NowSecure™

MOBILE APP SECURITY PROGRAM MANAGEMENT HANDBOOK

GETTING STARTED WITH:

PEOPLE

PROCESS

TECHNOLOGY

HELLO THERE,

Like everyone does, I enjoy meeting people I share common interests with. Because you're reading this guide, I suspect we share one -- making mobile apps more secure. So it's nice to meet you.

We're all spending more time with mobile apps these days, and apps are changing our personal and work lives. But because of the incredible amount of data generated, stored, and transmitted by these apps and our mobile devices, we as consumers, employees, IT professionals, and security practitioners need to be careful.

At NowSecure, I talk to a lot of people — CISOs, security analysts, quality assurance engineers, and others — about mobile app security. I'm exposed to many different application security programs at varying levels of maturity. I also manage the mobile app security program at NowSecure, which includes things like overseeing our mobile app security testing methodology and process, managing a high-performing team of mobile app penetration testers, developing reports, and debriefing clients.

Whether you are starting from square one with a mobile app security initiative, have a bunch of ad-hoc tasks you want to consolidate into a repeatable process, or just want to make some tweaks to improve an existing program -- chances are I've been there before.

While I call this document a "getting started guide," regardless of your program's maturity I think you'll find some helpful hints regarding the **people, process,** and **technology** that can elevate your mobile app security program to the next level.

If you find these tips helpful or have questions or suggestions, let me know via e-mail or Twitter!

Thanks for reading,

KATIE STRZEMPKA

Vice President, Customer Success and Services

kstrzempka@nowsecure.com

@kstrzemp

FIRST THINGS FIRST

What is a mobile app security program?

A mobile app security program identifies, analyzes, and manages the risk associated with your portfolio of mobile apps on a continual basis. Ideally that covers the complete spectrum from inception through design, development, deployment, and finally end-of-life. A program is ongoing and incorporates metrics that allow you to quantify your progress in meeting the objectives you've set. Creating a measurable program requires repeatable methods and processes that result in consistent outcomes.

Some key questions you might ask to help establish metrics include the following:

- How many mobile apps does your organization have?
- How critical is each mobile app to your business goals?
- How many mobile app security flaws are in production?
- How many security flaws are fixed before and/or after deployment?
- Are the number of identified flaws increasing or decreasing over time?

Mobile app security program challenges

Some enterprises have only just started developing their first native mobile apps and aren't sure where to start with mobile app security testing. The mobile attack surface differs from that of web applications and so legacy web application testing frameworks are not enough.

Identifying mobile app security flaws requires performing tests that account for the idiosyncrasies of the operating system or platform (e.g., Android and iOS for our purposes). Those tests require a wide range of techniques and fields of expertise, for example analyzing cryptographic algorithms and their implementation and reverse-engineering proprietary protocols within the binary.

Many of the app security tools available in the market today were originally developed to assess web applications and can't drill down to this level of detail. On top of that, many of these tools are doing static analysis only, and therefore require access to source code.

Another common problem for larger organizations is the sheer volume of mobile apps that need testing. Many enterprises have internal security teams that perform manual testing of apps but can't keep up with the demands on their time. These teams also experience friction when delivering testing results to their development teams. It's common for security analysts to receive a build for analysis with very little time before a scheduled deployment to test. Identifying vulnerabilities too late in the game results in a mad rush to remediate and oftentimes delays a release.

PEOPLE

BUILD A HIGH-EFFICIENCY MOBILE APP SECURITY TEAM

REQUIRED AREAS OF EXPERTISE

Some readers may already have a team in place that focuses on mobile application security testing. If so, HIGH FIVE! For readers whose organization is only beginning to develop mobile apps or has outsourced mobile app security testing in the past, a pillar of your program will be your core mobile team. Something very unique about mobile app security talent is the rarity of finding a single person that embodies all of the skills needed to cover the entire mobile attack surface.

