Jurnal Sistim Informasi dan Teknologi

https://jsisfotek.org/index.php

2023 Vol. 5 No. 2 Hal: 97-102 e-ISSN: 2686-3154

The Application of Mobile Security Framework (MOBSF) and Mobile Application Security Testing Guide to Ensure the Security in Mobile Commerce Applications

Chairul Anwar[1]✉, Chevy Herli Sumerli A[2], Sultan Hady[3], Novi Rahayu[4], Kraugusteeliana Kraugusteeliana[5]

[1]Politeknik Jakarta Internasional
[2]Universitas Pasundan
[3]Universitas Dayanu Ikhsanuddin
[4]STIA Bengkulu
[5]Universitas Pembangunan Nasional Veteran Jakarta

chairul.anwar@jihs.ac.id

Abstract

The use of mobile devices is one aspect of information technology that is now expanding quickly. In recent years, the use of mobile applications has increased in various areas of Indonesian society. However, cybercrimes such as data leaks are also increasing in Indonesia. One of them is the case of data theft in mobile commerce applications in Indonesia, where as many as more than 90 million user records were illegally traded by hackers on dark web sites. The mobile commerce application also stores sensitive user data for use in its business processes, such as email, passwords, addresses, telephone numbers, and account numbers. The goal of this study is to evaluate and identify security vulnerabilities or loopholes that could harm providers and users of the Android-based mobile commerce application using the Mobile Security Framework (MOBSF) and the OWASP Mobile Application Security Testing Guide (MASTG). This research was carried out in five stages: preparation, data collection, mapping the application (mapping vulnerabilities), exploitation, and reporting. The results of the study found that the mobile commerce application has a security gap issue in the data storage range in the parameter (MSTG-STORAGE-5) and in the authentication architecture range in the parameter (MSTG-AUTH-5 and MSTG-AUTH-6).

Keywords: Mobile Devices, Mobile Applications, MOBSF, Security.

1. Introduction

The progress of information technology is currently advancing at a rapid pace, and one aspect of this is the utilization of mobile devices. In recent times, the usage of mobile applications has surged in various sectors of Indonesian society. According to the Digital 2022: Indonesia report, the number of internet users in Indonesia has reached 204.7 million, with a penetration rate of 73.7 percent. This indicates an increase of 1 percent compared to the previous year (DataReportal, 2022). However, the rise in the number of internet users in Indonesia does not necessarily correspond to a high level of cybersecurity. Cybersecurity is a critical issue in Indonesia, with frequent occurrences of data breaches. As per the National Cyber Security Index (NCSI) report of 2022, Indonesia has a cyber security index score of 38.96 out of 100, which is a measure of the country's preparedness to prevent cyber threats and manage cybersecurity incidents. This places Indonesia at rank 83 out of 160 countries in the world, rank 6 in ASEAN, and is ranked 3rd lowest compared to the G20 countries (National Cyber Security Index, 2022). In the Vulnerabilities and Threats in Mobile Applications 2019 report on the ptsecurity.com website, it states that as many as 43% of Android applications have security vulnerabilities with high risk. Most of it is due to weaknesses in security mechanisms since application creation and also to granting privileges that are inappropriately exploited by attackers [1].

The increase in cybercrimes such as data leaks has been a succession in Indonesia. One of them is the case of data theft in mobile commerce applications in Indonesia. According to previous research, as many as 91 million data points from Tokopedia application users were traded illegally by hackers on the Dark Web site. User data that may be exposed is personal data such as email, name, address, date of birth, and telephone number. The perpetrator sold the data for $5,000, or around IDR 74 million. This is inseparable from carelessness and weak online technology security systems that have had a very bad impact on users and platform owners. Application providers, or in that case, mobile commerce applications in general, often ask for permission regarding personal data from their users. This data is used to support the application's business processes. Primary user identities such as email,

Diterima: 10-06-23 | Revisi: 15-06-23 | Diterbitkan: 20-06-23 | doi: 10.37034/jsisfotek.v5i1.231

address, and telephone number are unique and vulnerable to abuse. According to previous research, Indonesia has cases of account leaks that have increased by more than 100% in 2022 compared to 2021. Indonesia is in the first tier of countries where there have been increased cases of account leaks from 2021 to 2022 [2].

