ANDROID VULNERABILITIES

VERSION 12 & 13

PRESENTATION BY
M SATYA SAI TEJA
P SAI SIVAKESH

Agenda

Introduction

WHY ANDROID

- Most popular mobile operating system in the world, with over 2 billion active devices

Introduction

WHY ANDROID

- Most popular mobile operating system in the world, with over 2 billion active devices

- Not immune to vulnerabilities

Introduction

WHY ANDROID

- Most popular mobile operating system in the world, with over 2 billion active devices
- Not immune to vulnerabilities
- Attackers exploit vulnerabilities to gain unauthorized access, steal data, or install malware

Introduction

WHY ANDROID

- Most popular mobile operating system in the world, with over 2 billion active devices.
- Not immune to vulnerabilities.
- Attackers exploit vulnerabilities to gain unauthorized access, steal data, or install malware.
- 82% of Android devices were susceptible to at least one out of 25 vulnerabilities in the Android operating system

Introduction

OUR GOALS

- Raise awareness of Android vulnerabilities and how to protect yourself from attacks

Introduction

OUR GOALS

- Raise awareness of Android vulnerabilities and how to protect yourself from attacks
- Discuss the different types of Android vulnerabilities

Introduction

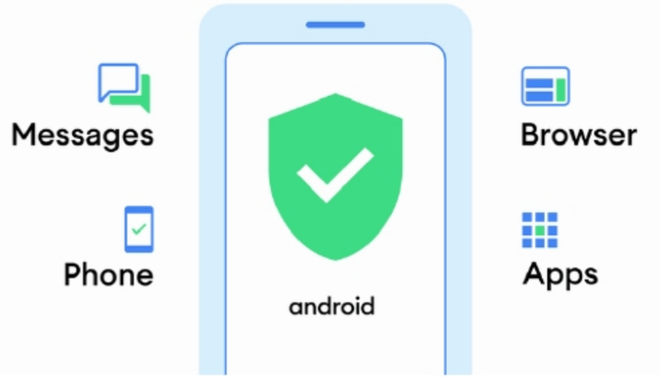
OUR GOALS

- Raise awareness of Android vulnerabilities and how to protect yourself from attacks.
- Discuss the different types of Android vulnerabilities.
- Provide code demos of some of the most critical Android 12 & 13 vulnerabilities.

Introduction

OUR GOALS

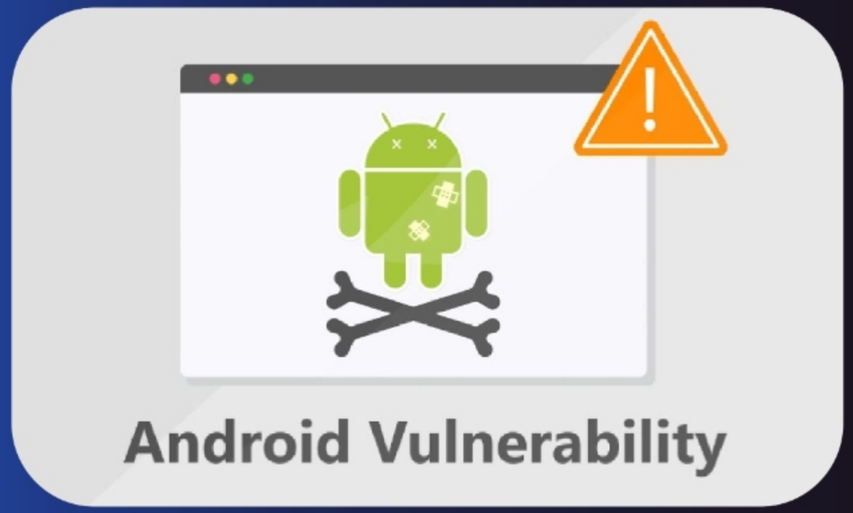
- Raise awareness of Android vulnerabilities and how to protect yourself from attacks.
- Discuss the different types of Android vulnerabilities.
- Provide code demos of some of the most critical Android 12 & 13 vulnerabilities.
- How they can be exploited, and what steps you can take to protect yourself.



