SAGA: Journal of Technology and Information Systems
Vol 4, Issue 1, February 2026, Page 585-590
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.v4i1.608

Implementation of Metasploit Framework on Android Platform in a Single Network

Muhammad Fitra Gemilang, Arif Senja Fitrani*, Mochamad Alfan Rosid, S Sumarno

Department of Computer Science, Universitas Muhammadiyah Sidoarjo, Sidoarjo, Indonesia

*Corresponding Email: arifsenjafitrani@umsida.ac.id

Abstract. This study discusses the implementation of the Metasploit Framework on the Android platform within a single network. The objective of this research is to analyze the vulnerability of Android devices to backdoor attacks using penetration testing tools. The payload in the form of an APK file was created using msfvenom and tested on Android devices within a local network. The results showed successful exploitation, including access to SMS, call logs, and device location. This study highlights the need for security awareness among Android users and the importance of preventive measures.

Keywords: Metasploit, Android, Penetration Testing, Backdoor, Payload

1 Introduction

The rapid development of information technology has driven the increased use of Android-based devices in various aspects of life, ranging from communication, digital transactions, to smart device control systems[1] . Android, as an open source operating system, provides high flexibility for developers and device manufacturers[2] . However, the openness of this system also makes it an easy target for various types of cyber attacks that exploit application and system security vulnerabilities[3] .

According to reports from Symantec and the National Cyber and Crypto Agency, Indonesia ranks high in the number of cyber attack incidents in the Southeast Asia region, including attacks involving backdoor techniques[3] . Backdoors are a method of attack that gives attackers hidden access to control the system without the user's knowledge, often through modified or compromised applications[4] . This technique is dangerous because it can evade detection and provide full control over the victim's device[5] .

One of the frameworks commonly used in penetration testing and Android system exploitation is the Metasploit Framework[6] . Metasploit is a tool in the field of ethical hacking that supports various attack techniques, including the creation and insertion of payloads into target devices[7] . With this tool, attackers can gain system access, download data, or activate microphones and cameras secretly. Knowledge of how these attacks work is crucial for improving cybersecurity resilience and user awareness in managing device security[8] .

Previous studies have explored the use of Metasploit for simulating attacks on Android systems[9] . The research by Nugroho et al.[10] shows the effectiveness of reverse TCP payload techniques in remotely accessing Android systems, while the study by Pratiwi and Miarsa highlights the importance of user education in detecting malicious applications[11] . However, further research is needed that focuses on mapping the level of access obtained through backdoor attacks and the mitigation measures that can be taken.

This study aims to determine the process of exploiting Android devices using the Metasploit Framework, describe the extent of access that can be obtained through backdoors, and raise awareness of the importance of securing Android devices from unauthorized access. This study is expected to contribute to the development of prevention and early detection strategies against cyber threats targeting the Android operating system.

2 Method

This research uses an experimental approach with a case study method on Android devices located in a local network[12] . The purpose of this method is to simulate the *backdoor* attack process using *the Metasploit Framework* in order to directly observe the stages of exploitation and the level of access that can be obtained by attackers.

SAGA: Journal of Technology and Information Systems
Vol 4, Issue 1, February 2026, Page 585-590
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.v4i1.608

2.1 Devices and Tools

The research was conducted using one computer device as the attacker and one Android device as the target/victim. The attacker's computer ran the Kali Linux operating system with *Metasploit Framework* and *msfvenom* installed, while the target device used the Android 8.0 (Oreo) operating system or higher. The connection between the two devices was made via a local Wi-Fi network (*Local Area Network*).

2.2 Experimental Steps

The stages of the experiment in this study included:

a. Payload Creation
   Researchers used the msfvenom command to create a payload of type android/meterpreter/reverse_tcp. The payload was packaged in .apk (Android Package) file format and disguised to resemble a legitimate application, so that it could be run on the target device without arousing suspicion[13] ..
b. Payload Distribution
   The compiled APK file is manually distributed to Android devices for simulation purposes (not through the Play Store or social media, in order to maintain experimental ethics).
c. Payload Injection and Execution
   After the application is installed and run on the Android device, *the Metasploit Handler* will listen for connections from the target device. If successful, a Meterpreter session will open, indicating that the device has been successfully exploited[14] .
d. Observation and Documentation of Access
   Researchers observed features that can be remotely controlled via the Meterpreter session, such as taking photos, reading files, accessing location data, recording audio, and opening the camera.
e. Security Analysis and Potential Vulnerabilities
   The results of the exploitation are then analyzed to identify potential security risks and the extent of control that attackers can gain through this *backdoor*.[15]

2.3 Research Ethics and Security

The entire experimental process was conducted in a closed and controlled environment, without involving third-party devices or other individuals. This research is purely educational and aims to raise awareness of the importance of Android device security. No personal data was misused during the research process.

