

## Question Answer

What is CV CVE-2023-36621 refers to a vulnerability in the Boomerang Parental Control application versions up to 1.0.0. How severe CVE-2023-36621 is rated as having a base score of 9.1, which classifies it as CRITICAL in terms of severity. What is the CVE-2023-36621 was published on 03 November 2023.

Where can Additional details about CVE-2023-36621 can be found in several references, including a Full Disclosure Project article. What is the CVE-2023-36621 allows a child to exploit the Boomerang Parental Control application by entering Safe Mode. Can you provide A potential attack scenario involving CVE-2023-36621 might entail a child who is aware of the vulnerability. What is CV CVE-2023-36620 is a security vulnerability identified in the Boomerang Parental Control application, specifically in the Android version. How would The security implications of CVE-2023-36620 include unauthorized access and control over the affected device. What is the CVSS base score of CVE-2023-36620 is 4.6, which is categorized as 'MEDIUM'. This score indicates the severity level. When was CVE-2023-36620 was published on 03 November 2023.

What version CVE-2023-36620 affects all versions of the Boomerang Parental Control Android app prior to version 1.0.0. What mitigation To mitigate CVE-2023-36620, users of the Boomerang Parental Control application for Android should update to the latest version. Can you provide Further information on CVE-2023-36620 can be found through the following sources: 'https://seclists.org/bugtraq/2023/11/03/cve-2023-36620'. Could you provide A possible attack scenario for CVE-2023-36620 would involve a malicious actor with physical or remote access to the device. What is CV CVE-2023-2968 is a security vulnerability identified in both LINE@ for Android version 1.0.0 and LINE@ for iOS version 1.0.0. What is the severity level of CVE-2023-2968 is rated as 'MEDIUM' with a base score of 5.9. This score indicates the severity level. When was CVE-2023-2968 was published on 31 October 2023.

What is the potential impact of CVE-2023-2968 includes unauthorized interception and modification of communications. Where can More information about CVE-2023-2968 can be found at these sources:- Japan Vulnerability Notes: <https://jvn.jp/en/ja/vuln/2023/JVNA-2023-0001/>. Can you describe A possible attack scenario for CVE-2023-2968 would be as follows: An attacker could position themselves to intercept and modify communications. What mitigation To mitigate CVE-2023-2968, users should ensure they update the LINE@ application to a version that addresses the vulnerability. What is CV CVE-2023-0897 is a Common Vulnerabilities and Exposures identification for a security flaw that was identified in the LINE@ application. What is the CVSS Base Score for CVE-2023-0897 is '5.9 MEDIUM.' The CVSS (Common Vulnerability Scoring System) score indicates the severity level. When was CVE-2023-0897 was published on 31 October 2023.

Can you provide For CVE-2023-0897, an example attack scenario involves an attacker being in a position to intercept communications. Where can Additional information about CVE-2023-0897 can be found through the references provided: the official LINE@ security advisory. What software The versions of LINE affected by CVE-2023-0897 are LINE@ for Android version 5.0.2 and earlier, as well as LINE@ for iOS version 5.0.2 and earlier. What kind CVE-2023-0897 is classified as a Man-in-the-Middle (MITM) attack vulnerability, wherein an attacker can intercept and modify communications. What is CV CVE-2023-46139 is a vulnerability found in KernelSU, a kernel-based root solution for Android devices. How does CVE-2023-46139 impacts device security by allowing a piece of malware to potentially obtain root privileges. What are the The verification logic vulnerability in CVE-2023-46139 lies in how KernelSU incorrectly obtains the signature. How was CVE-2023-46139 was fixed in KernelSU version 0.7.0. The update corrects the signature verification logic. What workarounds The recommended workarounds for CVE-2023-46139 include keeping the KernelSU manager installed and updated. What is the CVE-2023-46139 has been given a CVSS base score of 5.7, which falls into the 'MEDIUM' severity category. Can you provide An example of an attack scenario for CVE-2023-46139 involves an attacker crafting a malicious Android application. Where can The official advisory and patch for CVE-2023-46139 can be found on KernelSU's GitHub page under the 'CVE-2023-46139' issue. What is the vulnerability identified by CVE-2023-21307 pertains to a permissions bypass issue in Bluetooth or Bluetooth Low Energy (BLE) communication. What type CVE-2023-21307 is classified as an information disclosure vulnerability due to a permissions bypass in Bluetooth or BLE communication. What is the base score of CVE-2023-21307 is rated as 5.0, which falls within the 'MEDIUM' severity level in the CVSS score. When was CVE-2023-21307 was published on 30 October 2023.

Where can More details about CVE-2023-21307 can be found on the Android security bulletin page, accessible via <https://source.android.com/security/bulletin/2023-10-30>. Does CVE-2023-21307 require user interaction, as indicated by the description of the vulnerability. Are there code examples The CVE-2023-21307 description does not include code examples. However, it is related to a permissions bypass issue. Can you describe In a possible attack scenario of CVE-2023-21307, an attacker with access to a paired Bluetooth device can potentially access sensitive information. What is CV CVE-2023-40140 is a security vulnerability identified in the android.view.InputDevice.create function. How severe CVE-2023-40140 is considered to have a high severity level with a base score of 7.8.

When was CVE-2023-40140 published on 27 October 2023.

Does CVE-2023-40140 require user interaction for exploitation? No, user interaction is not required for the exploitation of CVE-2023-40140.

What type of attack could CVE-2023-40140 lead to? CVE-2023-40140 could lead to local escalation of privilege without the attacker needing any additional information.

Where can more information and any available patches for CVE-2023-40140 be found? More information and any available patches for CVE-2023-40140 can be found on the official Android Security Center.

Could you provide an attack scenario for CVE-2023-40140? An attack scenario for CVE-2023-40140 might involve a malicious application that is installed on the device and exploits the vulnerability to escalate privileges.

What mitigation steps are recommended for CVE-2023-40140? To mitigate risks associated with CVE-2023-40140, it is recommended to apply any patches or updates provided by the device manufacturer.

What is CVE-2023-46102? CVE-2023-46102 is a vulnerability identified in the Android Client application that, when enrolled with a device, allows an attacker to access sensitive information.

How was CVE-2023-46102 rated? CVE-2023-46102 was rated as having a high severity with a base score of 8.8 on the CVSS (Common Vulnerability Scoring System).

On what date was CVE-2023-46102 published? CVE-2023-46102 was published on 25 October 2023.

What protocol does CVE-2023-46102 exploit? CVE-2023-46102 exploits the MQTT protocol, which is used to facilitate remote management of devices.

Where can further details about CVE-2023-46102 be found? Further details about CVE-2023-46102 can be found in the security advisory published by Bosch at the following link: [https://www.bosch-iaas-platform.com/en-us/security-advisories/cve-2023-46102](#).

Can you describe an attack scenario for CVE-2023-46102? An attack scenario for CVE-2023-46102 involves an attacker positioning themselves within the same network as the device to intercept MQTT traffic.

Is there a code example for CVE-2023-46102? Since CVE-2023-46102 involves proprietary applications and encryption keys, a specific code example is not provided.

What is CVE-2023-45851? CVE-2023-45851 refers to a security vulnerability found in the Android Client application when it is enrolled with a device.

How severe is CVE-2023-45851? The vulnerability described by CVE-2023-45851 has been assigned a Base Score of 8.8, which categorizes it as HIGH severity.

