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

FRIDA UNDER ARMOUR Microsoft Google Uber

citi iRobot TESLA

GitHub Medtronic AT&T Ford

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® + 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

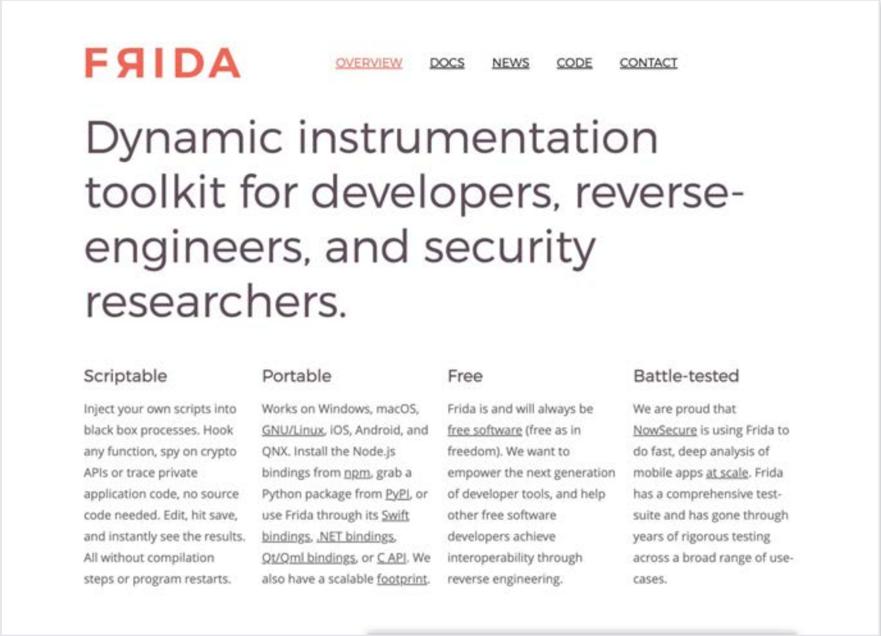
ioXt
internet of secure things
+ NowSecure

NowSecure IoXT Authorized Lab
Certify Mobile-Connected IoT devices

ADA + NowSecure

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

Open Source Community



FRIDA

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

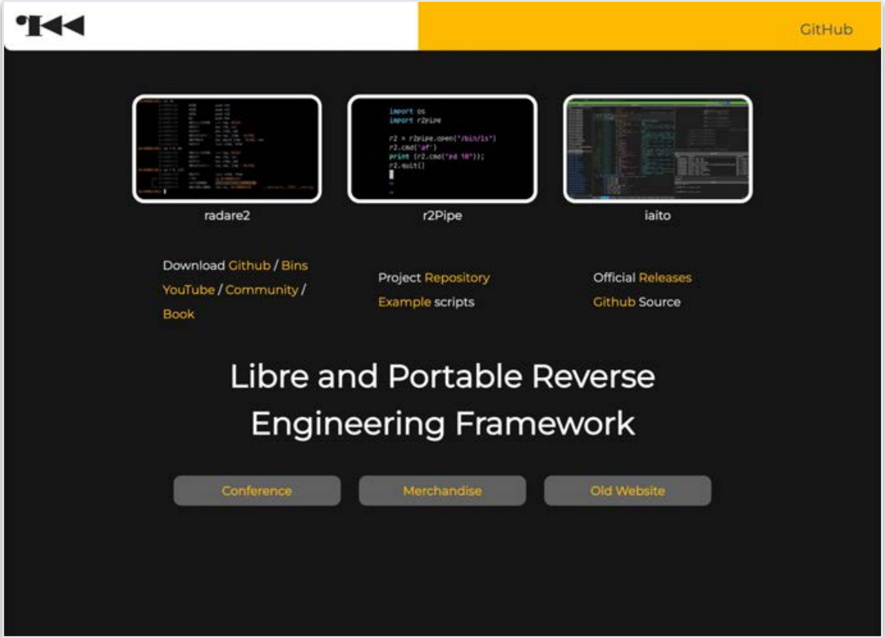
Works on Windows, macOS, GNU/Linux, iOS, Android, and QNX. Install the Node.js bindings from npm, grab a Python package from PyPI, or use Frida through its Swift bindings, .NET bindings, Qt/Qml bindings, or C API. We also have a scalable footprint.

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.



```
import os
import r2pipe

r2 = r2pipe.open("/bin/ls")
r2.cmd('af')
print (r2.cmd("pd 10"));
r2.quit()
```

Download Github / Bins
YouTube / Community / Book

Project Repository
Example scripts

Official Releases
Github Source

Libre and Portable Reverse Engineering Framework

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/

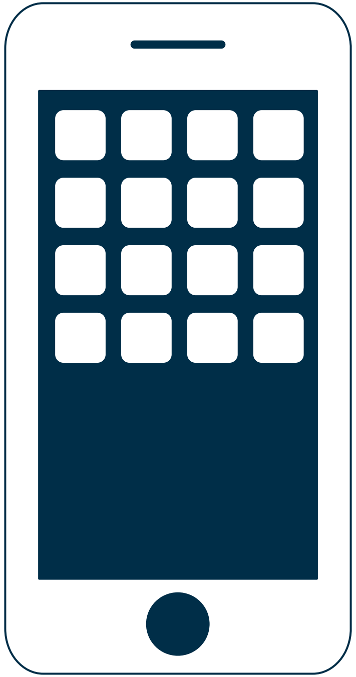
Mobile Powers the World, But Mobile Risk is Pervasive

69%

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

200bn

Mobile App Downloads in
2021



85%

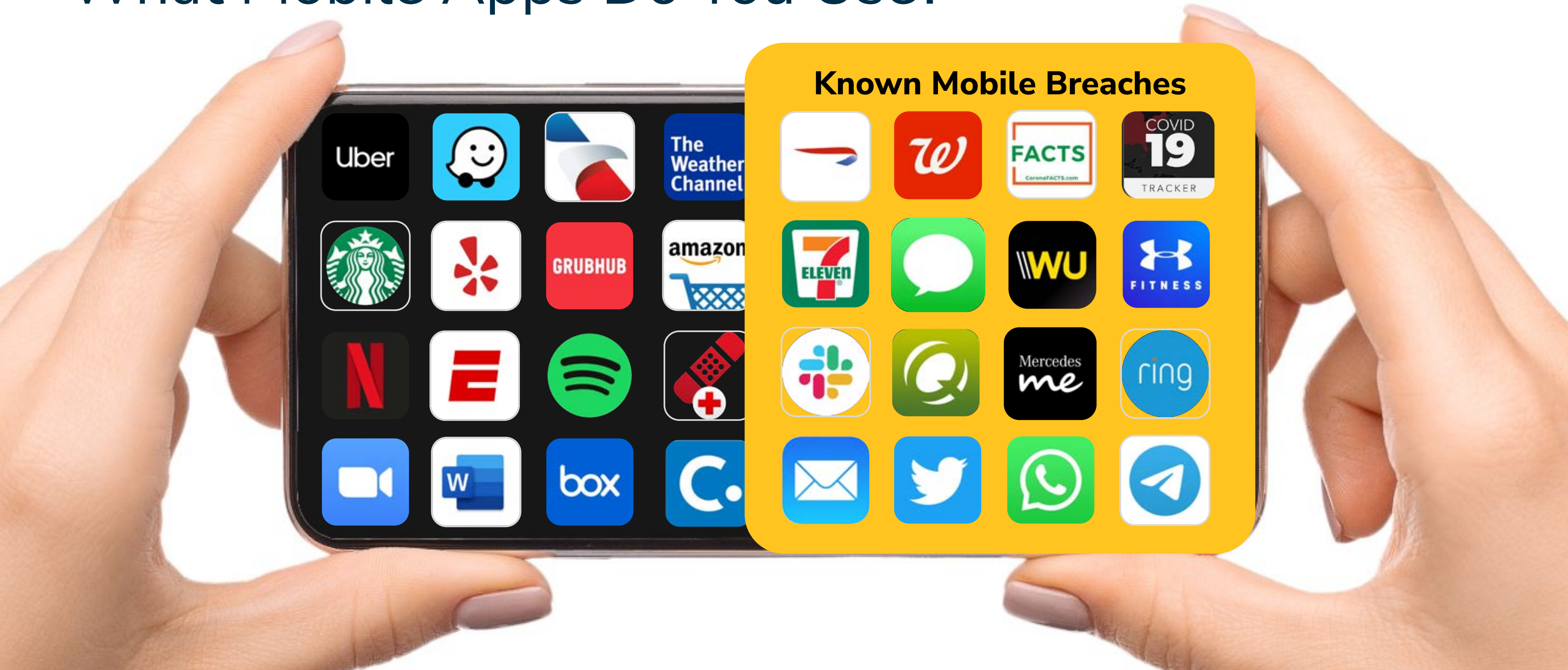
of Mobile Apps
have security risks
(Fail OWASP Mobile Top 10)

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

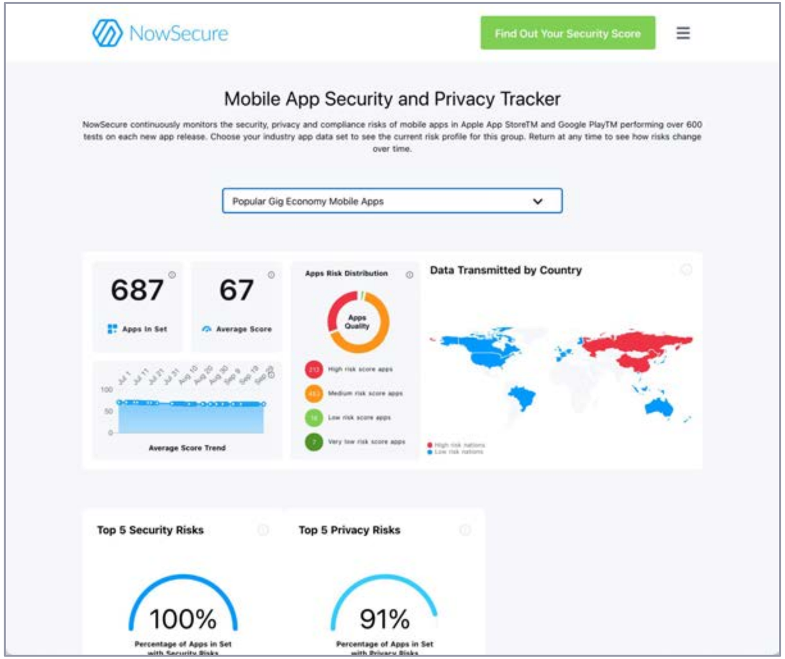
What Mobile Apps Do You Use?



Known Mobile Breaches

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

Benchmark Trackers to Learn More



Mobile App Security and Privacy Tracker

NowSecure continuously monitors the security, privacy and compliance risks of mobile apps in Apple App StoreTM and Google PlayTM performing over 600 tests on each new app release. Choose your industry app data set to see the current risk profile for this group. Return at any time to see how risks change over time.

Popular Gig Economy Mobile Apps

687 Apps In Set

67 Average Score

Average Score Trend

Apps Risk Distribution

- High risk score apps
- Medium risk score apps
- Low risk score apps
- Very low risk score apps

Data Transmitted by Country

Top 5 Security Risks

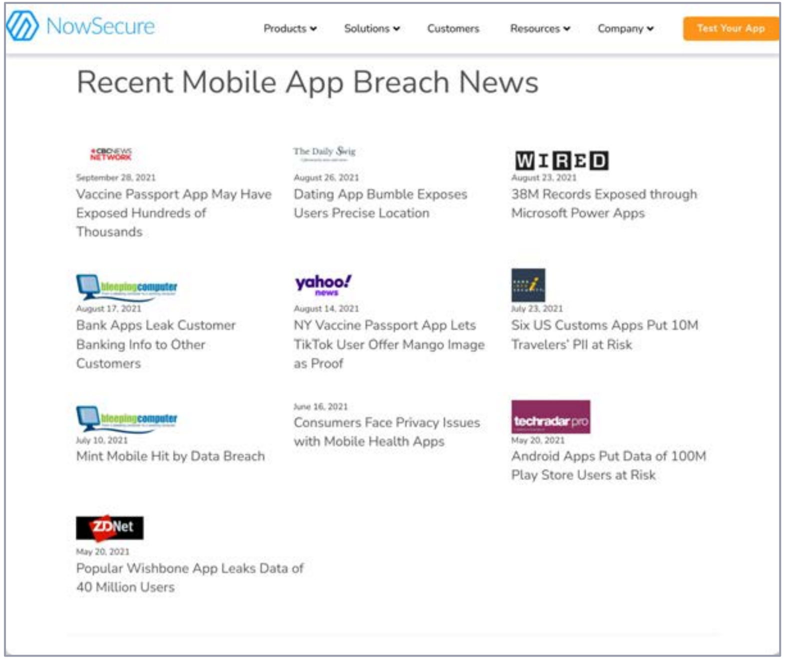
100%

Percentage of Apps in Set with Security Risks

Top 5 Privacy Risks

91%

Percentage of Apps in Set with Privacy Risks



Recent Mobile App Breach News

September 28, 2021
Vaccine Passport App May Have Exposed Hundreds of Thousands

August 26, 2021
Dating App Bumble Exposes Users Precise Location

August 23, 2021
38M Records Exposed through Microsoft Power Apps

August 17, 2021
Bank Apps Leak Customer Banking Info to Other Customers

August 14, 2021
NY Vaccine Passport App Lets TikTok User Offer Mango Image as Proof

