Mobile Application Security Report 2015

BY

Author : James Greenberg

AppSec Labs Ltd.
info@appsec-labs.com | https://appsec-labs.com | 10 HaTa'as St., Kfar Saba 44641 Israel | T: +972-9-7485005 | F: +972-9-7730595

Mobile Application Security Report 2015

Executive Summary

The mobile application industry is growing exponentially at an explosive pace, yet security issues of mobile applications are lagging behind.

As hacking incidents become more public and have a greater impact on businesses than ever before, it is clear that the evolution of the field of mobile application hacking is rapid both in numbers and in techniques.

In this report we intend to share our experience as an Expert Application Security Company.

AppSec Labs is **cutting edge** in mobile application security and has produced a platform + tools for testing Android applications (AppUse) and a tool for testing iOS applications (Inalyzer) which have been downloaded by tens of thousands of security personnel worldwide.

We have recently released **APPUSE PRO** our Android Application Testing Platform + full tool box for android application security testing

Read more about AppUse Pro - https://appsec-labs.com/appuse/

Likewise, we have been training security professionals globally on mobile application security at BlackHat USA (6 trainings in the last 2 years and 2 trainings scheduled for BlackHat 2015.

We have aggregated and analyzed the findings of all our 2014-15 mobile application penetration tests in order to gain insight into the current state of mobile application security.

In this document we will address the most common vulnerabilities we have found throughout testing cycles, we will address the severity and business risks which these vulnerabilities can cause, and yes, we will address the famous Android VS iOS rivalry, hint: we will shatter the myth of iOS apps being inherently more secure than Android apps.

Our intent in this report is to elevate mobile application developer's awareness to the importance of addressing security issues as part of the development life cycle.

We hope all readers of this report will gain insight and **act** to improve overall app security at their respective businesses.

James Greenberg, VP AppSec Labs

Introduction – Why are mobile applications different?

The last year was a direct continuum to the last decade's trend: **everything is going mobile**. More and more people are getting used to having an all-in-one device, their mobile.

No one wants to carry around their camera, their laptop, their wallet, and even their TV or radio; not when they can have everything in one place, all the time. Therefore, there is no surprise in the exponential rise of mobile applications; if you want to reach your customers, all the time, your business must have a mobile app.

The meteoric growth of mobile applications, brings a lot of possibilities and comfort to their users, but alongside should come the responsibility.

As a security company specializing in mobile application security, we have had the opportunity to test the security of many mobile applications and **the results were alarming**.

On average, every application has ~3.5 **major security issues**, the level of mobile security today is equivalent to where we stood, ten years ago, in the midst of the world wide web, even though we had the same concerns before, when we first replaced physically going to our bank or healthcare offices, with connecting to their websites over the internet.

Naturally, when new technology platforms are introduced to the world, a window of opportunities opens for the hackers, who take advantage of the "window of opportunity" they have until the awareness for security and the knowledge of how to defend is established.

For example, many applications rely on the assumption that the average user cannot access the device's file-system, but only the presented interface; such assumption is playing into the hands of the hackers, of course.

Gaining access to the device's file-system is only a matter of few seconds if the device is rooted/ jail broken and a little longer, if not.

Once the hacker is there – all your personal information is disclosed – your passwords, your private photos, your personal accounts, everything is unprotected based on the false assumption that it is not accessible, because it is not given to the user by default.

The exact same assumptions mostly lead into another security breach, bypassing your authentication or authorization on the application, in order to perform actions or view information that you were allowed to do only if you had your personal password, or pin code.

In order to explain this, let's assume that we have an application on our mobile device that should protect our private photos and documents; in order for anyone to gain access to this "vault", they require to provide a 6 digit pin code, previously selected by us.

Is it really protected now? That depends on various factors, yet, according to what we've seen so far, **probably not.** Most of the applications would simply take your pin code, write it in plaintext on a configuration file, and keep it there unprotected for the hacker to see. Even if the password is encrypted, a great deal of applications would forget this password even exists if you completely erase it from the configuration file; otherwise, how would the applications know you even had passwords? – Usually, they won't.

Why is mobile application security different than web and desktop applications?

1. Access

In addition to the common threats for standard computer stations, such as desktops, servers, etc.; mobile devices deal with a wider risk, being exposed to even more security threats.

The main claim for that, is the fact that mobile devices, by nature, are mobile, thus tend to get stolen or lost.

An attacker might need physical access to the device of no longer then few minutes in order to extract data or perform actions on behalf of its original owner.

If you accidently installed a malware, an application with malicious intents, your data is at immediate risk, even when it is lying by your side; stealing personal information from the device, sending SMS on your behalf, getting your photos and posting on your accounts in your name are only a small risk of what these apps can do.

There is no way around it, in order to keep you and your data protected - security decisions, like encryption/ decryption, authentication, authorization and even part of the applications flow itself - should be redesigned, and verified so hacker don't have access to the server.

According to what we've witnessed to in the past year, we are far from being there.

Fact - Most of the applications don't take security measures to protect you and your data and prefer to trust the user.

2. Volume

Another important figure to address in this discussion relates to the number of mobile applications which are in use and the number of downloads.

The more applications are being released into the market and the more these apps are being downloaded by users – the higher and more real the risks become, in other words the attack surface of a hacker who wants to attack mobile applications is greater:

a) The number of applications to choose from.

b) The number of users who download mobile apps, hence the potential amount of information which can be exposed is greater.

To give you an idea of the volumes we are talking about, let's take the 2 leading mobile platform, iOS and Android.

iOS apps were first released in 2008 – in July 2008 iOS users had a mere 500 applications to choose from.

