Bangladesh University of Engineering and Technology
Department of Computer Science and Engineering
CSE 406: Computer Security Sessional

Report on MobSF

(Mobile Security Framework)

Submitted by:

Ridwanul Haque

Student ID: 1705111

Date of submission : 12 September, 2023

Contents

1 Introduction

In an increasingly interconnected and digitized world, mobile devices have become an integral part of our daily lives. Smartphones and tablets have revolutionized the way we communicate, work, shop, and entertain ourselves. However, this ubiquity and dependence on mobile technology have brought forth new and complex security challenges that demand our attention and vigilance.

Mobile security, also known as mobile device security or mobile application security, refers to the comprehensive set of measures, practices, and technologies designed to protect mobile devices, their data, and the applications that run on them from various threats and vulnerabilities. These threats can range from malware and data breaches to privacy intrusions and device theft.

The rapid evolution of mobile technology has not only enhanced our lives but has also attracted the attention of malicious actors who exploit vulnerabilities for their gain. Cybercriminals continually adapt and develop sophisticated techniques to compromise mobile devices and access sensitive information, making mobile security an ever-evolving field that demands constant adaptation and innovation.

As we journey through the realm of mobile security, it becomes evident that our ability to harness the potential of mobile technology depends on our ability to protect it. Through a comprehensive understanding of mobile security principles, best practices, and the tools at our disposal, we can empower individuals and organizations to make the most of the mobile revolution while minimizing the associated risks.

2 Mobile Security Framework(MobSF)

2.1 What is MobSF?

Mobile Security Framework (MobSF) is an open-source, automated mobile application (app) security testing framework designed to help developers, security professionals, and organizations assess the security of mobile applications. MobSF provides a set of tools and features for analyzing the security of both Android and iOS mobile apps.

2.2 History and Development

2.2.1 Initial Development

MobSF started with a small group of developers or an individual who recognizes a need for a specific tool or framework in the field of mobile application security in May, 2015.MobSF began as a response to the growing need for automated mobile app security testing tools.

2.2.2 Community Involvement

Successful open-source project like MobSF tends to attract a community of contributors and users interested in improving and using the tool. These contributors come from various backgrounds, including mobile app developers, security researchers, and cybersecurity professionals.

2.2.3 Version Releases and Iterations

Over time, MobSF would have gone through multiple version releases, with each iteration bringing improvements, bug fixes, and new features. The development team and community would actively

address security vulnerabilities and adapt to changes in mobile technology and security standards.

2.2.4 Documentation and User Support

Maintaining comprehensive documentation and providing user support is crucial for the success of open-source projects. Clear and up-to-date documentation helps users understand how to use the tool effectively.

3 Installation and Setup

3.1 System Requirements

3.1.1 Hardware Requirements

Minimum 4GB RAM, 5GB HDD/SSD and Virtualization Support for running MobSF VM and Intel HAXM if user is running MobSF ARM Emulator.

3.1.2 Software Requirements

Python 3.6 — Python 3.10. Oracle JDK 1.7 or above. Mac OS Users must install Command-line tools. iOS IPA Analysis works only on Mac and Linux. Windows App Static analysis requires a Windows Host or Windows VM for Mac and Linux.

3.2 Installation Steps

- Clone the MobSF Repository:
  git clone https://github.com/MobSF/Mobile-Security-Framework-MobSF.git
- Navigate to the MobSF Directory:
  cd Mobile-Security-Framework-MobSF
- Install Python Dependencies:
  pip install -r requirements.txt
- Set Up Virtual Environment (Optional but Recommended):
  virtualenv venv
- Launch MobSF:
  ./setup.sh
- Access MobSF Web Interface:
  Go to web browser and navigate to http://127.0.0.1:8000
- Shutting Down MobSF:
  Press 'Ctrl+C' to terminate the server.

4 Key Features

4.1 Multi-Platform Support

MobSF is designed to work with both Android and iOS mobile applications, making it a versatile tool for assessing the security of apps on different platforms.

4.2 Static Analysis

MobSF performs static analysis of mobile app binaries and source code. It examines the app's code and configuration files to identify potential vulnerabilities and security issues.

4.3 Dynamic Analysis

It conducts dynamic analysis by running the mobile app in a controlled environment. This allows MobSF to detect runtime vulnerabilities and behaviors that might not be apparent through static analysis alone.

4.4 Reverse Engineering

MobSF includes features for reverse engineering mobile apps. This capability is useful for understanding an app's functionality, identifying vulnerabilities, and assessing potential security risks.

4.5 Reporting

MobSF generates detailed reports summarizing its findings. These reports typically include identified vulnerabilities, potential security risks, and recommendations for remediation. Clear and informative reports are essential for security professionals and developers to take action.

4.6 Integration

MobSF can be integrated into continuous integration/continuous deployment (CI/CD) pipelines. This enables automated security testing as part of the app development and release process, helping organizations maintain security throughout the development lifecycle.

