Security Analysis of the Lombok Tourism Android Application Using Penetration Testing (Pentesting) Methods Based on the OWASP Mobile Top 10-2024 Framework

Ida Bagus Adi Surya Kemenuh, Raphael Bianco Huwae, Andy Hidayat Jatmika
Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: bagusadi.bgcbg@gmail.com, *: [*raphael.bianco.huwae, andy]@unram.ac.id

Abstract **Android has become the most widely used operating system for mobile devices, playing an important role in supporting the tourism sector. Along with the growth of tourism in Indonesia, the need for quick and easy access to information for travel planning is increasing. However, the issue of user data security on Android applications is a major concern. This research aims to conduct penetration testing on tourism applications in Lombok, the application used as a test example was selected from one of the 17 Lombok tourism applications that had the highest number of vulnerabilities compared to other Lombok tourism applications. This research is focused on identifying vulnerabilities based on the OWASP Top 10 Mobile 2024 framework. The method used in this research is static analysis using the Mobile Security Framework (MobSF), focusing on the two main vulnerabilities identified: Insecure Data Storage and Insufficient Cryptography. These two vulnerabilities are the only ones listed in the OWASP Top 10 Mobile 2024 for the applications that have been analyzed. The penetration testing results show that, despite the risks associated with Insecure Data Storage, no sensitive user data was found in the Asap Lombok app database. In addition, this application uses outdated encryption (CBC with PKCS7 padding), which is potentially vulnerable to oracle padding attacks. However, this vulnerability was only found in the appearance of the application layout and did not have a significant impact on user data security.**

Key words**: Tourism, Android, OWASP Mobile top 10 2024, MobFS, Penetration Testings**

I. INTRODUCTION

Android has become the most widely used operating system globally, particularly for mobile devices**.** Its flexibility and openness allow developers and users to modify, develop, and utilize devices according to their needs. Android has become a hot topic in recent years, especially in supporting the tourism sector.[1].

The tourism sector is one of Indonesia's leading industries, as it has the potential to boost economic growth and holds a strategic position in supporting other sectors such as transportation, accommodation, entertainment, and other ancillary sectors**.** In April 2024, the number of tourists reached 1.07 million, an increase of 23.23% compared to the same period the previous year[2]. Along with this development, various challenges have arisen, including environmental sustainability issues, competition among destinations, and the importance of digitalization in the tourism sector.

For modern travelers, quick and easy access to information is an essential need when planning their journeys. Therefore, technology, particularly Android applications, plays a critical role in providing the necessary information and services to tourists more efficiently. The importance of the tourism sector as an economic driver has encouraged many countries, including Indonesia, to continually enhance the competitiveness and quality of destination services while ensuring the security and comfort of tourists[3]

In this context, digital innovation and sustainability have become key factors in addressing these challenges and maximizing the potential of the tourism sector**.** This development has increased the demand for efficient and easy information access among tourists when planning their travels. The rapid advancement of information technology has significantly enhanced the ability of tourists to access information effortlessly and efficiently during their travel planning process. Android applications have become a primary solution, offering various conveniences ranging from accommodation information to tourist destination recommendations**.** These applications allow tourists to quickly access real-time information they need. However, with the abundance of applications available on platforms such as Google Play, concerns about user data security have emerged. Some applications pose a risk of leaking personal or organizational data if not properly managed and secured[4]

Sensitive data breaches can occur due to insufficient mechanisms for storing and protecting information within applications, threatening user privacy and security. Therefore, more rigorous efforts are needed to ensure that Android applications used in the tourism sector are safe and reliable, preserving the confidentiality of user data.

Vulnerabilities and risks related to privacy are often introduced into applications during development, either

intentionally or through negligence. Additionally, many mobile applications initiate connections to websites, other applications, or services beyond their scope, exposing unsuspecting users to significant risks. Therefore, Android data encryption or related privacy controls must be considered during the application development stage[5].

