ZIMPERIUM®
MOBILE THREAT DEFENSE

Top 5 Ways Hackers Attack Your Apps

Table of Contents

INTRODUCTION

Securing apps against attacks and manipulations is a full-time job.

It is a full-time job since cybercriminals engage in hacking 24 hours a day to find innovative attack methods to uncover mobile app secrets and private user data.

To defend against constant attacks, application developers must implement security throughout the entire app development lifecycle to secure data and limit risk but at a low cost to the user experience. It's even more crucial for mobile app developers and security professionals to measure this risk and enable detection since mobile devices are incredibly vulnerable.

Mobile app security is challenging since the operating systems evolve so quickly, and there are so many apps. As mobile devices process more data, so are the chances of exposing secrets.

This paper describes the top five ways hackers attack apps looking for secrets. We describe particular attacks, the tools used in the attacks, and what specific information the bad actor hopes to obtain. We also look at how to protect against each attack and cite real-world examples of the attacks.

60%

Percentage of online fraud attributable to mobile platforms[1]

76%

Percentage of mobile apps that have flaws allowing hackers to steal passwords, money, and texts[2]

80%

Percentage of mobile fraud occurrences that use mobile apps instead of mobile web browsers[1]

89%

Percentage of mobile app vulnerabilities for which hackers do not need physical device access because they can exploit the device remotely via malware[2]

1 RSA. 2018 Current State of Cybercrime. https://www.rsa.com/content/dam/premium/en/white-paper/2018-current-state-of-cybercrime.pdf

2 Positive Technologies. Vulnerabilities and threats in mobile applications. https://www.ptsecurity.com/ww-en/analytics/mobile-application-security-threats-and-vulnerabilities-2019/

With that in mind, let's look at the first attack set, which is collectively authentication attacks.

AUTHENTICATION ATTACKS

STATIC REVERSE ENGINEERING

In this attack, the bad actor attempts to gain information about an app via its source code, without necessarily running the app.

What tools does the hacker use?

- Frida, Xposed, Substrate, QBDI, scriptable debuggers such as GDB/LLVM/IDA
- For hooking and swizzling: Frida, CaptainHook, MobileSubstrate, Cycript, Cynject
- IDA, Ghidra, BinaryNinja, Hopper, Radare2, JEB, jadx, apktool; dextra, jtool, joker

What is the primary goal of this attack?

There may be multiple purposes, such as:

- Understanding the implementation of an algorithm to replicate it or abuse it
- Finding embedded credentials, URLs, or configurations of an application;
- Identifying and bypassing security checks
- Vulnerability assessment

How do you defend against it?

The most common defenses are:

- Code obfuscation via methods ranging from code to data protection
- Tricks to thwart or break the analysis tools, such as malformed data that is correctly handled by the device but unhandled by the analysis tool

What are real-world examples?

Hackers uncovered communication and credential information by reverse engineering the Tesco Bank mobile app leading to a £2.5M loss from over 9000 accounts overnight.

Researchers hacked the Bitfi eWallet by using a local exploit to extract the memory allowing private keys to be generated and the funds stolen.

CREDENTIALS PHISHING

Here, the hacker attempts to acquire sensitive information (the credentials) by appearing to be a trustworthy entity, such as the user's employer.

What tools does the hacker use?
Tools for developing fraudulent sites trying to imitate the legit sites; tools for massive mail/SMS distribution; private tools such as malicious apps aiming to steal two-factor authentication tokens.

What is the primary goal of this attack?
Gain user credentials, e.g. user/password combinations that may also be usable in other sites.

How do you defend against it?
Defensive measures include two-factor authentication and the proper use of SSL/PKI.

What are real-world examples?
Hackers discovered the RSA private keys used to sign QR codes after hacking the Manchester Metrolink mobile app. After finding the private keys in the mobile app, the attackers created QR codes for free rides on the Metrolink and distributed the codes on the Internet.

