Date - 2nd March 2019

www.ncdrc.res.in

NCDRC MAST v2.0

National Cyber Defence Research Center (NCDRC)
Mobile Application Security Testing Framework v2.0

Event: National Cyber Defence Conference 19 | Venur: CUSAT, Kochi Campus, Kerala

An Initiative by Briskinfosec BINT LAB

TABLE OF CONTENTS

ABOUT NCDRC

NCDRC has been started with a great vision to safeguard the cyber world from current threats in cyber space. The multi-dimensional structure of technology in the cyber space poses a great challenge in handling the multifaceted problems in cyber domain.

Cyber world is fronting huge threats from various countries in cyber-attacks and information thefts. Cyber security challenges put sensitive data at risk and can cost your company time, revenue and resources. Cyber safety has taken an initiative to curb and enervate the notoriously spreading cyber threats from various directions and dimensions.

Today, Cyber Security is a daunting Security Problem and we're applying decades of expertise to the task. To keep systems safe and to foil attacks, National Cyber Safety and Security Standards develops protective technologies, conducts threat assessments, and analyses Government, Military, and Civilian computer networks.

National Cyber Safety and Security Standards have done an extensive research in the Cyber domain to understand the nature of cyber threats and cybercrimes. We have understood that the multi-faceted cyber technology cannot be handled by common standards and security policies. Thus, a Common Platform to facilitate the experts to provide an effective solution for the complex and alarming problems in the society towards cyber security domain. National Cyber Safety and Security Standards is developing innovative strategies and compliance procedures to curb the increasing complexity of the Global Cyber Threats.

NCDRC MAST V2.0 (MOBILE APPLICATION SECURITY TESTING) FRAMEWORK

NCRDC MAST v2.0 (Mobile Application Security Test) FRAMEWORK was proposed by NCDRC research lab. NCDRC partnered with Briskinfosec, a leading cybersecurity company based from India, made a research on NCDRC MAST v2.0 (Mobile Application Security Test) FRAMEWORK at their BINT LAB (Brisk Intelligence Lab). As official partners of NCDRC MAST V2.0 (Mobile Application Security Test) framework, Briskinfosec is constantly supporting and improving NCDRC MAST v2.0 (Mobile Application security) framework.

Briskinfosec is a global frontrunner in end-to-end information security services based on Chennai, India. We are an expert team of highly dedicated security specialists and researchers, supported by strategic and emerging technology partners, focused on information security for our enterprise customers. Our multi-disciplinary approach with deep, practical industry knowledge helps clients to meet various challenges and respond to opportunities. We deliver according to the highest standards of our knowledge and our experience is the best in class on global perspective

WHY MAST V2.0 ?

Making a guide like this is a massive responsibility, demanding hundreds of people around the world. MAST v2.0 (Mobile Application Security Test) Framework provides mobile security folks, the ability to work together and form a leading approach towards mobile security issues. The importance of this guide benefits anyone with the ability to understand the techniques, used for testing common mobile security issues.

WHAT MAKES MAST V2.0THROUGHT-PROVOKING?

Mobile applications have very different threat models than their web-based counterparts. Android, iPhone, Blackberry smartphone and tablet devices provide a variety of functions built into the hardware that makes them vividly different from desktop or laptop computers, presenting a unique set of security ramifications that must be dealt with at the application level.

It is important for mobile developers to understand how to design and build applications that securely influence a platform's capabilities without exposing the organization or the application's users to risk.

Developers of mobile applications need to understand:

- The competences of their chosen development platform(s)
- The threat model for the system they are building
- The mobile application itself is only a part of the system that attackers will attempt to compromise.

This guide must make its way into the hands of Mobile Application Penetration Testers. Keeping this information up to date is a critical aspect of the NCDRC MAST V2.0 project.

It is vital to adopt this guide in your organization. You may need to customize the information for matching your organizations technologies, processes, and organizational structure. Mobile Application Pentesters should use this guide in combination with other techniques as one way to verify that no security holes have been missed in a mobile application.

Executive summary

- Adopting mobile Application security without the correct policies and management infrastructure in place, increases the opportunities for attackers to breach sensitive data.
- The National Cyber Defence Research Centre (NCDRC) developed MAST v2.0 (Mobile Application Security Testing) Framework used for testing common mobile security issues.
- The security characteristics in this Framework are informed by guidance and best practices are from, as per the industry standards.
- The (NCDRC) approach uses non-commercially available products that can be included alongside your current products in your existing infrastructure.
- The example solutions are packaged as a "how to" guide that demonstrates implementation of standards-based, available cybersecurity technologies in the real world. The framework helps organizations utilize technologies to reduce the risk of intrusion via mobile application while saving them research and Proof Of Concept (POC) costs.

Challenge

Information technology (IT) environments have changed drastically because of the increasing popularity of smartphones, tablets, and other highly capable rapidly maturing mobile devices. These devices have many functional similarities to traditional IT systems — including access to a wide range of enterprise applications and data, as well as the additional functionality that is particular to mobile computing. This has greatly expanded the utility and value of mobile devices, enabling employees to do their jobs more effectively and efficiently. Unfortunately, security controls have not kept pace with the security risks that mobile application can pose. This gap in protection mechanisms means that data stored on or accessed from mobile application or device is at increased risk of being breached.

For example, suppose an organization has enabled mobile access to its email, calendaring, and contact management services regardless of the origin of the employee's mobile devices (organization-owned and employee-owned, organization-provisioned and employee-provisioned, etc.). If sensitive data is stored on a poorly secured mobile device that is lost or stolen, an attacker may be able to readily gain unauthorized access to that data. Even if the mobile application security is not minted, then it may lead to data breach and data theft in an Organisation.

SOLUTION

We demonstrate how security can be supported throughout the mobile testing life cycle. This framework includes how to perform a mobile application testing in an effective way.

This framework…

- Identifies the security characteristics needed to sufficiently reduce the risks from mobile testing.
- Maps security characteristics to standards and best practices from OWASP and other organizations.
- Provides example solutions that are suitable for organizations of all sizes, and evaluates those solutions.

Your organization's information security experts can adopt MAST v2.0 framework that will best integrate with your existing tools and IT system infrastructure. Your organization can adopt this solution or one that adheres to this framework. Implement industry standard mobile security controls, reducing long-term costs and decreasing the risk of vendor lock-in.

Version History

VERSION 2.0 2nd March 2019

With version 2.0, we released a new NCDRC MAST v2.0 (Mobile Application Security Test) Framework that will be the standard guide to perform Mobile Application Security Test.

The primary aim of the NCDRC MAST v2.0 (Mobile Application Security Test) Framework Project is to standardize the range in the coverage that is accessible, when it comes to performing Mobile Application testing. The standard provides

- Static Analysis
- Dynamic Analysis
- Communication Level Analysis
- Business Logic Analysis
- Server-side Analysis

VERSION 1.0 2016

With version 1.0, we released a NCDRC MAST v2.0 (Mobile Application Security Test) Framework that will be the standard guide to perform Mobile Application Security test.

NCDRC thanks the Project Leaders, Contributors, reviewers, and editors for their hard work in bringing this guide.

SHARE YOUR FEEDBACK

If you adopt this solution for your own organization, please share your experience and advice with us. We recognize that technical solutions alone will not fully enable the benefits of our solution. So, we encourage organizations to share lessons learned and provide best practices for transforming the processes, associated with implementing this framework. To provide comments or to learn more by arranging a demonstration of this example implementation, contact the NCDRC or BINT Lab.