Penetration testing or penetration testing (Pentest) is an important method for identifying vulnerabilities in application architecture. The purpose of this testing is to find and fix vulnerabilities before they are exploited by irresponsible parties. In addition to applications, penetration testing is also applied to networks and operating systems to detect and exploit discovered vulnerabilities with the aim of improving overall security. In the context of application security, penetration testing tends to focus on network and sensitive data extraction, including dangerous parameters such as dangerous permissions, weak crypto, root detection, SSL bypass, as well as malware domains. These elements, if not properly identified and addressed, can compromise the security of application user data. Thus, penetration testing is an important step to ensure that systems, applications and networks remain safe from potentially detrimental external threats, especially when it comes to protecting sensitive data. If these items are not properly identified and addressed, the security of application user data may be compromised. Therefore, penetration testing is an important step to ensure that systems, applications and networks are protected from potentially harmful external threats, especially when it comes to protecting sensitive data. As technology evolves and cyber threats become more complex, penetration testing becomes increasingly important. Penetration testing is an important part of maintaining information and data security in various digital environments[6]

In this research, penetration testing will be carried out on tourism applications on the island of Lombok to identify vulnerabilities that can be exploited from existing vulnerabilities based on the OWASP Top 10 Mobile Risk.

II. Literature Riview

In previous research conducted by [7], vulnerabilities were analyzed, and permissions were checked as part of a static analysis of tourism application security. This research used supporting tools to perform static analysis processes, such as emulators and MobFS, to scan for vulnerabilities in tourism applications. The previous research and this research shared several similar contexts; however, the difference lay in the development of further tests to ensure vulnerabilities in tourism applications that had been scanned by performing penetration testing on the application to identify security holes and their causes. The 10 vulnerability threats listed in the OWASP Top 10 Risks were as follows:

- M1: Improper Credential Usage: vulnerabilities involving storing or handling credentials inserted directly into source code or transmitted without encryption, which could be exploited by attackers to gain unauthorized access.
- M2: Inadequate Supply Chain Security : vulnerabilities that pose a risk from third party users such as SDK that can become an entry point for attackers by inserting malicious code.
- M3: Insecure Authentication/Authorization : vulnerabilities that have weaknesses in the authentication and authorization processes that allow attackers to steal or spoof a user's identity to access protected data or functions.
- M4: Insufficient Input/Output Validation : This risk refers to the lack of validation and sanitization of data entering or exiting the application, which can lead to attacks such as SQL injection or cross-site scripting (XSS).
- M5: Insecure Communication : This risk is that the transmission of data without encryption or through insecure channels, allows attackers to carry out attacks to steal or modify data.
- M6: Inadequate Privacy Controls : This vulnerability represents a failure to protect users' personal information, which can lead to leaks of sensitive data and breaches of privacy.
- M7: Insufficient Binary Protections : This refers to the lack of mechanisms to prevent reverse engineering or code modification, which attackers can exploit to identify vulnerabilities or create malicious versions of the application.
- M8: Security Misconfiguration : This vulnerability arises from improper security settings or unchanged defaults, which can provide an entry point for attackers to exploit the system.
- M9: Insecure Data Storage : This risk involves storing sensitive data without encryption or in easily accessible locations, embedding sensitive data in plain text that attackers can exploit.
- M10: Insufficient Cryptography : The use of weak cryptographic algorithms or incorrect implementations that do not provide adequate protection for sensitive data.

Research conducted by [8], analyzed malware using static MobSF on APK Android applications, emphasizing the importance of using tools such as MobSF for conducting static analysis. The study utilized the Mobile Security Framework (MobSF), which was an effective method for detecting vulnerabilities in Android applications before the applications were run. MobSF conducted a thorough examination of the APK file to identify potential malware risks, including insecure permissions, weak data encryption, and components vulnerable to exploitation, whether or not the vulnerabilities in the Android app were actually malicious.

In this research [9], the focus was on penetration testing, an important method that was often used to ensure security in mobile applications. Security testing was conducted on mobile applications using the OWASP methodology as a reference, focusing on the main aspects of security,

including session management, authentication, and insecure data storage. This research was carried out systematically, and the approach to OWASP also provided guidance or suggestions for addressing these security gaps. On the other hand, this research emphasized the importance of testing mobile applications to detect previously undetected risks by manually performing penetration testing, to identify potential data leaks, unencrypted storage of sensitive data, and weak authentication.