July 23, 2021
Six US Customs Apps Put 10M Travelers' PII at Risk

July 10, 2021
Mint Mobile Hit by Data Breach

June 16, 2021
Consumers Face Privacy Issues with Mobile Health Apps

May 20, 2021
Android Apps Put Data of 100M Play Store Users at Risk

May 20, 2021
Popular Wishbone App Leaks Data of 40 Million Users

https://mobilerisktracker.nowsecure.com

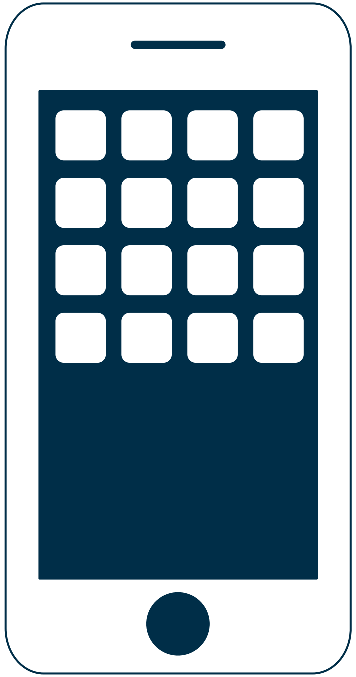
https://bit.ly/ns-breachtracker

Inside Mobile AppSec

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

NowSecure

Unique Characteristics of Mobile AppDev & AppSec

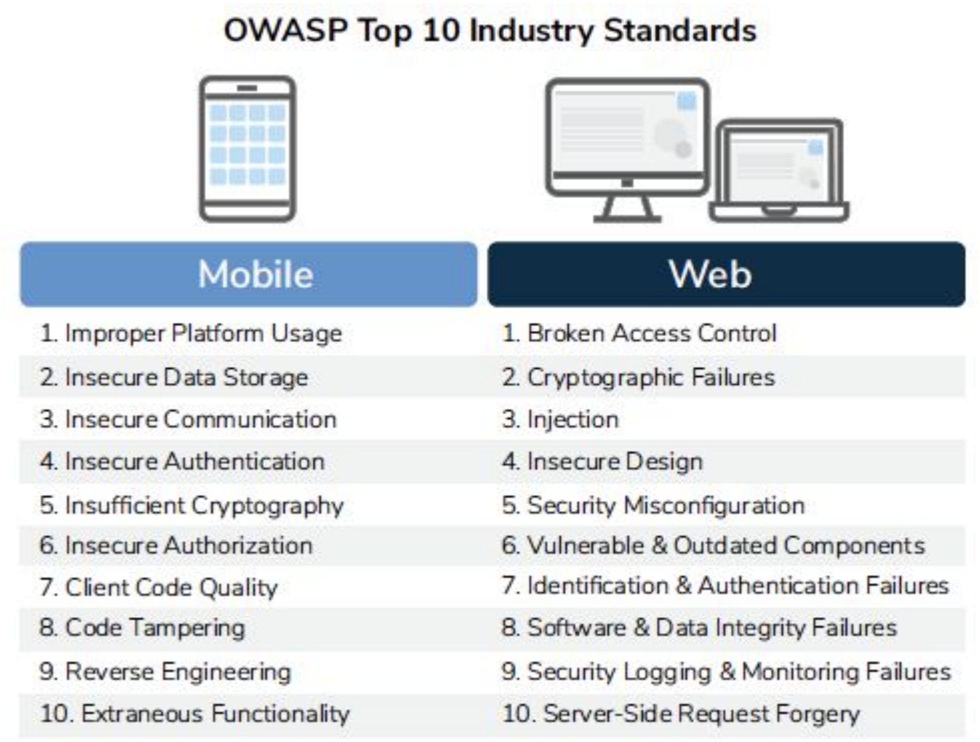


WEB	VS	MOBILE
98% of code behind perimeter with broad layered protection		Substantial code “in the wild”, running on untrusted device & easily reversible

- 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

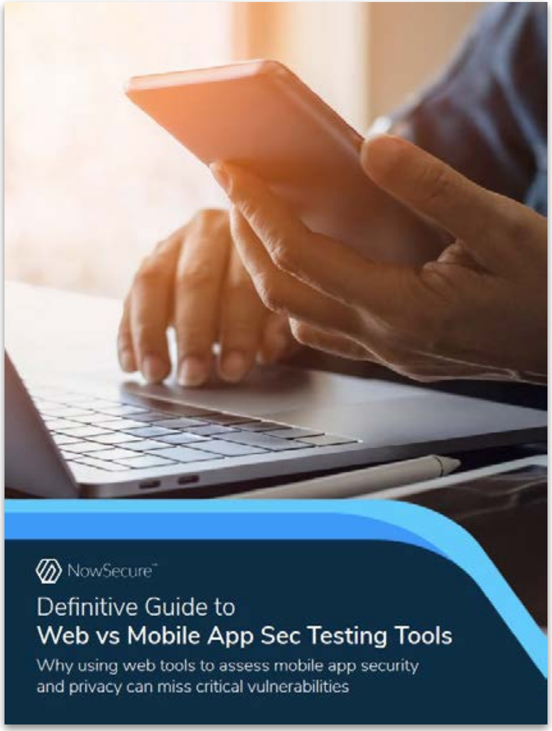
The OWASP MASVS is here to help!

NowSecure



OWASP Top 10 Industry Standards

Mobile	Web
1. Improper Platform Usage	1. Broken Access Control
2. Insecure Data Storage	2. Cryptographic Failures
3. Insecure Communication	3. Injection
4. Insecure Authentication	4. Insecure Design
5. Insufficient Cryptography	5. Security Misconfiguration
6. Insecure Authorization	6. Vulnerable & Outdated Components
7. Client Code Quality	7. Identification & Authentication Failures
8. Code Tampering	8. Software & Data Integrity Failures
9. Reverse Engineering	9. Security Logging & Monitoring Failures
10. Extraneous Functionality	10. Server-Side Request Forgery



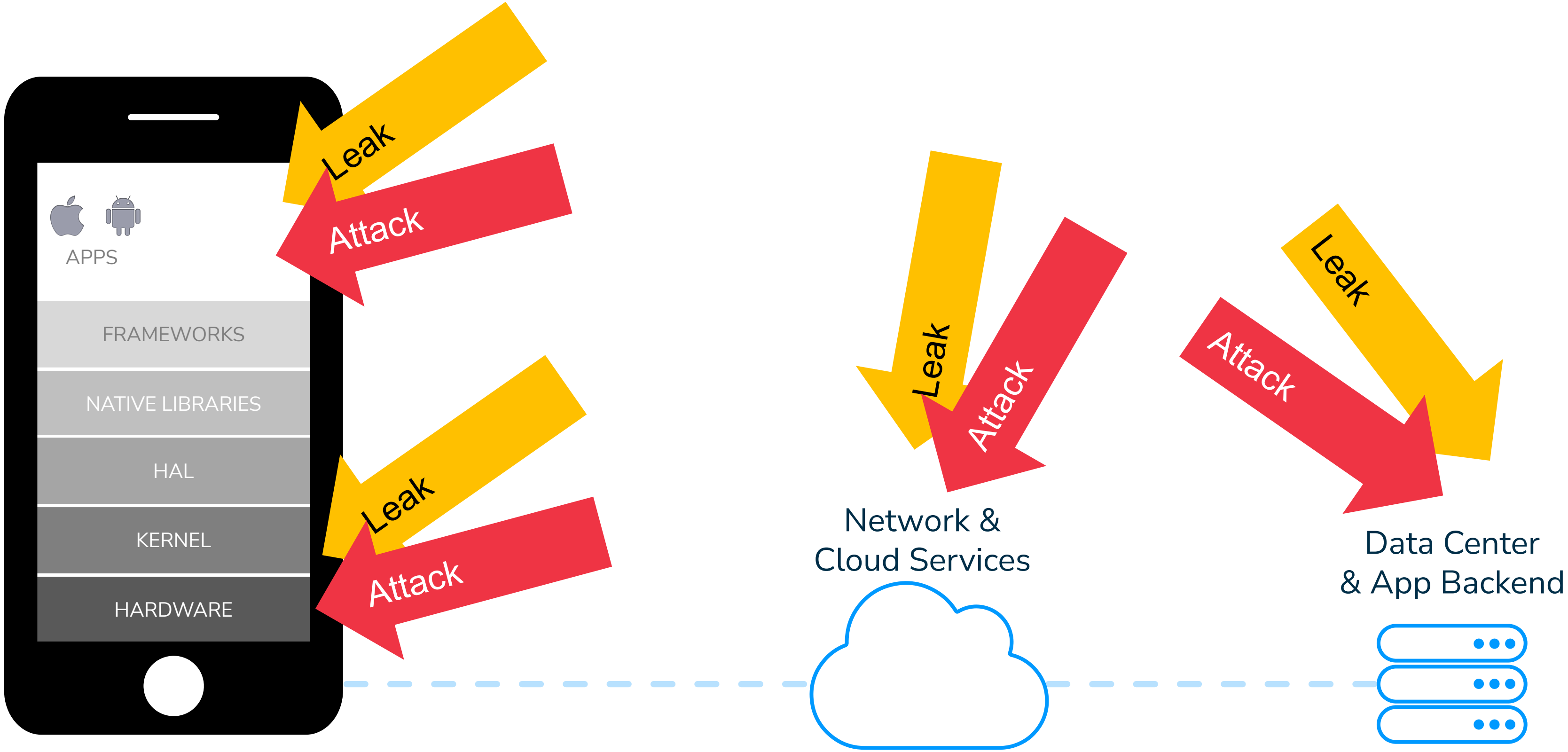
NowSecure™

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

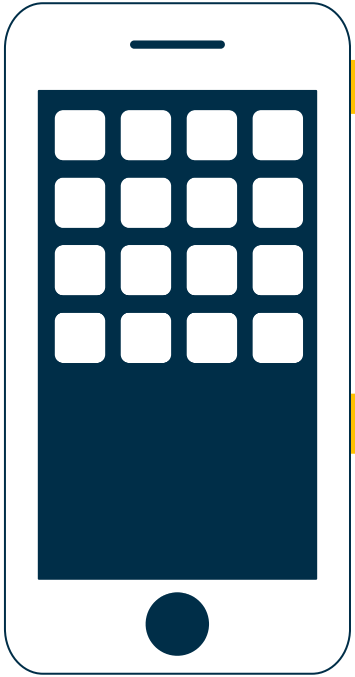
https://discover.nowsecure.com/nowsecure-ms/web-vs-mobile-app-security-testing-tools

Mobile Attack Surface



NowSecure

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

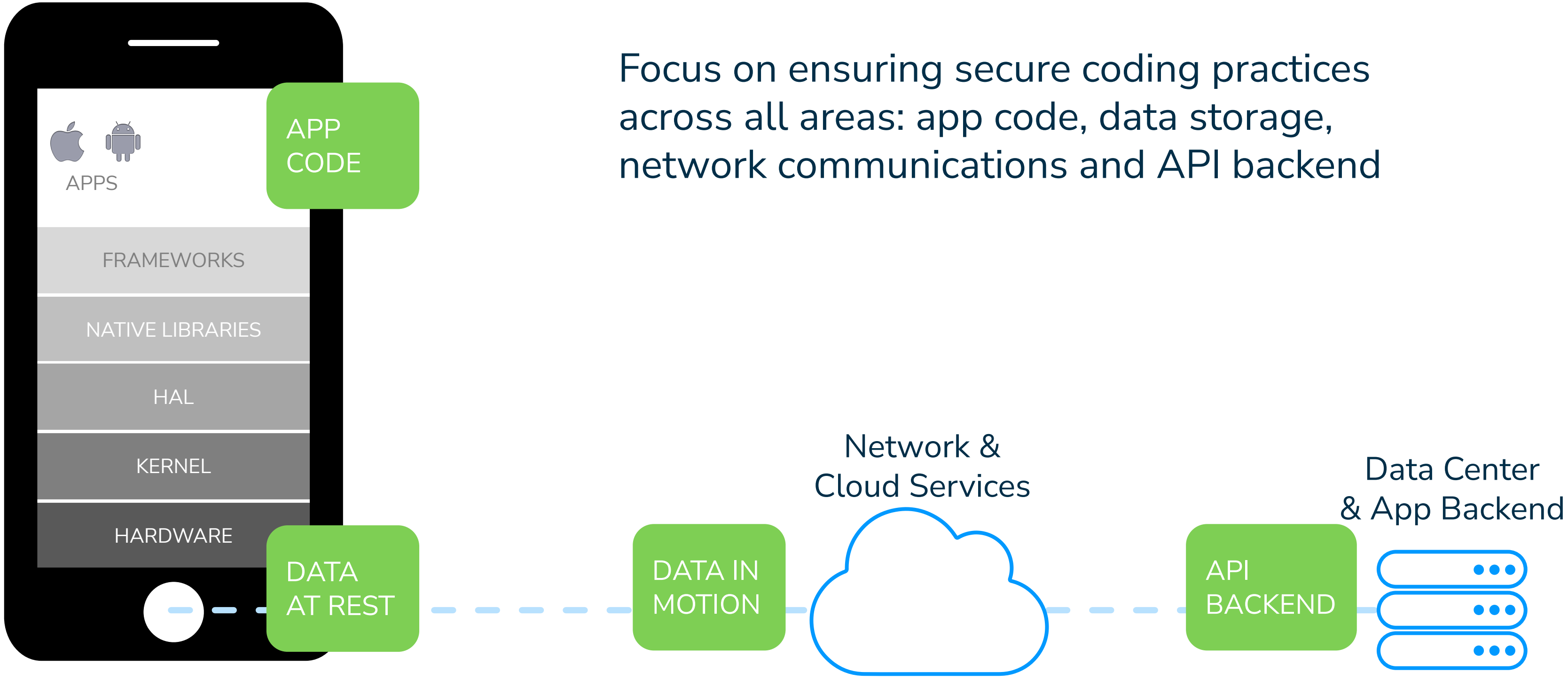
…

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

…

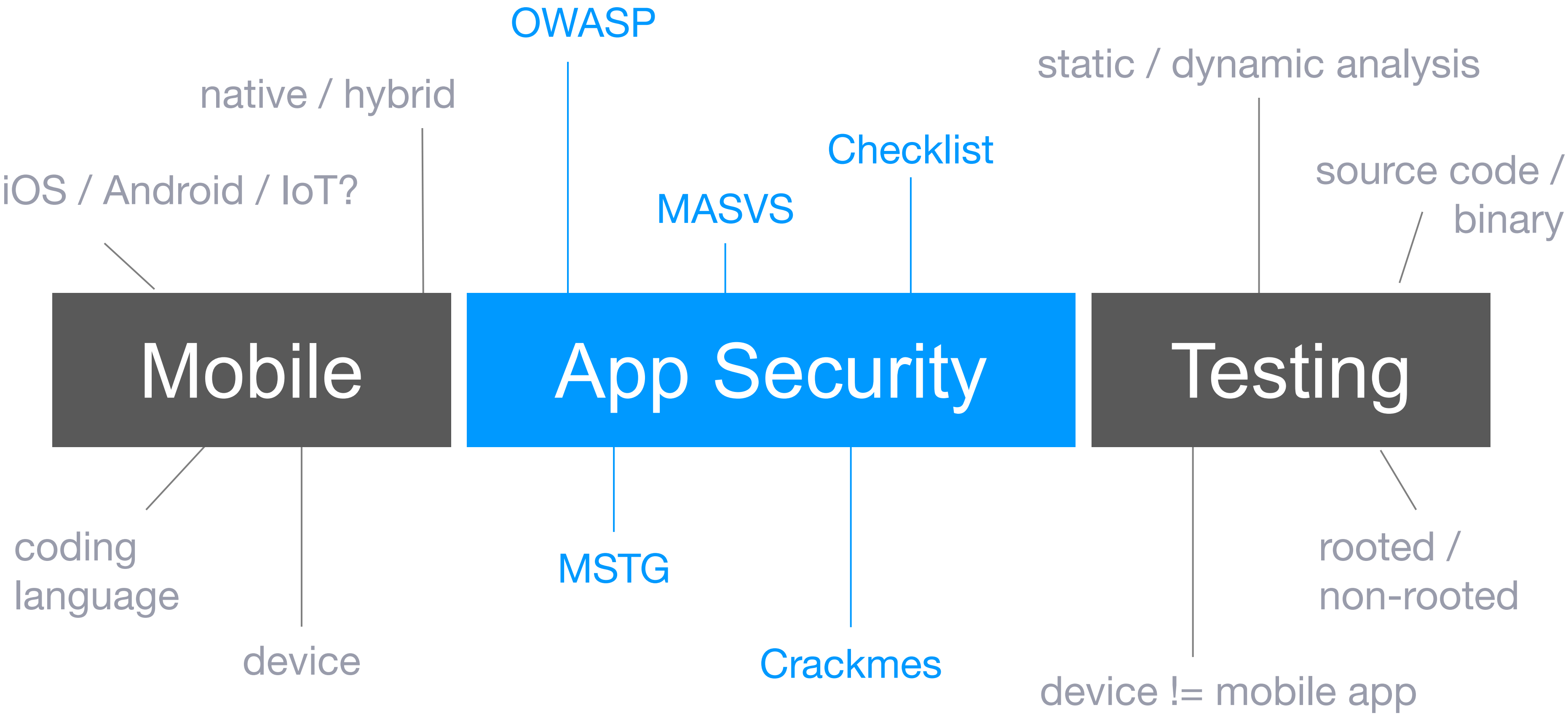
Reduce the Attack Surface to Protect Sensitive Data

