Insiders Guide to Mobile AppSec with OWASP MASVS

OWASP Meetup

Brian Reed, Chief Mobility Officer
br@nowsecure.com
@reed_on_the_run

NowSecure

NowSecure

12 years in Mobile Security
OWASP Sponsor & Contributor
Mobile AppSec Testing Tools, Training, Pen Testing
Creators of Frida and Radare

Brian Reed
Chief Mobility Officer

br@nowsecure.com
@reed_on_the_run

NowSecure

15+ Years in Mobile

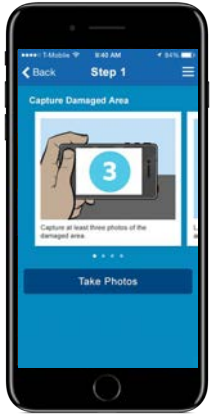
Remember when BlackBerry ruled the world? Now I live on iOS, Droid, Apple Watch, Oura....

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

OWASP Mobile Project Financial Sponsor & Contributor
NowSecure Security Researcher Carlos Holguera (@grepharder) is co-project lead for OWASP Mobile Project

OWASP CycloneDX SBOM Contributor
NowSecure Founder Andrew Hoog on the CycloneDX leadership board

Mobile Innovation



First Mobile Accident Claim

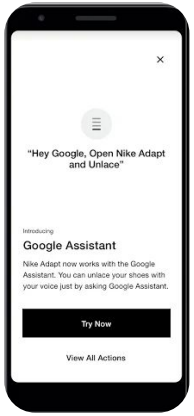
Multiple awards, Move up to #3 USA Insurer

First AR Furniture Shopping

Over 70% of millennials use feature, then shop in store

First Interactive Order Delivery Tracking

Mobile returned business to #1 pizza and drives >75% of all transactions



First Mobile Fit, First Mobile IOT

Dominant category brand, continuous innovation

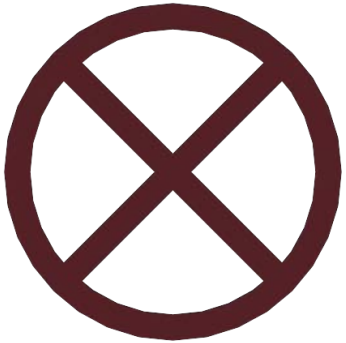
What is most often the biggest challenge or frustration teams face with developing & delivering secure apps?

Unpredictable
Security
Release Blockers

NowSecure

What’s going to block your next release?

A. Apple or Google App Store Dev Requirements
B. Internal Security Testing
C. Internal Governance Requirements
D. Feature Delays
E. Bug Fix Delays
F. All of the above?



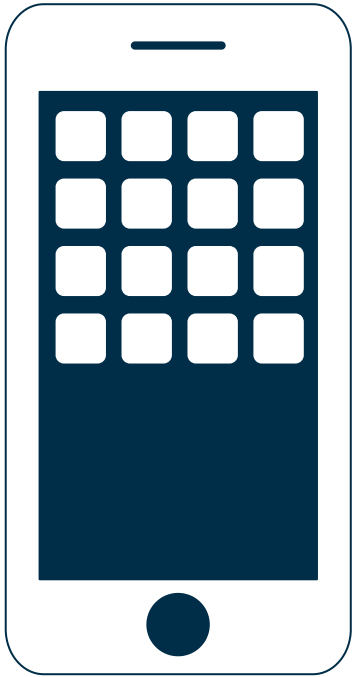
Mobile Powers the World, But Mobile Risk is Pervasive

69%

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

90%

Spike in Q2 2020 mobile app downloads YoY due to pandemic



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

NowSecure

What Mobile Apps Do You Use?

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

Peloton Responsible Disclosure from NowSecure

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

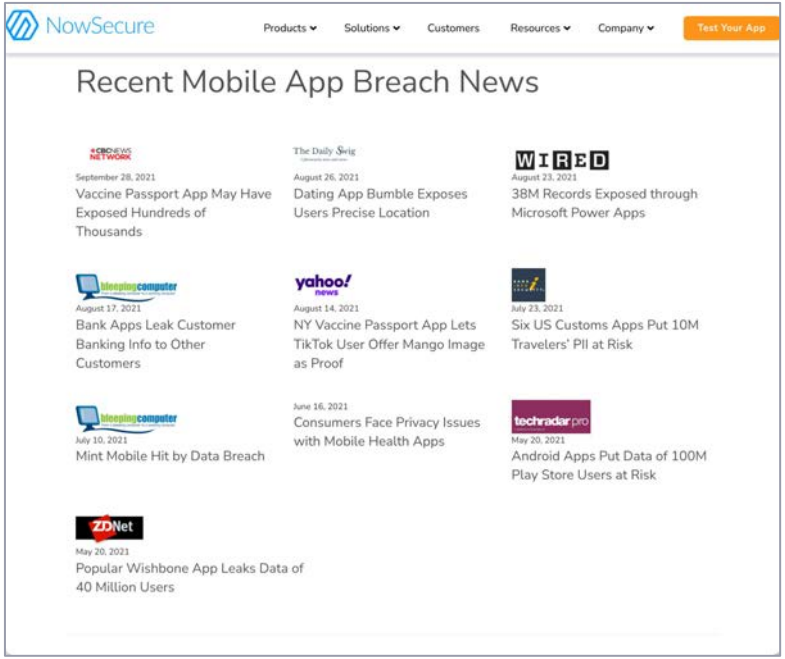
1. Peloton user exposure to account takeover
2. Peloton user exposure to phishing attack
3. Remote access to Peloton users' private personal info
4. Ability to remotely change device ID and serial number

