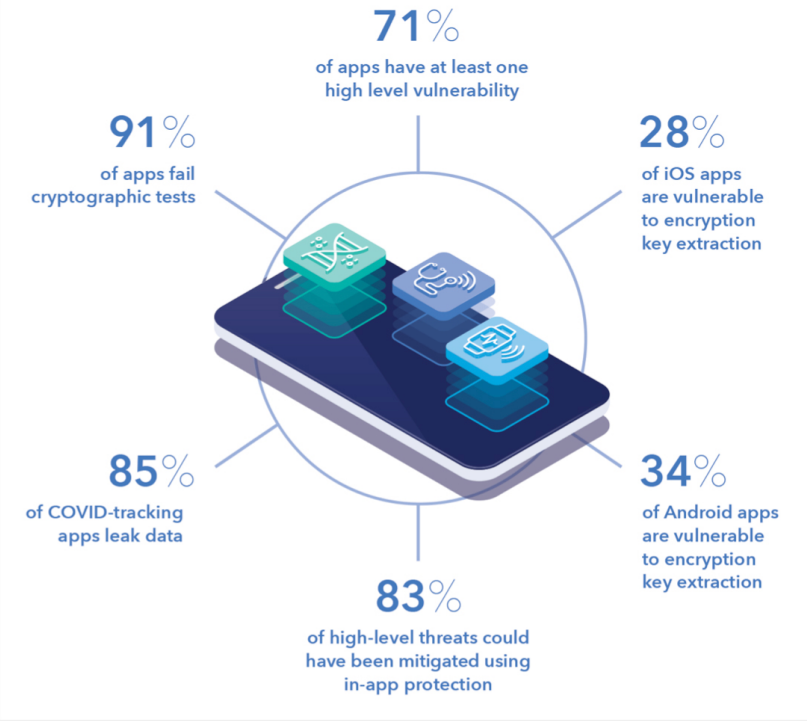

ZIMPERIUM®
ADVANCED MOBILE SECURITY
Mobile App Security:
The Only Guide You'll
Ever Need

The average person spends over four hours a day on mobile, and the average smartphone is home to nearly 100 apps. It's clear mobile apps claim a large and growing role in people's lives—likewise for businesses. Most companies with an online presence deploy an app; for many it's their core business driver. However, businesses aren't alone in recognizing the revenue potential of mobile apps, cybercriminals have as well.

Each year, attacks on mobile applications steadily increase. The costs to businesses in terms of lost customers, compensation, reputational damage, and regulatory fines make lax mobile app security a significant risk factor. Ensuring effective mobile app security, however, presents a major challenge for many organizations. For example, our proprietary research found that 71% of mHealth apps contain at least one serious vulnerability.



Our 2020 report on mHealth apps found that some of the most serious risks to apps were also the most prevalent.

What is mobile app security?

Mobile app security is the general term for the processes, strategies, and tools used to secure mobile applications from threats. It involves the entire application development ecosystem, including secure design, assessing applications for security flaws, mitigation measures, and defense protections. Mobile app security is an essential function in the app development process, and most app distributors either build an in-house app security team or engage third-party support. Some rely on a mix of the two.

Mobile app security encompasses the various mobile devices and platforms, like Android, iOS, and Windows Phone, that host mobile applications. It needs to be flexible enough to handle the specific vulnerabilities and contextual limitations in which the app appears (such as a lack of hardware security support). A holistic approach to mobile app security includes building security into the development process to create a security-by-design approach. Regular penetration testing, patching, and decommissioning are also necessary to maintain an app's security throughout its lifecycle.

How does mobile app security work?

Due to the large array of attack vectors, points of vulnerability, operating systems, and devices that affect how an app is attacked, mobile app security needs to adopt a multi-pronged and overlapping approach. This is done by prioritizing best practices during app development and having the necessary in-app instrumentation for real-time visibility and on-device protection after release. Integrating security across the application lifecycle allows security and risk teams to proactively identify risks and respond to threats in a timely fashion.

A diagram showing some of the major government departments affected by the SolarWinds hack. Image courtesy of David Wolpoff via TechCrunch.

