

Penetration Testing with OWASP Mobile for Android Security Optimization

Deco Aprilliansyah ¹, Imam Riadi ², Sunardi ³

¹ Master Program of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.

² Department of Information System, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.

³ Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.

Email: ¹ deco2007048003@webmail.uad.ac.id

*Corresponding Author

Abstract— Security and privacy are very important on android devices to prevent crimes such as the theft of data and confidential information for users. There are many attack methods that can be carried out by irresponsible parties, one of which is penetration testing. The need to improve the security of android devices from cyber crimes that can occur at any time so that the security and information belonging to users are more secure. Based on this, this study offers how attackers perform penetration testing on targets using android devices using the OWASP Mobile framework based on the steps in the Security Testing Guide (OWASP MSTG) manual. The penetration testing activity is carried out in five steps. Namely, injection of backdoors on the application, finding vulnerabilities, scanning, exploiting and making reports. The results of this study obtained some information on application IOCs and other information in the form of contact data, SMS data, and audio records belonging to the attacked device. Based on this, this research can be used by security parties to patch loopholes in their applications and systems.

Keywords— Android; Penetration Testing; OWASP MSTG; Exploit.

I. INTRODUCTION

The development of smartphone technology is progressing rapidly, and smartphones are used for communication, data exchange, information retrieval, and entertainment, such as video games, online learning, and listening to music[1]-[3]. This makes users dependent on their smartphones. The problem is that this reliance on mobile will never make smartphones free from cybersecurity risks. This development has negative consequences, such as data theft and crime in the form of confidential information for Android device users[4]-[6]. According to Statista, the number of cyberattacks against smartphone users worldwide in December 2021 totaled about 2.2 million attacks[7]. Meanwhile, products from cybersecurity companies detected approximately 3.5 million malicious installation packages on mobile devices in 2021. This is well below the 5.7 million detected the previous year. Note, however, that the

numbers recorded in 2021 are about the same as in 2019[8]. One of the most popular smartphone technologies is Android, the mobile operating system [9]. Android is based on the Linux kernel and was developed by Google. As the single most used OS on smartphones globally, Android is in great demand by bad programmers.

Security on the Android operating system is sometimes overlooked by users, such as malware and remote access exploits by third parties. Mobile app security threats are exploding. Bugs and security vulnerabilities in mobile apps can allow hackers to break in and access sensitive information [10]. Communication security and data protection are very important in application development [11]. For this reason, it is necessary to test the security of the application on an Android smartphone.

Testing is essential to do to know how to use android devices safely. This study experimented with doing penetration testing on Android devices using the OWASP Mobile Security Testing Guide to find out security holes.

The penetration tester attack testing system in this test follows the OWASP Mobile Security Testing Guide with five steps: preparation, intelligence gathering, mapping the application, exploitation, and report. This stage aims to obtain information such as Indicators of Compromise (IOCs), Hash Code, activity system files, signature files, SMS data, contacts data, and audio records [12]. This is important because most data is confidential. That's why safety is important. This testing process is expected to help us understand how third-party penetration testing attacks including the OWASP mobile top 10 issues and enable security parties to patch gaps in applications and systems [13].

II. LITERATURE REVIEW

A. Android OS

Android OS is the most pervasive, widespread, and easy-to-use mobile platform. This open-source Linux kernel-based operating system offers greater flexibility due to its customization properties, making it the leading mobile operating system [14]. An Android utility commonly consists of numerous utility components, such



B. Penetration Testing on Android OS

C. OWASP Mobile Security Testing Guide (MSTG)

D. Attack Scenarios

The diagram illustrates the Android malware attack cycle, showing the interaction between an Attacker, a Victim Application, and a Victim's Mobile Device. The cycle consists of the following steps:

- 1. Find Application:** The Attacker identifies a target application (APK).
- 2. Prepare Malware:** The Attacker prepares the malware, which is then injected into the application.
- 3. Distribute Malware:** The malware is distributed to the victim's mobile device.
- 4. Exploit Vulnerability:** The malware exploits a vulnerability in the victim's mobile device.
- 5. Escalate Privileges:** The malware escalates its privileges on the victim's mobile device.
- 6. Stealing & Manual Exploitation:** The malware steals data and performs manual exploitation.
- 7. Report:** The Attacker receives a report from the victim's mobile device.
- 8. Receive Data:** The Attacker receives data from the victim's mobile device.
- 9. Send Command:** The Attacker sends a command to the victim's mobile device.
- 10. Exploit Target Device:** The Attacker exploits the target device.