Mobile and computer applications have grown rapidly in recent years, posing certain threats to data security and user privacy. In a previous study conducted by [10], which focused on security solutions for mobile applications, including cryptographic encryption, data leakage mitigation, and security algorithm implementation. During application development, this study emphasized conducting penetration testing to identify vulnerabilities and threats before the application is released to the public. This study provides important insights into how penetration testing methodologies and new technologies can be used to strengthen mobile application security.

Based on previous research conducted by [11], entitled "Security Analysis of Mobile Commerce Applications using Mobile Security Framework (MOBSF) and OWASP Mobile Application Security Testing Guide (MSTG)," the importance of conducting penetration testing to validate vulnerability issues in testing local storage parameters for sensitive data in mobile applications after application mapping is emphasized. However, after penetration testing was performed as the exploitation stage, it turned out that the parameter issue was not identified as a vulnerability. This proves that simply performing a static scan or analysis using the MOBSF tool can also result in false positives.

Based on the results of research from exploitation conducted on the OWASP MASTG framework which found vulnerabilities of unencrypted and exposed sensitive data in communication between client and server. which hackers can carry out attacks in the form of brute force to be able to guess or hack application user passwords that allow account theft or sensitive information that can be misused.

III. Research Methodology

The methodology used in this research consists of five parts which develop the approach from previous research. In previous research, penetration testing was not carried out, but in this research a penetration testing section was added. The following is a picture of the methodology diagram for this research.

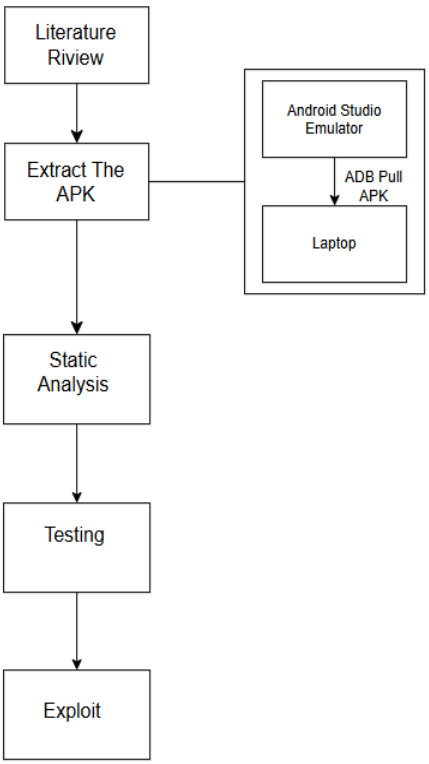

Fig. 1. Research Stages methodology diagram

The first step taken in this research is to conduct a literature review where data or information is collected as a literature review to understand the research that has been done with static testing. It also serves as a foundation for further development of dynamic research. The research begins by determining the target of one of the 17 android applications for tourism in Lombok that have been tested with static analysis before. To prove the MobFS vulnerability problem, this research makes penetration testing on the application.

Then the second step is to Extract the APK, at this stage of the research, the installation of the tourism mobile application which will be the target of the trial using the Google Play store in the Android Studio Emulator, this step is carried out so that the leptop can pull the tourism application using the Android Debug Bridge which then later the application can be static analyzed.

The next stage in this research is to conduct Static Analysis using several tools such as Mobile Security Framework (MobFS) as an android-based static analysis tool to scan application source code, IPA, or binary files without running the android application or scanning sensitive data management security vulnerabilities in the android application. then perform API and licensing checks that detect the potential for functional misuse of the application. After knowing there is a vulnerability issue, then conduct further analysis using JADX to find out and

check the code on the source of the vulnerable android application after scanning using MobFS which then engineers the android package file using JADX.

After that, the testing stage is a penetration testing method for testing vulnerabilities in the android application after scanning using MobFS, which scans several vulnerabilities in the android application, the results of the scan that have been carried out are identified as part of the OWASP top 10 risks, which are then carried out validation testing whether the vulnerabilities that have been identified will interfere with the android application system.