4.7 Extensibility

The framework supports the use of plugins and extensions. This means that users can customize and extend MobSF's functionality to meet specific testing requirements or to integrate it with other tools and systems.

4.8 Malware Analysis

MobSF can help identify and analyze malware or potentially harmful components within mobile apps, enhancing security assessments and threat detection.

4.9 Web API Testing

It can analyze the security of APIs used by mobile apps. This is crucial for identifying vulnerabilities in API endpoints that could be exploited by attackers.

4.10 Code Review Integration

MobSF can integrate with code review tools, making it easier for development teams to identify and fix security issues in the early stages of app development.

4.11 Database Assessment

The framework includes features for assessing the security of databases used by mobile apps, ensuring that sensitive data is properly protected.

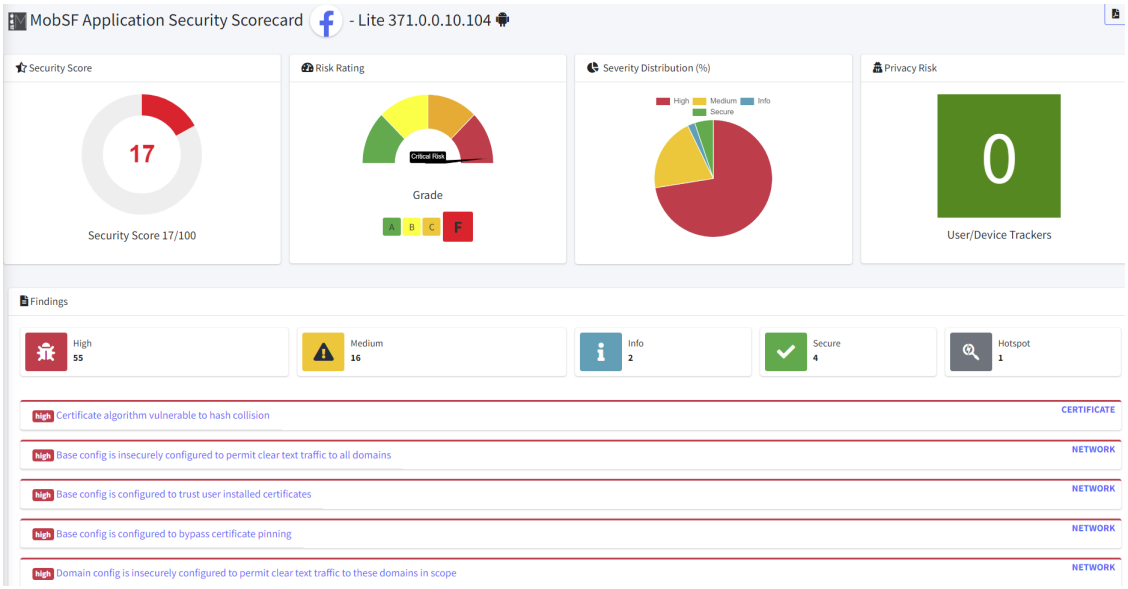
4.12 Ease of Use

While MobSF provides advanced security testing capabilities, it is designed to be user-friendly, with a straightforward interface that allows both security professionals and developers to use it effectively.

5 Functionality and Scanning Mobile Apps

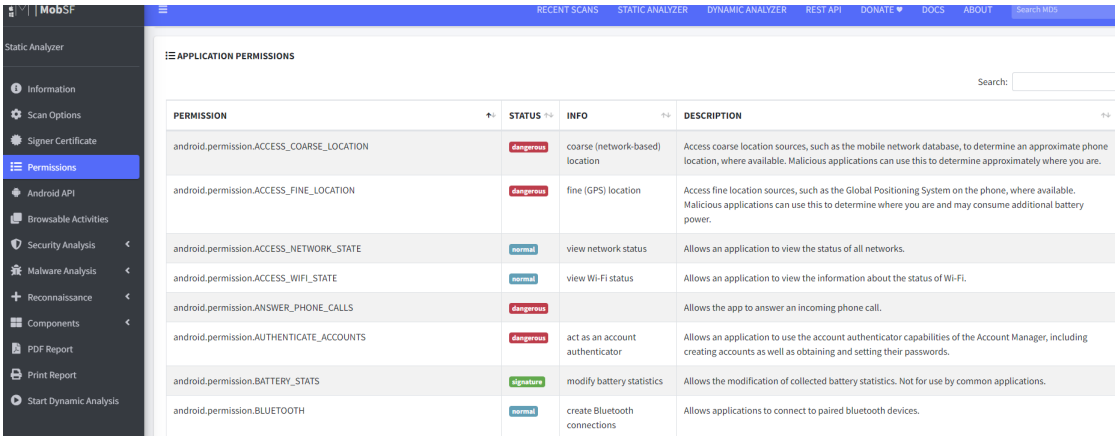
5.1 App Security Scorecard

MobSF can display a security scorecard to visualize app security.