When was CVE-2023-45851 published? CVE-2023-45851 was published on 25 October 2023.

What could happen if CVE-2023-45851 is exploited? If CVE-2023-45851 is exploited, an attacker could force the Android Client application to connect to a malicious MQTT broker.

Where can further information on CVE-2023-45851 be found? Further information on CVE-2023-45851 can be found in the security advisory published by Bosch at the following link: [https://www.bosch-iaas-platform.com/en-us/security-advisories/cve-2023-45851](#).

What is an MQTT broker? In the context of CVE-2023-45851, an MQTT broker is a server that the Android Client application uses to communicate with the cloud.

Can you provide an attack scenario for CVE-2023-45851? An attacker wishing to exploit CVE-2023-45851 could set up a malicious MQTT broker and then use network sniffing to intercept traffic.

What mitigation steps are recommended for CVE-2023-45851? To mitigate the threat of CVE-2023-45851, the Android Client application should be updated to enforce secure communication protocols.

What is CVE-2023-45844? CVE-2023-45844 refers to a security vulnerability that enables a low privileged user with access to a device to access sensitive information.

How severe is CVE-2023-45844? The CVE-2023-45844 vulnerability has been given a Base Score of 6.8, which categorizes it as a 'MEDIUM' severity.

When was CVE-2023-45844 published? CVE-2023-45844 was published on 25 October 2023.

Where can additional details about CVE-2023-45844 be found? Additional details about CVE-2023-45844 can be found in the security advisory provided by Bosch at the following link: [https://www.bosch-iaas-platform.com/en-us/security-advisories/cve-2023-45844](#).

What kind of attack scenarios are associated with CVE-2023-45844? Potential attack scenarios for CVE-2023-45844 could involve an attacker with physical access to a device attempting to access sensitive data.

What are the recommended mitigation steps for CVE-2023-45844? To mitigate CVE-2023-45844, it is advisable to follow the guidance issued by the device manufacturer to update the application.

Can you provide an example of how CVE-2023-45844 is exploited? As CVE-2023-45844 relates to installing an arbitrary application to access device settings, there is no specific code example provided.

What is the CVE ID for the vulnerability affecting the Android Client application by using HTTP to retrieve sensitive information? The CVE ID for the vulnerability affecting the Android Client application by using HTTP to retrieve sensitive information is CVE-2023-45321.

Can you describe CVE-2023-45321? CVE-2023-45321 describes a security issue in the Android Client application used for remote management of devices.

What is the severity of CVE-2023-45321? CVE-2023-45321 has a CVSS base score of 8.8, which is categorized as HIGH. This indicates that the vulnerability has a high impact on the system.

When was CVE-2023-45321 published? CVE-2023-45321 was published on 25 October 2023. Detailed information about this vulnerability can be found in the security advisory published by Bosch at the following link: [https://www.bosch-iaas-platform.com/en-us/security-advisories/cve-2023-45321](#).

What are some possible attack scenarios for CVE-2023-45321? Possible attack scenarios for CVE-2023-45321 include an attacker conducting a man-in-the-middle (MITM) attack to intercept sensitive data.

What are the recommended mitigation steps for CVE-2023-45321? While the CVE entry itself does not specify mitigation steps, general recommendations to mitigate the threat include updating the application and using secure communication protocols.

What is CVE-2023-45220? CVE-2023-45220 is a security vulnerability identified in the Android Client application. It involves the use of insecure communication protocols to access sensitive information.

When was CVE-2023-45220 published? CVE-2023-45220 was published on 25 October 2023.

What is the severity of CVE-2023-45220? The base score assigned to CVE-2023-45220 is 8.8, which categorizes it as HIGH severity.

What type of information is at risk due to CVE-2023-45220? The sensitive information at risk due to CVE-2023-45220 includes the IP address and credentials required for remote management.

How does CVE-2023-45220 affect the Android Client application's security? CVE-2023-45220 affects the Android Client application's security by using an insecure communication protocol (HTTP) instead of a secure one (HTTPS).

Can users configure the Android Client application to use HTTPS? No, users cannot configure the Android Client application to use HTTPS in place of HTTP for CVE-2023-45220.

Where can more information about CVE-2023-45220 be found? More information about CVE-2023-45220 is available in the security advisory published by Bosch, which can be found at the following link: [https://www.bosch-iaas-platform.com/en-us/security-advisories/cve-2023-45220](#).

What are some possible attack scenarios associated with CVE-2023-45220? Possible attack scenarios associated with CVE-2023-45220 include man-in-the-middle attacks where an attacker intercepts and modifies the communication between the device and the cloud.

Could you provide a hypothetical code example for CVE-2023-45220? Certainly, a hypothetical code example for CVE-2023-45220 might look like the following: ````java// Exar`

What is CVE-2023-41960? CVE-2023-41960 refers to a security vulnerability in which an unprivileged third-party application can access sensitive information.

What type of applications are affected by CVE-2023-41960? Applications affected by CVE-2023-41960 are Android Agent applications that expose content providers to untrusted applications.

How severe is CVE-2023-41960? CVE-2023-41960 has a Base Score of 3.3, which is categorized as LOW severity. This suggests that while the vulnerability exists, its impact is limited.

When was CVE-2023-41960 published? CVE-2023-41960 was published on 25 October 2023.

Where can More details about CVE-2023-41960 can be found in the advisory published by Bosch Security and Saf  
What can e By exploiting CVE-2023-41960, an attacker could potentially modify sensitive settings within the Andr  
Can you pr Imagine a third-party application that is installed on an Android device and does not require any speci  
What is CV CVE-2023-41372 is a security vulnerability identified in a particular Android Client application. This vu  
How sever CVE-2023-41372 has been classified with a Base Score of 7.8, which categorizes it as HIGH in terms of  
On what d CVE-2023-41372 was published on 25 October 2023.

Where can More information about CVE-2023-41372 can be found in the security advisory published by Bosch Se  
What are t A possible attack scenario for CVE-2023-41372 involves an unprivileged third-party application install  
Are there c While specific code examples that demonstrate the vulnerability in CVE-2023-41372 are not provided,  
What is the CVE ID for the host header injection vulnerability in sisqualWFM for Android is CVE-2023-36085.  
Which vers The versions of sisqualWFM for Android affected by CVE-2023-36085 range from 7.1.319.103 to 7.1.3  
What type CVE-2023-36085 addresses a host header injection vulnerability.

What is the The potential impact of the vulnerability described in CVE-2023-36085 includes the possibility of an at  
Can you pr An example of an attack using the CVE-2023-36085 vulnerability would involve an attacker sending a  
What is the The CVSS Base Score assigned to CVE-2023-36085 is 6.1, which is categorized as MEDIUM severity.  
When was CVE-2023-36085 was published on 25 October 2023.