And finally Exploit is the final step in this research where to evaluate vulnerabilities that can be utilized, in order to find out the extent of the danger of vulnerability issues that have been identified by MobFS. The process to exploit in this research uses several tools and is also done manually.

IV. Results and Discussion

A. *Extract The APK*

In this research, an experiment was conducted to conduct penetration testing on one of the Lombok tourism mobile applications that had been statically tested in previous studies while proving the vulnerabilities displayed by MobFS, the following 17 Lombok tourism applications.

TABLE I. List of Tourism Applications and APK Version

No	**Application**	**Version APK**
1	Wisata Lobar	1.0.0
2	ASAP Lombok	3.0.2
3	Lombok Astoria Hotel	1.1.5
4	DLU Ferry	3.0.1
5	Ferizy	3.3.3
6	GORAtiket	1.4.0
7	Lallo	2.0.2
8	Lombok Tourism	2.4.0
9	MyTL	2.159
10	Panorama	1.0
11	Rinjani Visitor	0.4.0
12	Gili Termena	1.0.1
13	Amaq Goncank Lombok Travel	1.4.0
14	LOTIMGO	1.0.0
15	Ayo Ke Lombok	1.0.0
16	SiGis Tetebatu	1.0.8
17	E-Ticket DLN	1.7.7

From the data collected by previous research, there are 17 Lombok tourism mobile applications that have been carried out static analysis and of the 17 applications above, there is one mobile application that will be tested penetration testing in order to find out further whether the vulnerability issues that have been analyzed using MobFS are dangerous or not.

The selection of the Asap Lombok application as the application to be tested from 17 other tourism applications was taken based on the results of MobFS Scanning which contained many vulnerabilities contained in this application from 17 Lombok tourism applications, the Smoke Lombok application and the Asap Lombok application have old APK versions and do not get updates which have a lot of security risks.

The first step taken in this research is to extract the Lombok tourism APK which will be the target for testing into a leptop so that it can upload the Lombok tourism APK into MobFS. The following is the display of the Lombok tourism APK that has been extracted into the leptop and installed into Android studio.

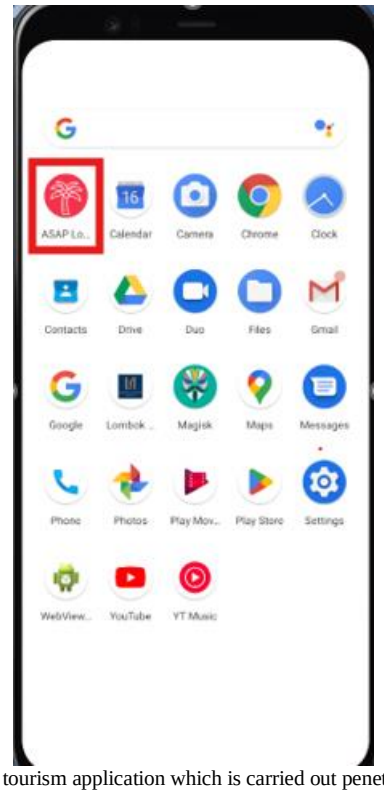

Fig.2. Lombok tourism application which is carried out penetration test

B. *Static Analysis*

In the static analysis, the reverse engineering method is used as a static analysis test method that decompiles the Asap Lombok APK using the MobFS tool, as shown in Figure 3.

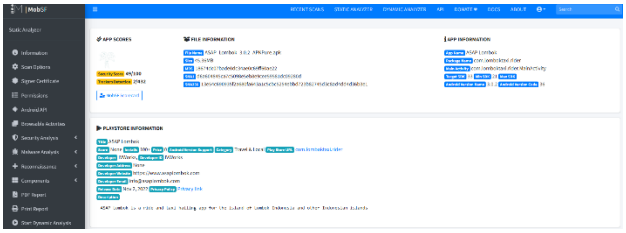

Fig.3. scanning using MobFS and location information of APK data files

After completing the scan, there are several vulnerability issues identified by MobFS where the detected vulnerabilities will be tested to ensure or validate whether the vulnerabilities mentioned are dangerous to users or only harmless vulnerability warning issues as shown in Figure 4.