It is virtually impossible to eliminate every vulnerability in an app, but deploying a range of application security strategies can frustrate attacks and make it cost-prohibitive to continue. This means only an extremely determined attacker, likely working with a large team (Microsoft estimates over a thousand engineers worked on the 2020 SolarWinds software attack) and expensive equipment, will spend the time and resources necessary to break an app's security. In the vast majority of situations, this is completely unfeasible. In the event of a successful hack, properly encrypted data would still be unreadable. By placing sophisticated and difficult-to-overcome barriers in the way of hackers, a layered mobile app security strategy prevents and deters attacks.

Why is mobile app security important?

Mobile app security should form a major part of the security function for any organization that develops or distributes apps. It should also be considered an overall priority given its repercussions for a business' risk management and revenue security. Robust mobile app security impacts multiple critical areas.

Protects data and sensitive information

Data is a trillion-dollar currency, which apps store, send, and receive constantly. This makes them a highly lucrative target for attackers. Sensitive personal identifiable information (PII) and other valuable data including names, passwords, credit card details, and medical records can be easily accessed through hacked mobile apps. Mobile banking trojans such as Anubis and Ghimob, and other mobile malware, use various techniques to exfiltrate data including keyloggers, overlay screens, and exploiting accessibility services.

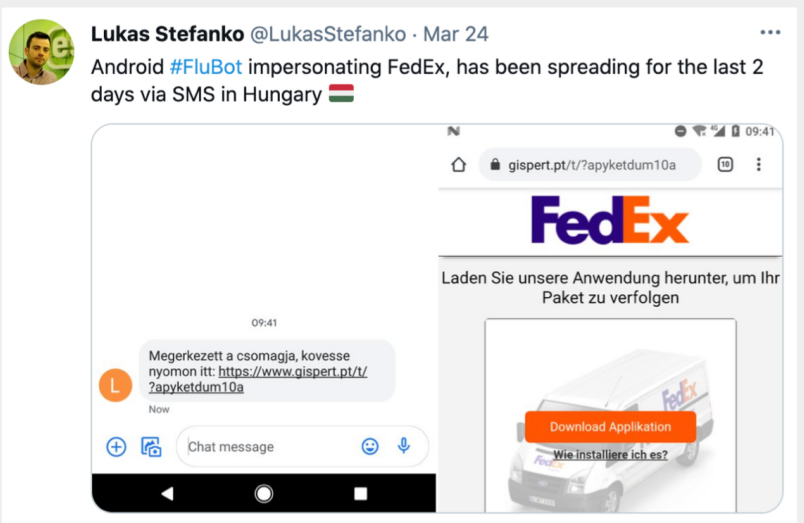
The consequences of breached data extend far beyond theft of information for businesses. Organizations face multiple regulatory violations and fines along with loss of customer trust. British Airways was hit with a £183 million fine for a web and mobile app breach that exposed personal data for nearly 500,000 users.

Prevents IP theft
Organizations spend billions on R&D to improve their services, gain a competitive advantage, and increase revenue. Cybercrime groups and state actors engage in industrial espionage to steal this valuable intellectual property.

Application code often includes proprietary algorithms and patented technology, which can be reverse engineered, copied, and utilized in competitor technology or sold. Any recourse requires lengthy legal processes—if possible at all. Mobile app security protects apps from compromise while also making code analysis difficult and cost-prohibitive.

Stops app hijacking
A common method of attack is to clone an app by reverse engineering it and creating a replica. Unaware users download the fake app and enter their private information, such as banking login details, which is captured by cybercriminals. Attackers also inject malware into app code then repackage and distribute it under guise of the legitimate app, or even use parts of legitimate app code inside their own malware. This is what happened when code from the popular Telegram messaging app was incorporated into the ToxicEye remote access trojan and used Telegram's own infrastructure to communicate with the attackers' servers.