A highly functional mobile application security testing team needs expertise covering the following aspects of the mobile app attack surface:

Mobile forensics and data recovery: Knowing how to forensically examine data at rest to ensure apps do not store sensitive data insecurely on the device.
Network security and web services/API testing: Evaluating whether apps properly encrypt the data sent to various endpoints. Vulnerabilities classified as high and critical severity according to the Common Vulnerability Scoring System (CVSS) result from failures to protect data in transit.
Server-side penetration testing: Diagnosing the insecure storage of sensitive data on the backend.
Reverse-engineering and code analysis: Identifying weaknesses in code that are vulnerable to exploit.

The size of your team will vary depending on the number of apps that need testing, as well as, the type and complexity of the security tools you use. The ideal team will not only include a mixture of skill-sets as described above but will include people of varying experience. Count on entry-level security analysts to handle some aspects of initial testing and pass their findings on to a more experienced analyst. More experienced analysts can then focus their time and energy on the areas that require their expert knowledge.

PEOPLE

SMALLER TEAMS

A small team might consist of one or two mobile app security analysts responsible for testing one to ten applications each year (along with bug fixes, feature additions, minor releases, or major updates for those apps). Analysts on smaller teams don't typically test mobile apps full time. It's usually only a subset of their overall job responsibilities.

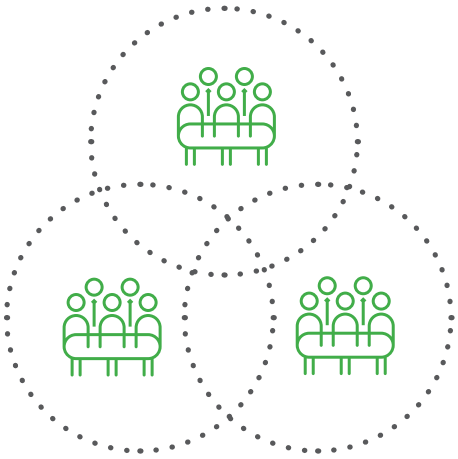
Oftentimes, a compilation of open-source tools might suffice for a small team provided that as a whole, the team has expertise in all of the areas mentioned above. Open-source tools can be problematic if the team already faces a backlog of app testing. Manually setting up different testing environments, troubleshooting, and compiling results from multiple tools into one hand-written report can eat up precious hours of a small security team's already limited time.

LARGER TEAMS

I define a large mobile app security team as one that includes three or more mobile app security analysts that continually test numerous mobile apps (and app updates) on an annual basis. Typically, this team's sole responsibility is app testing because the organization develops a large volume of apps that they release or update frequently.

A big challenge for larger teams is a hefty testing backlog and/or pressure from the business to perform testing and provide results more quickly. Automated mobile app security testing that provides a reasonable amount of coverage can free security analysts up to focus a majority of their time on areas that require more in-depth manual analysis.

ESTABLISH BUY IN AND CROSS-TEAM COLLABORATION



When you launch your mobile app security testing program, begin by reminding your developers, your DevOps team, and yourself that you're all on the same team in pursuit of the same business goals. The apps the team develops generate revenue, increase the organization's productivity, and contribute to business objectives in any number of ways. The security team contributes by protecting these valuable assets and eliminating mobile app security risks that can cost the business millions.

PROCESS

In general, the objective of most mobile app security testing programs is to make sure that the mobile apps deployed:

- Store data securely (if data storage is necessary)