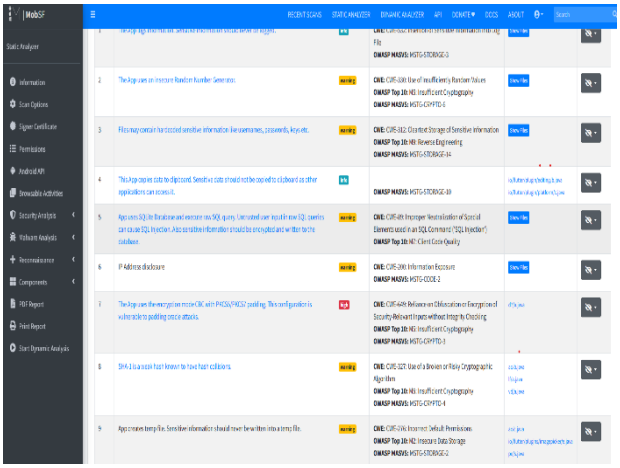

Fig.4. Vulnerability issues in tourism mobile applications Asap Lombok

In its calculation, MobSF has a rule that will be assigned a value of one hundred, which is considered the optimal condition for the application to be tested. For each result, an additional or subtracted value will be assigned depending on the characteristics of the vulnerability. If the vulnerability is harmless or in the good category, it will be given an additional value of 5 for each vulnerability, whereas if the vulnerability is in the high or warning category, it will be given a subtraction value of 15 or 10. to calculate the final result by using all the code analysis vulnerability findings. The final score will be rounded to 100 if it exceeds 100 and 10 if it is less than 0. To indicate the condition of the application, the MobSF test security score is divided into four score categories, as in table II.

TABLE II. VULNERABILITIES AND MOBFS VALUES

Vulnerability category	***Range of vulnerability values***
Critical	0-16
High	16-40
Medium	41-70
Low	71-100

TABLE III. MOBFS ISSUE SCAN RESULTS

Severity	**OWASP top 10 2024**	**Issue**
Warning	**M9**: Insecure Data Storage	The application creates temporary files. Sensitive information should not be written into temporary files, as these files are vulnerable to unauthorized access and may compromise data security.
High	**M10**: Insufficient Cryptography	The application uses CBC (Cipher Block Chaining) encryption mode with PKCS5 or PKCS7 padding. This configuration is vulnerable to oracle padding attacks.

The results of the static analysis using MobFS in Figure 4 there are several vulnerability issues displayed along with the code that has vulnerabilities in the application. but there are only two vulnerabilities that are included in the OWASP Mobile top 10 list in 2024 because several other issues displayed on MobFS have not been updated and are still included in the OWASP Mobile top 10 list in 2016, which then we can find out that only two vulnerability issues are listed in the OWASP Mobile top 10 2024 as in table III. After that, further analysis was carried out on the part of the code that was vulnerable to Insecure Data Storage and Insufficient Cryptography attacks. by uploading the Asap Lombok application code to the JADX tool to read the application code.

B.1. Source code analysis

The From the results of the scan conducted using MobFS, it was found that there were Insecure Data Storage and Insufficient Cryptography vulnerabilities in some source code in this application. Then after scanning, at this stage an examination is carried out first on the source code detected with Insecure Data Storage and Insufficient Cryptography vulnerabilities using JADX as shown in Figure.

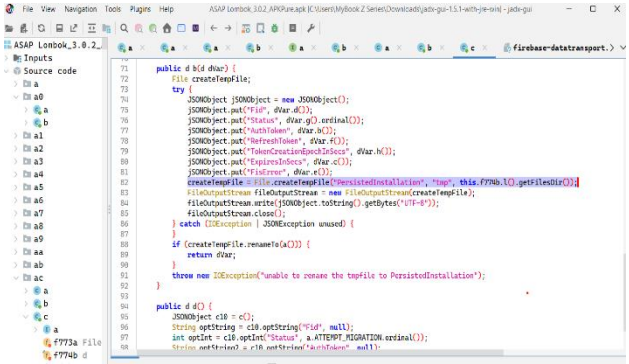

Fig.5. Insecure Data Storage vulnerability in databases

Figure 5 shows the Insecure Data Storage vulnerability issue identified in the Asap Lombok application database. This vulnerability indicates a potential risk related to insecure data storage, such as sensitive user data stored without adequate encryption.

