

# Insiders Guide to Mobile AppSec with OWASP MASVS

OWASP Meetup

Brian Reed, Chief Mobility Officer

[br@nowsecure.com](mailto:br@nowsecure.com)

[@reed\\_on\\_the\\_run](https://twitter.com/reed_on_the_run)





12 years in Mobile Security

OWASP Sponsor & Contributor

Mobile AppSec Testing Tools, Training, Pen Testing

Creators of Frida and Radare

FRIDA



Microsoft Google Uber



iRobot TESLA



Medtronic



AT&T



**Brian Reed**  
Chief Mobility Officer

[br@nowsecure.com](mailto:br@nowsecure.com)

@reed\_on\_the\_run



**15+ Years in Mobile**

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

**NowSecure, Good Technology, BlackBerry,  
ZeroFOX, BoxTone, and MicroFocus**



NowSecure

**OWASP Mobile Project Financial Sponsor & Contributor**

*NowSecure Security Researcher Carlos Holguera*

*([@grepharder](#)) is co-project lead for OWASP Mobile Project*

**OWASP MSTG Advocate**

*recognition for years of contributions*

**OWASP CycloneDX SBOM Contributor**

*NowSecure Founder Andrew Hoog on the CycloneDX leadership board*





**NowSecure IoXT Authorized Lab**  
*Certify Mobile-Connected IoT devices*



**NowSecure ADA Authorized Lab**  
*Independent Security Reviews for Google Play Data Safety*



# Open Source Community

## FRIDA

[OVERVIEW](#) [DOCS](#) [NEWS](#) [CODE](#) [CONTACT](#)

Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers.

### Scriptable

Inject your own scripts into black box processes. Hook any function, spy on crypto APIs or trace private application code, no source code needed. Edit, hit save, and instantly see the results. All without compilation steps or program restarts.

### Portable

Works on Windows, macOS, GNU/Linux, iOS, Android, and QNX. Install the Node.js bindings from [npm](#), grab a Python package from [PyPI](#), or use Frida through its [Swift bindings](#), [.NET bindings](#), [Qt/Qml bindings](#), or [C API](#). We also have a scalable [footprint](#).

### Free

Frida is and will always be [free software](#) (free as in freedom). We want to empower the next generation of developer tools, and help other free software developers achieve interoperability through reverse engineering.

### Battle-tested

We are proud that [NowSecure](#) is using Frida to do fast, deep analysis of mobile apps [at scale](#). Frida has a comprehensive test-suite and has gone through years of rigorous testing across a broad range of use-cases.

GitHub

radare2

```
import os
import r2pipe

r2 = r2pipe.open('libv(1)')
r2.cmd('p')
print (r2.cmd('va 10'))
r2.quit()
```

r2Pipe

lailo

Download [Github](#) / [Bins](#)  
[YouTube](#) / [Community](#) / [Book](#)

[Project Repository](#)  
[Example scripts](#)

[Official Releases](#)  
[Github Source](#)

## Libre and Portable Reverse Engineering Framework

[Conference](#) [Merchandise](#) [Old Website](#)

# Peloton Responsible Disclosure from NowSecure

NowSecure researcher Austin Emmitt found and disclosed 4 vulnerabilities to Peloton mobile, web & APIs that have now been fixed:

1. Peloton user exposure to account takeover
2. Peloton user exposure to phishing attack
3. Remote access to Peloton users' private personal info
4. Ability to remotely change device ID and serial number

There is NO evidence that any customers were breached

Read the two Blogs:

<https://www.nowsecure.com/blog/2021/12/08/its-not-about-the-bike-how-nowsecure-helped-peloton-secure-its-mobile-apps-apis/>

<https://www.nowsecure.com/blog/2022/02/09/a-zero-click-rce-exploit-for-the-peloton-bike-and-also-every-other-unpatched-android-device/>



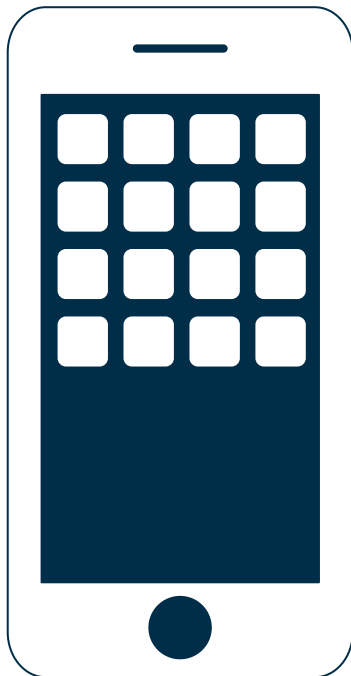
# Mobile Powers the World, But Mobile Risk is Pervasive

**69%**

of all digital traffic &  
time spent is on mobile  
vs. web

**200bn**

Mobile App Downloads in  
2021



**85%**

of Mobile Apps  
have security risks  
(Fail OWASP Mobile Top 10)

**70%**

of Mobile Apps leak  
personal data to  
violate GDPR/CCPA

Sources: AppAnnie, March, 2020; Comscore, January 2020  
Gartner, *Avoid Mobile App Security Pitfalls*, Zumerle, 27Jul2020  
NowSecure Privacy Benchmark, 2019; NowSecure Security Benchmark 2020



# What Mobile Apps Do You Use?



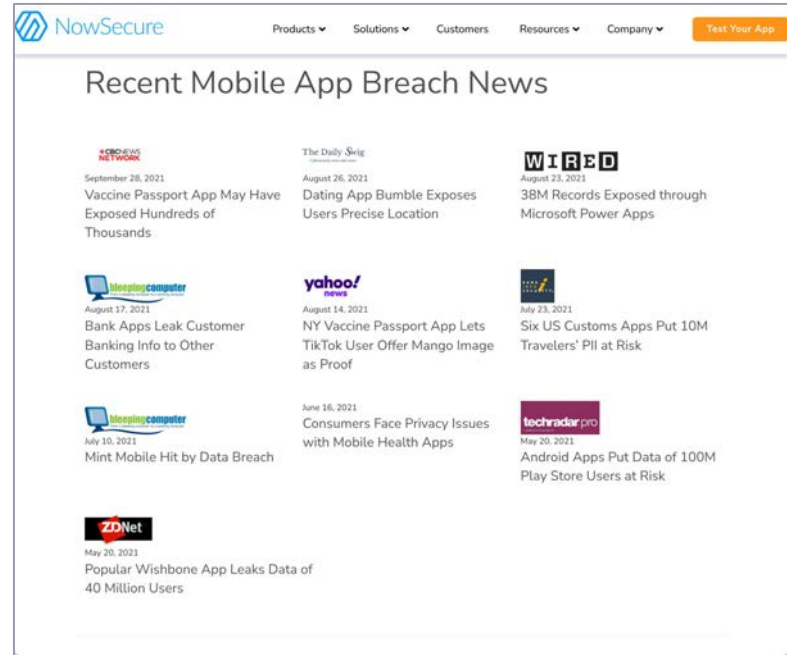
<https://www.nowsecure.com/mobile-app-security-news/>



# Benchmark Trackers to Learn More



<https://mobilerisktracker.nowsecure.com>

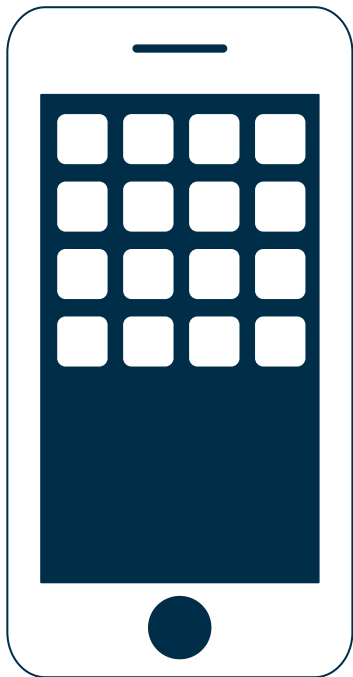


<https://bit.ly/ns-breachtracker>

# Inside Mobile AppSec

modern mobile  
**sensitive**  
padlock **hacker** personal computer user secure touch protect service virus  
password code protection businessman web people network  
intelligence support multimedia safety storage  
communication **data** computing **technology** private  
**security** crime screen **software** digital media  
**privacy** access attack **information** electronic encryption monitor  
help **internet** confidential technical safe finger system city  
business company display control  
street device cyber transfer

# Unique Characteristics of Mobile AppDev & AppSec



## WEB VS MOBILE

98% of code behind perimeter with broad layered protection

Substantial code "in the wild", running on untrusted device & easily reversible

- 2 Mobile OS with varying security capabilities
- 4 Dev Languages, Dozens of Frameworks, Thousands of libraries
- Continuous updates of Mobile OS and Dev tools
- Effective testing requires physical devices, not emulators
- Dynamic & APIsec testing are challenging, but can be automated

The OWASP MASVS is here to help!

## OWASP Top 10 Industry Standards



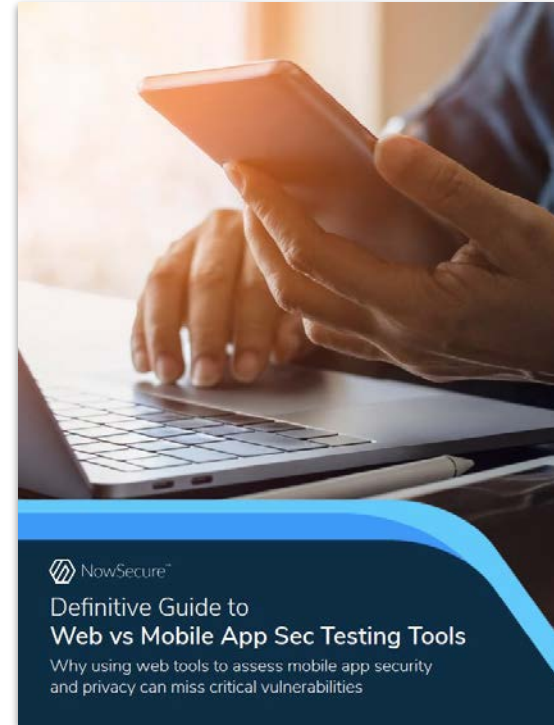
### Mobile

1. Improper Platform Usage
2. Insecure Data Storage
3. Insecure Communication
4. Insecure Authentication
5. Insufficient Cryptography
6. Insecure Authorization
7. Client Code Quality
8. Code Tampering
9. Reverse Engineering
10. Extraneous Functionality



### Web

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable & Outdated Components
7. Identification & Authentication Failures
8. Software & Data Integrity Failures
9. Security Logging & Monitoring Failures
10. Server-Side Request Forgery