- Transmit data securely

- Don't expose sensitive data via backend components

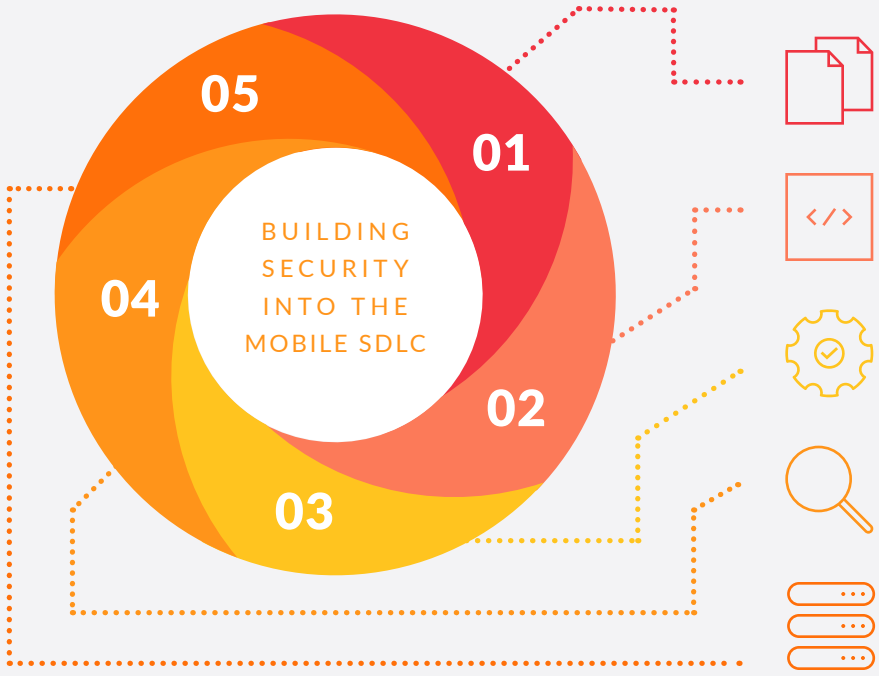
- Are developed in a way that doesn't allow for manipulation of their code (e.g. via reverse-engineering)

To build a process that achieves these objectives, you'll need to define mobile app security standards that govern what you will or will not accept based on your organization's risk appetite.

PROCESS
DOCUMENT MOBILE APP SECURITY STANDARDS

In order for the program to be successful, all of the players involved (security, development, product, engineering) must agree and adhere to a set of mobile app security standards. Documenting requirements for mobile app security, providing information about how apps will be tested against those requirements, and deciding what security issues will delay or block a release is a key step. It will also go a long way in building goodwill with your development team, as developers will know what to expect.

Even better is teaching developers how to code securely, thereby meeting the requirements, and fixing security issues themselves. This saves time for developers, security team members, and quality assurance staff. Depending on the maturity of your program, providing developers with automated mobile app security testing tools can multiply the time savings. Technology exists that will automatically test mobile app builds as part of continuous integration (CI), continuous delivery (CD), or other DevOps processes and provide feedback directly to developers.



Document mobile app security policies

Code using secure development best practices

Perform automated security assessments on each build

Perform penetration testing on each release candidate

Deploy more secure mobile apps to production

PROCESS
DEVELOP A TESTING CHECKLIST

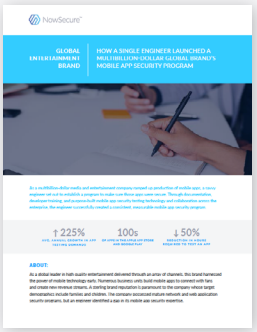
Once you document your mobile app security standards, you can establish your testing requirements. Ideally, you have tests that validate an app's compliance with each and every requirement. Obviously, those testing capabilities should be requirements in your evaluation criteria as you evaluate testing technology and tools for your team.

Testing checklists help ensure consistent, repeatable testing and reporting. To create your own checklist, start with de facto industry standards (e.g., the OWASP Top Ten Mobile Risks) and incorporate any additional tests you need in order to audit apps against your internal policies. However, be sure to also give your testers some leeway so that they can get creative and apply an attacker's perspective in testing an app.

A CHECKLIST WILL ALSO HELP YOU:

- On-board new team members more quickly
- Provide details to developers about the flaws you'll test for and how you will test for them
- Reduce the likelihood that developers will be surprised or bitter about security findings they need to fix

Learn how a multibillion dollar global entertainment brand built a mobile app security program.



