بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

Vulnerability Assessment of Mobile Financial Service Applications In Bangladesh

Authors
Md. Tanvir Rahman - 170042006
Tasnuva Mehnaz - 170042037
Mahbubur Rahman - 1700420054

Supervisor
Ashraful Alam Khan
Assistant Professor Department of CSE,
Islamic University of Technology (IUT)
Co-Supervisor
S.M. Sabit Bananee
Lecturer, Department of CSE,
Islamic University of Technology (IUT)
Imtiaj Ahmed Chowdhury
Lecturer, Department of CSE,
Islamic University of Technology (IUT)

A thesis submitted in partial fulfilment of the requirements for the degree of B. Sc. Engineering in Computer Science and Engineering (CSE)

Academic Year: 2020-2021

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT),
A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)
Gazipur-1704, Dhaka, Bangladesh

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by *Md. Tanvir Rahman*, *Tasnuva Mehnaz* and *Mahbubur Rahman Ashraful Alam Khan*, Asst. Professor of the Department of Computer Science and Engineering (CSE), *S.M. Sabit Bananee* , Lecturer of the Department of Computer Science and Engineering (CSE), *Imtiaj Ahmed Chowdhury* , Lecturer of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh.
We are very grateful to our supervisors *Prof. Ashraful Alam Khan*, Department of Computer Science and Engineering, Islamic University of Technology (IUT), *S.M. Sabit Bananee*, Department of Computer Science and Engineering, Islamic University of Technology (IUT), *Imtiaj Ahmed Chowdhury* , Department of Computer Science and Engineering, Islamic University of Technology (IUT), for their supervision, knowledge and support, which has been invaluable for us.
It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Tanvir Rahman

Student ID - 170042006

Tasnuva Mehnaz

Student ID - 170042037

Mahbubur Rahman Durjoy

Student ID - 170042054

.

Approved By:

Supervisor:

Ashraful Alam Khan
Assistant Professor, Department of CSE

Co-Supervisor:

S.M. Sabit Bananee	Imtiaj Ahmed Chowdhury
Lecturer, Department of CSE	Lecturer, Department of CSE

Abstract

In the era of the internet and smartphone, digital financial transaction services make people's lives easier. In Bangladesh there are several prominent Mobile Financial Service (MFS) provider companies which give service to their customers. Companies launch android and iOS applications to make this service more reachable to their clients.These applications help people to purchase, pay or transfer money with the smartphone caring in his/her hand. While developing, for many reasons like coding flaws, logical errors, misconfiguration or vulnerable architectural design in applications could lead to compromise of that system. So security is the main concern here to keep client's money safe. This study is able to analyze the vulnerability level of top five Mobile Financial Service (MFS) applications available in Bangladesh. Popular scanning tools like Mobile Security Framework (MobSF), QUIXXI and Immuniweb are used to analyze this study following the Open Source Web Application Project (OWASP) Mobile Application Security Verification Standard (MASVS). After analyzing and comparing all the reports from selected tools are merged together and break down the application level vulnerabilities. Among the common issues found in almost every application, the percentage of High risk is alarming. This indicates a concern for the organization to defend against any cyber attack and also for losing reputation.

Contents

List of Figures

1 Introduction

Mobile applications have been established as an intrinsic part of our daily life. We enjoy useful services through many mobile applications .There are many mobile financial services in Bangladesh which are being used everyday . A single vulnerability to these app can cost us financial loss, reveal our personal information and even endanger national security through theft of critical military data. Hackers stole 1 million usernames and passwords from Sony in 2011(1) , 45million customer credit cards from TJ Maxx in 2007(2) , 100 million customer credit cards from Heartland Payment Systems in 2009 (3) and also stole 24,000 Bitcoins from BitFloor in 2012(4), a major Bitcoin exchange . Akamai Technologies, a leading provider of content delivery services, reported in its state of the internet for the third quarter of 2014 that its customers are facing increased distributed denial-of-service in terms of both the bandwidth and the number of requests generated by the attackers(5).