There is NO evidence any customers were breached

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

NowSecure

Benchmark Trackers to Learn More

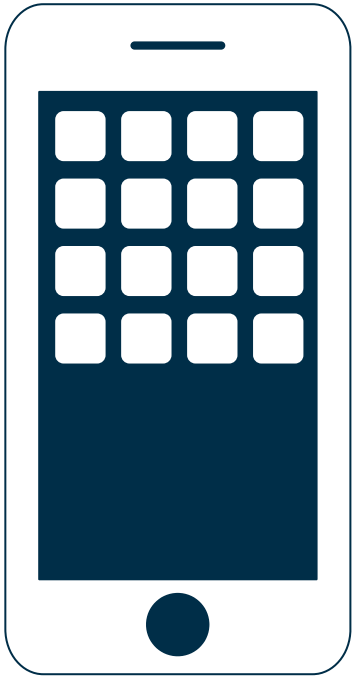


https://mobilerisktracker.nowsecure.com

https://bit.ly/ns-breachtracker

Inside Mobile AppSec

Unique Characteristics of Mobile AppDev & AppSec



Two different mobile OS with varying security capabilities

4 Dev Languages, Dozens of Frameworks, Thousands of libraries

Mobile OS and Dev tools update yearly

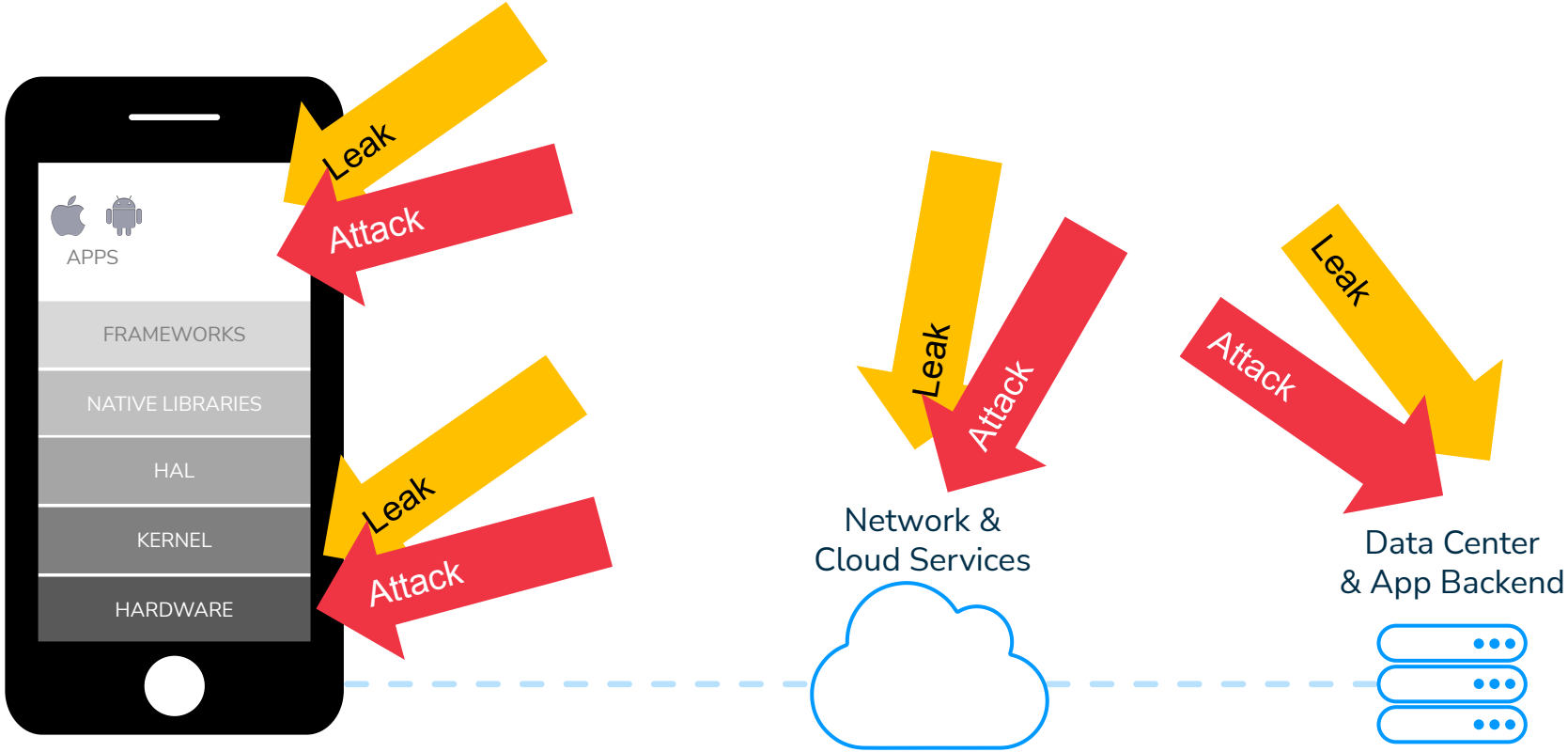
Mobile apps run on unprotected device, not behind web firewall

Effective testing requires physical devices, not emulators

Dynamic & APISec testing are challenging, but can be automated

OWASP MAVS is here to help!

Mobile Attack Surface



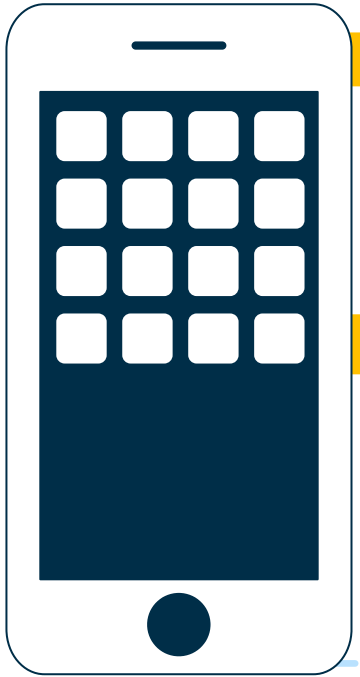
NowSecure

What's Inside the Mobile Attack Surface?

WEB VS MOBILE

98% of code behind perimeter with broad layered protection

Substantial code "in the wild" on uncontrolled OS & easily reversible



Code Functionality

- GPS spoofing
- Buffer overflow
- allowBackup Flag
- allowDebug Flag
- Code Obfuscation
- Configuration manipulation
- Escalated privileges
- Insecure 3rd party libs
- URL schemes
- GPS Leaking
- Integrity/tampering/repacking
- Side channel attacks
- App signing key unprotected
- JSON-RPC
- Automatic Reference Counting
- Media/file format parsers

Runtime

- Dynamic runtime injection
- Unintended permissions
- User-initiated code
- UI overlay/pin stealing
- Intent hijacking
- Zip directory traversal
- Clipboard data
- World Writable/Readable Files

API Backends

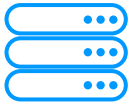
- Unauthenticated APIs
- Unprotected APIs
- Insecure URLs
- Excessive API Data
- API SQL Injection
- Remote code execution
- Privilege Escalation
- Denial of Service

Data at Rest

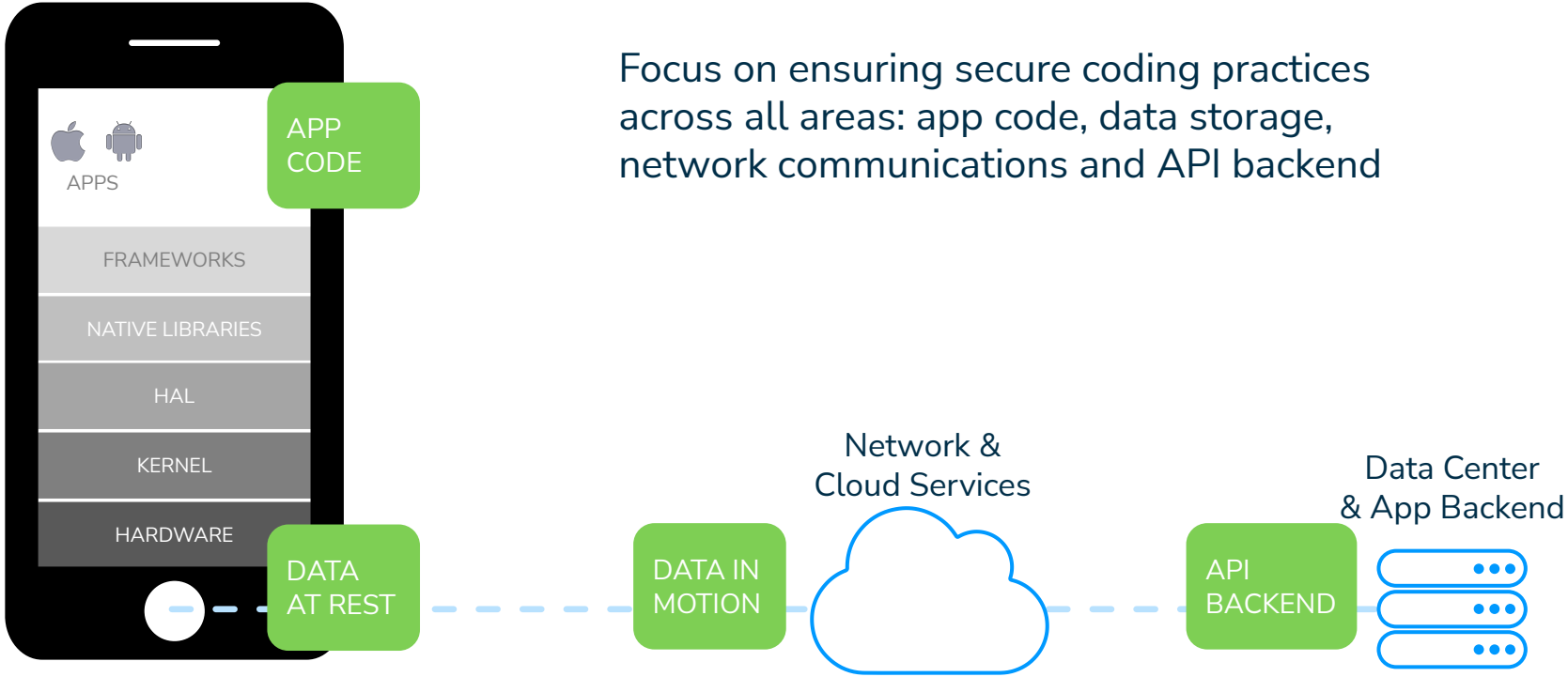
- Data caching
- Data stored in app directory
- Decryption of keychain
- Data stored in log files
- Data cached in memory/RAM
- Data stored in SD card
- Android rooting/iOS jailbreak
- OS data caching
- Passwords & data accessible
- No/Weak encryption
- TEE/Secure Enclave Processor
- Side channel leak
- SQLite database
- Emulator variance