3 Results and Discussion

This section will explain in detail the results of the research, starting from the payload creation process to testing on the target device. Each step is described with attention to the configuration, tools used, and results obtained at each stage. The process begins with the creation of a payload as the main component used to establish a connection between the attacker's device and the target device.

3.1 Payload Creation

The payload was created using the command msfvenom -p android/meterpreter/reverse_tcp LHOST=172.20.10.3 LPORT=4444 -o tess.apk, as shown in Figure 1. This command generates the tess.apk file containing the payload with the local host address (LHOST) 172.20.10.3 and port (LPORT) 4444 configured as the reverse communication path to the attacker's device. The command generates a *tess.apk* file containing a payload with a size of 10,237 bytes, which is later used to exploit the target Android device through *a reverse connection*.

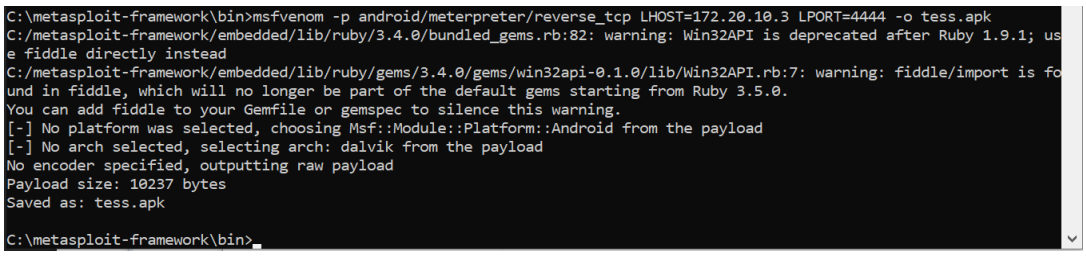

Figure 1. Payload creation

SAGA: Journal of Technology and Information Systems
Vol 4, Issue 1, February 2026, Page 585-590
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.v4i1.608

3.2 Installation on Android Devices

The APK file is sent to the Android device via a USB cable connection because this method offers higher transfer speeds, more stable connections, and a lower risk of detection compared to wireless connections. Testing is performed by running the application on the Android device and monitoring the Meterpreter connection session via the attacker's laptop, as shown in Figure 2.

