2024 IEEE 2nd International Conference on Electrical Engineering, Computer and Information Technology (ICEECIT)
University of Jember, Jember, Indonesia, November 22-23 2024

2024 IEEE 2nd International Conference on Electrical Engineering, Computer and Information Technology (ICEECIT) | 979-8-3315-0437-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICEECIT63698.2024.10859915

Evaluating Compliance of of the XYZ Ministry's Android Messaging Applications with OWASP MASVS: A Comprehensive Case Study

Grace Friscilla Margaretha Karo-Karo
Cybersecurity Engineering
National Cyber and Crypto Polytechnic
Bogor, Indonesia
grace.friscilla@student.poltekssn.ac.id

Susila Windarta
Cybersecurity Engineering
National Cyber and Crypto Polytechnic
Bogor, Indonesia
susila.windarta@poltekssn.ac.id

Amiruddin
Cybersecurity Engineering
National Cyber and Crypto Polytechnic
Bogor, Indonesia
amir@poltekssn.ac.id

Ira Rosianal Hikmah
Cybersecurity Engineering
National Cyber and Crypto Polytechnic
Bogor, Indonesia
ira.rosianal@student.poltekssn.ac.id

***Abstract*— Safeguarding the security of government communication tools is crucial for protecting sensitive information. This study presents a thorough case analysis that assesses the compliance of the XYZ Ministry's Android messaging app with the OWASP Mobile Application Security Verification Standard (MASVS). OWASP MASVS offers a robust framework for evaluating the security stance of mobile applications, encompassing vital areas such as data protection, authentication, and network security. The study applied the OWASP MASVS standard to the XYZ Ministry's messaging application using automated and manual testing approaches to gauge adherence to security standards. Specifically, the research concentrated on MASVS-NETWORK-2 security controls. The primary goal of this research was to uncover vulnerabilities, evaluate compliance, and put forth actionable recommendations for enhancing application security. The findings indicated that the tested application failed to meet two of the five test sections in MASVS-NETWORK-2. The recommended course of action is to modify the code in Network Security Settings to accept certificates from the system exclusively.**

Keywords*— Android, Application Security, Data Protection, OWASP MASVS, Security Compliance

I. INTRODUCTION

In today's digital era, instant messaging applications have become integral to communication and information exchange. According to a survey conducted by Statista [1] between July 2022 and June 2023 across 21 countries with respondents aged 18-64, an average of 75 percent of mobile phone users utilized their devices for chatting and messaging. The use of instant messaging applications is not confined to personal needs but has also become a critical communication tool in the business and professional sectors, including within government operations. According to a report by the Check Point Research Team [2], the government and military sectors ranked second among the most frequently targeted institutions in the cyber domain in 2023. The number of attacks increased by 8 percent, from 1,626 attacks recorded in the second quarter of 2022 to 1,772. Government ministries rely on these applications to exchange sensitive information and facilitate secure communication. Therefore, ensuring these applications comply with stringent security standards is crucial to protecting against data breaches and unauthorized access.

The Open Worldwide Application Security Project (OWASP) Mobile Application Security Verification Standard (MASVS) provides a comprehensive framework designed to assess and enhance the security posture of mobile applications. OWASP MASVS outlines requirements and best practices across various security domains, including data protection, authentication, and network security. Compliance with this standard helps mitigate potential vulnerabilities and safeguards the integrity and confidentiality of mobile applications. Among the seven security control groups in OWASP MASVS, this research focuses on MASVS-NETWORK security controls at Level 2, referred to as MASVS-NETWORK-2. This focus is based on a report from Positive Technologies [3] indicating that 38 percent of mobile applications are vulnerable to Insecure Communication, which currently ranks third in the OWASP Top 10 Mobile Risks 2016. This vulnerability poses a significant risk of exploiting critical information in mobile applications, particularly on Android.

According to data from Statista [4] in the third quarter of 2023, Android dominated the market with a 70.5% share, followed by Apple's iOS with a market share of approximately 28.83%. Therefore, this paper focuses on a case study of the XYZ Ministry's Android messaging application, evaluating its compliance with OWASP MASVS. By examining how well the application adheres to MASVS criteria, this research aims to identify security gaps and provide insights into areas requiring improvement. The evaluation process combines both automated and manual testing techniques to assess the application's compliance with established security benchmarks.