The threat of cyberattacks is inseparable from the carelessness and weakness of the online technology security system, which has a very bad impact on application users and platform owners. According to previous research, mobile commerce is an online channel that can be reached by someone through a mobile device and is used by sellers and customers to make transactions online. The increase in the number of electronic commerce users in Indonesia is predicted to reach 189.6 million in 2024. Therefore, it is important for all parties involved in online transactions to consider threats to information security and be equipped with relevant and up-to-date knowledge to minimize the risk. At the end of 2018, Jakmall, which is one of the largest e-commerce companies in Indonesia, sought to develop its company by releasing an Android-based mobile commerce application. To date, users of Android-based e-commerce applications have reached more than 100,000. E-commerce is a mobile commerce application that offers different things in the midst of intense e-commerce competition. E-commerce provides a platform that opens opportunities for resellers to pick up goods from stalls on this platform. Therefore, e-commerce conducts strict selection for users in order to provide products at low prices. Based on observations in this study, e-commerce collects user information for processing and expediting the process of using the application [3].

There exist various frameworks that aid in examining the security of Android apps. Two such frameworks are the Mobile Security Framework and the Open Web Application Security Project Mobile Application Security Testing Guide. These frameworks furnish unambiguous security analysis outcomes, which can serve as a basis for security enhancements for Android app developers. The Mobile Security Framework (MobSF) is an automated testing framework for open-source mobile apps (Android, iOS, and Windows) [4]. This framework performs malware analysis and penetration testing and can identify vulnerabilities that attackers can exploit in mobile apps. In prior research, the MOBSF framework was employed for static analysis, which yielded a true positive of up to 97%. The accuracy of the analysis results was verified, indicating that the true positive value remained unchanged [5]. The MobSF framework encompasses various frameworks, including the Open Web Application Security Project (OWASP) Mobile Application Security framework. The OWASP MASTG framework assists MobSF in analyzing and categorizing mobile security risks based on its guidelines. In 2018, OWASP created a guide on testing the security of mobile applications packaged in OWASP MASTG. OWASP MASTG is a security standard used in mobile applications accompanied by a comprehensive testing guide that includes processes, techniques, tools, and case studies related to carrying out mobile application security tests. According to previous research, OWASP MASTG can find more Android application vulnerabilities than the results of analysis on the AndroBugs framework [6].

A study carried out a static analysis of the security of Android-based mobile commerce applications using MobSF and found that there were security holes in five mobile commerce applications with relatively the same level of security [17]. In another study, creating a security testing system that functions to search for code to be analyzed by implementing the OWASP Mobile Application Security Testing Guide (MASTG) Static Analysis technique The results of the system can be used as material for further evaluation based on the OWASP MASTG guidelines [7][16]. For testing Android mobile security analysis, OWASP recommends a phase arrangement that is generally used by penetration testers, including the following: Preparation: This stage establishes the parameters of security testing, encompassing the identification of relevant security measures, testing objectives, entities, and sensitive data. Intelligent Gathering: In this stage, the environmental context and application architecture are analyzed to gain a comprehensive contextual understanding [8][15]. Application Mapping: This stage involves mapping the application based on information gathered in the previous stage, which can be done either through automated scanning or manual exploration. Mapping provides a thorough understanding of the application, its entry points, the data it stores, and key potential vulnerabilities [9]. These vulnerabilities are then ranked based on the damage their exploitation could cause, allowing security testers to prioritize them [10][11][12]. Exploitation: In this stage, security testers attempt to infiltrate the application by exploiting the vulnerabilities identified in the previous stage. This stage is crucial for verifying the veracity of the vulnerability [13][14]. Reporting: In this stage, security testers report identified vulnerabilities to relevant parties. This includes a detailed exploitation process, classifying the types of vulnerabilities, documenting the risk if an attacker can compromise the target, and deciphering which data could be accessed by testers unauthorize [18].