This research conducted an empirical analysis of vulnerability detection using MobSF, ImmuniWeb and Quixxi.The obtained results are analyzed and categorized as high, medium, low, and informational level security flaws depending upon their severity in terms of the potential damage they are capable of doing to the system or the end users. Then, results of both tools have been compared, where (write the tool name here)was more successful detecting vulnerabilities in the examined web applications.CSRF, one of the most lethal vulnerabilities, was prevalent among all the tested web applications, hence this vulnerability was aptly ranked as number one.

2 Background Study

One of the greatest advances in technology is the invention of mobile applications. 83.72% of the world population owns a smartphone and they use different types of mobile applications.Mobile Applications have become so important and useful in our daily life for their usefulness and usability. They are very much needed in our life to conduct daily activities (Social Media, Food Delivery, Transport, Banking etc.). But these mobile applications are exposed to various vulnerabilities .Mobile Applications are mostly targeted by attackers because they process a lot of users' personal and sensitive information, access them and in some cases enable remote control of devices. Higher possibility of attacks and compromises arise while there is more data flow in the mobile applications . Besides,most developers are very much focused on the features , performance , usability etc. because user requirements are very much focused on those areas . Also the developers are in continuous pressure to implement the features and thus the security is often neglected. Also they are focused on the requirements which are more relevant from a business perspective . The most important applications to be focused on to strengthen security are : Mobile Financial Service or MFS . In Bangladesh, most of the people use these applications for instant money transaction and deposition, for paying bills etc.In the following section ,different vulnerabilities are briefly discussed . (6)

2.1 Unsafe Files Deletion

Misconfiguration in a software system can damage the software greatly. Mobile apps often rely on connection with the server but most of the time mobile application developers don't have traditional server-side security knowledge. If the access permission for the critical files that are accessible only by the authorized people are not set properly then anyone may corrupt those files. And this arbitrary file is a file that allows malicious people to modify everything on a system. An attacker who is unauthenticated can exploit this vulnerability remotely while sending a crafted request to the target system. If exploitation is successful, arbitrary files from the target system are deleted and may cause to denial of service conditions if important executables or libraries are deleted. This issue enables an attacker to influence calls to the 'unlink()' function and delete arbitrary files. An attacker can supply directory traversal sequences followed by an arbitrary file name to delete specific files for the lack of input validation.

2.2 Enable Cleartext Traffic

encrypted.It is recommended that Android applications protect against unintended regressions to clear text communication and many Android applications have already included a secure-only mode. But some of them have a tendency to revert to cleartext communications by mistake. For example, a server component update might result in the app receiving HTTP URLs instead of HTTPS URLs from the server. A clear text message would then be sent to the user, and there would be no user-visible symptoms. The app's users may not be aware of this problem.

2.3 Undefined Protection Level Of Exported Service

Modern software architectural design helps to integrate one application's service to another application to make it more user friendly. This type of service is known as exported service. But this exported service's permission should be protected. Otherwise any other component has the chance to use it without permission.

2.4 Raw Execution Of SQL Query

Based on user input, an application performs a data query to retrieve or alter data in the database. Since databases are the main target for an attacker, it's very important to set up a strong protest against receiving malicious input from untrusted users. Proper setup of regular expression can help to mitigate the risk of unwanted change in the database. Besides that instead of supplying user input in sql query directly, it's better to practice using placeholders.

2.5 Missing Certificate Pinning

Certificate pinning is a method that depends on server certificate verification on the client side and server certificate or fingerprint is necessary to be known previously to the mobile app for this verification.While connection with the server is established the fingerprint is compared with a certificate from the remote server . Missing certificate pinning can lead to giving unwanted authorization or not being able to authorize the right person . This protection mechanism stops different kind of attack like man-in-the-middle, compromising CA etc

2.6 Insecure Webview Implementation

In Android applications, WebViews are used to load content and HTML sites. WebView's implementation must be secure in order to avoid exposing the application to a significant risk. Because WebView is less safe, this post aims to look at how to fix these problems.