This study presents three main contributions: (1) evaluating the security compliance of the XYZ Ministry's messaging application based on OWASP MASVS, (2) identifying common vulnerabilities and compliance gaps, and (3) providing actionable recommendations for improving the application's security posture. The insights gained from this case study offer practical value to developers, security professionals, and policymakers responsible for government communication tools. By assessing the security status of a critical government messaging platform, the research emphasizes the need for robust security standards to protect sensitive communications and ensure trustworthy digital communication platforms.

The structure of this paper is organized as follows: Section 2 discusses the literature review, providing the theoretical framework and context relevant to this study. Section 3 details the research methodology, including the evaluation approach

This research was supported by Politeknik Siber dan Sandi Negara.

and data collection techniques. Section 4 presents the findings, analysis, and discussion of the key results and their implications. Finally, Section 5 concludes the paper by summarizing its contributions.

II. Literature Review

A. Theoretical Background

1) Android Application

Android is an open-source operating system based on the Linux kernel [5]. This operating system supports applications designed to run on mobile devices such as smartphones. According to [6], applications on mobile devices are generally more practical and relatively compact, facilitating easier deployment and operation. Android applications are developed using the Android Software Development Kit (SDK) and are typically written in the Java.

Android applications can be pre-installed, meaning they come bundled with the device. Additionally, applications can be downloaded from app stores provided by the device, such as the Google Play Store. There is also the option to download applications from third-party sources, such as Android Package Kit (APK) files. APK files contain all necessary elements to run the application, including code, resources, images, and application descriptions [7].

2) OWASP Mobile Application Security Verification Standard (MASVS)

OWASP MASVS is a mobile application security standard designed to assist developers in creating secure mobile applications [8]. OWASP MASVS defines three levels of requirements: MASVS-L1, MASVS-L2, and MASVS-R. This framework is applicable in different contexts based on the testing level. Compliance with MASVS-L1 indicates that the application adheres to best practices and does not exhibit common vulnerabilities. MASVS-L2 introduces additional defensive controls, such as Secure Socket Layer (SSL) pinning, to protect against more sophisticated attacks. Meeting some or all of the MASVS-R requirements specifically addresses client-side threats.

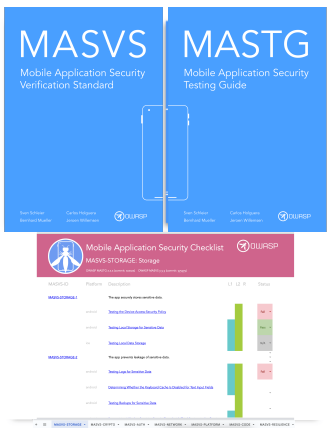


Fig. 1. OWASP MASVS, MASTG, and MAS

OWASP MASVS outlines several security control groups for assessing mobile application security, including:

a) MASVS-STORAGE
b) MASVS-CRYPTO
c) MASVS-AUTH
d) MASVS-NETWORK
e) MASVS-PLATFORM
f) MASVS-CODE
g) MASVS-RESILIENCE

MASVS-NETWORK is a security control designed to ensure that mobile applications provide secure connections under all circumstances. This control specifically focuses on verifying that secure and encrypted channels are used for network communication within the application. Additionally, this category addresses scenarios where developers may choose to trust only certain Certificate Authorities or implement certificate pinning or public key pinning [9].

MASVS-NETWORK is divided into two levels. MASVS-NETWORK Level 1, hereafter referred to as MASVS-NETWORK-1, covers basic requirements for network security in mobile applications, while MASVS-NETWORK Level 2, hereafter referred to as MASVS-NETWORK-2, encompasses more advanced and comprehensive requirements. The security controls outlined in MASVS-NETWORK-2 include all the controls present in MASVS-NETWORK-1. This research focuses on MASVS-NETWORK-2. The levels of MASVS-NETWORK are detailed in Table I above..

TABLE I. Levels in MASVS-NETWORK

Level	Control Description
MASVS-NETWORK-1	The application secures all network traffic in accordance with current best practices.
MASVS-NETWORK-2	The application performs identity pinning for all remote endpoints under the developer's control.

In practice, OWASP MASVS is integrated with the testing sections of OWASP Mobile Application Security Testing Guide (MASTG) [10] and OWASP Mobile Application Security (MAS) Checklist [11] as shown in Fi. OWASP MASTG provides an in-depth guide for performing penetration testing on mobile applications. It outlines the technical procedures required to verify the security controls specified in OWASP MASVS. In addition, OWASP MASVS is complemented by a comprehensive checklist for thoroughly re-evaluating tests that have already been conducted. The OWASP MAS contains a detailed list of tests from the OWASP MSTG, along with links to the corresponding test requirements categorized under the security control groups defined in the OWASP MASVS. In summary, MASVS sets the security standards, MSTG provides the testing techniques, and MAS serves as a checklist to ensure all relevant tests are conducted and aligned with MASVS requirements.