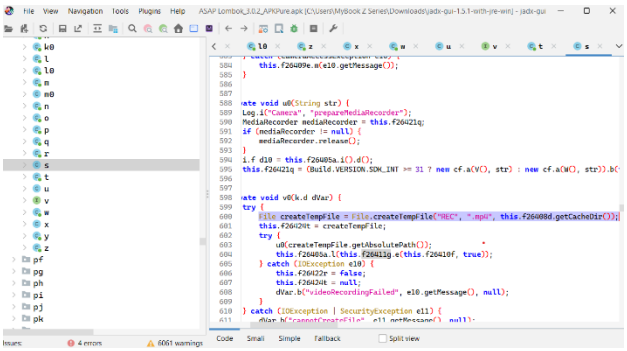

Fig.6. Insecure Data Storage vulnerability in databases

Figure 6 shows the Insecure Data Storage vulnerability issue found in the file.xml of the Asap Lombok application. This vulnerability occurs because the file.xml is used to store information that should be sensitive without adequate security protection, such as encryption.

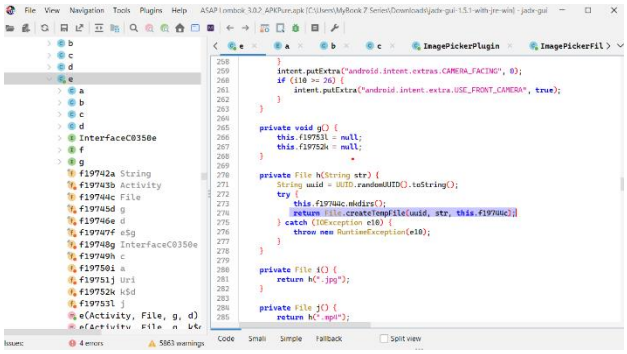

Fig.7. Insufficient Cryptography vulnerability in source code

in Figure 7 shows the Insufficient Cryptography issue identified in the Asap Lombok application source code. This vulnerability occurs due to inadequate implementation of cryptographic algorithms or the use of outdated encryption methods, such as CBC with PKCS7 padding.

C. Testing

1) M9 : *Insecure Data Storage*

location of the source code that has an Insecure Data Storage vulnerability after that testing is carried out with the Testing Method using several tools available on Android Studio and checking Databases as well as data on shared pref.xml which is located on the Lombok Smoke application data, as seen in Figure 8 and 9 the results of testing.

Fig.8. contents of the Asap Lombok application databases

Fig.9. file *sharedpref.xml*

2) M10 : *Insufficient Cryptography*

After scanning using MobFS, there is an Insufficient Cryptography vulnerability issue in the Asap Lombok application which is a Lombok tourism application which is tested on vulnerabilities that still use CBC encryption mode with PKC55 / PKC57 pandding which has been found in the source code as shown in Figure 10.

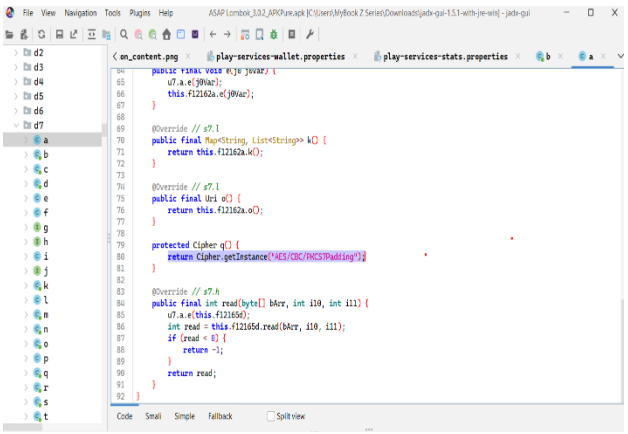

Fig.10. Insecure PKCS7Padding encryption mode

The PKCS7padding used is a development of PKCS5padding, there are several main differences in both which are significant, namely located in the block, in PKS5padding is designed only for some blocks that are only 8 bytes in size, while in PKCS7padding supports that can support any block size from 1 byte to up to 255 bytes which in this comparison PKCS5padding can be considered only as a subset of PKS7padding which is specifically for 8 byte blocks. Therefore PKCS7padding is more commonly used in Cryptographic Massage Syntax.