North Korea was linked to a massive phishing campaign targeting foreign ministries of three different countries. Researchers found that the threat actors used several subdomains and web pages to impersonate the French Ministry for Europe and Foreign Affairs (MEAE) to trick its victims.

INFO GATHERING

This attack is characterized by its purpose, which is to gather private information - usually intended for strategic or other importance to a business, government or other entity - to achieve a competitive advantage.

What tools does the hacker use?
Spear phishing campaigns are similar to those mentioned under credentials phishing, but are very targeted. Spear phishing seeks to gather information from an individual or to trigger the password recovery "protocol" on the victim's account and then attempt to replace the legitimate password with one from the attacker.

What is the primary goal of this attack?
Gain vital information such as email addresses, passwords, phone numbers, bank account details, etc. Attackers may use the stolen information or social engineering to impersonate the victim (e.g. by sending fraudulent emails under the victim's name).

How do you defend against it?
Enforcing the use of two-factor authentication mechanisms is key, along with proper use of HTTPS in cooperative portals and implementing appropriate BYOD policies.

What are real-world examples?
Microsoft saw hackers launch 2,700 attempts to identify specific target email accounts, including those belonging to current and former US government officials involved in presidential elections.

POOR CODING

This attack aims to exploit an app that can be both legitimate and non-malicious but whose coding is poor, such that the app incorporates identifiable vulnerabilities.

What tools does the hacker use?
Because this approach exploits design and security errors, hackers may attempt to guess credentials; for example, if security is dependent on knowing a user's email address, the hacker can make educated guesses as to what the user's email address may be.

What is the primary goal of this attack?
Gain personal or sensitive information about the victim via mobile app or communication methods.

How do you defend against it?
There are several coding best practices to follow, including the OWASP Mobile Top 10. It's also best to enforce the use of high-quality passwords and two-factor authentication mechanisms.

What are real-world examples?
A major flaw in the Conservatives' official conference mobile phone application has made the private data of senior party members – including cabinet ministers – accessible to anyone that logged in as that particular conference attendee.

AUTHORIZATION ATTACKS

BOTNETS OR SOCKET-BASED

A botnet attack leverages the distributed computing power of a group of devices for tasks such as infiltrating a network to which one of the infected devices has access.

What tools does the hacker use?
The primary vehicles are trojan applications that embed code to scan networks that devices are connected to. Persistence is usually a requirement and the fundamental capability is to execute commands (received by the attacker) on the discovered local services. For IoT devices, in particular, the attacks are

usually carried out against vulnerable devices that let the attacker install malicious ELF files that will connect to the command and control infrastructure of the attacker.

What is the primary goal of this attack?
The aim is to access private network services (e.g. content delivery networks) and resources (e.g. databases). Even though services may not be exposed to the Internet, a device connected to the same network (and connected to the Internet) acts as an entry point for the attacker, shifting the attack surface from mobile to server.

How do you defend against it?
Antivirus software and a properly configured firewall can monitor and stop trojan applications and detect suspicious accesses to the local network.

What are real-world examples?
Android Dresscode Malware was embedded into more than 40 apps, and found in more than 400 additional apps on third party app stores. DressCode creates a botnet that uses proxied IP addresses, which Check Point researchers suspect were used to disguise ad clicks and generate false traffic, generating revenue for the attacker.

DressCode created a botnet using proxied IP addresses. Researchers suspect the botnet was used to disguise ad clicks and generate false traffic generating revenue for the attacker.

ROOT DETECTION BYPASS, FRIDA DETECTION BYPASS

This attack aims to enable a rooted device to run an application and bypass root detection. Bypass tools enable this by falsifying root detection and publishing a false reading.

What tools does the hacker use?
Frida and anti-anti-root detection mechanisms like Magisk are used to bypass root detection. Magisk includes a module called MagiskHide to hide falsify root detection. The root bypass reflects the cat-and-mouse battle in which the attacker and defender continuously try to find new ways to overcome limitations imposed by the other.