Email: c-defence@ncdrc.res.in and contact@briskinfosec.com

ACKNOWLEDGEMENT

Mr. KHALIRAJ NAIDU,
Additional Director General – National Cyber Safety and Security Standards (NCSSS)

PROJECT LEADERS

Mr. ARULSELVAR THOMAS,
Founder & Director, Briskinfosec Technology and Consulting Pvt Ltd,

Technical Head – National Cyber Defence Research Centre (NCDRC)

Mr. DINESH.C,
Security Consultant, Briskinfosec Technology and Consulting Pvt Ltd,

Member - National Cyber Defence Research Centre (NCDRC)

CONTRIBUTERS

Mr.VENU GOPAL
Advisor – Briskinfosec Technology and Consulting Pvt Ltd

Member – National Cyber Defence Research Centre (NCDRC)

Mr. SATHISH ASHWIN,
Architect – Cybersecurity Consulting & Advisory Services - Virtusa

Head – National Cyber Defence Research Centre (NCDRC)

Mr. .SATHISH KANNAN,
Senior Security Researcher - Briskinfosec Technology and Consulting Pvt Ltd

Chair member, National Cyber Defence Research Centre (NCDRC)

CALL TO ACTION

NCDRC and BRISKINFOSEC strongly encourages you to get familiar with the security testing guidance in this guide. It is a great road map for testing the most common issues fronting in mobile security today, but it is not comprehensive. If you find fault, please add a note to the discussion page and consider joining us as an individual or as a corporate member so that we can continue to produce materials like this Mobile Application Security testing guide and all the other great projects at NCDRC.

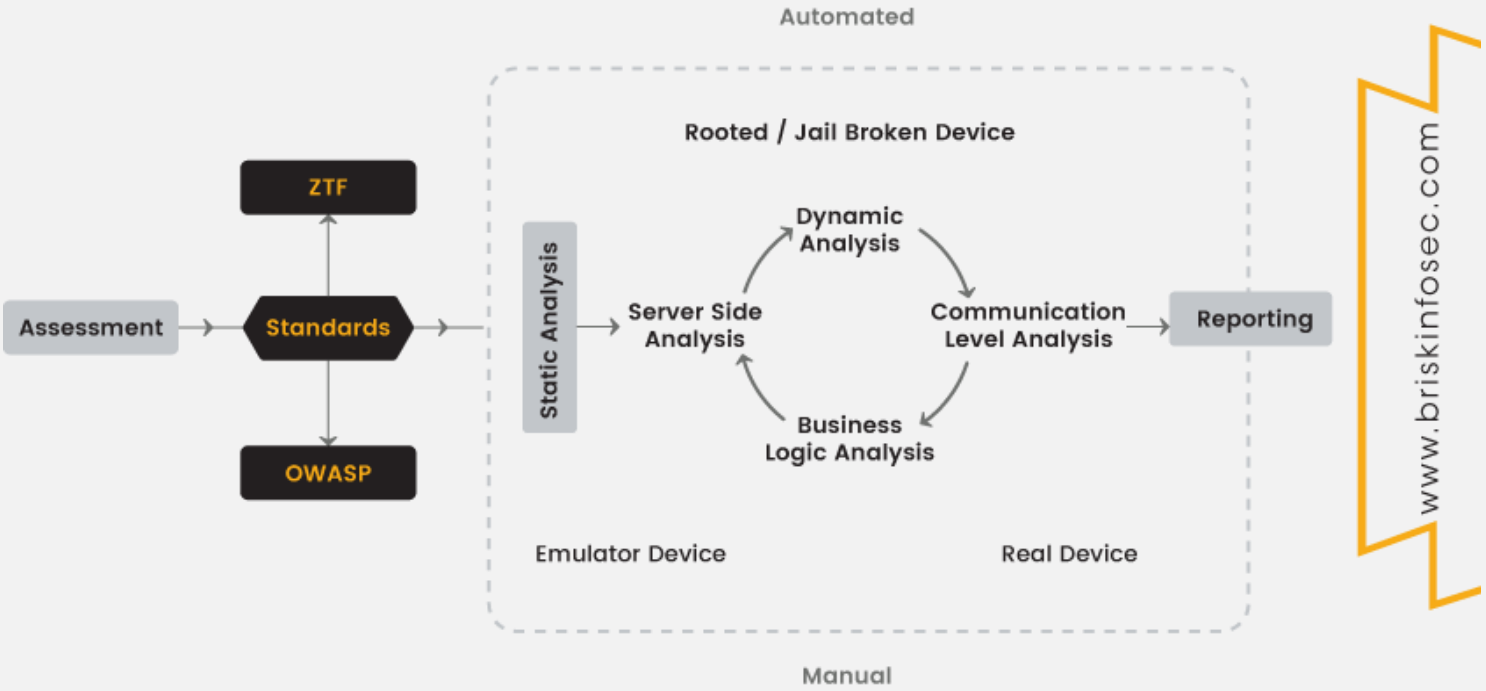
We appreciate and invite your contribution and support towards NCDRC MAST v2.0 (Mobile Application Security Test) Framework. Feel free to reach us over email MAST v2.0@ncdrc.co.in and bintlab@briskinfosec.com

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by MAST v2.0 or NCDRC, nor is it intended to imply that the entities, equipment, products, or materials are necessarily the best available for the purpose.

MOBILE APP SECURITY TEST METHODOLOGY (MAST V2.0)

NCDRC and BRISKINFOSEC have instrumented complex mobile app security testing into crystal clear basic approach. All mobile apps are revealing different security test depending on the way it was tested. Briskinfosec's BINT LAB researchers concluded four stages of mobile app security test which covers every aspect of security issues in target mobile application.



SECURITY TEST PHASES:

Our customized approach trails word class mobile security testing frameworks such as OWASP, OSAM, and SANS and our other testing phases such as:

- Static Analysis
- Dynamic Analysis
- Communication Level Analysis
- Business Logic Analysis
- Server-Side Analysis

SECURITY TEST STAGES:

- Emulator Level
- Real Device Level
- Rooted/Jail Broken Level

Our customized approach is to satisfy the end to end mobile app security requirements and meeting most of the information security compliance such as ISO 27001, PCI: DSS, HIPAA, etc.

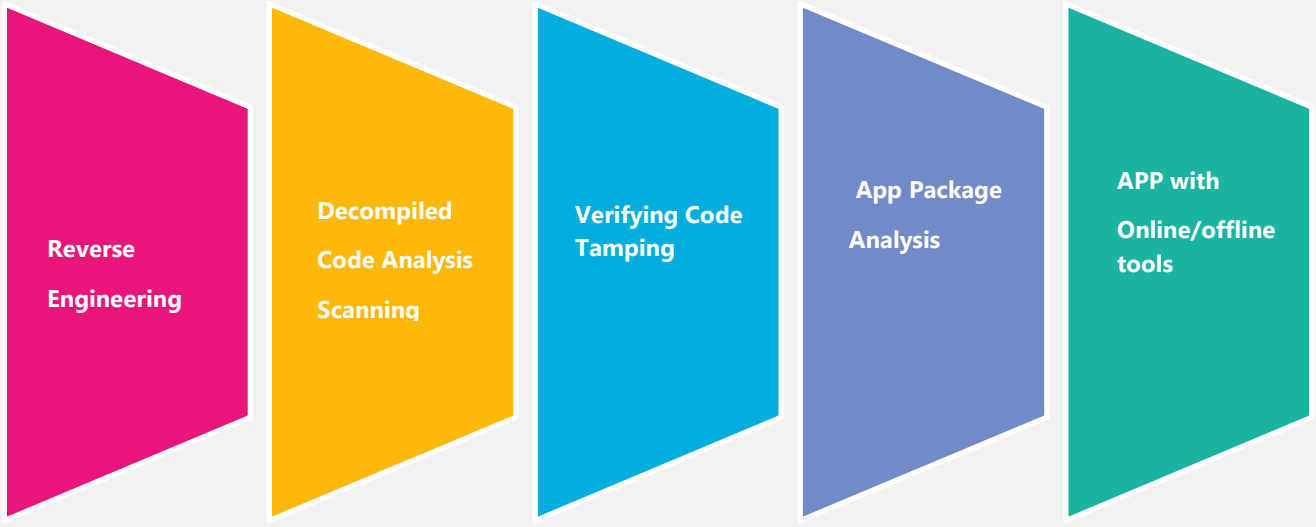
STATIC ANALYSIS:

App level security test is to execute the security test with independent mobile App package (APK, IPA) level security test.