After less than 7 years statistics show that the iOS platform (including tablets) offer users a staggering 1.4 million different applications.

Apple reported in June 2014 that downloads have crossed the 75 billion mark!!!

Android applications which got off to a later start yet have caught up in the race, present similar figures both in number of apps and in download figures.

The bottom line is that the explosion of the mobile application industry in the last 7 years has created a whole new battle field in the rat race between hackers and security experts.

But the most important players in the game – the developers, well, they are way behind……

In depth analysis

When coming to analyze the data of thousands of hours spent in penetration testing, it is important to define our targets.

Which questions do we wish to get answers for and what is our message to you, the reader.

In order to start the discussion we will present a list of questions which this report will address.

A) Are mobile applications secure?

B) How severe are the security issues in mobile applications?

C) What are the main issues mobile app developers should be aware?

D) Are iOS applications more secure than Android applications?

E) How can the development community take action in order to improve application core security?

In the following chapters we will address all the above issues and present a current status of mobile application security.

Classifying Security Vulnerabilities – by severity and type

Before we begin to dive into statistics it is important to clarify how are security vulnerabilities classified?

There are 2 fundamental parameters to discuss when addressing security vulnerabilities:

- **Severity** of risk exposed
- **Type** of risk exposed

Vulnerabilities were not "created equal", some vulnerabilities pose greater threat to the organization than others, some are easier to exploit than others, the level of exposure caused by each vulnerability varies between applications and sometimes even within the same application we may find 2 vulnerabilities belonging to the same type yet one will be classified as Low severity and the other as Critical.

The "application ecosystem" plays a major role in assessing the level of risk of a vulnerability.

Classifying Severity of exposed vulnerabilities

In order to achieve a unified categorization of the severity of an exposed vulnerability we use the OWASP and WASC methodologies:

Critical

- A security breach that exposes a major security risk with a direct exploit (not needing user involvement). If exploited, the security threat might cause major damage to the application and/or have major impact on the company. The likelihood of such attack to occur is high, considering the architecture/business-logic/complexity of the exploit.

High

- The weakness identified has the potential to **directly** compromise the confidentiality, integrity and/or availability of the system or data, but the likelihood to occur is not high, considering the architecture/business-logic/complexity of the exploit. The possible damage to the application or the company is high, but not a total disaster.

- In applications involving sensitive data, the risk might be considered high in case the weakness by itself is against common regulations (e.g. PCI).

Medium

- A medium security issue that imposes some affect/damage to the application. Often it cannot be used directly, but can assist an attacker to launch further attacks.

Low

- No direct threat exists. It is a risk much more rather than a threat and does not cause damage by itself. The vulnerability may be leveraged with other vulnerabilities in order to launch further attacks.

- The risk reveals technical information which might assist an attacker in launching future attacks.

The presented above severity types serve as indicators to the application owner regarding how acute the issues are? And in accordance decide what measures (if any at all) should be taken in order to mitigate the issues reported.

Vulnerability types – "The 7 sins"

From the data collected, we have divided the vulnerabilities into seven applications "pitfalls":

1. **Authentication/ Authorization –** all security issues that result in performing actions or accessing data without sufficient permissions (e.g. bypassing security pin code).
2. **Availability** - all issues to result in denying from the application, or part of it – to work on the way it was intended to (e.g. crashing the application).
3. **Configuration Management** - issues related to incorrect or inappropriate configurations.
4. **Cryptography Weaknesses** – Breaches related to insecure way of data protection based on cryptography.
5. **Information Disclosure** – any unwanted technical information exposed to the client (e.g. application logs).
6. **Input validation handling** – issues occur due to mishandling data received from the user.
7. **Personal/ Sensitive information leakage** – Any exposure of our personal data or other sensitive data to the client (secret documents, credit card numbers, etc.)

Mobile application security stature

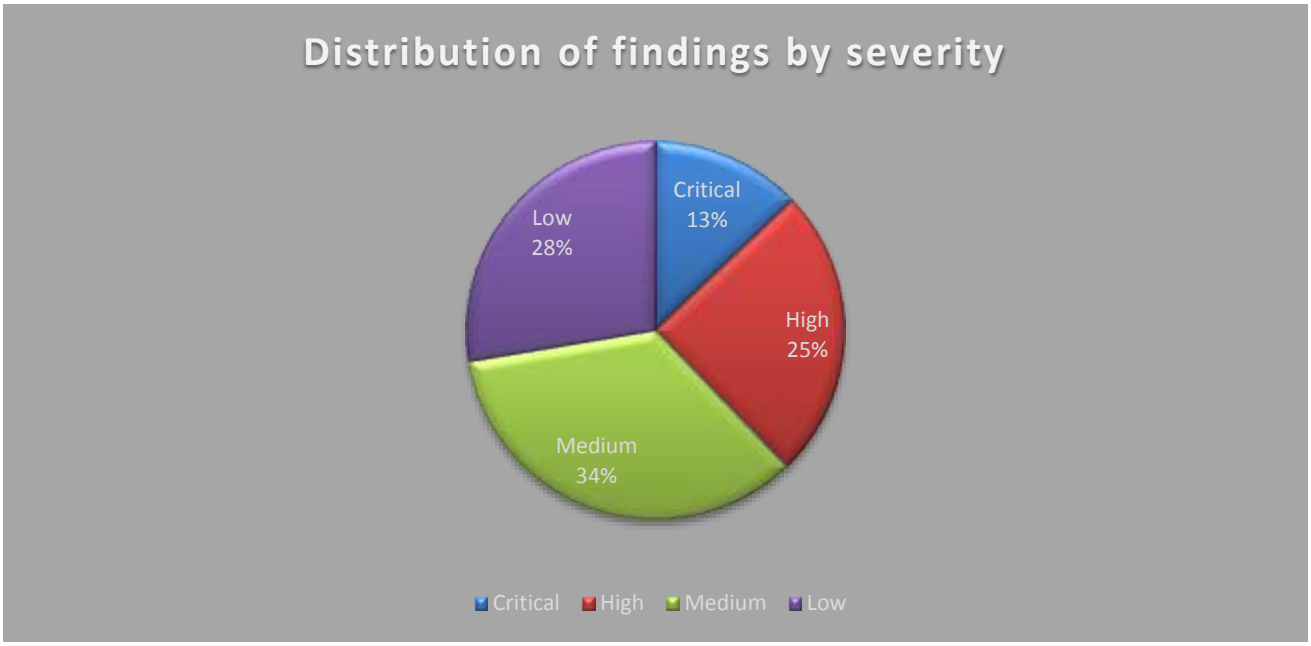
In 2014-15, we have tested hundreds of mobile applications, of all types – banking, utilities, retail, gaming and even security oriented applications. It is safe to say, we are unprotected and our privacy is exposed to many threats.