It can be seen in Fig. 2. that the steps taken include 5 steps. Namely, the first step of preparation is done by injecting the backdoor and installing the application, the second step of

III. RESULT AND DISCUSSION

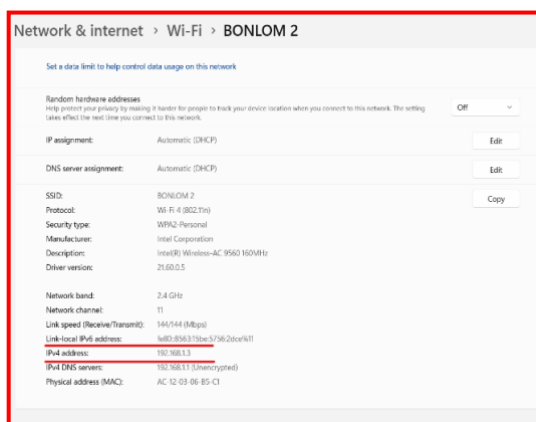
A typical security test is structured as follows:

- ### B. Inject Application

At this stage, the tester will enter the payload into the

The screenshot shows the L3MON APK Builder web interface. The header includes the L3MON logo and navigation links for Devices, APK Builder, and Recent Log. The main content area is titled 'APK Builder' and features a form with a 'http://' dropdown, the IP address '192.168.1.3', a port dropdown set to '22222', and a green 'Build' button.

Fig. 3. shows the IP used by the attacker is 192.168.1.3, and the PORT used is 22222. This IP and PORT follow the attacker's HOST, which can be seen in Fig. 4.



It can be seen in Fig. 4. that the IP Address information on the attacker's device is 192.168.1.3.

This stage shows the analysis of the scope of the application that has been injected with the backdoor. Analysis of gathering information can be seen by identifying the permissions requested by the application. The list of permissions on the application can be seen in Fig. 4. It can be seen in Fig. 4. application permission requests in the form of Wake lock, Camera, Write & read external storage, Internet, Access network state, Read SMS, Send SMS, Access wifi state, Read phone state, Read call log, Record Audio, Access location, Read contacts.

```

<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_CALL_LOG" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.READ_CONTACTS" />

```

Malicious applications can use this to send and receive data and other important information by using this permission.

The results of the manual mapping application found the host entry point contained in the `MyReceiver.java` file, which can be seen in Fig. 5.

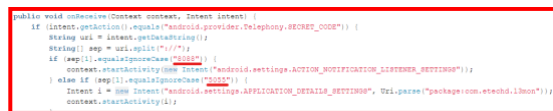


Fig. 6. shows the code listener the incoming port used, namely "8088" and "5055".

The victim's device that has been connected to the attacker's HOST will then be subjected to the Exploitation process, which can be seen in Fig. 6.



It can be seen in Fig. 6. that the attacker's IP is 192.168.1.3, and the PORT is 22222. The victim's device will be connected via the IP Address and PORT. Next, the exploitation process will be carried out.

This stage is a phase where the examiner reports on the attack process and the information that has been obtained during the attack process. The following in Table 1 is some information on the analysis application that has been included in the payload.

Table. 1. IOCs Application

<i>POVERTY</i>	<i>VALUE</i>
<i>MD5</i>	623614337e89113b3ebdc0325bdc0918
<i>SHA1</i>	f7b476070905f35986cdb5764026150d61ab76cb
<i>SHA256</i>	cbea811972b9b87de06eb6c2ea2e291c2981be5fe4c73589352aef4dd0a08f25
<i>Service NAME</i>	com.eteched.l3mon.MainService

Table. 1. shows the information on giving signatures information in the form of hash code md5, sha1, sha256, service name, signature file “rsassa_pkcs1v15,” and file type “apk.” Based on data, files can be retrieved IOCs SHA-256bea811972b9b87de06eb6c2ea2e291c2981be5fe4c73589352aef4dd0a08f25.

After successfully exploitation, the attacker managed to get some data between getting contact, Read SMS, and Record audio which can be seen in Figs 8.



Fig 8 attacker who managed to get contact

Fig. 8. Shows an attacker who managed to get contact information data from the victim's device. The contact data obtained are `***27663368378***`, `08080808` (Exploit 1), `22222` (Exploit 2).



Fig. 9. Dumping data SMS

Fig. 9. shows that the attacker managed to get 3 SMS data from the victim's device.