B. Related Work

An earlier work [12] explored vulnerability assessment on Parental Control Mobile Applications Security based on OWASP Security Requirements. The evaluation utilized the control groups from OWASP MASVS as security requirements and identified common vulnerabilities in three parental control applications. However, the study did not use OWASP MSTG as a manual guideline for compliance evaluation. Instead, it relied on the Quixxi Security scanning tool for automated scanning.

Another evaluation in the research [13] focused on determining Android mobile banking application's compliance with the requirements of OWASP MASVS. However, research solely concentrated on unauthenticated client-side testing using a black-box testing approach and did

not reference OWASP MASTG as a guideline or OWASP MAS as a checklist.

Finally, the study [14] conducted penetration testing on mobile applications based on the OWASP Top 10 Mobile Application Vulnerabilities. Although the study provided valuable information on risk levels, impacts, and recommendations based on compliance evaluation through testing according to the OWASP Top 10, it did not include OWASP MASVS and OWASP MASTG.

This emphasizes the need for a compliance evaluation plan that conducts tests aligned with the security controls in OWASP MASVS and OWASP MASTG.

III. RESEARCH METHODOLOGY

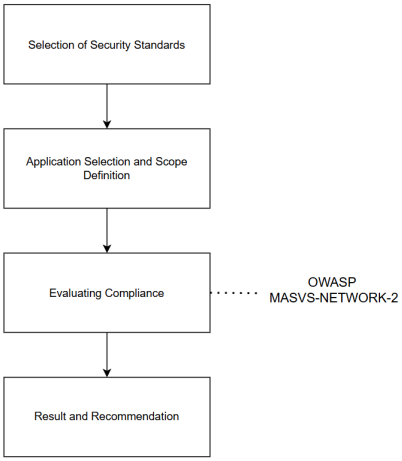


Fig. 2. Research Methodology for Evaluating Compliance

Fig. 2 provides an overview of the methodology utilized to assess the compliance of the XYZ Ministry's Android messaging applications with the OWASP MASVS. The research was meticulously crafted as a comprehensive case study, honing in on a specific application used within the ministry to ascertain its alignment with well-defined security standards. The structured and detailed methodology draws upon OWASP's frameworks to thoroughly evaluate security compliance.

A. Selection of Security Standards

The process of selecting security standards aims to comprehensively identify and implement the appropriate standards and controls throughout the research. This meticulous approach ensures the study is grounded in a strong security framework aligned with industry best practices, addressing the application's security requirements and challenges. The careful selection of these standards ensures a comprehensive and reliable research assessment.

B. Application Selection and Scope Definition

This stage is focused on clearly defining the scope of the study by carefully selecting the application(s) that will be evaluated. In addition, it involves establishing the specific boundaries and parameters that will guide the research, ensuring that the chosen application is thoroughly assessed within a well-defined framework. This includes determining the aspects of the application to be scrutinized, the criteria for evaluation, and the overall objectives of the assessment, all of which are crucial for a comprehensive and focused analysis.

C. Evaluating Compliance

The objective of this stage is to evaluate the extent to which the application in question adheres to the requirements set forth by MASVS. To ascertain this, a series of tests are conducted using the standards set forth by OWASP MASTG, encompassing both static and dynamic analyses. The analyses were conducted using a combination of manual testing with tools such as JADX GUI, Wireshark, and automated scanning with tools such as MobSF and Frida. Subsequently, the compliance evaluation results of will be re-examined using the OWASP MAS checklist. The outcome of this stage will be the compliance evaluation results for each test section as per OWASP MASTG.

D. Result and Recommendation

This stage involves consolidating the findings from the compliance evaluation phase and formulating actionable recommendations to address any identified security deficiencies. The goal is to assess the application's security posture clearly and provide practical guidance for enhancing compliance with OWASP MASVS. The deliverables of this stage include a detailed compliance evaluation results table and a recommendations table, both derived from the evaluation findings and aimed at improving the application's security measures and aligning them with industry standards.

IV. RESULT

A. Selection of Security Standards

Security Standard is needed as the core for evaluating compliance of an application. There are several standards that can be used as references for compliance evaluation. The comparison table between OWASP MASVS and several other standards is provided below in Table II. This comparison aids in understanding the similarities and differences among these standards and assists in determining the most appropriate and comprehensive framework for evaluating security compliance.