Before we dive into the details and classifications of the specific vulnerability types prevalent in mobile apps – let get a little perspective of what the general security status of mobile apps is.

As you can imagine the results are not very comforting, we have found on average **9.041 vulnerabilities per application tested.**

Now, this number represents findings classified low, medium, high and critical.

In the following pie you may see the distribution of findings by severity:



Analysis:

38% of vulnerabilities exposed are of critical or high severity, this indicates a relatively high risk for the average mobile applications.

By the statistic stated earlier, with an average exposure of 9.041 vulnerabilities per app, **the average number of high and critical vulnerabilities is 3.435 per app**

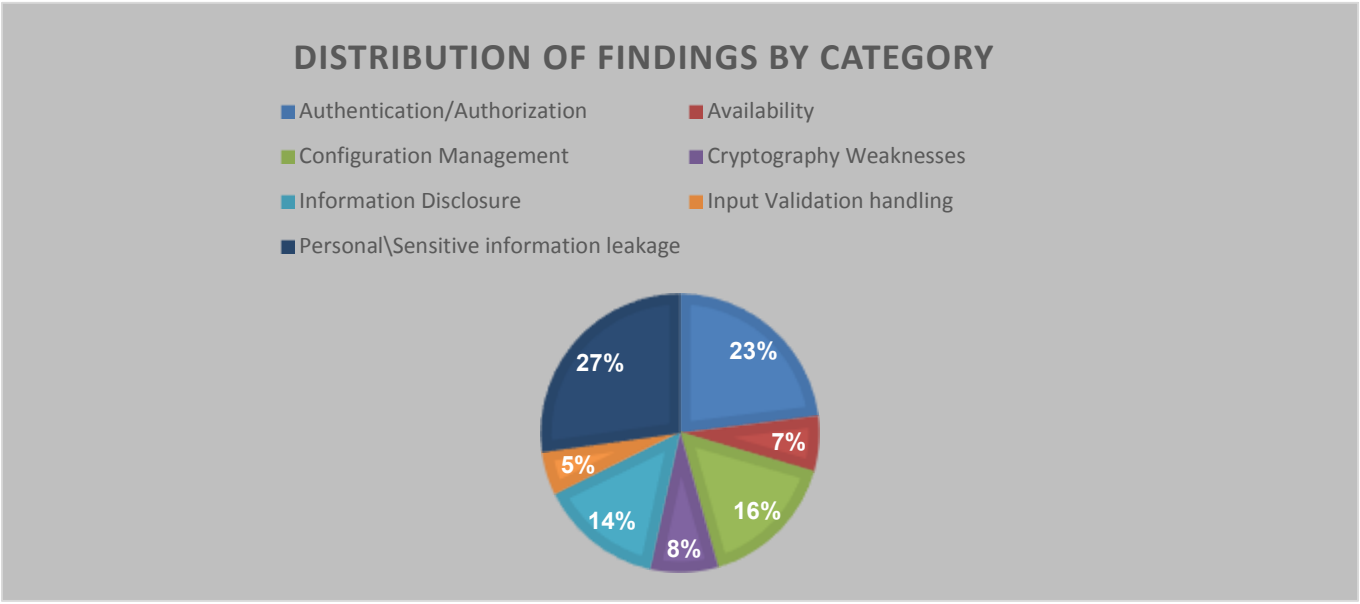
This means that the average mobile application is, pretty much, wide open for being the cause of major risk to its owners and users business.

So, this is a very alarming piece of information, but we wish for this report to be of value for the mobile app developing community and its leaders.

In order to have a deeper understanding of what must be done in order to create change by developers it is vital to map out what are the areas in which organizations ought to invest in improving security.

In order to map out and define the most acute areas it is important to see the distribution of vulnerabilities by types (as defined in the "7 sins" paragraph).

The table below presents the distribution of vulnerabilities by category as follows:



Analysis:

The most interesting statistic in the table above shows that 2 vulnerability "families", namely, *Personal / Sensitive information leakage* and *Authentication and Authorization* consist 50% of all vulnerabilities exposed.

Having said that, the statistic presented above does not correlate the **type** of vulnerability with the **severity** level of the exposed vulnerability.

We will later see that the dimension of severity is a major component.

For example : As *Personal / Sensitive information leakage* vulnerabilities are primarily classified as medium to low risk in comparison to *Authentication and Authorization* which are by majority high or critical risk, we can conclude that the first priority should be in addressing *Authentication and Authorization* issues.

As mentioned above, not all instances of a vulnerability classified into a certain category share the same severity level.