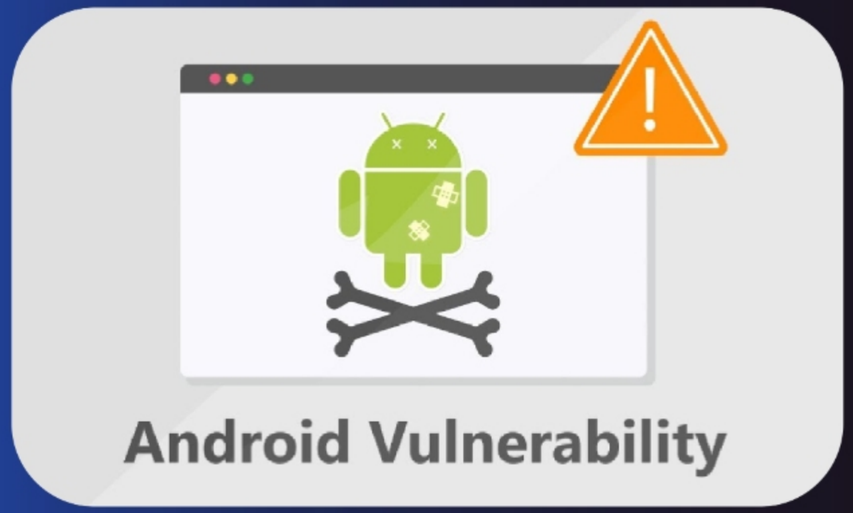
Android Vulnerabilities

WHAT

Security hole or weak spot in the Android operating system (OS)



Android Vulnerabilities



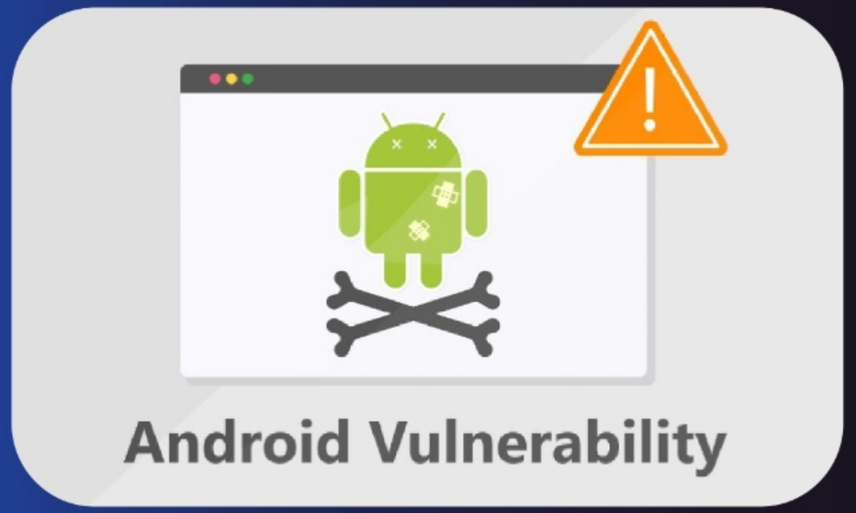
WHAT

Security hole or weak spot in the Android operating system (OS)

WHERE

In the kernel, system libraries, device drivers, and apps, among other parts of the Android OS.

Android Vulnerabilities

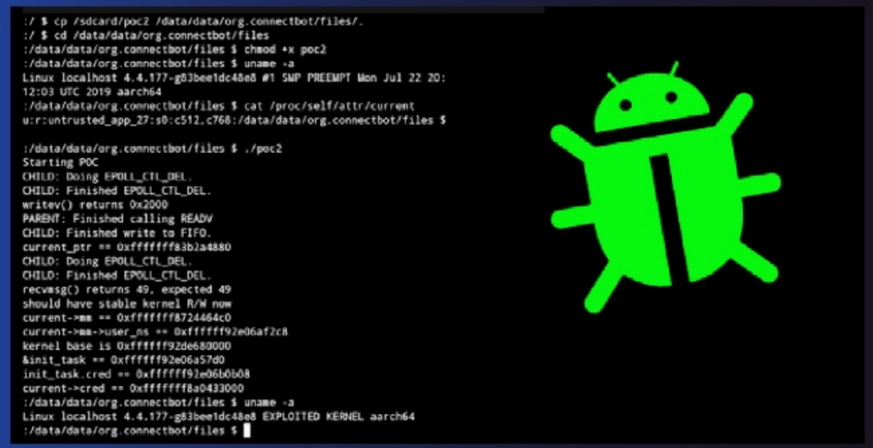


WHAT Security hole or weak spot in the Android operating system (OS)

WHERE In the kernel, system libraries, device drivers, and apps, among other parts of the Android OS.

