

**ITMC411**

# **Security in mobile computing**

---

## **LECTURE 7**

### **Mobile Application Security Verification Standard**



**MASVS**

# Mobile Application Security Verification Standard

The **MASVS** can be used to **establish** a level of confidence in the security of mobile apps.

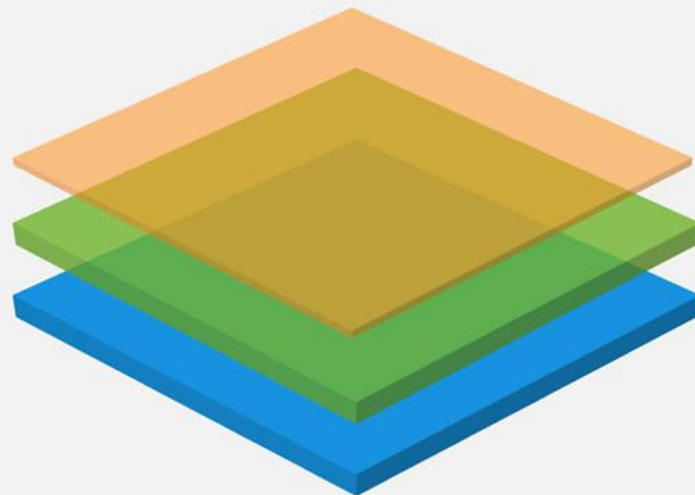
The **requirements** were **developed** with the following objectives in mind:

- **Use as a metric** - To provide a security standard against which existing mobile apps can be compared by developers and application owners;
- **Use as guidance** - To provide guidance during all phases of mobile app development and testing;
- **Use during procurement** - To provide a baseline for mobile app security verification.

# Mobile AppSec Model

The MASVS defines two security verification levels (**MASVS-L1** and **MASVS-L2**) and set of reverse engineering resiliency requirements (**MASVS-R**)

1. **MASVS-L1** contains generic security requirements that are recommended for all mobile apps.
2. **MASVSL2** should be applied to apps handling highly sensitive data.
3. **MASVS-R** covers additional protective controls that can be applied if preventing client-side threats is a design goal.



R – Resiliency Against Reverse Engineering and Tampering

L2 – Defense-in-Depth

L1 – Standard Security

# Verification Levels in Detail

## MASVS-L1: Standard Security

A mobile app that achieves **MASVS-L1** adheres to mobile application security best practices. It fulfills basic requirements in terms of :

- code quality,
- handling of sensitive data,
- interaction with the mobile environment.
- A testing process must be in place to verify the security controls.

**This level is appropriate for all mobile applications.**

# Verification Levels in Detail

## MASVS-L2: Defense-in-Depth

**MASVS-L2** introduces **advanced security controls** that go beyond the standard requirements. To fulfill **MASVS-L2**,

- a threat model must exist,
- security must be an integral part of the app's architecture and design. Based on the threat model,
- the right **MASVS-L2** controls should have been selected and implemented successfully.

**This level is appropriate for apps that handle highly sensitive data, such as mobile banking apps.**

# Verification Levels in Detail

## MASVS-R: Resiliency Against Reverse Engineering and Tampering

The app has state-of-the-art security, and is also resilient against specific, clearly defined client-side attacks, such as

- **tampering**,
- **modding**
- **reverse engineering** to extract sensitive code or data.

**MASVS-R** is applicable to apps that handle **highly sensitive data** and may serve as a means of protecting intellectual property or tamper-proofing an app.

# Using the OWASP Mobile Security Testing Guide (MSTG)

The **OWASP MSTG** is a manual for testing the security of mobile apps.

- describes the technical processes for verifying the requirements listed in the **MASVS**.
- includes a list of test **cases**, each of which map to a requirement in the **MASVS**.
- While the **MASVS** requirements are **high-level** and generic, the **MSTG** provides **in-depth** recommendations and testing procedures on a per-mobile-OS basis.

# Document Structure

The requirements have been grouped into eight categories (V1 to V8) based on technical objective / scope.

The following nomenclature is used throughout the **MASVS** and **MSTG**:

- **V1: Architecture, Design and Threat Modeling Requirements**
- **V2: Data Storage and Privacy Requirements**
- **V3: Cryptography Requirements**
- **V4: Authentication and Session Management Requirements**
- **V5: Network Communication Requirements**
- **V6: Platform Interaction Requirements**
- **V7: Code Quality and Build Setting Requirements**
- **V8: Resilience Requirements**





# Document Structure

The requirements have been grouped into eight categories (V1 to V8) based on technical objective / scope.

The following nomenclature is used throughout the **MASVS** and **MSTG**:

- **MASVS-STORAGE**: Secure storage of sensitive data on a device (data-at-rest).
- **MASVS-CRYPTO**: Cryptographic functionality used to protect sensitive data.
- **MASVS-AUTH**: Authentication and authorization mechanisms used by the mobile app.
- **MASVS-NETWORK**: Secure network communication between the mobile app and remote endpoints (data-in-transit).
- **MASVS-PLATFORM**: Secure interaction with the underlying mobile platform and other installed apps.
- **MASVS-CODE**: Security best practices for data processing and keeping the app up-to-date.
- **MASVS-RESILIENCE**: Resilience to reverse engineering and tampering
- **MASVS-PRIVACY**: Privacy controls to protect user privacy.



# Mobile\_App\_Security\_Checklist

## Checklist



V2	Data Storage and Privacy	
2.2	Verify that no sensitive data is written to application logs.	✓

Test Cases



Requirements



**OWASP Mobile Security Testing Guide (MSTG)**

**OWASP Mobile Application Security Verification Standard (MASVS)**

OMTG-DATAST-002: Test for Sensitive Data in Logs

### Overview

There are many legit reasons to create log files on a mobile device, for example to keep track of crashes or errors that are stored locally when being offline and being sent to the application developer/company once online again or for usage statistics. However, logging sensitive data such as credit card number and session IDs might expose the data to attackers or malicious applications. Log files can be created in various ways on each of the different operating systems. The following list shows the mechanisms that are available on Android:

- Log Class, `Log[a-Z]`
- Logger Class
- StrictMode
- `System.out/System.err.print`

Classification of sensitive information can vary between different industries, countries and their laws and regulations. Therefore laws and regulations need to be known that are applicable to it and to be aware of what sensitive information actually is in the context of the App.

#	
2.2	No sensitive data is written to application logs.