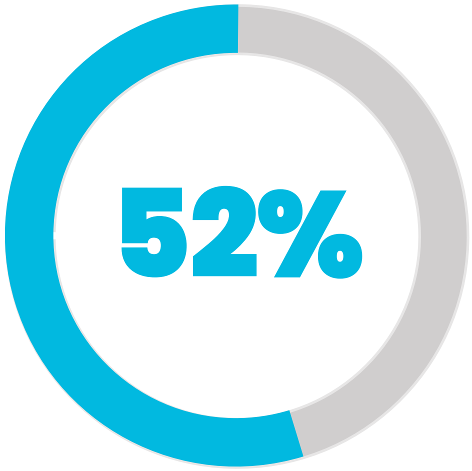
of Singaporeans say they **seek out info about security and privacy** measures in mobile apps before using them



will **abandon a mobile brand after a breach** and will also tell their friends to do

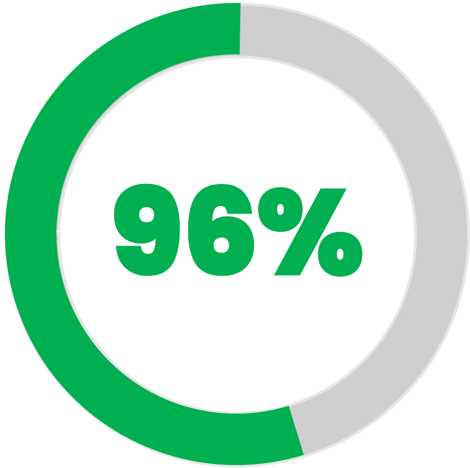


are likely or **very likely to stop using an app** if it failed to protect them



said they have **deleted or stopped using a mobile app** due to security or privacy concerns

Use Security As a Differentiator to Grow Your Business



of Singaporean consumers confirmed their **willingness to promote security-conscious brands** with visible and public forms of advocacy

(with likes, hashtags,
positive app store reviews and
brand advocacy)

Download Now

appdome

2024 Singapore Consumer Expectations of Mobile App Security

© 2024 – Appdome Inc. All Rights Reserved

2024

Singapore Consumer Report

appdome | OWASP

2024 Philippines Consumer Expectations of Mobile App Security

© 2024 – Appdome Inc. All Rights Reserved

2024

Philippines Consumer Report

appdome | OWASP

2024 Australia Consumer Expectations of Mobile App Security

© 2024 – Appdome Inc. All Rights Reserved

2024

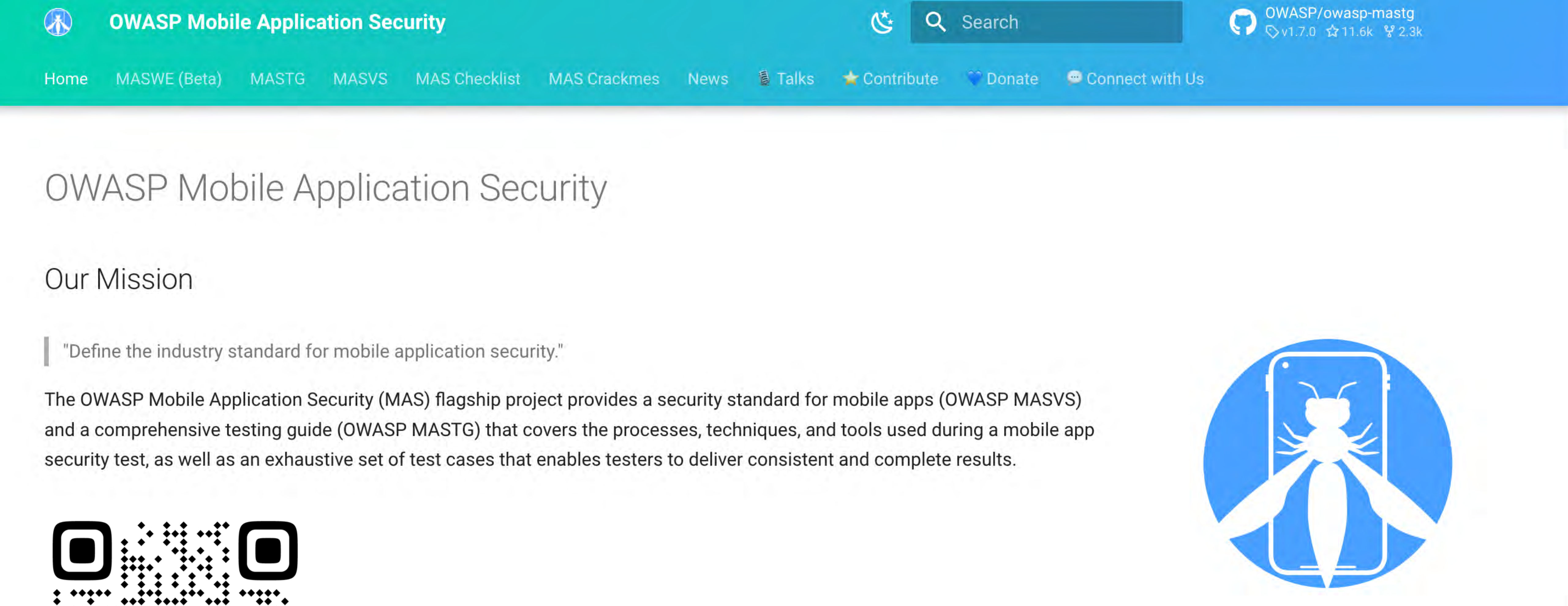
Australia Consumer Report

Scan to Download

Who thinks the Consumer Survey will help convince their business to prioritize mobile security?

Tap into OWASP MAS as The Foundation of Your Mobile AppSec Program

Who uses the OWASP Mobile Project?



OWASP Mobile Application Security

Our Mission

> "Define the industry standard for mobile application security."

The OWASP Mobile Application Security (MAS) flagship project provides a security standard for mobile apps (OWASP MASVS) and a comprehensive testing guide (OWASP MASTG) that covers the processes, techniques, and tools used during a mobile app security test, as well as an exhaustive set of test cases that enables testers to deliver consistent and complete results.

https://mas.owasp.org

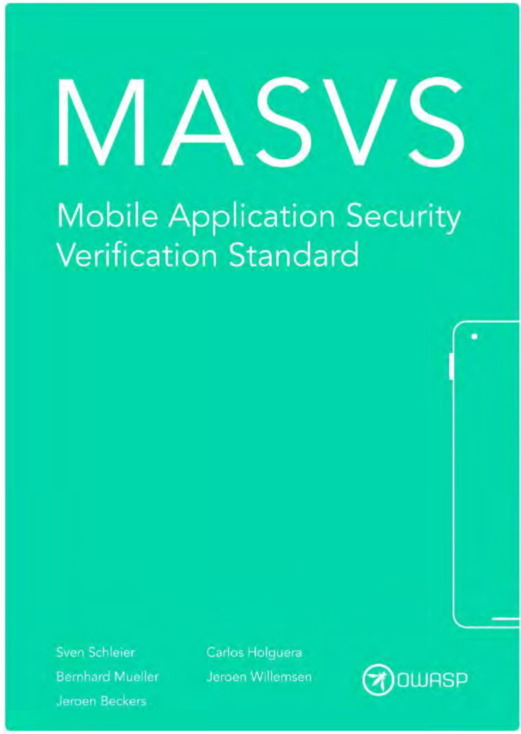
OWASP Mobile Application Security

SAFE APP STANDARD

CSA SINGAPORE

Cyber Security Agency of Singapore

OWASP MASVS



MASVS

Mobile Application Security Verification Standard

Sven Schleier
Bernhard Mueller
Jeroen Beckers

Carlos Holguera
Jeroen Willemsen

OWASP

OWASP MASWE

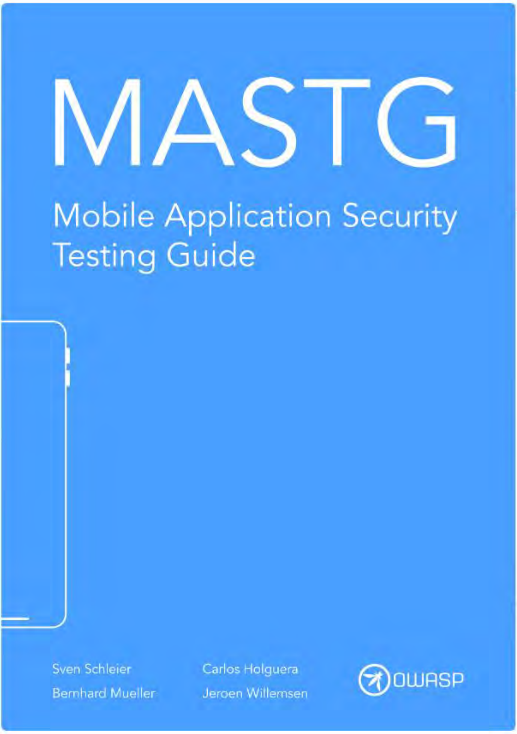
MASWE

Mobile Application Security Weakness Enumeration

Sven Schleier
Carlos Holguera
Jeroen Beckers

OWASP

OWASP MASTG



MASTG

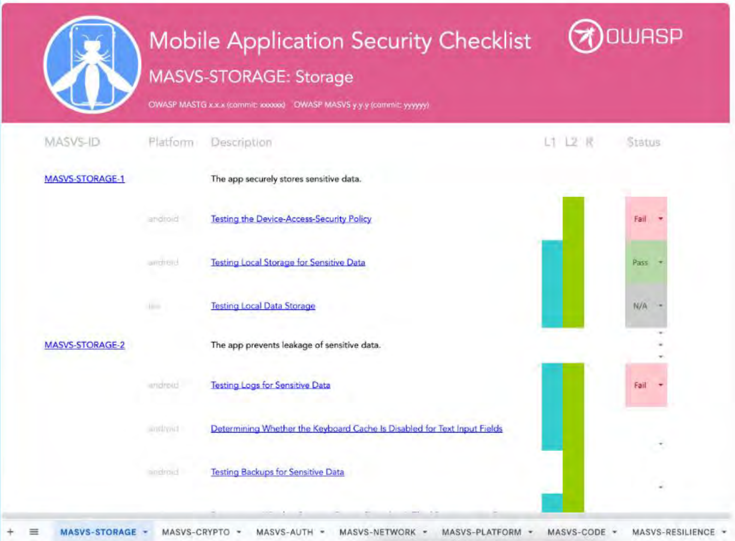
Mobile Application Security Testing Guide