Data in Motion

- Wi-Fi (no/weak encryption)
- Rogue access point
- Packet sniffing
- Man-in-the-middle
- Session hijacking
- TLS Downgrade
- Fake TLS certificate
- Improper TLS validation
- HTTP Proxies
- Weak VPNs
- Weak/No Local authentication
- App transport security
- Transmitted to insecure server
- Zip files in transit
- Cookie "httpOnly" flag
- Cookie "secure" flag

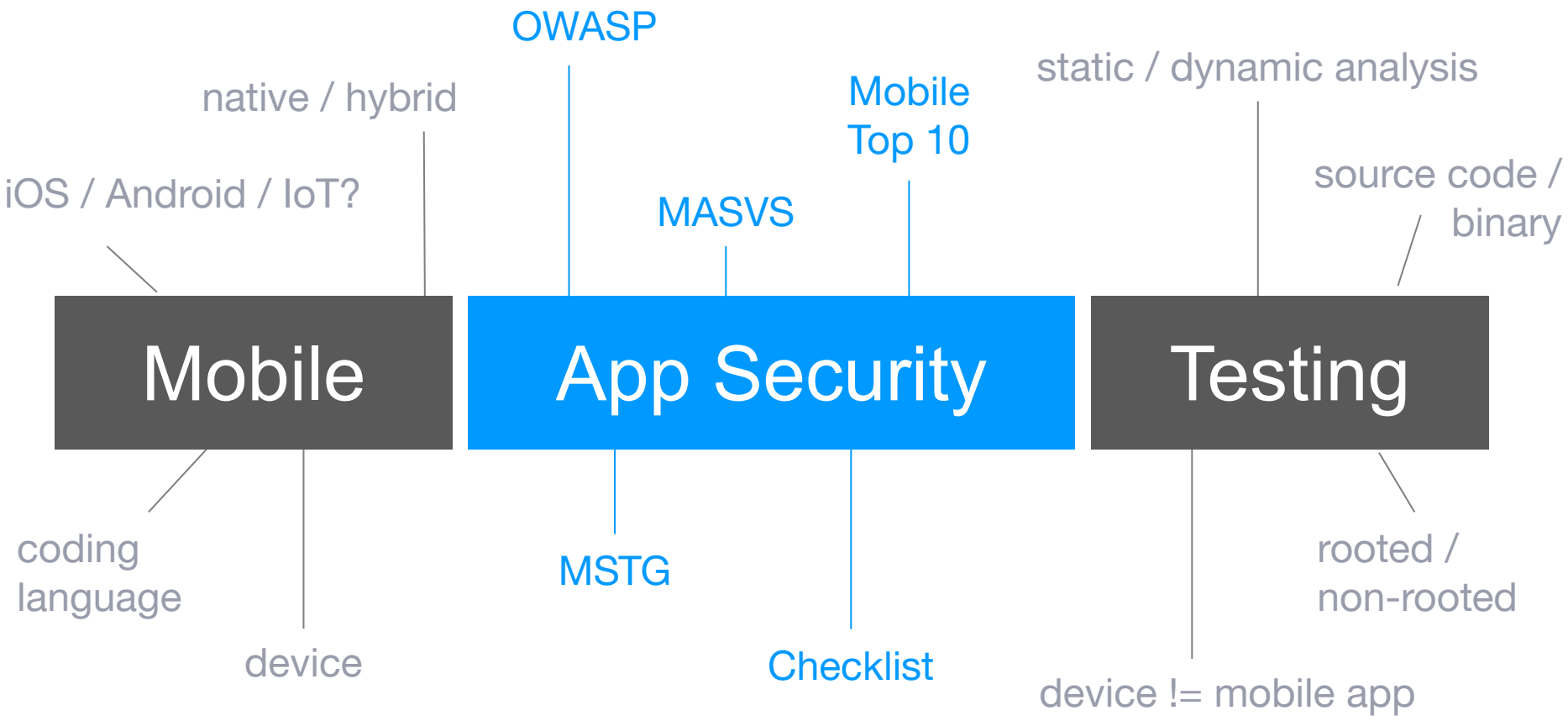


Reduce the Attack Surface



Focus on ensuring secure coding practices across all areas: app code, data storage, network communications and API backend

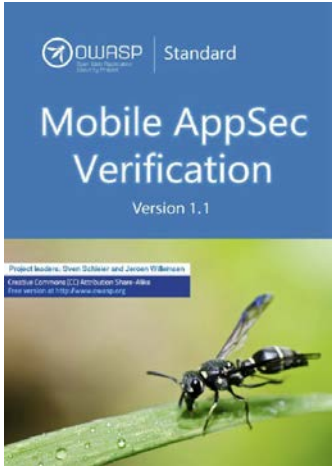
Mobile App Security Testing



OWASP Mobile Security Project Resources

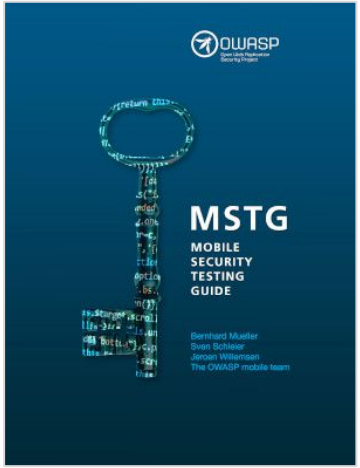
[Mobile Top 10]

Big issues in Mobile App Development



[Mobile App Security Verification Standards MASVS]

Establish security baseline for mobile apps



[Mobile Security Testing Guide MSTG]

Cookbook for mobile app security testing

[Mobile Security Testing Checklist]

Requirements for mobile app security testing

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

MASVS Key Differences

L1 expects standard security best practices to be met

L2 expects deeper defense
- Hardened against “Lost device” scenario
- Certificate Pinning
- Multi-factor authentication
- Corp policy for Architecture and Risk controls