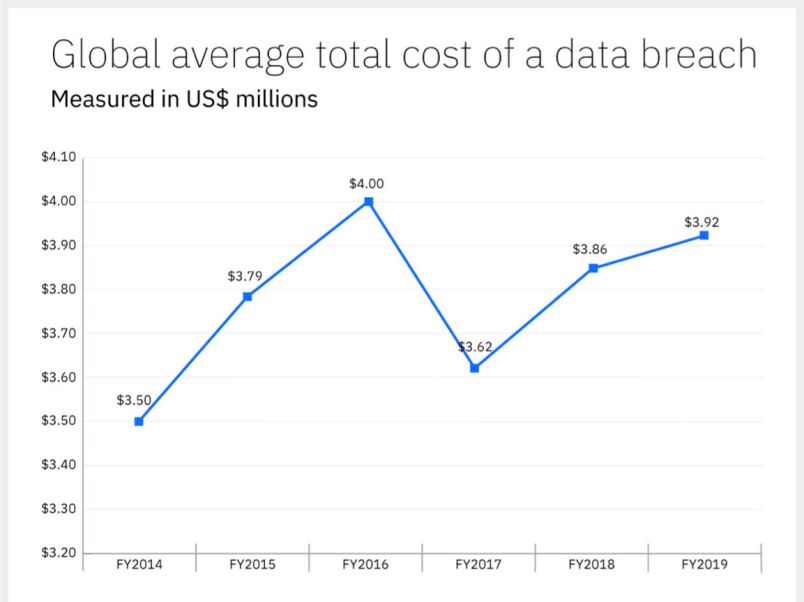
This hijacking of apps not only leads to data being stolen but can damage brand reputation. Victims of hacking or fraud by a cloned app may associate the experience with an organization's name and branding, despite the company's lack of direct involvement.



Tweet from an ESET security researcher about spyware disguised in a fake FedEx app. Image courtesy of Lukas Stefanko, ESET.

Prevents revenue loss
For most organizations, the major risk from inadequate mobile app security comes down to their bottom line. The average data breach loss in the U.S. was $8.19 million in 2020, and the global average was $3.86 million, with costs averaging out to nearly $150 per record compromised in the breach.

The biggest factor in the cost of data breaches is lost business. Lost business costs arise from increased customer churn, increased customer acquisition costs, and disruption in operations due to system downtime. Additional data breach-associated costs come from consumer compensation, higher insurance premiums, and regulatory fines. These have a lingering impact on available cash for investment in other areas of the organization, leading to even more lost revenue.



Graph showing the lasting impact of a data breach. Image courtesy of IBM Security.

Data breaches are just one of the sources of lost revenue stemming from compromised app security. As mentioned earlier, IP theft can have serious business consequences. Organizations also take significant financial hits when hackers bypass authentication protocols or use other methods to access premium services for free, for example with gaming or streaming apps.

Instills brand confidence

Brand loyalty is essential to long-term revenue stability. Unfortunately, consumers in the digital age are reactive to cybersecurity incidents, with 25% of U.S. and 44% of U.K. consumers saying they would never again do business with a brand after a data breach. In certain sectors, consumer trust around data security is even more important, with research from Carnegie Mellon University indicating that customers were far more likely to leave their bank in the six months following a data breach.

Building a reputation for being attentive to security and caring for your customers' data encourages brand loyalty and uptake on new product features and app updates. However, app developers can only maintain this confidence through constant vigilance and effective mobile app security.

Mobile app security risks and threats

Security risks and threats are an unfortunate fact of life for mobile applications. A mobile application is a complex undertaking with countless lines of code, much of which will be third-party. Mistakes in even a fraction of the code equate to hundreds or even thousands of flaws per app, increasing the likelihood of security issues. Additional factors include inherent problems in programming languages, vulnerabilities in third-party libraries, insecure coding practices, operating system security flaws, and running on a jailbroken or rooted device.

To track all of these different risks and security threats, the Open Web Application Security Project (OWASP) produces a running list of the biggest threats to mobile security, known as the OWASP Mobile Top 10. The list forms a core guideline to build more secure mobile apps for developers, and the businesses and consumers who use them. Every app developer should be familiar with these top mobile security risks, how they affect mobile apps, and what can be done to mitigate them.

OWASP Mobile Top 10 vulnerabilities

M1: Improper Platform Usage
Platforms and operating systems (such as Android or iOS) have built-in features to improve the function of applications and their security. When these features are not used properly, they create vulnerabilities that hackers can exploit to attack the app. For example, attackers can use permissions granted to an app to inject inputs or execute commands and steal data. Similarly, if an app fails to correctly use security features such as iOS keychain, then secret data can be exposed.