Sven Schleier
Bernhard Mueller

Carlos Holguera
Jeroen Willemsen

OWASP

OWASP MAS Checklist



Mobile Application Security Checklist

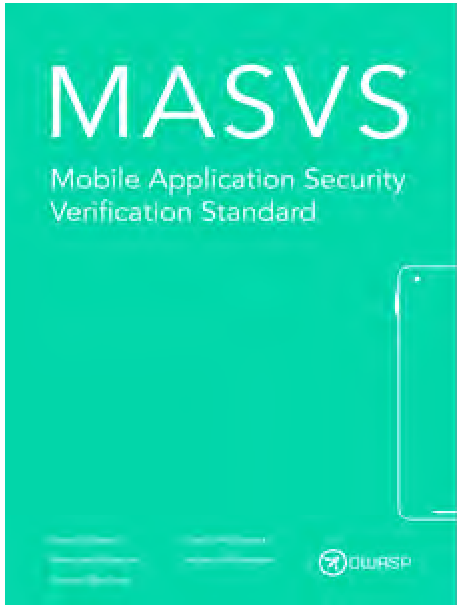
MASVS-STORAGE: Storage

OWASP MASTG x.x.x (commit: xxxxxx) OWASP MASVS y.y.y (commit: yyyyyy)

OWASP

MASVS-ID	Platform	Description	L1	L2	R	Status
MASVS-STORAGE-1		The app securely stores sensitive data.				
	android	Testing the Device-Access-Security Policy		🟩		Fail
	android	Testing Local Storage for Sensitive Data	🟦	🟩		Pass
	ios	Testing Local Data Storage	🟦	🟩		N/A
MASVS-STORAGE-2		The app prevents leakage of sensitive data.				
	android	Testing Logs for Sensitive Data	🟦	🟩		Fail
	android	Determining Whether the Keyboard Cache Is Disabled for Text Input Fields	🟦	🟩		
	android	Testing Backups for Sensitive Data		🟩		

New MASWE (Beta)