Table II. Comparison Table of Security Standard

Criteria	**OWASP MASVS** [9]	**ISO/IEC 27034** [15]	**NIST 800-163 Rev 1** [16]	**NIAP** [17]
Scope	Mobile App	App Security	Mobile App	Mobile App
Focus Area	Mobile app security requirements	Security in Information Systems and security risk management	Risk assessment and mitigation plans	Security requirements for IT products
Compliance Levels	Level 1 and 2	Level 0, Level 1, and Level 2	Level A, B, and C	Functional and Assurance
Testing Guide	OWASP MASTG	In the Document	In the Document	In the Document
Checklist	OWASP MAS	No Checklist	No Checklist	No Checklist
Update Frequency	Regular updates based on threats	Regular updates and revisions	Periodic updates as per NIST guidelines	Periodic updates and revisions

The OWASP MASVS was selected as the primary standard for evaluation due to its comprehensive nature and alignment with mobile application security requirements. The corresponding OWASP MASTG was used as the guideline for performing specific tests and MAS checklist providing comprehensive resources to ensure thorough and effective security evaluations. OWASP MASVS, cover various security controls and requirements, addressing all critical

security aspects. This comprehensive approach helps in identifying potential vulnerabilities that may not be apparent without a structured framework.

B. Application Selection and Scope Definition

The case study will focus on the XYZ Ministry's Android messaging application, which is utilized for internal communications. The application will be installed in the Android Emulator within Android Studio on an Android device running on the Android 9 platform. The assessment will concentrate on MASVS Level 2 network security control (MASVS-NETWORK-2) due to the sensitive nature of government communications. Both static and dynamic analyses will be employed during the evaluation. MASVS NETWORK-2 comprises five sections of tests conducted by MASTG, which are as follows: MASTG-NETWORK-0020, MASTG-NETWORK-0021, MASTG-NETWORK-0019, MASTG-NETWORK-0023, and MASTG-NETWORK-0022.

C. Evaluating Compliance

1) MASTG-NETWORK-0020

This section examines the Transport Layer Security (TLS) configurations utilized within the application. The methodology entails a static analysis of the application's APK file to scrutinize the TLS settings. This analysis is executed by extracting the APK file and an in-depth examination using the JADX GUI tool. Within the confines of the application, the connection parameters are pinpointed within the TlsVersion class. The extensive test outcomes demonstrate that the application has efficaciously instituted robust security measures across a spectrum of TLS versions, including TLS v1.0, TLS v1.1, TLS v1.2, TLS v1.3, and SSLv3. These findings are graphically represented in Fig. 3 for elucidation.

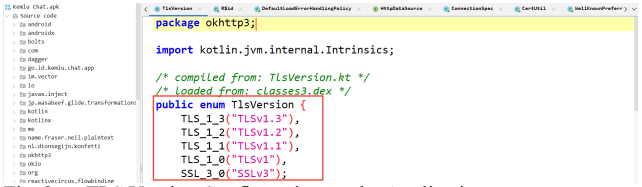


Fig. 3. TLS Version Configuration on the Application

After conducting a thorough analysis using Wireshark to monitor the incoming and outgoing packets of the program, we traced the discovered IP addresses to confirm the Wireshark traffic results, as depicted in Fig. 4. Based on the comprehensive findings from both static and dynamic assessments; it is evident that the application unequivocally meets the compliance assessment for MASTG-TEST-0020.

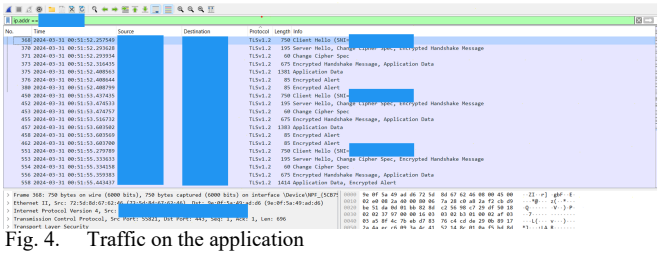


Fig. 4. Traffic on the application

2) MASTG-NETWORK-0021

In this section, Endpoint Identification Verification testing is conducted. This process involves verifying that the certificates used originate from trusted sources and ensuring that the endpoint, specifically the server, presents the correct certificate. Static analysis is performed using the Apktool. This tool is utilized to decompile the APK file, allowing the application's structure and configuration to be analyzed in detail.