M2: Insecure Data Storage
Mobile apps often store all kinds of PII and data about their users, from email addresses and passwords to credit card details or their route to work. If this data is not stored securely, hackers can access and exploit or steal the information for several purposes. Weak encryption, compromised devices, and insecure access protocols all contribute to insecure data storage. Deploying mobile app security to keep sensitive data safe should be a priority for all app vendors.

M3: Insecure Communication
Mobile applications connect users to the digital world in a multitude of ways. As such, they constantly send and receive data. Mobile apps are vulnerable to insecure communication if transport security is not sufficient or is implemented incorrectly. This lets hackers eavesdrop on transmissions to steal sensitive data from a user's device or change the data sent to it.

M4: Insecure Authentication
Secure authentication is difficult for mobile apps as they are limited by their device form factor and passwords often consist of 4 or 6-digit PINs, especially if they authenticate locally. An attacker that bypasses app authentication procedures could anonymously execute functions and transactions within the app or backend server. For example, an attack on a smart home system could allow a hacker to alter temperature settings or even speak through the device's connected speakers.

M5: Insufficient Cryptography
As discussed in our deep dive of the OWASP Mobile Top 10, insufficient cryptography is when the data on an app has inferior cryptographic protection. This occurs when weak or compromised encryption is deployed or in instances when the vendor fails to provide extremely secure encryption for the most valuable data (such as medical records) using crypto key protections such as white-box cryptography.

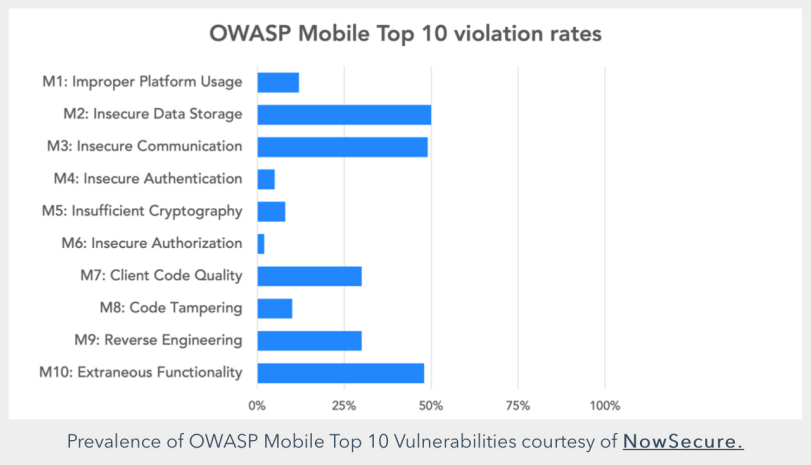
M6: Insecure Authorization
This refers to an attacker's ability to manipulate authorization procedures to escalate app privileges and perform functions that they should not have permissions to access. While app developers may seek more permissions to deliver greater functionality, these can also become a significant risk without strong authorization procedures. Easy access to permissions or over-privileged users could even allow hackers to navigate to databases and directly steal data or take over other accounts.

M7: Client Code Quality
Completely avoiding coding flaws is almost impossible due to the sheer amount of code present in an app. However, some bugs are more serious than others and can create major vulnerabilities if not addressed. An effective mobile app security function will constantly perform penetration testing and bug hunts to seek out critical code flaws.

M8: Code Tampering
Code tampering is when an attacker modifies an application's code and creates an altered or fake version. These cloned apps can serve many purposes, such as accessing premium gaming or streaming services for free, stealing an online shopper's bank details, or installing trojans that infiltrate backend servers and IT systems. Whatever the use, code tampering means a loss of control and revenue for the app vendor.

M9: Reverse Engineering
While not necessarily critically dangerous by itself, the foundations of nearly all attacks on mobile applications arise out of or utilize reverse engineering. As we explored in part three of our series on the OWASP Mobile Top 10, reverse engineering involves an attacker running the application through debuggers and other tools to analyze the app for its source code, algorithms, libraries, and other elements. With this information, they can identify other important elements of the app's functionality, discover vulnerabilities, steal or copy information, and find other ways to bypass protections.