MASVS

Mobile Application Security Verification Standard

OWASP

MASWE

Mobile Application Security Weakness Enumeration

Sven Schleier
Carlos Holguera
Jeroen Beckers

OWASP

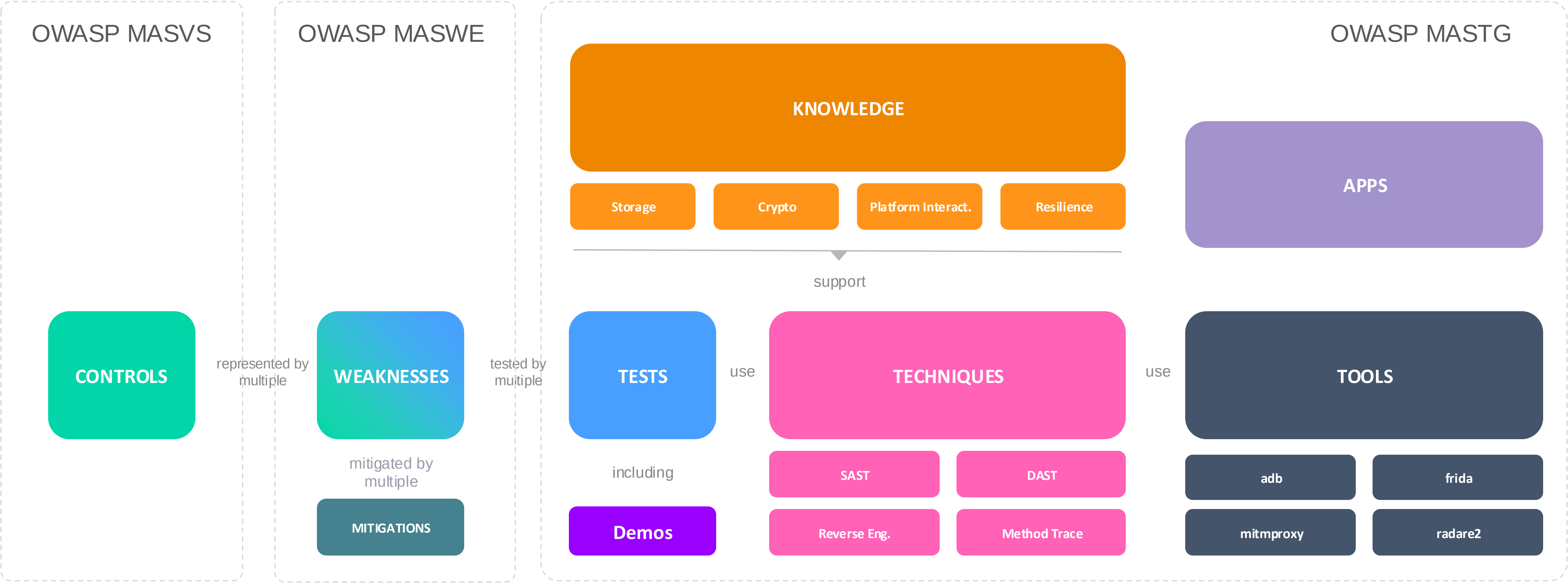
MASTG

Mobile Application Security Testing Guide

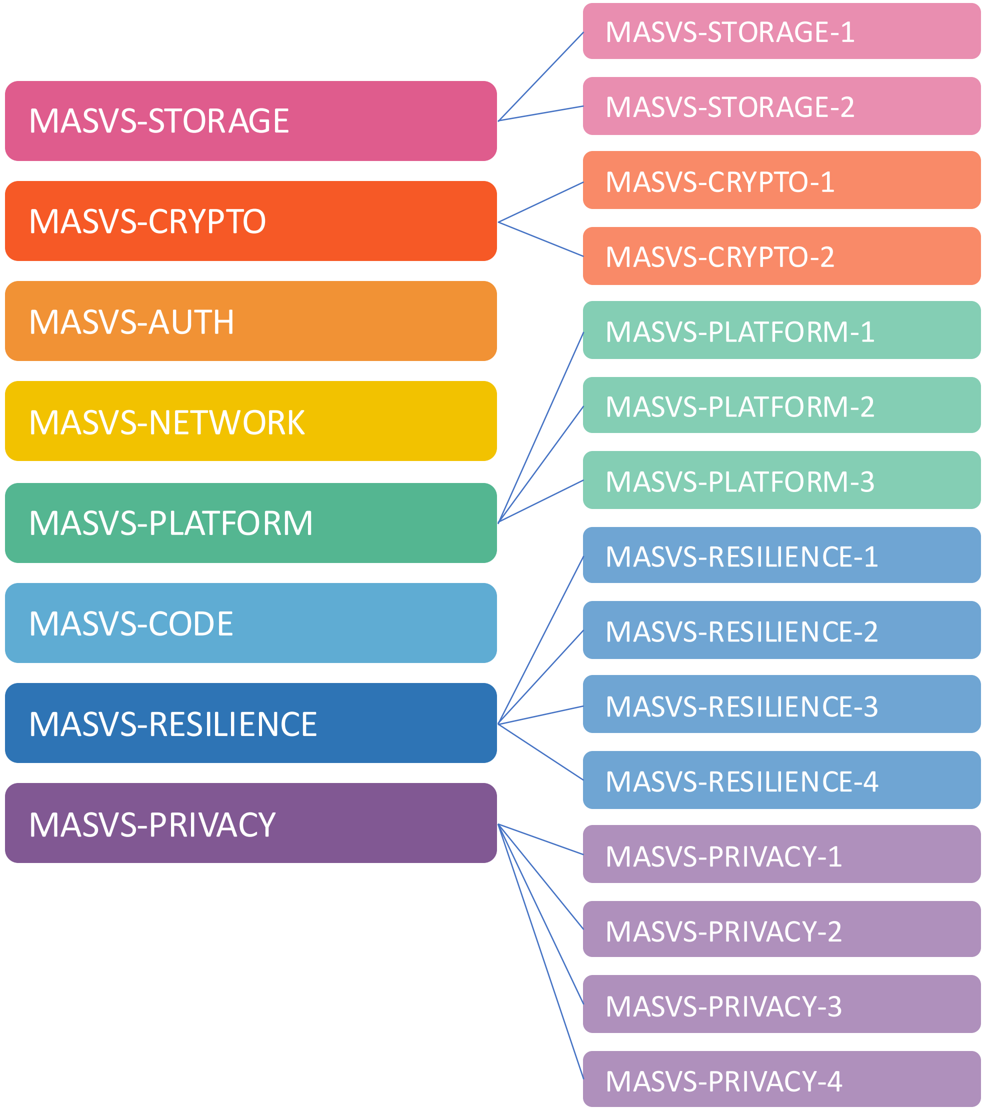
OWASP

https://mas.owasp.org/news/2024/07/30/new-maswe/

OWASP MAS V2

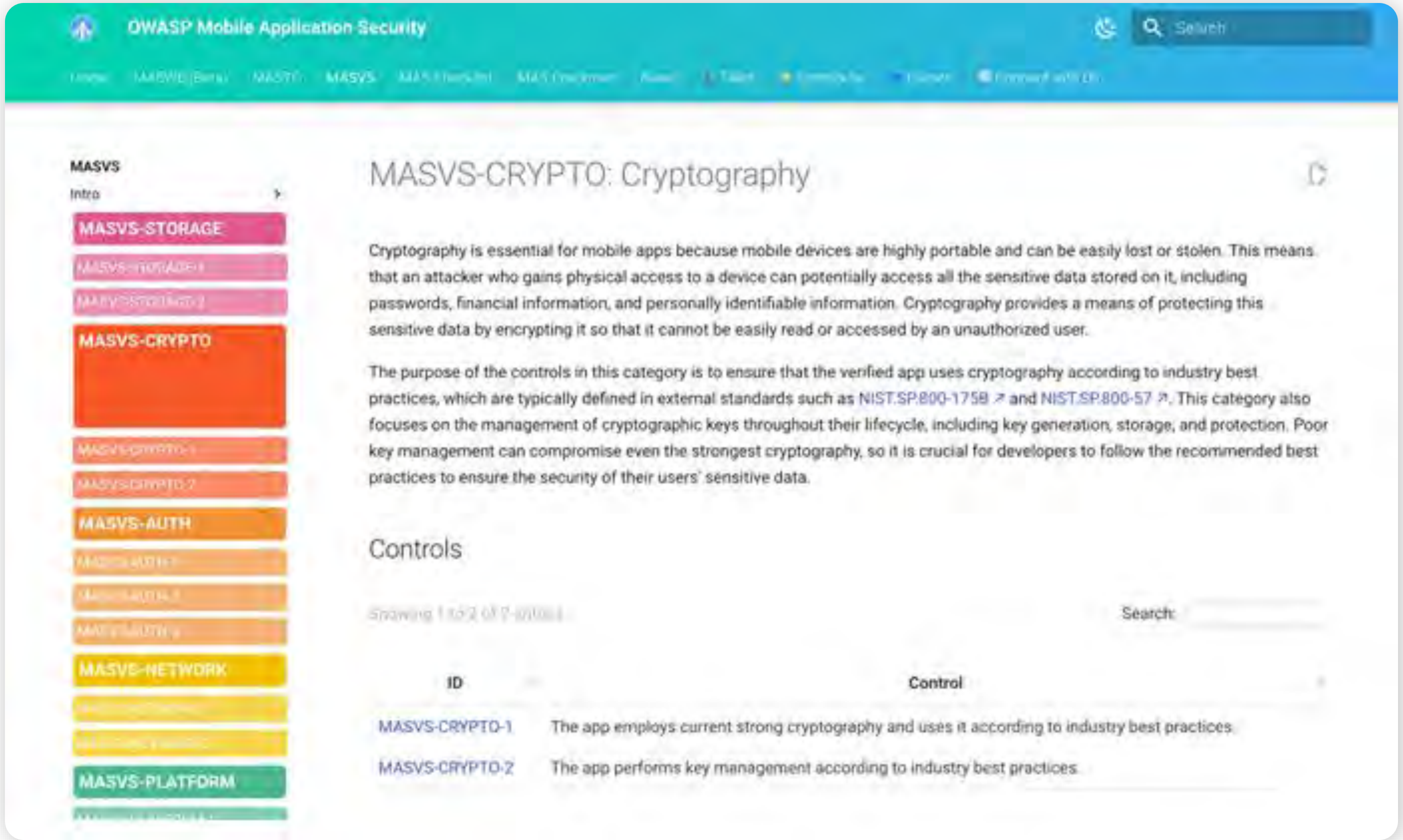

OWASP MASVS
OWASP MASWE
OWASP MASTG
KNOWLEDGE
Storage
Crypto
Platform Interact.
Resilience
support
APPS
CONTROLS
represented by multiple
WEAKNESSES
tested by multiple
TESTS
use
TECHNIQUES
use
TOOLS
mitigated by multiple
MITIGATIONS
including
Demos
SAST
DAST
Reverse Eng.
Method Trace
adb
frida
mitmproxy
radare2

OWASP MASVS V2 Controls



Category	Control	Description
MASVS-STORAGE	MASVS-STORAGE-1	The app securely stores sensitive data
	MASVS-STORAGE-2	The app prevents leakage of sensitive data
MASVS-CRYPTO	MASVS-CRYPTO-1	The app employs current strong crypto & uses it according to industry best practices
	MASVS-CRYPTO-2	The app performs key management according to industry best practices
MASVS-AUTH		
MASVS-NETWORK		
MASVS-PLATFORM	MASVS-PLATFORM-1	The app uses IPC mechanisms securely
	MASVS-PLATFORM-2	The app uses WebViews securely
	MASVS-PLATFORM-3	The app uses the user interface securely
MASVS-CODE		
MASVS-RESILIENCE	MASVS-RESILIENCE-1	The app validates the integrity of the platform
	MASVS-RESILIENCE-2	The app implements anti-tampering
	MASVS-RESILIENCE-3	The app implements anti-static analysis mechanisms
	MASVS-RESILIENCE-4	The app implements anti-dynamic analysis
MASVS-PRIVACY	MASVS-PRIVACY-1	The app minimizes access to sensitive data and resources
	MASVS-PRIVACY-2	The app prevents identification of the user
	MASVS-PRIVACY-3	The app is transparent about data collection and storage
	MASVS-PRIVACY-4	The app offers user control over their data

MASVS Detail



MASVS-CRYPTO: Cryptography

Cryptography is essential for mobile apps because mobile devices are highly portable and can be easily lost or stolen. This means that an attacker who gains physical access to a device can potentially access all the sensitive data stored on it, including passwords, financial information, and personally identifiable information. Cryptography provides a means of protecting this sensitive data by encrypting it so that it cannot be easily read or accessed by an unauthorized user.

The purpose of the controls in this category is to ensure that the verified app uses cryptography according to industry best practices, which are typically defined in external standards such as NIST.SP.800-175B ↗ and NIST.SP.800-57 ↗. This category also focuses on the management of cryptographic keys throughout their lifecycle, including key generation, storage, and protection. Poor key management can compromise even the strongest cryptography, so it is crucial for developers to follow the recommended best practices to ensure the security of their users' sensitive data.

Controls

Showing 1 to 2 of 2 entries

Search:

ID	Control
MASVS-CRYPTO-1	The app employs current strong cryptography and uses it according to industry best practices.
MASVS-CRYPTO-2	The app performs key management according to industry best practices.

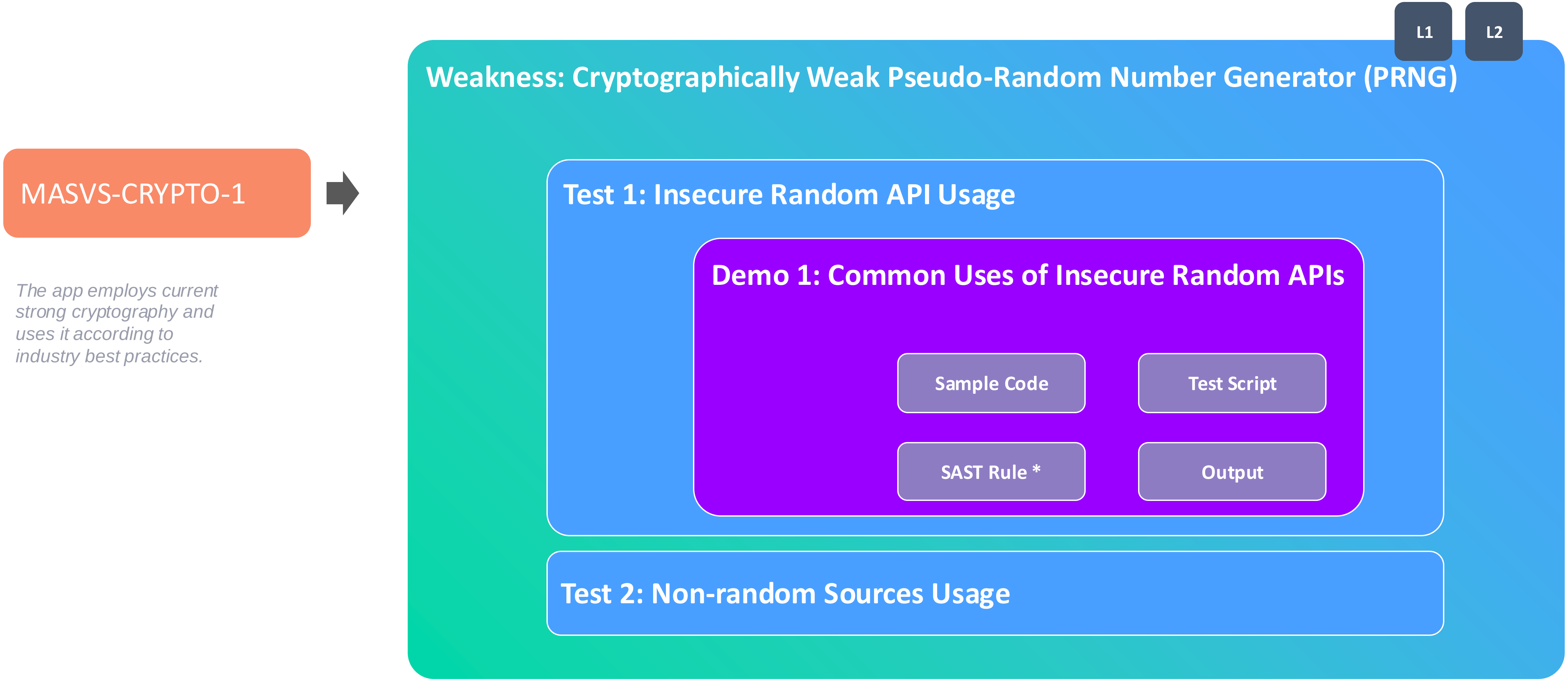
What is Crypto?

What can happen?

What is purpose?

What are relevant industry standards?

OWASP MAS V2 Example



MASVS-CRYPTO-1

The app employs current strong cryptography and uses it according to industry best practices.