3 Literature Review

We have selected two related paper with our work. The first paper is(7) : Vulnerability Assessment on the Parental Control Mobile Applications' Security: Status based on the OWASP Security Requirements. In this paper, they have performed SAST(Static Application Security Testing)using Checkmarx, to identify flaws and vulnerabilities. First, they have reviewed the advantages and drawbacks of SAST and also presented a survey on OWASP risk rating methodology.Then they mapped the co occurrence of high profile vulnerabilities types from both OWASP Top 10 and CWE/SANS 25. By performing SAST, they have detected most vulnerable files and again scanned without those files.

The second paper(8) is Vulnerability Assessment on the Parental Control Mobile Applications' Security: Status based on the OWASP Security Requirements . Quixxi security is applied to check vulnerabilities of parental apps and some common vulnerabilities are found among 3 apps and average of the fail rate in every severity level is calculated .

4 Data Collection

This segment discusses a list of vulnerability scanning tools and targeted applications, environment setup, and data gathering strategies for discovering vulnerabilities in Bangladeshi mobile financial service applications. These vulnerabilities result primarily from deviations from standard coding practices and security configuration issues. Occasionally, smartphone hardware difficulties are also to blame. This study analyzes the vulnerabilities in MFS applications using a variety of measurements and approaches.

4.1 Vulnerability Scanner Selection

Using a high-quality vulnerability scanner can assist detect weak areas and reduce the attack surface that an attacker may exploit. There are several commercial and open-source android application scanning tools available which provide static and dynamic analysis features along with detailed reports. Since the list is long, three tools are selected in this research which are quite remarkable in the industry. MobSF (Mobile security Framework), Immuniweb are open-source and QUIXXI is a paid scanner. A short outline of these tools is provided below.

- **MobSF (Mobile Security Framework) :** Open source, a complete package for mobile penetration testing which follows the OWASP MSTG (Mobile Security Testing Guideline) and capable of operating dynamic and static testing of an android or iOS application effectively. While analyzing statically, it converts APK/IPA files to native code and does a very extended test on code level. Additionally, in dynamic testing applications are installed in a real device or simulator that is integrated with mobSF. Different frida scripts can be injected while performing dynamic tests. (9)

- **QUIXXI :** A sophisticated paid tool that is inspired by OWASP which checks the integrity and detects threats with CVE reference for android platform. The key advantage of utilizing this tool is that it shows the potential impact of the identified issues and provides proposed remedies. (10)

- **Immuniweb-A SaaS-based (Software as a Service) :** application security solution developed based on machine learning and artificial intelligence that checks vulnerability against different business logic testing, SANS Top 25 standards, PCI DSS OWASP coverage. (11)

4.2 MFS Application Crawling

In Bangladesh, several prominent digital mobile financial service providers operate under the supervision of various banks and the Bangladesh Post office. These enterprises all offer Android and iOS applications to their customers for daily uses. Five Android MFS applications have been chosen for vulnerability research in this paper. Due to confidentiality concerns, the names of the applications are kept anonymous.

5 Experimental Workflow

In figure 1, App1, App2, … App5 denotes the selected five MFS applications for this research. APK files of these applications are uploaded to the scanning tool as an input and after analysis it generates vulnerability reports. Since three individual reports are generated from selected tools for each application, an overall report analysis has been performed to generate the final report which shows the application wise vulnerability breakdown. Vulnerability assessment summary and best performance tool - these two things will come up in out research by following this workflow.