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



NowSecure

Mobile App Security Testing



Mobile

iOS / Android / IoT?

native / hybrid

coding language

device

App Security

OWASP

MASVS

Checklist

MSTG

Crackmes

Testing

static / dynamic analysis

source code / binary

rooted / non-rooted

device != mobile app

NowSecure

NowSecure

OWASP Mobile Security Project Resources

MASVS

Mobile Application Security Verification Standard

Sven Schleier
Bernhard Mueller

Carlos Holguera
Jeroen Willemsen

OWASP

[Mobile App Security Verification Standard]()

Establish security baseline for mobile apps

Latest Release: 2022

MSTG

Mobile Security Testing Guide

Sven Schleier
Bernhard Mueller

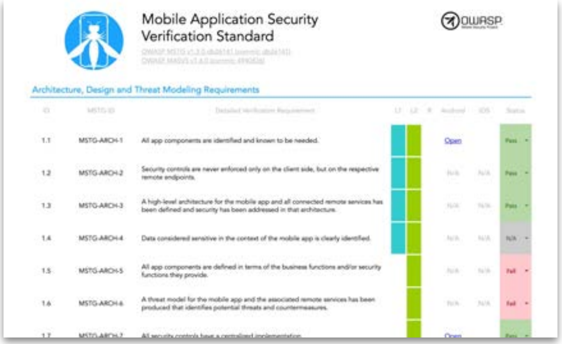
Carlos Holguera
Jeroen Willemsen

OWASP

[Mobile Security Testing Guide]()

Cookbook for mobile app security testing

Latest Release: 2022



Mobile Application Security
Verification Standard

OWASP MSTG v1.3.0 [illegible]
OWASP MASVS v1.4.0 [illegible]

Architecture, Design and Threat Modeling Requirements

ID	MSTG-ID	Detailed Verification Requirement	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.	🟦	🟩		N/A	N/A	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.	🟦	🟩		N/A	N/A	Pass
1.4	MSTG-ARCH-4	Data considered sensitive in the context of the mobile app is clearly identified.	🟦	🟩		N/A	N/A	N/A
1.5	MSTG-ARCH-5	All app components are defined in terms of the business functions and/or security functions they provide.		🟩		N/A	N/A	Fail
1.6	MSTG-ARCH-6	A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.		🟩		N/A	N/A	Fail
1.7	MSTG-ARCH-7	All security controls have a centralized 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**

L2 expects **defense-in-depth**

- 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

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

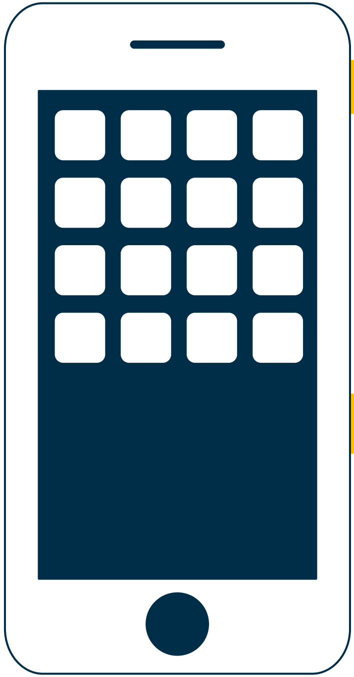
Example: Healthcare Drug Delivery App

R expects **hardening**

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

NowSecure

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

…

MASVS-ARCH

App Architecture

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

…

MASVS-PLATFORM

Data in Use

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

…

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

…

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-CRYPTO

MASVS-STORAGE

- 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

…

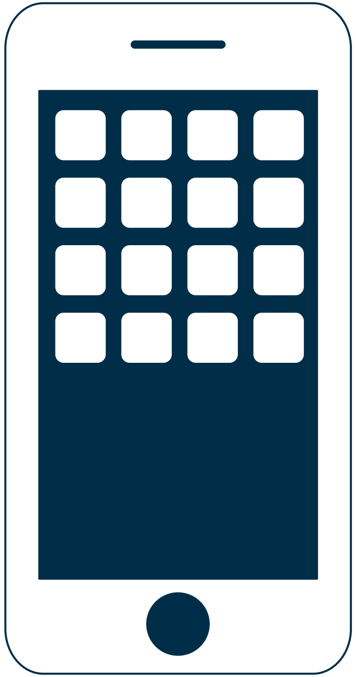
MASVS-AUTH

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

NowSecure

Top 5 Areas To Focus
OWASP MASVS

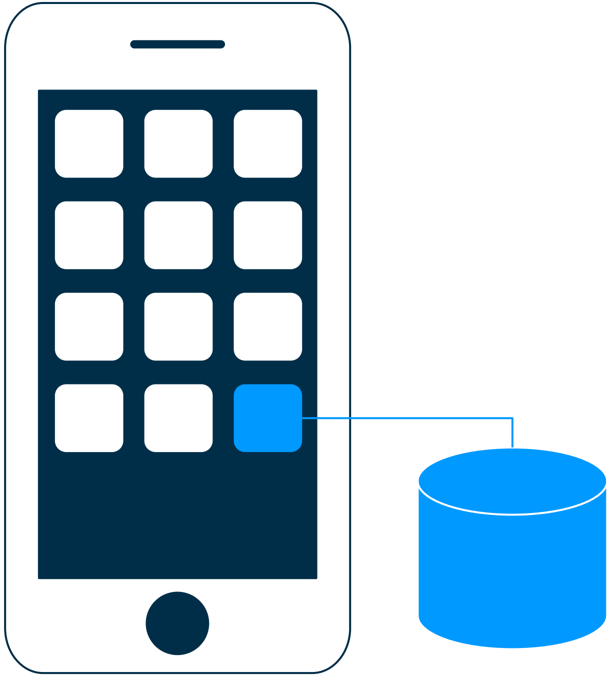
Let’s Use Both the Builder and Breaker POV

1 - Insecure Data Storage & Crypto

Insecure Data Storage & Crypto

50%

of Mobile Apps have insecure data storage



NowSecure

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

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

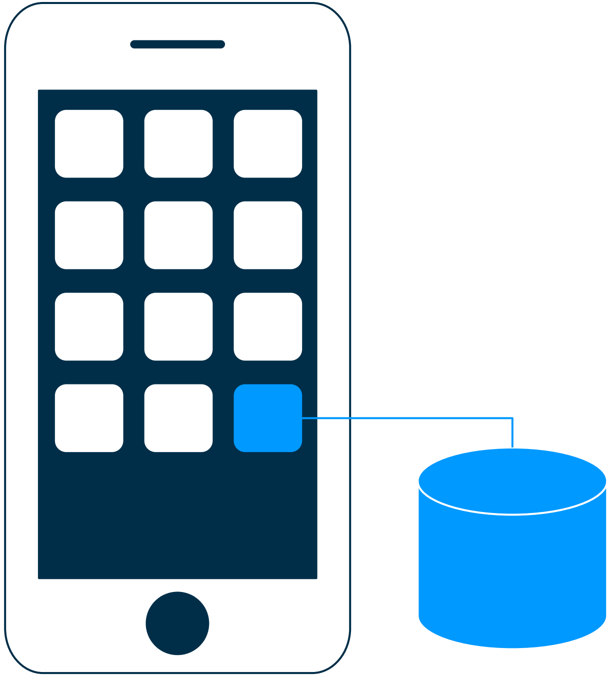
Insecure Data Storage & Weak Crypto

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

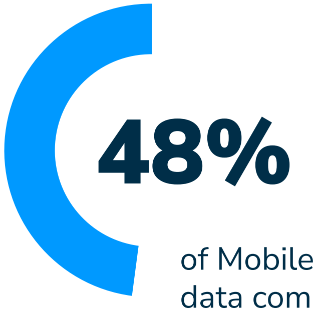
- Test for credentials & PII in files, logs, IPC
- Test for data removed when background
- Test Crypto libs & storage
- Confirm req use of device password
- Check for weak crypto & bad practices



NowSecure

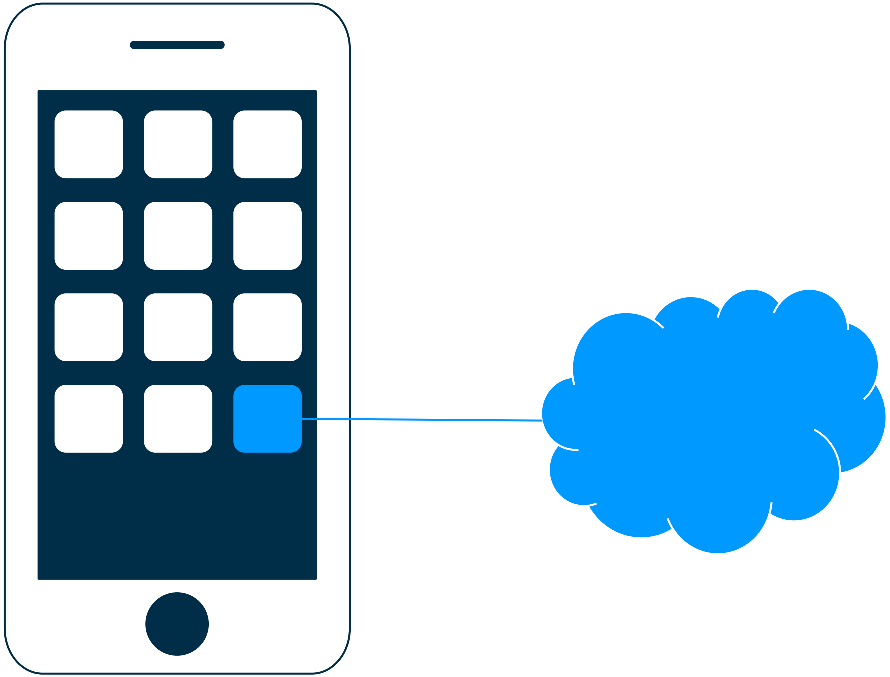
2- Insecure Network Communication

Insecure Network Communication



48%

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

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

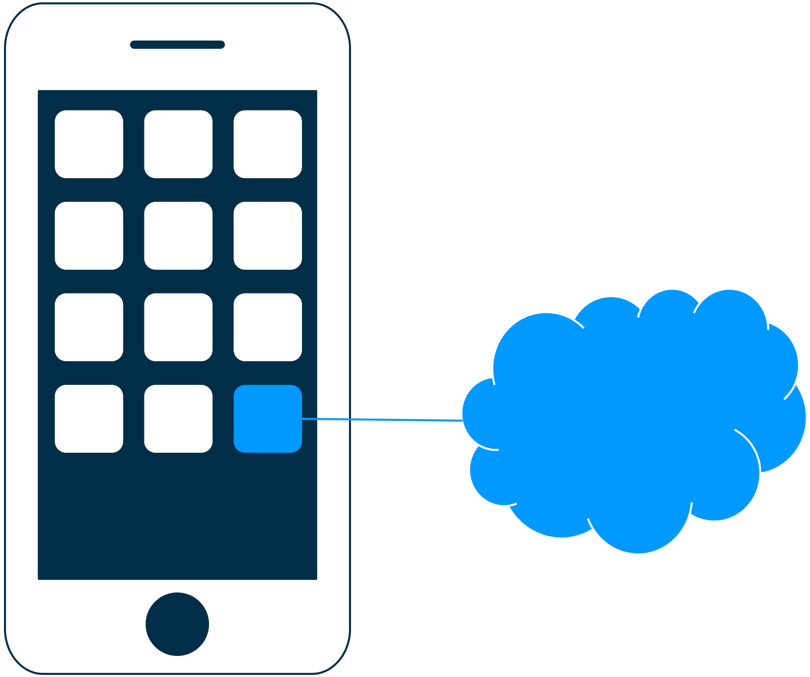
Improperly Coded Network Calls

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



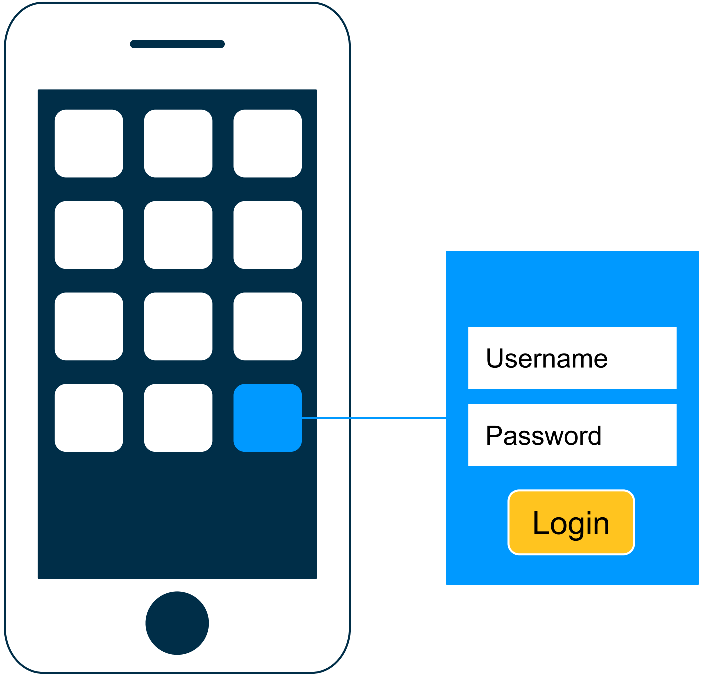
NowSecure

3- Insecure Authentication or Authorization

Insecure Authentication or Authorization

14%

of Mobile Apps have insecure authentication



NowSecure

Insecure Authentication or Authorization

OWASP MASVS Mapping

- V4: Authentication & Session Mgmt

Resources:

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

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

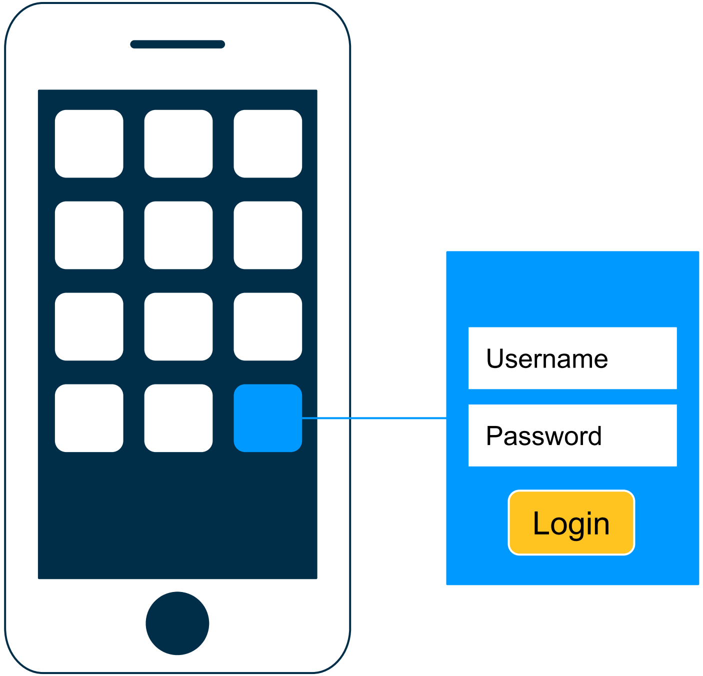
Insecure Authentication or Authorization