Fig. 5. Custom CA Usage Permissions on the Application

The data shown in Fig. 5 indicates that the application uses a unique network security configuration that establishes a separate trust anchor. The configuration of this system requires the application to rely on Certificate Authorities (CAs) that the user supplies. This feature allows the application to trust both the CAs provided by the system and the operator.

The Burp Suite engine executed a Man-in-the-Middle (MITM) attack while conducting the dynamic analysis. To facilitate the interception and monitoring of network data, the self-signed certificate of Burp Suite was implemented, and the software was modified to function as a proxy. A test was conducted to confirm the certificate's validity by transmitting a message over the application. The findings revealed that the application placed complete trust in the PortSwigger certificate, enabling monitoring of its operations through the Burp Suite proxy. An effective intercept of the request at (https://xxx.xxx.xxx was documented in the Burp Suite proxy, as shown in Fig. 6. From the static and dynamic analysis findings, it can be concluded that the application did not meet the compliance evaluation requirements for the MASTG-TEST-0021 assessment.

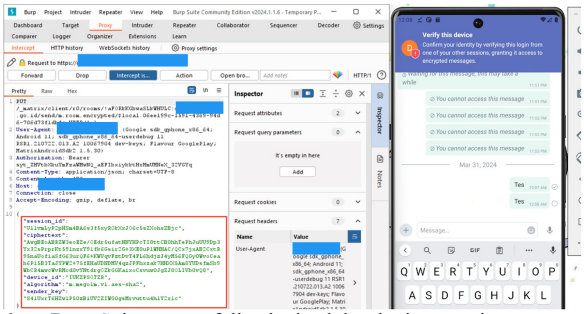


Fig. 6. Burp Suite successfully obtained data by interception

3) MASTG-NETWORK-0019

We are investigating whether the application permits unencrypted HTTP cleartext traffic. To conduct static analysis, we inspect the code related to the 'cleartextTrafficPermitted' setting using the JADX GUI tool. Additionally, we are examining the information in the 'network_security_config.xml' file, as illustrated in Fig. 7, to identify specific domains that may use plaintext traffic.

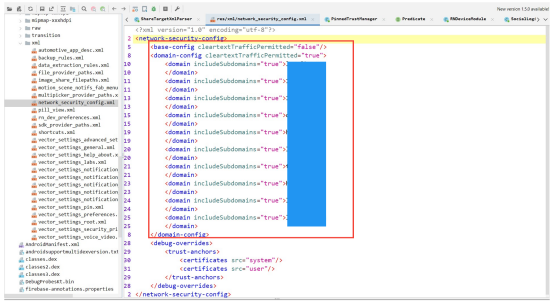


Fig. 7. Network Security Policy Configuration

As part of the dynamic analysis, we intercepted the network traffic originating from the application to confirm the encrypted data sent. To accomplish this, we utilized the proxy

interception function of Burp Suite. The interception results, depicted in Fig. 8, reveal that while the traffic was intercepted successfully, specific data like 'room_id,' the employed algorithm,'session_id,"sender_key,' 'code,' 'from_device,' and 'org.matrix.msgid' were discovered to be encrypted. The application successfully satisfies the compliance assessment for the MASTG-TEST-0019 based on the findings from both static and dynamic assessments.

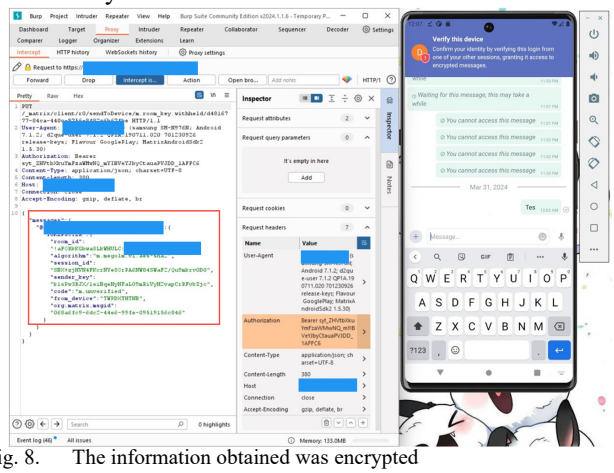
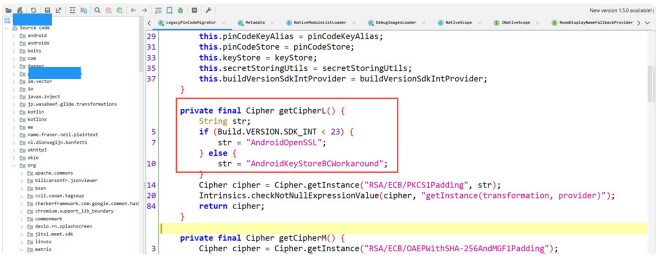