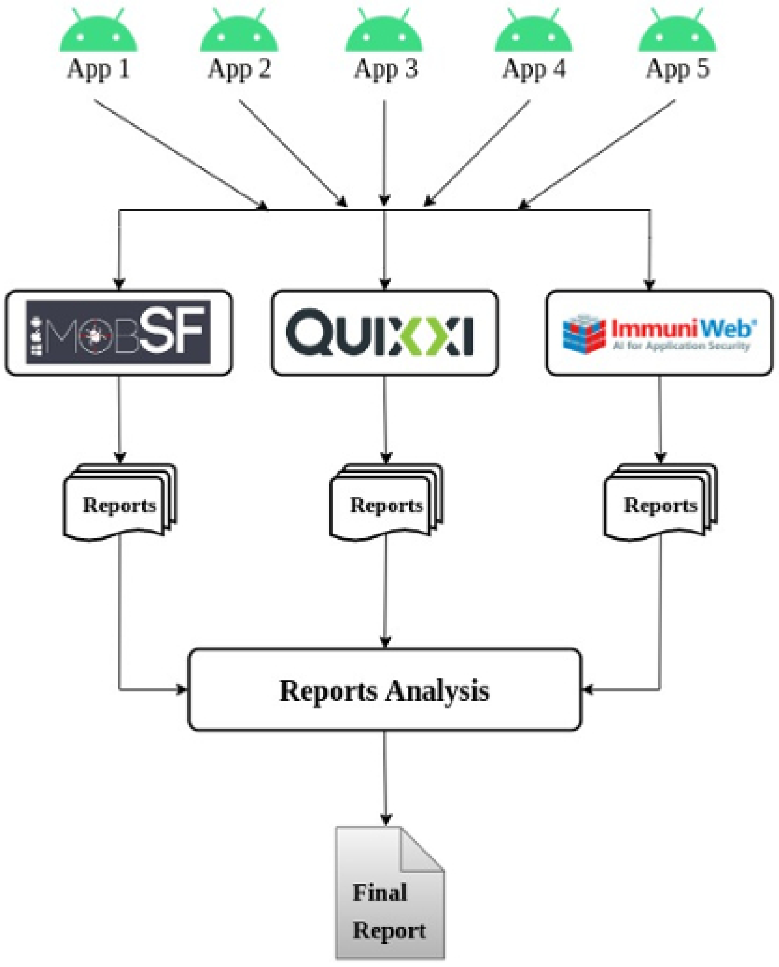

Figure 1: Experimental Workflow

6 Experimental Findings Discussion

In this section we analyzed scanned results and mapped on the basis of different risk levels. Some identified vulnerabilities must be resolved immediately because they allow attackers to compromise or damage the application, while others have a lower priority. In its scans and reports, scanner categorizes vulnerabilities based on their risk rankings to aid developers in deciding which flaws should be patched first. In software security risks are classified into 4 categories named High, Medium, Low and Information.

6.1 Risk Level Classification

The risks are classified based on their CVSS score (12)

- High Risk : The vulnerabilities identified as High Severity may grant unauthorized access to application resources and data.
- Medium Risk : Medium Severity problems are typically caused by faults and shortcomings in the application configuration.
- Low Risk : Low Severity concerns include information leakage, installation errors, and the absence of certain security measures.

In addition to the risk alerts listed above, a scanner also uncovers informational messages that are regarded to be of interest. Understanding the application's technological stack and its dependencies make it easier for attackers to design attack strategy.

6.2 Vulnerability Detection

Results of detected vulnerabilities by selected tools are plotted statistically in pae chart and the numbers at tables here. Numerous failed evaluations are observed by Vulnerability scanners for all applications and detect severity level, related risk, and security weaknesses. Most vulnerability scanners either search for existing vulnerabilities or look for common flaws to discover new ones. In the report of the scans, an overall risk score is calculated.

Since each scanner's testing process, debugging script, and database of common vulnerabilities are different. So a different outcome for a targeted application from multiple scanning tools is not surprising. The disadvantages of relying only on automation testing is either it misses major issues or provides false positives results sometimes. However, the false positive results have been excluded for getting accurate results by analyzing each application's permission list, manifest, code, shared library, APKID, Browsable Activities and tracker manually in this research.