What is the primary goal of this attack?
Bypass the restrictions hindering the usability of root-based and instrumentation tools.

How do you defend against it?
The top priority is always to find new ways to detect the aforementioned tools and methods buy using a real-time mobile threat defense technology like Zimperium's zIAP.

What are real-world examples?
Magisk is an Android rooting application based on phh's SuperUser. Using Magisk Hide will allow you to bypass root detection and continue to run apps on a rooted device that would normally not allow that state.

LEGIT APPS WITH VULNERABILITIES

This attack seeks to identify and then exploit inadvertent vulnerabilities in legitimate apps from legitimate vendors.

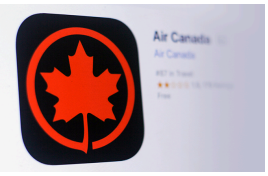
What tools does the hacker use?
Automated tools to scan for vulnerable or misused APIs (e.g. Hackode, Andriller, zANTI). The tools vary according to the vulnerability, which may reside in the main code of the application, in an imported third-party library, or one of the external services used by the application (e.g. a misconfigured Amazon instance or CDN).

What is the primary goal of this attack?
Gather sensitive and/or valuable information, or where possible to execute malicious actions.

How do you defend against it?
Monitoring and updating third-party SDKs and libraries in your application are critical. Continuously performing and refining internal testing procedures can help identify new security issues.

What are real-world examples?
The British Airways mobile app and ecommerce systems were hacked and later leaked more than 500,000 customer records including credit card numbers. Hackers infiltrated a third-party system to siphon the data.

Air Canada's mobile app was hacked and leaked more than 20,000 customer records after hackers infiltrated a third-party system used in the mobile app.



OBFUSCATION / PROTECTION BYPASS

The approach of this attack is to defeat the security measures put in place by app developers; specifically, the aim is to defeat attempts to obfuscate an app's source code and thereby make the app harder to reverse engineer.

What tools does the hacker use?

- Frida, Xposed, Substrate, QBDI, scriptable debuggers such as GDB/LLVM/IDA
- For hooking and swizzling: Frida, CaptainHook, MobileSubstrate, Cycript, Cynject
- Privately developed tools, Emulators, or hardware devices

What is the primary goal of this attack?

Gain access to the actual code of the application or to reverse engineer the implementation of a specific algorithm.

How do you defend against it?

- Proper use of strong protections, properly implemented encryption, combined with non-trivial obfuscation can be effective in hindering obfuscation bypass
- Known obfuscation and protection techniques include control flow flattening, opaque predicate insertion, virtual machines obfuscation, constant unfolding, and hardware bindings insertion
- Embedding emulator, debugger and file system detection software, like zIAP, in your app can be effective in stopping real-time attacks

What are real-world examples?

Dissecting Mobile Native Code Packers. As mobile malware advances to the levels of desktop malware, it's not uncommon to stumble upon protected APKs while analyzing malware. Most of the times, the sample is simply obfuscated via classes/variables name stripping from the DEX file and/or strings obfuscation; but other times several layers divide the researcher from the original code.

SPOOFING AND TAMPERING ATTACKS

BANK IMPERSONATIONS

These attacks leverage malicious mobile apps designed in such a way as to trick users into thinking they are from a legitimate financial institution.

What tools does the hacker use?

- Specially crafted apps for graphically and behaviorally imitating legitimate banking apps
- Trojans running services and displaying overlays on top of legitimate banking apps mimicking legitimate banking applications
- Invisible activities abusing the accessibility services of the device to steal the inserted data (e.g. username, password, PIN)

What is the primary goal of this attack?

Criminals seek to gain account credentials and/or credit card numbers. Once an account is breached, criminals can execute transfers or payments after logging into the account or can sell the credentials to another party.

How do you defend against it?

Avoid exposing intents/services to unknown apps when possible. Also, take extreme precautions when registering a new accessibility service (Android) and check proper use of theme/visibility attributes by activities.