PRE-REQUESTS:

1. App Package (APK/IPA)

Static analysis is the first phase. Primarily, we must analyse the application package by reverse engineering the APK/IPA file, which is further proceeded with code analysis and verifying code tamping. Lastly, perform complete scanning of the application in the cloud environment, along with offline tools.



After Static analysis phase, it goes to the security stages triangle. Each security stage includes four different phases which makes security testing more efficient. Through these various stages and phases, this approach is a crystal clear prospect which enables us to encompass attack vectors in mobile applications.

SECURITY TEST STAGES:

- Emulator Level
- Real Device Level
- Rooted/Jail Broken Level

SECURITY TEST PHASES:

- Dynamic Analysis

- Communication Level Analysis
- Business Logic Analysis
- Server-Side Analysis

REAL DEVICE LEVEL:

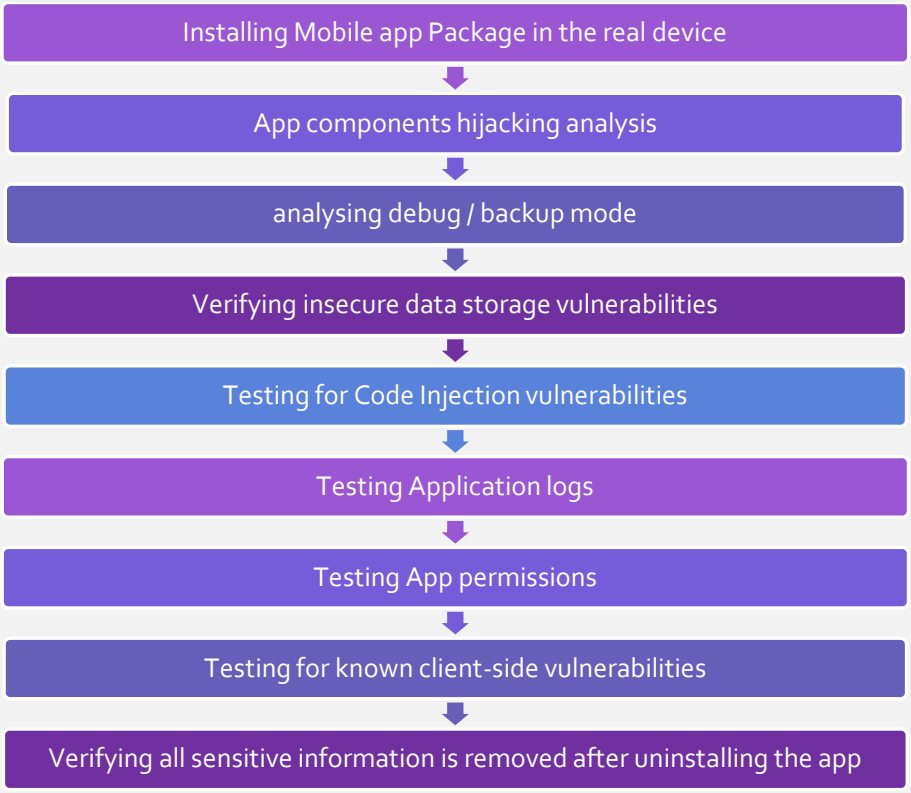
Given app will integrate with corresponding real device to do the complete security test.

PRE-REQUESTS:

1. App Package (APK/IPA)
2. Mobile devices for corresponding mobile platform
3. Login details if applicable

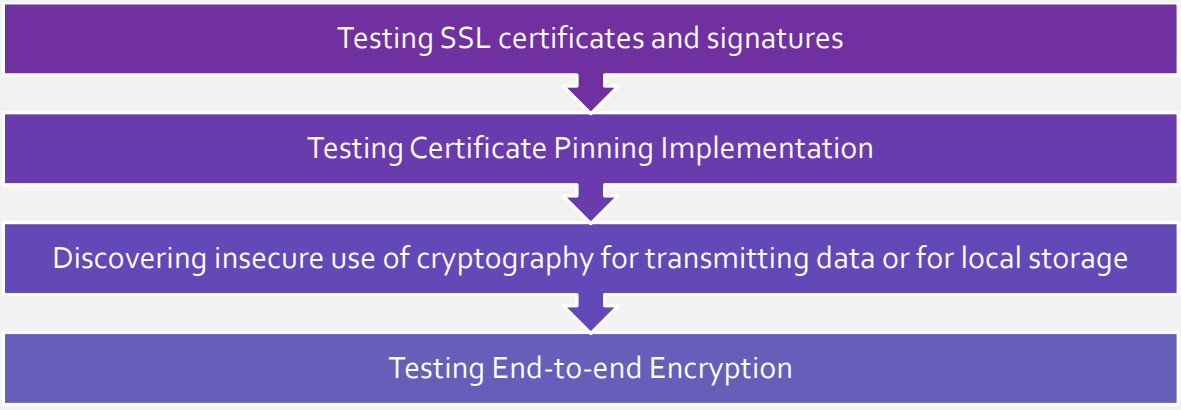
PHASE 1: DYNAMIC ANALYSIS

In the real device stage, the first phase is to install mobile app packages in the real device, post which we have to perform dynamic analysis. We need to verify the security of local file system and verify the Application's logs by simulating different test cases. We need to perform test for code injection vulnerabilities, application permission, known client-side vulnerabilities. Finally, we must verify all the sensitive information which is removed after uninstalling the application.



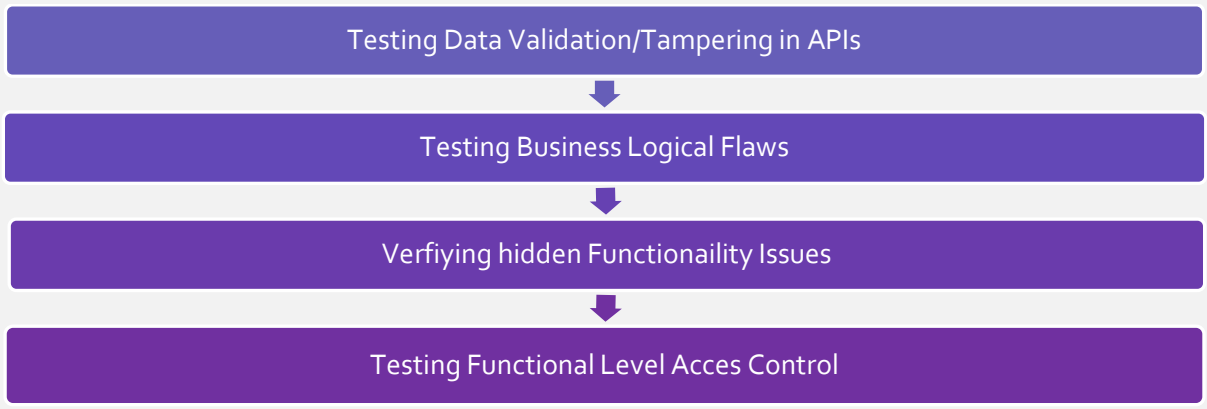
PHASE 2: COMMUNICATION LEVEL ANALYSIS

In real device security stage, the second phase is communication level security test. Primarily test SSL certificates and signatures. Next, we need to verify the certificate pinning implementation and insecure use of cryptography for transmitting data for the local storage. Following, we should test for the end to end encryption.