M10: Extraneous Functionality
This threat to mobile app security concerns elements such as log files, switches, or back doors that get left behind following development or are poorly protected after an update. Depending on the type of functionality, If discovered, bad actors could execute admin-level actions or even gain access to the system's backend.



Prevalence of OWASP Mobile Top 10 Vulnerabilities courtesy of NowSecure.

Top mobile app security best practices

Mobile app vendors face a vast array of challenges to keep apps, users, and their data safe. To counter these risks and threats, it is vital for app developers and security professionals to employ mobile app security best practices. Here we'll take a look at a number of actions and preventive measures that can be put in place to improve the security of mobile applications, including:

- Follow secure app design best practices
- Use strong encryption
- Manage keys securely
- Perform risk analysis
- Code obfuscation
- Deploy tamper-detection technologies
- RASP
- Encrypt communications
- Enforce session logout
- Use multi-factor authentication
- Perform regular penetration testing
- Regularly patch vulnerabilities
- Be cautious with third-party libraries

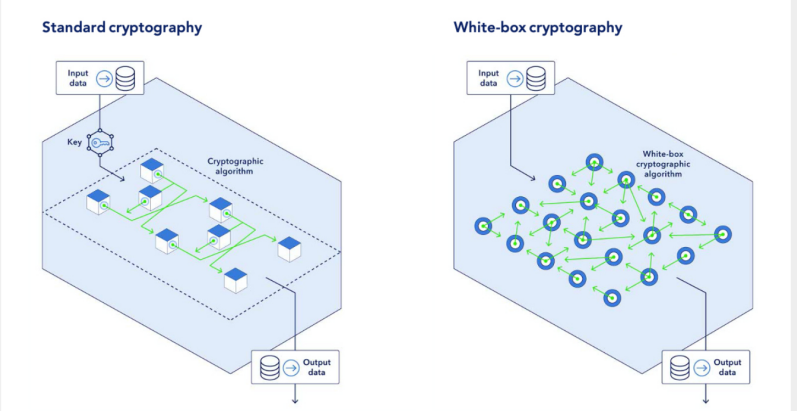
Follow secure app design best practices
App developers should take a security-by-design approach, implementing app design best practices from the onset. For example, avoid the local storage of critical information unless necessary, ensure all data is subject to input validation, and confirm that passwords are only allowed to be stored when strong encryption is present.

Following a DevSecOps framework will build security into the development lifecycle.

Use strong encryption
Encryption is the process of translating a message or information into ciphertext that only authorized parties with access to a secret key or password can read it. Generally, all sensitive data held by an application should be encrypted. Not only should it be encrypted, but it should also use an extremely secure cryptographic standard, such as AES. This ensures that even if an attacker does get through an application's defenses, the data they steal is unreadable.

Manage keys securely
Even the most complex encryption is useless if an attacker gets a hold of the cryptographic keys. As encryption algorithms get more difficult for hackers to crack, they increasingly resort to the simpler target of stealing the keys used to encrypt and decrypt mobile app data. This can be done in a variety of ways, such as employing dynamic analysis to identify them when in use or through side-channel attacks that work off physical effects produced by a device.

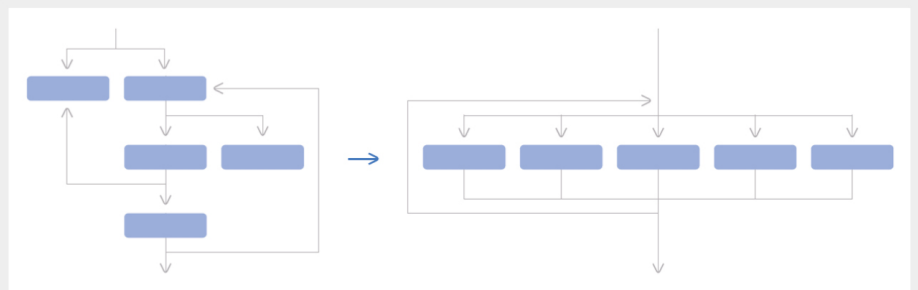
Mobile apps cannot be guaranteed access to hardware-based key protection or may be running in compromised environments such as jailbroken or rooted devices. To manage keys securely, it is essential that they employ strong software-based key protection such as white-box cryptography. This ensures that keys are kept keys safe, even when in use.