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

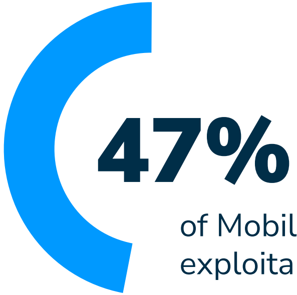
- Test session validation
- Test data in memory



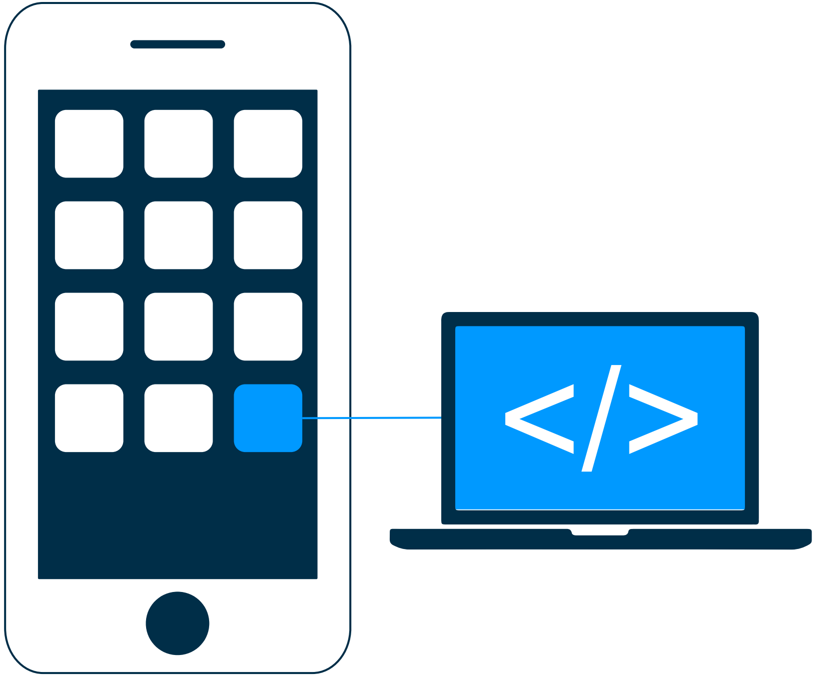
NowSecure

4- Insecure Coding Practices

Insecure Coding Practices



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]

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

NowSecure

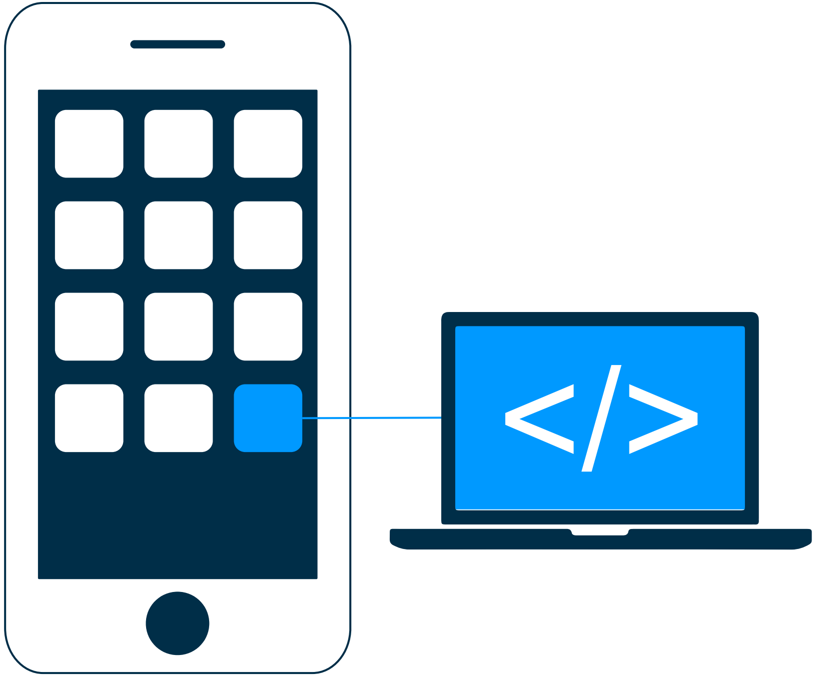
Insecure Coding Practices

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

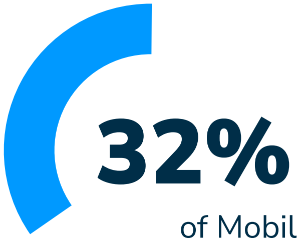
- Test app signed with valid cert
- Test for debug build, hardcoded keys
- Test error conditions, verbose log files
- Check 3rd party libraries



NowSecure

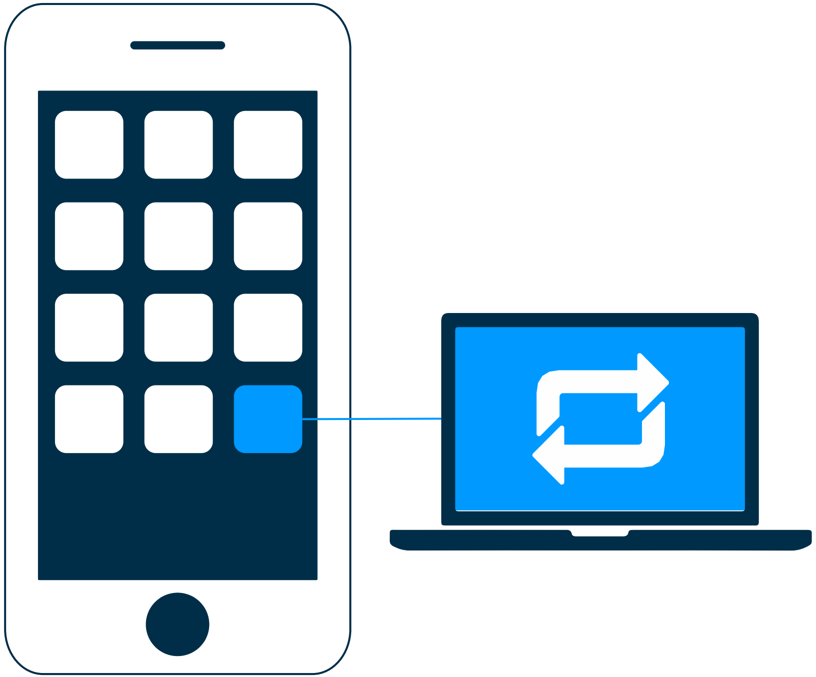
5- Reverse Engineering & Anti-Tampering

Exposure to Reverse Engineering



32%

of Mobile Apps have exposure to reverse engineering



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

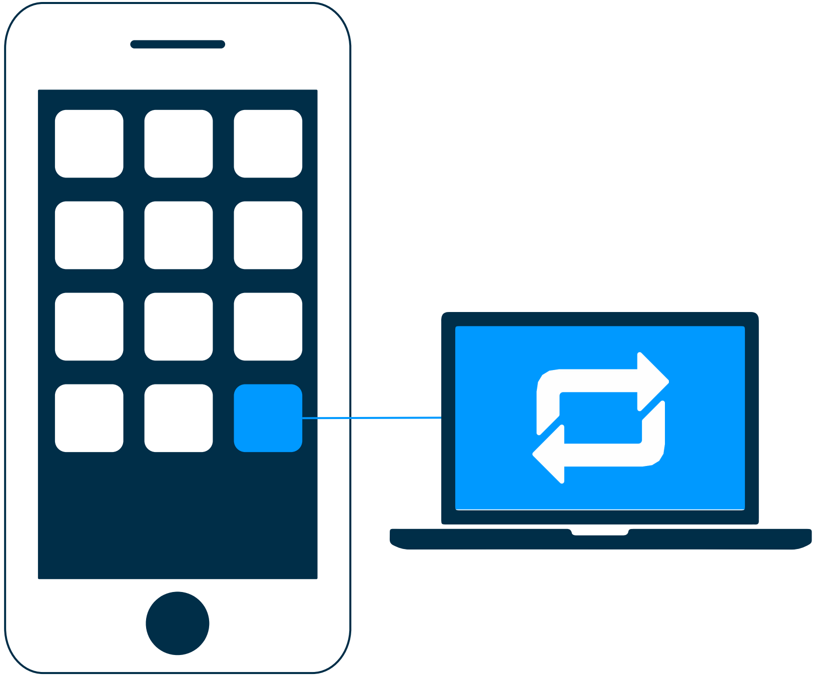
Exposure to Reverse Engineering

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

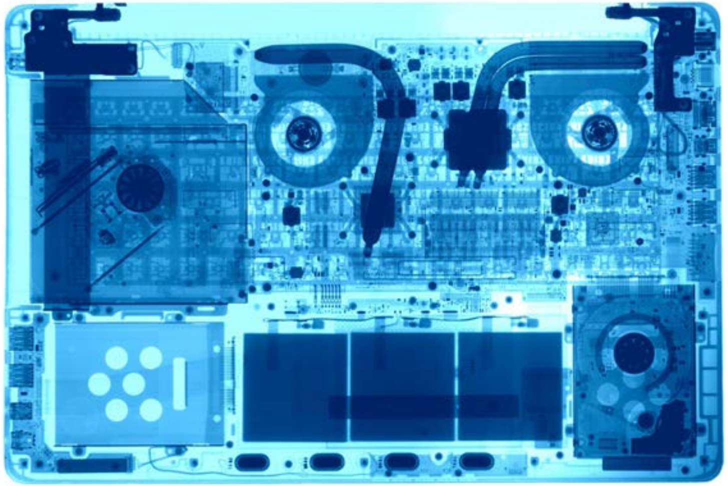


Resiliency Against Reverse Engineering & Tampering

Testing Tip

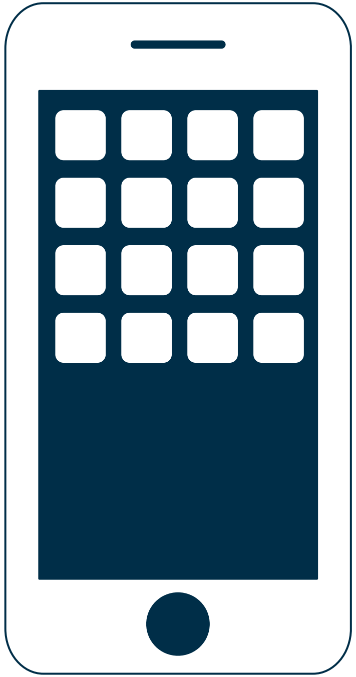
Tamper proofing helps, but only so far...

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



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

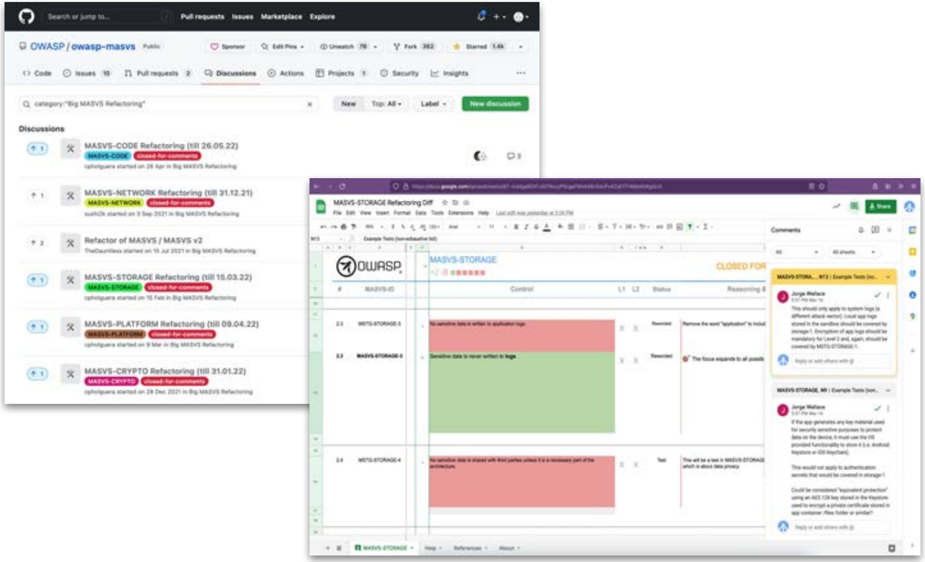
Start with the Big 5 (storage, network, auth, code, RE)

Get involved in the OWASP Mobile Project - Sign Up!

OWASP 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



category:"Big MASVS Refactoring"

Discussions

- ↑ 1 MASVS-CODE Refactoring (till 26.05.22) MASVS-CODE closed-for-comments
  cpholguera started on 28 Apr in Big MASVS Refactoring
- ↑ 1 MASVS-NETWORK Refactoring (till 31.12.21) MASVS-NETWORK closed-for-comments
  sushi2k started on 3 Sep 2021 in Big MASVS Refactoring
- ↑ 2 Refactor of MASVS / MASVS v2
  TheDauntless started on 15 Jul 2021 in Big MASVS Refactoring
- ↑ 1 MASVS-STORAGE Refactoring (till 15.03.22) MASVS-STORAGE closed-for-comments
  cpholguera started on 15 Feb in Big MASVS Refactoring
- ↑ 1 MASVS-PLATFORM Refactoring (till 09.04.22) MASVS-PLATFORM closed-for-comments
  cpholguera started on 9 Mar in Big MASVS Refactoring
- ↑ 1 MASVS-CRYPTO Refactoring (till 31.01.22) MASVS-CRYPTO closed-for-comments
  cpholguera started on 28 Dec 2021 in Big MASVS Refactoring

#	MASVS-ID	Control	L1	L2	Status	Reasoning
2.3	MSTG-STORAGE-3	No sensitive data is written to application logs.	X	X	Reworded	Remove the word "application" to includ
3.3	MASVS-STORAGE-3	Sensitive data is never written to logs.	X	X	Reworded	The focus expands to all possib
2.4	MSTG-STORAGE-4	No sensitive data is shared with third parties unless it is a necessary part of the architecture.	X	X	Test	This will be a test in MASVS-STORAGE which is about data privacy.

Comments

MASVS-STORA... M13 | Example Tests (no...

Jorge Wallace: This should only apply to system logs (a different attack vector). Local app logs stored in the sandbox should be covered by storage-1. Encryption of app logs should be mandatory for Level 2 and, again, should be covered by MSTG-STORAGE-1.

MASVS-STORAGE, M9 | Example Tests (non...

Jorge Wallace: If the app generates any key material used for security sensitive purposes to protect data on the device, it must use the OS provided functionality to store it (i.e. Android Keystore or iOS Keychain).