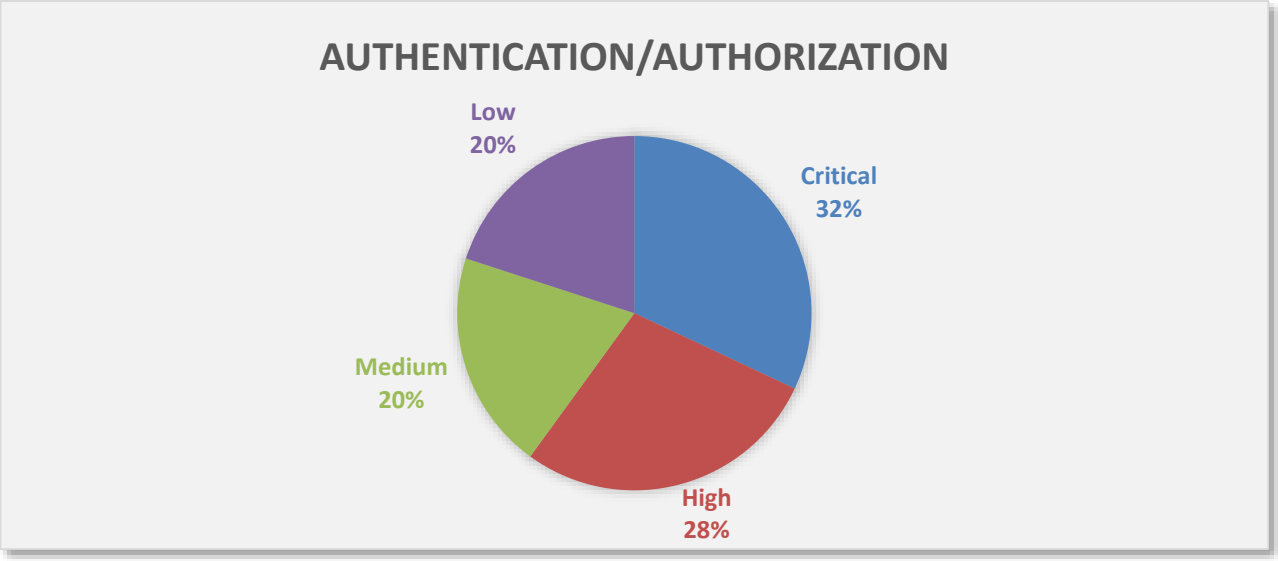
The context of the exposure plays a major role in the definition of the severity.

For instance we may expose in the same application 2 vulnerabilities belonging to the *Authentication and Authorization* category, yet one will be defined as critical and the other as low risk.

In order to get a clear breakdown per vulnerability type, we took each "family" of vulnerabilities and checked the severity classification of all its instances reported.

This analysis enables us to determine which vulnerabilities are more likely to be of high or critical impact and therefore necessitate extra attention.

1. **Authentication/ Authorization –** all security issues that result in performing actions or accessing data without sufficient permissions (e.g. bypassing security pin code).



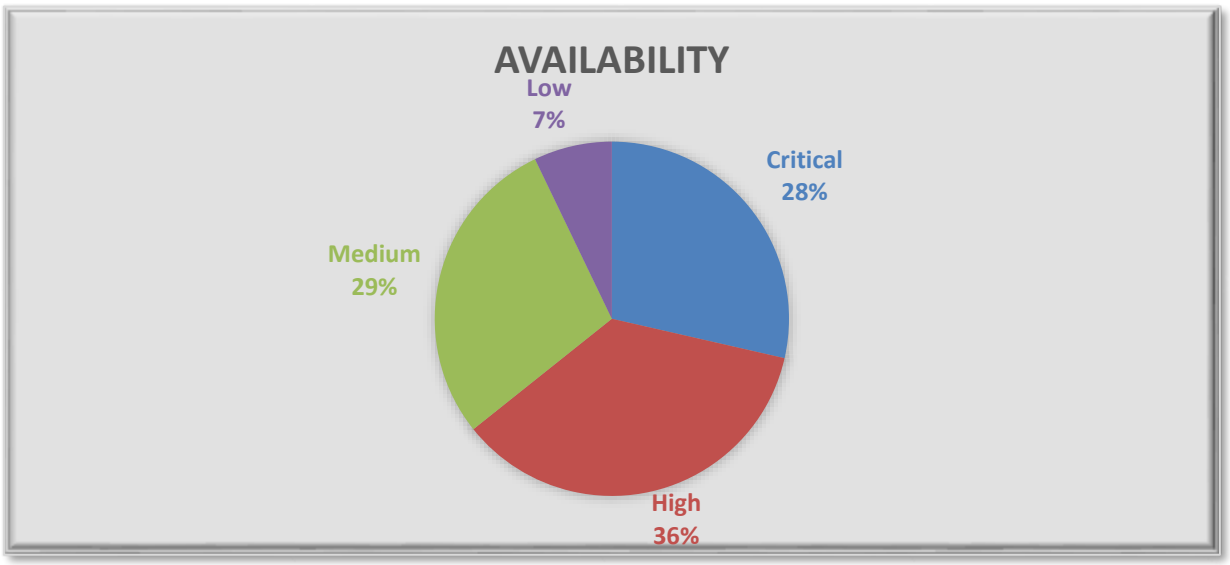
Analysis:

60% of Authentication and Authorization issues are of critical or high severity clearly indicating that development teams:

A) Should take measures to improve developer awareness to securing Authentication and Authorization mechanisms.

B) Prioritize securing Authentication and Authorization mechanism.