L1

L2

Weakness: Cryptographically Weak Pseudo-Random Number Generator (PRNG)

Test 1: Insecure Random API Usage

Demo 1: Common Uses of Insecure Random APIs

Sample Code

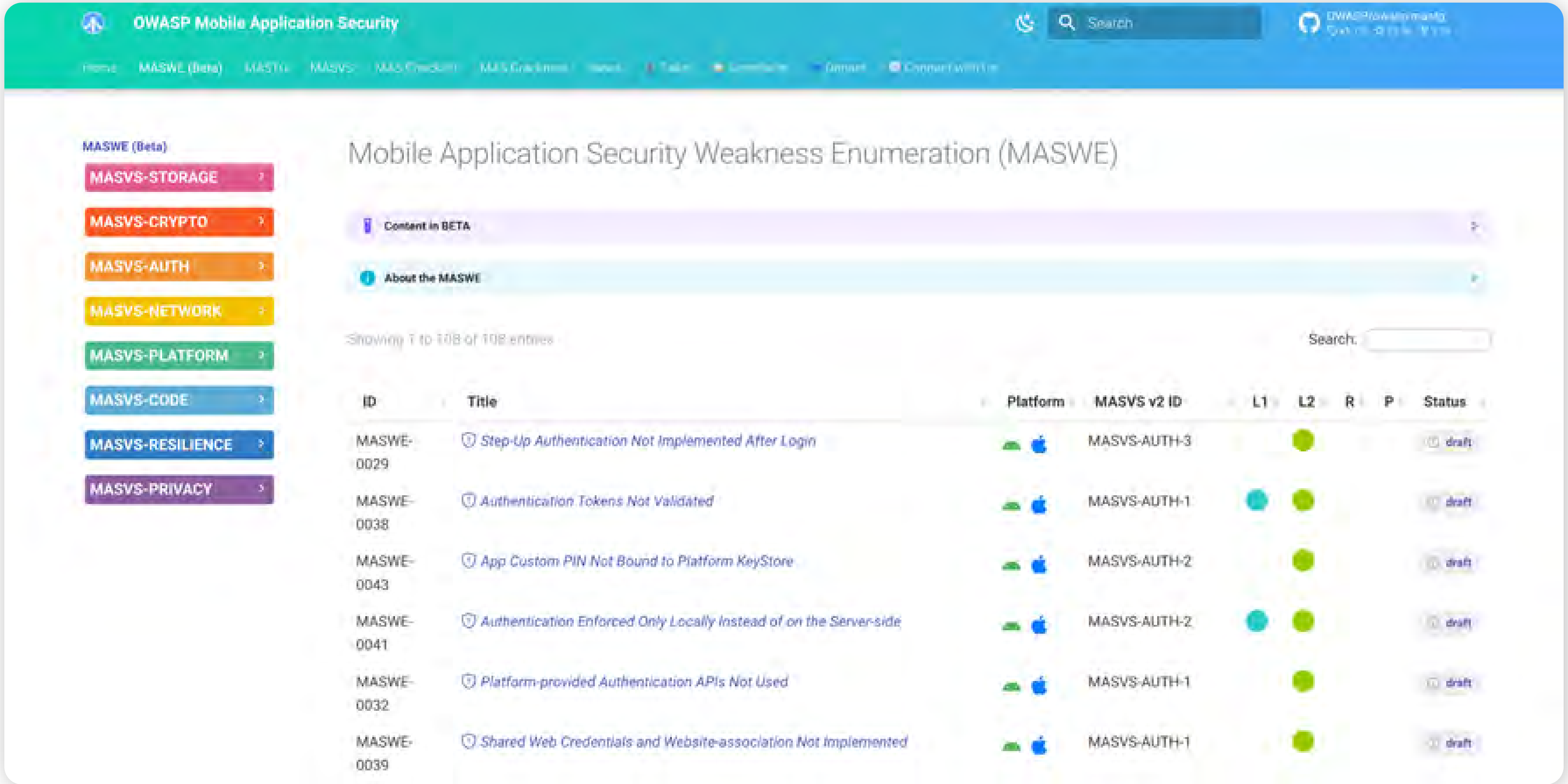
Test Script

SAST Rule *

Output

Test 2: Non-random Sources Usage

MASWE



OWASP Mobile Application Security

Search

OWASP/owasp-mastg

MASWE (Beta)

- MASVS-STORAGE
- MASVS-CRYPTO
- MASVS-AUTH
- MASVS-NETWORK
- MASVS-PLATFORM
- MASVS-CODE
- MASVS-RESILIENCE
- MASVS-PRIVACY

Mobile Application Security Weakness Enumeration (MASWE)

Content in BETA

About the MASWE

Showing 1 to 108 of 108 entries

Search:

ID	Title	Platform	MASVS v2 ID	L1	L2	R	P	Status
MASWE-0029	*Step-Up Authentication Not Implemented After Login*	🤖 🍎	MASVS-AUTH-3		●			draft
MASWE-0038	*Authentication Tokens Not Validated*	🤖 🍎	MASVS-AUTH-1	●	●			draft
MASWE-0043	*App Custom PIN Not Bound to Platform KeyStore*	🤖 🍎	MASVS-AUTH-2		●			draft
MASWE-0041	*Authentication Enforced Only Locally Instead of on the Server-side*	🤖 🍎	MASVS-AUTH-2	●	●			draft
MASWE-0032	*Platform-provided Authentication APIs Not Used*	🤖 🍎	MASVS-AUTH-1		●			draft
MASWE-0039	*Shared Web Credentials and Website-association Not Implemented*	🤖 🍎	MASVS-AUTH-1		●			draft

Unique IDs

Titles

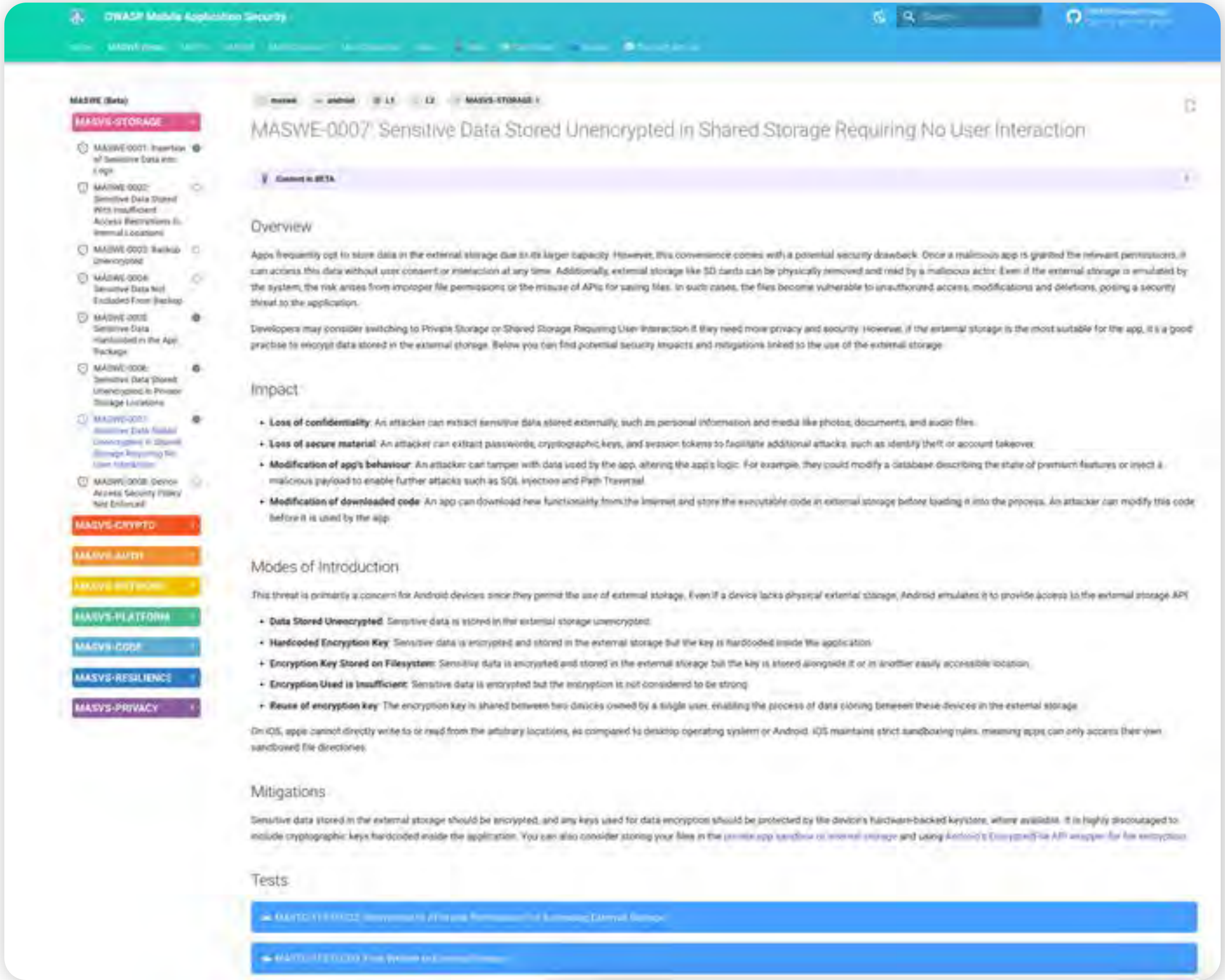
Platform

MASVS IDs

Profiles

Status with links to Github Issues

MASWE Update



MASWE (Beta)

MASVS-STORAGE

- MASWE-0001: Insertion of Sensitive Data into Logs
- MASWE-0002: Sensitive Data Stored With Insufficient Access Restrictions in Internal Locations
- MASWE-0003: Backup Unencrypted
- MASWE-0004: Sensitive Data Not Excluded From Backup
- MASWE-0005: Sensitive Data Hardcoded in the App Package
- MASWE-0006: Sensitive Data Stored Unencrypted in Private Storage Locations
- MASWE-0007: Sensitive Data Stored Unencrypted in Shared Storage Requiring No User Interaction
- MASWE-0008: Device Access Security Policy Not Enforced

MASVS-CRYPTO

MASVS-AUTH

MASVS-NETWORK

MASVS-PLATFORM

MASVS-CODE

MASVS-RESILIENCE

MASVS-PRIVACY

masvs android L1 L2 MASVS-STORAGE-1

MASWE-0007: Sensitive Data Stored Unencrypted in Shared Storage Requiring No User Interaction

Content in BETA

Overview

Apps frequently opt to store data in the external storage due to its larger capacity. However, this convenience comes with a potential security drawback. Once a malicious app is granted the relevant permissions, it can access this data without user consent or interaction at any time. Additionally, external storage like SD cards can be physically removed and read by a malicious actor. Even if the external storage is emulated by the system, the risk arises from improper file permissions or the misuse of APIs for saving files. In such cases, the files become vulnerable to unauthorized access, modifications and deletions, posing a security threat to the application.