HOW Programming errors, design faults, or weaknesses in the underlying software or hardware

Android Vulnerabilities



Types of Android Vulnerabilities

1. Remote Code Execution (RCE)

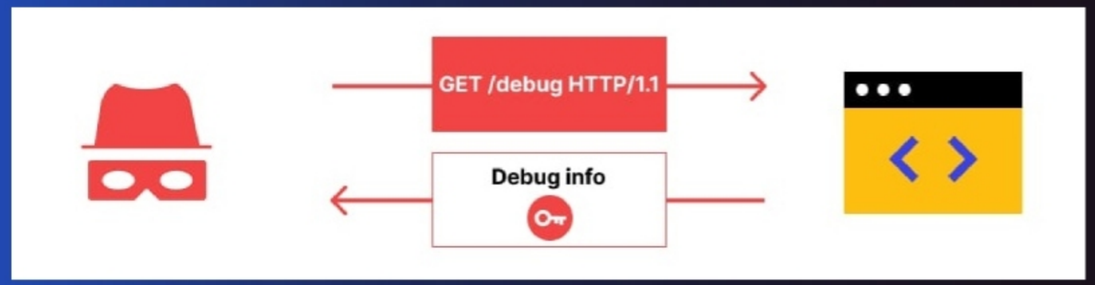
Android Vulnerabilities

Types of Android Vulnerabilities

1. Remote Code Execution (RCE)

2. Escalation Of Privilege (EOP)

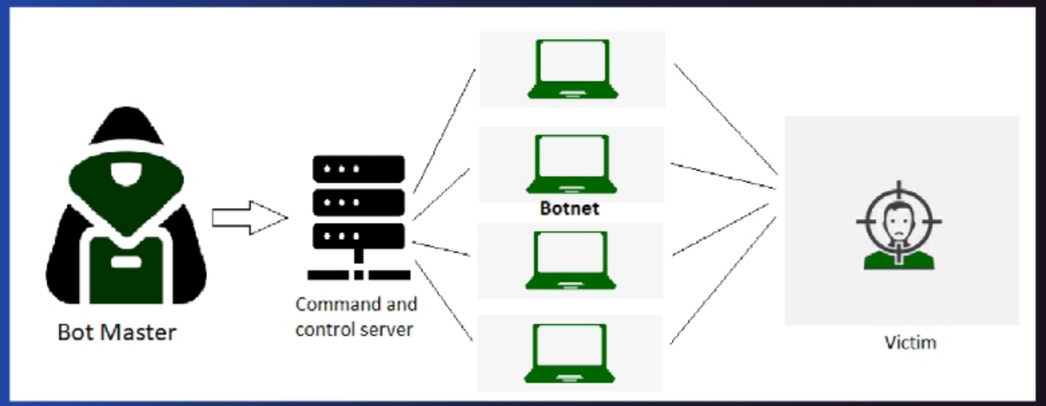
Android Vulnerabilities



Types of Android Vulnerabilities

1. Remote Code Execution (RCE)
2. Escalation Of Privilege (EOP)
3. Information Disclosure (ID)

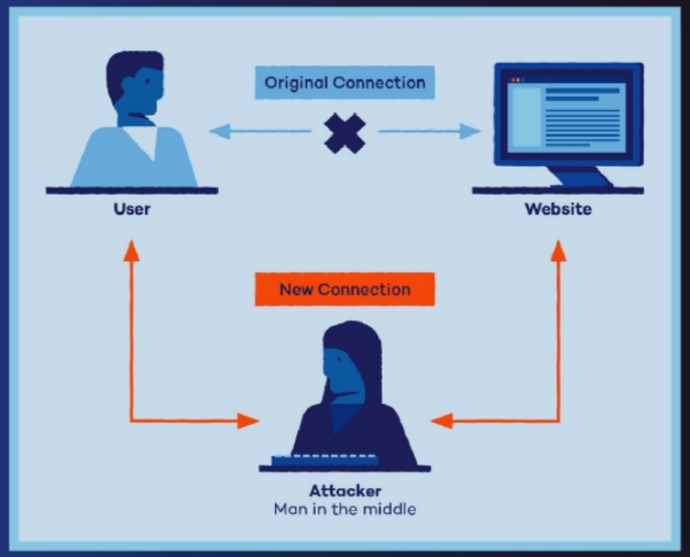
Android Vulnerabilities



Types of Android Vulnerabilities

1. Remote Code Execution (RCE)
2. Escalation Of Privilege (EOP)
3. Information Disclosure (ID)
4. Denial of Service (DoS)

Android Vulnerabilities



Types of Android Vulnerabilities

1. Remote Code Execution (RCE)
2. Escalation Of Privilege (EOP)
3. Information Disclosure (ID)
4. Denial of Service (DoS)
5. Man-in-the-Middle (MitM) Attacks

Most Critical Android Vulnerabilities

- CVE-2023-20951 (RCE, Base Score 9.8, Critical)
- In gatt_process_prep_write_rsp of gatt_cl.cc, there is a possible out of bounds write due to a missing bounds check.

Most Critical Android Vulnerabilities

- CVE-2023-20951 (RCE, Base Score 9.8, Critical)
- In gatt_process_prep_write_rsp of gatt_cl.cc, there is a possible out of bounds write due to a missing bounds check.
- This could lead to remote code execution with no additional execution privileges needed.

Most Critical Android Vulnerabilities

- CVE-2023-20951 (RCE, Base Score 9.8, Critical)
- In gatt_process_prep_write_rsp of gatt_cl.cc, there is a possible out of bounds write due to a missing bounds check.
- This could lead to remote code execution with no additional execution privileges needed.
- User interaction is not needed for exploitation. Product: Android Versions: Android-11 Android-12 Android-12L Android-13 Android ID: A-258652631

Loss of Integrity and Availability

Most Critical Android Vulnerabilities