Where can More information or a proof of concept for CVE-2023-36085 can be found on the provided GitHub rep  
What is CV CVE-2023-41898 is a security vulnerability in the Home Assistant Companion for Android app, which a  
How sever The base score for CVE-2023-41898 is rated as 7.8, which classifies it as HIGH severity. It denotes a sig  
Which vers CVE-2023-41898 affects the Home Assistant Companion for Android app versions up to 2023.8.2. Use  
How do us Users can mitigate the risk associated with CVE-2023-41898 by upgrading the Home Assistant Compai  
What are t Possible attack scenarios for CVE-2023-41898 include an attacker exploiting the arbitrary URL loading

Where can Official advisories and information related to CVE-2023-41898 can be found in the security advisory U  
Is there a c As CVE-2023-41898 involves a security vulnerability with WebView in an Android application, a code e  
What is the The GitHub Security Lab (GHSL) Vulnerability Report identifier associated with CVE-2023-41898 is `GH  
What is CV CVE-2023-5365 refers to a reported vulnerability in the HP LIFE Android Mobile application, which is s  
How sever The vulnerability described by CVE-2023-5365 has been assigned a base score of 9.8, which categorize  
When was CVE-2023-5365 was published on 09 October 2023.

Where can More information about CVE-2023-5365 can be found in the HP support document at the following UI  
What type CVE-2023-5365 affects the HP LIFE Android Mobile application, which is designed for mobile devices r  
Can you pr While specific code examples may not be publicly disclosed to prevent exploitation, a potential attack  
What is CV CVE-2023-44121 is a security vulnerability that was identified within the LG ThinQ Service, specifically  
How sever CVE-2023-44121 has been rated with a base score of 6.3, which classifies it as a MEDIUM severity vuln  
On what d CVE-2023-44121 was published on 27 September 2023.

How could CVE-2023-44121 could be exploited by a third-party app on an LG device by sending a broadcast with  
Can you pr Certainly. An attacker could craft a malicious app that, once installed on the LG device, sends a specifi  
What make CVE-2023-44121 is especially dangerous because the vulnerability exists in a system-level app (LG Thi  
Where can More information about CVE-2023-44121 can be found at LG's official security bulletin by visiting the  
What are t Possible attack scenarios of CVE-2023-44121 include a third-party application executing unauthorized  
What is the The CVE ID for the vulnerability found in NVIDIA GeForce Now for Android is CVE-2023-31014.

Can you de CVE-2023-31014 is a vulnerability in NVIDIA's GeForce Now for Android where the game launcher con  
What comp CVE-2023-31014 affects the game launcher component of NVIDIA GeForce Now for Android.  
What is the The CVSS base score of CVE-2023-31014 is rated as 4.8, which categorizes it as a MEDIUM severity vul  
What type: The security risks associated with CVE-2023-31014 include limited information disclosure, denial of se  
When was CVE-2023-31014 was published on 20 September 2023.

Where can Further information and updates for CVE-2023-31014 can be found at the NVIDIA support page: <https>  
What might A possible attack scenario for exploiting CVE-2023-31014 could involve a malicious application install

Are there any? As of my knowledge, specific code examples or proof of concept exploits for CVE-2023-31014 have not been disclosed.  
What mitigation? To mitigate the risks posed by CVE-2023-31014, users should ensure they apply any updates or patches as soon as they are available.  
What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-42468.

Can you explain? CVE-2023-42468 is a security flaw in the com.cutestudio.colordialer application versions up to 2.1.8-2.  
What is the base severity score assigned to CVE-2023-42468? The base severity score assigned to CVE-2023-42468 is 5.3, which is categorized as MEDIUM.

When was CVE-2023-42468 published? CVE-2023-42468 was published on 13 September 2023.

Where can I find more information? More information or references regarding CVE-2023-42468 can be found at the following URLs: <https://github.com/cutestudio/colordialer/issues/10>

Can you provide an attack scenario? An attack scenario for CVE-2023-42468 could involve a malicious application installed on the user's device that attempts to make unauthorized phone calls.

What is the CVE ID for the vulnerability that allows unintended phone calls in an Android application? The CVE ID for the vulnerability is CVE-2023-42469.

What is the base severity score assigned to CVE-2023-42469? CVE-2023-42469 has been assigned a severity level of LOW with a base score of 3.3.

Which Android application is affected by CVE-2023-42469? The Android application affected by CVE-2023-42469 is the com.full.dialer.top.secure.encrypted application.

How can CVE-2023-42469 be exploited? CVE-2023-42469 can be exploited by any installed application without needing permissions, by sending a crafted intent.

On which date was CVE-2023-42469 published? CVE-2023-42469 was published on 13 September 2023.

Where can I find more information and a proof of concept? More information and a proof of concept for CVE-2023-42469 can be found on GitHub at the following link: <https://github.com/full-dialer/top-secure-encrypted-app/pull/1>

Describe a possible attack scenario. A possible attack scenario for CVE-2023-42469 could involve a malicious app installed on the user's device that attempts to make unauthorized phone calls.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-4907.

Can you describe CVE-2023-4907? CVE-2023-4907 refers to an inappropriate implementation in Intents within Google Chrome on Android.

What is the base severity score assigned to CVE-2023-4907? CVE-2023-4907 has been assigned a severity level of 'Low' by Chromium security, and it has a CVSS base score of 3.3.

When was CVE-2023-4907 published? CVE-2023-4907 was published on 12 September 2023.

Are there any references? Yes, there are multiple references available for CVE-2023-4907, including a detailed post on the Google Security blog.

What might be a possible attack scenario? A possible attack scenario exploiting CVE-2023-4907 would involve a remote attacker creating a malicious HTML page designed to trigger the vulnerability.

Are there any code examples? As CVE-2023-4907 relates to obfuscating security UI via a crafted HTML page, specific code examples are not provided here.

Has the vulnerability been addressed? Yes, the vulnerability identified by CVE-2023-4907 has been addressed in Google Chrome version 117.0.5938.62.

What is the CVE ID of the vulnerability related to Google Chrome on Android? The CVE ID of the vulnerability related to Google Chrome on Android is CVE-2023-4903.

What does CVE-2023-4903 pertain to? CVE-2023-4903 pertains to an inappropriate implementation in Custom Mobile Tabs in Google Chrome on Android.

What type of attack could CVE-2023-4903 potentially enable? CVE-2023-4903 could potentially enable a phishing attack where an attacker deceives a user by spoofing a legitimate Google Chrome interface.

What is the severity score of CVE-2023-4903? The severity score of CVE-2023-4903 is 4.3, and it is classified as MEDIUM according to the CVE report.

When was CVE-2023-4903 publicly disclosed? CVE-2023-4903 was publicly disclosed on 12 September 2023.

How can users protect themselves? Users can protect themselves from the vulnerability identified in CVE-2023-4903 by updating to Google Chrome version 117.0.5938.62 or later.

Can you provide more information? Certainly, more information on CVE-2023-4903 can be found at the following references: <https://chromium.googlesource.com/chromium/src/+/main:chrome/android/CustomMobileTabsView.java>

Describe a possible attack scenario. A possible attack scenario exploiting CVE-2023-4903 could involve an attacker crafting a malicious HTML page designed to trigger the vulnerability.

What is CVE-2023-4900? CVE-2023-4900 is a security vulnerability that was found in the implementation of Custom Tabs in Google Chrome on Android.

What versions are affected? CVE-2023-4900 affects versions of Google Chrome on Android prior to 117.0.5938.62. Users with versions 117.0.5938.62 or later are not affected.

What is the severity of CVE-2023-4900? The security severity of CVE-2023-4900 is assessed as Medium by Chromium security.

When was CVE-2023-4900 published? CVE-2023-4900 was published on 12 September 2023.

How is CVE-2023-4900 exploited? An attacker can exploit CVE-2023-4900 by creating a crafted HTML page designed to obfuscate a permission request.