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

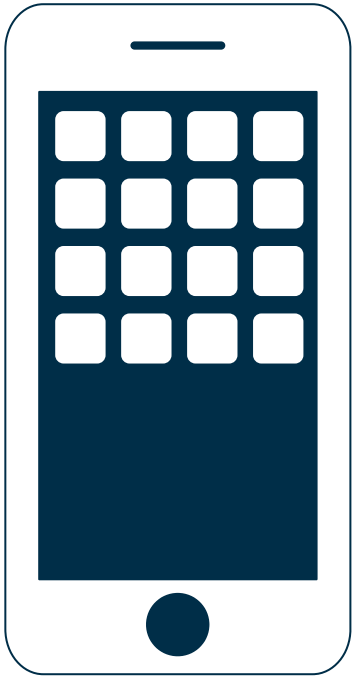
MASVS L2 + R
Defense-in-Depth + High RE Resilience

- 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

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

8 Domains of MASVS



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: Environmental Interaction Requirements

V7: Code Quality and Build Setting Requirements

V8: Resiliency Against Reverse Engineering Requirements

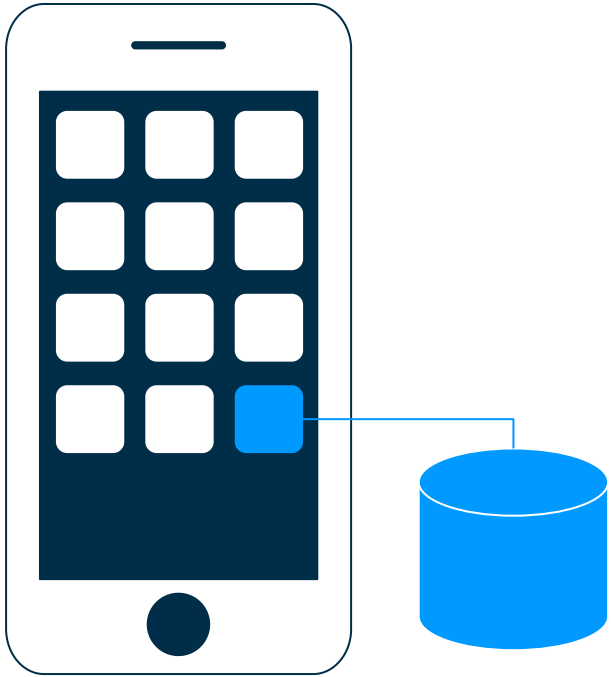
Top 5 Areas To Focus OWASP MASVS

1 - Data Stored in an Insecure Location

Data Stored in an Insecure, Exposed Location

50%

of Mobile Apps have insecure data storage



NowSecure

Data Stored in an Insecure Location

OWASP MASVS Mapping

- V2: Data Storage & Privacy
- V3: Cryptography

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

Resources:

- OWASP MASVS V2: Insecure Data Storage
- OWASP MASVS V3: Cryptography
- Android: Data and file storage overview
- Apple: File system basics

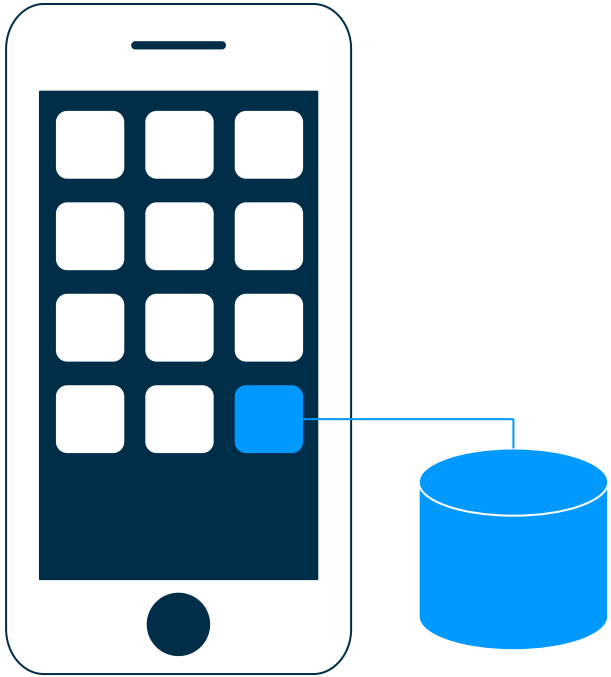
Data Stored in an Insecure, Exposed Location

Best Practices for Secure Coding

- Avoid writing sensitive data to device
- Encrypt sensitive files
- Use Android Scoped Storage
- Avoid query strings in sensitive data
- Implement secure data storage

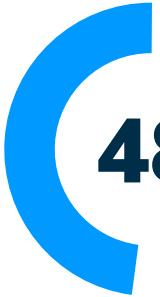
Best Practices for AppSec Testing

- 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 use of Android Scoped Storage



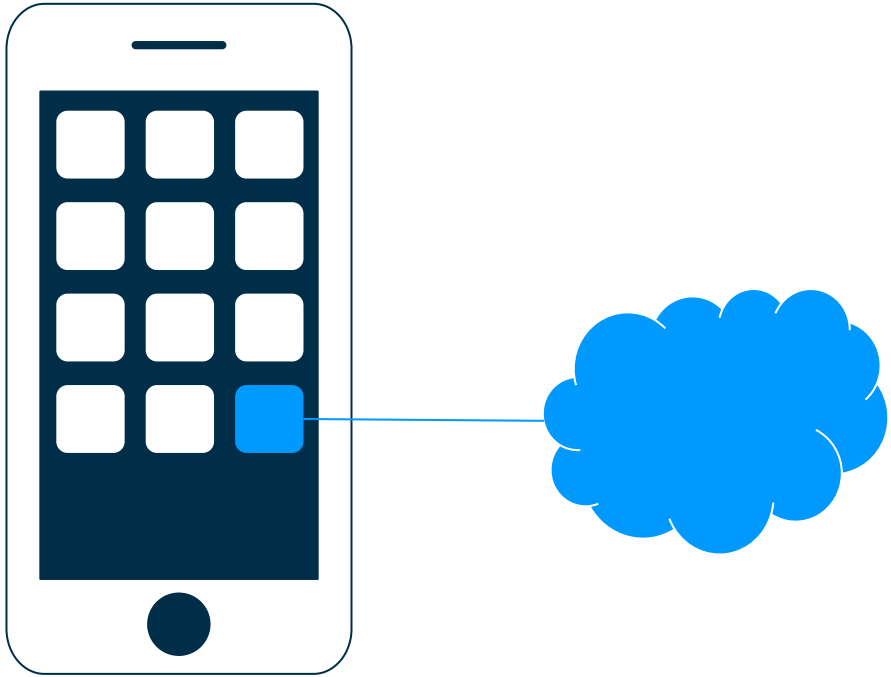
2- Improperly Coded Network Calls

Improperly Coded Network Calls



48%

of Mobile Apps have insecure data communication



NowSecure

Improperly Coded Network Calls

OWASP MASVS Mapping

- V5: Network Communication

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

Resources:

- OWASP MASVS V5: Network Comms
- Android: Network security configuration
- Apple: Preventing insecure network connection