- CVE-2023-20951 (RCE, Base Score 9.8, Critical)

```
if (len < GATT_PREP_WRITE_RSP_MIN_LEN) {
+   if (len < GATT_PREP_WRITE_RSP_MIN_LEN ||
+       len > GATT_PREP_WRITE_RSP_MIN_LEN + sizeof(value.value)) {
      LOG(ERROR) << "illegal prepare write response length, discard";
      gatt_end_operation(p_clcb, GATT_INVALID_PDU, &value);
      return;
@@ -604,7 +605,7 @@
    STREAM_TO_UINT16(value.handle, p);
    STREAM_TO_UINT16(value.offset, p);

-   value.len = len - 4;
+   value.len = len - GATT_PREP_WRITE_RSP_MIN_LEN;

    memcpy(value.value, p, value.len);
```

CWE - 787

Dirty Pipe Vulnerability

- **CVE-2022-0847 (EOP, Base Score 7.8, High)**
- Linux Kernel Version > 5.8, Android 12
- Arbitrary write read-only files (No depend on any CAPs)
- **Similar as CVE-2016-5195 (Dirty Cow)**
- But more easier to trigger

Gives unprivileged users root access

Android Vulnerabilities

Dirty Pipe Vulnerability

CVE-2022-0847 Detail

Description

A flaw was found in the way the "flags" member of the new pipe buffer structure was lacking proper initialization in copy_page_to_iter_pipe and push_pipe functions in the Linux kernel and could thus contain stale values. An unprivileged local user could use this flaw to write to pages in the page cache backed by read only files and as such escalate their privileges on the system.

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 7.8 HIGH **Vector:** CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

QUICK INFO

CVE Dictionary Entry:
CVE-2022-0847
NVD Published Date:
03/10/2022
NVD Last Modified:
12/09/2022
Source:
Red Hat, Inc.

Dirty Pipe Vulnerability

- How the dirty pipe vulnerability can be exploited :
- Injecting code into a process.
- Overwriting data in a file.

Gives unprivileged users root access

Dirty Pipe Vulnerability

- The PIPE_BUF_FLAG_CAN_MERGE flag was added to Linux pipes in kernel version 5.8, released in May 2020.
- This flag enables the kernel to merge pages from multiple pipes, leading to improved performance.
- This flag introduced the "dirty pipe" vulnerability.