What is the base score assigned to CVE-2023-4900? The Base Score assigned to CVE-2023-4900 is 4.3, and it is categorized as MEDIUM in terms of severity.

Where can I find more information or updates? More information or updates related to CVE-2023-4900 can be found on various resources such as the Chromium Security page.

What could be a potential attack scenario? A potential attack scenario involving CVE-2023-4900 could be a phishing campaign where attackers seek to trick users into granting permissions.

Are there any code examples? Specific proof of concept code examples for exploiting CVE-2023-4900 are not provided here. However, a fix was included in Google Chrome version 117.0.5938.62.

Has CVE-2023-4900 been addressed? Yes, CVE-2023-4900 has been addressed by the Google Chrome team, and a fix was included in Google Chrome version 117.0.5938.62.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-42471.

Which version is affected? The wave.ai.browser application for Android is affected through version 1.0.35 by CVE-2023-42471.

What kind of attack is possible? CVE-2023-42471 is a vulnerability that allows a remote attacker to execute arbitrary JavaScript code via a crafted URL.

What component is exploited? CVE-2023-42471 exploits a WebView component within the wave.ai.browser.ui.splash.SplashScreen activity.

How does the attack work? CVE-2023-42471 allows the execution of arbitrary JavaScript due to a manifest entry that exports the 'android.webkit.WebView' module.

What is the severity of CVE-2023-42471? CVE-2023-42471 has been assigned a Base Score of 9.8, which categorizes it as CRITICAL severity.

On what date CVE-2023-42471 was published on 11 September 2023.

Where can More information and proof-of-concept examples related to CVE-2023-42471 can be found on the following GitHub repository: <https://github.com/0x00sec/Android-CVE-2023-42471>

Can you provide A possible attack scenario for CVE-2023-42471 could involve a malicious application installed on the smartphone.

What is the severity A vulnerability with a CVSS Base Score of 9.8, like CVE-2023-42471, indicates that it is of CRITICAL severity.

What is CV CVE-2023-42470 refers to a security vulnerability in the Imou Life com.mm.android.smartlifeiot application.

How serious The vulnerability detailed in CVE-2023-42470 has been given a Base Score of 9.8, which classifies it as CRITICAL.

When was CVE-2023-42470 was published on 11 September 2023.

What kinds Possible attack scenarios for CVE-2023-42470 include attackers crafting malicious intents that could be used to exploit the vulnerability.

Are there any Yes, there are references that provide additional details and proof of concept (POC) for CVE-2023-42470.

What Android The Android application affected by CVE-2023-42470 is the Imou Life com.mm.android.smartlifeiot application.

What is the cause The primary cause of the security issue in CVE-2023-42470 is due to JavaScript execution being enabled.

What is the CVE ID The CVE ID for the vulnerability discovered in the MyCrops HiGrade application is CVE-2023-40040.

Which versions The affected version of the MyCrops HiGrade application by CVE-2023-40040 is 1.0.337 for Android.

What is the base score The base score assigned to CVE-2023-40040 is 5.3 MEDIUM.

What is the CVE-2023-40040 describes a vulnerability where a remote attacker can start the camera feed via the camera app.

On which version CVE-2023-40040 applies to Android Lollipop, version 5.1.1 API 22, because this is the version consistent with the affected version.

How can CVE-2023-40040 can be exploited by a remote attacker who can trigger the camera feed on affected devices.

What percentage As of the publication of CVE-2023-40040, less than five percent of Android devices are affected, as the vulnerability is not widespread.

When was CVE-2023-40040 was published on 11 September 2023.

Where can More information about CVE-2023-40040 can be found at the following URL: <https://github.com/actua11y/android-cve-2023-40040>

What are the Possible attack scenarios for CVE-2023-40040 include an attacker remotely activating the camera on the affected device.

Can you provide A vulnerable component in CVE-2023-40040 would involve the CameraActivity component in the MyCrops application.

What is the CVE The vulnerability affecting the Camera app on various Android versions is identified as CVE-2023-30730.

What type CVE-2023-30730 represents an implicit intent hijacking vulnerability.

Which versions The versions of the Android Camera app impacted by CVE-2023-30730 are prior to 11.0.16.43 in Android 11.

What is the CVSS The CVSS Base Score assigned to CVE-2023-30730 is 5.5, which classifies it as a MEDIUM severity vulnerability.

When was CVE-2023-30730 was published on 06 September 2023.

Where can More information about CVE-2023-30730 can be found on the Samsung Mobile Security webpage at <https://www.samsung.com/global/mobility/security/cve-2023-30730/>

Can you describe A possible attack scenario for CVE-2023-30730 involves a local attacker exploiting the implicit intent hijacking vulnerability.

What kind To exploit CVE-2023-30730, an attacker would need to have the ability to execute code locally on the affected device.

What is the impact The impact of CVE-2023-30730 on affected devices includes the possibility of an attacker accessing sensitive data.

What is CV CVE-2023-30718 refers to a security vulnerability in the WifiApAutoHotspotEnablingActivity of some Android versions.

How severe CVE-2023-30718 is considered to have a LOW severity with a Base Score of 3.3. Despite being a security issue, it is not critical.

When was CVE-2023-30718 was published on 06 September 2023.

Which devices CVE-2023-30718 affects android devices with a specific application component named WifiApAutoHotspotEnablingActivity.

What could By exploiting CVE-2023-30718, a local attacker could potentially change the Auto Hotspot settings on the affected device.

Where can More information about CVE-2023-30718 can be found on the Samsung Mobile Security website, specifically <https://www.samsung.com/global/mobility/security/cve-2023-30718/>

What kind CVE-2023-30718 is categorized as an 'Improper export of android application components' vulnerability.

Can you provide While code examples that exploit vulnerabilities like CVE-2023-30718 are not typically shared to avoid further exploitation.

Has CVE-2023-30718 should have been addressed by the manufacturer in the security maintenance release for the affected versions.

What is CV CVE-2023-0654 refers to a security vulnerability that was found in the WARP Mobile Client for Android.

How does Users affected by CVE-2023-0654 are at risk of falling victim to a tapjacking attack. If an attacker convinces the user to interact with the malicious app, they can steal sensitive information.

What is a tap In the context of CVE-2023-0654, a tapjacking attack involves a malicious application creating a transparent overlay on top of the legitimate application's interface.

What is the CVSS The CVSS base score for CVE-2023-0654 is 3.7, which is categorized as LOW severity. This score is determined by the impact and exploitability of the vulnerability.

What versions CVE-2023-0654 affects WARP Mobile Client versions prior to 6.29 for Android. Users with versions 6.29 and later are not affected.

How can users Users can protect themselves from CVE-2023-0654 by updating their WARP Mobile Client for Android to the latest version.

Can you provide An example attack scenario for CVE-2023-0654 could involve an attacker creating a malicious app that mimics the WARP Mobile Client interface.

Where can Users seeking more information about CVE-2023-0654 can refer to the security advisory published by What is the CVE ID for the vulnerability is CVE-2023-0238.

What versi The WARP Mobile Client versions 6.29 and below for Android are affected by CVE-2023-0238.

Can you de CVE-2023-0238 describes a vulnerability that arises because of the absence of a security policy in the '...

What is the severity rating of CVE-2023-0238? The severity rating of CVE-2023-0238 is 5.5 and is classified as MEDIUM. This indicates that the vulnerability is of moderate severity and should be addressed.