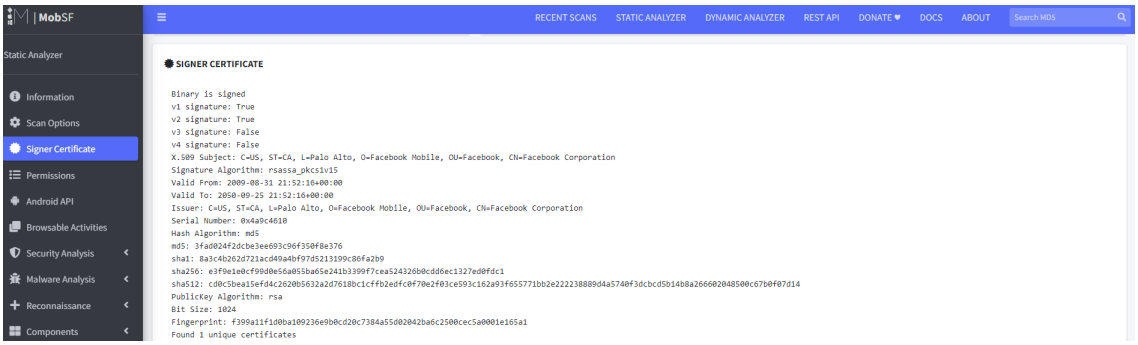
5.2 App Permissions

Application Permissions are seen for the app.



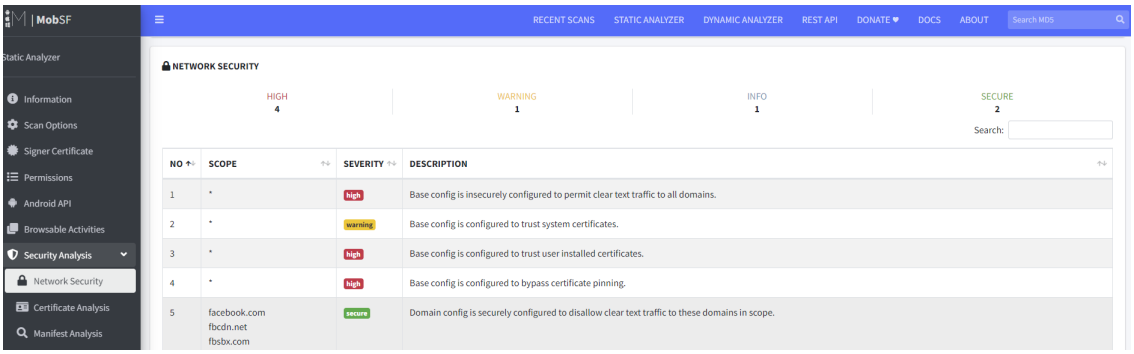
5.3 Signer Certificate

MobSF allows to see signer certificate.



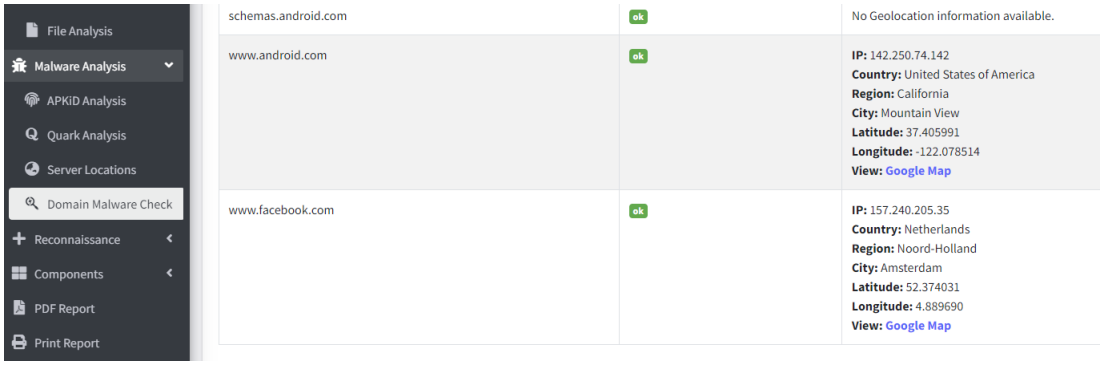
5.4 Network Security

Provides a brief description on application's network security.



5.5 Malware Analysis

MobSF can detect potential malware threats in the application.