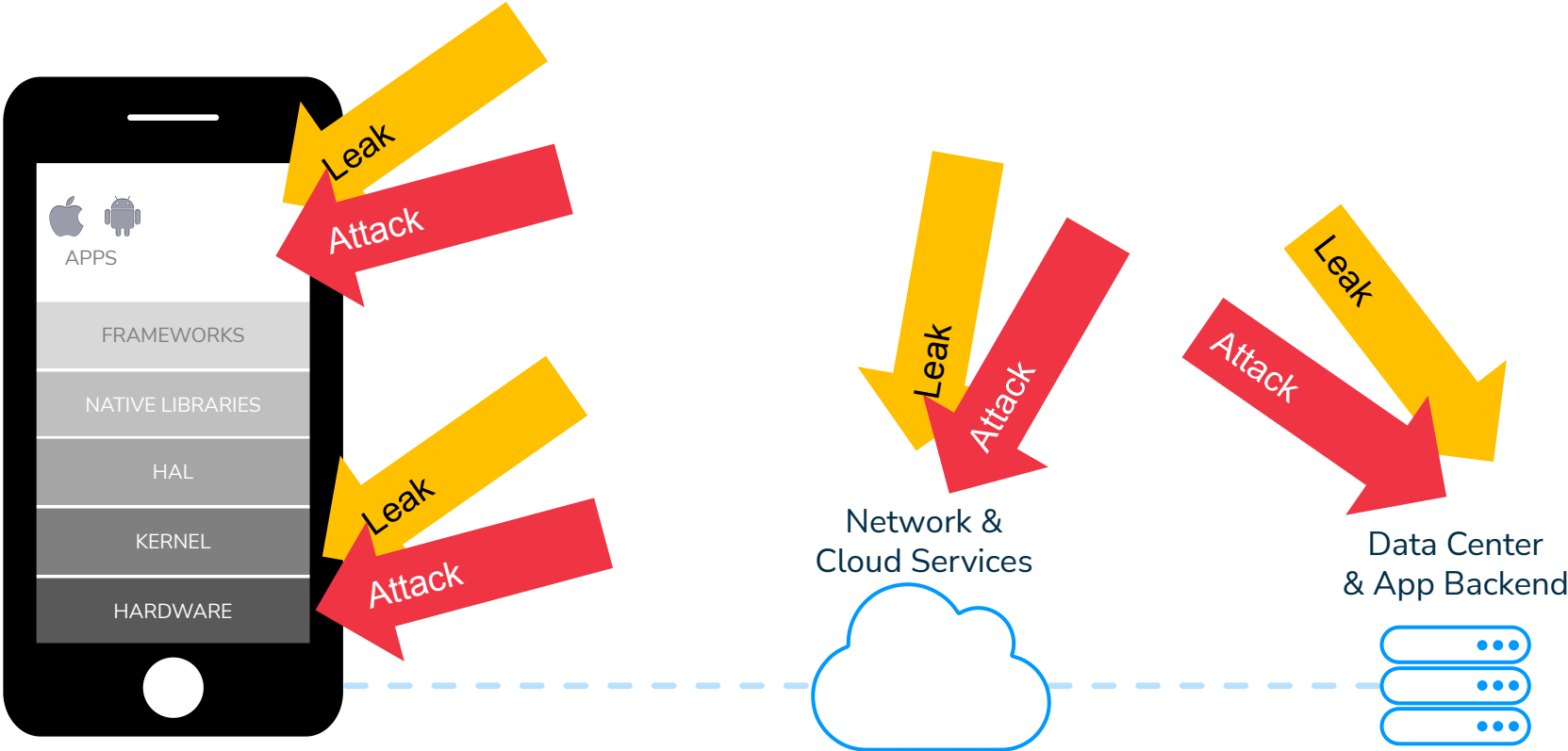


### Definitive Guide to Web vs Mobile App Sec Testing Tools

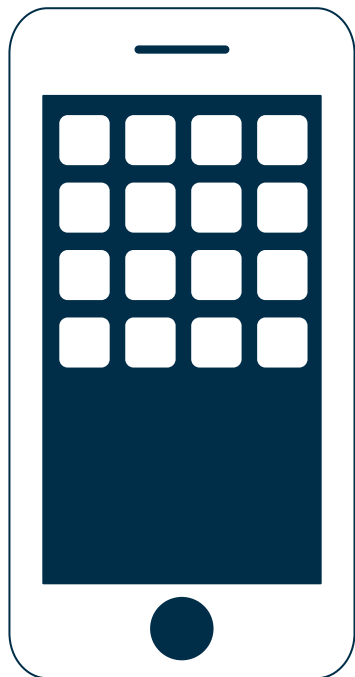
Why using web tools to assess mobile app security and privacy can miss critical vulnerabilities

<https://discover.nowsecure.com/nowsecure-ms/web-vs-mobile-app-security-testing-tools>

# Mobile Attack Surface



# What's Inside the Mobile Attack Surface?



## App Code

- App signing key unprotected
- Buffer overflow
- App Debuggable
- Configuration manipulation
- Missing User-input validation
- Insecure 3rd party libs
- Tampering/repacking possible
- No rooting/jailbreak detection
- No Code Obfuscation
- ...

## App Architecture

- Lack of Threat Modeling
- Insecure SDLC
- Bad Security Architecture
- Lack of Sensitive Data overview
- ...

## Data in Use

- Dynamic runtime injection
- Insecure URL schemes
- UI Data leaks
- Clipboard data leaks
- Unnecessary permissions
- ...

## API Backends

- Unauthenticated APIs
- Unprotected APIs
- Excessive API Data
- API SQL Injection
- Remote code execution
- Privilege Escalation
- Denial of Service
- ...

## Data at Rest

- Sensitive Data caching
- Lack of keychain usage
- Sensitive Data in log files
- Sensitive Data in memory
- Sensitive Data in World Writable/Readable Files
- ...

- Passwords & data accessible
- No/Weak encryption
- TEE/Secure Enclave Processor
- Side channel leak
- Sensitive Data in unencrypted databases
- ...

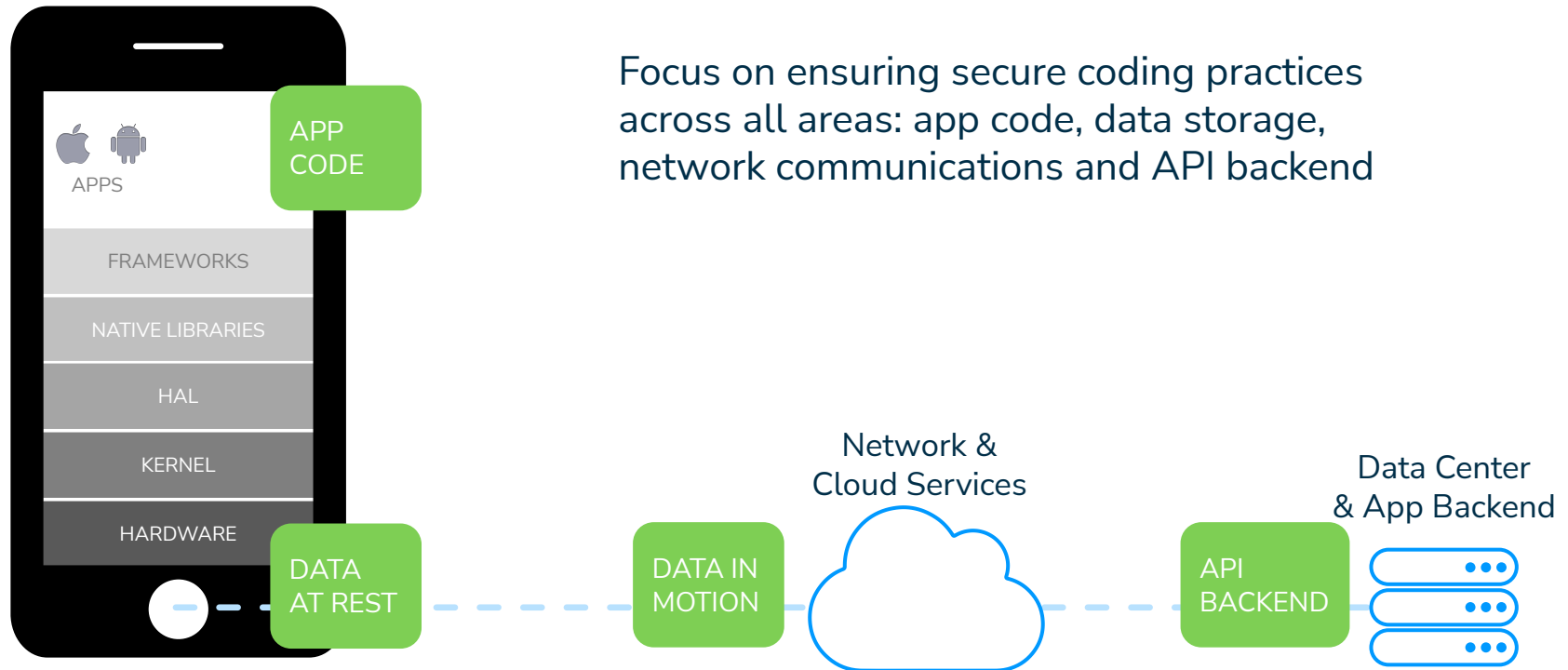
## Data in Motion

- Vulnerable to MITM attacks
- Vulnerable to session hijacking
- Improper TLS validation
- Weak App transport security
- Use of insecure protocols
- Insecure Cookies
- ...

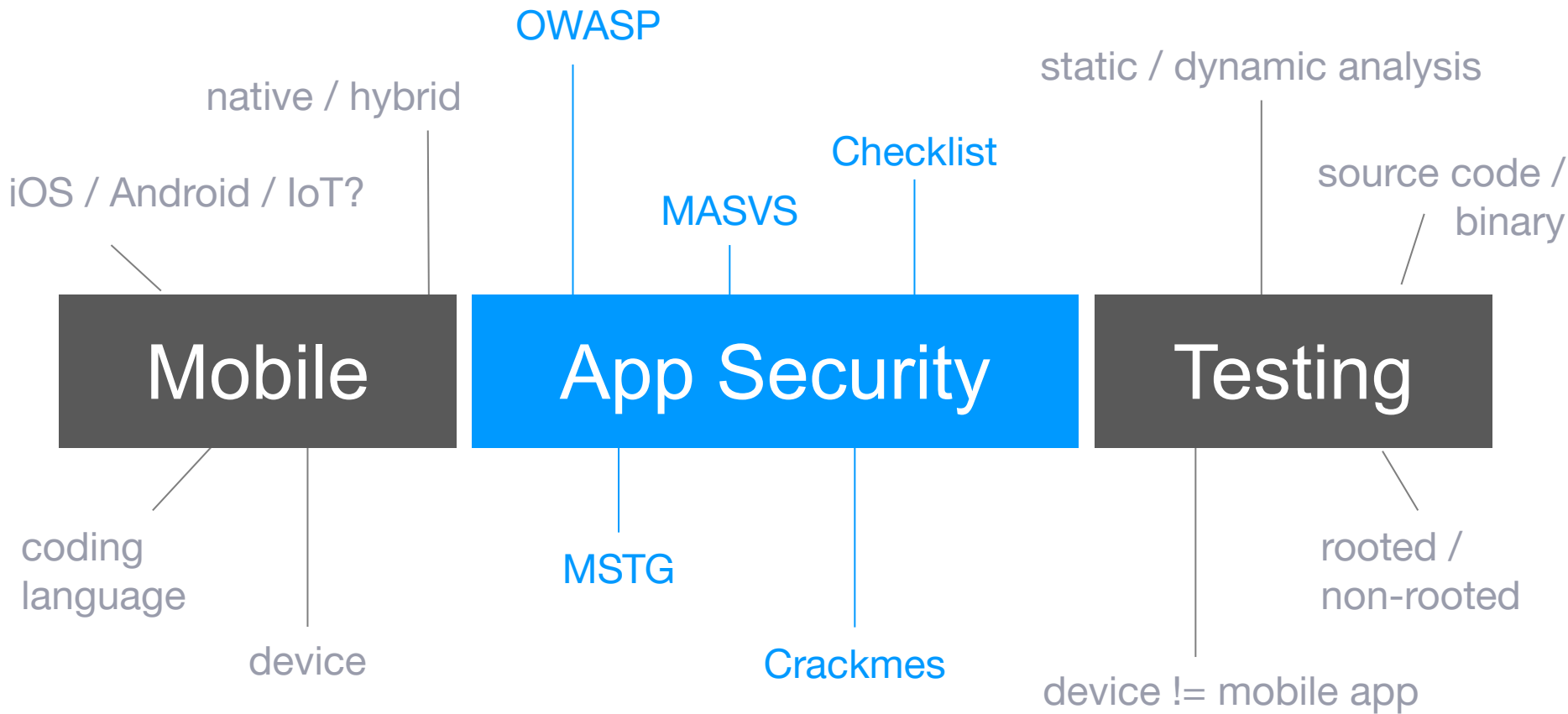
- Unauthenticated APIs
- Excessive API Data
- API SQL Injection
- Remote code execution
- Privilege Escalation
- Denial of Service
- ...



# Reduce the Attack Surface to Protect Sensitive Data

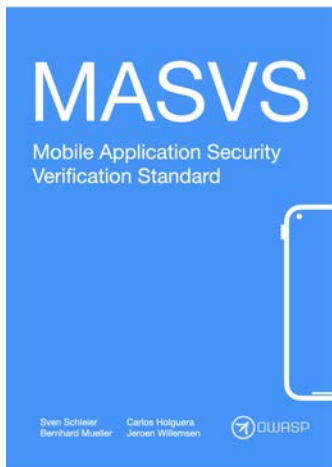


# Mobile App Security Testing





# OWASP Mobile Security Project Resources



## [Mobile App Security Verification Standard](#)

*Establish security baseline for mobile apps*

Latest Release: 2022



## [Mobile Security Testing Guide](#)

*Cookbook for mobile app security testing*

Latest Release: 2022

ID	ID	Description	L1	L2	R	Android	iOS	Status
1.1	MSTG-ARCH-1	All app components are identified and known to be needed.				Open		Pass
1.2	MSTG-ARCH-2	Security controls are never enforced only on the client side, but on the respective remote endpoints.				Pass	Pass	Pass
1.3	MSTG-ARCH-3	A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture.				Pass	Pass	Pass
1.4	MSTG-ARCH-4	Data considered sensitive in the context of the mobile app is clearly identified.				Pass	Pass	Fail
1.5	MSTG-ARCH-5	All app components are defined in terms of the business functions and/or security functions they provide.				Pass	Pass	Fail
1.6	MSTG-ARCH-6	A threat model for the mobile app and the associated remote services has been produced that identify potential threats and countermeasures.				Pass	Pass	Fail
1.7	MSTG-ARCH-7	All security controls have a canonical implementation.				Open		Pass

## [Mobile Security Testing Checklist](#)

*Checklist for mobile app security testing linking the MASVS to the MSTG*

Latest Release: 2022

# MASVS Mobile AppSec Model

## MASVS L1

*Standard Security*

- The minimum
- No compliance or regulatory needs
- Simple apps

*Example: Healthcare WebMD App*

## MASVS L1 + R

*Standard Security + High RE Resilience*

- Prioritize IP protection
- Prevent malicious modification or tampering

*Example: Medical Formulary App*

## MASVS L2

*Defense-in-Depth*

- Regulated industry data
- Compliance consideration
- Apps that perform simple tasks, but handled highly sensitive data.