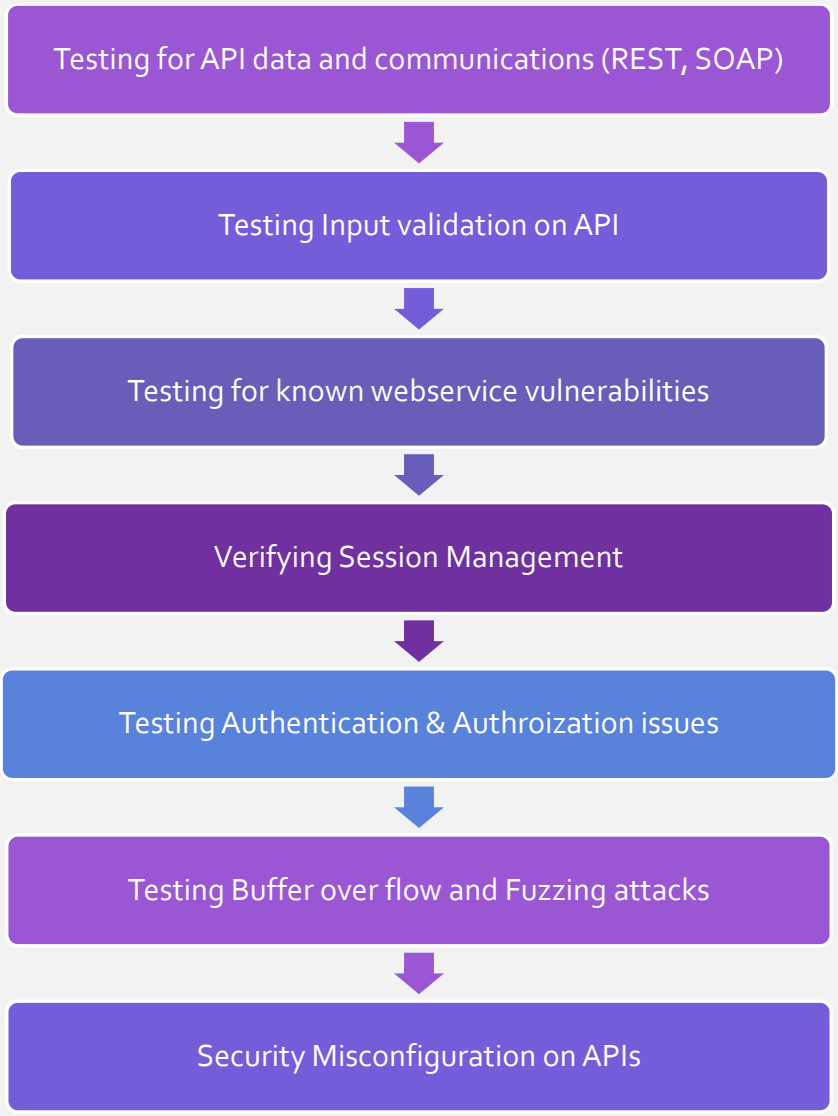
PHASE 3: BUSINESS LOGIC ANALYSIS

In real device security stage, the third phase is business logic testing. Primarily, we need to check for the logical flaws that are leading to business logic issues. Further, we need to check if there are any loop on data flow or functional flow. We need to perform data validation and tampering data attacks, verify if there are any hidden functions and test for known & unknown client-side vulnerabilities.



PHASE 4: SERVER-SIDE ANALYSIS

In real device security stage, the third phase is server-side security test. Primarily we need to test the API used for the data and communication (reset and soap). Further, we need to check the input validation, testing generic web application attacks through web services. We need to perform testing buffer over flow and fuzzing attacks, test for unknown vulnerabilities. Testing for Authentication & Authorization issues such BAC, IDOR, MFLAC, etc must also be done.



EMULATOR LEVEL:

Given app will integrate with corresponding emulator to do the complete security test.

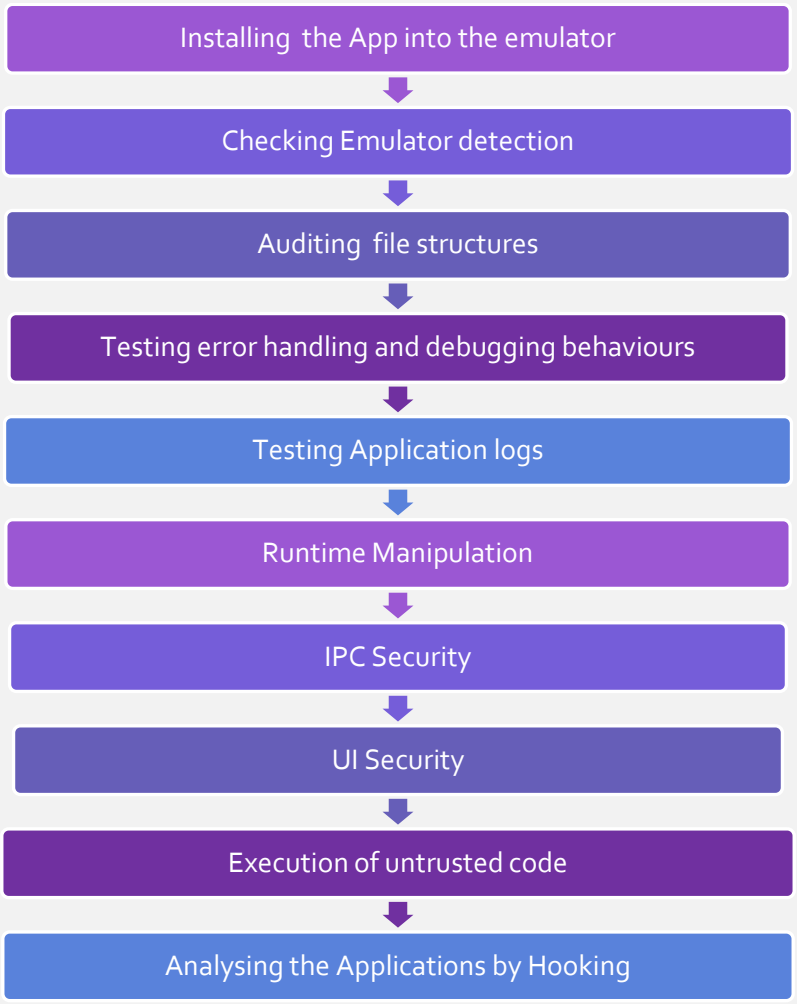
PRE REQUESTS:

1. App Package (APK/IPA)

2. Login details if required
3. Emulator for corresponding mobile platform

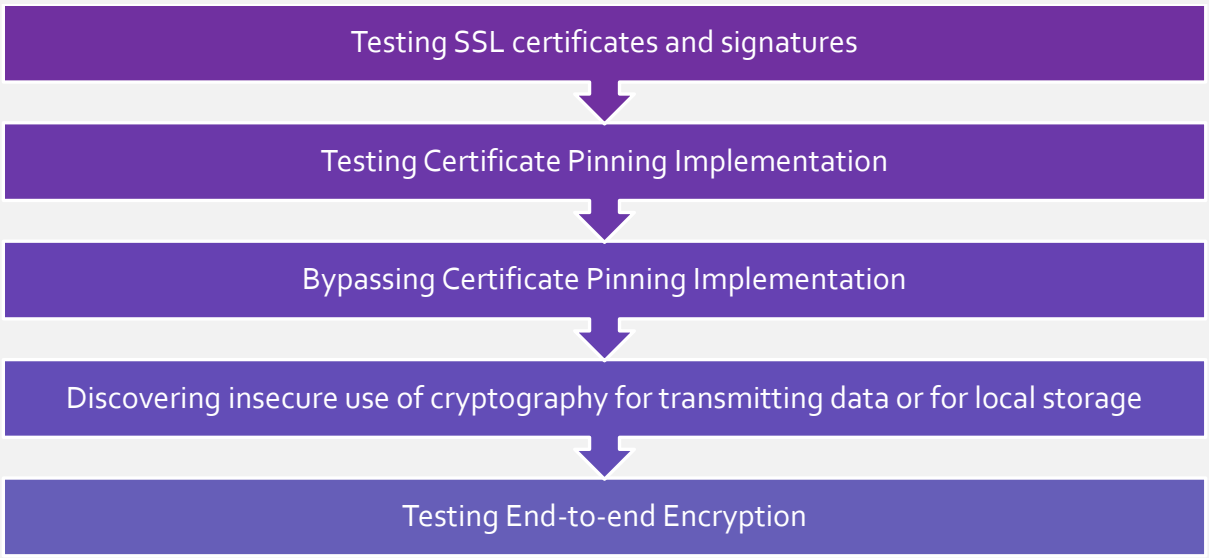
PHASE 1: DYNAMIC ANALYSIS

In the Emulator level stage, the first phase is Dynamic Analysis. Primarily we install the App in to the emulator. First, we need to verify emulator detections. Then, we need to Audit the file structure and test error handling and debugging behaviours. We need to verify the security of local file system and the Application's logs by simulating different test cases. We need to verify IPC (Inter-Process Communication), UI related issues. We also need to verify the runtime manipulation of classes by hooking the applications.



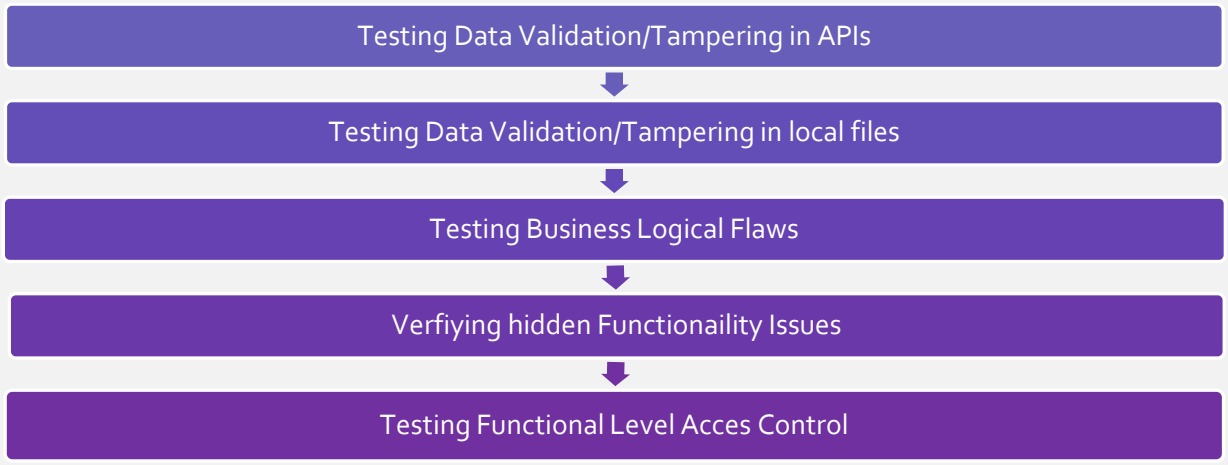
PHASE 2: COMMUNICATION LEVEL ANALYSIS:

In Emulator level stage, the second phase is communication level security test. Primarily test SSL certificates and signatures. Next, we need to verify the certificate pinning implementation and check whether it is by-passable or not. Test insecure use of cryptography for transmitting data for the local storage. Following, we should test for the end to end encryption as well.



PHASE 3: BUSINESS LOGIC ANALYSIS:

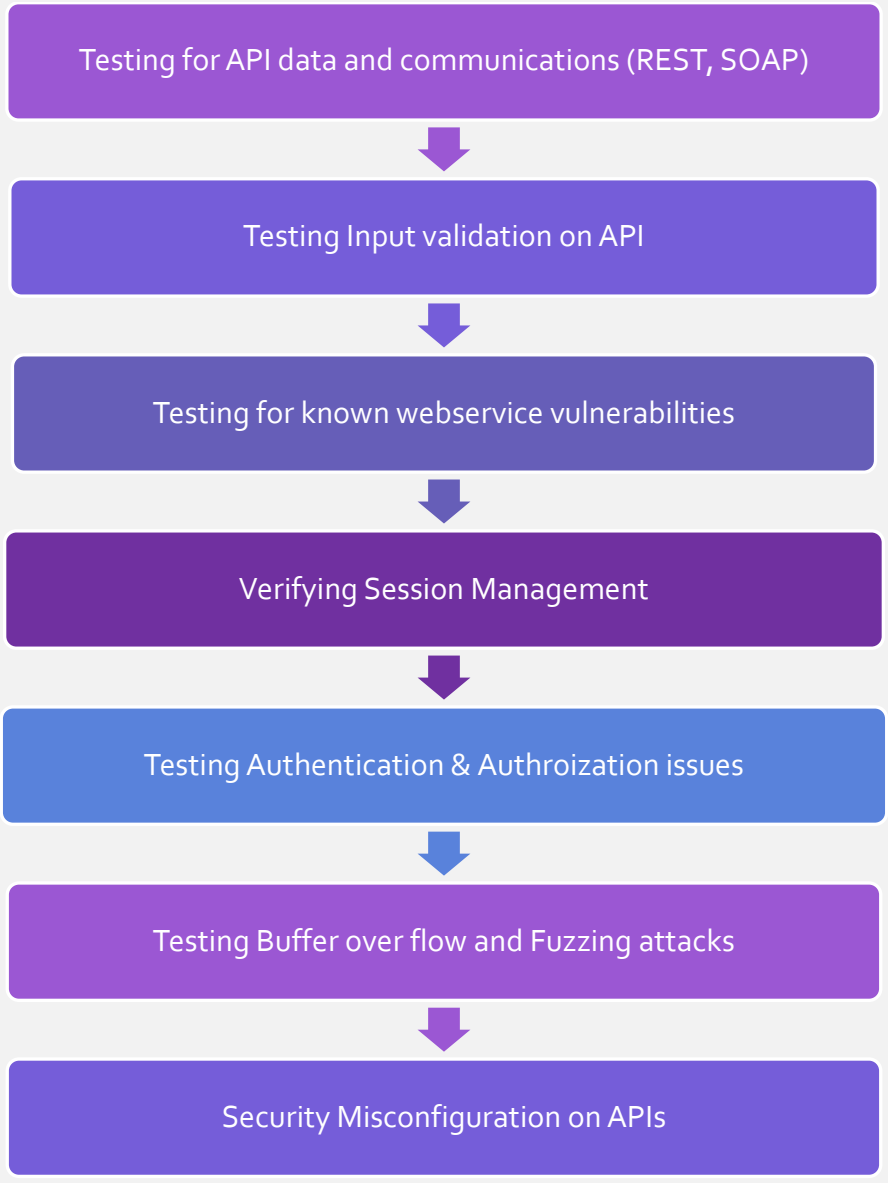
In Emulator Level stage, the third phase is business logic testing. Primarily we need to check for the logical flaws that leads to business logic issues. Further, we need to check if there are any loopholes on data flow or functional flow. We need to perform data validation and tampering data attacks in APIs also with local files created by application. Then, we must verify if there are any hidden functions and must test for known & unknown client-side vulnerabilities.