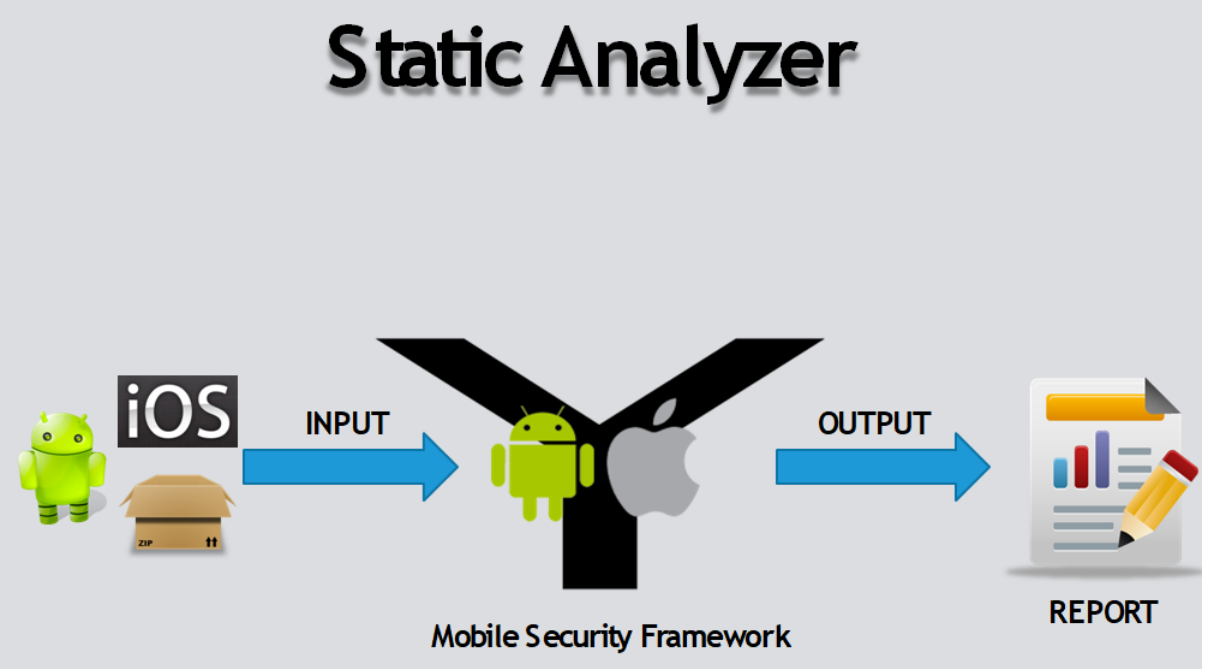
5.6 Supported File Formats

MobSF supports mobile app binaries (APK, XAPK, IPA APPX) along with zipped source code and provides REST APIs for seamless integration with your CI/CD or DevSecOps pipeline.The Dynamic Analyzer helps to perform runtime security assessment and interactive instrumented testing.

6 Security Assessments

6.1 Static Analysis

Static analysis in Mobile Security Framework (MobSF) is the process of examining the source code, binary files, and configuration files of an Android or iOS mobile app without executing the app. This analysis helps identify potential vulnerabilities, security issues, and coding mistakes that could pose risks to the app's security.