*Example: Healthcare Weight Monitoring App*

## MASVS L2 + R

*Defense-in-Depth + High RE Resilience*

- Apps that perform complex activities between users and handle high sensitive data
- Compliance and IP protection are key
- Preventing Malware based attacks is in your threat model

*Example: Healthcare Drug Delivery App*

# Inside the MASVS Levels

L1 expects standard security best practices

## MASVS L1

*Standard Security*

- The minimum
- No compliance or regulatory needs
- Simple apps

*Example: Healthcare WebMD App*

## MASVS L1 + R

*Standard Security + High RE Resilience*

- Prioritize IP protection
- Prevent malicious modification or tampering

*Example: Medical Formulary App*

L2 expects defense-in-depth

- Hardened against "Lost device" scenario
- Certificate Pinning
- Multi-factor authentication
- Corp. policy for Architecture and Risk controls

## MASVS L2

*Defense-in-Depth*

- Regulated industry data
- Compliance consideration
- Apps that perform simple tasks, but handled highly sensitive data.

*Example: Healthcare Weight Monitoring App*

## MASVS L2 + R

*Defense-in-Depth + High RE Resilience*

- High sensitive operations & data handling
- Compliance and IP protection are key
- Preventing Malware based attacks is in your threat model

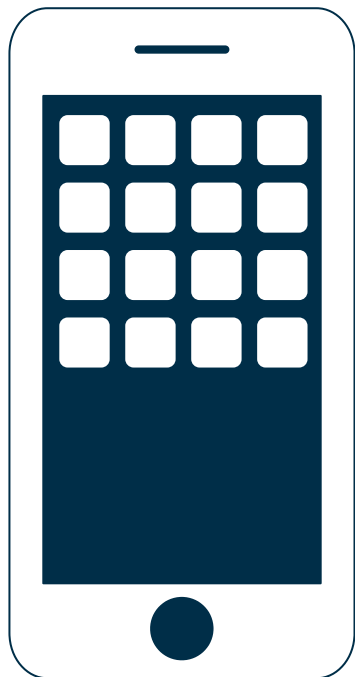
*Example: Healthcare Drug Delivery App*

R expects hardening

- Device Binding
- Obfuscation
- Anti-Tamper
- Not meant to compensate for poor security



# OWASP MASVS Addresses the Mobile Attack Surface



## MASVS-CODE

### MASVS-RESILIENCY

#### App Code

- App signing key unprotected
- Buffer overflow
- App Debuggable
- Configuration manipulation
- Missing User-input validation
- Insecure 3rd party libs
- Tampering/repacking possible
- No rooting/jailbreak detection
- No Code Obfuscation
- ...

#### Data at Rest

- Sensitive Data caching
- Lack of keychain usage
- Sensitive Data in log files
- Sensitive Data in memory
- Sensitive Data in World Writable/Readable Files
- ...

### MASVS-ARCH

#### App Architecture

- Lack of Threat Modeling
- Insecure SDLC
- Bad Security Architecture
- Lack of Sensitive Data overview
- ...

### MASVS-CRYPTO

#### MASVS-STORAGE

- Passwords & data accessible
- No/Weak encryption
- TEE/Secure Enclave Processor
- Side channel leak
- Sensitive Data in unencrypted databases
- ...

### MASVS-PLATFORM

#### Data in Use

- Dynamic runtime injection
- Insecure URL schemes
- UI Data leaks
- Clipboard data leaks
- Unnecessary permissions
- ...

#### Data in Motion

- Vulnerable to MITM attacks
- Vulnerable to session hijacking
- Improper TLS validation
- Weak App transport security
- Use of insecure protocols
- Insecure Cookies
- ...

### OWASP API Top 10 & ASVS

#### API Backends

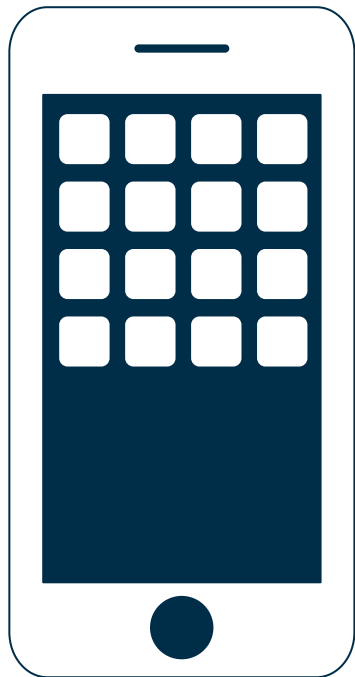
- Unauthenticated APIs
- Unprotected APIs
- Excessive API Data
- API SQL Injection
- Remote code execution
- Privilege Escalation
- Denial of Service
- ...

### MASVS-AUTH

#### MASVS-NETWORK

- Unauthenticated APIs
- Excessive API Data
- API SQL Injection
- Remote code execution
- Privilege Escalation
- Denial of Service
- ...

# 8 Domains of MASVS Requirements



**V1:** Architecture, Design and Threat Modeling

**V2:** Data Storage and Privacy

**V3:** Cryptography

**V4:** Authentication and Session Management

**V5:** Network Communication

**V6:** Environmental Interaction

**V7:** Code Quality and Build Setting

**V8:** Resiliency Against Reverse Engineering

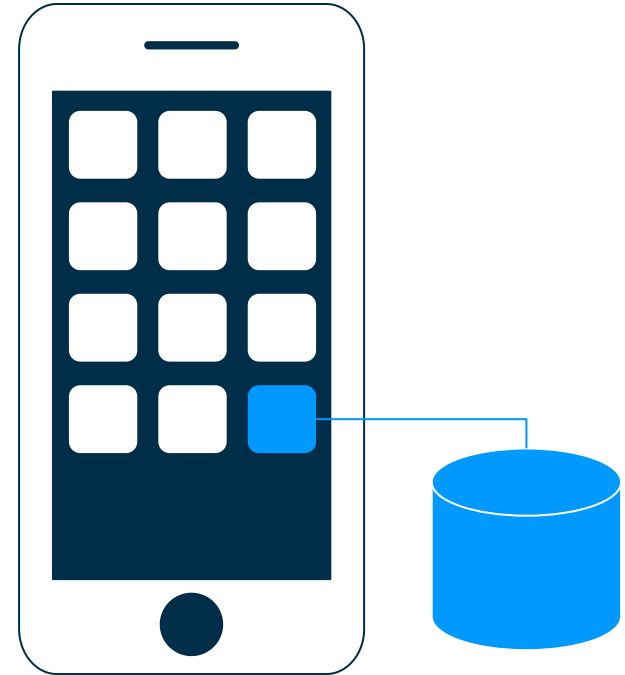
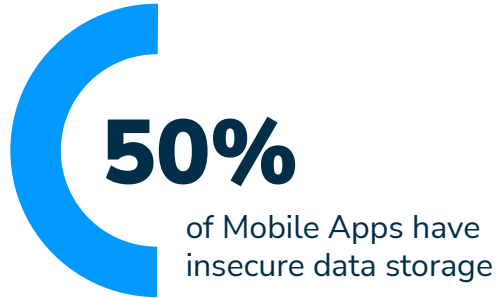
# Top 5 Areas To Focus OWASP MASVS

Let's Use Both the Builder and Breaker POV



# 1 - Insecure Data Storage & Crypto

# Insecure Data Storage & Crypto



# Insecure Data Storage & Crypto



## OWASP MASVS Mapping

- V2: Data Storage & Privacy
- V3: Cryptography

### Resources:

- [OWASP MASVS V2: Insecure Data Storage](#)
- [OWASP MASVS V3: Cryptography](#)
- [Android: Data and file storage overview](#)
- [Apple: File system basics](#)

<b>Security bug:</b>	Use of the device file system without security controls
<b>Attack vector:</b>	Malware, lost/stolen device, malicious USB charger
<b>Business impact:</b>	Identity theft, fraud, policy/compliance violation, data loss, reputational risk



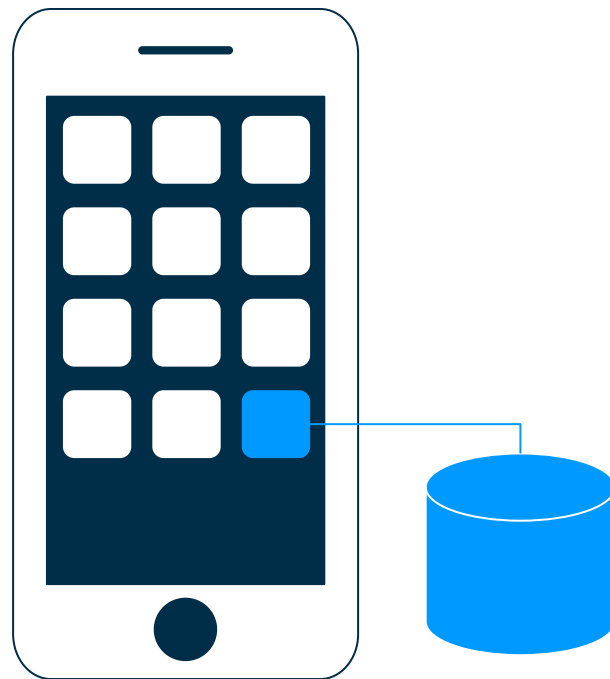
# Insecure Data Storage & Weak Crypto

## Best Practices for Secure Coding

- Avoid writing sensitive data to device
- Encrypt sensitive files
- Avoid query strings in sensitive data
- Implement secure data storage
- Use strong current Crypto (e.g. SHA3)
- Use SecureRandom
- Use Key with a length of at least 2048 bits (preferably 4096 bits)

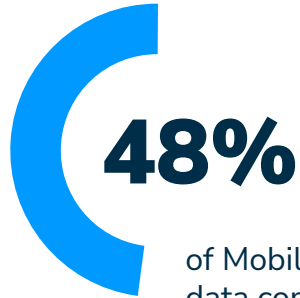
## Best Practices for AppSec Testing

- Test for credentials & PII in files, logs, IPC
- Test for data removed when background
- Test Crypto libs & storage
- Confirm req use of device password
- Check for weak crypto & bad practices

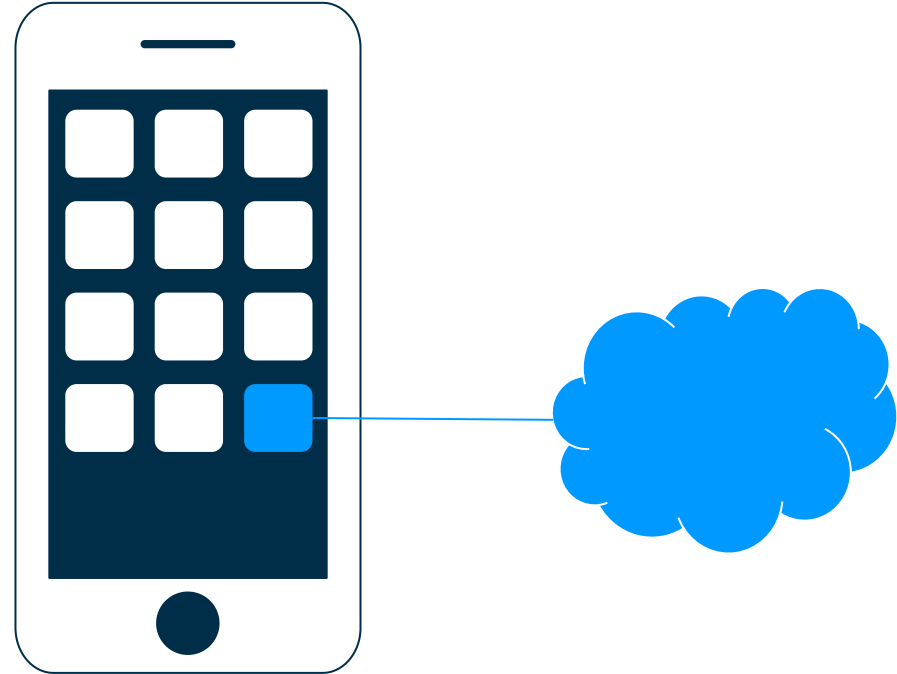


# 2- Insecure Network Communication

# Insecure Network Communication



of Mobile Apps have insecure data communication



# Insecure Network Communication



## OWASP MASVS Mapping

- V5: Network Communication

### Resources:

- [OWASP MASVS V5: Network Comms](#)
- [Android: Network security configuration](#)
- [Apple: Preventing insecure network connection](#)