Read the case study at: www.nowsecure.com/go/case-study-program/.

PROCESS

MOBILE APP SECURITY TRAINING AND EDUCATION

To ensure consistency and repeatability, it's crucial to train both your security analysts, as well as, your developers on what makes the mobile app attack surface unique. While there is some overlap between the types of exploits and vulnerabilities found in web versus native apps, there are quite a few unique differences as well.

To ensure proper test coverage, make sure both your security and development teams are aware of these nuances. Resources such as the OWASP Mobile Security Project and the NowSecure Secure Mobile Development Best Practices are good starting points for training material.

EDUCATIONAL RESOURCES:

OWASP

Mobile Security Project

NowSecure

Secure Mobile Development Best Practices

INTEGRATE WITH DEVELOPMENT AND DEVOPS PROCESSES

IDEALLY, YOU'LL ESTABLISH BUY-IN WITH YOUR DEVELOPMENT AND DEVOPS TEAMS THROUGH THE FOLLOWING:

- Educating them on how mobile app security flaws put the business at risk
- Explaining how identifying and eliminating security flaws earlier in the software development lifecycle (SDLC) saves them time and reduces stress
- Documenting the security policies against which you'll assess mobile apps
- Training them on secure coding practices that prevent those mobile app security issues
- Demonstrating how you will test apps for those issues
- Integrating automated mobile app security testing into their existing technology stack so they can perform basic assessments of their apps without hassle

PROCESS

MAKE AUTOMATED SECURITY TESTING A DEVOPS REALITY

Many developers have bad memories of delayed releases and panicked, eleventh-hour scrambles to fix security issues. I've seen even the most stubborn developers come around once they understood the possibility and the benefits of baking security testing directly into their continuous integration (CI) and/or continuous delivery (CD) processes.

Automating a portion of security testing up front reveals defects earlier so that developers can fix them before passing the app along for a final assessment. Remediating vulnerabilities earlier in the SDLC reduces the likelihood of the security team finding flaws that could delay a release.

In addition, by giving developers feedback about security defects with every submitted build, they receive real-world training on how to build secure apps. Continuous security testing also reduces the passing back-and-forth of findings and results between security and development teams, which, again, reduces the risk of missing release deadlines.

But, you'll also need to make sure the tool you select provides accurate results that provide value to developers. Static-only testing that spits out pages of false positives will quickly erode any goodwill you've established with developers or your DevOps team.

AS YOU BEGIN TO STANDARDIZE YOUR PROCESS, REMEMBER THAT THE END GOAL IS SOME SEMBLANCE OF CONSISTENCY IN COVERAGE, RESULTS, AND REPORTING:

- Consistent coverage ensures you're spending your time wisely and evaluating the security of an app across the entire mobile attack surface
- Consistent results help you track progress against your program's objectives
- Consistent reporting helps the multiple teams involved in the mobile app development process understand what needs to be done to reduce risk in an enterprise's mobile apps

WHY IS CONSISTENCY SO IMPORTANT?

MAKES YOUR TEAMS MORE EFFICIENT

MAKES YOUR PROGRAM MEASUREABLE

KEEPS TEAMS FOCUSED ON ACHIEVING THE SAME OBJECTIVE

TECHNOLOGY

To start, building a mobile app security program requires setting your objectives and metrics, documenting your security standards and methodology, and assembling your team. From there, you need to arm your team with the tools they need to thoroughly test your organization's mobile apps. The size of your team can affect what tools will work for you. If you test one or two mobile apps each year, you might choose different tools than if you test 40, 50, or even hundreds of app releases annually.