Gives unprivileged users root access

Dirty Pipe Vulnerability

- This vulnerability enables a user with write access to a pipe to overwrite data in any user-readable file.
- This can be done by creating a pipe with the PIPE_BUF_FLAG_CAN_MERGE flag.
- Then the splice() system call is used to merge pages from the pipe into the target file.

Gives unprivileged users root access

Dirty Pipe Vulnerability

- Suppose if we have a file /etc/passwd that contains the user account information for a system.
- This file is owned by root and has permissions of 644 (only root can write to the file, but all users can read it)
- Now, if we have a user named Bob who does not have root privileges. Bob can still create a pipe with the PIPE_BUF_FLAG_CAN_MERGE flag.
- He can then use the splice() system call to merge pages from the pipe into the /etc/passwd file.

Dirty Pipe Vulnerability

- If Bob is able to merge a page from the pipe into the /etc/passwd file, he can overwrite any data in the file.
- This means that he could change his own user account information, or he could change the user account information for any other user on the system.

Dirty Pipe Vulnerability

- The dirty pipe vulnerability was patched in kernel version 5.16, released in May 2022.
- However, systems running older versions of the kernel are still vulnerable.
- One of the major limitations of this exploit is that the attacker must have at least read permission to the file.

Gives unprivileged users root access

Dirty Pipe Vulnerability

- Some ways to mitigate the dirty pipe vulnerability are :
- Use a firewall to block unauthorized access to your system.
- Use a security solution that can detect and block exploitation of the dirty pipe vulnerability.
- Be careful about what you run on your system.
- Upgrade to a kernel version that has the vulnerability patched.

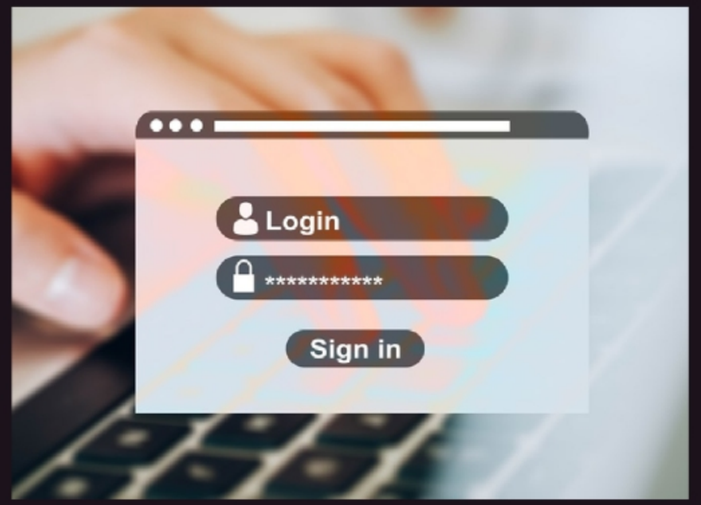
Dirty Pipe Vulnerability

Lets look at a

DEMO

Set Secure Passwords

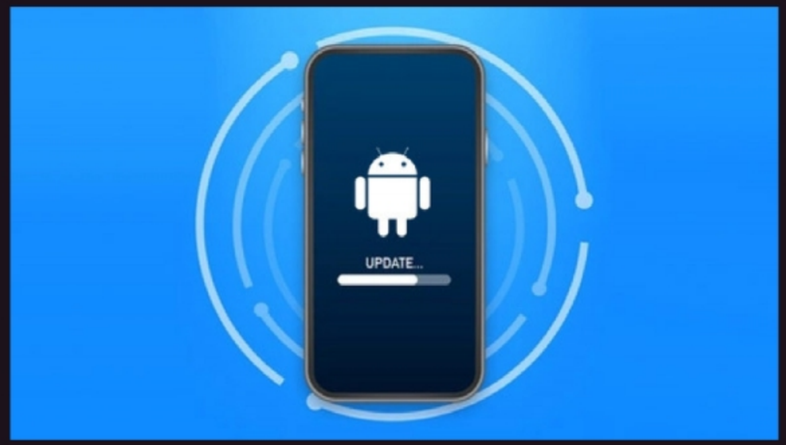
Measures to protect Android devices from vulnerabilities