What are real-world examples?

Over 17,000 new samples of the Anubis Android banking malware targeted a total of 188 finance and banking applications. Anubis is able to take screenshots, record audio, send, receive, and delete SMS messages, steal contact lists and account credentials, open URLs -- potentially to download additional payloads.

XcodeGhost allowed a developer to distribute malicious apps in the App Store. The malicious apps upload the device and app information to its command and control server. Attackers can then send commands through this command and control server, telling it to perform send phishing alerts, monitor the pasteboard, or hijack the browser.

PAYMENT SCAMS

This attack focuses on hijacking data in motion. It is possible to gather data from a device from the pasteboard or received via SMS.

What tools does the hacker use?
Specially crafted apps that monitor the clipboard, SMS inbox or user's emails and notifications.

What is the primary goal of this attack?
Criminals seek to obtain bitcoin and other cryptocurrencies addresses and identify e-wallet apps installed on the device. Once identified, criminals will monitor the app activities like notifications, SMS, emails in order to steal information and funds.

How do you defend against it?
Avoid copying and pasting sensitive data on your device since many applications have access to your pasteboard. Take special precautions when installing a new app, especially if it tries to use the Clipboard system service or pasteboard. If you have a mobile management platform like InTune, MobileIron, or AirWatch, you can license an application analysis engine to monitor for abusive apps.

What are real-world examples?
The Metamask Malware on Android impersonates a legitimate service called MetaMask. The malware's primary purpose is to steal the victim's credentials and private keys to gain control over the victim's Ethereum funds.

MALVERTISING

This attack displays advertisements on mobile devices through malicious, rather than legitimate, methods as a path to monetization.

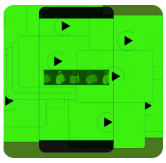
What tools does the hacker use?
Rogue ad SDKs that act maliciously based on GPS location, usage of the phone or other conditions.

What is the primary goal of this attack?
Criminals see to collect human clicks on ads and also load hidden scripts to automatically click ads that are invisible to users since they run in the background.

How do you defend against it?
Antivirus software can embed signatures to detect most of the malicious ads SDKs but for full protection you will need to install a mobile threat defense app (zIPS) on the device or install a mobile threat SDK (zIAP).

What are real-world examples?
Clicking Bot Applications use various methods to simulate user clicks to generate revenue. They can control what Ads are to be targeted, how the bots are controlled by their command and control (C&C) server and how to avoid the detection methods that are commonly used against them.

Video Ad Bots hijacked Twitter's MoPub ad platform to hide and create malicious clicks via millions of mobile devices.



REPACKAGING LEGITIMATE APPLICATIONS

In this attack, the hacker repackages a legitimate application to include malicious code and then resigns the app using a different certificate.

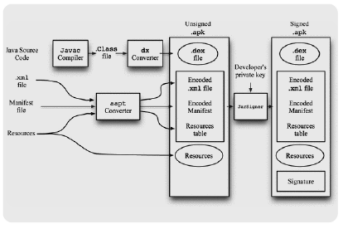
What tools does the hacker use?
On Android, apktool, and in rare cases custom-made injection utilities.

What is the primary goal of this attack?
Embed malware in a legitimate application and distribute it on third party stores (e.g. in countries where the official Google Play Store is blocked); infect as many devices as possible; distribute paid applications for free, specifically cracked to bypass the protection mechanism and with embedded malicious code.

How do you defend against it?
Identify repackaged applications by checking the signing certificate; however, this is ineffective if the official certificate with which the original application was signed has been leaked.

What are real-world examples?

Android-based smartphone banking apps were attacked to verify whether a money transfer could be made to an unintended recipient. The results showed that an attack of this kind is possible without having to illegally obtain any of the sender's personal information, such as the senders public key certificate, the password to their bank account, or their security card.