PHASE 4: SERVER-SIDE ANALYSIS

In Emulator level stage, the third phase is server-side security test. Primarily, we need to test the APIs used for data and communication (reset and soap). Further, we need to check the input validation, testing generic web application attacks through web services. We need to perform

testing buffer over flow and fuzzing attacks and must test for unknown vulnerabilities. Testing for Authentication & Authorization issues such BAC, IDOR, MFLAC, etc. must also be mandatorily done.



v

ROOTED/JAIL BROKEN LEVEL:

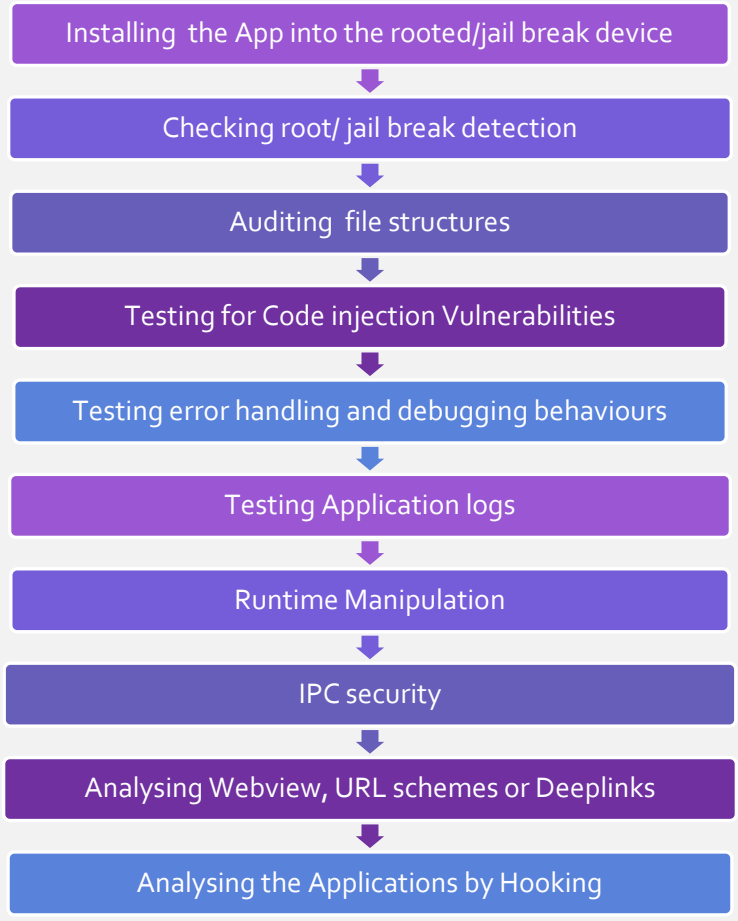
Given app will integrate with the corresponding Rooted/Jail broken device to do the complete security test.

PRE-REQUESTS:

1. App Package (APK/IPA)
2. Rooted / jail breaking device for corresponding mobile platform
3. Login details if required

PHASE 1: DYNAMIC ANALYSIS

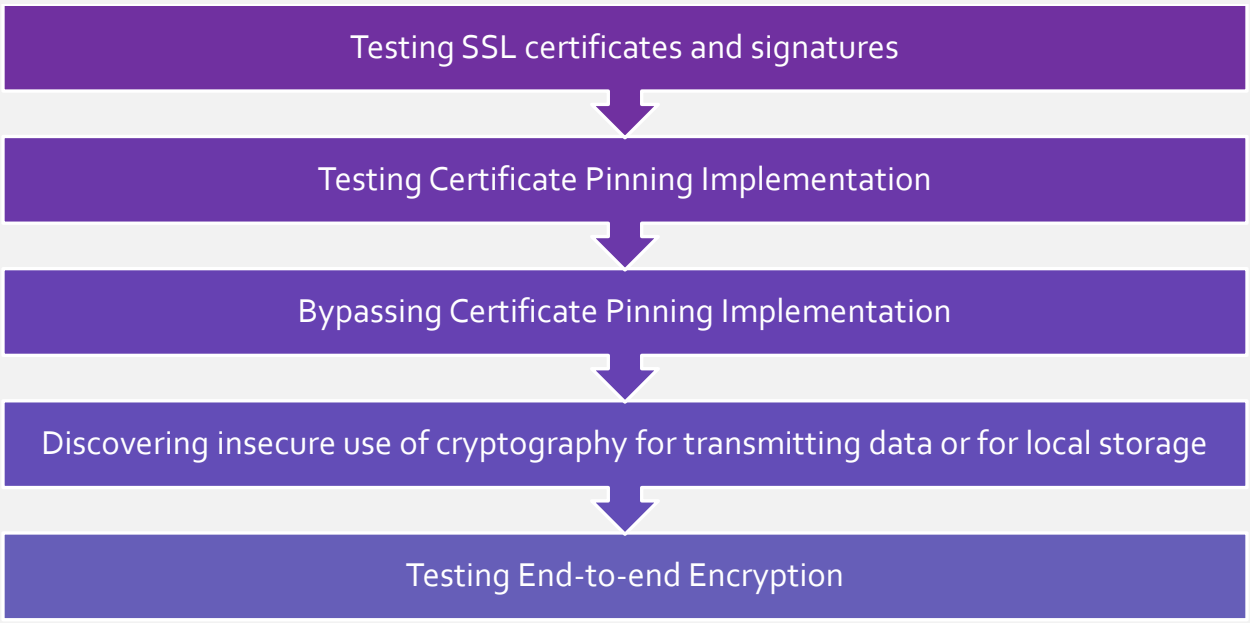
In the rooted level stage, the first phase is Dynamic Analysis. Primarily, we install the App in to the rooted device. First, we need to verify root detection mechanism. Then, we need to Audit the file structure and test error handling and debugging behaviours. We need to verify the security of local file system and must verify the Application's logs by simulating different test cases. We need to verify IPC (Inter-Process Communication), UI related issues. We need to also verify runtime manipulation of classes by hooking the applications.