When was CVE-2023-0238 was publicly disclosed on 29 August 2023.

Where can More information and advisories related to CVE-2023-0238 can be found at the following URLs:- <https://www.cisa.gov/ncas/known-vulnerabilities/advisories/advisory-0001>

What are the Potential attack scenarios for CVE-2023-0238 include a scenario where a user installs a seemingly harmless application, which then exploits the vulnerability to gain unauthorized access to the system's memory and execute arbitrary code.

Can you pr Since CVE-2023-0238 pertains to a vulnerability in a commercial application, providing a code example

What is the CVE ID for the vulnerability discovered in the 'Skylark' app for Android and iOS is CVE-2023-40530

What kind CVE-2023-40530 identifies an improper authorization issue in the handler for custom URL scheme in t

Which version? The vulnerability CVE-2023-40530 affects the 'Skylark' app for Android and iOS versions 6.2.13 and earlier.

How severe The CVE-2023-40530 vulnerability has been given a Base Score of 4.7, which is categorized as MEDIUM

On what date was the CVE-2023-40530 vulnerability published? 25 August 2023.

Where can More information about the CVE-2023-40530 vulnerability can be found at the following links: the Go

What attack? An attacker could exploit the CVE-2023-40530 vulnerability by leading a user to access an arbitrary website.

Can you give While specific code examples cannot be provided without more information about the nature of the v

What is the CVE ID for the vulnerability in the 'Rikunabi NEXT' App for Android is CVE-2023-39507.

What type CVE-2023-39507 involves an improper authorization issue in the custom URL scheme handler of the 'F

What are t The vulnerability in CVE-2023-39507 allows a malicious intent to exploit the vulnerability to cause the

What is the CVE-2023-39507 has been assigned a base score of 6.1, which classifies it as a MEDIUM severity vulne

When was CVE-2023-39507 was published on 16 August 2023.

Which vers The 'Rikunabi NEXT' App for Android is affected by CVE-2023-39507 prior to version 11.5.0.

Where can More information about CVE-2023-39507 can be found at the following reference link: <https://jvn.jp/>

Can you de An attacker could craft a malicious intent in the form of a custom URL that, when processed by the vu

What is the CVE ID for the Chrome WebShare vulnerability disclosed is CVE-2023-4363.

Can you pr CVE-2023-4363 is a vulnerability that involves an inappropriate implementation in the WebShare feat

What is the severity rating of CVE-2023-4363? The severity rating of CVE-2023-4363 is Medium with a base score of 4.3.

On what date CVE-2023-4363 was published on 15 August 2023.

Could you | Yes, more information on CVE-2023-4363 can be found at the following URLs:- <https://chromereleases>

Are there : Officially, code examples demonstrating the exploitation of CVE-2023-4363 are not provided for secur

What pote A potential attack scenario for CVE-2023-4363 would involve an attacker creating a malicious HTML p:

How can u: Users can protect themselves from the vulnerability CVE-2023-4363 by updating their Google Chrome

What is the CVE ID related to the Autofill issue in Google Chrome on Android is CVE-2023-4361.

Can you explain the vulnerability in CVE-2023-4361 involves an inappropriate implementation of the Autofill feature in

What is the The security severity of CVE-2023-4361 is rated as 'Medium' by the Chromium security team, and it ha

As of what The issue described in CVE-2023-4361 is addressed in Google Chrome on Android as of version 116.0..

What could A possible attack scenario for CVE-2023-4361 could involve an attacker creating a malicious website w

Where can More information and updates regarding CVE-2023-4361 can be found through several sources: The c

What is the CVE ID for the vulnerability is CVE-2023-4350.

What is the CVE-2023-4350 is associated with an inappropriate implementation in Fullscreen mode in Google Chrome

What versi The vulnerability mentioned in CVE-2023-4350 was fixed in Google Chrome version 116.0.5845.96 for

What seve The Chromium security team has assigned a severity level of 'High' to CVE-2023-4350.

How has the CVSS base score for CVE-2023-4350 has been rated as '6.5 MEDIUM'.

When was CVE-2023-4350 was published on 15 August 2023.

Can you pr Yes, more information about CVE-2023-4350 can be found in the following references:- <https://chromium.com>

What pote An attack scenario for CVE-2023-4350 could involve a remote attacker crafting a malicious HTML page

Are there any Due to the responsible disclosure of vulnerabilities, specific code examples that exploit CVE-2023-435  
What is CV CVE-2023-2312 refers to a security vulnerability found in Google Chrome on Android. It is classified as  
How severe The CVE-2023-2312 vulnerability is considered to have a high severity with a base score of 8.8, indicat  
When was The CVE-2023-2312 vulnerability was published on 15 August 2023.

Which vers Google Chrome on Android versions prior to 116.0.5845.96 are affected by the CVE-2023-2312 vulner

What kind The CVE-2023-2312 vulnerability can lead to an attack in which a remote attacker who has compromised

Where can You can find more information or updates about CVE-2023-2312 from the following references: the of

Could you? A possible attack scenario for CVE-2023-2312 could involve a user visiting a web page containing mali

What is CV CVE-2023-32609 refers to a security vulnerability found in the Intel Unite® Android application. Prior 1

What type CVE-2023-32609 is classified as an 'Improper Access Control' vulnerability, which is a security flaw wh

What is the severity level of CVE-2023-32609 is rated as '5.5 MEDIUM' according to its Base Score.

How can C To mitigate CVE-2023-32609, users of the Intel Unite® Android application should update the applicat

What are t Exploiting CVE-2023-32609 could potentially allow an authenticated user to disclose sensitive informa

Which Inte The Intel Unite® Android application is affected by CVE-2023-32609.

What are t To exploit CVE-2023-32609, the attacker would need to be authenticated, meaning they require some

When was CVE-2023-32609 was published on 11 August 2023.

Where can More information about CVE-2023-32609 can be found on Intel's official security advisory page at the

Has Intel re Yes, Intel has addressed the issue in CVE-2023-32609 and released a patch. Users should update their

What is the CVE ID for the vulnerability found in the Intel Support Android application is CVE-2023-27392.

What type CVE-2023-27392 is a vulnerability caused by incorrect default permissions that may allow a privileged

The severity score assigned to CVE-2023-27392 is 4.4, which is categorized as MEDIUM.

As of which The vulnerability designated by CVE-2023-27392 is patched in the Intel Support Android application as:

On what date CVE-2023-27392 was published on 11 August 2023.

Where can More information about CVE-2023-27392 can be found on Intel's security advisory page at: <http://www.intel.com/content/www/us/en/security-center/advisory.aspx?product=processors&advisory=CVE-2023-27392>

What are p Possible attack scenarios for CVE-2023-27392 might involve a privileged user, such as someone with a

What app i CVE-2023-39957 affects the Nextcloud Talk Android application, which is used for placing video and a

What is the vulnerability in CVE-2023-39957 was due to an unprotected intent which allowed malicious third-

Has CVE-2023-39957 been addressed in Nextcloud Talk Android version 17.0.0, which includes a patch for CVE-2023-39957?

What is the base score assigned to CVE-2023-39957 is 7.8, which is categorized as HIGH severity.

When was CVE-2023-39957 was published on 10 August 2023.