2. Methods

To accomplish this study, data and information are required as resources to substantiate the accuracy of the material depiction and discourse. Thus, prior to conducting this research, the investigators performed preliminary research to acquire pertinent data and information. The methodologies used by the researchers for data collection are as follows: literary review and observation. For this examination, data analysis techniques were employed, utilizing static and dynamic analysis based on the OWASP Mobile Security Testing Guide (MASTG) with the aid of tools and the MOBSF framework. Static analysis is carried out by examining the results of the source code and the

results of application reverse engineering without having to interact with the application being tested. Meanwhile, dynamic analysis is carried out by examining the application directly when it is run on an Android device or emulator. In this study, 5 stages of testing were carried out based on the OWASP Mobile Security Testing Guide: preparation, intelligence gathering, mapping the application, exploitation, and reporting (MASTG, 2022).

3. Results and Discussion

Based on the results of automatic scanning with MOBSF, it provides information that the APK file that is the target for analysis is version 1.3.83 with version code 35. Android-based e-commerce can run with a minimum of API (Application Programming Interface) level 21 or Android 5.0 and with a target SDK (Software Development Kit) on API level 32 or Android 12. The authenticity of e-commerce APK files can be checked using the SHA256 cryptographic algorithm (Secure Hasing Algorithm) with MD5 (Message Digest) and SHA1 hashes. One of the results of the analysis carried out by MOBSF is to provide information about Android components, including activity, service, broadcast receiver, and content provider. In the Android-based mobile commerce application, there are 76 activities, 16 services, 19 receivers, and 5 providers. The static analysis from MOBSF found the issue of exposed components, including 7 exported activities out of 76 activities, 2 exported services out of 16 services, and 7 exported receivers out of 19 receivers. Based on the results of the static analysis from MOBSF, there are several security issues found in the code analysis. This stage performs the mapping or categorization of vulnerabilities found from the results of static analysis based on the OWASP MASVS (Mobile Application Security Verification Standard) within the scope of data storage and authentication architectures. In the mobile commerce application, a vulnerability was found in the MSTG-STORAGE-2 parameter, which is included in the scope of OWASP MASTG Data Storage in the Testing Local Storage for Sensitive Data section. MOBSF does not find security issues in other parameters, so the authors carry out further tests to validate the findings from MOBSF. In this exploitation stage, testing of security issue cases is carried out against the findings from the results of code analysis that has been carried out with MOBSF tools. The purpose of this stage is to verify the test cases from the OWASP Mobile Application Security Testing Guide (MASTG) related to the assessment of the OWASP Mobile Application Security Verification Standard (MASVS) parameters generated from the automated testing of the MOBSF code analysis tools. In the results of the code analysis that was carried out previously, security issues were found with the MSTG-STORAGE-2 parameter with a medium risk or warning level. These parameters can be tested with test cases from OWASP MASTG Testing Local Storage for Sensitive Data. These test cases focus on identifying potentially sensitive data stored by applications and verifying that it is stored securely. This is intended so that sensitive data stored is not leaked due to malicious application attacks (OWASP MASVS, 2022). Android devices have several information storage media, such as internal storage, external storage, and cloud storage. In this test case, it is carried out to identify the source of the type of storage in the application. In the AndroidManifest.xml file, you can see that this application is asking for permission to read and write to the external storage media. Under these conditions, it allows external storage to store cache. Anything stored in this type of storage can be accessed by other applications on the user's device, and data is retained even after these applications are removed from the user's device. However, after further testing the shared preferences (/data/data/com.jakmall/shared_prefs) to look for XML files that hold sensitive data, The result is that there is no sensitive user data stored in the application's shared preferences.

In addition, the authors conducted tests using the adb tool to check permissions on /data/data/com.jakmall to ensure that only registered users when the application was installed could read, write, and execute. The screenshot results from the adb tools state that the Jakmall mobile commerce application meets the standards for this test parameter. Only the u0_a45 user that has been installed can read, write, and execute. After testing the security issues on this parameter, the results do not match the results of the static analysis test carried out automatically with the MOBSF tool, which can be called a false positive. False positives are very likely to occur when performing tests automatically with existing tools. Therefore, it is highly recommended to carry out further testing to validate the findings from automated testing. So in this parameter, it can be said that the application uses sensitive data or information storage. Any publicly accessible data store, such as login forms or any process, can modify that data. This can be used by irresponsible parties. Then it is necessary to apply input validation when the data is read back. In this test, an input validation analysis of the login mechanism was carried out based on the source code in the mobile commerce application's AuthHolder file obtained from the reverse engineering results.