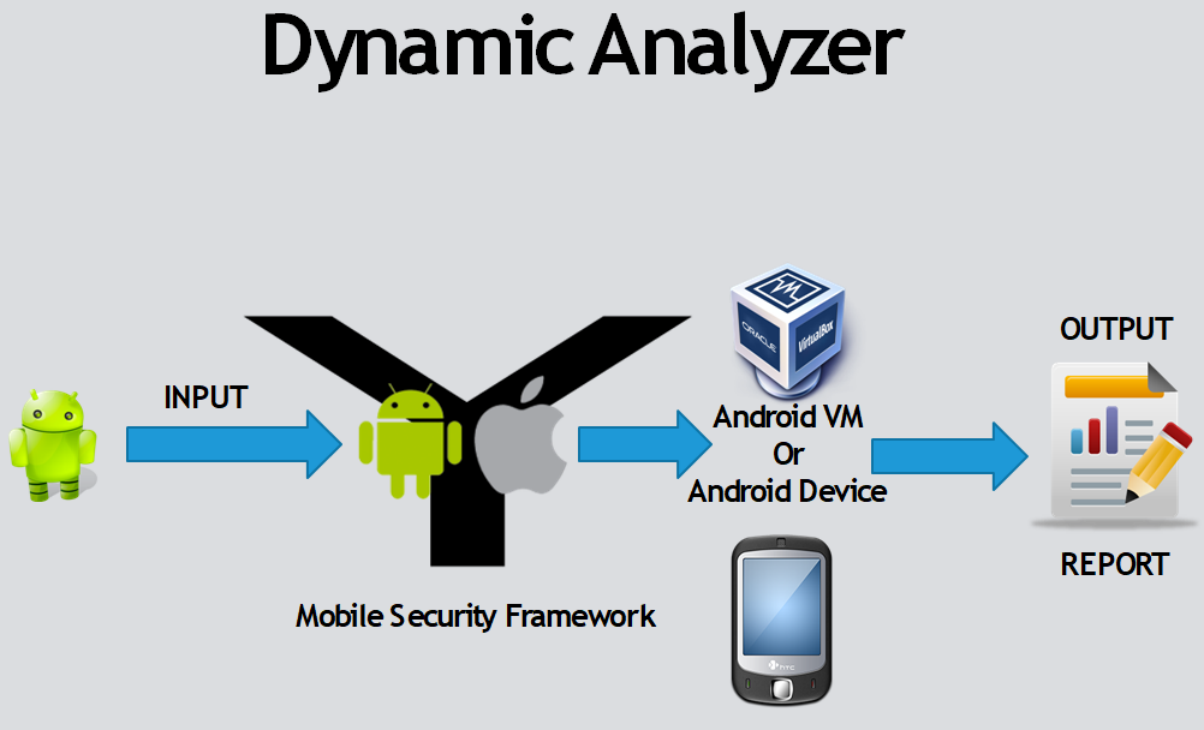


Static analysis on top apps-criteria:

- SSL Bypass in Native Code
- SSL Bypass in Web View
- Remote Web View Debugging

6.2 Dynamic Analysis

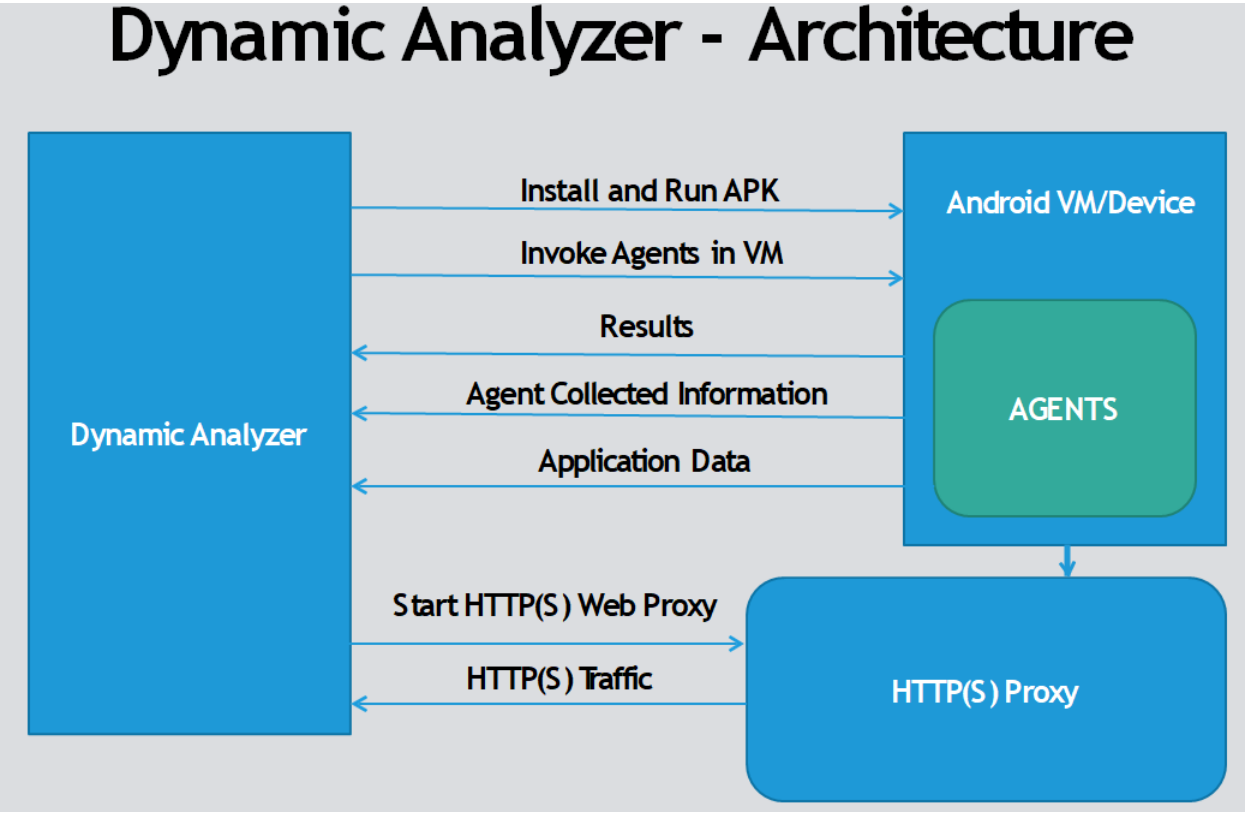
Dynamic analysis in Mobile Security Framework (MobSF) involves running a mobile app in a controlled environment to observe its behavior, monitor network interactions, and identify runtime vulnerabilities. This type of analysis provides insights into how the app behaves when executed and helps detect issues that may not be apparent through static analysis alone.