2. **Availability** - all issues to result in denying from the application, or part of it – to work on the way it was intended to (e.g. crashing the application).



Analysis

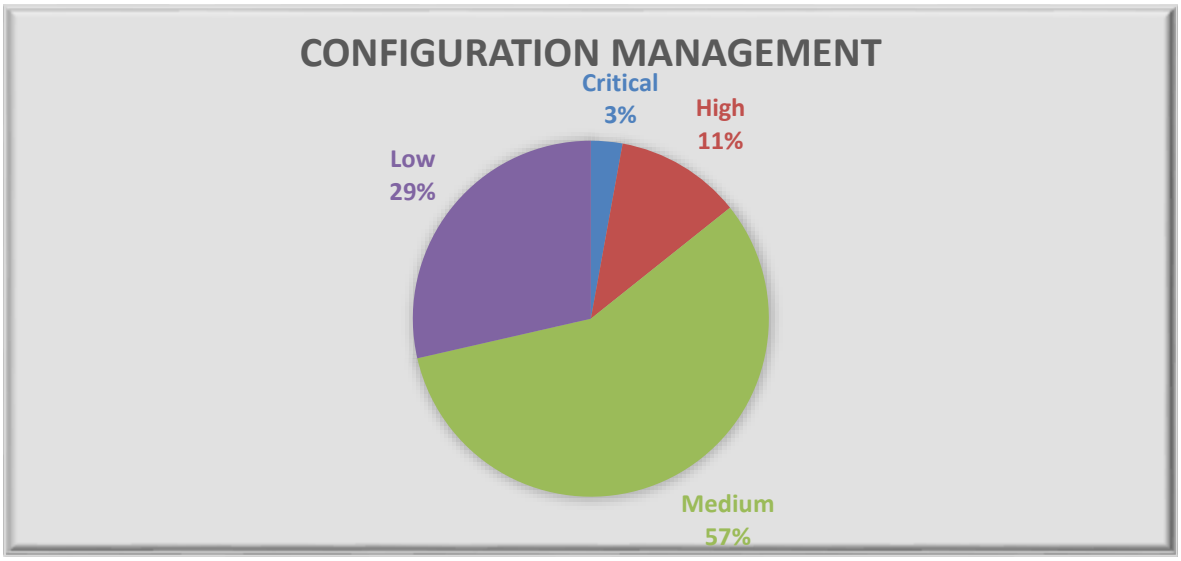
As we see from the stats presented above Availability issues as well, are at "high risk" category with 64% of all findings in this category posing major threat to application owners and users.

Only 7% of Availability issues reported are classified as low severity issues which indicates that availability issues are a high impact issues.

In order to mitigate Availability issues it is important for developers to understand the vectors of attack which enable hackers to cause availability issues, for example, understanding how to defend requests from crashing the system by causing an overflow of requests which will challenge the processing abilities of the application.

By understanding which functionalities in your application are potential victims of this risk and limiting the amount of permitted requests (per second), hereby mitigating or at least minimizing the risk of causing availability issues.

3. **Configuration Management** - issues related to incorrect or inappropriate configurations.

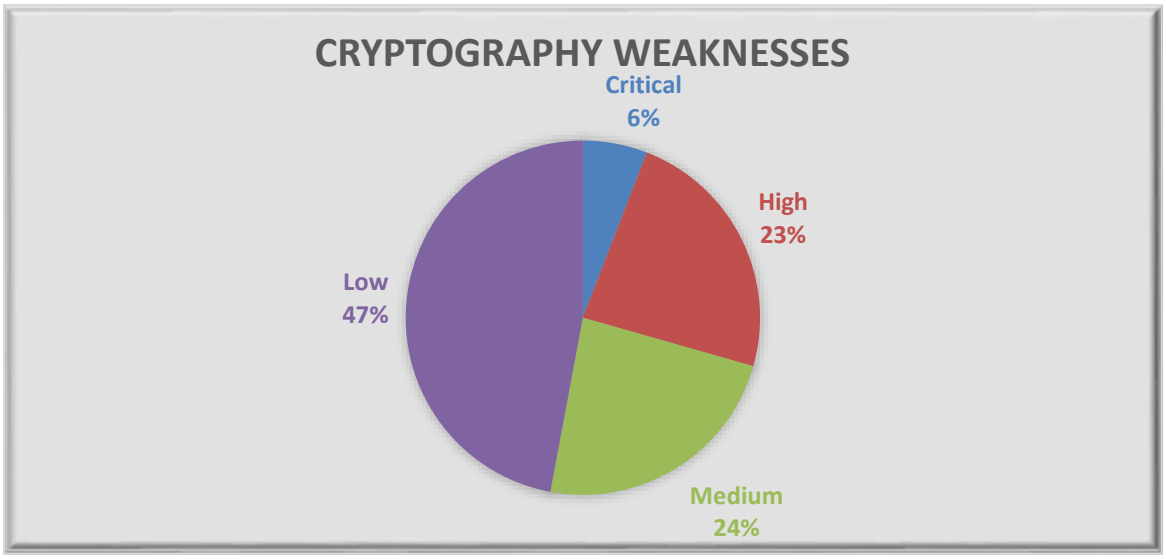


Analysis:

As can be seen in the table above, this category of vulnerabilities is of lower criticality, though we cannot overlook the fact that 14% of the reported vulnerabilities in this category are of critical and high severity,

the majority of reported incidents are medium (57%) ,positioning this family of vulnerabilities as less critical in comparison to the others.

4. **Cryptography Weaknesses** – Breaches related to insecure way of data protection based on cryptography.