PHASE 2: COMMUNICATION LEVEL ANALYSIS

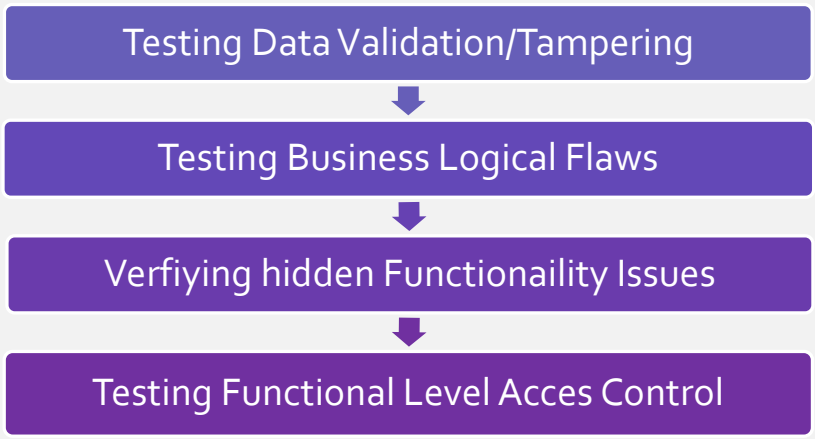
In rooted device level stage, the second phase is communication level security test. Primarily test SSL certificates and signatures. Next, we need to verify the certificate pinning implementation and

check whether it is by-passable or not. Then, we must test the insecure use of cryptography for transmitting data for the local storage. Following, we should test for the end to end encryption.



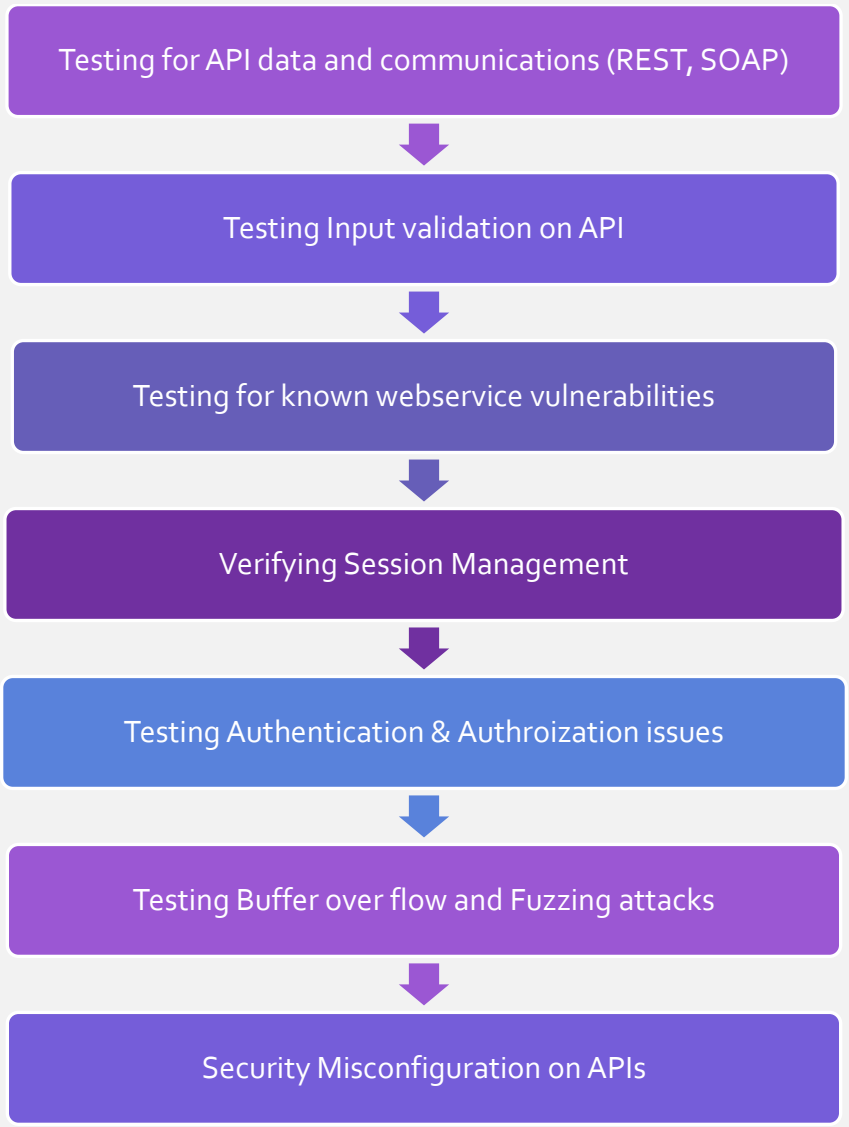
PHASE 3: BUSINESS LOGIC ANALYSIS

In rooted device Level stage, the third phase is business logic testing. Primarily, we need to check for the logical flaws that leads to business logic issues. Further, we need to check if there is any loop on data flow or functional flow. We need to perform data validation and tampering data attacks, verify if there are any hidden functions and test for known & unknown client-side vulnerabilities.



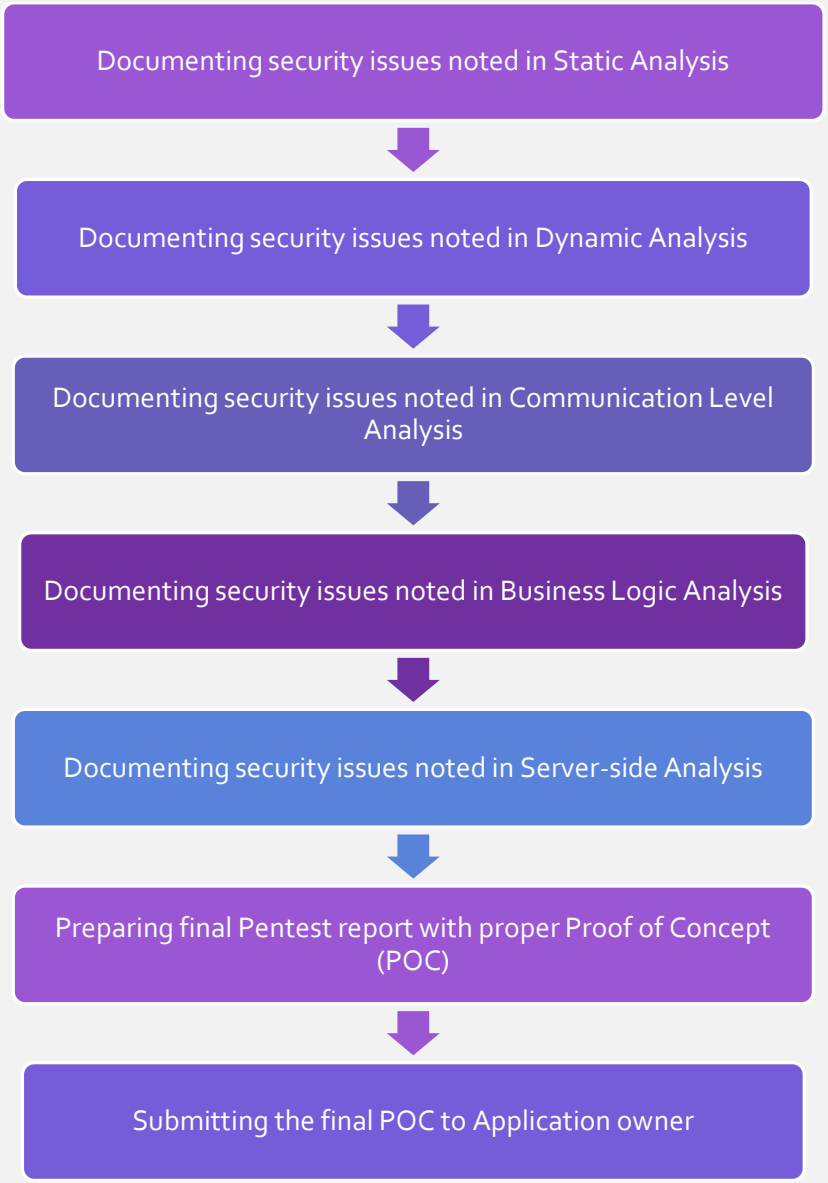
PHASE 3: SERVER-SIDE ANALYSIS

In rooted device level stage, the third phase is server-side security test. Primarily, we need to test the APIs used for the data and communication (reset and soap). Then, we need to check the input validation, testing generic web application attacks through web services. Mandatorily, we need to perform testing buffer over flow and fuzzing attacks, test for unknown vulnerabilities. Testing for Authentication & Authorization issues such BAC, IDOR, MFLAC, etc. must also be done.