The use of padding has some significant security flaws in that when CBC is used without additional safeguards, the application may become vulnerable to padding oracle attacks. This attack simply utilizes the app's different responses when the padding is valid or invalid so that it can decrypt the chipper.text byte by byte without requiring an encryption key to crack the app.

D. Exploit

The next stage in exploiting the two vulnerabilities. at vulnerability M9: Insecure Data Storage found gaps in databases as well as file.xml and then in vulnerability M10: Insufficient Cryptography found vulnerabilities in application padding.

In the vulnerability of databases after checking and testing the exploit by taking the databases file and compressing it using SQLite as shown in Figure 11, there is no important data such as personal data of application users. And in testing the shared pref.xml file by opening the contents of the file.xml using a Command Prompt that has been connected to ADB which can connect directly to the device as shown in Figure 9. And using android studio to try to open and try to exploit by changing one of the files.xml as shown in Figure 12.

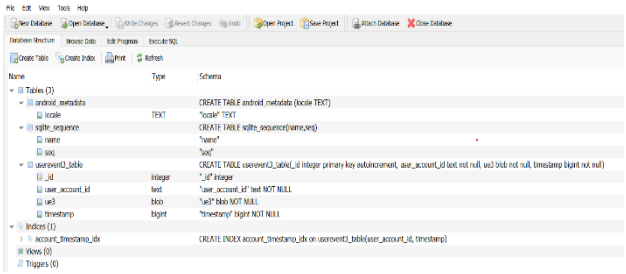

Fig.11. view of the Asap Lombok app databases that were exploited

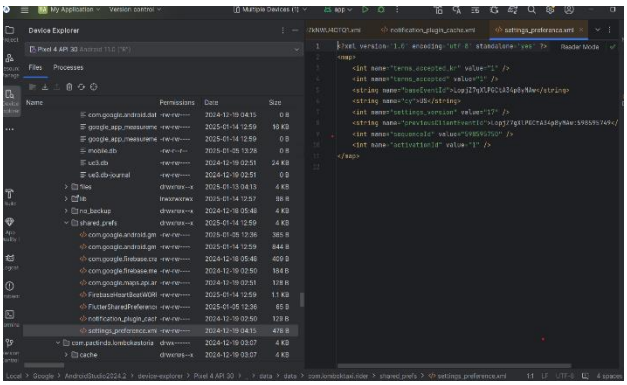

Fig.12. test exploit file.xml using Android Studio

From Figure 12 the.xml file can be further exploited and can be opened by the attacker but the.xml file cannot be changed. there is no sensitive data information to be stolen, because there is no display that displays the user's personal data and also the application user password.

Furthermore, the Insufficient Cryptography vulnerability is located in the layout application padding of the display using PKCS7 which can be seen in Figure 13. Based on exploitation testing carried out on the Insufficient Cryptography issue detected by MobFS which has a vulnerability in the layout application padding, there is no impact that will harm the user. because there is only map location data in the application.

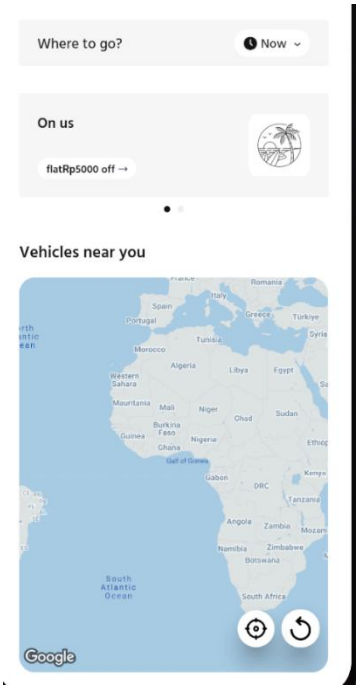

Fig.13. display of the Lombok smoke application that PKCS7padding uses that has been exploit