Dynamic SSL Testing:

- Dynamically verify if SSL connections are securely implemented.
- Disable JustTrustMe and Remove MobSF Root CA.
- If we can still access the decrypted HTTPS Web Traffic then that means the app is bypassing SSL errors.

Dynamic Analyzer Architecture:



Dynamic Exported Activity Tester:

```
<activity
    android:name=".ExportedActivity"
    android:label="ExportedActivity"
    android:exported="true">
</activity>
<activity
    android:name=".ImplicitlyExportedActivity"
    android:label="ImplicitlyExportedActivity" >
    <intent-filter>
        <action android:name="opensecurity.vulnapp.INTENT"/>
    </intent-filter>
</activity>
```

Challenges in Dynamic Analysis

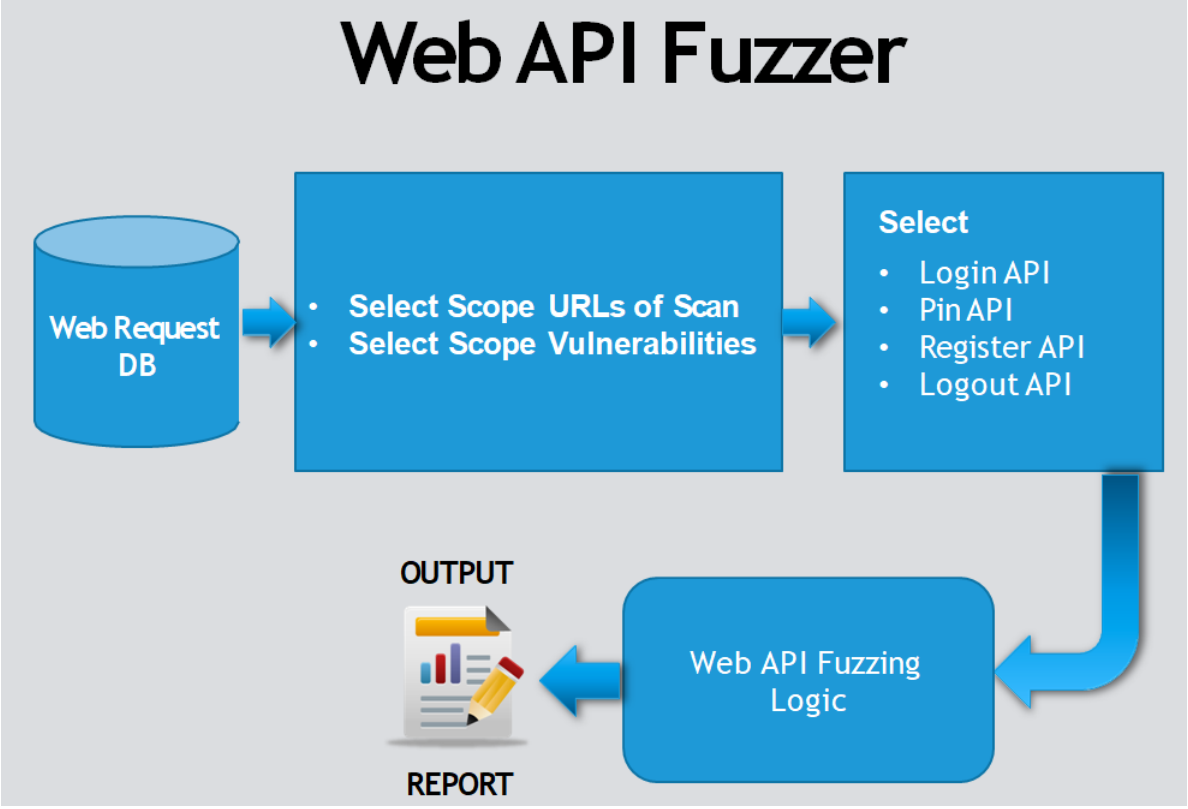
- Some Android Apps are built with security in mind.
  - Anti VM Detection
  - Anti Root Detection
  - Anti MITM with Certificate Pinning.
  - Some Apps / Malwares have sophisticated methods to detect Virtual Machines.