Developers may consider switching to Private Storage or Shared Storage Requiring User Interaction if they need more privacy and security. However, if the external storage is the most suitable for the app, it's a good practise to encrypt data stored in the external storage. Below you can find potential security impacts and mitigations linked to the use of the external storage.

Impact

- **Loss of confidentiality**: An attacker can extract sensitive data stored externally, such as personal information and media like photos, documents, and audio files.
- **Loss of secure material**: An attacker can extract passwords, cryptographic keys, and session tokens to facilitate additional attacks, such as identity theft or account takeover.
- **Modification of app's behaviour**: An attacker can tamper with data used by the app, altering the app's logic. For example, they could modify a database describing the state of premium features or inject a malicious payload to enable further attacks such as SQL injection and Path Traversal.
- **Modification of downloaded code**: An app can download new functionality from the Internet and store the executable code in external storage before loading it into the process. An attacker can modify this code before it is used by the app.

Modes of Introduction

This threat is primarily a concern for Android devices since they permit the use of external storage. Even if a device lacks physical external storage, Android emulates it to provide access to the external storage API.

- **Data Stored Unencrypted**: Sensitive data is stored in the external storage unencrypted.
- **Hardcoded Encryption Key**: Sensitive data is encrypted and stored in the external storage but the key is hardcoded inside the application.
- **Encryption Key Stored on Filesystem**: Sensitive data is encrypted and stored in the external storage but the key is stored alongside it or in another easily accessible location.
- **Encryption Used is Insufficient**: Sensitive data is encrypted but the encryption is not considered to be strong.
- **Reuse of encryption key**: The encryption key is shared between two devices owned by a single user, enabling the process of data cloning between these devices in the external storage.

On iOS, apps cannot directly write to or read from the arbitrary locations, as compared to desktop operating system or Android. iOS maintains strict sandboxing rules, meaning apps can only access their own sandboxed file directories.

Mitigations

Sensitive data stored in the external storage should be encrypted, and any keys used for data encryption should be protected by the device's hardware-backed keystore, where available. It is highly discouraged to include cryptographic keys hardcoded inside the application. You can also consider storing your files in the private app sandbox or internal storage and using Android's EncryptedFile API wrapper for file encryption.

Tests

MASTG-TEST-0202: [illegible] Permissions for Accessing External Storage

MASTG-TEST-0200: Files Written to External Storage

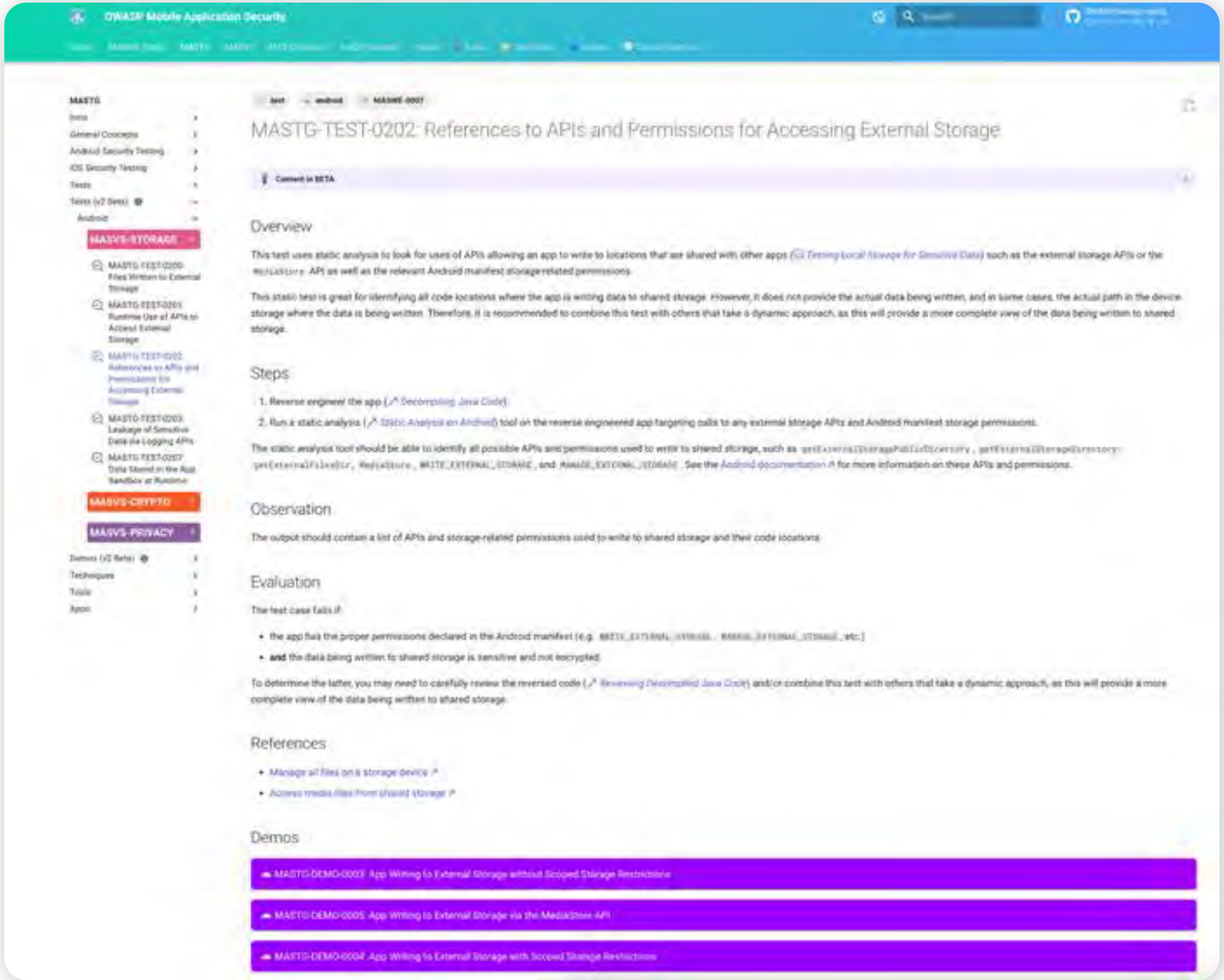
What's bad?

What can happen?

How can this happen?

How to fix it?

Links to tests


OWASP Mobile Application Security
MASTG
Intro
General Concepts
Android Security Testing
iOS Security Testing
Tests
Tests (v2 Beta)
Android
MASVS-STORAGE
MASTG-TEST-0200: Files Written to External Storage
MASTG-TEST-0201: Runtime Use of APIs to Access External Storage
MASTG-TEST-0202: References to APIs and Permissions for Accessing External Storage
MASTG-TEST-0203: Leakage of Sensitive Data via Logging APIs
MASTG-TEST-0207: Data Stored in the App Sandbox at Runtime
MASVS-CRYPTO
MASVS-PRIVACY
Demos (v2 Beta)
Techniques
Tools
Apps
test
android
MASWE-0007
Content in BETA

MASTG-TEST-0202: References to APIs and Permissions for Accessing External Storage

Overview

This test uses static analysis to look for uses of APIs allowing an app to write to locations that are shared with other apps (Testing Local Storage for Sensitive Data) such as the external storage APIs or the `MediaStore` API as well as the relevant Android manifest storage-related permissions.

This static test is great for identifying all code locations where the app is writing data to shared storage. However, it does *not* provide the actual data being written, and in some cases, the actual path in the device storage where the data is being written. Therefore, it is recommended to combine this test with others that take a dynamic approach, as this will provide a more complete view of the data being written to shared storage.

Steps

1. Reverse engineer the app (↗ Decompiling Java Code).
2. Run a static analysis (↗ Static Analysis on Android) tool on the reverse engineered app targeting calls to any external storage APIs and Android manifest storage permissions.

The static analysis tool should be able to identify all possible APIs and permissions used to write to shared storage, such as `getExternalStoragePublicDirectory`, `getExternalStorageDirectory`, `getExternalFilesDir`, `MediaStore`, `WRITE_EXTERNAL_STORAGE`, and `MANAGE_EXTERNAL_STORAGE`. See the Android documentation ↗ for more information on these APIs and permissions.

Observation

The output should contain a list of APIs and storage-related permissions used to write to shared storage and their code locations.

Evaluation

The test case fails if:

- the app has the proper permissions declared in the Android manifest (e.g. `WRITE_EXTERNAL_STORAGE`, `MANAGE_EXTERNAL_STORAGE`, etc.)
- **and** the data being written to shared storage is sensitive and not encrypted.

To determine the latter, you may need to carefully review the reversed code (↗ Reviewing Decompiled Java Code) and/or combine this test with others that take a dynamic approach, as this will provide a more complete view of the data being written to shared storage.