DYNAMIC ANALYSIS AND INSTRUMENTATION

This attack seeks to reverse engineer an app by running it in a virtual environment so that its inputs and outputs can be observed and analyzed (whereas static reverse engineering takes place when the app under attack is not running).

What tools does the hacker use?

- Frida, Xposed, Substrate, QBDI, scriptable debuggers such as GDB/LLVM/IDA
- For hooking and swizzling: Frida, CaptainHook, MobileSubstrate, Cycript, Cynject

What is the primary goal of this attack?

Obtain loaded/written files, allocated memory buffers, encrypted/decrypted information, and/or received/sent network packets; dynamically bypass security checks, forcing the call of specific functions with particular inputs.

How do you defend against it?

Common techniques include:

- Anti-instrumentation: instrumentation typically takes advantage of assembly trampolines to divert the execution of the original code; the code has to be modified at runtime and it would therefore differ from the one in the binary
- Anti-debugging checks: determining if the debugger is connected or if some internal state structures have been modified in a way that only a debugging session would do

What are real-world examples?

QBDI features to assess native code and speedup reverse engineering.

How hackers can use Frida to inject javascript into a vulnerable mobile banking application.

The Xposed Framework allows you to make changes to individual elements of the OS or any app without requiring the source code.

RACING RESOURCES ATTACKS

MALICIOUS PAYLOAD INJECTION, MAN-IN-THE-DISK

This attack takes advantage of careless storage protocols in third-party applications in order to crash a victim's Android mobile device.

What tools does the hacker use?
Malicious SDKs (e.g. SWAnalytic).

What is the primary goal of this attack?
Gather sensitive and/or valuable information, or where possible to execute malicious actions.

How do you defend against it?
Antivirus software can detect known-malicious payloads or SDKs. However, for full protection, you will need to install a mobile threat defense app (zIPS) on the device or install a mobile threat SDK (zIAP).

What are real-world examples?

Operation Sheep is a group of Android applications massively harvesting contact information on mobile phones without the user's consent.

COIN MINING

This attack makes unauthorized use of computing resources on the victim's device to assist the bad actor in cryptocurrency mining.

What tools does the hacker use?

Hackers will inject malicious JavaScript payloads by embedding (and usually hidden) WebViews in a legitimate application.

What is the primary goal of this attack?

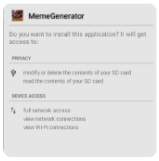
Using the computational power someone else's device to mine cryptocurrencies.

How do you defend against it?

Use an antivirus software that detects known-malicious payloads or SDKs.

What are real-world examples?

A free meme generator Android app also [happens to generate Monero cryptocurrency]. The app uses device processing power unbeknownst to the device owner.



INFORMATION LEAKAGE ATTACKS

INFORMATION STEALING

This attack method is to exploit vulnerabilities on a device such as bad communications, weak or missing encryption, and side-channel leakages (logs, files, backups, cloud). This attack seeks to gather information from sources such as the device's microphone, camera and messaging/SMS apps.

What tools does the hacker use?

- Spyware installed on the victim's device
- Network monitoring tools that detect information leaks

What is the primary goal of this attack?

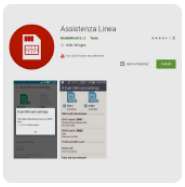
Obtain unique device identifiers, geolocation information, photos, emails, SMS/MMS, contacts, IM messages.

How do you defend against it?

Follow appropriate coding standards and best practices.

What are real-world examples?

Exodus Spyware was disguised as service applications from legitimate mobile operators. Once the malware was installed, it was able to continuously run and remain persistent on the victim's device. The spyware then downloaded data, messages and call logs from its victims.



Missing Link spyware was used to surveil senior members of Tibetan groups by sending malicious links in individually tailored WhatsApp text exchanges with operators posing as NGO workers, journalists, and other fake personas. The links led to code designed to exploit web browser vulnerabilities to install spyware on iOS and Android devices.