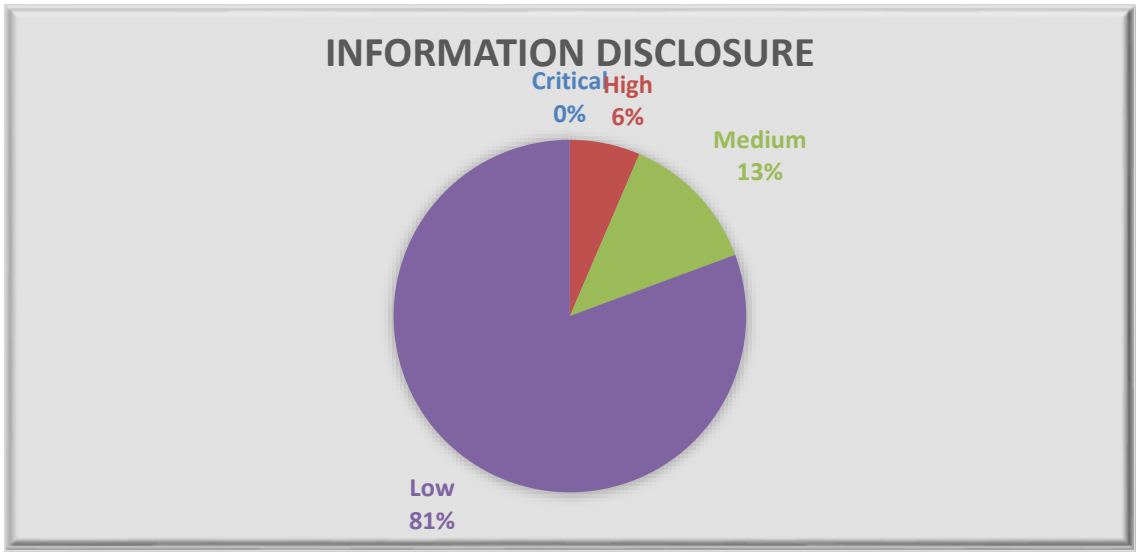
Analysis:

In the cryptography weaknesses category we find a very interesting statistic.

29% of the vulnerabilities reported are of high and critical severity, yet almost 50% of the reported cases were of low severity.

Which indicates that this category is tricky and does necessitate close attention since there is a probability that overlooking cryptography weakness can result in causing major risk.

5. **Information Disclosure** – any unwanted technical information exposed to the client (e.g. application logs).



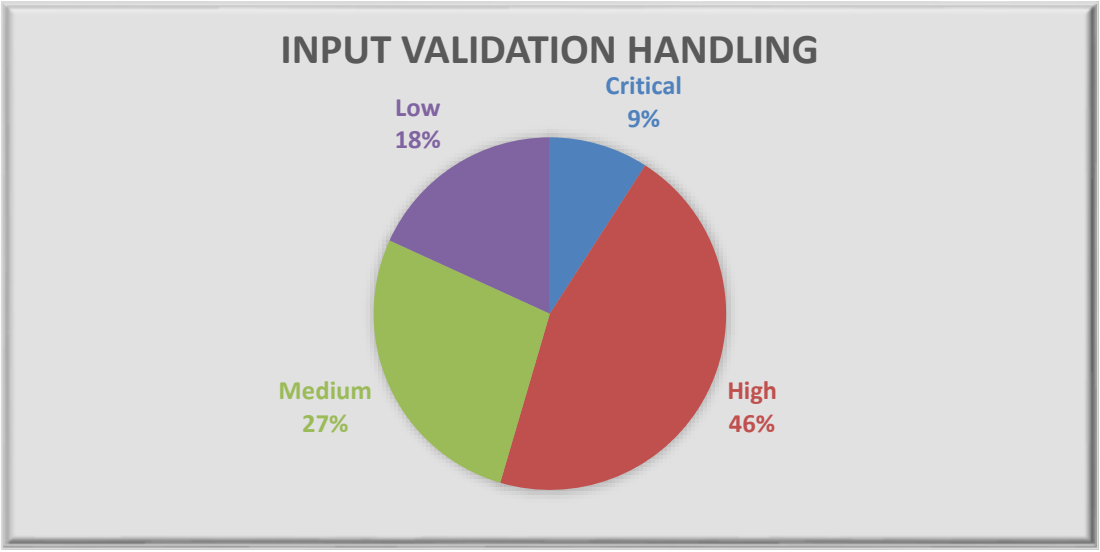
Analysis:

Information disclosure vulnerabilities are an interesting category, though the statistics show that the vast majority of the cases reported are of low severity and only 6% are classified as high severity.

This vulnerability category should not be overlooked.

Though, Information disclosure usually cannot cause direct damage, yet it can serve as a source of information which can be used as a precursor for more severe attack vectors.

6. **Input validation handling** – issues occur due to mishandling data received from the user