References

- Manage all files on a storage device ↗
- Access media files from shared storage ↗

Demos

- MASTG-DEMO-0003: App Writing to External Storage without Scoped Storage Restrictions
- MASTG-DEMO-0005: App Writing to External Storage via the MediaStore API
- MASTG-DEMO-0004: App Writing to External Storage with Scoped Storage Restrictions

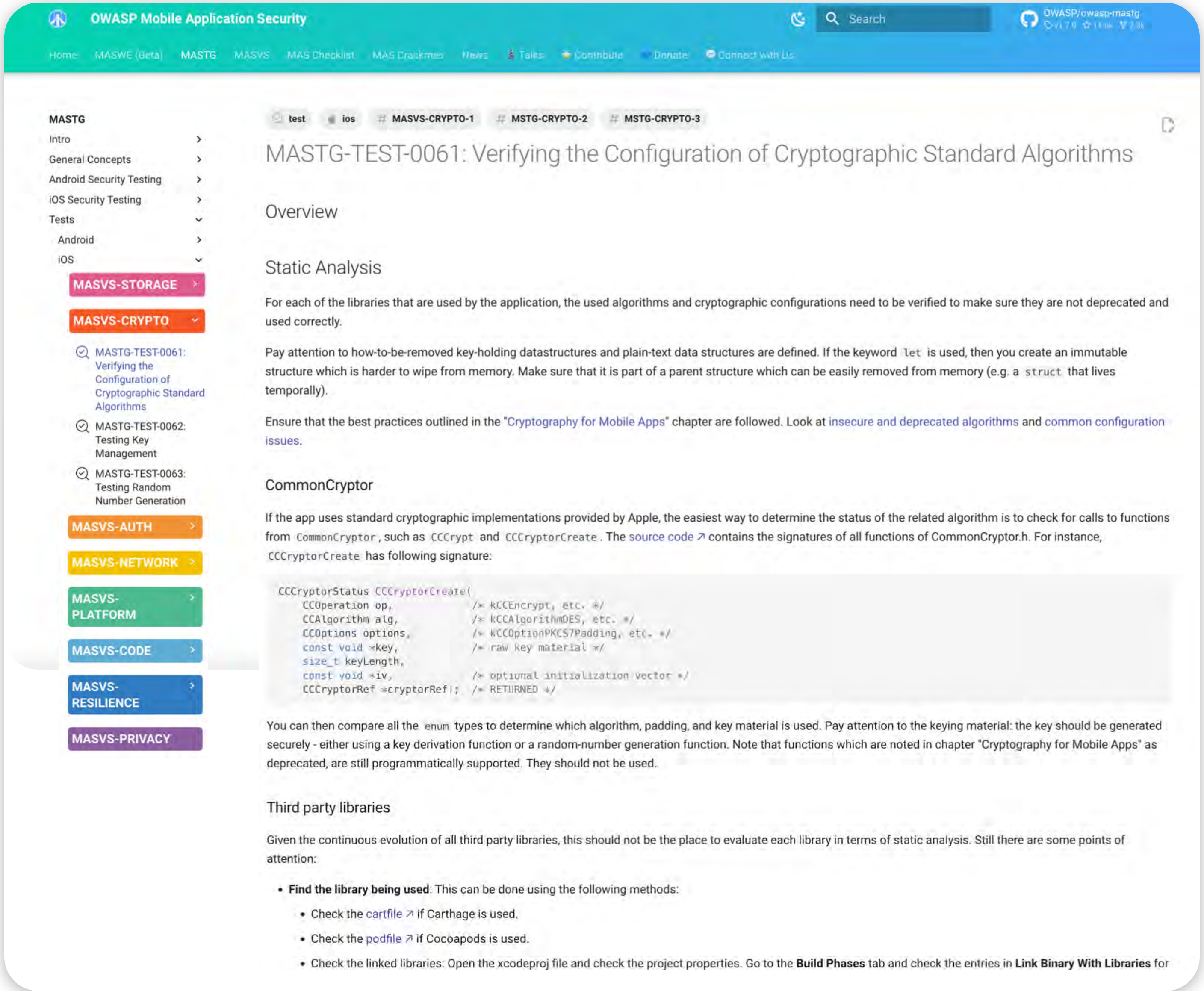
What are we testing & why?

Steps to test

Output of test

How to evaluate?

Links to demos



MASTG-TEST-0061: Verifying the Configuration of Cryptographic Standard Algorithms

Overview

Static Analysis

For each of the libraries that are used by the application, the used algorithms and cryptographic configurations need to be verified to make sure they are not deprecated and used correctly.

Pay attention to how-to-be-removed key-holding datastructures and plain-text data structures are defined. If the keyword `let` is used, then you create an immutable structure which is harder to wipe from memory. Make sure that it is part of a parent structure which can be easily removed from memory (e.g. a `struct` that lives temporally).

Ensure that the best practices outlined in the "Cryptography for Mobile Apps" chapter are followed. Look at insecure and deprecated algorithms and common configuration issues.

CommonCryptor

If the app uses standard cryptographic implementations provided by Apple, the easiest way to determine the status of the related algorithm is to check for calls to functions from `CommonCryptor`, such as `CCCrypt` and `CCCryptorCreate`. The source code ↗ contains the signatures of all functions of CommonCryptor.h. For instance, `CCCryptorCreate` has following signature:

```
CCCryptorStatus CCCryptorCreate(
    CCOperation op,             /* kCCEncrypt, etc. */
    CCAlgorithm alg,            /* kCCAlgorithmDES, etc. */
    CCOptions options,          /* kCCOptionPKCS7Padding, etc. */
    const void *key,            /* raw key material */
    size_t keyLength,
    const void *iv,             /* optional initialization vector */
    CCCryptorRef *cryptorRef);  /* RETURNED */
```

You can then compare all the `enum` types to determine which algorithm, padding, and key material is used. Pay attention to the keying material: the key should be generated securely - either using a key derivation function or a random-number generation function. Note that functions which are noted in chapter "Cryptography for Mobile Apps" as deprecated, are still programmatically supported. They should not be used.

Third party libraries

Given the continuous evolution of all third party libraries, this should not be the place to evaluate each library in terms of static analysis. Still there are some points of attention:

- **Find the library being used**: This can be done using the following methods:
  - Check the cartfile ↗ if Carthage is used.
  - Check the podfile ↗ if Cocoapods is used.
  - Check the linked libraries: Open the xcodeproj file and check the project properties. Go to the **Build Phases** tab and check the entries in **Link Binary With Libraries** for

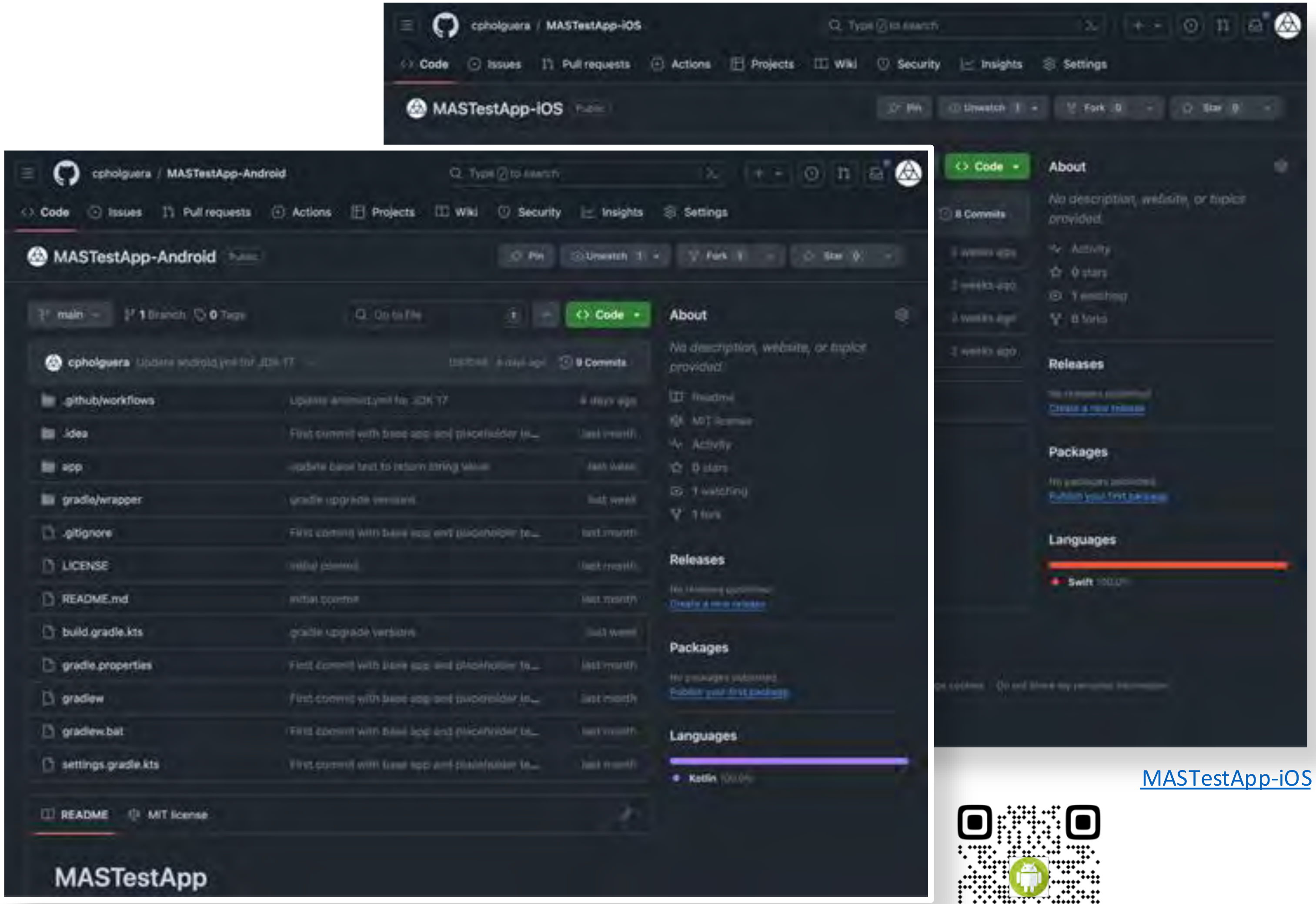
What to test?