Tables

App	High	Medium	Low	Info
App1	7	5	5	4
App2	12	7	10	11
App3	8	6	6	6
App4	9	4	5	5
App5	7	4	3	5
Total	43	25	29	31

Table 1: Vulnerability Detected By MobSF Scanner

App	High	Medium	Low	Info
App1	6	6	2	3
App2	10	8	3	3
App3	8	3	1	1
App4	8	3	1	1
App5	5	4	1	2
Total	37	27	8	11

Table 2: Vulnerability Detected By QUIXXI Scanner

App	High	Medium	Low	Info
App1	4	3	2	4
App2	6	4	7	11
App3	3	3	4	6
App4	3	2	3	4
App5	4	2	3	5
Total	21	14	19	30

Table 3: Vulnerability Detected By Immuniweb Scanner

List of Figures

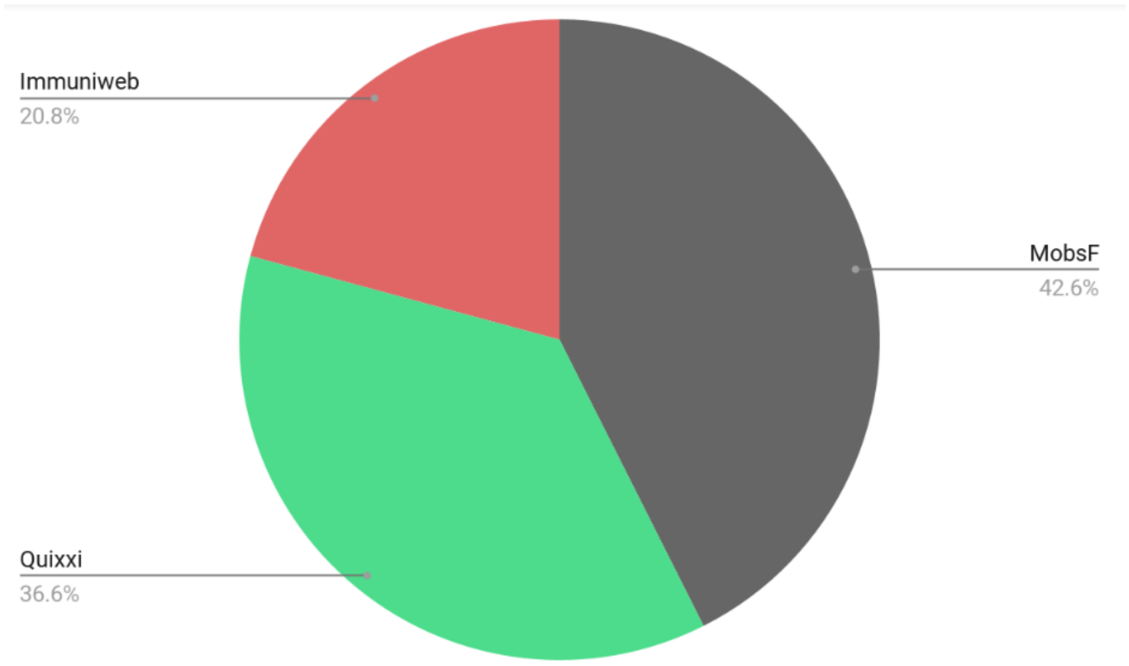

Figure 2: High Vulnerability detected by Scanners

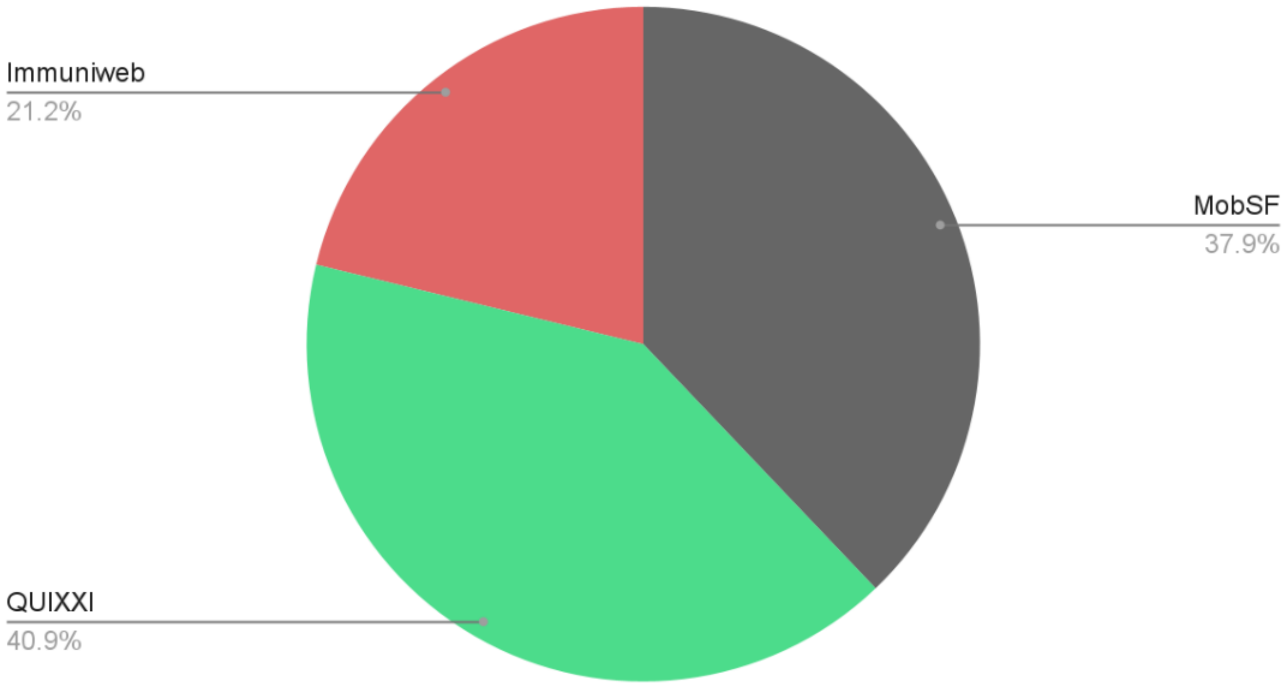

Figure 3: Medium Vulnerability detected by Scanners

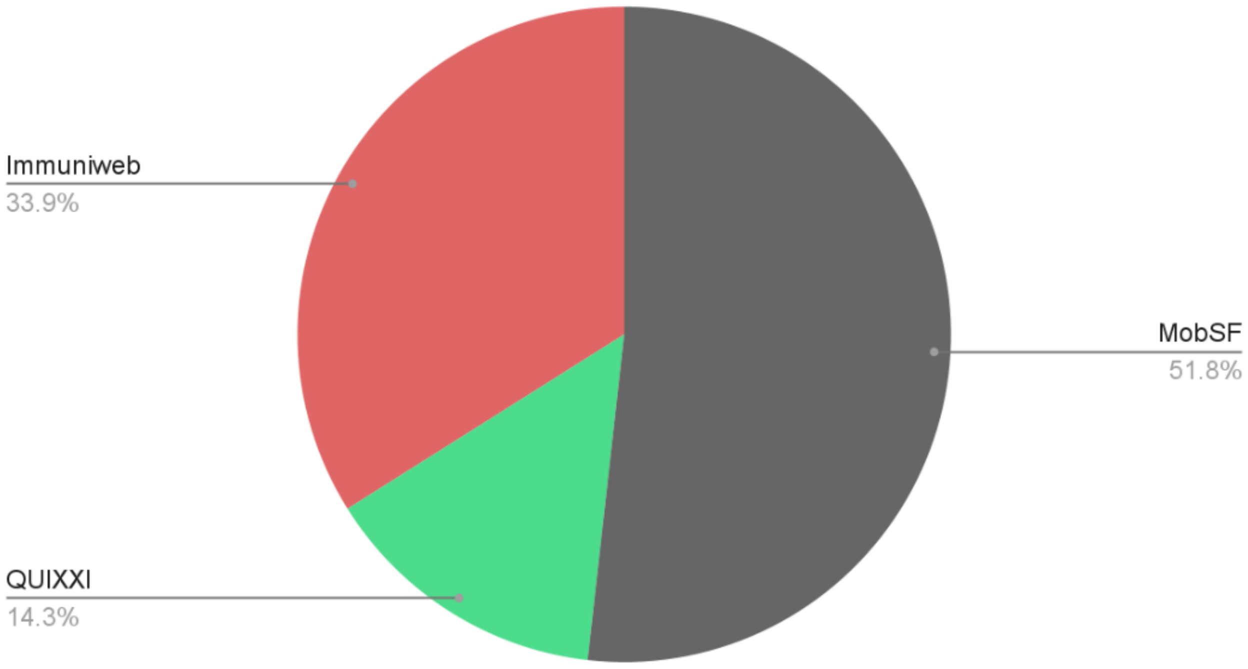

Figure 4: Low Vulnerability detected by Scanners

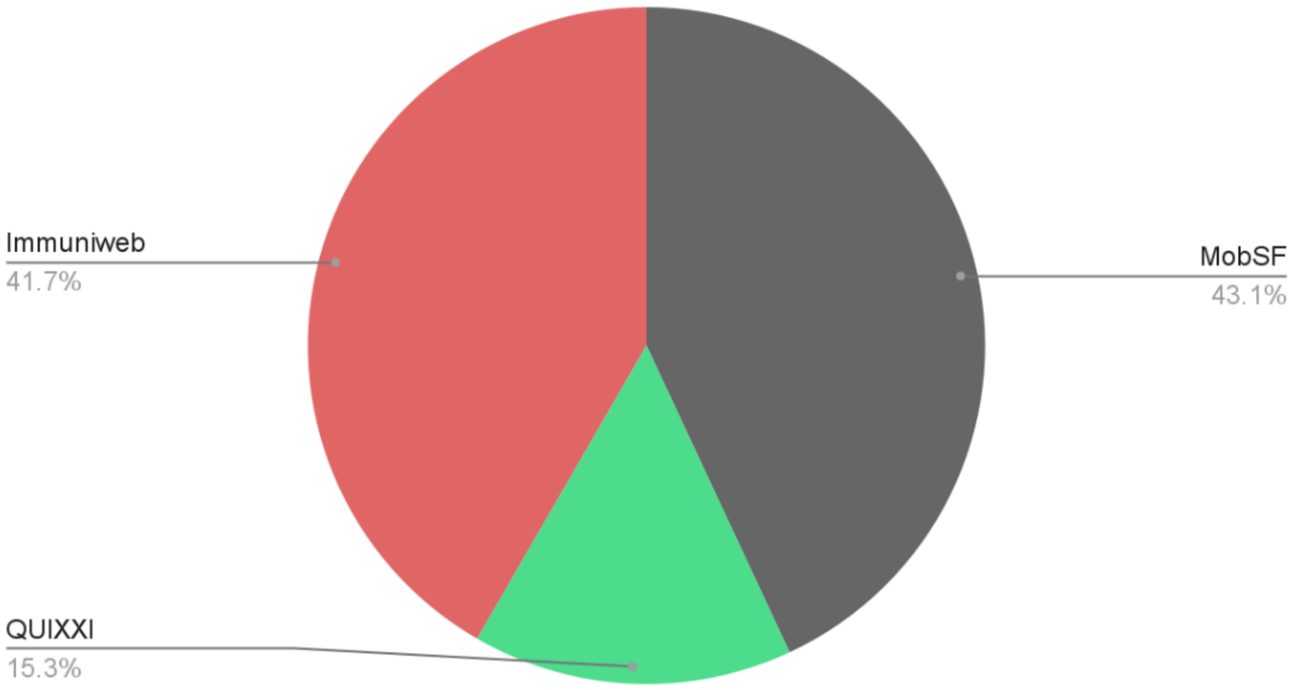

Figure 5: Informational Vulnerability detected by Scanners

7 Discussion

In this paper the most dangerous web application flaws evaluated , such as Enabled Clear text communication, SSL certificate missing, unsafe file deletion, execution of raw SQL query, improper permission setting and also medium risk like Weak Random