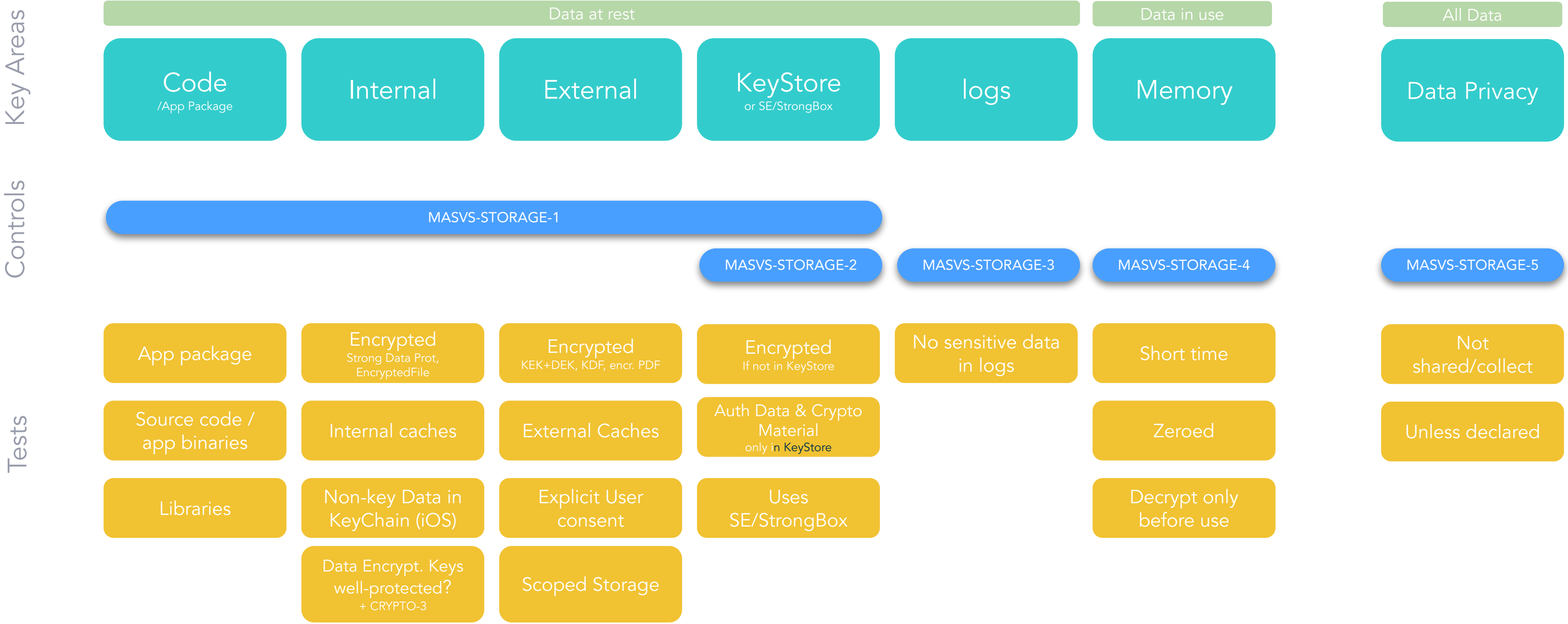
This would not apply to authentication secrets that would be covered in storage-1.

Could be considered "equivalent protection" using an AES 128 key stored in the Keystore used to encrypt a private certificate stored in app container /files folder or similar?

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

NowSecure

OWASP MASVS Refactoring Process



	Data at rest					Data in use	All Data
Key Areas	Code /App Package	Internal	External	KeyStore or SE/StrongBox	logs	Memory	Data Privacy
Controls	MASVS-STORAGE-1						
				MASVS-STORAGE-2	MASVS-STORAGE-3	MASVS-STORAGE-4	MASVS-STORAGE-5
Tests	App package	Encrypted Strong Data Prot, EncryptedFile	Encrypted KEK+DEK, KDF, encr. PDF	Encrypted If not in KeyStore	No sensitive data in logs	Short time	Not shared/collect
	Source code / app binaries	Internal caches	External Caches	Auth Data & Crypto Material only in KeyStore		Zeroed	Unless declared
	Libraries	Non-key Data in KeyChain (iOS)	Explicit User consent	Uses SE/StrongBox		Decrypt only before use	
		Data Encrypt. Keys well-protected? + CRYPTO-3	Scoped Storage				

NowSecure

OWASP MASVS V2 Compliance-as-Code

Human
+
excel/PDF/Word

Read and interpret manually

Hard to prove control and test coverage

Compare providers manually

Hard to maintain

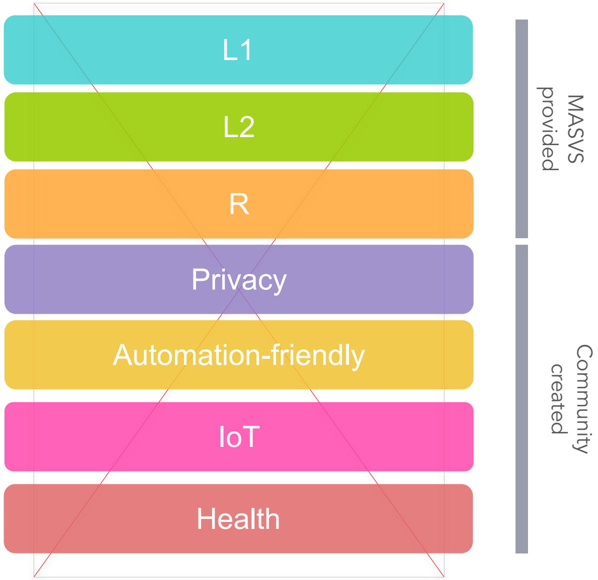
Automation
+
yaml/json/xml

Machine-readable

Easy to prove control and test coverage

Compare providers with benchmarking

Fully traceable



Standard and fully tailored testing

MASVS + proprietary + cross-standards

NowSecure

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

Help us automate & GitHub Actions

Answer Discussions

Give feedback to the MASVS Refactoring

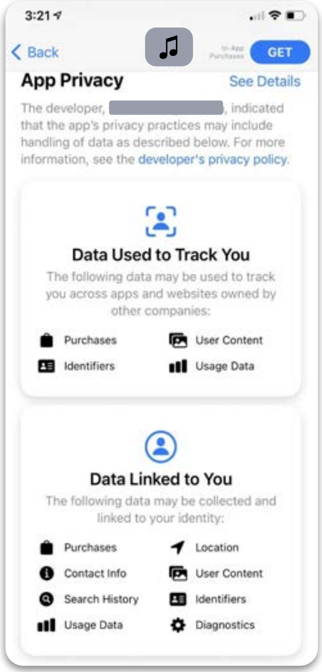
Contribute & connect with us!

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

...

Apple and Google Updates

Apple Privacy And Google Play Data Safety



App Privacy

See Details

The developer, [redacted], indicated that the app's privacy practices may include handling of data as described below. For more information, see the developer's privacy policy.

Data Used to Track You

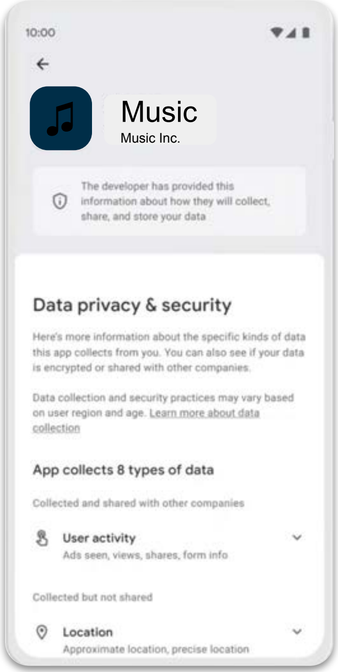
The following data may be used to track you across apps and websites owned by other companies:

- Purchases
- User Content
- Identifiers
- Usage Data

Data Linked to You

The following data may be collected and linked to your identity:

- Purchases
- Location
- Contact Info
- User Content
- Search History
- Identifiers
- Usage Data
- Diagnostics



Music

Music Inc.

The developer has provided this information about how they will collect, share, and store your data

Data privacy & security

Here's more information about the specific kinds of data this app collects from you. You can also see if your data is encrypted or shared with other companies.

Data collection and security practices may vary based on user region and age. Learn more about data collection

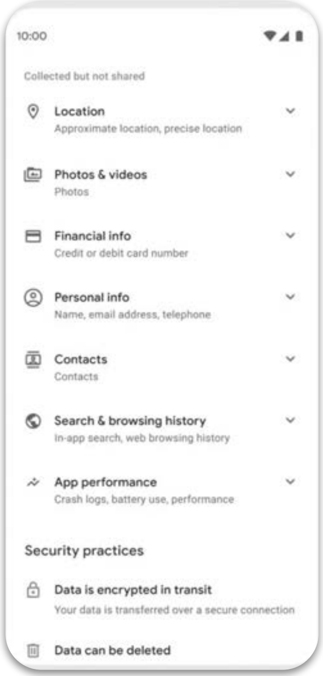
App collects 8 types of data

Collected and shared with other companies

User activity
Ads seen, views, shares, form info

Collected but not shared

Location
Approximate location, precise location



Collected but not shared

Location
Approximate location, precise location

Photos & videos
Photos

Financial info
Credit or debit card number

Personal info
Name, email address, telephone

Contacts
Contacts

Search & browsing history
In-app search, web browsing history

App performance
Crash logs, battery use, performance

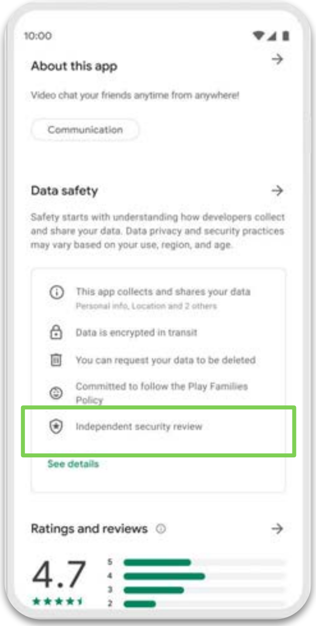
Security practices

Data is encrypted in transit
Your data is transferred over a secure connection

Data can be deleted

NowSecure

ADA Authorized Labs with MASA Verification



10:00

About this app

Video chat your friends anytime from anywhere!

Communication

Data safety

Safety starts with understanding how developers collect and share your data. Data privacy and security practices may vary based on your use, region, and age.

This app collects and shares your data
Personal info, Location and 2 others

Data is encrypted in transit

You can request your data to be deleted

Committed to follow the Play Families Policy

Independent security review

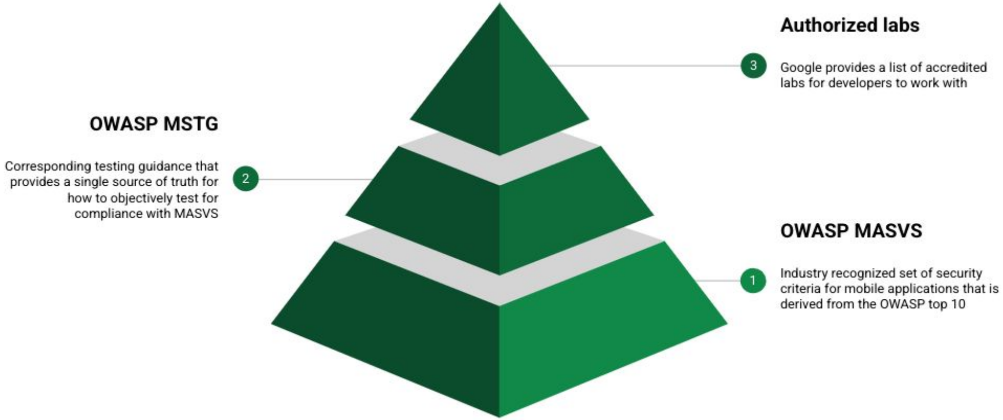
See details

Ratings and reviews

4.7

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.*



Authorized labs

3 Google provides a list of accredited labs for developers to work with

OWASP MSTG

2 Corresponding testing guidance that provides a single source of truth for how to objectively test for compliance with MASVS

OWASP MASVS

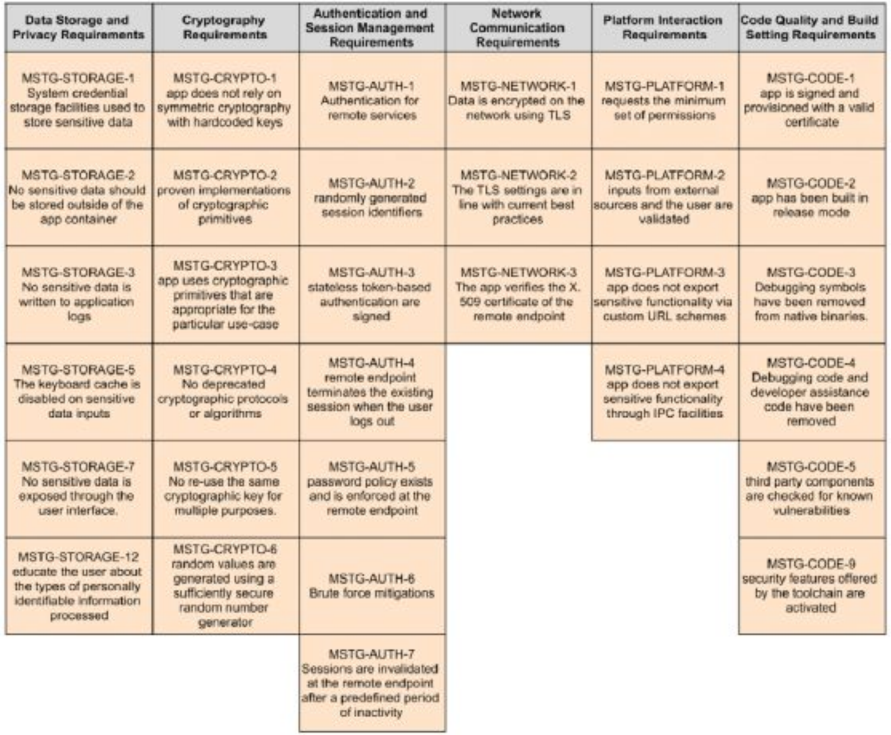
1 Industry recognized set of security criteria for mobile applications that is derived from the OWASP top 10

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

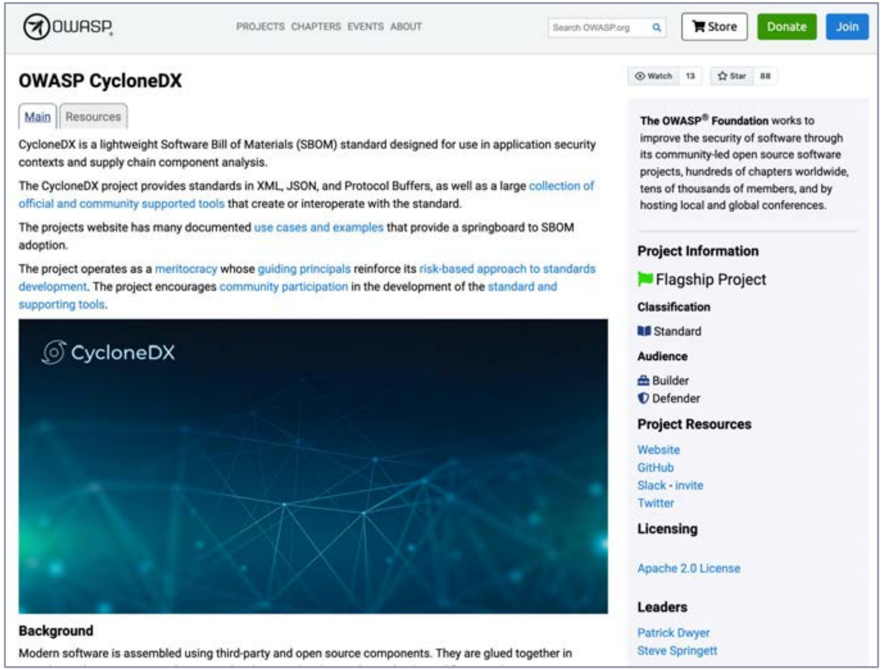


Data Storage and Privacy Requirements	Cryptography Requirements	Authentication and Session Management Requirements	Network Communication Requirements	Platform Interaction Requirements	Code Quality and Build Setting Requirements
MSTG-STORAGE-1 System credential storage facilities used to store sensitive 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 predefined period of inactivity			

OWASP CycloneDX for SBOM

What is OWASP CycloneDX?