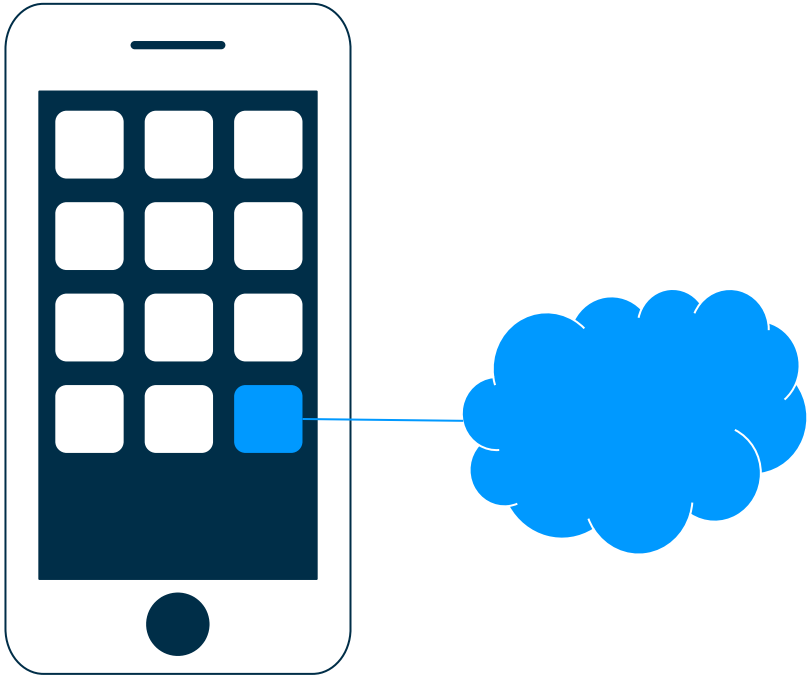
Improperly Coded Network Calls

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

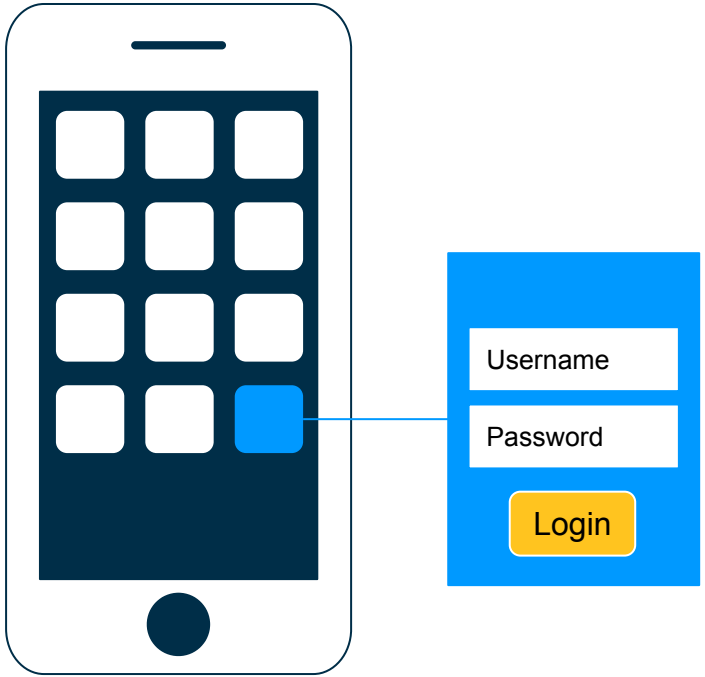


3- Insecure Authentication or Authorization

Insecure Authentication or Authorization

14%

of Mobile Apps have insecure authentication



Insecure Authentication or Authorization

OWASP MASVS Mapping

- V4: Authentication & Session Mgmt

Security bug:	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)
Attack vector:	API endpoints, stolen device
Business impact:	Unauthorized access, theft, and reputational risk

Resources:

- OWASP MASVS V4: Auth & Session Mgmt
- Android: Authenticate Users
- Apple: User Authentication

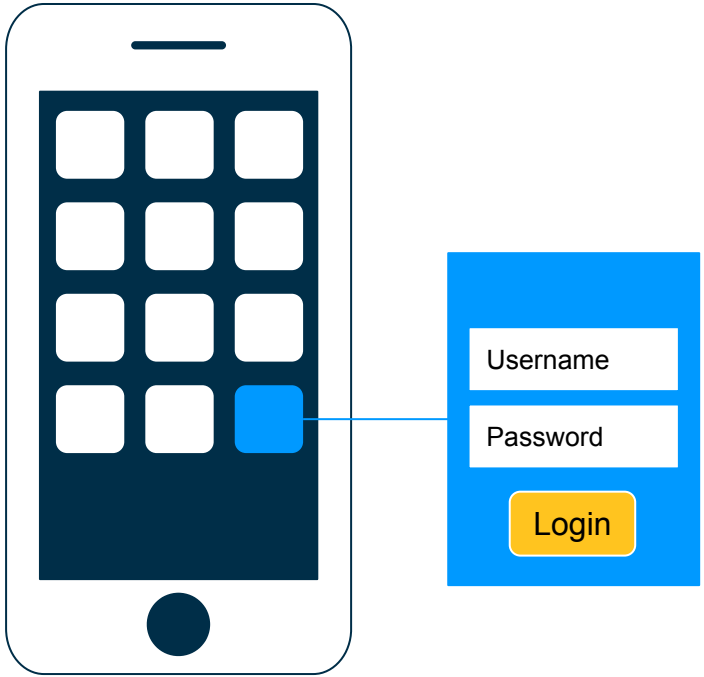
Insecure Authentication or Authorization

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

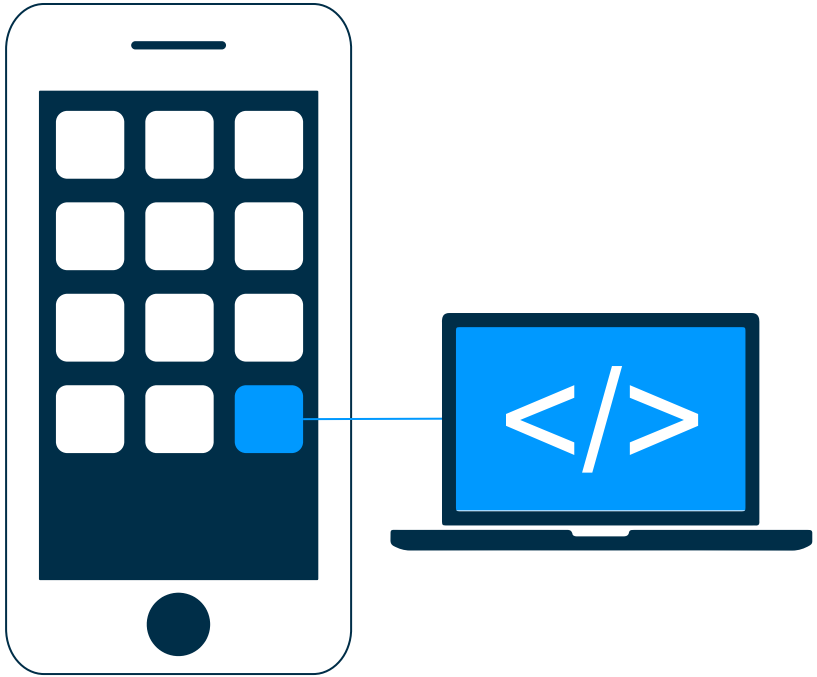
Insecure Coding Practices

47%

of Mobile Apps have insecure exploitable extraneous functionality

OWASP MASVS Mapping

- V6: Environmental Interaction
- V7: Code Quality & Build Setting



Insecure Coding Practices

OWASP MASVS Mapping

- V7: Code Quality & Build Setting Requirements

Resources:

- [OWASP MASVS V7: Code Quality]

Security bug:	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)
Attack vector:	Malware, phishing, unsuspected user, extraneous func. feature
Business impact:	Data theft, reputational risk, fraud, unauthorized access

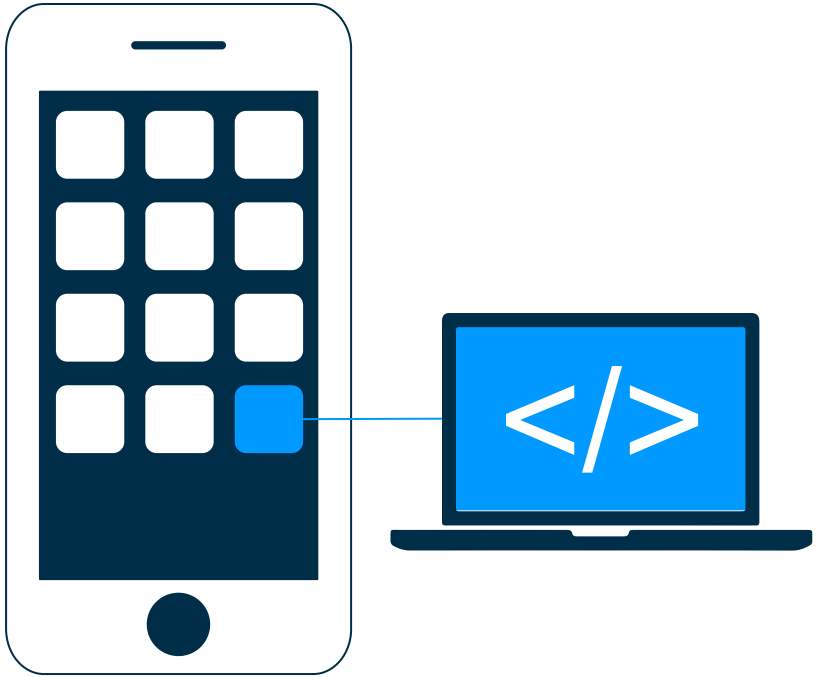
Insecure Coding Practices

Best Practices for Secure Coding

- Ensure Crypto meets minimum standards using SHA3 or greater
- Use dynamic values (not static) such as SecureRandom
- Use Key with a length of at least 2048 bits (preferably 4096 bits)

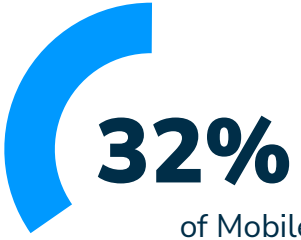
Best Practices for AppSec Testing

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



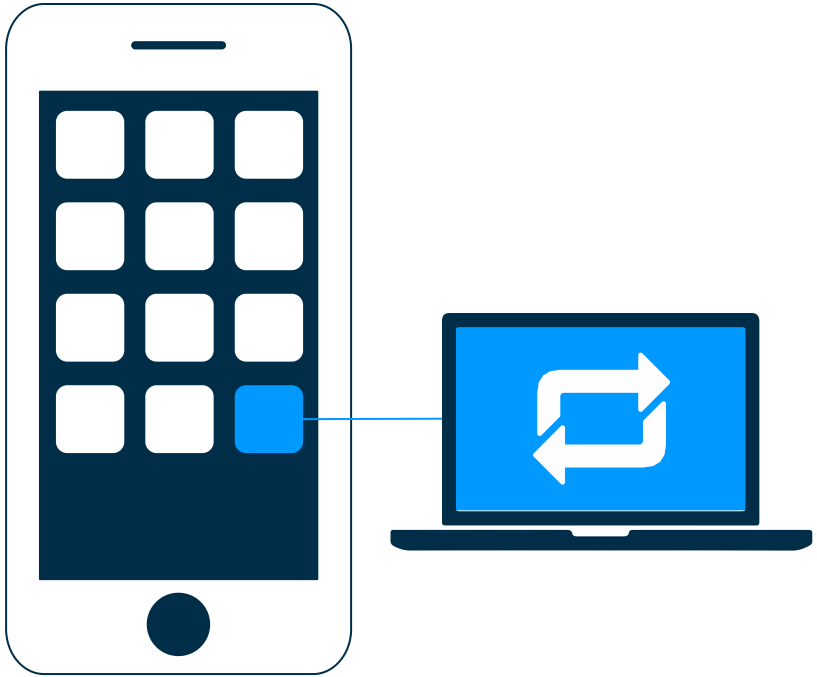
5- Reverse Engineering & Anti-Tampering

Exposure to Reverse Engineering



32%

of Mobile Apps have exposure to reverse engineering



NowSecure

Reverse Engineering

OWASP MASVS Mapping

- V8: Resiliency Against Reverse Engineering & Tampering

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

Resources:

- OWASP MASVS V8: Resiliency
- OWASP Reversing Prevention Project
- Reversing tools: Frida, Radare, 2Frida Repo