Number Generator, Weak Java Hashcode implementation, Insecure cryptographic protocols are most prevalent in all applications. Low level of vulnerabilities like poor string security, missing protection of tapjacking, and insecure temporary file creation are also found.

According to the common vulnerabilities table, a hypothesis has been established in figure to understand the security level of mobile financial applications in Bangladesh. It shows that 52.6% of vulnerabilities are categorized as High. The percentage of medium and low risks are around 21.4%. And more than 5% are leveled as low.

8 Conclusion

In this study, the overall vulnerability assessment of mobile financial service applications in respect of Bangladesh were identified by MobSF, QUIXXI and Immuniweb following mobile application security standards provided by OWASP. According to the survey, the alarming rate of high-risk vulnerabilities prevalence in digital transaction applications is worrying. Since the money is converted to digital currency and everything is logged on the system, it is possible to follow every transaction. Therefore, if an attacker has a foothold and is able to elevate privileges, the system cloud is easily compromised. Consequently, consumers will incur monetary losses.So these application development company should pay more and more attention to mitigate this risk.

9 References

References

[1] N. Bilton and B. Stelter, "Sony says playstation hacker got personal data," *The New York Times*, 2011.

[2] M. Jewell, "Data theft believed to be biggest hack," *The Washington Post*, 2007.

[3] B. Acohido, "Hackers breach heartland payment credit card system," *Retrieved October*, vol. 18, no. 2011, pp. 2009–0, 2009.

[4] M. A. Rahman, M. Amjad, B. Ahmed, and M. S. Siddik, "Analyzing web application vulnerabilities: an empirical study on e-commerce sector in bangladesh," pp. 1–6, 2020.

[5] D. Gillman, Y. Lin, B. Maggs, and R. K. Sitaraman, "Protecting websites from attack with secure delivery networks," *Computer*, vol. 48, no. 4, pp. 26–34, 2015.

[6] Y. Cifuentes, L. Beltrán, and L. Ramírez, "Analysis of security vulnerabilities for mobile health applications," *International Journal of Health and Medical Engineering*, vol. 9, no. 9, pp. 1067–1072, 2015.

[7] J. Li, "Vulnerabilities mapping based on owasp-sans: a survey for static application security testing (sast)," *Annals of Emerging Technologies in Computing (AETiC), Print ISSN*, pp. 2516–0281, 2020.

[8] E. B. Blancaflor, G. A. J. Anson, A. M. V. Encinas, *et al.*, "A vulnerability assessment on the parental control mobile applications' security: Status based on the owasp security requirements,"

[9] G. LaMalva and S. Schmeelk, "Mobsf: Mobile health care android applications through the lens of open source static analysis," in *2020 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pp. 1–4, IEEE, 2020.

[10] M. Patil and D. Pramod, "Andrev: Reverse engineering tool to extract permissions of android mobile apps for analysis," in *Computer Networks and Inventive Communication Technologies*, pp. 1199–1207, Springer, 2021.

[11] M. Naja, A. Shafana, and A. Musfira, "Automated software testing and tool selection: case study based on security testing of popular e-commerce applications in malaysia," 2021.

[12] Z. Han, X. Li, Z. Xing, H. Liu, and Z. Feng, "Learning to predict severity of software vulnerability using only vulnerability description," pp. 125–136, 2017.