Best Practice for MASVS-CODE



OWASP CycloneDX

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/

New Flagship Project at OWASP

A new industry standard for SBOM interoperability

Chaired by Steve Springett & Patrick Dwyer

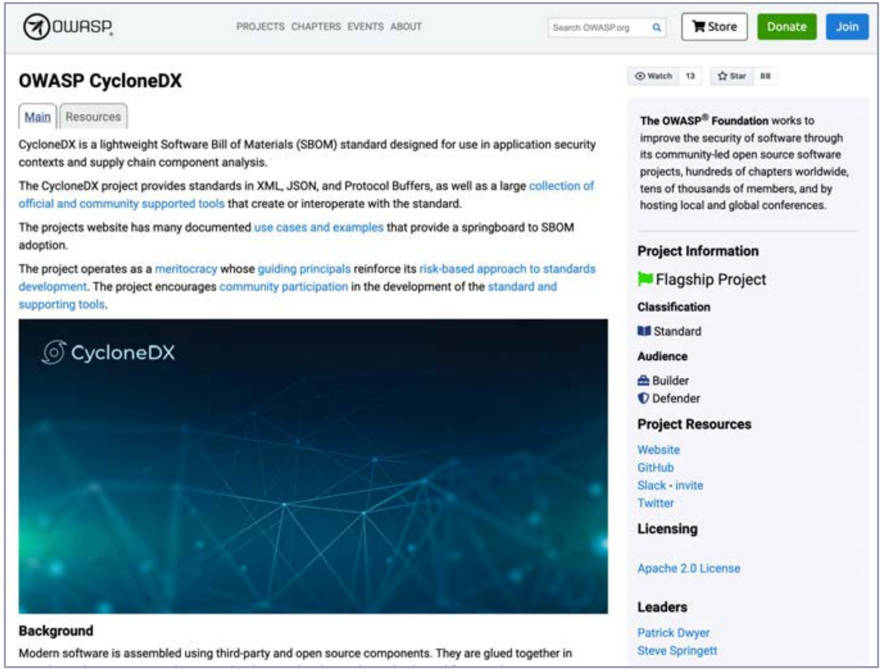
“The CycloneDX SBOM standard is a result of security experts and industry coming together to create an SBOM standard that delivers the transparency and interoperability necessary to communicate software inventory and the relationships across different systems.”

Cross links with OWASP MASVS Poject as well

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

NowSecure

What is OWASP CycloneDX?



OWASP CycloneDX

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

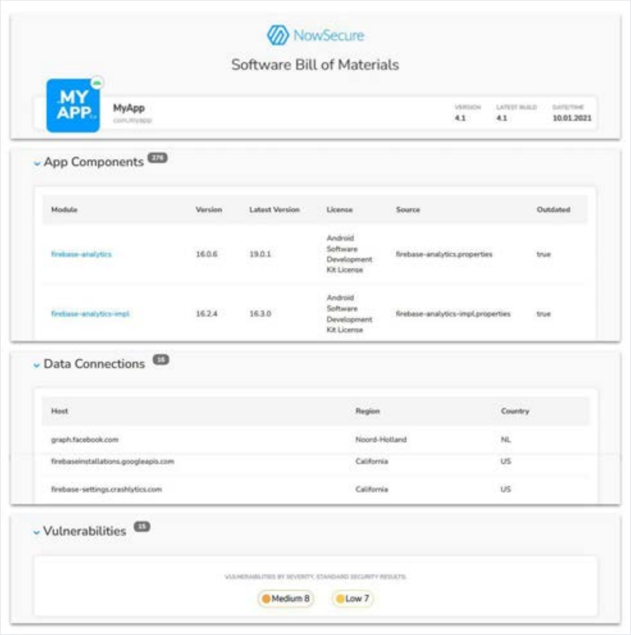
Website
GitHub
Slack - invite
Twitter

Licensing

Apache 2.0 License

Leaders

Patrick Dwyer
Steve Springett



Software Bill of Materials

MyApp
com.myapp

VERSION	LATEST BUILD	DATE/TIME
4.1	4.1	10.01.2021

App Components 276

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 18

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

Vulnerabilities 15

VULNERABILITIES BY SEVERITY, STANDARD SECURITY RESULTS

● Medium 8 ● Low 7

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

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

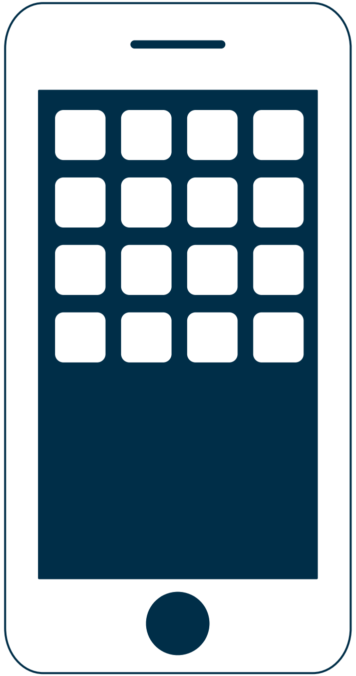
NowSecure

Resources Resources Resources

Mobile Pen Tester’s Toolkit

Manual & OSS Testing Resources

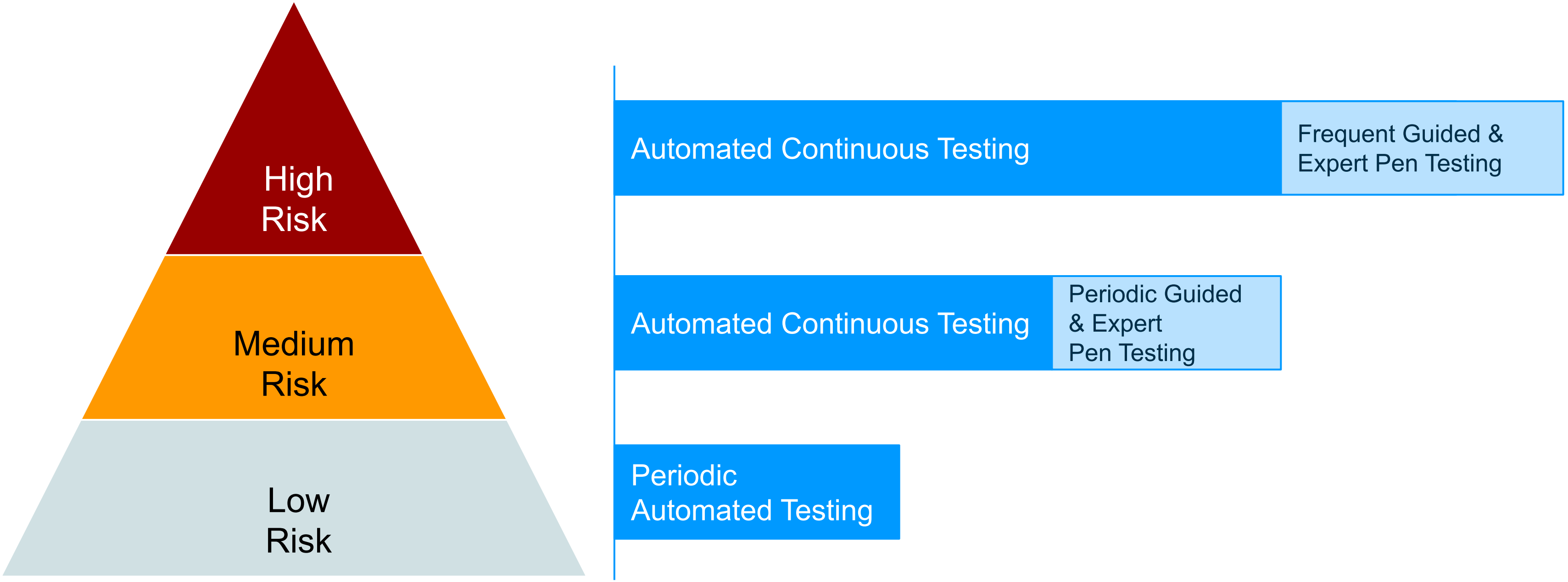
- MASVS repo
- MSTG repo
- MSTG Hacking Playground
- Frida Dynamic Instrumentation Toolkit
- Radare Portable Reversing Framework
- Burp Suite or ZedAttackProxy
- Jailbroken & Rooted devices



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



High Risk

Automated Continuous Testing | Frequent Guided & Expert Pen Testing

Medium Risk

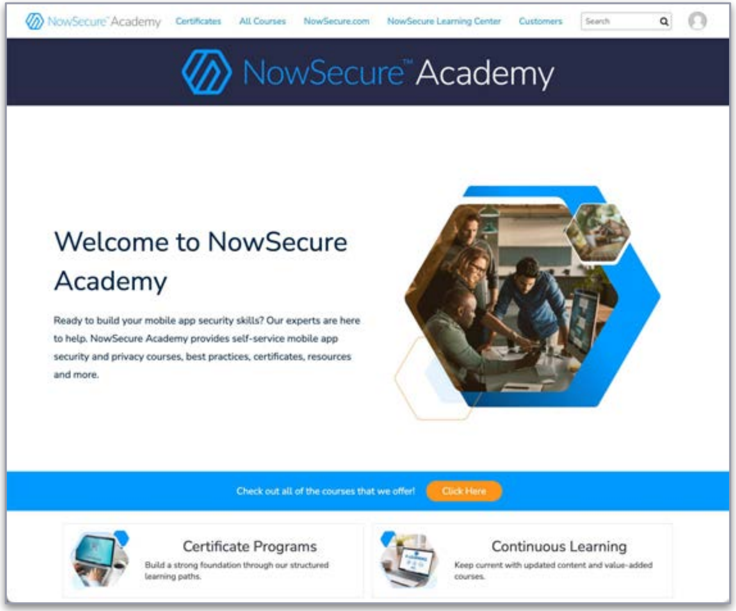
Automated Continuous Testing | Periodic Guided & Expert Pen Testing

Low Risk

Periodic Automated Testing

NowSecure

Free Training



Welcome to NowSecure Academy

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.

Check out all of the courses that we offer! Click Here

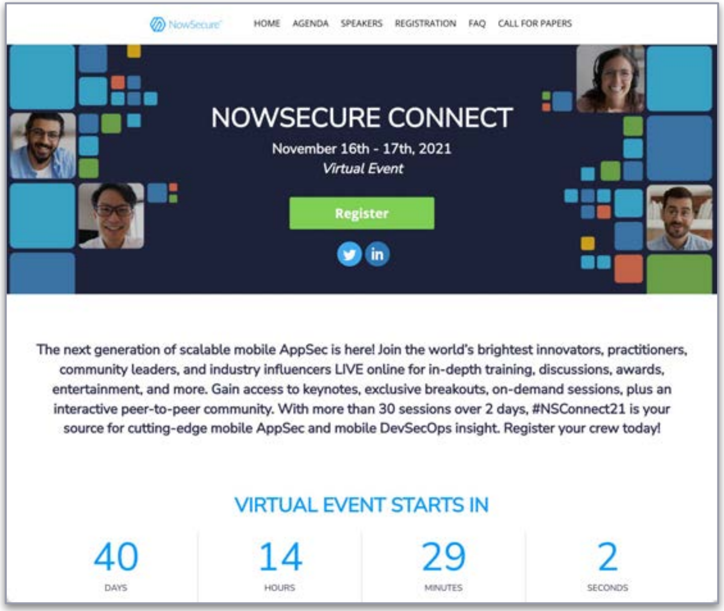
Certificate Programs

Build a strong foundation through our structured learning paths.

Continuous Learning

Keep current with updated content and value-added courses.

Online Courseware
https://academy.nowsecure.com



NOWSECURE CONNECT

November 16th - 17th, 2021
Virtual 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!

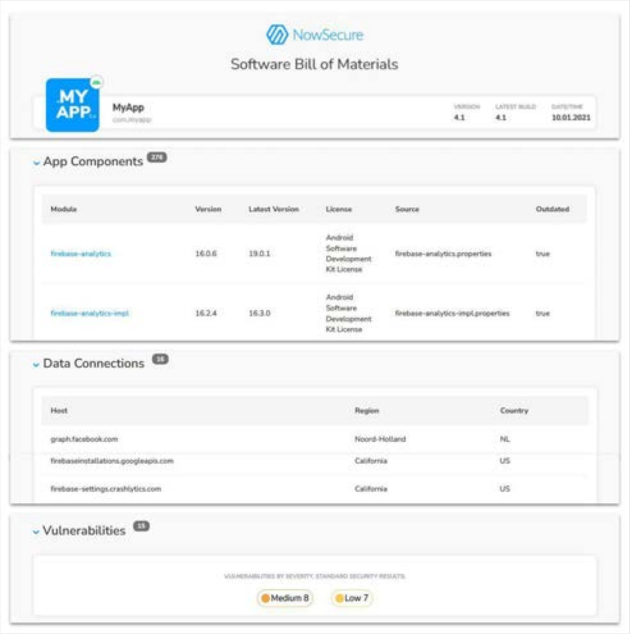
VIRTUAL EVENT STARTS IN

40	14	29	2
DAYS	HOURS	MINUTES	SECONDS

Full Replays
https://bit.ly/ns-connect

NowSecure

Checkout Your Own Mobile Apps



NowSecure

Software Bill of Materials

MyApp

VERSION	LATEST BUILD	DATE/TIME
4.1	4.1	10.01.2021

App Components 276

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 16

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

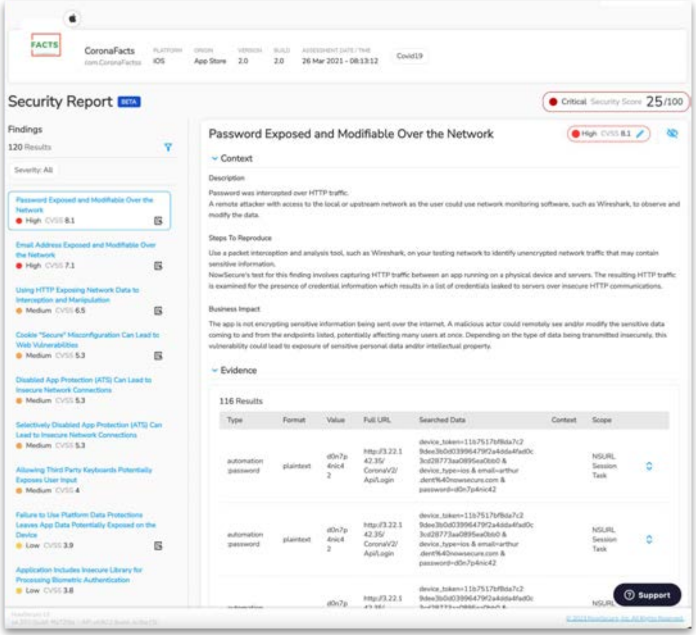
Vulnerabilities 15

VULNERABILITIES BY SEVERITY, STANDARD SECURITY RESULTS

● Medium 8 ● Low 7

Free SBOM

https://bit.ly/ns-SBOM10



CoronaFacts

PLATFORM	ORIGIN	VERSION	BUILD	ASSESSMENT DATE / TIME
iOS	App Store	2.0	2.0	26 Mar 2021 - 08:13:12

Covid19

Security Report BETA

● Critical Security Score 25/100

Findings

120 Results

Severity: All

- 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
- Cookie "Secure" Misconfiguration Can Lead to Web Vulnerabilities — ● Medium CVSS 5.3
- Disabled App Protection (ATS) Can Lead to Insecure Network Connections — ● Medium CVSS 5.3
- Selectively Disabled App Protection (ATS) Can Lead to Insecure Network Connections — ● Medium CVSS 5.3
- Allowing Third Party Keyboards Potentially Exposes User Input — ● Medium CVSS 4
- Failure to Use Platform Data Protections Leaves App Data Potentially Exposed on the Device — ● Low CVSS 3.9
- Application Includes Insecure Library for Processing Biometric Authentication — ● Low CVSS 3.8