Measures to protect Android devices from vulnerabilities

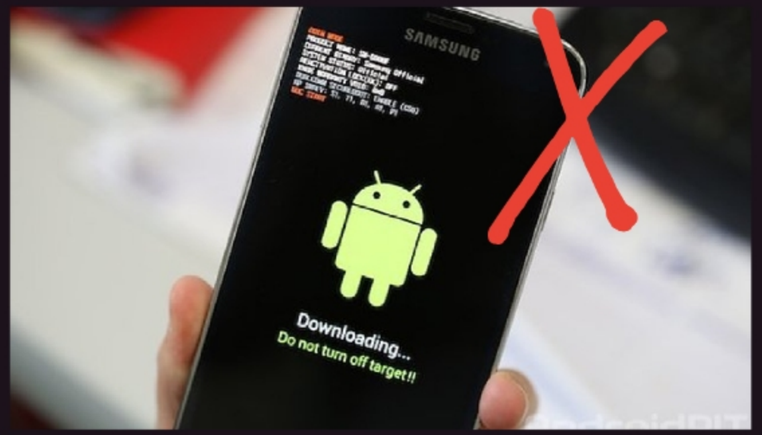
Set Secure Passwords

Keep your device's OS Up-To-Date



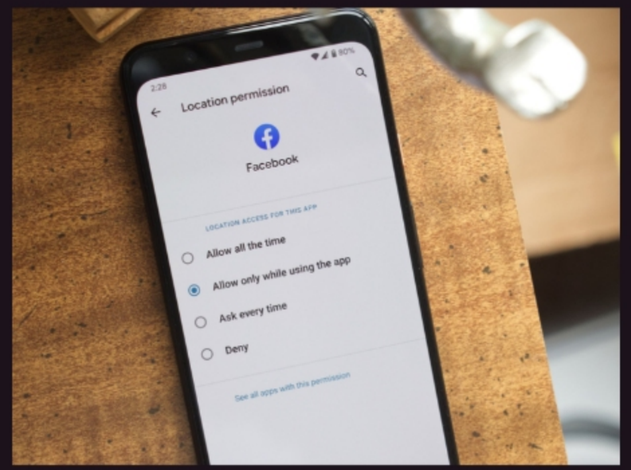
Measures to protect Android devices from vulnerabilities

- Set Secure Passwords
- Keep your device's OS Up-To-Date
- Don't Jailbreak or Root your phone



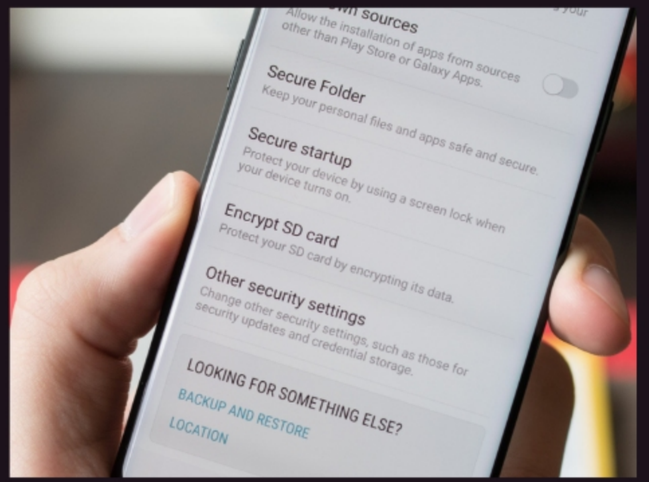
Measures to protect Android devices from vulnerabilities

- Set Secure Passwords
- Keep your device's OS Up-To-Date
- Don't Jailbreak or Root your phone
- Be cautious with app permissions