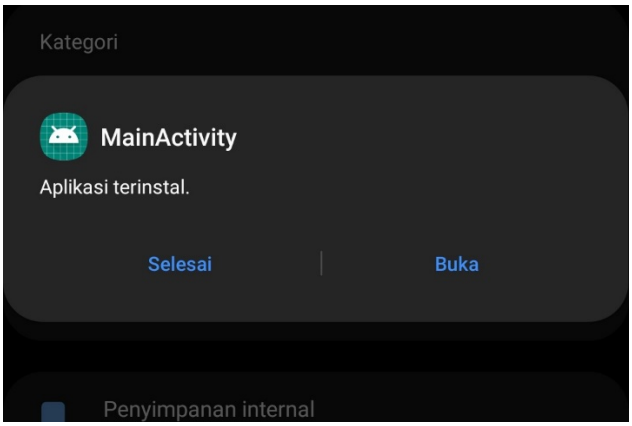

Figure 2. APK installed

3.3 Create a Backdoor APK with Msfvenom

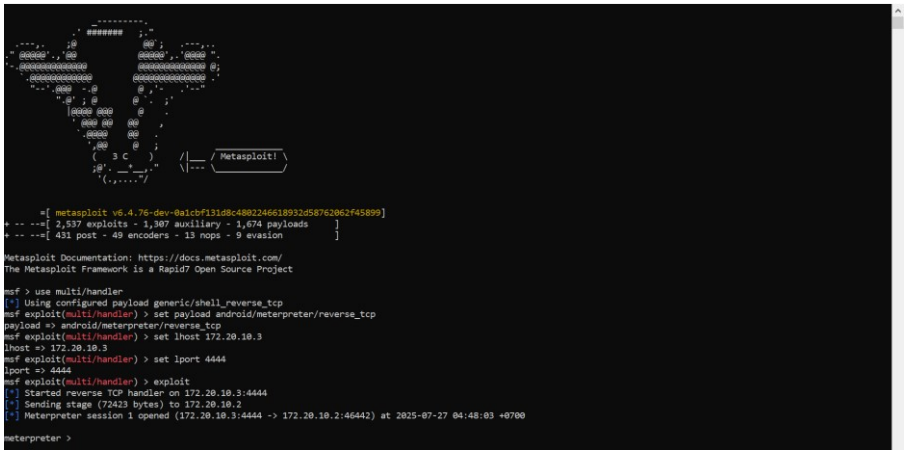

Figure 3. Msfconsole Display

Msfconsole is a tool used to run Meterpreter and execute previously created payloads. This process begins with the use multi/handler command to prepare the listener, followed by configuring the Local Host (LHOST) and Local Port (LPORT) parameters. Once the configuration is complete, the exploit command is run to wait for a connection from the target device, as shown in Figure 3.

3.4 Exploitation Commands

Exploitation can be carried out by running a number of specific commands through the Meterpreter session. Some of these are sysinfo, which is used to display system information such as the Android version and CPU architecture; dump_sms to retrieve SMS message data from the device; dump_calllog to obtain call history; app_list to display all applications installed on the device; and geolocate, which is used to retrieve GPS coordinates from the target device's location.

3.5 Visualization of Results

Figure 4 shows the output of the sysinfo command in the Metasploit Meterpreter environment. The information displayed describes the system details of the Android device that was successfully accessed. The device has the hostname localhost and runs the Android 11 operating system on top of Linux kernel version 4.14.113 with the

SAGA: Journal of Technology and Information Systems
Vol 4, Issue 1, February 2026, Page 585-590
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.v4i1.608

name Lengyue-beyond0lte-Kernel-V7. The processor architecture used is aarch64, indicating that the device is based on ARM 64-bit. The detected system language is in_ID, indicating the use of Indonesian. The active Meterpreter type is dalvik/android, confirming that this exploitation session is connected to an Android device.

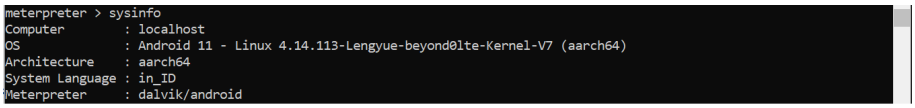

Figure 4. Sysinfo

Figure 5 shows the interaction within the Meterpreter environment, which is an advanced payload from the Metasploit Framework used to perform post-exploitation activities on a compromised system. In this scenario, the dump_sms command was successfully executed and extracted 1,740 SMS messages from the target device. The data was then automatically saved to the file sms_dump_20250726234928.txt. However, after the data retrieval process was complete, the Meterpreter session that was previously connected to the IP address 172.20.10.2 was suddenly disconnected with the status "Reason: Died". This indicates that the exploitation session ended unexpectedly, likely due to network connection issues, a crash in the Meterpreter agent on the target device, or the device being forcibly shut down.

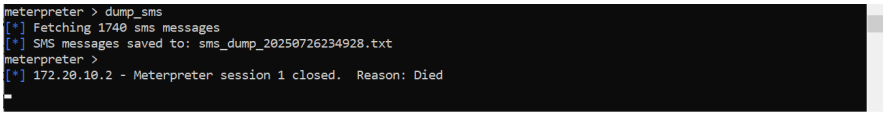

Figure 5. Dump_sms

Figure 6 shows one of the operations in the Meterpreter session, where the dump_calllog command was successfully executed to retrieve the call history from the target device. The results of this process included 315 entries or call records that were successfully extracted. After the retrieval process was complete, all of the data was saved to a text file named calllog_dump_20250727081520.txt. The file naming format is believed to include information on the date and time when the dumping process was performed, in order to facilitate the identification of extraction results based on the execution time.

Figure 6. Dump_callog

Figure 7 shows the results of executing the app_list command in a Meterpreter session, which is used to obtain a complete list of applications installed on the target Android device. Each entry in the list includes several important columns of information, namely Name (the application's display name), Package (the unique package name that identifies the application), Running (the status of whether the application is running, marked as "true" or "false"), and IsSystem (an indication of whether the application is part of the operating system). From the data displayed, we can observe the diversity of installed applications, ranging from utility applications such as "1.1.1.1" (which is likely a VPN application from Cloudflare) and "3 Button Navigation Bar", to Android system services such as "ANT+ Plugins Service", "Android System Intelligence", and manufacturer-built-in applications such as "AlwaysOnDisplay" and "MTP Application".. This information provides a comprehensive overview of the application ecosystem running on the successfully exploited device.