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

116 Results

Type	Format	Value	Full URL	Searched Data	Context	Scope
automation password	plaintext	d0n7p4nic42	http://3.22.142.35/CoronaV2/Api/Login	device_token=11b7517bf8da7c29dee3b0d03996479f2a4dda4fad0c3cd28773aa0895ea0bb0 & device_type=ios & email=arthur.dent%40nowsecure.com & password=d0n7p4nic42		NSURL Session Task
automation password	plaintext	d0n7p4nic42	http://3.22.142.35/CoronaV2/Api/Login	device_token=11b7517bf8da7c29dee3b0d03996479f2a4dda4fad0c3cd28773aa0895ea0bb0 & device_type=ios & email=arthur.dent%40nowsecure.com & password=d0n7p4nic42		NSURL Session Task

Free Security Report

https://bit.ly/ns-report

NowSecure

More Free Resources

A Manager's Guide to the OWASP Mobile Security Project

Building & Executing Risk-Based Security Policy for Mobile Apps

NowSecure

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

The Top 5

Mobile App Security Issues

— and How Developers Can Avoid Them —

Mobile app developers can reduce the chance of attackers exploiting their apps by applying secure coding best practices. Knowing the most frequently found mobile application security issues to avoid enables developers to work smarter and faster.

The Open Web Application Security Project (OWASP) Mobile Top 10 enumerates the most common areas of risk in mobile applications and provides a foundation to educate developers about prevalent security gaps and secure mobile app development best practices.

Here are the key problems that compromise mobile application security and best practices developers should adopt to avoid these issues.

1 Data stored in an insecure, exposed location

OWASP Mobile Top 10 Mapping
M1: Improper Platform Usage
M2: Insecure Data Storage
M5: Insufficient Cryptography
M7: Client Code Quality

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

NowSecure

MOBILE APP SECURITY PROGRAM MANAGEMENT HANDBOOK

GETTING STARTED WITH:

PEOPLE PROCESS TECHNOLOGY

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

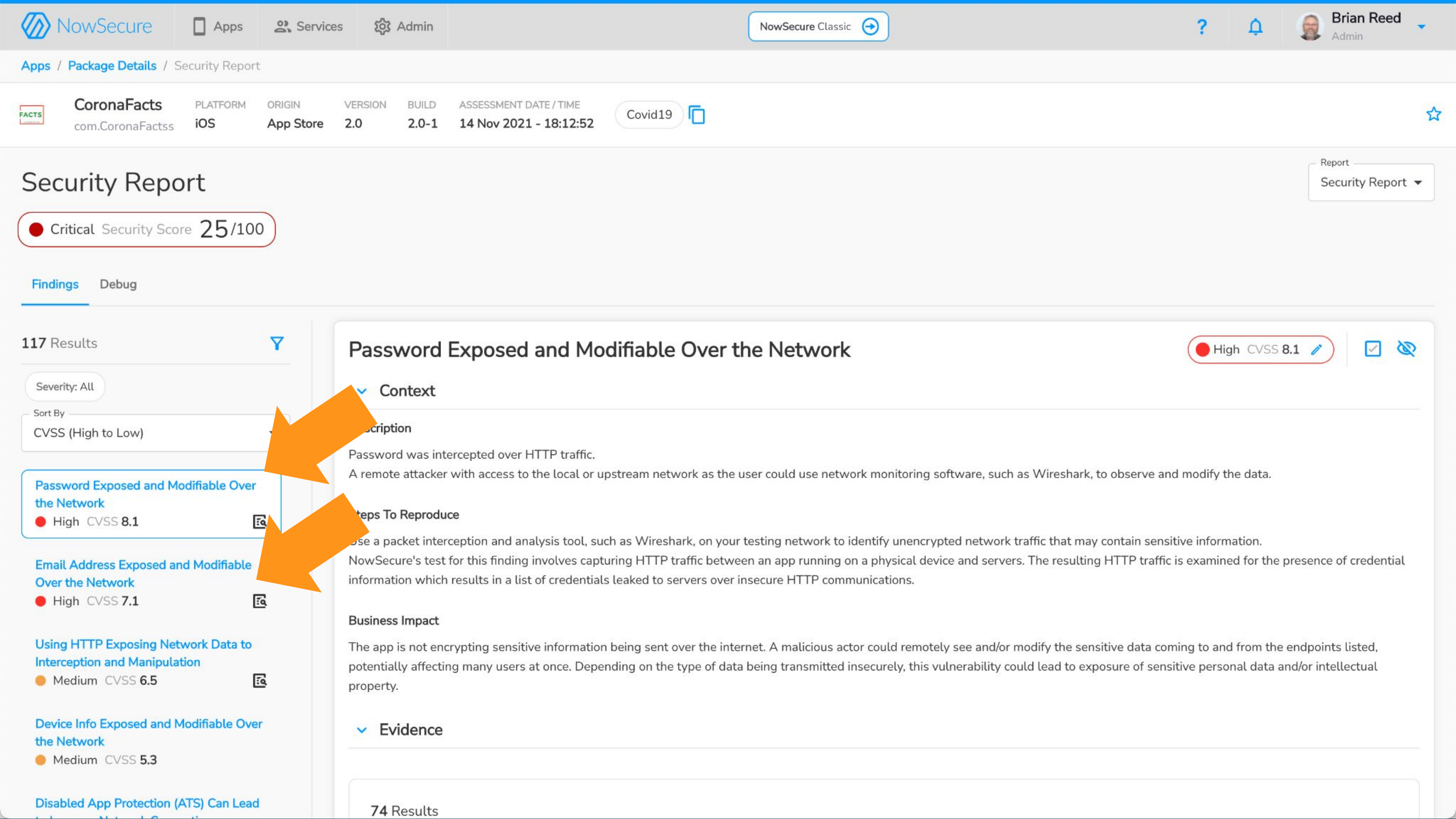
NowSecure Workstation
All-in-one mobile pen tester toolkit for productivity

NowSecure Academy
Online courseware and certification for mobile

NowSecure Pen Testing
Expert full scope mobile pen testing services & remediation

NowSecure Mobileverse™
Customer community to onboard, learn & network with peers

NowSecure



CoronaFacts
com.CoronaFactss

PLATFORM	ORIGIN	VERSION	BUILD	ASSESSMENT DATE / TIME
iOS	App Store	2.0	2.0-1	14 Nov 2021 - 18:12:52

Covid19

Report: Security Report

Security Report

● Critical Security Score 25/100

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

Password Exposed and Modifiable Over the Network

● High CVSS 8.1

Context

cription

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.

teps To Reproduce

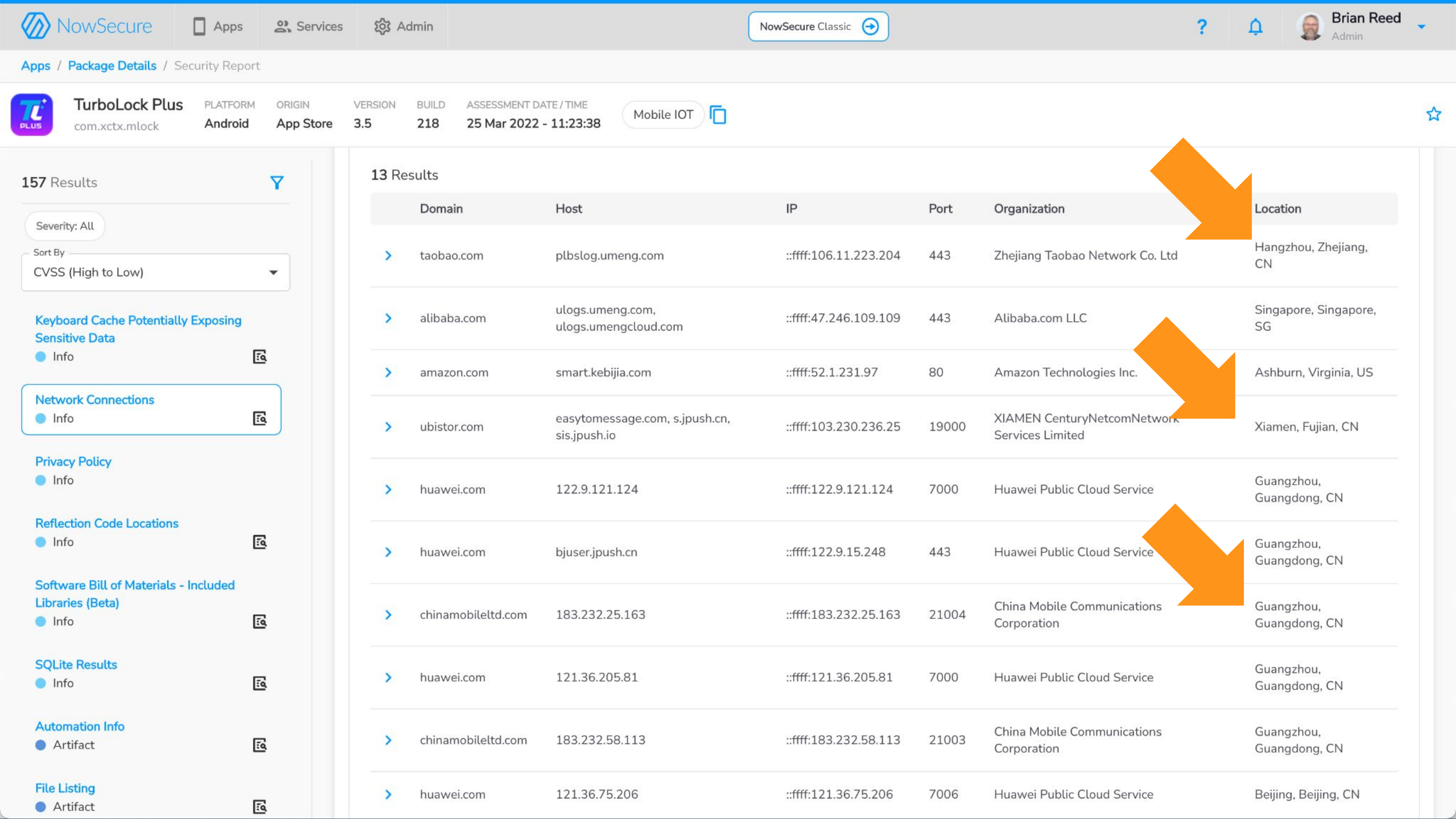
se a packet interception and analysis tool, such as Wireshark, on your testing network to identify unencrypted network traffic that may contain sensitive information.
NowSecure's test for this finding involves capturing HTTP traffic between an app running on a physical device and servers. The resulting 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



Apps / Package Details / Security Report

TurboLock Plus
com.xctx.mlock

PLATFORM	ORIGIN	VERSION	BUILD	ASSESSMENT DATE / TIME
Android	App Store	3.5	218	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

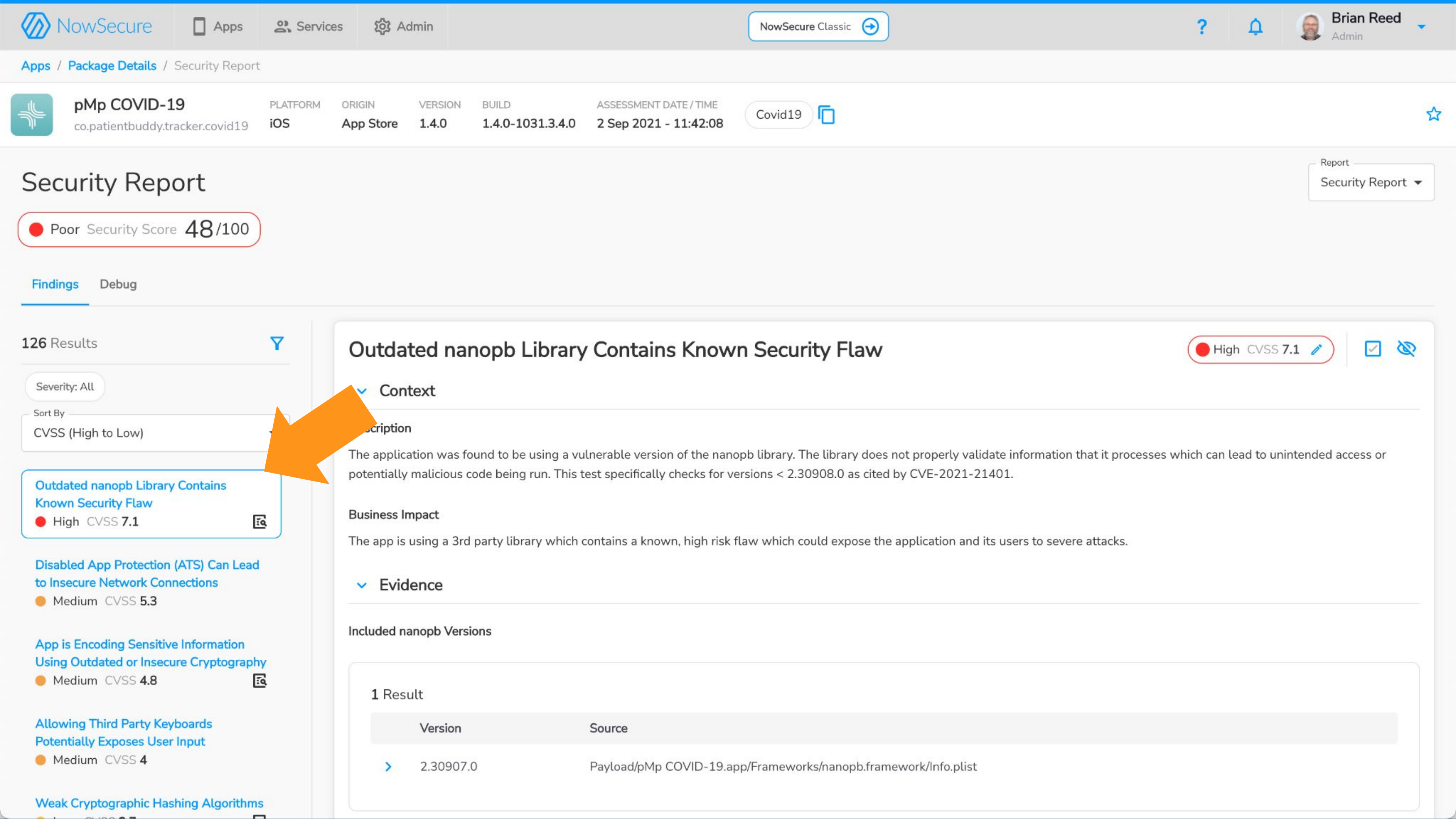
SQLite Results
● Info

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, s.jpush.cn, sis.jpush.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.jpush.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