Fig. 8. The information obtained was encrypted

4) MASTG-NETWORK-0023

This section evaluates whether the application includes a security provider that meets the requirements. The testing is conducted through static analysis by examining the compiled code using the JADX GUI tool. The code related to the security providers used in the application, specifically `AndroidOpenSSL` and `AndroidKey StoreBCWorkaround, is shown in Fig. 9.



```
this.pinCodeKeyAlias = pinCodeKeyAlias;
this.pinCodeStore = pinCodeStore;
this.keyStore = keyStore;
this.secretStoringUtils = secretStoringUtils;
this.buildVersionSdkIntProvider = buildVersionSdkIntProvider;
}

private final Cipher getCipherL() {
    String str;
    if (Build.VERSION.SDK_INT < 23) {
        str = "AndroidOpenSSL";
    } else {
        str = "AndroidKeyStoreBCWorkaround";
    }
    Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding", str);
    Intrinsics.checkNotNullExpressionValue(cipher, "getInstance(transformation, provider)");
    return cipher;
}

private final Cipher getCipherM() {
    Cipher cipher = Cipher.getInstance("RSA/ECB/OAEPWithSHA-256AndMGF1Padding");
```

Fig. 9. Security Providers on the Application

Frida was used to perform dynamic analysis, which involved connecting it to the Android Studio emulator to analyze the installed application. This was accomplished by establishing a TCP connection between 'frida-core' and monitoring 'localhost:27042'. When the security provider hooking script was executed in Frida using Python, it was found that the application employs many security providers, as depicted in Fig. 10. The static and dynamic analysis findings complete the compliance assessment for the MASTG-TEST-0023 method.

```
C:\Users\ASUS\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\Scripts>python run_frida.py
Providers: AndroidOpenSSL version 1.0,CertPathProvider version 1.0,AndroidKeyStoreBCWorkaround version 1.0,BC version 1.61,HarmonyJSSE version 1.0,AndroidKeyStore version 1.0
```

Fig. 10. Security Providers Based on Frida Results

5) MASTG-NETWORK-0022

We conduct testing for Custom Certificate Storage and Certificate Pinning. Static analysis is performed by examining the code within the Network Security Configuration to identify elements related to custom certificate storage and certificate pinning functionality, as shown in Fig. 11.

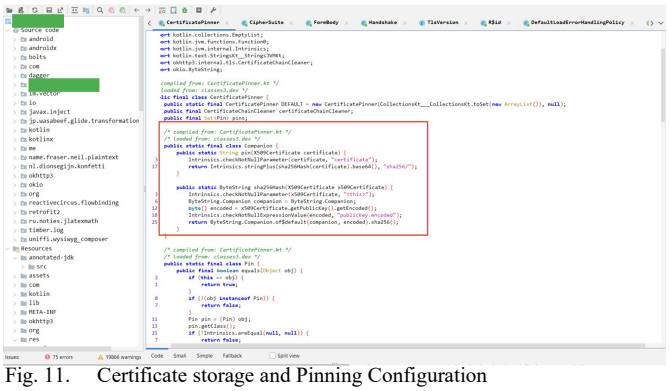


Fig. 11. Certificate storage and Pinning Configuration

During dynamic analysis, the application's traffic was intercepted using Burp Suite. As shown in Fig. 12, the application's activities were successfully intercepted and recorded in Burp Suite. This achievement was made possible by configuring the application in the `trust-anchors` section of the Network Security Configuration, as depicted in Fig. 5, which allowed trust for both system and user certificate sources. As a result, certificate pinning was not effectively enforced. Based on the results of both static and dynamic analyses, the application is deemed to have failed the compliance evaluation for the MASTG-TEST-0022 evaluation criteria.