Measures to protect Android devices from vulnerabilities

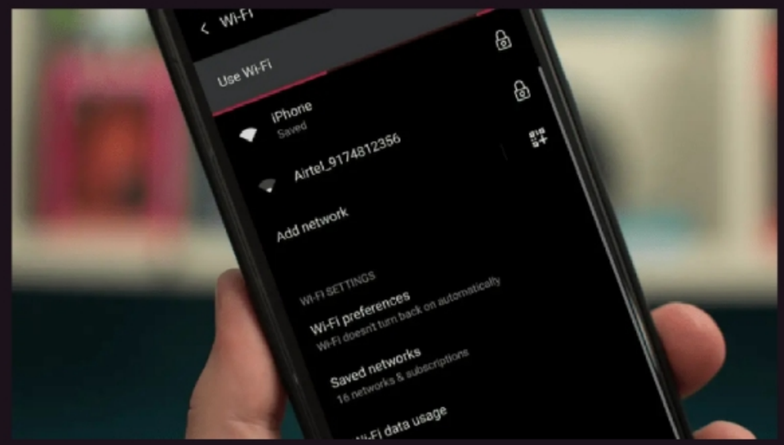
- Set Secure Passwords
- Keep your device's OS Up-To-Date
- Don't Jailbreak or Root your phone
- Be cautious with app permissions
- Enable app and device encryption



Android Vulnerabilities

Measures to protect Android devices from vulnerabilities

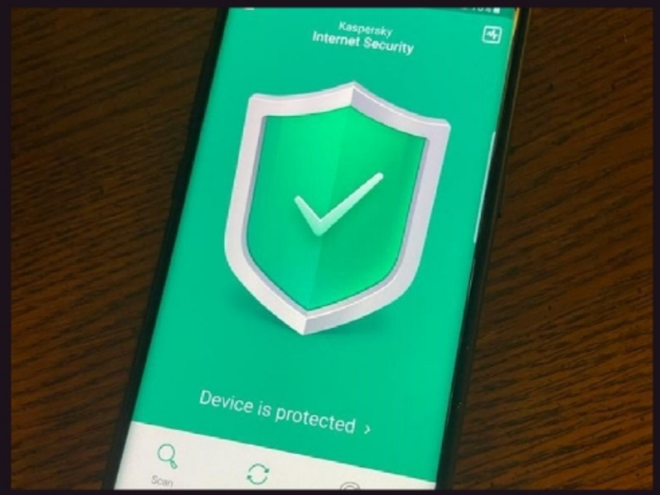
- Set Secure Passwords
- Keep your device’s OS Up-To-Date
- Don’t Jailbreak or Root your phone
- Be cautious with app permissions
- Enable app and device encryption
- Connect to Secure Wifi



Android Vulnerabilities

Measures to protect Android devices from vulnerabilities

- Set Secure Passwords
- Keep your device's OS Up-To-Date
- Don't Jailbreak or Root your phone
- Be cautious with app permissions
- Enable app and device encryption
- Connect to Secure Wifi
- Install Anti-Virus Software



Android Vulnerabilities

Measures to protect Android devices from vulnerabilities

- Set Secure Passwords
- Keep your device's OS Up-To-Date
- Don't Jailbreak or Root your phone
- Be cautious with app permissions
- Enable app and device encryption
- Connect to Secure Wifi
- Install Anti-Virus Software
- Stay cautious of phishing attacks