Input validation analysis was carried out on the login mechanism in the mobile commerce application source code. Where in the AuthHolder file used in the LoginActivity mechanism there is a getSharedPreference, it can be said that the input validation mechanism has been carried out in shared preferences. So that it can be stated that the input validation carried out in shared preference does not generate sensitive data or information. Logging is useful for application developers in tracking crashes, application usage errors, and viewing application usage statistics. Log file storage can be done on internal storage. But you have to be careful when the application allows you to write logs to reveal sensitive user data that can be misused by irresponsible parties. Testing in these cases focuses on identifying any sensitive application data in the application logs. The examination is performed through static

scrutiny of the origin code, which archives correlated code like Log.d, Log.e, Log.i, Log.v, Log.w, and Log.wtf. It is generally advised to eliminate logging declarations from the production releases to prevent the potential exposure of confidential application data, except when it is deemed essential for the application or explicitly recognized as secure, such as after a security evaluation. So then the author will conduct a test with dynamic analysis of log activity on the device.

Testing is done using the adb tool by giving the command "adb logcat" through the command prompt. The process is carried out in the form of login features, changing user data, and checking out. Based on this test, no sensitive data was found in the activity log of the mobile commerce application. The test in this case is an attempt to intercept HTTP (Hypertext Transfer Protocol) traffic between the client and server when logging in to the mobile commerce application. Testing was carried out using the Burpsuite tool. The purpose of this test is to find out what information is obtained during the communication process between the client and server. Testing of the application login process is carried out by entering data in the form of a user username, email, and password. In this test, the username and password used are the author's. The system sends data to the server in the form of input from the user without going through the process of encrypting data against usernames and passwords, thus allowing intercepted HTTP requests that easily change data to learn how applications behave when performing different actions. So that, with these conditions, it can be categorized as HTTP traffic for mobile commerce applications in unsafe conditions.

This test case identifies whether there is sensitive data exposed through the keyboard cache. When the user types in the input field, the application can suggest data; this means that the application allows storing data in the keyboard cache. Nonetheless, the keyboard buffer might exhibit confidential information when the user selects an input field that employs this sort of data. Analysis is conducted on the input column that contains sensitive data. If the column suggests previously entered data, it indicates that the app may store the keyboard buffer. Nevertheless, the mobile commerce app demonstrates that it does not store the keyboard buffer with regard to sensitive data, such as usernames and passwords. The act of entering sensitive data, like registering an account or logging in, is a critical aspect of operating numerous apps. This data can take the form of information, such as user passwords. If the app fails to conceal it properly while it is being typed, data could be exposed. To prevent disclosure and minimize risks such as shoulder surfing (direct observation) cybercrime, it is essential to verify that no sensitive data is exposed through the user interface. In this instance, sensitive data, user passwords, must be appropriately disguised, usually by displaying asterisks or dots instead of plain text.

Based on the test in this case, it is carried out by checking the account registration form. The entered text is hidden on the program interface by a dot symbol that has been placed over the password column. When data is utilized for authentication, password strength is an important consideration. The password policy outlines the conditions that the user must meet. The password policy usually determines the length of the password, the complexity of the password, and the topology of the password. A "strong" password policy makes cracking manual or automated passwords difficult or impossible. In this test case, it is done by testing the password policy when registering an account. Testing was carried out using Burpsuite tools by looking at requests and responses during the process of registering a new account. The data used in this test is the new user data that the author created. tested by creating a new account. The author creates a new account by entering the password "panthers". This password is weak because all characters only use lowercase letters (OWASP, 2022). The application gives the response "200 OK," which means the use of a weak password in the application is allowed. This proves that the mobile commerce application does not have a password policy for new users. Weak passwords are easier to attack and can be abused. The author will test this mechanism by carrying out a brute-force attack technique. This technique is done by trial and error in breaking the password. Mobile commerce applications can be targeted with brute-force attacks on the login feature. The author performs a brute force attack technique on the user's password by trying to log in 1000 (one thousand) times with the user's username, and the result is that only one password matches that username. Determining Whether Sensitive Data is Sent to Third Parties (MSTG-STORAGE-4), one of the earlier parameters in the Data Storage scope, is linked to this as well, making the login function vulnerable to brute force attack strategies.