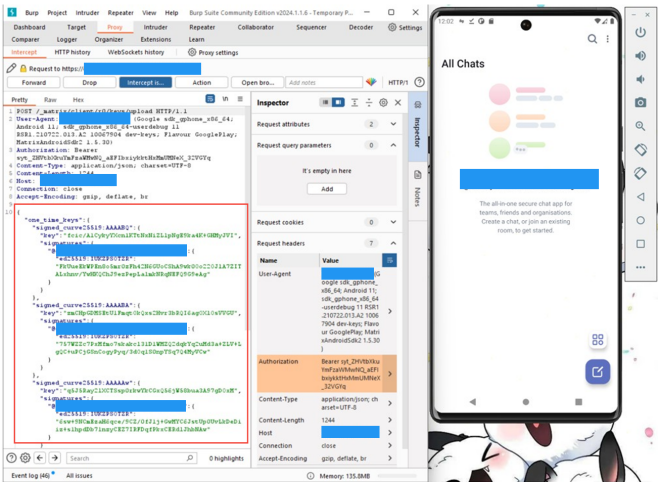


Fig. 12. The activities of the application can be intercepted

D. Result and Recommendation

The compliance evaluation results indicate that the application fails to fulfill the criteria for MASTG-TEST-0021 and MASTG-TEST-0022. Table III provides an overview of the outcomes of the Compliance Evaluation phase.

TABLE III. RESULTS OF THE COMPLIANCE EVALUATION

No.	Test Section	Meets The Requirements	
		Yes	No
1.	MASTG-TEST-0020	☑	
2.	MASTG-TEST-0019	☑	
3.	MASTG-TEST-0021		☑
4.	MASTG-TEST-0023	☑	
5.	MASTG-TEST-0022		☑

After completing the Compliance Evaluation phase, we conducted a thorough review by filling out the MAS Checklist in Fig. 13.

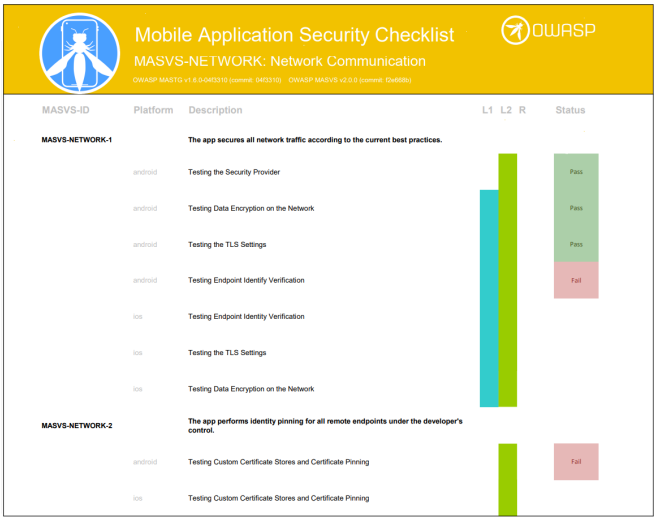


Fig. 13. MAS Checklist for MASVS-NETWORK

1) Recommendation

Based on the results of the Compliance Evaluation phase, recommendations for improvement are provided for the test sections that did not meet the requirements, serving as a guide for application enhancement. The improvement recommendations are detailed in Table 4.

Table 4 Improvement Recommendations

Test Section	Recommendation
MASTG-TEST-0021	Revise the code in the Network Security Configuration section by altering the certificate source settings. This involves changing the configuration to accept certificates from the system only, rather than from both the system and user sources. This modification is necessary to prevent interception or eavesdropping on the information exchange process through the application when user-provided certificates are still accepted.
MASTG-TEST-0022	Revise the code in the Network Security Configuration section by altering the certificate source settings. This involves changing the configuration to accept certificates from the system only, rather than from both the system and user sources. Although there is code that handles certificate pinning, it is necessary to configure the application to accept certificates only from the system. This adjustment ensures that the application does not accept certificates from sources like Burp Suite, which could be used for interception purposes.

V. CONCLUSION

Out of the five test sections in MASTG, two test sections did not meet the requirements during the Compliance Evaluation phase. These are Testing Endpoint Identification Verification (MASTG-TEST-0021) and Testing Custom Certificate Storage and Certificate Pinning (MASTG-TEST-0022). The results of the Compliance Evaluation indicate that the Kemlu Chat Android application has not yet met all the requirements of the MASTG test sections according to MASVS-NETWORK-2. Therefore, it is concluded that the application does not adhere to best practices and has not yet implemented additional, more in-depth defense controls as recommended by OWASP in the OWASP MAS Checklist.

The recommendations for MASTG-TEST-0021 and MASTG-TEST-0022 are to revise the code in the Network Security Configuration section by changing the certificate source settings. Specifically, the configuration should be adjusted to accept certificates only from the system and remove the acceptance of certificates from user sources. This measure is intended to prevent interception and eavesdropping through the use of user-provided certificates that are not validated by the system.