DATA LEAKAGE

This attack exploits back-end databases used to store data from mobile apps. An un-protected or inadequately-protected database can leak inadvertently expose user data and violate privacy laws.

What tools does the hacker use?

Hackers will sniff and intercept data using Bettercap, BurpSuite, CharlesProxy, Fiddler, or custom scripts.

What is the primary goal of this attack?

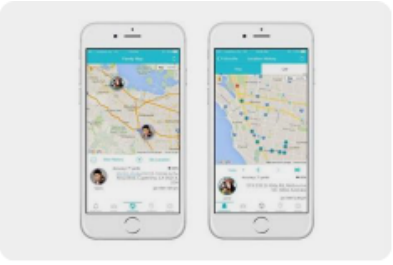
Obtain location data for specific devices and individuals.

How do you defend against it?

Encrypting data on the back and using appropriate security measures to secure back-end databases is vital.

What are real-world examples?

The Family Locator app was leaking the real-time locations of more than 238,000 users for weeks after the developer left a server exposed without a password.



NETWORK TRAFFIC MONITORING

This is a network-based attack that intercepts data (including potentially unencrypted data) sent and received by a mobile device across a network.

What tools does the hacker use?

- Wireshark, BurpSuite, Fiddler
- Sniffing and interception: Bettercap, BurpSuite, CharlesProxy, Fiddler

What is the primary goal of this attack?

The goal of monitoring a mobile device's network communication is to obtain sensitive data (encrypted or unencrypted) shared over the network, such as passwords, emails, links to private resources, or sensitive binary data such as photos or OTA updates. Hackers can also use a hijacked network connection to install code via a mobile vulnerability in order to gain persistence for further monitoring.

How do you defend against it?

A secure HTTPS connection with SSL pinning is useful and encrypting all of the data is advantageous. In order to detect a network attack, you will need to install a mobile threat defense app like zIPS to protect the device. If you want to reduce chances of a network attack on an unmanaged device, you can install mobile threat defense in your mobile app using an SDK like zIAP.

What are real-world examples?

Some Western countries, including the United States and Australia, have restricted Huawei from building next-generation mobile networks, citing concerns that its equipment may contain 'back doors' opening it to cyber-espionage.

SUMMARY AND CONCLUSION

Enterprises have been advocating mobile device use for years because of its superior user experience and efficiency. Mobile drives more conversions and, therefore, more revenue. Developers rush to deliver innovative apps to capture available market share but sometimes at security's expense. Mobile development teams invest many hours in designing and building intuitive mobile apps but less so on security. Many perform standard checks for static code analysis, authentication, and cryptography, however, they overlook

how critically important obfuscating the code and securing the application runtime environment is. Mobile apps rely on the OS to provide a secure foundation to operate correctly, but it is possible to compromise the entire device and to bypass native security functions. If a device is compromised, the security foundation underneath the mobile apps is compromised as well.

To reduce the risk of criminals exploiting vulnerabilities in your mobile apps and mobile devices it is critically important for you to enable them to defend themselves. Cybercriminals engage in hacking and reverse engineering at all hours. They use tools and attack methods, like those mentioned above, to uncover mobile app secrets, vulnerabilities, and private user data. To defend against constant attacks, application developers must implement in-app security throughout the entire app development lifecycle to secure data and limit risk but at a low cost to the user experience. In-app defense not only secures the data on the device but limits a compromised devices' ability to connect to your back-end systems.

Mobile apps with runtime security technology and in-app protection are capable of detecting malicious activity independently from native security features. The independent detection identifies real-time threats and reduces risk exposure while protecting transactions and data on both the app and server sides.

It is critically important for mobile app developers to install app shielding and real-time in-app protection. For comprehensive risk reduction and security, you must leverage these technologies in your mobile app for resiliency and obtain customized mobile attack data and intelligence.

CONTACT US

4055 Valley View, Dallas, TX 75244
Tel: (1) 844.601.6760
info@zimperium.com
www.zimperium.com