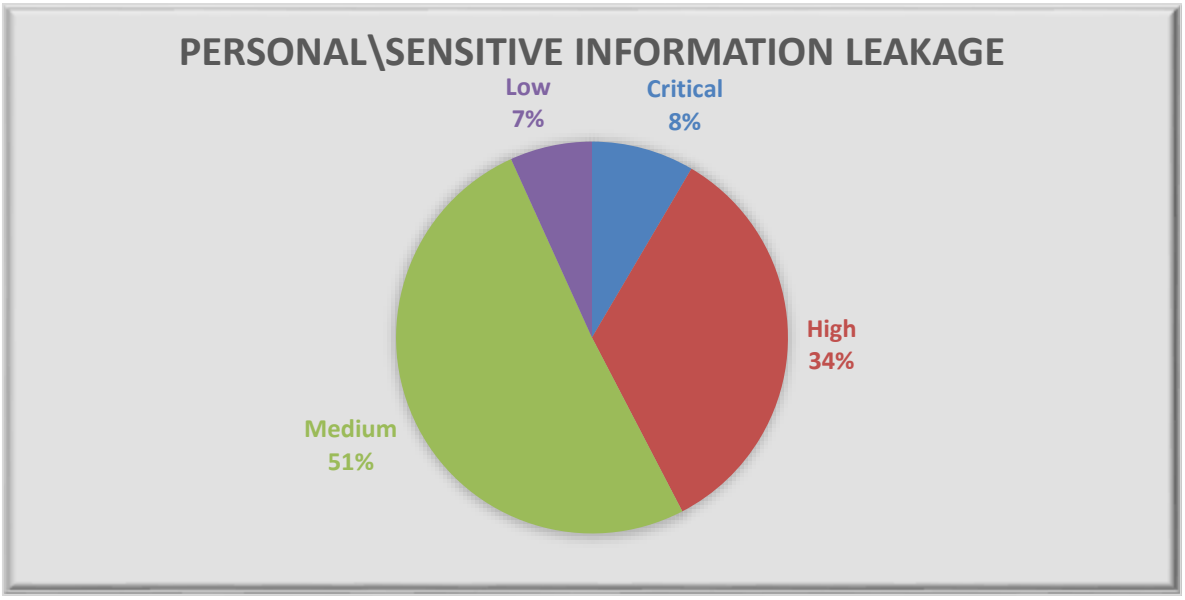
Analysis:

Input validation handling is a major topic which organizations should have a clear directive regarding secure implementation of secure input validation methods.

As we can see, 55% of the reported exposed vulnerabilities are of high and critical severity positioning input validation issues as overall high risk.

With only 18% of vulnerabilities exposed in this category classified as low severity – this category of issues demands close attention.

7. **Personal/ Sensitive information leakage** – Any exposure of our personal data or other sensitive data to the client (secret documents, credit card numbers, etc.)



Analysis:

Personal and sensitive information leakage is a very interesting category, consisting 27% of all exposed vulnerabilities –more reported incidents than any other category.

This vulnerability category is not only the most commonly found vulnerability, it is relatively of high risk with a total of 42% of reported incidents being of major risk to the application owner and users.

Only 7% of reported incidents in this category were reported as being of low risk.

The implications of these statistics are that securing personal and sensitive information should be a top priority of system designers and developers since these are the both common and of high risk.

It may sound obvious that personal and sensitive data must be secured, yet de facto our experience shows that lack of awareness to security issues are the direct cause of this serious attack vector.

Summary - Statistics aggregated

To summarize the findings above please view below chart depicting the distribution of findings by category and severity.

	Critical	High	Medium	Low
Authentication/Authorization	32 %	28%	20%	20%
Availability	28 %	36%	29%	7%
Configuration Management	3%	11%	57%	29%
Cryptography Weaknesses	6%	23%	24%	47%
Information Disclosure	0 %	6%	13%	81%
Input Validation handling	9%	46%	27%	18%
Personal\Sensitive information leakage	8%	34%	51%	7%

Mitigation- What can developers do to improve app security?

So, we have mapped out all the categories and have a good idea of what areas of our applications need extra attention (Authentication and Authorization, Availability and input validation handling).

Yet what are the best ways to address each category of vulnerabilities?

In the next paragraph we will present high level mitigation directives for the different vulnerability types.

Availability

1. Perform Input validation on all received intents and ignore badly formatted intents.
2. Catch all Exceptions, in order to block a DoS attack using system exceptions.

Authentication/Authorization

1. Never trust the client. Ensure the user who requests **any page/action** has the legitimate permissions by validating the session permission in the server side.

2. Allow the system users 3-5 failed login attempts. If the user fails more times than the allowed amount, deploy an active CAPTCHA mechanism

Cryptography Weaknesses

1. Due to the sensitivity of information (example – user and pin code) the server must require the transport layer to be over SSL/TLS.
2. It is recommended to use AES128/256 instead of RC4

Information Disclosure

1. Use extreme obfuscation in order to prevent an attacker from retrieving useful data from the APK file.

Personal\Sensitive information leakage

1. Do not store sensitive information on device

Configuration Management

Since configuration issues vary from application to application it is important to implement a control mechanism which will assure adequate configuration management.

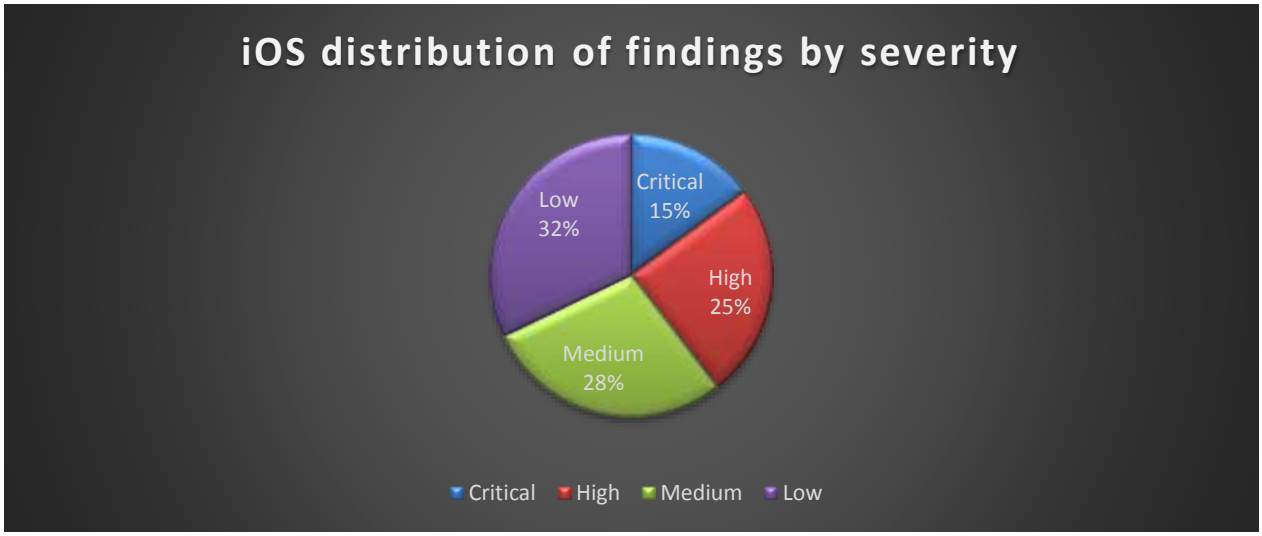
Shattering myths - Android VS. iOS Application Security