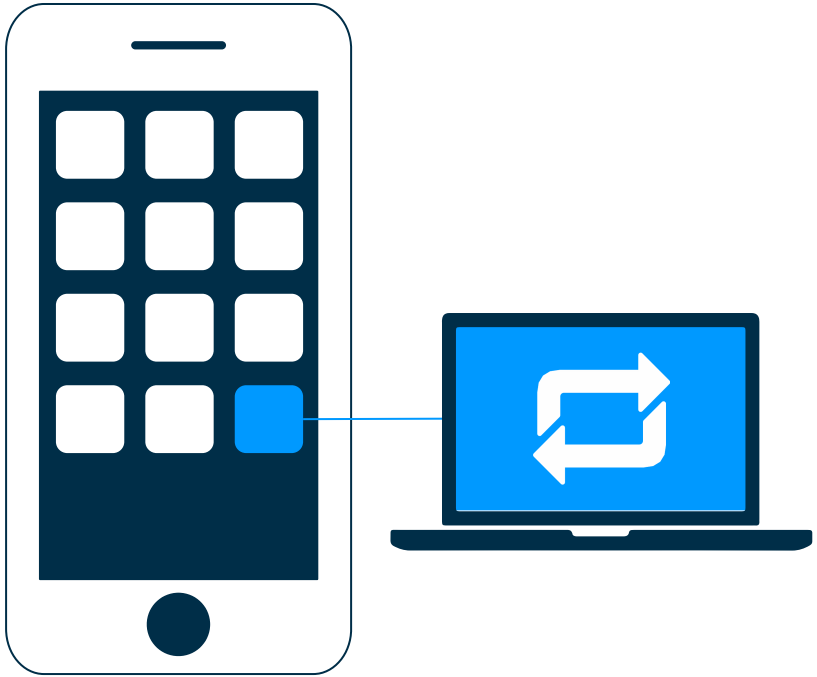
Exposure to Reverse Engineering

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

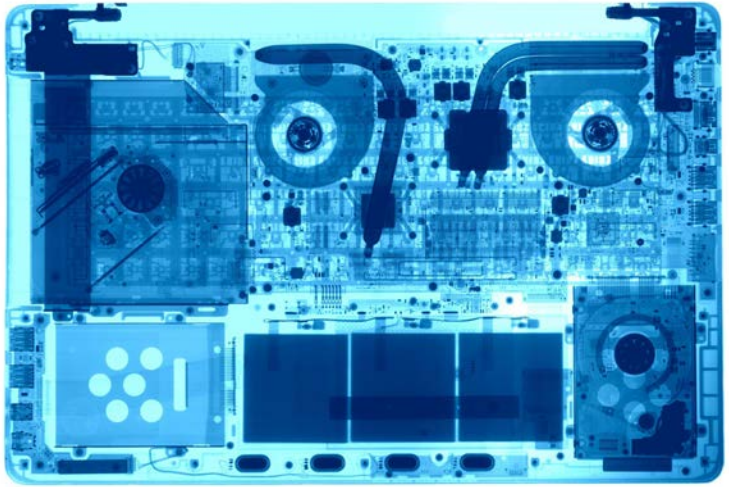
Best Practices for AppSec Testing

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



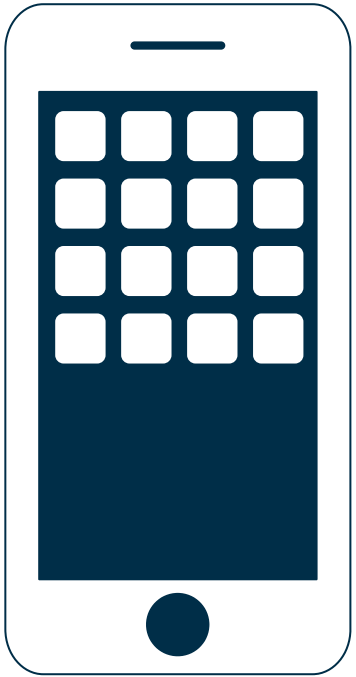
Tamper proofing helps, but only so far…

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



NowSecure

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 & Industry Updates

OWASP Mobile Security Testing Guide
@OWASP_MSTG

RELEASE @owasp MASVS v1.4.0 is out 🎉

Mobile apps must be transparent about the sensitive user data they collect and share, without exceptions. We promoted MSTG-STORAGE-12 to be L1 to reflect this.

🚀🥷 #privacy boost

github.com/OWASP/owasp-ma...

#MobileSecurity

New Release v1.4.0

v1.4.0

What's Changed

Changes in MASVS Requirements

- MSTG-STORAGE-12 is now L1 and L2 by @cpholguera in...

2 Contributors

OWASP Mobile Security Testing Guide
@OWASP_MSTG

The slides and recording from our latest talk at #NSConnect21 are available, enjoy 😊

🖼️ drive.google.com/file/d/18-SMID...

▶️ events.bizzabo.com/NSConnect21/ag...

MASVS & M
Refactoring

drive.google.com
NSConnect - MASVS & MSTG Refactoring.pdf

Store Mac iPad iPhone Watch TV Music Support

About privacy information on the App Store and the choices you have to control your data

The App Store now includes detailed privacy information that helps you understand each app's data collection practices.

In June 2020, Apple announced a new privacy information section for product pages on the App Store. This is the beginning of an innovative new programme to help customers have more transparency and understanding about what data apps may gather about them. This new programme creates an easy-to-understand system for all apps, where the information is self-reported by the developer. Apple will continue to provide resources to developers to help them fill in this information accurately. This privacy information section will evolve over time as we all learn what works best for everyone.

About the privacy information section

The new privacy information section helps you understand an app's privacy practices on any Apple platform. On each app's product page, you can find out about some of the data types the app may collect, and whether that data is linked to you or used to track you.

Find out how the App Privacy section defines the different types of data an app may collect – including location, contact info, health info and more – and some of the ways in which the developer or its third-party partners may use it, e.g. for advertising or analytics.

Data Linked to You

Data that is listed as linked to you means that the data is collected in a way that is linked to your identity, such as to your account, your device or your details (for example, your phone number). To declare that

developers Platform Android Studio Google Play More

Android Developers Blog

The latest Android and Google Play news for app and game developers.

Preparing for Google Play's new safety section

28 July 2021

Posted by Suzanne Frey, VP, Product, Android Security and Privacy

Today, we're announcing additional details for the upcoming safety section in Google Play. At Google, we know that feeling safe online comes from using products that are secure by default, private by design, and give users control over their data. This new safety section will provide developers a simple way to showcase their app's overall safety. Developers will be able to give users deeper insight into their privacy and security practices, as well as explain the data the app may collect and why — all before users install the app.

Ultimately, all Google Play store apps will be required to share information in the safety section. We want to give developers plenty of time to adapt to these changes, so we're sharing more information about the data type definitions, user journey, and policy requirements of this new feature.

Search blog...

Labels

Archive

Feed

Newsletter

Android Developers

Google Play

NowSecure

OWASP Just Launched Bonus Track

What is OWASP CycloneDX?

OWASP PROJECTS CHAPTERS EVENTS ABOUT Search OWASP.org Store Donate Join

OWASP CycloneDX

Main Resources

CycloneDX is a lightweight Software Bill of Materials (SBOM) standard designed for use in application security contexts and supply chain component analysis.

The CycloneDX project provides standards in XML, JSON, and Protocol Buffers, as well as a large collection of official and community supported tools that create or interoperate with the standard.

The projects website has many documented use cases and examples that provide a springboard to SBOM adoption.

The project operates as a meritocracy whose guiding principals reinforce its risk-based approach to standards development. The project encourages community participation in the development of the standard and supporting tools.

Background

Modern software is assembled using third-party and open source components. They are glued together in

Watch 13 Star 88

The OWASP® Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

Project Information

Flagship Project

Classification

Standard

Audience

Builder
Defender

Project Resources

Website
GitHub
Slack - invite
Twitter

Licensing

Apache 2.0 License

Leaders

Patrick Dwyer
Steve Springett

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

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."

Link to Dependency Track SBOM tool
https://dependencytrack.org/

What is OWASP CycloneDX?

OWASP

PROJECTS CHAPTERS EVENTS ABOUT

Search OWASP.org | Store | Donate | Join

OWASP CycloneDX

Watch 13 | Star 88

Main | Resources

CycloneDX is a lightweight Software Bill of Materials (SBOM) standard designed for use in application security contexts and supply chain component analysis.

The CycloneDX project provides standards in XML, JSON, and Protocol Buffers, as well as a large collection of official and community supported tools that create or interoperate with the standard.

The projects website has many documented use cases and examples that provide a springboard to SBOM adoption.

The project operates as a meritocracy whose guiding principals reinforce its risk-based approach to standards development. The project encourages community participation in the development of the standard and supporting tools.

Background

Modern software is assembled using third-party and open source components. They are glued together in