V. CONCLUSION

Based on the results of the research that has been done, namely Security Analysis of the Lombok tourism application based on Android using the penetration testing method, there are several findings when conducting static analysis using MobFS, there are two vulnerability issues on the OWASP TOP 10-2024 list such as:

In the Insecure Data Storage identified, a vulnerability that can open the.xml file but the.xml file in the APK cannot be changed, which has no harmful impact on the Asap Lombok APK.

Insufficient Cryptography in APKs that still use PKCS7 padding has some security flaws, such as that when CBC is used without additional protection, the app can be vulnerable to padding oracles but it doesn't have much impact on the APK.

REFERENCES

[1] T. Yuniati, A. R. Tambunan, and Y. A. Setyoko, "Implementasi Static Analysis Dan Background Process Untuk Mendeteksi Malware Pada Aplikasi Android Dengan Mobile Security Framework," *LEDGER : Journal Informatic and Information Technology*, vol. 1, no. 2, pp. 24–28, Oct. 2022, doi: 10.20895/ledger.v1i2.848.

[2] N. Salsabila, "ANALISIS PENGARUH SEKTOR PARIWISATA TERHADAP PERTUMBUHAN EKONOMI (Studi Kasus Pada Negara Asia Tenggara Maritim)."

[3] "Official Celebration of World Tourism Day 2018 - Tourism and the digital transformation."

[4] A. S. Rusdi, N. Widiyasono, and H. Sulastri, "Analisis Infeksi Malware Pada Perangkat Android Dengan Metode Hybrid Analysis."

[5] C. Montealegre, C. R. Njuguna, M. I. Malik, P. Hannay, and I. N. McAteer, "SECURITY VULNERABILITIES IN ANDROID APPLICATIONS," in *Proceedings of the 16th Australian Information Security Management Conference, AISMC 2018*, Edith Cowan University, 2018, pp. 14–28. doi: 10.25958/5c5274d466691.

[6] D. D. Bertoglio, G. Girotto, C. V. Neu, R. C. Lunardi, and and A. F. Zorzo, "Pentest on an Internet Mobile App: A Case Study using Tramonto," Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.09779

[7] R. B. Huwae, A. H. Jatmika, A. Z. Mardiansyah, and A. Zubaidi, "ANALISA STATIS KEAMANAN APLIKASI PARIWISATA DI LOMBOK BERBASIS ANDROID BERDASARKAN FRAMEWORK OWASP MOBILE TOP 10-2023 (STATIC ANALYSIS OF TOURISM APPLICATION SECURITY IN LOMBOK BASED ON ANDROID BASED ON OWASP MOBILE TOP 10-2023 FRAMEWORK)."

[8] I. Himawan, K. Septianzah, and I. Setiadi, "ANALISA RESIKO MALWARE DENGAN STATIC MOBSF TERHADAP APLIKASI ANDROID APK," *Technologia : Jurnal Ilmiah*, vol. 14, no. 4, p. 364, Oct. 2023, doi: 10.31602/tji.v14i4.11460.

[9] A. Alanda, D. Satria, H. A. Mooduto, and B. Kurniawan, "Mobile Application Security Penetration Testing Based on OWASP," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, May 2020. doi: 10.1088/1757-899X/846/1/012036.

[10] Kefei. Chen, Qi. Xie, Weidong. Qiu, Ninghui. Li, and W.-Guey. Tzeng, *ASIA CCS'13 : proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security : May 8-10, 2013, Hangzhou, China*. Association for Computing Machinery, 2013.

[11] Shaqila Erbeliza, "ANALISIS KEAMANAN APLIKASI MOBILE COMMERCE MENGGUNAKAN MOBILE SECURITY FRAMEWORK (MOBFS) DAN OWASP MOBILE APPLICATION SECURITY TESTING GUIDE (MASTG)," Jakarta, 2023.