It was discovered that two test parameters that were verified as security issue vulnerabilities, namely determining whether sensitive data is sent to third parties and testing best practices for passwords, were security issue vulnerabilities based on the exploitation stage that was carried out on seven parameters in the scope of data storage and authentication architectures. A security flaw is discovered when the program sends user input to the server without encrypting the login and password, as was the case when the Determining Whether Sensitive Data is Sent to Third Parties setting was tested. So that sensitive data is exposed to HTTP (Protocol Transfer Interface) traffic. When the hacker can intercept traffic from the client, he can either explore and understand what he is doing or rewrite it to get another response from the server. Then, in testing the Testing Best Practices for Password parameter, a security issue was found where the application did not apply a password policy to the mechanism for creating a new account or registering an account. The use of weak passwords is a serious vulnerability problem because it makes it easier for hackers to take control of user accounts. By not applying the blocking mechanism or

account locking against incorrect login requests many times, it gives an attacker the opportunity to freely guess an account's password with a brute-force attack technique. After that, recommendations for improvements are made based on the vulnerabilities found to improve the security of mobile commerce applications. The security recommendations provided are based on the OWASP MASTG guidelines, namely: Implement data encryption techniques when communicating between clients and servers, especially in the login mechanism. Enforce password policies when users register accounts, so users are required to use strong passwords. Limiting the number of logins attempts to no more than five times will make brute force techniques difficult. Implement an account lock system (temporary or permanent) if a user logs in more than 15 times. Added the Captcha feature to the login form to distinguish between human and robot logins. Maximizing the login mechanism by implementing two-factor authentication with validation on other devices so as to prevent brute-force attacks.

4. Conclusion

Based on the results of mobile commerce application security analysis using the MOBSF and OWASP Mobile Application Security Testing Guide (MASTG) methods in the scope of Data Storage and Authentication Architecture at the Mapping the Application stage, a security vulnerability issue was found in the Testing Local Storage for Sensitive Data parameter. However, after validation testing was carried out at the exploitation stage, these parameters did not identify any security vulnerability issues. This may occur as a result of false positives generated by the MOBSF tools during static analysis. A vulnerability in the parameter determining whether sensitive data is sent to third parties was discovered during the exploitation stage based on the findings of the validation test with the OWASP MASTG framework, specifically sensitive data that was not encrypted and exposed in communication between the client and server. Apart from that, other security issues were also found in the testing best practices for passwords parameter, where hackers can freely guess passwords by carrying out brute force attacks because existing applications do not implement a strong password creation policy. The vulnerabilities of the two parameters found can enable account theft where sensitive data or information belongs to users and can be misused by irresponsible parties. can perform tests with other scopes of the OWASP MASTG framework, such as cryptographic APIs, network communication, platform APIs, code quality and build settings, and anti-reversing defenses. using an iOS-based mobile commerce application by implementing the same framework. This can be done with applications in the health, education, banking, or government sectors, which also use sensitive user data and information.

References

[1] Kurniawan, C., & Trianto, N. (2021). Security Assessment on the XYZ Android Mobile Application With Reference to the OWASP Mobile Top Ten 2016 Vulnerabilities. *Info Kripto*, 15(1), 11–18. https://doi.org/10.56706/ik.v15i1.2

[2] DataReportal. (2022). *DataReportal – Global Digital Insights*. Datareportal. https://datareportal.com/reports/digital-2022-indonesia

[3] Darmawan, J., Wijaya, A. H., Hakim, L., & Tannady, H. (2021, February). Comparing Freeman Chain Code 4 Adjacency Algorithm and LZMA Algorithm in Binary Image Compression. In *Journal of Physics: Conference Series* (Vol. 1783, No. 1, p. 012045). IOP Publishing.