Steps to be taken if an Android device has been compromised

1. Disconnect from the internet

Steps to be taken if an Android device has been compromised

1. Disconnect from the internet

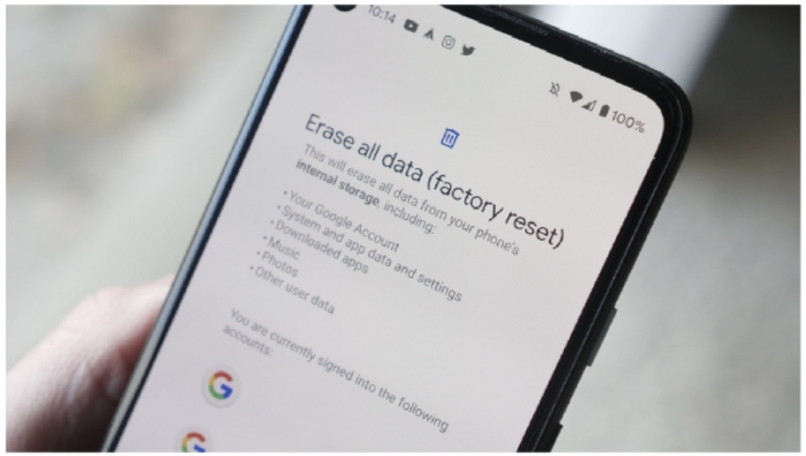
2. Remove suspicious apps

Steps to be taken if an Android device has been compromised

1. Disconnect from the internet

2. Remove suspicious apps

3. Backup and factory reset



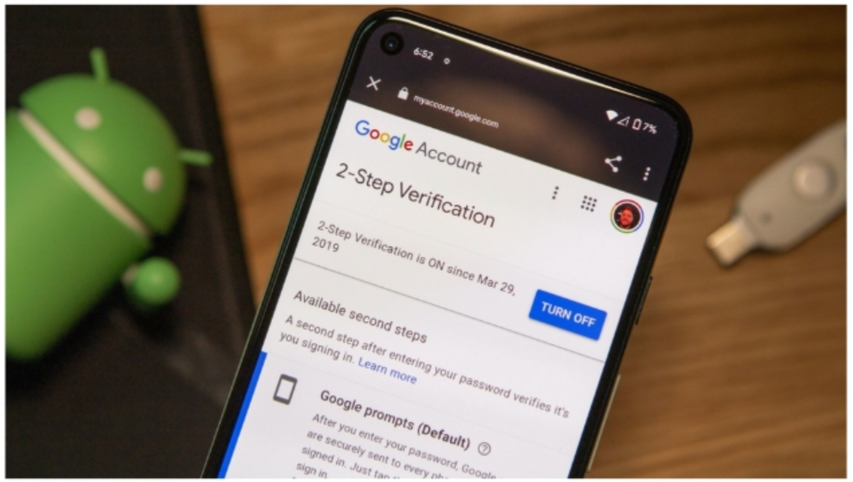
Steps to be taken if an Android device has been compromised

1. Disconnect from the internet

2. Remove suspicious apps

3. Backup and factory reset

4. Enable two-factor authentication (2FA)



Steps to be taken if an Android device has been compromised

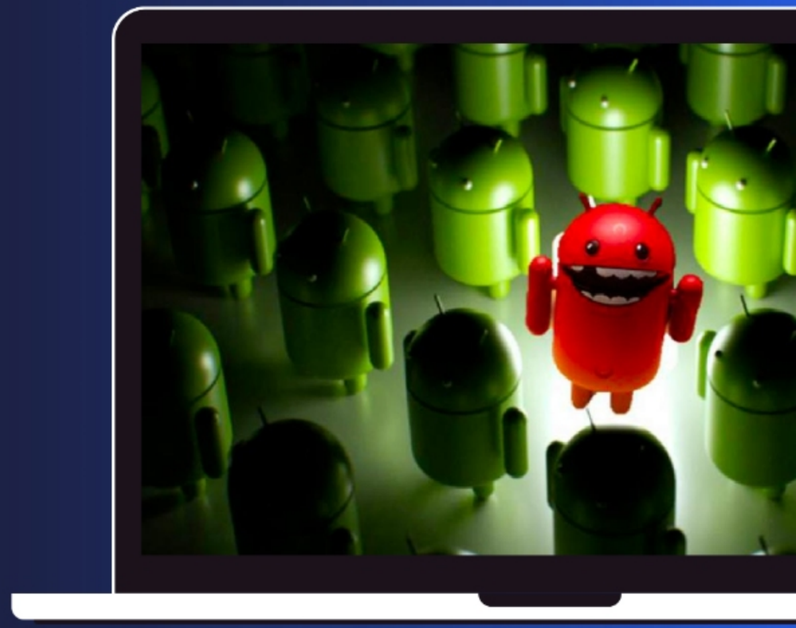
1. Disconnect from the internet
2. Remove suspicious apps
3. Backup and factory reset
4. Enable two-factor authentication (2FA)
5. Be cautious of unknown sources

Steps to be taken if an Android device has been compromised

1. Disconnect from the internet
2. Remove suspicious apps
3. Backup and factory reset
4. Enable two-factor authentication (2FA)
5. Be cautious of unknown sources
6. Install a reputable antivirus app

References

- Android Security Documentation
- NVD (National Vulnerability Database)
- NIST (National Institute of Standards and Technology)
- CWE (Common Weakness Enumeration) Community
- (Dirty Pipe) Max Kellerman's Blog



Thank you