So, it is a common myth that the iOS development platform is more secure than the Android equivalent for several legitimate reasons:

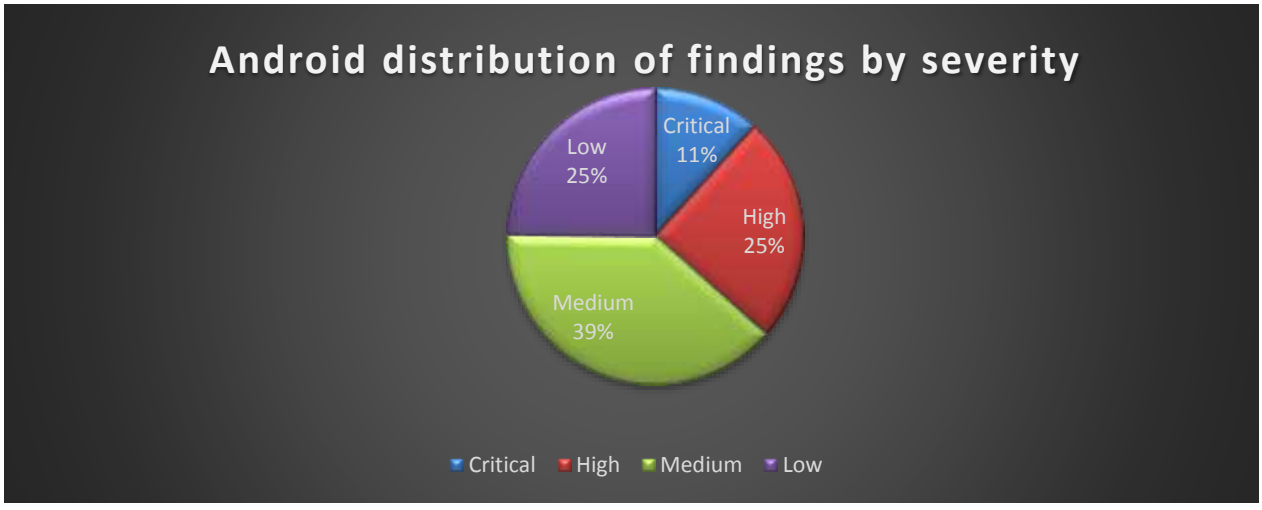
a) iOS has more restrictive controls over what developers can do and tight application sandboxing
b) iOS Applications are fully vetted before being released to customers - preventing malware from entering the Apple App Store

Yet, in the field of pure application security where vulnerabilities are built in the code or into the application logic the story is quite different.
Our statistics show that the distribution of vulnerability exposed by severity are almost identical between iOS and Android Applications with a slightly higher percentage of critical vulnerabilities in iOS applications.

Let's look at the stats…..



As we can see -40% of the detected vulnerabilities on iOS tested application were critical or high severity in comparison to Android applications in which the percentage of high and critical vulnerabilities add up to 36% of all findings as depicted below.



The statistics clearly show that iOS applications are no safer on the application level than their Android equivalents.

It is important to state that many of the penetration testing projects we performed were for the same applications Android and iOS versions, providing us with a better comparison between the platforms.

The statistics show that there are marginal differences between the 2 leading and rivaling platforms for developing mobile applications.

The significance of this information is vital for iOS development team leaders.

The false sense of security that iOS developers possess – has no foundation when discussing security flaws embedded in the code itself.

Final message – take your destiny into your own hands – secure the core of your business

The final and vital message of this report is very simple.

- There is clearly a lacking in awareness to application security and implementation of secure coding best practices by mobile app developers
- The risk is real! – the levels of risk which were detected – indicate **real risk** to application integrity of **almost all mobile applications**
- We will be experiencing more major hacks being performed via the mobile application vector than before.
- Organizations must not rely on external defense mechanisms only - code level security is a serious player.
- It is highly recommended to address mobile software security by**:**
    - **Integrating secure coding best practices into the development life cycle.**
    - **Educating developers – Knowledge is a great tool, empowering developers to protect their own apps.**
    - **Get your app tested by professionals before it hits the market and is exposed to hungry hackers.**

Bottom line:

The developer community can and must protect its products better by enhancing knowledge of application security issues and implementing Secure Development Lifecycle directives – it will have a real impact on your product security by minimizing risks.

Just because your gate is locked doesn't mean you don't need to lock the door!

Don’t rely on external security mechanisms when you can develop your app to have internal resilience at the core.

We hope you found the information presented in this report interesting and eye opening.
For questions, inquiries, suggestions or if you are interested in our services please contact:

James Greenberg
James@appsec-labs.com