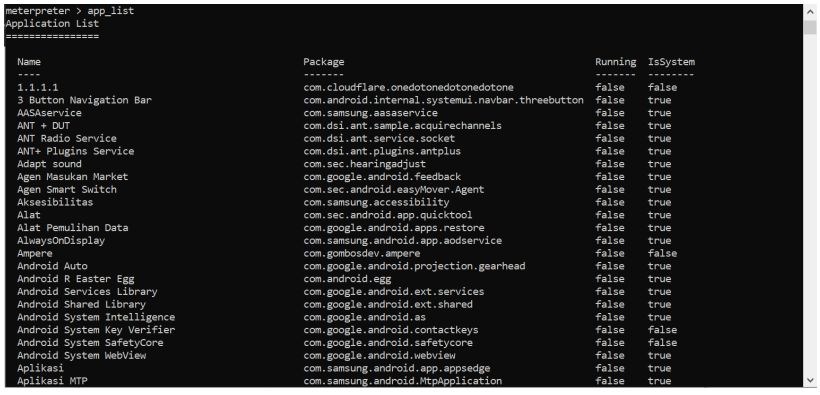

Figure 7. App_list

SAGA: Journal of Technology and Information Systems
Vol 4, Issue 1, February 2026, Page 585-590
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.v4i1.608

Figure 8 shows the output from an unusual Meterpreter session, where the text appears upside down and difficult to read. This indicates a possible disruption in the terminal display orientation or the use of non-standard character encoding. However, after manually reading the text backwards, several elements can be identified, such as the phrase "current location" and the geolocate command, which indicate that the user has executed the geolocation feature in Meterpreter to obtain the geographical coordinates of the target device. Although the results displayed are not immediately clear, the context of the command still describes an attempt to retrieve location data from the compromised system.

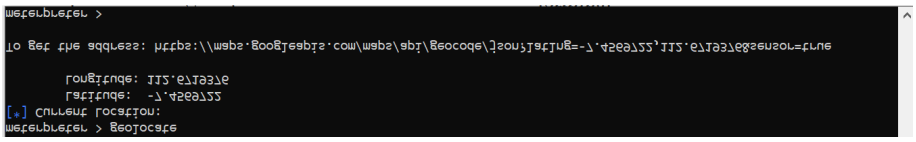

Figure 8. Geolocate

4 Conclusion

This research shows that the Android operating system, despite being open and flexible, is highly vulnerable to cyber attacks that utilize backdoor techniques. By utilizing the Metasploit Framework and the android/meterpreter/reverse_tcp payload, attackers can gain full access to the target device. The experiments conducted successfully demonstrated how a malicious application can be created, disguised, installed, and executed to open a Meterpreter session that allows the retrieval of sensitive data such as SMS messages, call history, application lists, and even the geographical location of the device.

5 Acknowledgments

We would like to express our gratitude to Mr. Arif Senja Fitrani, M.Kom., as the supervising lecturer, as well as to the Informatics Laboratory of Muhammadiyah University Sidoarjo for the facilities provided during this research process. This research is aware that it still has a number of shortcomings. Therefore, feedback in the form of criticism and suggestions is highly appreciated so that similar research in the future can be conducted more effectively.

References

[1] N. A. Pandusaputri, R. B. Ramadhan Mokodompit, and E. P. Simangunsong, "KENYAMANAN PENGGUNA IOS DAN ANDROID DI KALANGAN GENERASI Z.," *J. Syntax Lit.*, vol. 9, no. 5, 2024, Accessed: Jul. 29, 2025. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=25410849&AN=177805927&h=uksAeuWpjxLKL48E9O%2Fto7SdL8xkVBLco0X5X%2FEOyRvwH5Jruu%2BuqgaDJKekD2JdsFvYpsQDvNgIDykjqXhRGg%3D%3D&crl=c

[2] F. F. Navintino, M. F. Fahreza, and E. Rilvani, "Literature Review: Comparison of Security Systems in Android and iOS Applications," *STORAGE J. Ilm. Tek. Dan Ilmu Komput.*, vol. 4, no. 1, pp. 22–28, 2025.