A depiction of white-box cryptography.

Perform risk analysis
Like any area in business, actively evaluating the risks posed to the organization by your mobile apps is vital for properly evaluating your security needs. Vendors of apps with high value intellectual property and sensitive data, and brands that prioritize their customers' security and data privacy will have different mobile app security requirements than lower value apps. A risk analysis of your mobile application deployments will reveal the major areas of concern for your particular assets, the potential financial or reputational repercussions of a breach, and what your security team should focus on.

Code obfuscation
Code obfuscation is a security strategy that deliberately disguises code to frustrate and delay hackers in their attempts to understand how an application's code works. There are numerous methods involved in code obfuscation, such as inserting decoy logic or nonsense statements, encrypting segments of the binary code, and obfuscating the control flow. These techniques aim to confuse attackers and cost them more time and resources, making it economically unviable for them to try and break in.

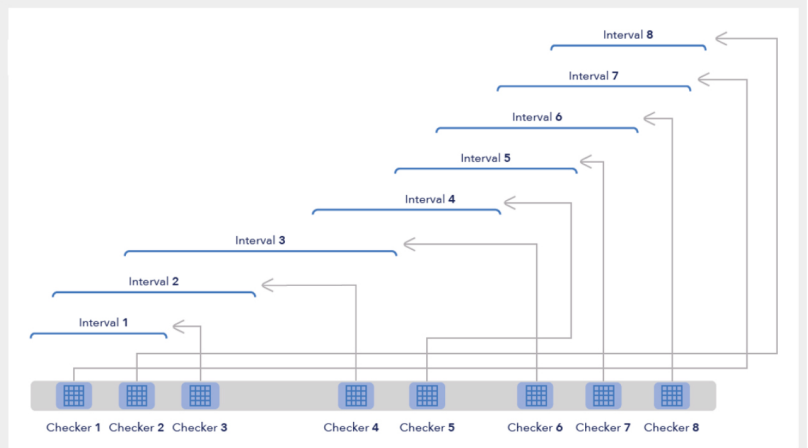


Flattening the control flow, a common code obfuscation technique.

The level of code obfuscation deployed should depend on your risk evaluation. A free obfuscator tool may be good enough for low value apps. Most, however, will require a higher grade obfuscation tool. As with any app security strategy, app performance and usability must be balanced against security needs. Get some tips on how to optimize code obfuscation and performance.

Deploy tamper-detection technologies

A common method for attackers to discover an app's weak points is to run it through debuggers or emulators and modify variables to see what happens. They also may try to insert code to bypass device and system security controls, deploy malware, or otherwise hijack the app for malicious purposes. To protect against these scenarios, applications can be equipped to monitor their runtime environment and any attempts at manipulating their code. For example, by detecting if a device is rooted or the presence of debuggers, or using integrity checkers to test segments of code for tampering. They can then deploy a pre-programmed response to keep themselves secure.



An illustration of integrity protection using checksum checkers.

Deploy RASP

While an application is running, more of its code and even its secure keys will be called into use, allowing hackers to observe them. Runtime application self-protection, often referred to as RASP, is a safety mechanism that equips applications to detect when they are being analyzed or attacked while in use. If they detect this, they can execute a predefined defense action, such as shutting down, deleting sensitive data, or entering into a sandbox mode.

Encrypt communications

In addition to the data held on an application, encryption needs to be applied to the data being sent and received. With many users running their applications over insecure public WiFi, such as at airports, coffee shops, or conference centers, the potential for serious data breaches through unencrypted communications becomes even larger.

Enforce session logout
Forcing an application to log a user out after a certain period of inactivity reduces the potential for a hacker to hijack an app and use a valid session ID to access the app with all the user's permissions. Without a session timeout, any attack which gains access to a device will automatically bypass any authentication controls, rendering a large part of your mobile app security irrelevant.