The OWASP® Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

Project Information

Flagship Project

Classification

Standard

Audience

Builder
Defender

Project Resources

Website
GitHub
Slack - invite
Twitter

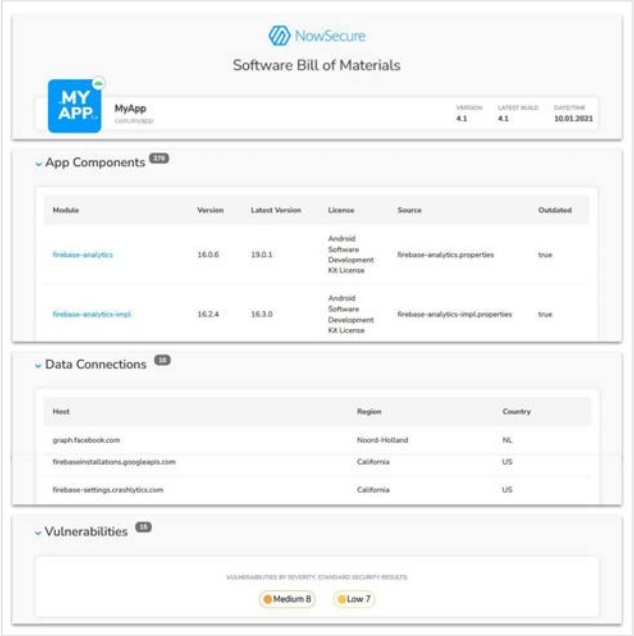
Licensing

Apache 2.0 License

Leaders

Patrick Dwyer
Steve Springett

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



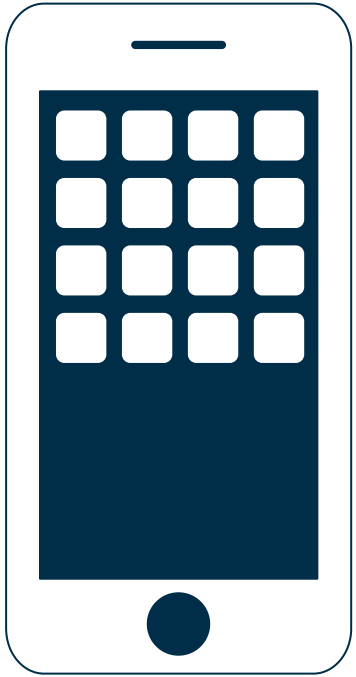
Get Free Mobile SBOMS
https://bit.ly/ns-SBOM10

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 or ZedAttackProxy
- Jailbroken & Rooted devices

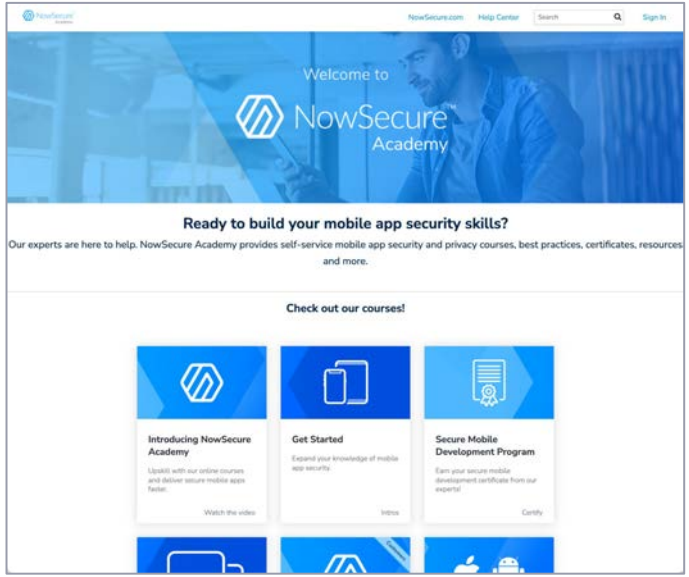


Automated Testing Resources

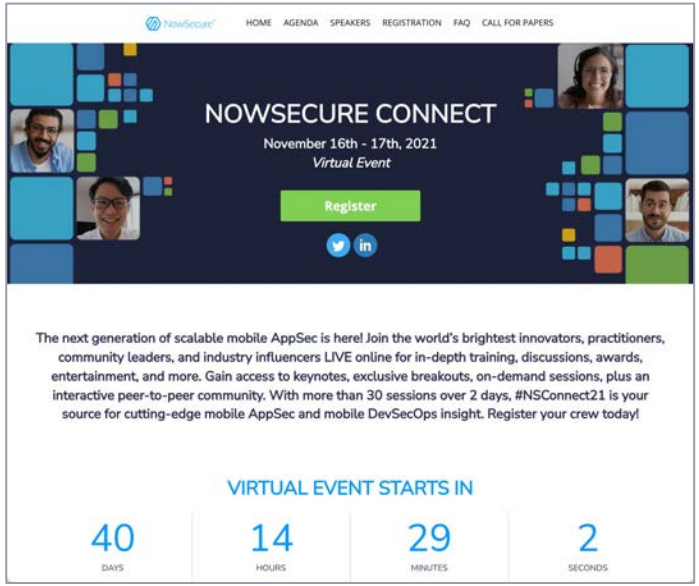
- Free Mobile SBOMs
- Free Mobile Analysis Report
- NowSecure Workstation Toolkit
- NowSecure Platform Automation

- Full mobile appsec testing
  - 600+ security, privacy and compliance tests
  - SAST+DAST+IAST+APISec
  - Automated & Interactive Modes
  - Embedded remediation

Free Training

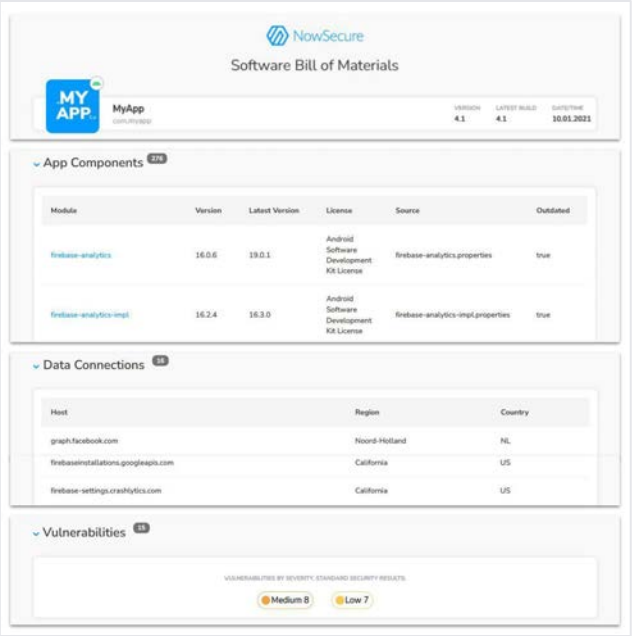


https://academy.nowsecure.com

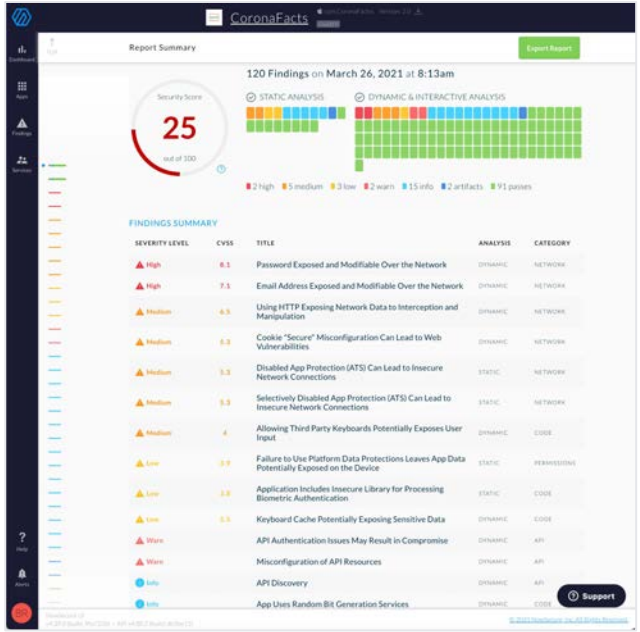


https://bit.ly/ns-connect

Checkout Your Own Mobile Apps



https://bit.ly/ns-SBOM10



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 CrackeMe 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 Services
Expert support, enterprise risk assessments & mobile programs

THANK YOU!

OWASP Meetup

Brian Reed, Chief Mobility Officer
br@nowsecure.com
@reed_on_the_run

NowSecure