[3] R. Vice, N. C. Simaremare, and S. Manalu, "The Development of Cybercrime: Its Impact on National Security and Resilience and Its Prevention," *Mimb. KEADILAN J. Ilmu Huk.*, pp. 71–83, 2024.

[4] S. Alhidamkara, S. Somantri, and I. L. Kharisma, "Security Analysis and Exploitation of Android 13 Kernel Using Metasploit Reverse_Tcp," *Kesatria J. Penerapan Sist. Inf. Komput. Dan Manaj.*, vol. 5, no. 3, pp. 845–856, 2024.

[5] A. S. Mubarak, M. N. Insirat, and M. N. Lutfiya, "Ransomware: Evolution, Classification, Attack Phase, Detection and Prevention," in *Proceedings of the National Seminar on Electrical Engineering, Information Systems, and Informatics Engineering (SNESTIK)*, 2024, pp. 1–6. Accessed: Jul. 29, 2025. [Online]. Available: https://ejurnal.itats.ac.id/snestik/article/view/5588

[6] M. R. A. Fitra, A. A. S. Effendi, S. A. Priscilia, and D. Kiswanto, "PENETRATION TESTING USING HYDRA AND METASPLOIT ON THE SECURE SHELL PROTOCOL," *JATI J. Mhs. Tek. Inform.*, vol. 9, no. 1, pp. 1017–1024, 2025.

[7] S. Alhidamkara, S. Somantri, and I. L. Kharisma, "Security Analysis and Exploitation of Android 13 Kernel Using Metasploit Reverse_Tcp," *Kesatria J. Penerapan Sist. Inf. Komput. Dan Manaj.*, vol. 5, no. 3, pp. 845–856, 2024.

[8] M. F. Aska, D. P. Putra, and C. J. M. Sinambela, "Effective Strategies for Cybersecurity Implementation in the Digital Age," *J. Inform. Inf. Secur.*, vol. 5, no. 2, pp. 187–200, 2024.

[9] H. SETIAWAN, “VISUALIZATION OF METASPLOIT TROJAN ATTACKS ON ANDROID USING THE K-MEANS METHOD”, Accessed: Jul. 31, 2025. [Online]. Available: https://repository.unsri.ac.id/146546/1/RAMA_56201_09011182025004_0003047905_01_front_ref.pdf

[10] T. H. F. NUGROHO, “CONGESTION CONTROL OPTIMIZATION IN SOFTWARE-DEFINED NETWORKS (SDN) BASED ON REINFORCEMENT LEARNING,” Accessed: Jul. 29, 2025. [Online]. Available: https://digilib.itb.ac.id/assets/files/2025/MjMyMjIwNzYgVHJpIEhlbmRhcnRvIEZOIERyYWZ0IFRlc2lzIEZpbmFsIC0gVHJpIEhlbmRhcnRvIEZhamFyIE51Z3JvaG8gKDEpLnBkZg.pdf

[11] F. Y. Pratiwi and F. R. D. Miarsa, “The Urgency of Public Education in Countering Personal Data Misuse,” *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 4, no. 2, pp. 1342–1349, 2025.

[12] H. W. Putra and I. G. L. P. E. Prismana, “Application of the Murder Collaborative Learning Model in Improving the Learning Competence of Basic Programming Subjects for Vocational School Students Based on PjBL Case Study: Smks Semen Gresik,” *IT-Edu J. Inf. Technol. Educ.*, vol. 10, no. 02, pp. 55–67, 2025.

[13] W. Darmawan and T. Aprilia, “Analysis of Metasploit Framework ‘msfvenom’ Backdoor Trojan and Fully Undetected (FUD) Trojan,” *Techno Com*, vol. 23, no. 1, 2024, Accessed: Jul. 29, 2025. [Online]. Available: https://core.ac.uk/download/pdf/604543777.pdf

[14] S. Alhidamkara, S. Somantri, and I. L. Kharisma, “Security Analysis and Exploitation of Android 13 Kernel Using Metasploit Reverse_Tcp,” *Kesatria J. Penerapan Sist. Inf. Komput. Dan Manaj.*, vol. 5, no. 3, pp. 845–856, 2024.

[15] G. P. I. Fanani, M. A. Mu'min, and N. Tristanti, "Analysis and Testing of Website Vulnerabilities Using OWASP ZAP," *J. Res. Syst. and Inf. Tech*., vol. 3, no. 1, pp. 36–50, 2025.