How to test?

Code Samples

First-party and third-party code

New MASTestApps



cpholguera / MASTestApp-iOS

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

MASTestApp-iOS Public

Pin · Unwatch 1 · Fork 0 · Star 0

8 Commits

2 weeks ago

2 weeks ago

2 weeks ago

2 weeks ago

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Swift 100.0%

cpholguera / MASTestApp-Android

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

MASTestApp-Android Public

Pin · Unwatch 1 · Fork 1 · Star 0

main · 1 Branch · 0 Tags · Go to file · Code

cpholguera Update android.yml for JDK 17 · 4 days ago · 9 Commits

.github/workflows	Update android.yml for JDK 17	4 days ago
.idea	First commit with base app and placeholder te...	last month
app	update base test to return string value	last week
gradle/wrapper	gradle upgrade versions	last week
.gitignore	First commit with base app and placeholder te...	last month
LICENSE	initial commit	last month
README.md	initial commit	last month
build.gradle.kts	gradle upgrade versions	last week
gradle.properties	First commit with base app and placeholder te...	last month
gradlew	First commit with base app and placeholder te...	last month
gradlew.bat	First commit with base app and placeholder te...	last month
settings.gradle.kts	First commit with base app and placeholder te...	last month

README · MIT license

MASTestApp

About

No description, website, or topics provided.

Readme

MIT license

Activity

0 stars

1 watching

1 fork

Releases

No releases published

Create a new release

Packages

No packages published

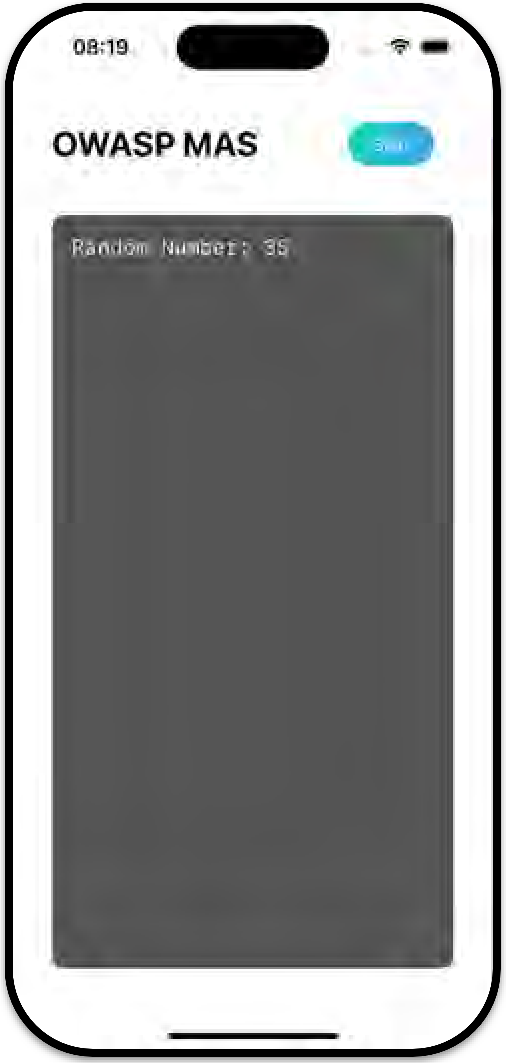
Publish your first package

Languages

Kotlin 100.0%

MASTestApp-iOS

MASTestApp-Android

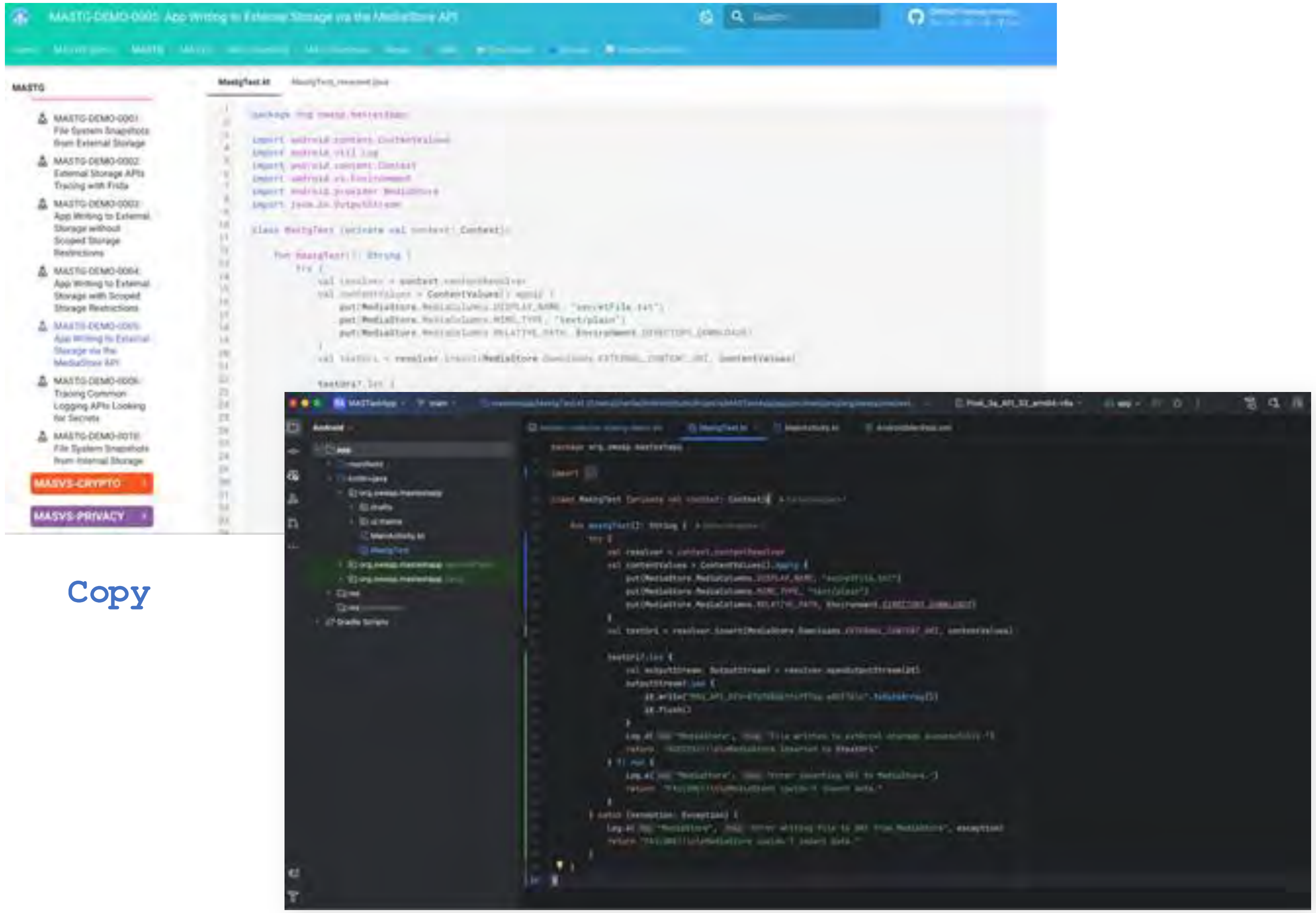


08:19

OWASP MAS

```
Random Number: 35
```

New MASTestApps



MASTG-DEMO-0005: App Writing to External Storage via the MediaStore API

MASTG

- MASTG-DEMO-0001: File System Snapshots from External Storage
- MASTG-DEMO-0002: External Storage APIs Tracing with Frida
- MASTG-DEMO-0003: App Writing to External Storage without Scoped Storage Restrictions
- MASTG-DEMO-0004: App Writing to External Storage with Scoped Storage Restrictions
- MASTG-DEMO-0005: App Writing to External Storage via the MediaStore API
- MASTG-DEMO-0006: Tracing Common Logging APIs Looking for Secrets
- MASTG-DEMO-0010: File System Snapshots from Internal Storage

MASVS-CRYPTO

MASVS-PRIVACY

MastgTest.kt | MastgTest_reversed.java

```
package org.owasp.mastestapp

import android.content.ContentValues
import android.util.Log
import android.content.Context
import android.os.Environment
import android.provider.MediaStore
import java.io.OutputStream

class MastgTest (private val context: Context){

    fun mastgTest(): String {
        try {
            val resolver = context.contentResolver
            val contentValues = ContentValues().apply {
                put(MediaStore.MediaColumns.DISPLAY_NAME, "secretFile.txt")
                put(MediaStore.MediaColumns.MIME_TYPE, "text/plain")
                put(MediaStore.MediaColumns.RELATIVE_PATH, Environment.DIRECTORY_DOWNLOADS)
            }
            val textUri = resolver.insert(MediaStore.Downloads.EXTERNAL_CONTENT_URI, contentValues)

            textUri?.let {
```

```
package org.owasp.mastestapp

import ...

class MastgTest (private val context: Context){

    fun mastgTest(): String {
        try {
            val resolver = context.contentResolver
            val contentValues = ContentValues().apply {
                put(MediaStore.MediaColumns.DISPLAY_NAME, "secretFile.txt")
                put(MediaStore.MediaColumns.MIME_TYPE, "text/plain")
                put(MediaStore.MediaColumns.RELATIVE_PATH, Environment.DIRECTORY_DOWNLOADS)
            }
            val textUri = resolver.insert(MediaStore.Downloads.EXTERNAL_CONTENT_URI, contentValues)

            textUri?.let {
                val outputStream: OutputStream? = resolver.openOutputStream(it)
                outputStream?.use {
                    it.write("[illegible]".toByteArray())
                    it.flush()
                }
                Log.d( tag: "MediaStore",  msg: "File written to external storage successfully.")
                return "SUCCESS!!\n\nMediaStore inserted to $textUri"
            } ?: run {
                Log.e( tag: "MediaStore",  msg: "Error inserting URI to MediaStore.")
                return "FAILURE!!\n\nMediaStore couldn't insert data."
            }
        } catch (exception: Exception) {
            Log.e( tag: "MediaStore",  msg: "Error writing file to URI from MediaStore", exception)
            return "FAILURE!!\n\nMediaStore couldn't insert data."
        }
    }
}
```

Copy

Paste

Run

Random Number: 35

Who plans to go investigate the OWASP MAS Project?