Are there any known workarounds? No known workarounds are available for CVE-2023-39957. The recommended action is to update Nex

Where can More information about CVE-2023-39957 can be found at the provided references, which include a Gi

What are s For CVE-2023-39957, a possible attack scenario could involve a malicious application installed on the s

What is the CVE ID of the vulnerability found in Custom Tabs in Google Chrome on Android is CVE-2023-3736.

Can you do CVE-2023-3736 refers to an inappropriate implementation in Custom Tabs in Google Chrome on Android

What is the CVE-2023-3736 has been assigned a severity level of Medium with a base score of 4.3.

When was CVE-2023-3736 was published on 01 August 2023.

Which version of CVE-2023-3736 affects Google Chrome on Android versions prior to 115.0.5790.98.

How can I learn more about the technical details of CVE-2023-3736 by referring to the Google Chrome F

What are t The possible attack scenarios for CVE-2023-3736 include a remote attacker creating a crafted HTML p:

What should Users do to protect themselves from the vulnerability identified by CVE-2023-3736 by updating their

Are code e Providing an exact code example of how CVE-2023-3736 can be exploited would be irresponsible and

Has CVE-2023-3736 been addressed in Google Chrome version 115.0.5790.98 for Android. Users :

What is CV CVE-2023-36351 is a security vulnerability identified in the Viatom Health ViHealth app for Android. S

**How severe:** The severity of CVE-2023-36351 is rated as 7.8, which is classified as 'HIGH' on the CVSS (Common Vul

When was CVE-2023-36351 was published on 01 August 2023.

Where can Additional details about CVE-2023-36351 can be found in the references provided in the CVE report. T

What are s Potential attack scenarios for CVE-2023-36351 could involve an attacker crafting malicious input or co  
How can C' To mitigate or fix CVE-2023-36351, users should update their Viatom Health ViHealth app for Android  
What is the CVE ID of the vulnerability is CVE-2022-4926.

Can you de CVE-2022-4926 refers to a vulnerability in Google Chrome on Android where insufficient policy enforc  
What is the CVE-2022-4926 has a CVSS base score of 6.5, which is categorized as MEDIUM severity.

When was CVE-2022-4926 was published on 29 July 2023.

Where can More details about CVE-2022-4926 can be found at the following references:- <https://crbug.com/1368>

Has Google Yes, the CVE-2022-4926 vulnerability has been addressed by Google Chrome in version 109.0.5414.11

What would A possible attack scenario exploiting CVE-2022-4926 could involve an attacker creating a malicious HT

Are there a Due to the nature of CVE listings and responsible disclosure practices, code examples that directly exp

Which corr CVE-2022-4926 affects the policy enforcement component regarding Intents on the Android version o

What is the CVE ID for the vulnerability found in Google Chrome's Notifications on Android is CVE-2022-4917.

Can you de CVE-2022-4917 refers to an incorrect security UI issue in the Notifications feature of Google Chrome c

What are t Potential attack scenarios for CVE-2022-4917 include a remote attacker creating a specially crafted H1

What is the CVE-2022-4917 was assigned a base severity level of 4.3, which is considered 'MEDIUM'.

When was CVE-2022-4917 was published on 29 July 2023.

Where can More information about CVE-2022-4917 can be found at the following references: The official Google

Was there Yes, Google Chrome on Android version 103.0.5060.53 resolved the issue described in CVE-2022-4917

What is the CVE ID of the vulnerability fixed in Apple Music 4.2.0 for Android is CVE-2023-32427.

What secu The fix for CVE-2023-32427 involved using HTTPS when sending information over the network to prev

What kind An attacker in a privileged network position, such as on the same local network or with control over a

What is the CVE-2023-32427 has been given a CVSS Base Score of 5.9, classifying it as a MEDIUM severity vulnera

When was CVE-2023-32427 was published on 28 July 2023.

Where can More information about CVE-2023-32427 can be found at the official Apple support page: <https://sup>

What might A possible attack scenario for CVE-2023-32427 could involve an attacker conducting a man-in-the-mid

Could you Before the fix for CVE-2023-32427, if a user connected their Android device running an older version c

What is the CVE ID for the vulnerability is CVE-2023-28203.

Which App The issue referenced in CVE-2023-28203 was fixed in Apple Music 4.2.0 for Android.

What does The vulnerability described in CVE-2023-28203 involves an app being able to improperly access the us

What is the CVE-2023-28203 has been assigned a Base Score of 5.5 MEDIUM, indicating it is a mid-level severity is

When was The CVE-2023-28203 was published on 28 July 2023.

Where can Official information or an advisory about CVE-2023-28203 can be found on the Apple support website

Can you pr An attacker could potentially exploit CVE-2023-28203 by creating a malicious application which, when

Is there an As an ethical practice, the sharing of actual exploitation code for vulnerabilities like CVE-2023-28203 i

What is the CVE ID of the vulnerability in TeleAdapt RoomCast TA-2400 is CVE-2023-33743.

What devic The TeleAdapt RoomCast TA-2400 models ranging from version 1.0 through 3.1 are affected by CVE-2

What is the CVE-2023-33743 reports an issue of Improper Access Control in TeleAdapt RoomCast TA-2400, where

What is the CVE-2023-33743 has a CVSS Base Score of 9.8, which categorizes it as CRITICAL in terms of severity.

When was CVE-2023-33743 was published on 27 July 2023.

Where can Additional information about CVE-2023-33743 can be found at the following URL: <http://packetstorm>

What might Attack scenarios for CVE-2023-33743 could include attackers exploiting the accessible Android Debug

Can you pr While not explicitly providing exploit code, an example of exploiting CVE-2023-33743 would be an att

What is CV CVE-2023-38173 is a security vulnerability identified in Microsoft Edge for Android, which is describec

When was The CVE-2023-38173 was published on 21 July 2023.

What type CVE-2023-38173 is identified as a spoofing vulnerability, which implies that it could enable an attacke

How sever The severity of the CVE-2023-38173 vulnerability is rated as medium with a base score of 4.3.

Where can More information about CVE-2023-38173 can be found on the Microsoft Security Response Center's v



What prod Microsoft Edge for Android is affected by the CVE-2023-38173 vulnerability.

What are p Possible attack scenarios for CVE-2023-38173 could involve an attacker creating a fake website or use

What is CV CVE-2023-34625 refers to a security vulnerability found in the ShowMojo MojoBox Digital Lockbox ver

How can C' CVE-2023-34625 can be exploited in two main ways. First, an attacker could intercept the BLE commu

What is thè The Base Score for CVE-2023-34625 is rated 8.1, indicating a HIGH severity level.

When was CVE-2023-34625 was published on 20 July 2023.

Can you pr One potential attack scenario for CVE-2023-34625 would be an attacker lurking nearby a property sec

Where can More information and details about CVE-2023-34625 can be found in the references provided by secu

What versi The version of ShowMojo MojoBox Digital Lockbox affected by CVE-2023-34625 is version 1.4.

What mea: To mitigate the risks associated with CVE-2023-34625, it is advisable for ShowMojo to release a softw

What is CV CVE-2023-21994 is a security vulnerability identified in the Oracle Mobile Security Suite product of Or

How can C' The CVE-2023-21994 vulnerability can be exploited by an unauthenticated attacker who has access to

What is thè The CVSS Base Score of CVE-2023-21994 is 6.5, which is categorized as MEDIUM severity. A score of 6

On what dè CVE-2023-21994 was published on July 18, 2023.