How to deal with these Challenges

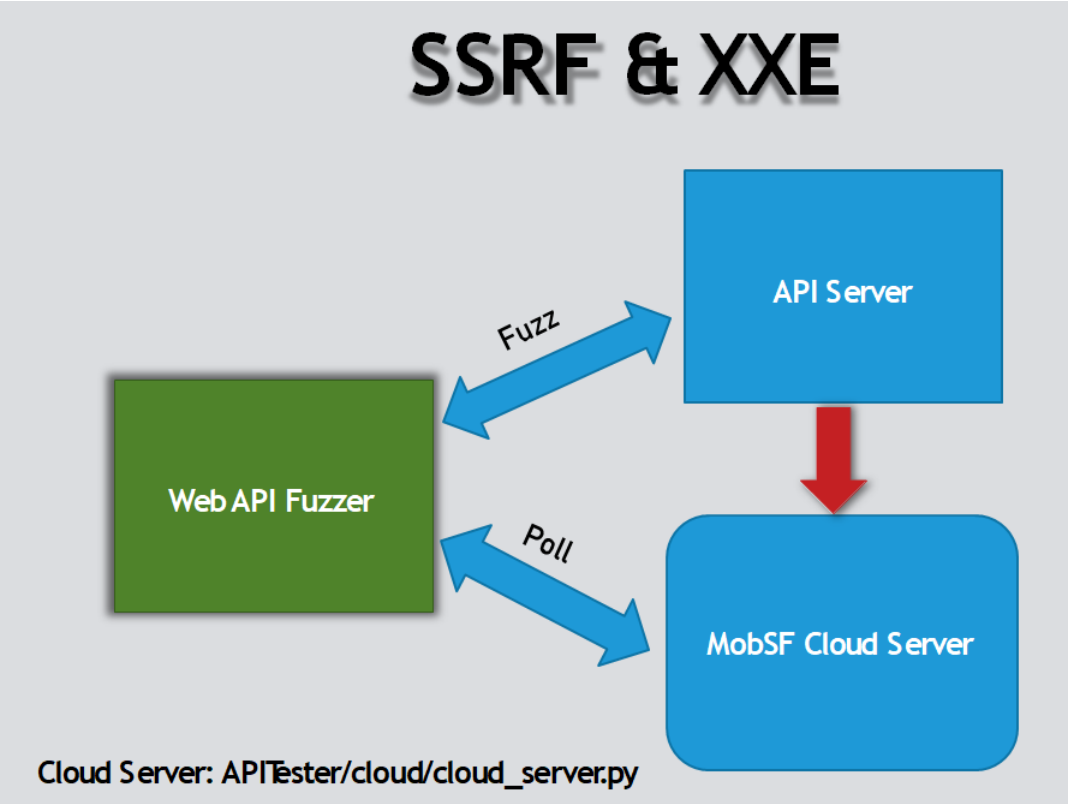
- API overriding with Xposed Framework
  - Anti VM Detection Bypass -> Android Blue Pill
  - Anti Root Detection Bypass -> RootCloak
  - Anti MITM Certificate Pinning Bypass -> JustTrustMe
- APK smali Patching.
- For sophisticated apps and malware, Use a real device for dynamic analysis.

6.3 Web API Fuzzer

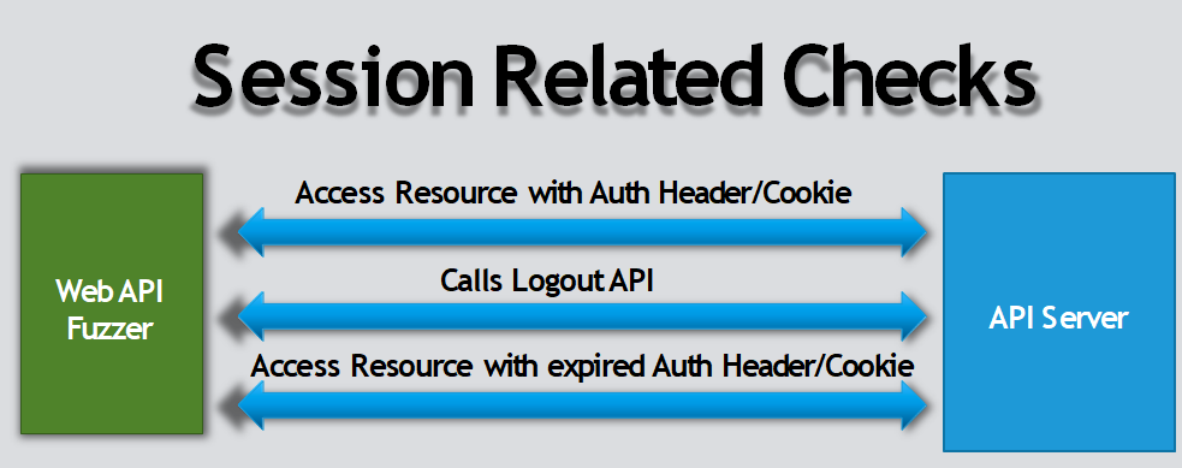
Web API fuzzing: Web API fuzzing performs fuzz testing of API operation parameters. Fuzz testing sets operation parameters to unexpected values in an effort to cause unexpected behavior and errors in the API backend. This helps us discover bugs and potential security issues that other QA processes may miss.



How to Detect?