[4] Fathur, M. (2020). Tokopedia's Responsibility for Data Leakage. *National Conference on Law Studies (NCOLS)*, 2(1), 43–60.

[5] Hanifurohman, C., & Hutagalung, D. D. (2020). Static Analysis Using the Mobile Security Framework for Testing the Security of Android-Based E-Commerce Mobile Applications. *Sebatik*, 24(1), 22–28. https://doi.org/10.46984/sebatik.v24i1.920

[6] Tannady, H., Andry, J. F., Suyoto, Y. T., & Herlian, A. (2020). Business Architecture of Public Guest Service for University Using TOGAF ADM Framework. *Technology Reports of Kansai University*, *62*(5), 2421-2428.

[7] Hermawan, K., Iskandar, A. A., & Hartono, R. N. (2011). Development of ECG signal interpretation software on Android 2.2. *2011 2nd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*, 259–264.

[8] Indarta, Y., Ranuharja, F., Ashari, I. F., Sihotang, J. I., Simarmata, J., Harmayani, H., Algifari, M. H., Muslihi, M. T., Mahmudi, A. A., Fatkhudin, A., & others. (2022). *Cyber Security: Challenges in the Era of the Industrial Revolution 4.0*. Yayasan Kita Menulis.

[9] Meisarah, F., Octiva, C. S., Sucipto, P. A., Satyaninrum, I. R., & Bakri, A. A. (2023). Improving Student Text Writing Ability by Utilizing the Use of Augmented Reality Feature. *Jurnal Sistim Informasi dan Teknologi, 5*(1), 129-134.

[10] Negi, C., Mishra, P., Chaudhary, P., & Vardhan, H. (2021). A Review and Case Study on Android Malware: Threat Model, Attacks, Techniques and Tools. *Journal of Cyber Security and Mobility*, 10(1), 231–260. https://doi.org/10.13052/jcsm2245-1439.1018

[11] Kumar, G. S., Priyadarshini, R., Parmenas, N. H., Tannady, H., Rabbi, F., & Andiyan, A. (2022, November). Design of Optimal Service Scheduling based Task Allocation for Improving CRM in Cloud Computing. In *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 438-445). IEEE.

[12] Parulian, S., Pratiwi, D. A., & Cahya Yustina, M. (2021). Cyber Attack Threats and Solutions in Indonesia. *Telecommunications, Networks, Electronics, and Computer Technologies (TELNECT)*, 1(2), 85–92. http://ejournal.upi.edu/index.php/TELNECT/

[13] Putri, A. S., & Zakaria, R. (2020). Mapping analysis of the largest e-commerce in Indonesia based on the Digital economy power model. *Seminar Dan Konferensi Nasional IDEC*, 1–14.
[14] Gunawan, F. E., Andry, J. F., Tannady, H., & Sebastian, B. (2020). Evaluation and measurement of automobile service and maintenance company performance using cobit framework and balanced scorecard. *Evaluation*, *62*(07).
[15] Rohmah, R. N. (2022). Efforts to Build Cyber Security Awareness among E-commerce Consumers in Indonesia. *Cendekia Niaga: Journal of Trade Development and Studies*, 6(1), 1–11.
[16] Anwar, C., & Riyanto, J. (2019). Perancangan Sistem Informasi Human Resources Development Pada PT. Semacom Integrated. *International Journal of Education, Science, Technology, and Engineering*, *2*(1), 19-38.
[17] Widyarto, E. Y., & Anwar, C. (2022). Build Social Networks-Based Audio Engineering (Design And Build An Audio-Based Social Network). *Eduvest-Journal of Universal Studies*, *2*(1), 48-54.
[18] Yumnun, L., Kusyanti, A., & Kartikasari, D. (2020). Implementation of the OWASP Mobile Security Testing Guide (MSTG) for Security Testing on Android-Based Applications. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(11), 10579–10585.