<b>Security bug:</b>	Unprotected network communications (e.g., use of HTTP, lack of TLS validations)
<b>Attack vector:</b>	Malicious VPN, exploited networks, public Wi-Fi
<b>Business impact:</b>	Identity theft, fraud, reputational risk

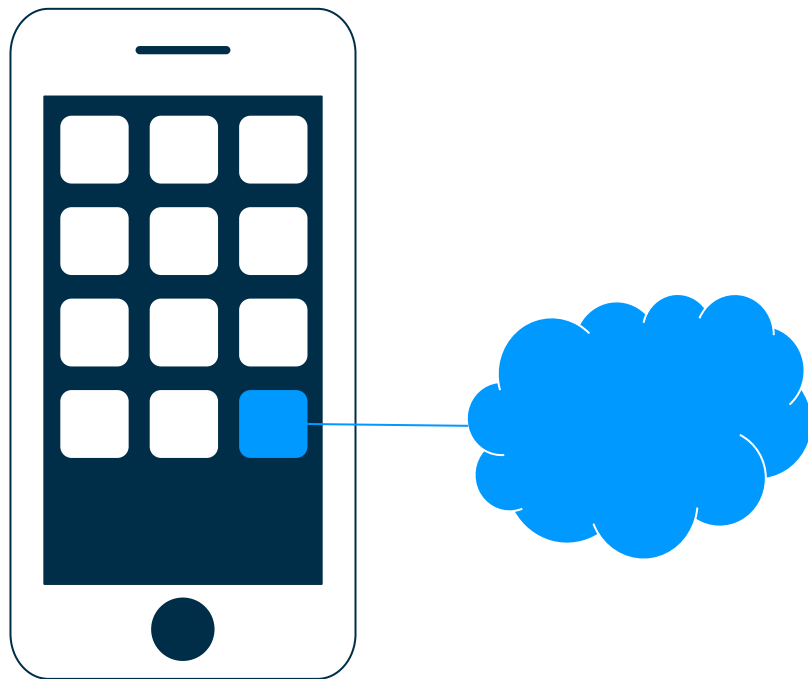
# Improperly Coded Network Calls

## Best Practices for Secure Coding

- Only generate TLS sessions after a successful trust evaluation and a valid DNS name
- Perform certificate pinning for connections carrying regulated data
- Leverage iOS App Transport Security and Android Network Security Configuration
- Learn about how to prevent man-in-the-middle attacks

## Best Practices for AppSec Testing

- Test TLS, Cert Pinning, zip files in transit
- Check for use of ATS & NSC
- Check 3rd party libraries



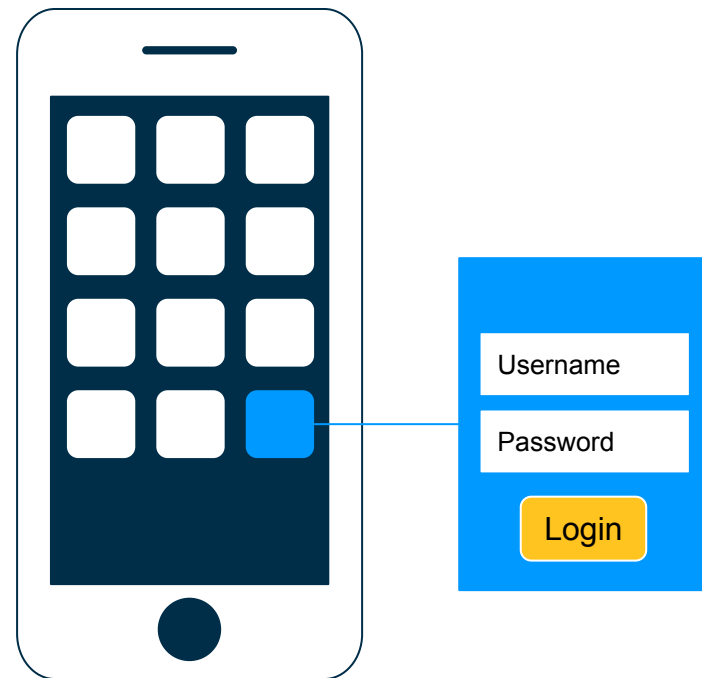
# 3- Insecure Authentication or Authorization

# Insecure Authentication or Authorization



**14%**

of Mobile Apps have  
insecure authentication





# Insecure Authentication or Authorization



## OWASP MASVS Mapping

- V4: Authentication & Session Mgmt

### Resources:

- [OWASP MASVS V4: Auth & Session Mgmt](#)
- [Android: Authenticate Users](#)
- [Apple: User Authentication](#)

<b>Security bug:</b>	Improper authentication scheme (e.g., weak password acceptance), design flaws in session management or authorization scheme (e.g., flaws in user's privilege level, authorization permissions provided through the client-side code)
<b>Attack vector:</b>	API endpoints, stolen device
<b>Business impact:</b>	Unauthorized access, theft, and reputational risk

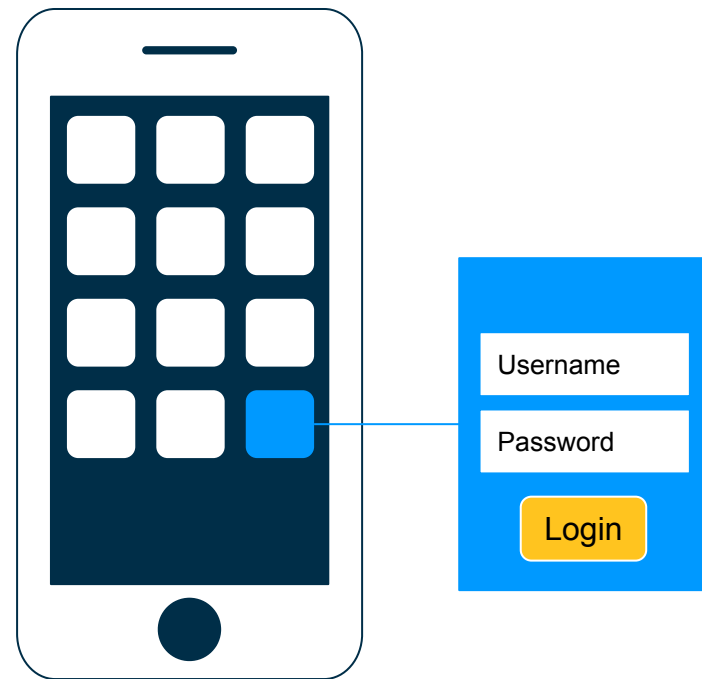
# Insecure Authentication or Authorization

## Best Practices for Secure Coding

- Terminate the active session after a given amount of time
- Ensure no app data is visible when session is invalidated
- Discard and clear all memory associated with the user data and encryption
- Run authorization checks for roles and permissions of an authenticated user at the server, not client side


## Best Practices for AppSec Testing

- Test session validation
- Test data in memory



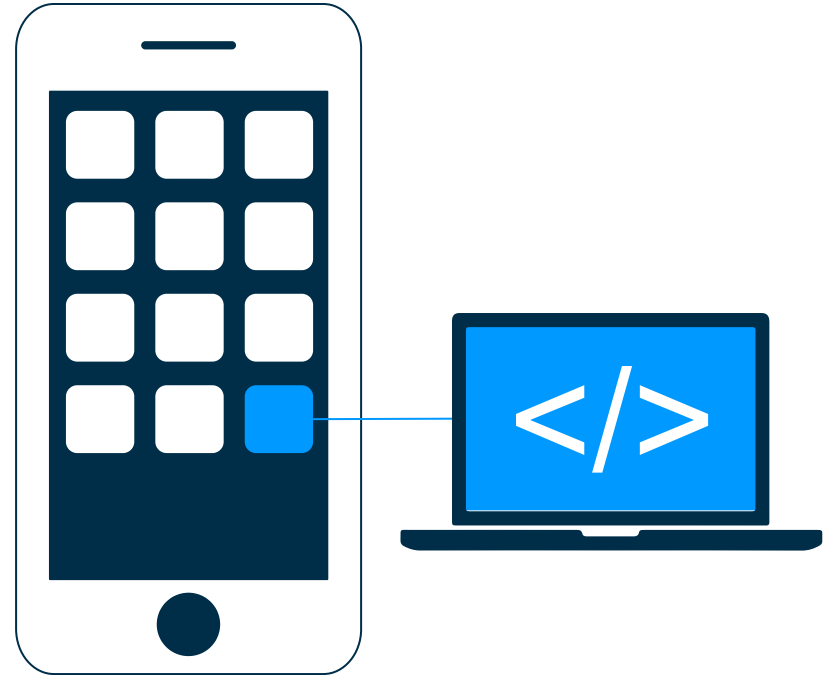
# 4- Insecure Coding Practices

# Insecure Coding Practices



**47%**

of Mobile Apps have insecure  
exploitable extraneous functionality



# Insecure Coding Practices



## OWASP MASVS Mapping

- V7: Code Quality & Build Setting Requirements

### Resources:

- [OWASP MASVS V7: Code Quality](#)

<b>Security bug:</b>	Issue as a result of poor coding practices (e.g., logic flaws in code, vulnerable third-party library, buffer overflows and memory leaks), unnecessary component built into app (e.g., debug features, security controls)
<b>Attack vector:</b>	Malware, phishing, unsuspected user, extraneous func. feature
<b>Business impact:</b>	Data theft, reputational risk, fraud, unauthorized access

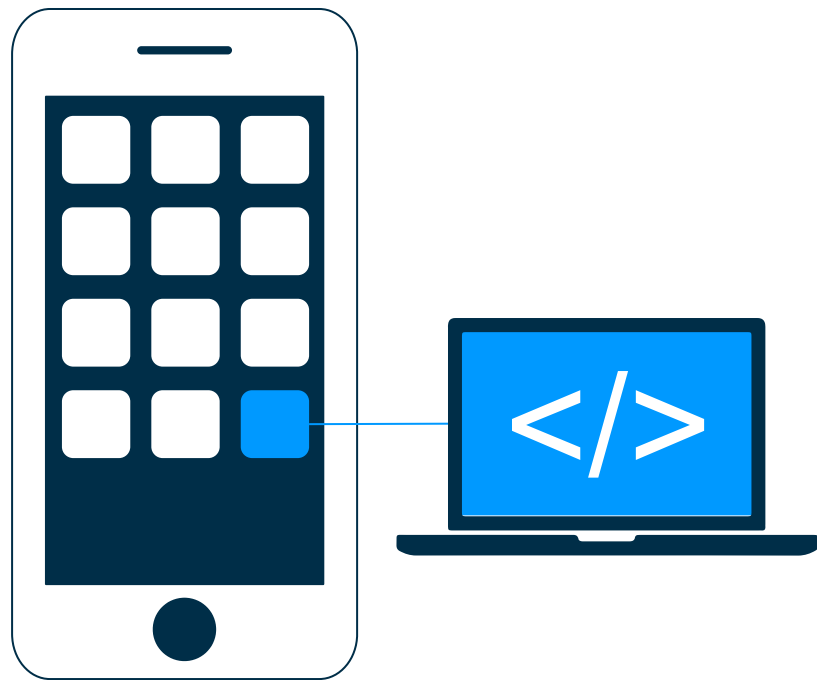
# Insecure Coding Practices

## Best Practices

- Remove Debug symbols & code
- Ensure Secure Coding practices
- Use free security features offered by the toolchain (stack protection, ARC, etc.)
- Keep track of 3rd party dependencies with an SBOM! Scan for well-known vulnerabilities

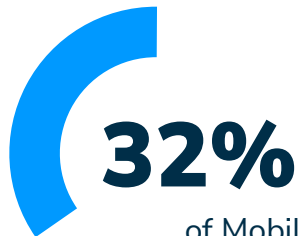
## Best Practices for AppSec Testing

- Test app signed with valid cert
- Test for debug build, hardcoded keys
- Test error conditions, verbose log files
- Check 3rd party libraries