What are t A successful exploit of CVE-2023-21994 can result in unauthorized access to critical data or complete a

What are s To mitigate CVE-2023-21994, it is recommended that users of the Oracle Mobile Security Suite upgrad

Are there è No specific code examples are provided for CVE-2023-21994 because the details of the vulnerability a

Can you de A possible attack scenario for CVE-2023-21994 could involve an attacker gaining physical access to the

What is CV CVE-2023-36888 refers to a specific cybersecurity vulnerability found in Microsoft Edge for Android, tl

How serio: The CVE-2023-36888 vulnerability has been given a Base Score of 6.3, which categorizes it as a MEDIU

On what dè The CVE-2023-36888 vulnerability was published on 14 July 2023.

Where can Detailed information about CVE-2023-36888 can be found on the Microsoft Security Response Center

What type The CVE-2023-36888 represents a tampering vulnerability. An attacker could exploit this by making u

What kind CVE-2023-36888 affects the Microsoft Edge browser for Android that is Chromium-based.

What are t If an attack successfully exploits CVE-2023-36888, it could result in data corruption, interception or m

What is CV CVE-2023-30678 refers to a security vulnerability that was identified in the Calendar application used

How sever CVE-2023-30678 has been given a base score of 5.5, which categorizes it as a MEDIUM severity vulner

When was The CVE-2023-30678 vulnerability was published on 06 July 2023.

Where can More information about CVE-2023-30678 can be found at the following link provided by Samsung Mo

What versi CVE-2023-30678 affects the Calendar application on Android 13 before version 12.4.07.15.

Can you pr An attack scenario for CVE-2023-30678 could involve a malicious actor crafting a specially designed zip

What is a z A zip path traversal vulnerability, such as the one found in CVE-2023-30678, involves manipulating file

What is thè The CVE ID for the vulnerability related to an open redirect issue in the Brave Browser Android app is

Can you de CVE-2023-28364 describes an Open Redirect vulnerability in the Brave Browser for Android, which oc

What mea: To address the Open Redirect vulnerability CVE-2023-28364, Brave Browser has been updated. Since v

What is thè The severity level of CVE-2023-28364 is classified as 'MEDIUM', and it has a base score of 6.1.

When was The CVE-2023-28364 vulnerability was published on 01 July 2023.

Where can More details about the CVE-2023-28364 vulnerability can be found in the report on HackerOne, which

What kind Exploiting the CVE-2023-28364 vulnerability could lead to a phishing attack scenario. An attacker coul

What is CV CVE-2023-28387 is a security vulnerability identified in the "NewsPicks" App for both Android (versio

How sever The CVE-2023-28387 vulnerability has been assigned a Base Score of 5.5, indicating a MEDIUM severit

On what dè CVE-2023-28387 was published on 30 June 2023.

Where can More information about the CVE-2023-28387 vulnerability can be found on the Japan Vulnerability Nc

Can you ex Potential attack scenarios for CVE-2023-28387 include a local attacker gaining access to the device wh

What is thè The CVE ID for the out-of-bounds read and write vulnerability is CVE-2023-26085.

What type CVE-2023-26085 describes an out-of-bounds read and write vulnerability due to an improper length c

What is thè CVE-2023-26085 has been assigned a CVSS Base Score of 7.8, which is categorized as HIGH severity.

In which version? The vulnerability designated as CVE-2023-26085 was identified in Arm NN Android-NN-Driver version 1.3.1.  
On what date? The CVE-2023-26085 vulnerability was published on the 29th of June, 2023.

Where can I find more information? More information about CVE-2023-26085 can be found on the ARM Security Center website and the relevant GitHub repository.  
Could you describe the vulnerability? In the case of CVE-2023-26085, an attacker who is able to exploit the out-of-bounds read and write vulnerability could potentially access sensitive data or execute arbitrary code.  
Are there any mitigations? As CVE-2023-26085 pertains to a specific security flaw in Arm NN Android-NN-Driver, it would be inappropriate to provide a general mitigation.  
Has a patch been released? Yes, a patch for the CVE-2023-26085 vulnerability has been released in Arm NN Android-NN-Driver version 1.3.2.  
What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-34761.

Can you describe the vulnerability? CVE-2023-34761 describes a vulnerability in the Hello Cup 1.3.1 for Android app used with the 7-Eleven LED.  
What is the CVSS base score assigned to CVE-2023-34761? The CVSS base score assigned to CVE-2023-34761 is 6.5, which is considered MEDIUM severity.  
When was CVE-2023-34761 published? CVE-2023-34761 was published on 28 June 2023.

Where can I find references for CVE-2023-34761? References for CVE-2023-34761 can be found at the following GitHub URLs: - <https://github.com/actua>  
Could you describe a likely attack scenario for CVE-2023-34761? A likely attack scenario for CVE-2023-34761 involves an attacker within BLE range of the 7-Eleven LED.  
What measures can be taken to mitigate the risks of CVE-2023-34761? To mitigate the risks of CVE-2023-34761, the application developers should release a patch or update.  
What is the CVE ID for the vulnerability discovered in Android's applyRemoteView method? The CVE ID for the vulnerability discovered in Android's applyRemoteView method is CVE-2023-21237.  
What versions of Android are affected by CVE-2023-21237? The versions of Android affected by CVE-2023-21237 are Android 13.

What is the severity rating assigned to CVE-2023-21237? CVE-2023-21237 has been assigned a severity rating of 5.5, which is categorized as MEDIUM.  
What type of vulnerability is CVE-2023-21237? CVE-2023-21237 involves a possibility to hide foreground service notifications due to misleading or incomplete user interaction.  
Is user interaction needed for the exploitation of CVE-2023-21237? No, user interaction is not needed for the exploitation of CVE-2023-21237.  
When was CVE-2023-21237 published? CVE-2023-21237 was published on the 28th of June, 2023.

Where can I find technical details regarding CVE-2023-21237? Technical details regarding CVE-2023-21237 can be found on the Android Security Bulletin website, specifically in the section for CVE-2023-21237.  
Can you provide an example of how an attacker could potentially exploit CVE-2023-21237? An attacker could potentially exploit CVE-2023-21237 by creating a malicious application that uses the vulnerable function.  
What is the Android ID associated with CVE-2023-21237? The Android ID associated with CVE-2023-21237 is A-251586912.

What is the CVE ID for the vulnerability involving an out-of-bounds write in aoc\_service\_set\_read\_blocked? The CVE ID for the vulnerability involving an out-of-bounds write in aoc\_service\_set\_read\_blocked of aoc.c is CVE-2023-21236.  
Can you describe CVE-2023-21236? CVE-2023-21236 refers to a vulnerability in the aoc\_service\_set\_read\_blocked function of aoc.c, where an out-of-bounds write occurs.  
What is the severity rating assigned to CVE-2023-21236? CVE-2023-21236 has been assigned a Base Score of 6.7, which is categorized as MEDIUM severity. This is based on the CVSS v3.1 score.  
When was CVE-2023-21236 published? CVE-2023-21236 was published on 28 June 2023.

Where can I find more information or updates about CVE-2023-21236? More information or updates about CVE-2023-21236 can be found at the following link: <https://source.android.com/security/bulletin/2023-06>  
For which Android versions is CVE-2023-21236 applicable? CVE-2023-21236 is applicable to the Android operating system, specifically for the Android kernel. Examples of affected versions include Android 12 and Android 13.  
Is user interaction needed to exploit the vulnerability described in CVE-2023-21236? No, user interaction is not needed to exploit the vulnerability described in CVE-2023-21236.