How Session Related Checks are Performed



7 Use cases

7.1 Mobile App Developers

Mobile app developers can use Mobile Security Framework (MobSF) for:
Security Testing: Identify and fix security issues during app development.
Code Review: Analyze code for vulnerabilities and make improvements.
CI/CD Integration: Automate security testing in development pipelines.
API Security: Ensure APIs used in apps are secure.
Custom Plugins: Extend MobSF for specific testing needs.
Security Education: Learn about mobile app security.
Library Security: Assess third-party library security.
Variant Testing: Test different app builds for security.
Rapid Assessment: Quickly evaluate app security before releases.

7.2 Security Professionals

Security professionals can use Mobile Security Framework (MobSF) for:
Mobile App Assessment: Analyze apps for vulnerabilities.
Penetration Testing: Test app security for weaknesses.
Incident Response: Investigate and analyze mobile security incidents.
Forensics: Examine mobile apps during digital investigations.
Threat Hunting: Identify and mitigate mobile threats.
Security Research: Explore emerging mobile security issues.
Security Awareness: Educate teams about mobile threats and defenses.

7.3 Penetration Testers

Penetration testers can use Mobile Security Framework (MobSF) for:
Mobile App Security Testing: Assess apps for vulnerabilities.
Dynamic Analysis: Analyze runtime behavior and network interactions.
Static Analysis: Inspect source code and binaries for issues.
Exploitation Testing: Identify and exploit app weaknesses.
API Security Assessment: Test APIs for vulnerabilities.
Reporting: Generate detailed reports for clients or teams.
Custom Plugin Development: Extend MobSF for unique testing needs.

7.4 Researchers

Researchers can use Mobile Security Framework (MobSF) for:
Security Studies: Investigate mobile app vulnerabilities and threats.
Experimentation: Conduct research on emerging mobile security issues.
Tool Enhancement: Develop custom plugins to extend MobSF's capabilities.
Data Collection: Gather data on mobile app security trends.
Security Innovation: Contribute to mobile security research and advancements.

8 Challenges and Limitations

8.1 Limitations of MobSF

1. Static Analysis Limitations:
   MobSF's static analysis may not uncover all vulnerabilities, especially those that are context-dependent or require runtime conditions to manifest.

2. Dynamic Analysis Challenges:
   The accuracy of dynamic analysis may be affected by the complexity of the app and the effectiveness of the controlled execution environment.

3. False Positives/Negatives:
   Like many security scanning tools, MobSF can produce false positives and negatives, requiring manual verification of results.

4. Limited Platform Support:
   While it supports Android and iOS, some specific app types or versions may not be fully compatible.

5. Outdated Vulnerability Definitions:
   MobSF relies on vulnerability definitions and databases, which may not always be up to date with the latest threats and vulnerabilities.

6. Custom Code Detection:
   Detecting custom or proprietary security measures implemented by app developers can be challenging for automated tools.

7. Lack of Real-World Threat Simulation:
   While it provides valuable security insights, MobSF does not simulate real-world threat scenarios, such as social engineering or phishing attacks.

8. Privacy Testing Limitations:
   The tool may not comprehensively assess privacy-related issues, including data leakage and privacy policy violations.

9. Performance Impact:
   Dynamic analysis can be resource-intensive and may slow down the scanning process, making it less practical for large apps.

10. Community Support Dependency:
    Users rely on community support for updates, bug fixes, and new features, which can be inconsistent.

9 Conclusion

9.1 Mobile Security Framework

In today's digital landscape, where mobile apps play a pivotal role in our daily lives, the importance of Mobile Security Framework (MobSF) cannot be overstated. MobSF serves as a critical line of defense against the ever-evolving landscape of mobile security threats.

For developers, MobSF is a tool that empowers them to build apps with confidence, ensuring they meet the highest security standards. For security professionals, it provides a robust platform for comprehensive assessments, helping to identify vulnerabilities and protect against potential breaches. Researchers benefit from MobSF's capabilities in exploring and understanding the dynamic world of mobile app security.

In an age where mobile devices are central to our personal and professional activities, the need for robust mobile app security has never been greater. MobSF plays a pivotal role in strengthening this security, contributing to safer, more trustworthy, and resilient mobile applications that protect both users and organizations. Its importance lies not only in what it is today but also in its potential to evolve and adapt to future challenges in the dynamic realm of mobile security.

9.2 Final Thoughts

In conclusion, the Mobile Security Framework (MobSF) serves as a valuable tool for assessing the security of mobile applications on both Android and iOS platforms. Its capabilities, including static and dynamic analysis, reporting, and extensibility, make it an essential asset for security professionals, developers, and researchers.

While MobSF has its limitations, its continued development and community support are likely to result in further improvements, enhancing its effectiveness in identifying and mitigating mobile app vulnerabilities and security threats. Whether used for securing mobile apps during development or conducting in-depth security assessments, MobSF plays a crucial role in the ongoing effort to bolster mobile app security.

10 References

- *https://github.com/MobSF/Mobile-Security-Framework-MobSF*
- *https://medium.com/@kshitishirke/mobile-security-framework-mobsf-static-analysis-df22fcdae46e*