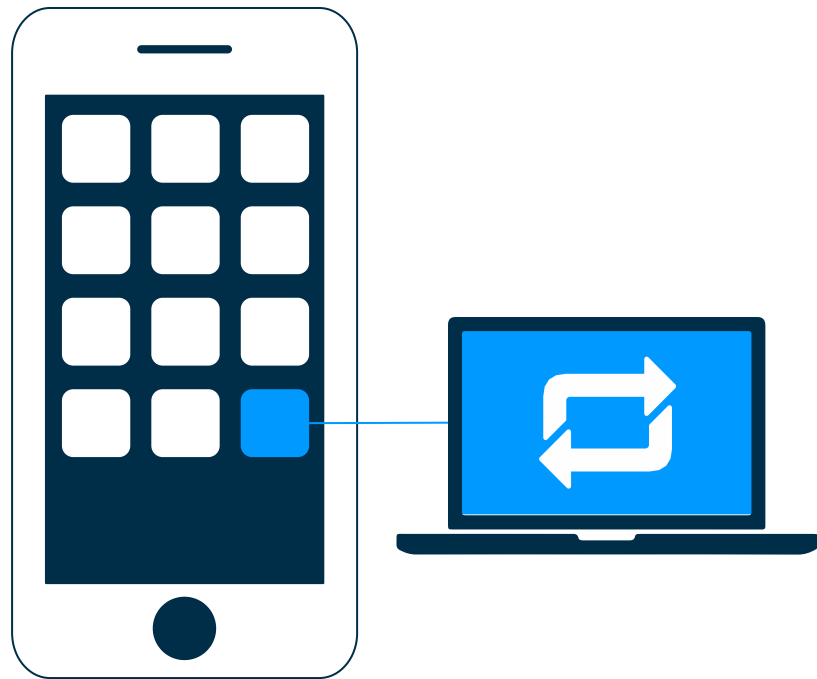


# 5- Reverse Engineering & Anti-Tampering

# Exposure to Reverse Engineering



of Mobile Apps have exposure to reverse engineering





# Reverse Engineering



## OWASP MASVS Mapping

- V8: Resiliency Against Reverse Engineering & Tampering

### Resources:

- [OWASP MASVS V8: Resiliency](#)
- [OWASP Reversing Prevention Project](#)
- Reversing tools: [Frida](#), [Radare](#), [2Frida Repo](#)

<b>Security bug:</b>	Unprotected IP and binary enables attackers to reverse engineer process and data to exploit in other ways
<b>Attack vector:</b>	Reverse engineering of mobile app binary
<b>Business impact:</b>	Data theft, IP theft, reputational risk, fraud, unauthorized access

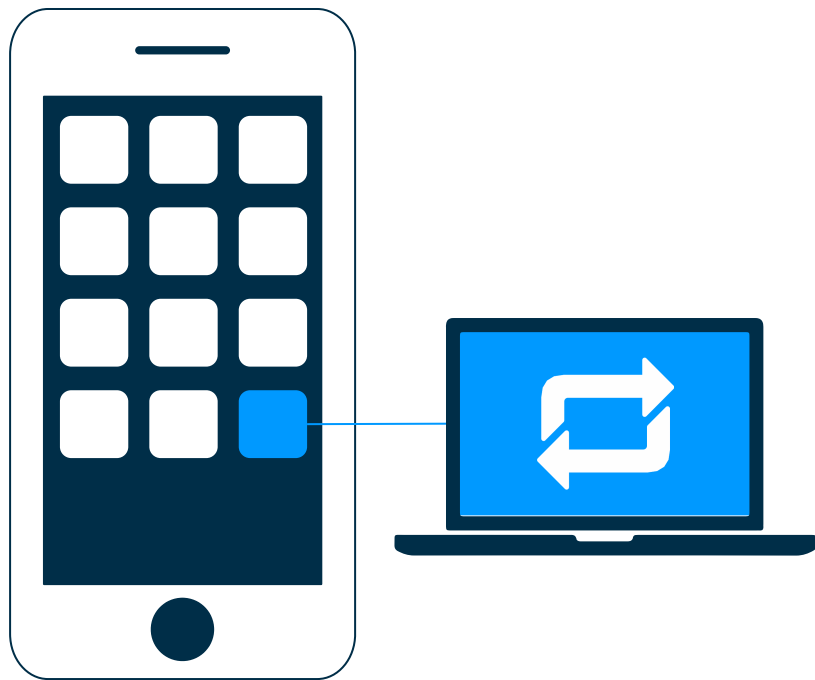
# Exposure to Reverse Engineering

## Best Practices for Secure Coding

- Use third-party code obfuscation tools, especially for Android apps
- Use Android SafetyNet API to check for Android device tampering
- Implement anti-tampering techniques

## Best Practices for AppSec Testing

- Test for reversibility via detect JB/root, debugger, data/file manipulation
- Test String tables & methods
- Check for Android SafetyNet API

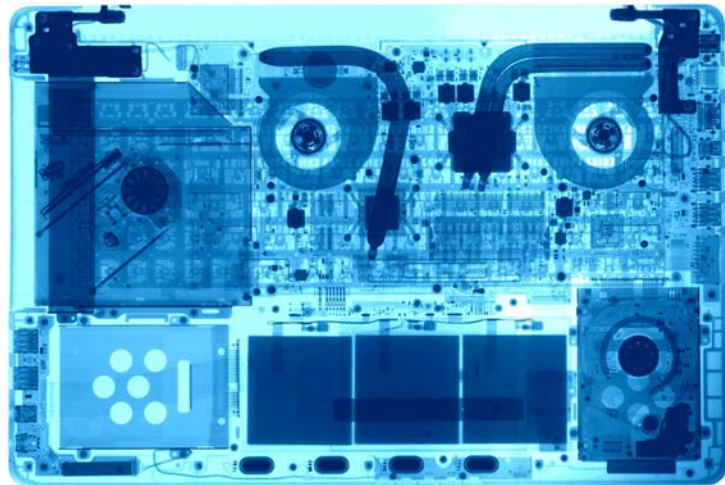


# Resiliency Against Reverse Engineering & Tampering

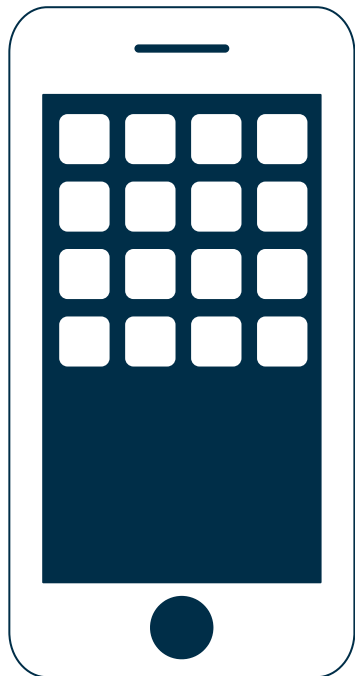
## Testing Tip

Tamper proofing helps, but only so far...

“Anti tampering doesn’t fix security bugs, or protect security bugs in production code...”



# Key Takeaways



Recognize Mobile & Web are different

Get to know the OWASP Mobile Project

Start exploring, leverage the great resources!

Build your skills and toolkit

Threat modeling is your friend

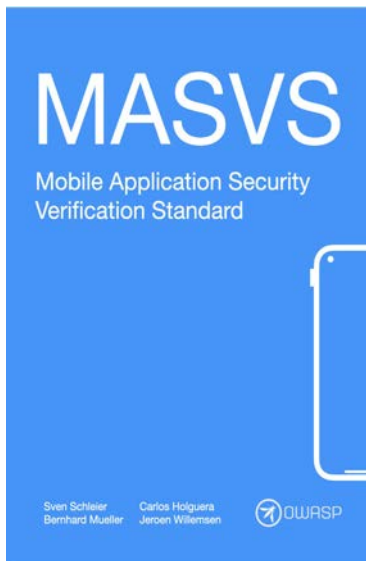
The 8 Requirements help break down the problem

Start with the Big 5 (storage, network, auth, code, RE)

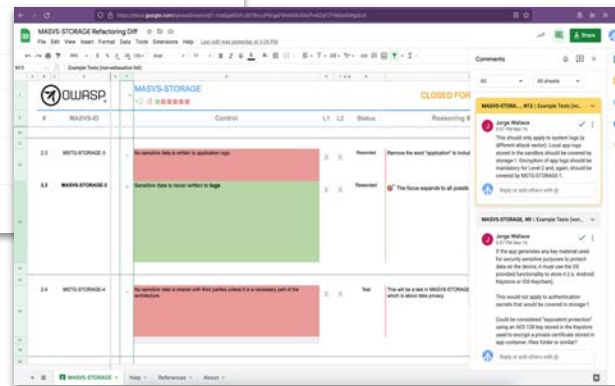
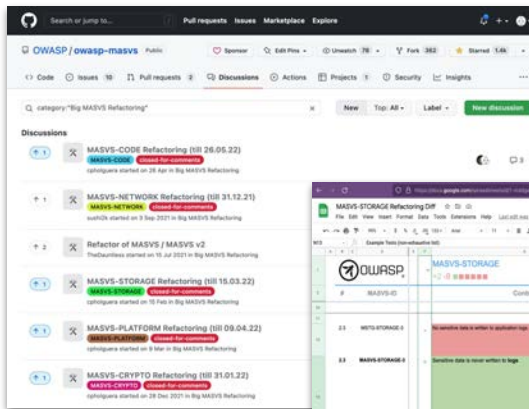
Get involved in the OWASP Mobile Project - Sign Up!

# OWASP MASVS Project Updates

# OWASP MASVS V2 Refactoring Process Update



- MASVS-NETWORK
- MASVS-CRYPTO
- MASVS-STORAGE
- MASVS-PLATFORM
- MASVS-CODE
- MASVS-AUTH
- MASVS-ARCH
- MASVS-RESILIENCY



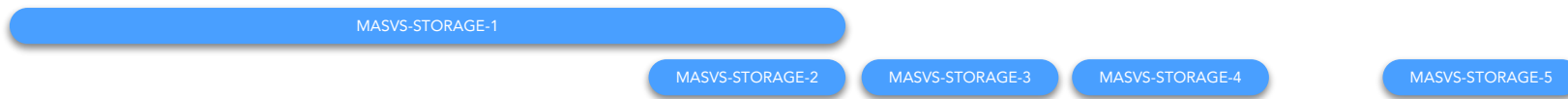
<https://github.com/OWASP/owasp-masvs/discussions/categories/big-masvs-refactoring>

# OWASP MASVS Refactoring Process

Key Areas



Controls



Tests



# OWASP MASVS V2 Compliance-as-Code



Read and interpret manually

Hard to prove control and test coverage

Compare providers manually

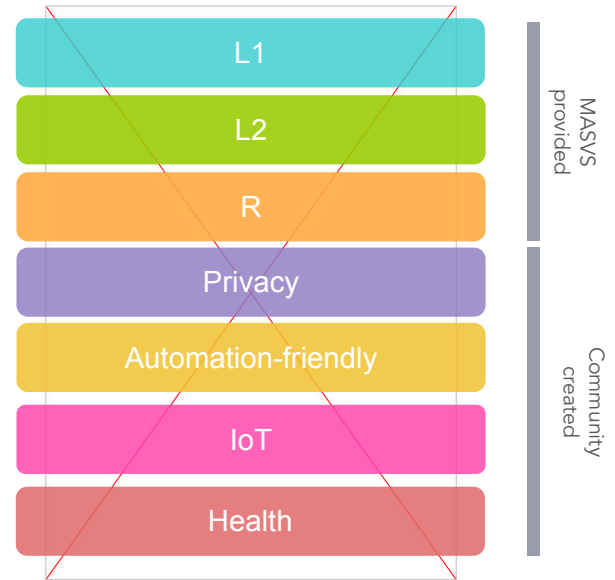
Hard to maintain

Machine-readable

Easy to prove control and test coverage

Compare providers with benchmarking

Fully traceable



**Standard and fully tailored testing**

MASVS + proprietary + cross-standards



# Join Our OWASP Project Team

Fix  
typos

Improve our  
Android / iOS  
Crackme apps

Review PRs

Enhance / write  
new Test Cases

Try out new  
hacking tools



Design our Swag

Answer  
Discussions

Help us automate &  
GitHub Actions

Give feedback  
to the MASVS  
Refactoring

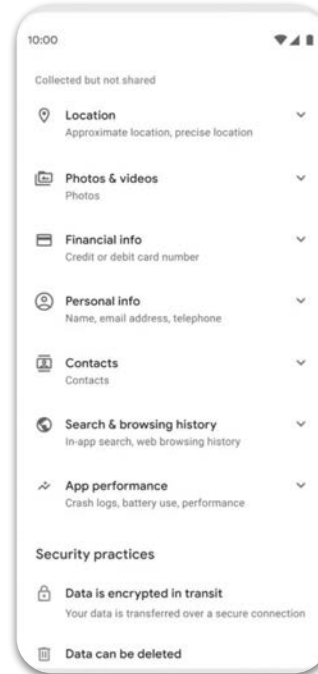
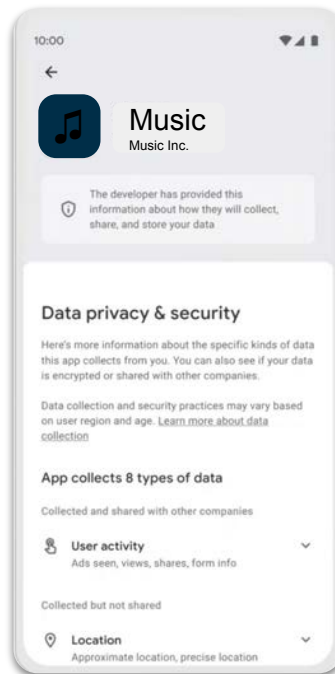
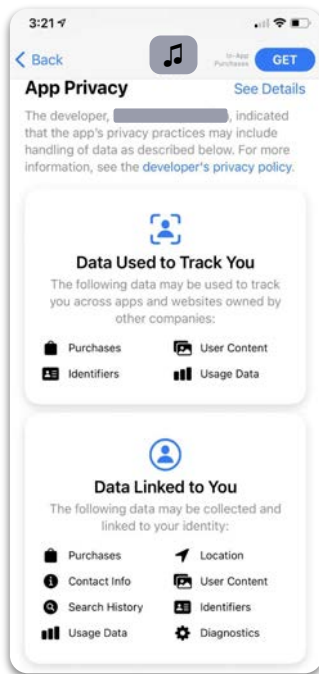
**Contribute & connect with us!**