1 SET OBJECTIVES AND ESTABLISH METRICS

2 DOCUMENT SECURITY STANDARDS AND TESTING METHODOLOGY

3 ASSEMBLE YOUR TEAM

4 EVALUATE AND IMPLEMENT TOOLS

TECHNOLOGY
OPEN-SOURCE TOOLS

Make no mistake — there's a steep learning curve for many of the open-source mobile app security testing tools listed below. In addition, some of the tools are not updated regularly, and technical support is unavailable. So using the tools effectively requires a certain level of technical acumen. But experienced analysts may be able to meet the needs of the business, at low testing volumes, with a collection of open-source tools.

A SAMPLING OF OPEN-SOURCE SECURITY TESTING TOOLS FOR MOBILE APPS:

Santoku: A virtual machine that contains a number of open source tools specific to mobile application security testing, forensics and data recovery, and malware analysis.

Drozer: Identifies security vulnerabilities in Android apps and devices and supports the use and sharing of public exploits.

Mobile Security Framework (MobSF): A penetration testing framework including static and dynamic analysis.

Frida: An analyst can use Frida to inject JavaScript snippets into native Windows, Mac, Linux, iOS, and Android apps.

Mitmproxy: Allows a user to intercept and modify requests and responses exchanged between an app and backend services in order to inspect any data transferred.

Radare: A reverse-engineering framework used to analyze and inspect iOS and Android binaries.

TECHNOLOGY
COMMERCIAL TOOLS

For teams that are new to mobile or don't have the requisite expertise, I highly recommend a commercial tool. A commercial tool will drive consistency, reduce on-boarding time for analysts, and make setup of the testing environment easier.

CONFIGURING A TESTING ENVIRONMENT IS A TIME-INTENSIVE, FRUSTRATING PROCESS THAT REQUIRES, AT LEAST, THE FOLLOWING:

- Jailbreaking or rooting a test device
- Setting up ad-hoc Wi-Fi networks for various network attacks
- Reverse-engineering app binaries to evaluate source code

MOST COMMERCIAL SOFTWARE LICENSES WILL ALSO INCLUDE SOME LEVEL OF TECHNICAL SUPPORT. THIS HELPS IN TWO WAYS:

- First, vendor tutorials and resources will teach an analyst how to use the tool and ingrain in them a process for evaluating the security of mobile apps.

- Second, if an analyst experiences problems trying to use the tool, help is just a chat-box or phone call away (rather than at the end of hours spent troubleshooting independently).

Need help evaluating commercial mobile app security testing tools? Download the evaluation guide at www.nowsecure.com/go/evaluation-guide-program

TECHNOLOGY

IDEAL MIX OF TOOLS FOR AUTOMATED AND MANUAL TESTING

More sophisticated mobile app security programs require a combination of tools to increase efficiency, reduce turnaround time, and deliver consistency in reporting from one analyst to the next. Advanced open source tools should certainly be used for areas of test coverage where automation is not possible, which may include web-services/API penetration testing and manual analysis of reverse-engineered code to identify weaknesses.

A LARGER, MORE MATURE MOBILE APP SECURITY TEAM NEEDS A TOOLSET THAT CAN DELIVER THE FOLLOWING:

Automated testing for basic coverage: Basic coverage should at least include checks for improper certificate validation or hostname verification, insecure storage of sensitive data, and personally identifiable information (PII) saved in device logs.

Manual testing across the entire attack surface: Security analysts will need tools for forensic analysis and data recovery, network analysis, web penetration testing, reverse-engineering, and code analysis.

Flexible reporting: Customizable reporting will maintain consistency while allowing your team of experts to adjust or add new findings beyond those uncovered by automated testing. Ideally reporting maps directly to your program's requirements, as well as, standards such as CVSS, the OWASP Mobile Top 10, and the Common Weakness Enumeration (CWE) framework.