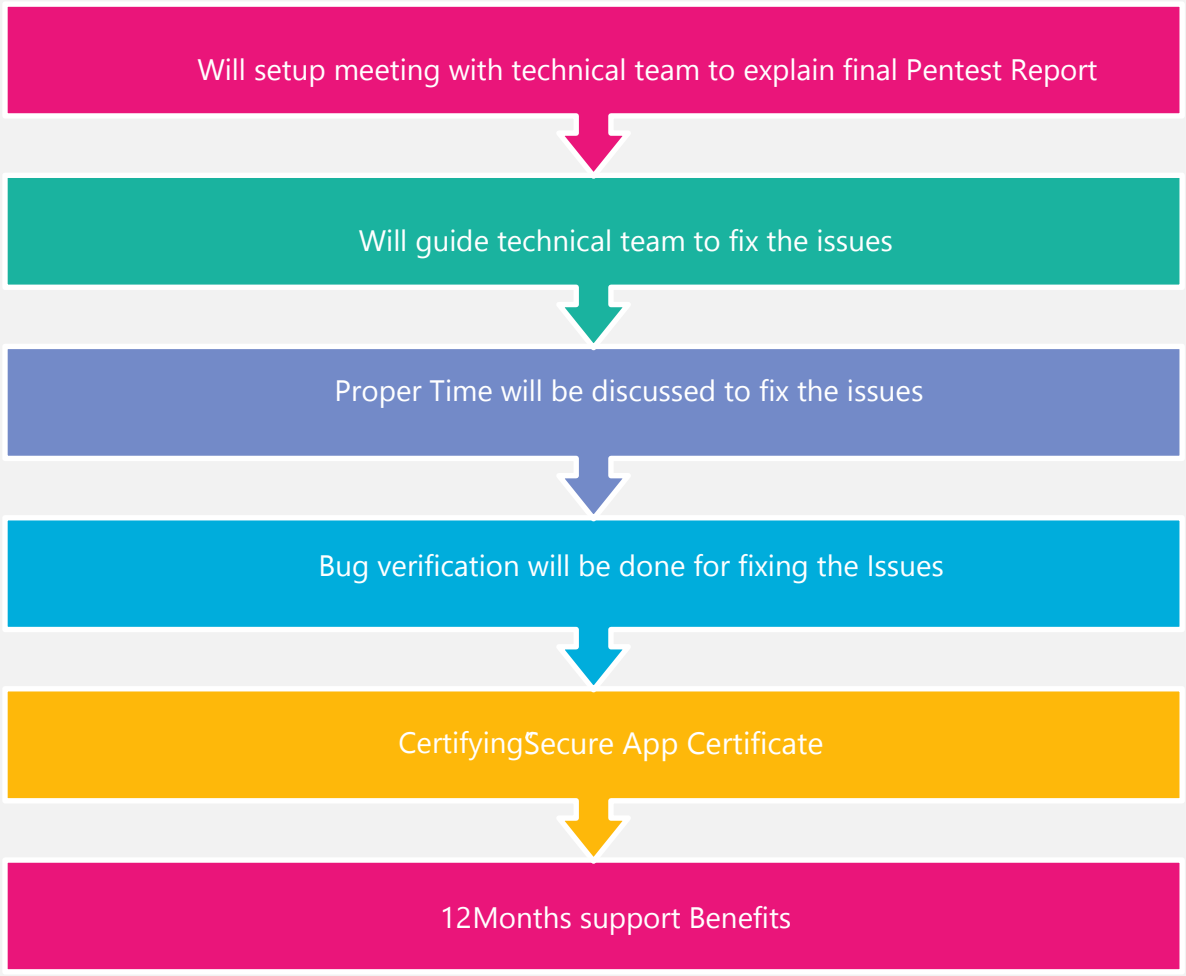
DOCUMENTATION AND SUBMISSION

Security tester should document security issues noted in the Static Analysis, Dynamic Analysis, Communication Level Analysis, Business Logic Analysis and Server-Side Analysis with three different stages. Then, we need to prepare the pen test report with Proof of Concept (POC) and finally we need to submit the POC (Proof of Concept) to the application owner.

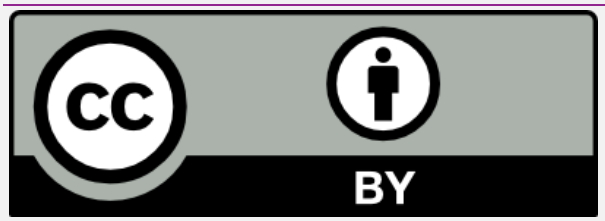


BUG VERIFICATION

In the bug verification process, we will setup a meeting with technical team to explain the final Pentest Report, guide the technical team to fix the issues and we will also perform bug verification for the fixed issue. Finally, we should provide a certificate "Secure App Certificate" and a 12 month support benefit is provided.



INTERNALIONALIZATION



Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons Public Licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subjected to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only and aren't exhaustive nor form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material that is not subjected to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors.

Considerations for the public: By using one of our public licenses, a licensor grants the public, permission to use the licensed material under specified terms and conditions. If the licensor's permission is not necessary for any reason–for example, because of any applicable exception or limitation to copyright–then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests which were reasonable. More considerations for the public.

GLOSSARY

- .APK – Android application package (APK) is the package file format used by the Android operating system for distribution and installation of mobile apps.
- .IPA - An .IPA file is an iOS application archive file which stores an iOS app.
- REVERSE ENGINEERING - Reverse engineering is the processes of extracting knowledge or design information from anything man-made and re-producing it or reproducing anything based on the extracted information.
- SSL - The Secure Sockets Layer (SSL) is a protocol that manages server authentication, client authentication and encrypted communication between servers and clients.
- GET - A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
- POST - A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
- PUT - Replaces all current representations of the target resource with the uploaded content.
- DELETE - Removes all current representations of the target resource given by a URI.
- REST - Representational state transfer (REST) or RESTful web services are one of the ways for providing interoperability between computer systems on the internet.
- SOAP - SOAP is an XML-based messaging protocol. It defines a set of rules for structuring messages that can be used for simple one-way messaging but is particularly useful for performing RPC-style (Remote Procedure Call) request-response dialogues.

- HTTP - The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.
- CRYPTOGRAPHY - Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it.
- API - Application Program Interface (API) is a set of routines, protocols, and tools for building software application.
- XML - Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine readable.
- JSON - JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write.
- BUFFER OERFLOW - A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold or when a program attempts to put data in a memory area past a buffer.
- FUZZING - Fuzz testing is a software testing technique which uses a random data as the inputs to the system. If the application fails, then those issues/defects are to be addressed by the system.
- EMULATOR - An Emulator is a program that pretends to be another particular device or program that other components expect to interact with.
- ROOTED DEVICES - Rooting is the process of allowing users of smartphones, tablets and other devices running the Android mobile operating system to attain privileged control (known as root access) over various Android subsystems.
- MITM ATTACK - The Man-In-the Middle (MITM) attack intercepts a communication between two systems.
- SNIFFFING - Any eavesdropping on existing network traffic is called sniffing,