<https://github.com/OWASP/owasp-mstg#connect-with-us>

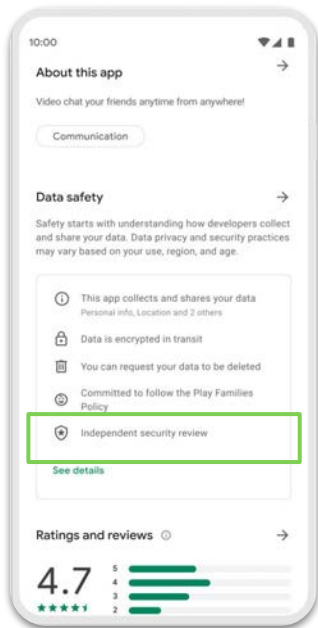
...

# Apple and Google Updates

# Apple Privacy And Google Play Data Safety

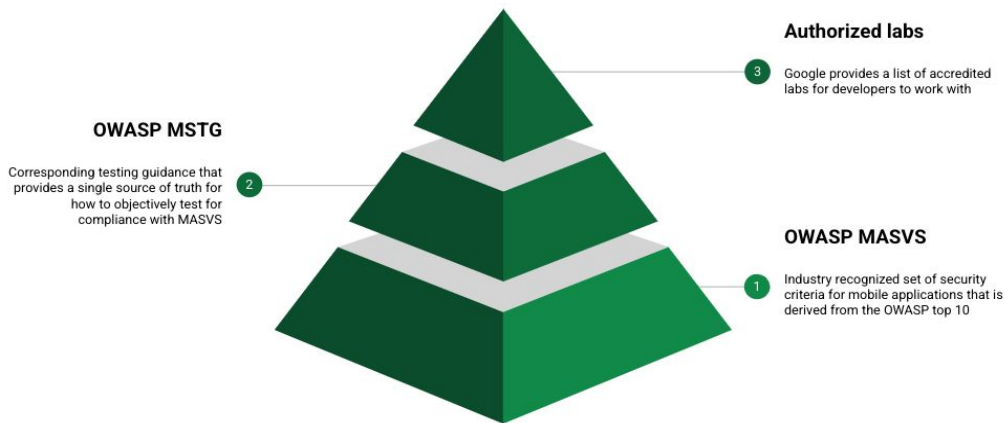


# ADA Authorized Labs with MASA Verification



Thanks to Google's App Defense Alliance (ADA), Developers can showcase [key privacy and security practices](#), at a glance.

By [July 20th 2022](#), the Data safety section for all your apps must be approved.



[App Defense Alliance: Mobile Application Security Assessment](#)

# ADA Mobile App Security Assessments (MASA)

MASA has a published formal set of requirements

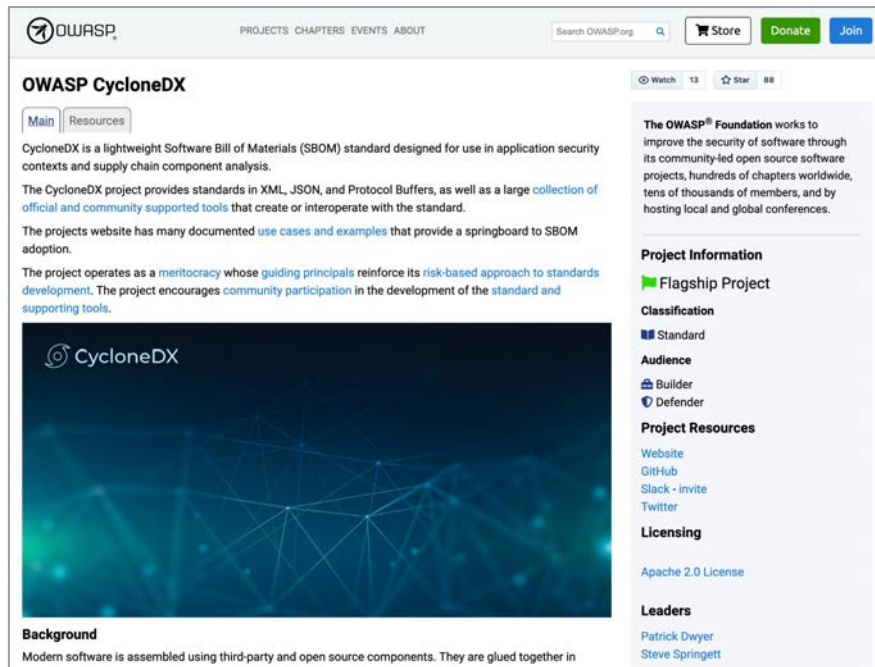
Based on OWASP MASVS and MSTG

Data Storage and Privacy Requirements	Cryptography Requirements	Authentication and Session Management Requirements	Network Communication Requirements	Platform Interaction Requirements	Code Quality and Build Setting Requirements
MSTG-STORAGE-1 System credential storage facilities used to store sensitive data	MSTG-CRYPTO-1 app does not rely on symmetric cryptography with hardcoded keys	MSTG-AUTH-1 Authentication for remote services	MSTG-NETWORK-1 Data is encrypted on the network using TLS	MSTG-PLATFORM-1 requests the minimum set of permissions	MSTG-CODE-1 app is signed and provisioned with a valid certificate
MSTG-STORAGE-2 No sensitive data should be stored outside of the app container	MSTG-CRYPTO-2 proven implementations of cryptographic primitives	MSTG-AUTH-2 randomly generated session identifiers	MSTG-NETWORK-2 The TLS settings are in line with current best practices	MSTG-PLATFORM-2 inputs from external sources and the user are validated	MSTG-CODE-2 app has been built in release mode
MSTG-STORAGE-3 No sensitive data is written to application logs	MSTG-CRYPTO-3 app uses cryptographic primitives that are appropriate for the particular use-case	MSTG-AUTH-3 stateless token-based authentication are signed	MSTG-NETWORK-3 The app verifies the X.509 certificate of the remote endpoint	MSTG-PLATFORM-3 app does not export sensitive functionality via custom URL schemes	MSTG-CODE-3 Debugging symbols have been removed from native binaries.
MSTG-STORAGE-5 The keyboard cache is disabled on sensitive data inputs	MSTG-CRYPTO-4 No deprecated cryptographic protocols or algorithms	MSTG-AUTH-4 remote endpoint terminates the existing session when the user logs out		MSTG-PLATFORM-4 app does not export sensitive functionality through IPC facilities	MSTG-CODE-4 Debugging code and developer assistance code have been removed
MSTG-STORAGE-7 No sensitive data is exposed through the user interface.	MSTG-CRYPTO-5 No re-use the same cryptographic key for multiple purposes.	MSTG-AUTH-5 password policy exists and is enforced at the remote endpoint			MSTG-CODE-5 third party components are checked for known vulnerabilities
MSTG-STORAGE-12 educate the user about the types of personally identifiable information processed	MSTG-CRYPTO-6 random values are generated using a sufficiently secure random number generator	MSTG-AUTH-6 Brute force mitigations			MSTG-CODE-9 security features offered by the toolchain are activated
		MSTG-AUTH-7 Sessions are invalidated at the remote endpoint after a pre-defined period of inactivity			

# OWASP CycloneDX for SBOM

# What is OWASP CycloneDX?

Best Practice for MASVS-CODE



The screenshot shows the OWASP CycloneDX project page. The header includes the OWASP logo, navigation links for PROJECTS, CHAPTERS, EVENTS, and ABOUT, a search bar, and buttons for Store, Donate, and Join. The main content area features the title "OWASP CycloneDX" with tabs for Main and Resources. The text describes CycloneDX as a lightweight SBOM standard for application security and supply chain analysis. It mentions standards in XML, JSON, and Protocol Buffers, and a collection of tools. A sidebar on the right provides project information, including its classification as a "Flagship Project", audience (Builder and Defender), project resources (Website, GitHub, Slack, Twitter), licensing (Apache 2.0), and leaders (Patrick Dwyer and Steve Springett). A large image of the CycloneDX logo is also visible.

## New Flagship Project at OWASP

A new industry standard for SBOM interoperability

Chaired by Steve Springett & Patrick Dwyer

*“The CycloneDX SBOM standard is a result of security experts and industry coming together to create an SBOM standard that delivers the transparency and interoperability necessary to communicate software inventory and the relationships across different systems.”*

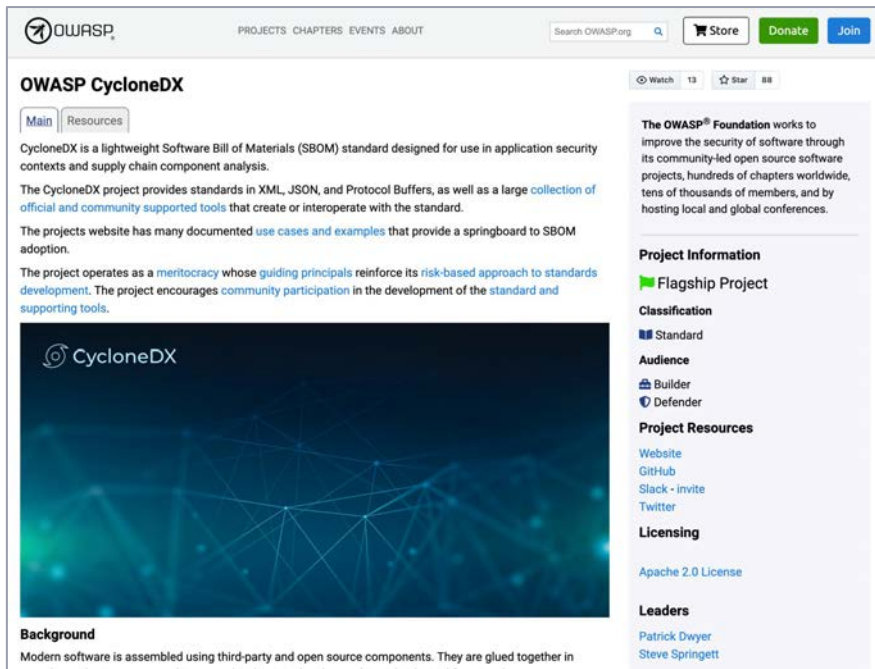
Cross links with OWASP MASVS Project as well

*Link to Dependency Track SBOM tool*

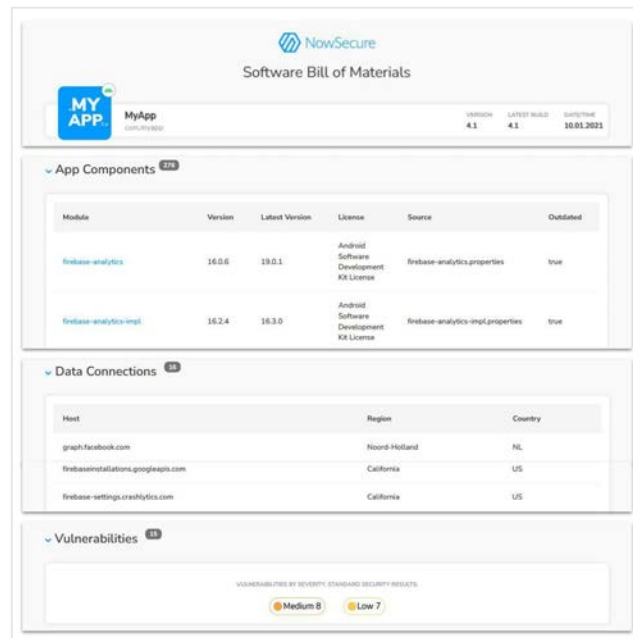
<https://dependencytrack.org/>

<https://owasp.org/www-project-cyclonedx/>

# What is OWASP CycloneDX?



The screenshot shows the OWASP CycloneDX website. The header includes the OWASP logo, navigation links for PROJECTS, CHAPTERS, EVENTS, and ABOUT, a search bar, and buttons for Store, Donate, and Join. The main content area is titled "OWASP CycloneDX" and has tabs for Main and Resources. The text describes CycloneDX as a lightweight SBOM standard for application security and supply chain analysis. It mentions standards in XML, JSON, and Protocol Buffers, and a collection of official and community-supported tools. It also notes that the project operates as a meritocracy with guiding principals reinforcing a risk-based approach to standards development. A large image of the CycloneDX logo is shown below the text. To the right, there is a sidebar with "Project Information" (Flagship Project, Classification: Standard, Audience: Builder, Defender), "Project Resources" (Website, GitHub, Slack, Twitter), and "Licensing" (Apache 2.0 License). A "Background" section at the bottom states that modern software is assembled using third-party and open source components.