Fig. 10. Record audio from target device

Fig. 10. shows the attacker managed to record audio from the target device. The audio file was recorded on 28/09/22 with the file name “sound2107994325983723535.mp3.”

IV. CONCLUSION

Penetration testing on targets using android devices using the OWASP Mobile framework based on the steps in the Security Testing Guide (OWASP MSTG) manual. The penetration testing activity is carried out in five steps. Namely, injection of backdoors on the application, finding vulnerabilities, scanning, exploiting and making reports. The results of this study obtained some information on application IOCs and other information in the form of contact data, SMS data, and audio records belonging to the attacked device. Based on this, this research can be used by security parties to patch loopholes in their applications and systems.

REFERENCE

- [1] T. T. Allen, Introduction to discrete event simulation and agent-based modeling: Voting systems, health care, military, and manufacturing, Springer, New York, 2011.
- [2] T. T. Ballen, Introduction to discrete event simulation and agent-based modeling: voting systems, health care, military, and manufacturing, 2 ed., Springer, New York, 2011
- [3] G. Blanchard and R. Loubere, High-order conservative remapping with a posteriori MOOD stabilization on polygonal meshes, 2015, Available from: <http://www.emn.fr/z-info/choco-solver/> [last accessed May 2011].
- [4] R. Boggs, J. Bozman, and R. Perry, Reducing downtime and business loss: addressing business risk with effective technology, Tech. Report Technical report 91-18, InternationalData Corporation (IDC), Sernageomin, 2002.
- [5] S. Elbaum, A. G. Malishevsky, and G. Rothermel, Test case prioritization: A family of empirical studies, IEEE Trans. Softw. Eng. 28 (Feb. 2002), no. 2, 159–182.
- [6] G. Rothermel, A safe efficient regression test selection technique, ACM Trans. Soft. Eng. Methodology 6 (1997), no. 2, 173–210. MR 2000f:91046
- [7] G. Rothermel, M. J. Harrold, C. W. Hirt, A. A. Amsden, and J. L. Cook, A safe efficient regression test selection technique, ACM Trans. Soft. Eng. Methodology 6 (1998), no. 2, 173–210.
- [8] A. Schulz and G. Doblhammer, Aktueller und Zukünftiger Krankenbestand von Demenz in Deutschland auf Basis der Outinedaten der AOK. (Current and future number of people suffering from dementia in Germany based on routine data from the AOK.), Versorgungs-Report (Piscataway, NJ, USA) (C. Gnster, J. Klose, and N. Schmacke. eds.). IEEE Press. 2012. pp. 161–175.

- [9] S. Yoo and M. Harman, Pareto efficient multi-objective test case selection, (Proceedings Of The 2007 International Symposium On Software Testing And Analysis, London, UK), 2007, pp. 140–150.
- [10] R. S. Kusuma, “Forensik Serangan Ransomware Ryuk pada Jaringan Cloud”, JURNAL MULTIMEDIA NETWORKING INFORMATICS, vol. 9, no. 2, pp. 99–107, Oct. 2023.
- [11] K. B. Sarmila and S. V. Manisekaran, "A Study on Security Considerations in IoT Environment and Data Protection Methodologies for Communication in Cloud Computing," *2019 International Carnahan Conference on Security Technology (ICCSST)*, Chennai, India, 2019, pp. 1-6, doi: 10.1109/CCST.2019.8888414.
- [12] S. -, I. Riadi, and P. Ananda, “Vulnerability Analysis of E-voting Application using Open Web Application Security Project (OWASP) Framework,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, 2019, doi: 10.14569/ijacsa.2019.0101118.
- [13] A. Alanda, D. Satria, H. A. Mooduto, and B. Kurniawan, “Mobile Application Security Penetration Testing Based on OWASP,” *IOP Conference Series: Materials Science and Engineering*, vol. 846, no. 1, p. 012036, May 2020, doi: 10.1088/1757-899x/846/1/012036.
- [14] R. Ponakala and M. N. Dailey, “LineageOS Android Open Source Mobile Operating System: Strengths And Challenges,” Oct. 2020, doi: 10.35543/osf.io/4gch5.
- [15] X. Li, L. Yu, and X. P. Luo, “On Discovering Vulnerabilities in Android Applications,” *Mobile Security and Privacy*, pp. 155–166, 2017, doi: 10.1016/b978-0-12-804629-6.00007-9.
- [16] “A Process of Penetration Testing Using Various Tools,” *Mesopotamian Journal of Cyber Security*, pp. 94–104, Apr. 2023, doi: 10.58496/mjcs/2023/014.