REFERENCES

[1] Statista Research Department, "Most popular smartphone activities for global users 2022-2023," Statista. Accessed: Nov. 01, 2023. [Online]. Available: https://www.statista.com/statistics/1337895/top-smartphone-activities/

[2] Check Point Research Team, "Average Weekly Global Cyberattacks peak with the highest number in 2 years, marking an 8% growth year over year, according to Check Point Research," Check Point. Accessed: Jun. 15, 2024. [Online]. Available: https://blog.checkpoint.com/security/average-weekly-global-cyberattacks-peak-with-the-highest-number-in-2-years-marking-an-8-growth-year-over-year-according-to-check-point-research/

[3] Positive technologies, "Vulnerabilities and threats in mobile applications, 2019," positive technologies. Accessed: Nov. 07, 2023. [Online]. Available: https://www.ptsecurity.com/ww-en/analytics/mobile-application-security-threats-and-vulnerabilities-2019/

[4] Statista Research Department, "Mobile OS Market share worldwide 2009-2023," Statista. Accessed: Dec. 06, 2023. [Online]. Available: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/

[5] Android, "Android Enterprise Security White Paper," Jan. 2019. Accessed: Dec. 16, 2023. [Online]. Available: https://static.googleusercontent.com/media/www.android.com/en//static/2016/pdfs/enterprise/Android_Enterprise_Security_White_Paper_2019.pdf

[6] M. R. Islam and T. A. Mazumder, "Mobile Application and Its Global Impact," *International Journal of Engineering & Technology IJET-IJENS*, vol. 10, pp. 72–78, 2010.

[7] Z. Muhammad, Z. Anwar, A. R. Javed, B. Saleem, S. Abbas, and T. R. Gadekallu, "Smartphone Security and Privacy: A Survey on APTs, Sensor-Based Attacks, Side-Channel Attacks, Google Play Attacks, and Defenses," *Technologies (Basel)*, vol. 11, no. 3, p. 76, Jun. 2023, doi: 10.3390/technologies11030076.

[8] K. U. Sarker, F. Yunus, and A. Deraman, "Penetration Taxonomy: A Systematic Review on the Penetration Process, Framework, Standards, Tools, and Scoring Methods," *Sustainability*, vol. 15, no. 13, p. 10471, Jul. 2023, doi: 10.3390/su151310471.

[9] J. Willemsen, C. Holguera, B. Mueller, S. Schleier, and J. Beckers, *M A S V S Mobile Application Security Verification Standard*, 2.0.0. 2023. [Online]. Available: https://mas.owasp.org

[10] J. Willemsen, C. Holguera, B. Mueller, and S. Schleier, *MASTG Mobile Application Security Testing Guide*, 2.0.0. 2023. [Online]. Available: https://mas.owasp.org/MASTG,

[11] OWASP, "OWASP MAS Checklist," 2023. Accessed: Nov. 15, 2023. [Online]. Available: https://mas.owasp.org/checklists/

[12] E. B. Blancaflor, G. Anne J. Anson, A. Mae V. Encinas, K. C. T. Huplo, M. A. V. Marin, and S. L. G. Zamora, "A Vulnerability Assessment on the Parental Control Mobile Applications' Security: Status based on the OWASP Security Requirements," in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Michigan, USA: IEOM Society International, Mar. 2021. doi: 10.46254/AN11.20211104.

[13] T. H. Chiboora, L. Chacha, T. Byagutangaza, and A. Gueye, "Evaluating Mobile Banking Application Security Posture Using the OWASP's MASVS Framework," in *Proceedings of the 6th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*, New York, NY, USA: ACM, Aug. 2023, pp. 99–106. doi: 10.1145/3588001.3609367.

[14] A. Alanda, D. Satria, H. A. Mooduto, and B. Kurniawan, "Mobile Application Security Penetration Testing Based on OWASP," *IOP Conf Ser Mater Sci Eng*, vol. 846, no. 1, p. 012036, May 2020, doi: 10.1088/1757-899X/846/1/012036.

[15] ISO/IEC, "ISO/IEC 27034 - Information technology — Security techniques — Application security," Switzerland, Nov. 2011.

[16] M. Ogata, J. Franklin, J. Voas, V. Sritapan, and S. Quirolgico, "Vetting the security of mobile applications," Gaithersburg, MD, Apr. 2019. doi: 10.6028/NIST.SP.800-163r1.

[17] NIAP, "Requirements for Vetting Mobile Apps from the Protection Profile for Application Software." Accessed: Nov. 15, 2023. [Online]. Available: https://www.niap-ccevs.org/MMO/PP/394.R/pp_app_v1.2_table-reqs.htm