The screenshot shows a NowSecure Software Bill of Materials (SBOM) interface. At the top, it displays the NowSecure logo and the title "Software Bill of Materials". Below this, there is a "MY APP" section with a "MyApp" label and a table showing version information: VERSION 4.1, LATEST BUILD 4.1, and DATE/TIME 10.01.2021. The interface is divided into several sections: "App Components" (2 items), "Data Connections" (3 items), and "Vulnerabilities" (3 items). The "App Components" section contains a table with columns for Module, Version, Latest Version, License, Source, and Outdated. The "Data Connections" section contains a table with columns for Host, Region, and Country. The "Vulnerabilities" section shows a summary of vulnerabilities by severity: Medium 8 and Low 7.

Module	Version	Latest Version	License	Source	Outdated
firebase-analytics	16.0.6	19.0.1	Android Software Development Kit License	firebase-analytics.properties	true
firebase-analytics-impl	16.2.4	16.3.0	Android Software Development Kit License	firebase-analytics-impl.properties	true

Host	Region	Country
graph.facebook.com	Noord-Holland	NL
firebaseinstallations.googleapis.com	California	US
firebase-settings.crashlytics.com	California	US

<https://owasp.org/www-project-cyclonedx/>

Get Free Mobile SBOMS  
<https://bit.ly/ns-SBOM10>

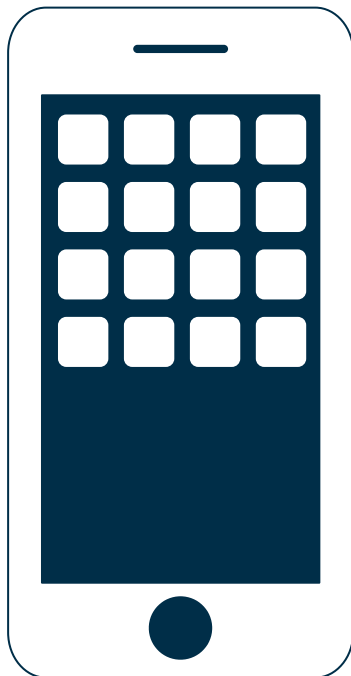


Resources Resources Resources

# Mobile Pen Tester's Toolkit

## Manual & OSS Testing Resources

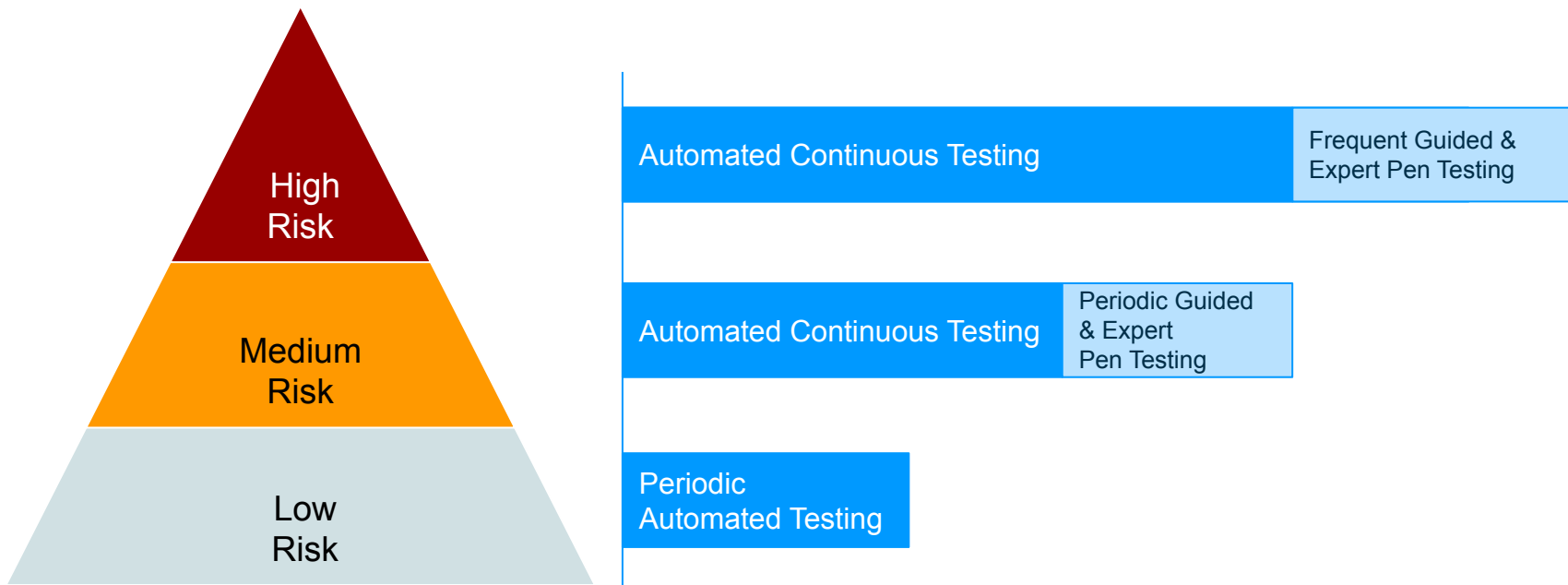
- MASVS [repo](#)
- MSTG [repo](#)
- MSTG [Hacking Playground](#)
- Frida [Dynamic Instrumentation Toolkit](#)
- Radare [Portable Reversing Framework](#)
- [Burp Suite](#) or [ZedAttackProxy](#)
- Jailbroken & Rooted devices



## Automated Testing Resources

- Free Mobile [SBOMs](#)
- Free Mobile Analysis [Report](#)
- Free Online Training [Academy](#)
- NowSecure Workstation [Toolkit](#)
- NowSecure Platform [Automation](#)
  - ✓ 600+ security, privacy and compliance tests
  - ✓ SAST+DAST+IAST+APISec
  - ✓ Automated & Interactive Modes
  - ✓ Embedded remediation

# Best Practice Tuning Security Test Coverage & Frequency



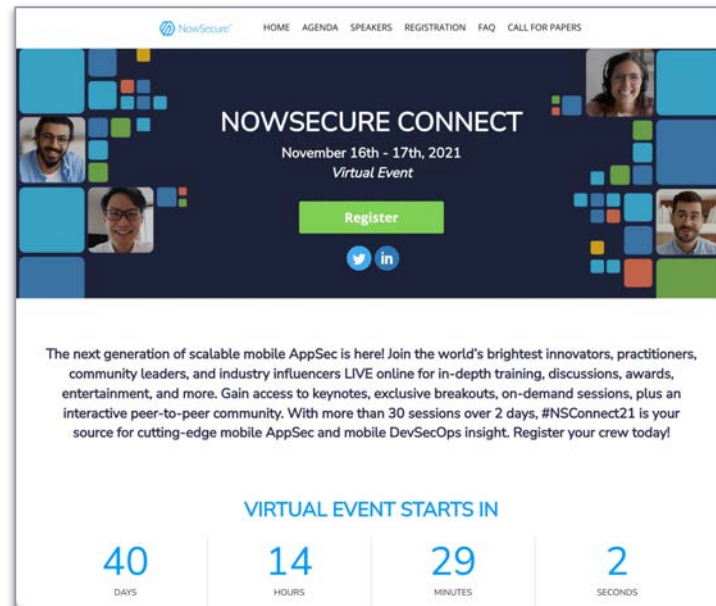
# Free Training



The screenshot shows the NowSecure Academy website. At the top, there is a navigation bar with links for Certificates, All Courses, NowSecure.com, NowSecure Learning Center, and Customers, along with a search bar. The main header features the NowSecure Academy logo. Below the header, a large blue hexagonal graphic contains an image of people working together. To the left of this graphic, the text reads "Welcome to NowSecure Academy" and "Ready to build your mobile app security skills? Our experts are here to help. NowSecure Academy provides self-service mobile app security and privacy courses, best practices, certificates, resources and more." At the bottom of the page, there are two sections: "Certificate Programs" with the subtext "Build a strong foundation through our structured learning paths." and "Continuous Learning" with the subtext "Keep current with updated content and value-added courses." A blue bar at the very bottom contains the text "Check out all of the courses that we offer!" and a "Click Here" button.

Online Courseware

<https://academy.nowsecure.com>



The screenshot shows the registration page for the "NOWSECURE CONNECT" virtual event. The top navigation bar includes links for HOME, AGENDA, SPEAKERS, REGISTRATION, FAQ, and CALL FOR PAPERS. The main header features the event title "NOWSECURE CONNECT" and the dates "November 16th - 17th, 2021 Virtual Event". A prominent green "Register" button is centered on the page, with social media icons for Twitter and LinkedIn below it. The page is decorated with a grid of colorful squares and several small video thumbnails of speakers. Below the registration section, a paragraph of text describes the event: "The next generation of scalable mobile AppSec is here! Join the world's brightest innovators, practitioners, community leaders, and industry influencers LIVE online for in-depth training, discussions, awards, entertainment, and more. Gain access to keynotes, exclusive breakouts, on-demand sessions, plus an interactive peer-to-peer community. With more than 30 sessions over 2 days, #NSConnect21 is your source for cutting-edge mobile AppSec and mobile DevSecOps insight. Register your crew today!" At the bottom, a countdown timer displays "VIRTUAL EVENT STARTS IN" followed by four columns: "40 DAYS", "14 HOURS", "29 MINUTES", and "2 SECONDS".

Full Replays

<https://bit.ly/ns-connect>

# Checkout Your Own Mobile Apps

The screenshot shows the NowSecure interface for a Software Bill of Materials (SBOM) for an application named 'MyApp'. The app version is 4.1, the latest build is 4.1, and the submission date is 10.01.2021. The interface is divided into three main sections: App Components, Data Connections, and Vulnerabilities.

**App Components**

Module	Version	Latest Version	License	Source	Outdated
firebase-analytics	16.0.6	19.0.1	Android Software Development Kit License	firebase-analytics.properties	true
firebase-analytics-impl	16.2.4	16.3.0	Android Software Development Kit License	firebase-analytics-impl.properties	true

**Data Connections**

Host	Region	Country
graph.facebook.com	Noord-Holland	NL
firebaseinstallations.googleapis.com	California	US
firebase-settings.crashlytics.com	California	US

**Vulnerabilities**

At the bottom, there are two vulnerability indicators: a yellow circle with 'Medium 8' and a green circle with 'Low 7'.

Free SBOM  
<https://bit.ly/ns-SBOM10>

The screenshot shows a Security Report for an application named 'CoronaFacts'. The report includes a 'Critical Security Score' of 25/100. The findings section lists 120 results, with several high-severity issues highlighted, such as 'Password Exposed and Modifiable Over the Network' (High, CVEs 8.1), 'Email Address Exposed and Modifiable Over the Network' (High, CVEs 7.1), and 'Using HTTP Exposing Network Data to Interception and Manipulation' (Medium, CVEs 6.5). A detailed view of the 'Password Exposed and Modifiable Over the Network' finding is shown on the right, including a description, steps to reproduce, business impact, and evidence. The evidence table shows 116 results, with columns for Type, Format, Value, Full URL, Searched Data, Context, and Scope.