NowSecure | Apps | Services | Admin | NowSecure Classic | ? | Brian Reed Admin

Apps / Package Details / Security Report

pMp COVID-19
co.patientbuddy.tracker.covid19

PLATFORM	ORIGIN	VERSION	BUILD	ASSESSMENT DATE / TIME
iOS	App Store	1.4.0	1.4.0-1031.3.4.0	2 Sep 2021 - 11:42:08

Covid19

Report: Security 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

cription

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

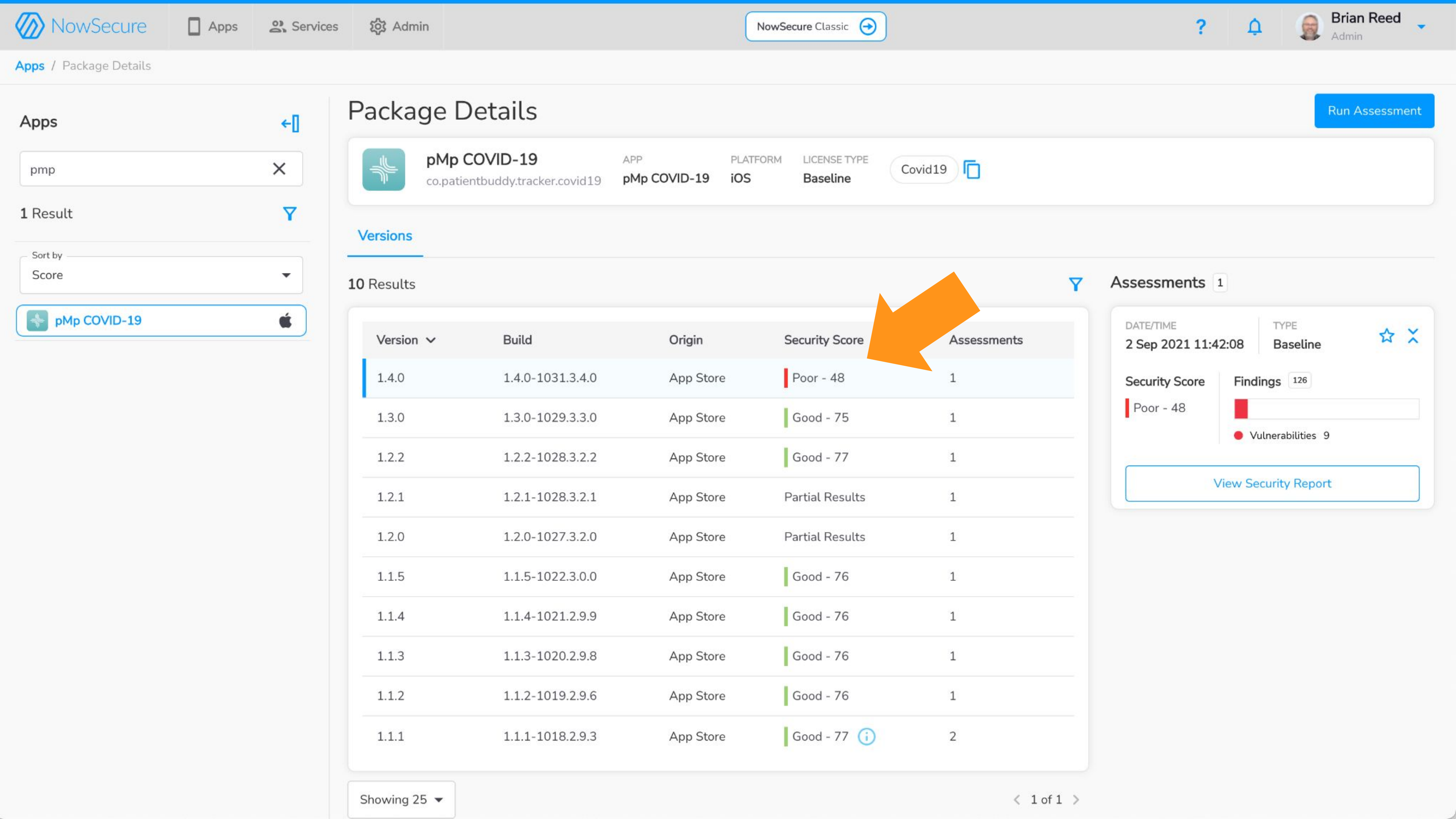
The app is using a 3rd party library which contains a known, high risk flaw which could expose the application and its users to severe attacks.

Evidence

Included nanopb Versions

1 Result

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



Apps / Package Details

Apps

pmp

1 Result

Sort by
Score

pMp COVID-19

Package Details

Run Assessment

pMp COVID-19
co.patientbuddy.tracker.covid19

APP	PLATFORM	LICENSE TYPE
pMp COVID-19	iOS	Baseline

Covid19

Versions

10 Results

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

Showing 25

1 of 1

Assessments 1

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

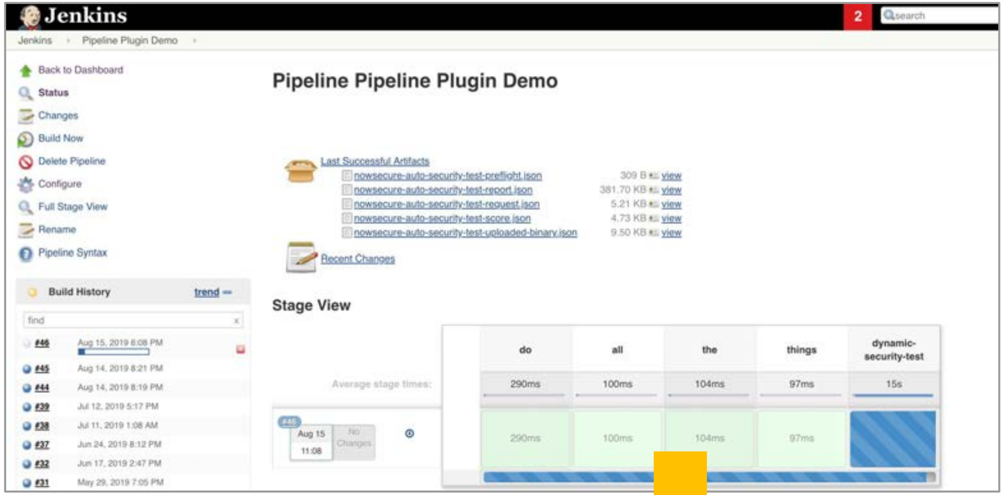
Security Score
Poor - 48

Findings 126

● Vulnerabilities 9

View Security Report

Sample Automated Workflow: Build, Test, Ticket, Repair



Jenkins | 2 | search

Jenkins › Pipeline Plugin Demo ›

- Back to Dashboard
- Status
- Changes
- Build Now
- Delete Pipeline
- Configure
- Full Stage View
- Rename
- Pipeline Syntax

Pipeline Pipeline Plugin Demo

Last Successful Artifacts

nowsecure-auto-security-test-preflight.json	309 B	view
nowsecure-auto-security-test-report.json	381.70 KB	view
nowsecure-auto-security-test-request.json	5.21 KB	view
nowsecure-auto-security-test-score.json	4.73 KB	view
nowsecure-auto-security-test-uploaded-binary.json	9.50 KB	view

Recent Changes

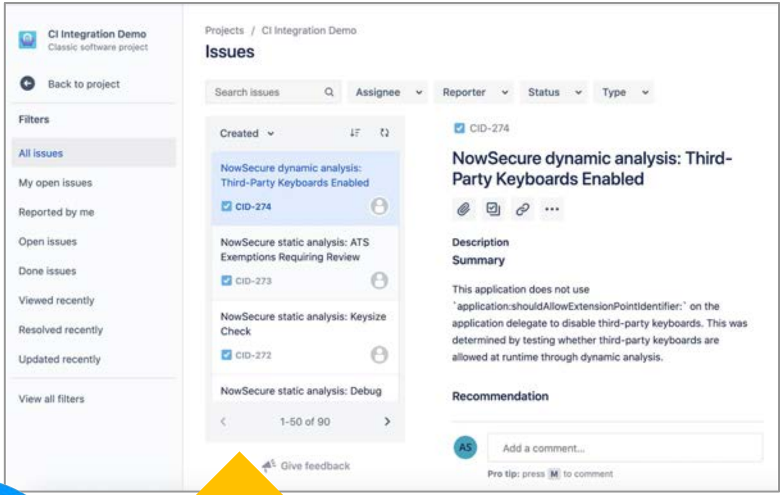
Build History trend —

find

#46	Aug 15, 2019 6:08 PM	
#45	Aug 14, 2019 8:21 PM	
#44	Aug 14, 2019 8:19 PM	
#39	Jul 12, 2019 5:17 PM	
#38	Jul 11, 2019 1:08 AM	
#37	Jun 24, 2019 8:12 PM	
#32	Jun 17, 2019 2:47 PM	
#31	May 29, 2019 7:05 PM	

Stage View

	do	all	the	things	dynamic-security-test
Average stage times:	290ms	100ms	104ms	97ms	15s
#46 Aug 15 11:08 No Changes	290ms	100ms	104ms	97ms	



CI Integration Demo
Classic software project

Back to project

Filters

- All issues
- My open issues
- Reported by me
- Open issues
- Done issues
- Viewed recently
- Resolved recently
- Updated recently
- View all filters

Projects / CI Integration Demo

Issues

Search issues | Assignee | Reporter | Status | Type

Created

- NowSecure dynamic analysis: Third-Party Keyboards Enabled — CID-274
- NowSecure static analysis: ATS Exemptions Requiring Review — CID-273
- NowSecure static analysis: Keysize Check — CID-272
- NowSecure static analysis: Debug

1-50 of 90

Give feedback

CID-274

NowSecure dynamic analysis: Third-Party Keyboards Enabled

Description

Summary

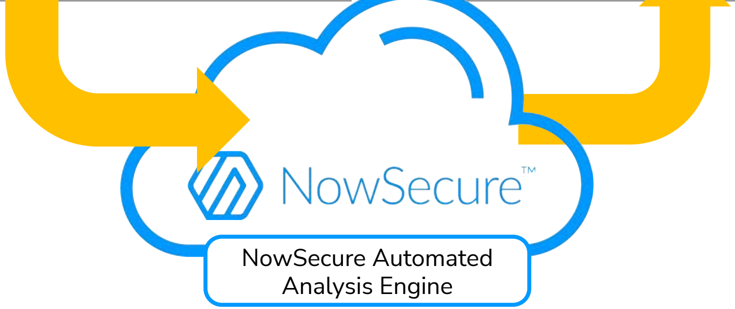
This application does not use `application:shouldAllowExtensionPointIdentifier:` on the application delegate to disable third-party keyboards. This was determined by testing whether third-party keyboards are allowed at runtime through dynamic analysis.

Recommendation

AS Add a comment...

Pro tip: press M to comment

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



NowSecure Automated Analysis Engine

NowSecure

NowSecure™

All-in-one SAST+DAST+IAST

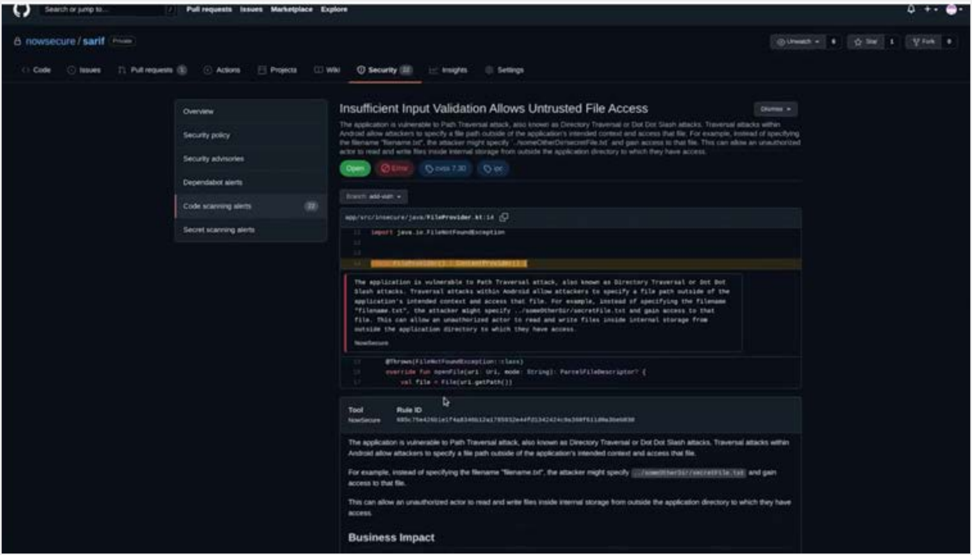
Security test any iOS/Android app binary, any language

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



Insufficient Input Validation Allows Untrusted File Access

The application is vulnerable to Path Traversal attack, also known as Directory Traversal or Dot Dot Slash attacks. Traversal attacks within Android allow attackers to specify a file path outside of the application's intended context and access that file. For example, instead of specifying the filename "filename.txt", the attacker might specify ../someOtherDir/secretFile.txt and gain access to that file. This can allow an unauthorized actor to read and write files inside internal storage from outside the application directory to which they have access.

app/src/insecure/java/FileProvider.kt:14

```
import java.io.FileNotFoundException


[illegible] FileProvider() : ContentProvider() {
```

```
The application is vulnerable to Path Traversal attack, also known as Directory Traversal or Dot Dot
Slash attacks. Traversal attacks within Android allow attackers to specify a file path outside of the
application's intended context and access that file. For example, instead of specifying the filename
"filename.txt", the attacker might specify ../someOtherDir/secretFile.txt and gain access to that
file. This can allow an unauthorized actor to read and write files inside internal storage from
outside the application directory to which they have access.
NowSecure
```

```
    @Throws(FileNotFoundException::class)
    override fun openFile(uri: Uri, mode: String): ParcelFileDescriptor? {
        val file = File(uri.getPath())
```

Tool	Rule ID
NowSecure	[illegible]

The application is vulnerable to Path Traversal attack, also known as Directory Traversal or Dot Dot Slash attacks. Traversal attacks within Android allow attackers to specify a file path outside of the application's intended context and access that file.

For example, instead of specifying the filename "filename.txt", the attacker might specify `../someOtherDir/secretFile.txt` and gain access to that file.

This can allow an unauthorized actor to read and write files inside internal storage from outside the application directory to which they have access.

Business Impact

NowSecure GitHub Action Resources

Demo Video Link   GitHub Action Link

GitHub

THANK YOU!

OWASP Meetup

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

NowSecure