TECHNOLOGY
TESTING AUTOMATION

No matter their size, a mobile app security team can benefit from automating aspects of their security testing. It all depends on volume. A smaller team can't test as many apps as a larger team can. If testing demands surge, automated testing reduces the amount of time it takes to test an app, gives developers feedback more quickly, and frees up analysts for more in-depth mobile app penetration testing. Larger organizations need automated security testing as part of their continuous integration and delivery practices and need it to integrate with other aspects of their DevOps toolchain.

Choose mobile app security testing automation tools carefully. Look for a combination of static and dynamic analysis capabilities. Some automation tools require source code and/or are only capable of static analysis, which can lead to more false positives. When test results continue to sound the same false alarms, the security team can lose credibility with developers. A combination of static and dynamic analysis helps filter out false positives. I tend to think of dynamic analysis as a way to confirm the results of a static check.

FOR EXAMPLE, IF THE OBJECTIVE OF A CHECK IS TO DETERMINE WHETHER AN APP IS LOGGING SENSITIVE DATA:

- Static analysis will flag whether or not certain debugging and logging flags are enabled or disable
- Dynamic analysis will actually run the app and search device logs for sensitive values (e.g., user credentials)
- In this example, if dynamic analysis finds user credentials in log files, there's no denying the existence of the issue

SECURITY TESTING TECHNOLOGY IMPERATIVES

AUTOMATED TESTING

MANUAL TESTING

FLEXIBLE REPORTING

TECHNOLOGY
TOOLS FOR DEVELOPERS AND DEVOPS

If you plan to introduce mobile app security testing into your development team's technology stack, you need to make it as seamless and easy as possible. In general, development and DevOps teams worry that mobile app security testing will complicate their processes and slow them down. By now, most of them are familiar with the automation of functional testing. Now, mobile app security testing technology has advanced to the point that it can also be automated and integrated with continuous integration and continuous delivery practices.

Whatever tool you choose should make tying automated testing into your build cycle easy, simple to set-up, and allow an admin to "set it and forget it." Output from the continuous security testing tool should also automatically log findings in the development team's favorite issue-tracking system.

KEEP THE FOLLOWING IN MIND AS YOU EXPLORE AUTOMATED MOBILE APP SECURITY TESTING SOLUTIONS:

- Collaborate with the development team in identifying options, evaluating technology, and choosing the right solution.
- Make sure selection criteria include static and dynamic analysis capabilities and a low false positive rate.
- Look for a solution that includes detailed remediation instructions that include code examples for any identified defects.

The volume of mobile apps developed and needing testing will only increase for the foreseeable future. Make sure any tools you choose to incorporate into your testing environment can scale with your development team's productivity.

NOW WHAT?

Taking your program to the next level

As you launch and manage your mobile app security program, remember that the entire enterprise needs to work together to make mobile apps more secure. Reducing friction by integrating the program with existing development and DevOps tools and workflows makes it that much easier for stakeholders to get on board.

What you do next will depend on the maturity of your program. NowSecure has a whole lot of experience supporting our customers as they launch, maintain, and improve mobile app security programs. We've helped enterprises build and implement a program from beginning to end, or, helped them address one particular aspect of their program that challenged them.

If you're ready to take your mobile app security testing program to the next level, NowSecure can help:

- If you're ready to evaluate mobile app security testing technology for use at your organization and/or upgrade current tools, download the NowSecure Evaluation Guide for Mobile App Security Testing for help in your search.

- If you don't yet have enough resources to build a complete internal program, learn more about our mobile app security services.

NEED HELP EVALUATING MOBILE APP SECURITY TESTING TECHNOLOGY?

Download our the NowSecure Evaluation Guide for Mobile App Security Testing at www.nowsecure.com/go/evaluation-guide-program/

NowSecure is the mobile app security technology company enterprises trust to help them deliver secure customer experiences through mobile apps and manage risk associated with mobile-centric workforces using dual-use devices. We deliver mobile app security testing, mobile app vetting, managed services, incident response, and compliance solutions.

(312) 878-1100
info@nowsecure.com
www.nowsecure.com