Type	Format	Value	Full URL	Searched Data	Context	Scope
automation	password	plaintext	40402	http://3.22.1.42:397/ContentV2/AppLogin	device_id=11975179f86a7f2f8be39c033996479ca4a64f4dc3e2077faed89fae0260&device_type=android & email=arthur.dent@nowsecure.com & password=40402y4knc42	NSURL, Session Task
automation	password	plaintext	40402	http://3.22.1.42:397/ContentV2/AppLogin	device_id=11975179f86a7f2f8be39c033996479ca4a64f4dc3e2077faed89fae0260&device_type=android & email=arthur.dent@nowsecure.com & password=40402y4knc42	NSURL, Session Task

Free Security Report  
<https://bit.ly/ns-report>

# More Free Resources



<http://bit.ly/ns-mgr-masvs>



<http://bit.ly/ns-owasp-top5>



<http://bit.ly/ns-maspmh>



OWASP Android  
CrackMe r2Comm

<http://bit.ly/ns-owasp-acme>

# NowSecure Full Mobile AppSec Solution Suite

## NowSecure Platform

Continuous security testing for mobile DevSecOps



## NowSecure Supply Chain

Continuous monitoring of app store mobile risk



## NowSecure Workstation

All-in-one mobile pen tester toolkit for productivity



## NowSecure Academy

Online courseware and certification for mobile



## NowSecure Pen Testing

Expert full scope mobile pen testing services & remediation



## NowSecure Mobileverse™

Customer community to onboard, learn & network with peers

	<b>CoronaFacts</b> com.CoronaFacts	PLATFORM iOS	ORIGIN App Store	VERSION 2.0	BUILD 2.0-1	ASSESSMENT DATE / TIME 14 Nov 2021 - 18:12:52	Covid19
--	---------------------------------------	-----------------	---------------------	----------------	----------------	--	---------



# Security Report

Report  
Security Report

● Critical Security Score **25/100**

Findings **Debug**

117 Results

Severity: All  
Sort By  
CVSS (High to Low)

- Password Exposed and Modifiable Over the Network**  
● High CVSS 8.1
- Email Address Exposed and Modifiable Over the Network**  
● High CVSS 7.1
- Using HTTP Exposing Network Data to Interception and Manipulation**  
● Medium CVSS 6.5
- Device Info Exposed and Modifiable Over the Network**  
● Medium CVSS 5.3
- Disabled App Protection (ATS) Can Lead to...**

## Password Exposed and Modifiable Over the Network

● High CVSS 8.1

### Context

**Description**  
Password was intercepted over HTTP traffic. A remote attacker with access to the local or upstream network as the user could use network monitoring software, such as Wireshark, to observe and modify the data.

### Steps To Reproduce

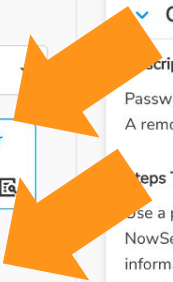
Use a packet interception and analysis tool, such as Wireshark, on your testing network to identify unencrypted network traffic that may contain sensitive information. NowSecure's test for this finding involves capturing HTTP traffic between an app running on a physical device and servers. The resulting HTTP traffic is examined for the presence of credential information which results in a list of credentials leaked to servers over insecure HTTP communications.

### Business Impact

The app is not encrypting sensitive information being sent over the internet. A malicious actor could remotely see and/or modify the sensitive data coming to and from the endpoints listed, potentially affecting many users at once. Depending on the type of data being transmitted insecurely, this vulnerability could lead to exposure of sensitive personal data and/or intellectual property.

### Evidence

74 Results







**TurboLock Plus**

com.xctx.mlock

PLATFORM

Android

ORIGIN

App Store

VERSION

3.5

BUILD

218

ASSESSMENT DATE / TIME

25 Mar 2022 - 11:23:38

Mobile IOT



157 Results



Severity: All

Sort By

CVSS (High to Low)

Keyboard Cache Potentially Exposing Sensitive Data

Info



Network Connections

Info



Privacy Policy

Info

Reflection Code Locations

Info



Software Bill of Materials - Included Libraries (Beta)

Info



SQLite Results

Info



Automation Info

Artifact



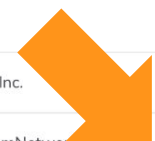
File Listing

Artifact



13 Results

Domain	Host	IP	Port	Organization	Location
> taobao.com	plbslog.umeng.com	::ffff:106.11.223.204	443	Zhejiang Taobao Network Co. Ltd	Hangzhou, Zhejiang, CN
> alibaba.com	ulogs.umeng.com, ulogs.umengcloud.com	::ffff:47.246.109.109	443	Alibaba.com LLC	Singapore, Singapore, SG
> amazon.com	smart.kebijia.com	::ffff:52.1.231.97	80	Amazon Technologies Inc.	Ashburn, Virginia, US
> ubistor.com	easytomessage.com, sjpush.cn, sis.jp.push.io	::ffff:103.230.236.25	19000	XIAMEN CenturyNetcomNetwork Services Limited	Xiamen, Fujian, CN
> huawei.com	122.9.121.124	::ffff:122.9.121.124	7000	Huawei Public Cloud Service	Guangzhou, Guangdong, CN
> huawei.com	bjuser.jp.push.cn	::ffff:122.9.15.248	443	Huawei Public Cloud Service	Guangzhou, Guangdong, CN
> chinamobileltd.com	183.232.25.163	::ffff:183.232.25.163	21004	China Mobile Communications Corporation	Guangzhou, Guangdong, CN
> huawei.com	121.36.205.81	::ffff:121.36.205.81	7000	Huawei Public Cloud Service	Guangzhou, Guangdong, CN
> chinamobileltd.com	183.232.58.113	::ffff:183.232.58.113	21003	China Mobile Communications Corporation	Guangzhou, Guangdong, CN
> huawei.com	121.36.75.206	::ffff:121.36.75.206	7006	Huawei Public Cloud Service	Beijing, Beijing, CN





## pMp COVID-19

co.patientbuddy.tracker.covid19

PLATFORM

IOS

ORIGIN

App Store

VERSION

1.4.0

BUILD

1.4.0-1031.3.4.0

ASSESSMENT DATE / TIME

2 Sep 2021 - 11:42:08

Covid19



## Security Report

Report

Security Report

● Poor Security Score 48/100

Findings

Debug

126 Results



Severity: All

Sort By

CVSS (High to Low)

## Outdated nanopb Library Contains Known Security Flaw

● High CVSS 7.1



## Disabled App Protection (ATS) Can Lead to Insecure Network Connections

● Medium CVSS 5.3

## App is Encoding Sensitive Information Using Outdated or Insecure Cryptography

● Medium CVSS 4.8



## Allowing Third Party Keyboards Potentially Exposes User Input

● Medium CVSS 4

## Weak Cryptographic Hashing Algorithms

## Outdated nanopb Library Contains Known Security Flaw

● High CVSS 7.1



## Context

## Description

The application was found to be using a vulnerable version of the nanopb library. The library does not properly validate information that it processes which can lead to unintended access or potentially malicious code being run. This test specifically checks for versions < 2.30908.0 as cited by CVE-2021-21401.

## Business Impact

The app is using a 3rd party library which contains a known, high risk flaw which could expose the application and its users to severe attacks.

## Evidence

## Included nanopb Versions

1 Result

Version	Source
> 2.30907.0	Payload/pMp COVID-19.app/Frameworks/nanopb.framework/Info.plist

Run Assessment

Apps

pmp

1 Result

Sort by Score

pMp COVID-19

Package Details



pMp COVID-19 co.patientbuddy.tracker.covid19

APP pMp COVID-19

PLATFORM iOS

LICENSE TYPE Baseline

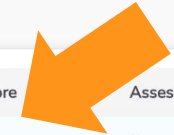
Covid19

Versions

10 Results

Assessments 1

Version	Build	Origin	Security Score	Assessments
1.4.0	1.4.0-1031.3.4.0	App Store	Poor - 48	1
1.3.0	1.3.0-1029.3.3.0	App Store	Good - 75	1
1.2.2	1.2.2-1028.3.2.2	App Store	Good - 77	1
1.2.1	1.2.1-1028.3.2.1	App Store	Partial Results	1
1.2.0	1.2.0-1027.3.2.0	App Store	Partial Results	1
1.1.5	1.1.5-1022.3.0.0	App Store	Good - 76	1
1.1.4	1.1.4-1021.2.9.9	App Store	Good - 76	1
1.1.3	1.1.3-1020.2.9.8	App Store	Good - 76	1
1.1.2	1.1.2-1019.2.9.6	App Store	Good - 76	1
1.1.1	1.1.1-1018.2.9.3	App Store	Good - 77	2



DATE/TIME: 2 Sep 2021 11:42:08

TYPE: Baseline

Security Score: Poor - 48

Findings: 126

Vulnerabilities: 9

[View Security Report](#)

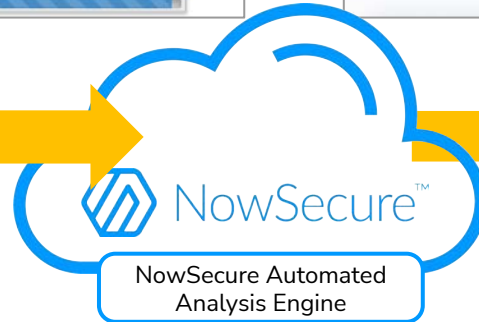
Showing 25

# Sample Automated Workflow: Build, Test, Ticket, Repair

The screenshot shows the Jenkins Pipeline Plugin Demo interface. On the left, there's a sidebar with navigation options like 'Back to Dashboard', 'Status', 'Changes', 'Build Now', etc. The main area is titled 'Pipeline Pipeline Plugin Demo'. It displays 'Last Successful Artifacts' with a list of files and their sizes. Below that is the 'Stage View' showing a progress bar for a pipeline with stages: 'do' (290ms), 'all' (100ms), 'the' (104ms), 'things' (97ms), and 'dynamic-security-test' (15s). A yellow arrow points from the 'dynamic-security-test' stage to the right.

The screenshot shows the NowSecure CI Integration Demo interface. It displays a list of issues under the heading 'Issues'. The issues include 'NowSecure dynamic analysis: Third-Party Keyboards Enabled', 'NowSecure static analysis: ATS Exemptions Requiring Review', 'NowSecure static analysis: Keysize Check', and 'NowSecure static analysis: Debug'. A yellow arrow points from the 'dynamic-security-test' stage in the Jenkins interface to this list of issues.

*FASTER FEEDBACK LOOPS  
FASTER MEAN TIME TO REPAIR  
LOWER DEFECT ESCAPE RATE*





**NowSecure™**  
All-in-one SAST+DAST+IAST

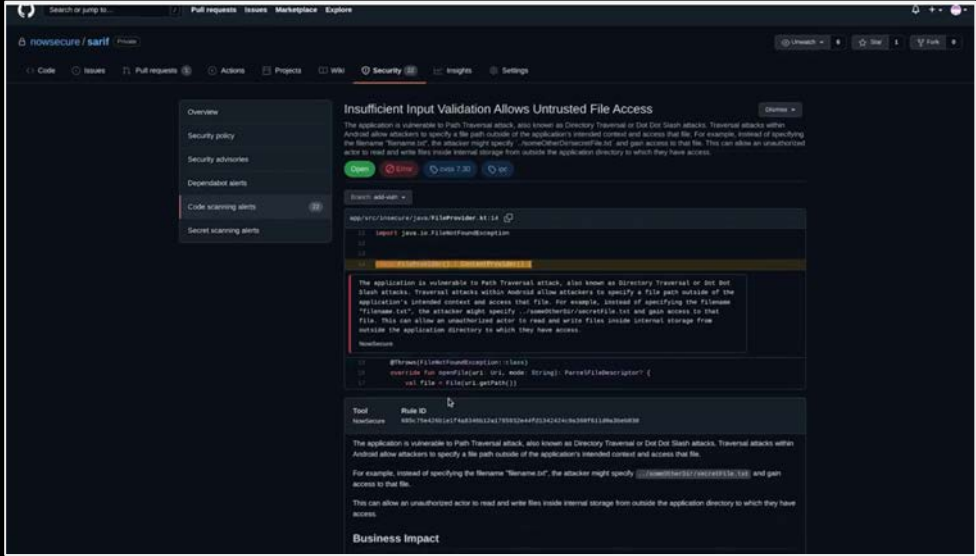
Security test any iOS/Android app binary, any language

Supports Kotlin, Java, Swift, ObjectiveC & More

Analyzes all code including 3rd party SDKs & transitive dependencies

Tickets w/ embedded dev guide & sample code to fix fast

# NowSecure GitHub Action for Developer-First Mobile AppSec Testing



NowSecure GitHub Action Resources  
[Demo Video Link](#)   [GitHub Action Link](#)



# THANK YOU!

OWASP Meetup

Brian Reed, Chief Mobility Officer

[br@nowsecure.com](mailto:br@nowsecure.com)

[@reed\\_on\\_the\\_run](https://twitter.com/reed_on_the_run)