Great Mobile App Security and Privacy Is a Team Effort

A Team Effort Focused on Protecting the Business and Consumers with a Great User Experience is as Easy as

1

Use the Consumer Report to make security and privacy a priority with your Business Leaders, Product and Engineering Teams

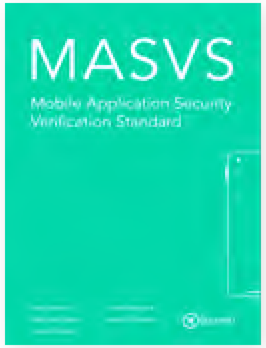
appdome

2024

Singapore Consumer Expectations of Mobile App Security

2

Leverage the industry standard OWASP MAS as the foundation of your mobile AppSec program



MASVS

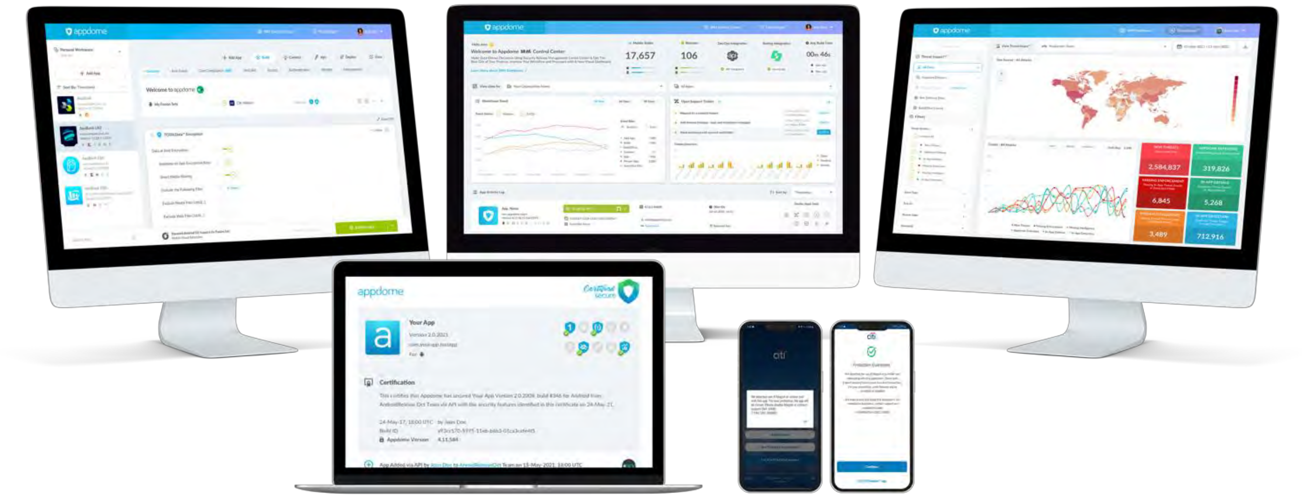
Mobile Application Security Verification Standard

3

Use automation so that it is easy for Security and Development to deliver high-quality releases fast

Appdome Unified Mobile App Defense Automation Platform

Get OWASP MASVS Compliant in <2 mins

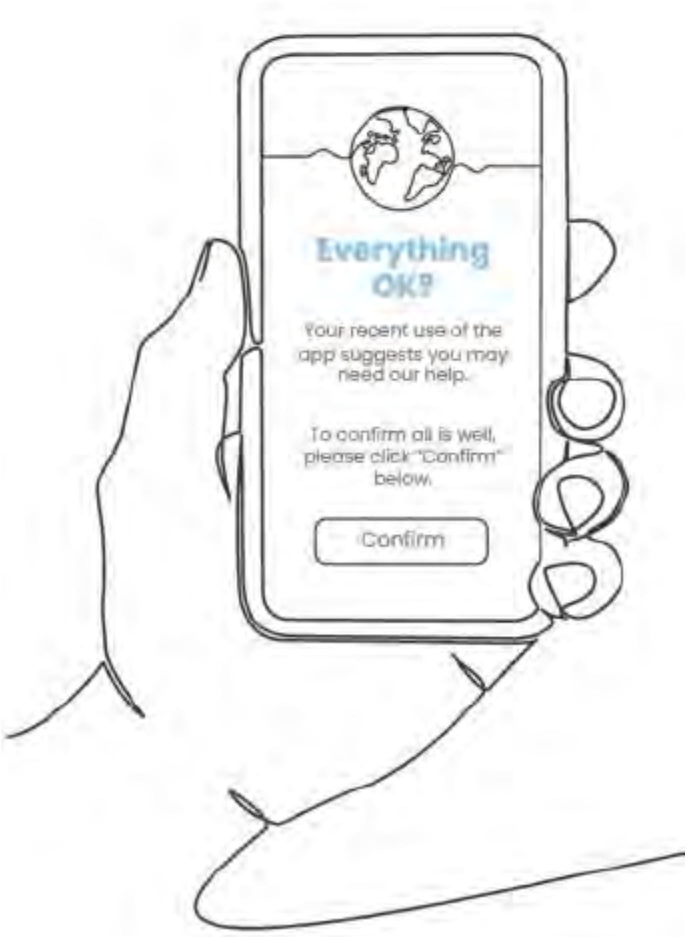


340+ Protections Against

Mobile App Security Threats, Fraud, Malware, Social Engineering Scams, Geo-Fraud, Mobile Bot Defense and more

Automated Certified Secure™ + Great User Experience

Block Social Engineering Scams



Everything OK?

Your recent use of the app suggests you may need our help.

To confirm all is well, please click "Confirm" below.

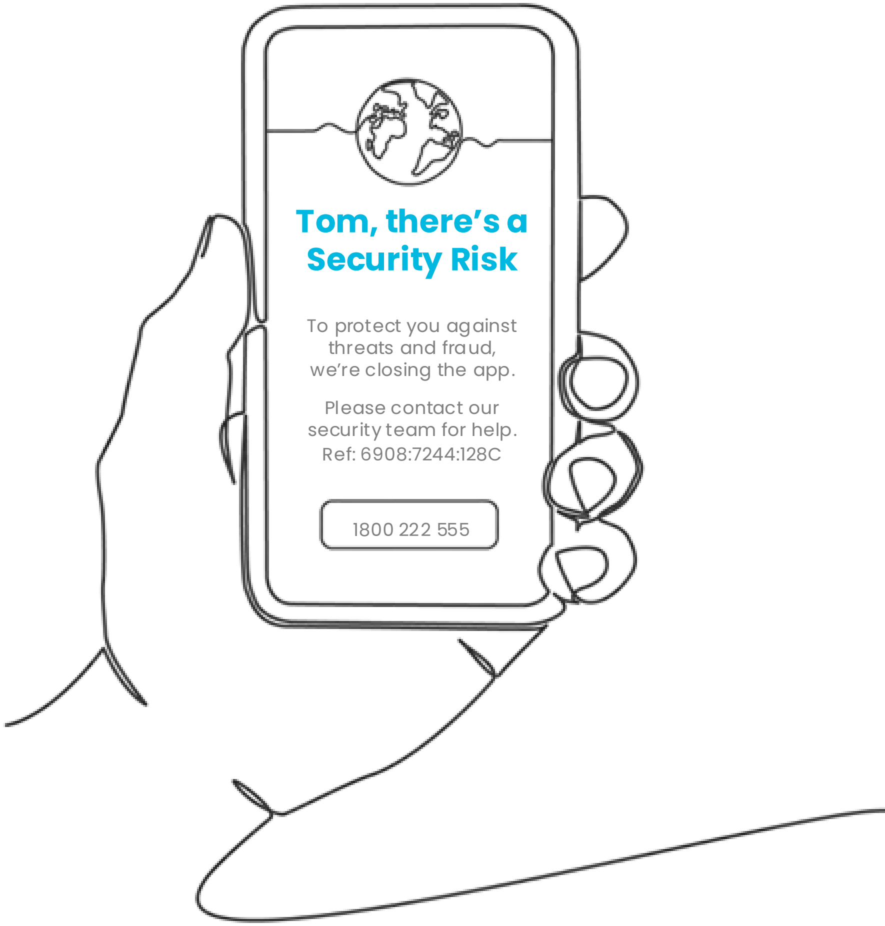
Stop Geo-Location Fraud

Are you travelling?

We'd love for you to keep using our app.

To Confirm it's you, please click "Confirm" below.

Protect Against Malware Attacks



Tom, there's a Security Risk

To protect you against threats and fraud, we're closing the app.

Please contact our security team for help. Ref: 6908:7244:128C

Thank you!

Brian Reed | SVP Mobile Defense

brianr@Appdome.com

Connect with me on Linkedin in and DM me for this deck

Scan to Download the Reports

appdome | OWASP

2024
Singapore Consumer Expectations of Mobile App Security

2024
Singapore Consumer Report

appdome | OWASP

2024
Philippines Consumer Expectations of Mobile App Security

2024
Philippines Consumer Report

appdome | OWASP

2024
Australia Consumer Expectations of Mobile App Security

2024
Australia Consumer Report

appdome | OWASP

2024
Global Consumer Expectations of Mobile App Security

2024
Global Consumer Report