Use multi-factor authentication
Multi-factor authentication (MFA) is when a user must provide two or more forms of identification to access an app, such as a password and a fingerprint or time-limited code. This limits the possibility of bad actors gaining access to an app's data through a stolen or brute-force password. MFA is also an important part of data security legislation, such as Europe's GDPR.

Perform regular application security testing
Application security testing (AST) analyzes application source code and tests applications for security vulnerabilities. AST is a broad term that covers various testing methodologies:

- Static AST (SAST): Inspects app source code before it is compiled to identify security flaws at the code level. SAST is often used in the initial stages of development.
- Dynamic AST (DAST): Analyzes app behavior as it runs in production to find vulnerabilities. DAST simulates attacks against an application and analyzes the application's reactions to find security weaknesses.
- Interactive AST (IAST): Inserts agents into the application that analyze the app's code and behavior as it interacts with manual or automated tests. IAST tests functionality at certain points as defined by the tester rather than the entire codebase.

Penetration testing finds and exploits vulnerabilities in system architecture. It generally is a manual process performed by experienced pen testers that try to think like a hacker and get past the security functions that have been deployed. Testing should be an ongoing process to identify vulnerabilities before an attacker does and help you prioritize for remediation. There are many free tools that help automate the mobile app security testing process, such as:

- Quick Android Review Kit (QARK)
- The Mobile Security Framework
- NMap (for port scanning)
- OWASP Zed Attack Proxy (ZAP)
- Devknox (like spellcheck for security-testing)
- Metasploit (a multi-purpose pen testing tool)

Regularly patch vulnerabilities
Most applications are published with bugs, despite the best efforts of security teams to identify and fix them. Moreover, new threats emerge each day. Vulnerabilities revealed by pen testing, bug bounty programs, and other sources should be remedied through regular updates. Unfortunately the patch rollout process is more complicated for mobile apps as it requires making a new build, submitting it to the app store and waiting for it to be accepted, and then hoping users update. This underscores the importance of secure development processes and application protection solutions that mitigate vulnerabilities until they can be patched.

Be cautious with third-party libraries
Third-party libraries and SDKs expedite the development process enormously, offering important integrations and allowing app-makers to offer basic functions, like a login page or notifications, without having to build them from scratch. It's estimated that third-party and open-source libraries make up an average of 60% of the code in a mobile app. However, many libraries contain vulnerabilities. In fact, a recent Veracode report found that 70% of applications use at least one open source library with a security flaw. It's recommended that app developers limit their use of third-party libraries, removing any that are unnecessary. For the rest, application protection

techniques like code obfuscation can make it so a hacker can't discover the third-party libraries you are using to exploit their vulnerabilities.

Find the right partners

Mobile app security is a core part of all digital security and should be an essential consideration for risk management and revenue security. To effectively protect mobile applications, a varied and multi-pronged approach needs to be adopted to counter the many possible attack vectors hackers use to gain access and steal data from an app.

This layered approach to mobile app security needs to be deployed throughout the app's entire lifecycle, including adhering to design best practices in development, anti-reverse engineering and anti-tampering protections, regular testing, and patching post-release. Understandably, adding the weight of mobile app security to an already-burdened infosec team can stretch resources, and building a new team in-house is an expensive exercise. That's why it's critical to find the right technologies and expert partners to support a holistic mobile app security strategy.

Zimperium's Mobile Application Protection Suite (MAPS) is the only unified solution that identifies security, privacy and compliance risks during app development and protects apps from attacks while in use. MAPS includes:

zScan to help your mobile app development organization to discover and fix compliance, privacy, and security issues within the development process before you publicly release your apps;

zKeyBox to protect your secrets and keys so they cannot be discovered, extracted, or manipulated;

zShield to protects the source code, intellectual property (IP), and data from potential attacks like reverse engineering and code tampering; and

zDefend, which is an SDK embedded in apps, to help detect and defend against device, network, phishing and malware attacks while mobile apps are in use.

To learn more about protecting your mobile apps, contact us and talk to one of our security experts.

ZIMPERIUM.

4055 Valley View Dallas, TX 75244
844.601.6760
info@zimperium.com