What kind of attack is CVE-2023-21236? An attacker needs to have System execution privileges to potentially exploit the vulnerability described in CVE-2023-21236.  
What are the potential consequences of an exploit of CVE-2023-21236? The potential consequences of an exploit of CVE-2023-21236 include a local escalation of privilege. An attacker could potentially gain root access to the device.  
What is the Android ID assigned to the vulnerability tracked as CVE-2023-21236? The Android ID assigned to the vulnerability tracked as CVE-2023-21236 is A-270148537.

What is the CVE ID of the vulnerability affecting the Android kernel? The CVE ID of the vulnerability affecting the Android kernel is CVE-2023-21226.  
Can you describe the vulnerability with CVE ID CVE-2023-21226? The vulnerability with CVE ID CVE-2023-21226 is a possible out of bounds read issue due to an incorrect bounds check in the kernel.  
What versions of Android are affected by CVE-2023-21226? The versions of Android affected by CVE-2023-21226 are not specified in the CVE description beyond the fact that it affects the kernel.  
How severe is the CVE-2023-21226 vulnerability? The severity of the CVE-2023-21226 vulnerability is rated as 'HIGH' with a base score of 7.5, indicating a high level of severity.  
What type of attack is CVE-2023-21226? No additional execution privileges are required for an attacker to exploit the vulnerability denoted by CVE-2023-21226.  
Is user interaction needed to exploit the vulnerability noted as CVE-2023-21226? No, user interaction is not required to exploit the vulnerability noted as CVE-2023-21226, which makes it more dangerous.  
When was CVE-2023-21226 published? The vulnerability CVE-2023-21226 was published on 28 June 2023.

Where can I find more information about the CVE-2023-21226 vulnerability? More information about the CVE-2023-21226 vulnerability can be found on the Android Security Bulletin website, specifically in the section for CVE-2023-21226.  
What are some possible attack scenarios for CVE-2023-21226? Possible attack scenarios for CVE-2023-21226 could include an attacker sending a maliciously crafted packet to the device.  
What is the CVE ID for the vulnerability allowing a potential bypass of the protected confirmation screen on Android? The CVE ID for the vulnerability allowing a potential bypass of the protected confirmation screen on Android is CVE-2023-21225.  
Can you describe CVE-2023-21225? CVE-2023-21225 describes a security flaw where there is a possibility to bypass the protected confirmation screen on Android.

What is the base score assigned to CVE-2023-21225? The base score assigned to CVE-2023-21225 is 7.8, which is categorized as HIGH.  
When was CVE-2023-21225 published? CVE-2023-21225 was published on 28 June 2023.  
Are there any official sources of information provided for CVE-2023-21225? Yes, an official source of information provided for CVE-2023-21225 is the Android Security Bulletin, which provides details about the vulnerability and its impact.  
What versions of Android are affected by CVE-2023-21225? The Android versions affected by CVE-2023-21225 are not explicitly specified, but it pertains to a security flaw in the Android operating system.

What type Exploiting CVE-2023-21225 can lead to local escalation of privilege on the affected Android device.

Is user inte Yes, user interaction is needed to exploit the vulnerability described in CVE-2023-21225.

Can you pr A possible attack scenario for CVE-2023-21225 might involve an attacker with physical access to a vuln

What is the CVE ID of the vulnerability is CVE-2023-21224.

What kind CVE-2023-21224 is associated with a possible out of bounds read due to a heap buffer overflow in the

With CVE-2 No additional execution privileges are needed to exploit the vulnerability described by CVE-2023-212

Is user inte No, user interaction is not needed for the exploitation of CVE-2023-21224.

What type CVE-2023-21224 could lead to remote information disclosure.

What is the CVSS Base Score assigned to CVE-2023-21224 is 7.5, categorized as HIGH.

When was CVE-2023-21224 was published on 28 June 2023.

Can you pr Yes, more details about CVE-2023-21224 can be found at: <https://source.android.com/security/bullet>

Can you de A possible attack scenario for CVE-2023-21224 might involve a malicious actor crafting a specific paylo

Are there a Unfortunately, without access to the specific source code of the vulnerability in question, we cannot p

What is the CVE ID of the vulnerability is CVE-2023-21223.

Can you de CVE-2023-21223 describes an issue where there is a possible out of bounds read due to a missing bou

What is the CVE-2023-21223 has been assigned a CVSS base score of 7.5, which is categorized as HIGH severity.

When was CVE-2023-21223 was published on 28 June 2023.

Does explo No, exploitation of CVE-2023-21223 does not require user interaction.

What prod The product affected by CVE-2023-21223 is Android, specifically the Android kernel.

Where can More information about CVE-2023-21223 can be found at the provided reference URL: <https://source>

Is there a c While the exact code for CVE-2023-21223 is not provided, the issue arises from a missing bounds chec

What kind Possible attack scenarios for CVE-2023-21223 include an attacker exploiting the vulnerability to read s

What is CV CVE-2023-21222 identifies a vulnerability in the load\_dt\_data function of storage.c, where there is a p

What is the CVSS Base Score assigned to CVE-2023-21222 is 6.7, categorized as MEDIUM severity.

Has CVE-2023-21222 was published on 28 June 2023.

Are there a One reference is provided for further information on CVE-2023-21222: the Android Security Bulletin In

Which And CVE-2023-21222 affects multiple versions of the Android kernel, as noted in the technology-specific ic

What are t Exploiting CVE-2023-21222 could lead to local escalation of privilege, giving an attacker execution priv

What prer To exploit CVE-2023-21222, the attacker would need to already have System execution privileges on t

Is user inte No, user interaction is not required to exploit CVE-2023-21222. An attacker can take advantage of this

Can you pr While an exact code example is not provided for CVE-2023-21222, a conceptual example would involv

What pote Potential attack scenarios for CVE-2023-21222 involve an attacker who already has system-level acces

What is CV CVE-2023-21220 is a security vulnerability that affects the Android kernel. It is described as a potentia

What versi The affected versions for CVE-2023-21220 include Android kernel versions that were available before

What is the CVSS Base Score for CVE-2023-21220 is 7.5, which is cate

Where can More details about CVE-2023-21220 can be found at the following URL: <https://source.android.com/s>

What type The CVE-2023-21220 vulnerability allows for remote information disclosure attacks. An attacker could

What are t To exploit CVE-2023-21220, an attacker would need a means to monitor or intercept cellular network

What are t If CVE-2023-21220 is exploited, the potential consequences include unauthorized access to sensitive i

Has CVE-2023-21220 has been made public, with the published date being 28 June 2023. Individuals and c

What is the CVE ID of the vulnerability is CVE-2023-21219.

Can you de CVE-2023-21219 addresses a security issue where there is a possible use of unencrypted transport ov

What is the severity rating assigned to CVE-2023-21219 is 7.5, which is categorized as HIGH.

Was user i No, user interaction is not needed for the exploitation of the vulnerability mentioned in CVE-2023-212

On what d CVE-2023-21219 was published on 28 June 2023.

Where can Detailed information about CVE-2023-21219 can be found at <https://source.android.com/security/bu>

What versi CVE-2023-21219 affects the Android kernel, but the specific versions of Android impacted have not be

What are s Possible attack scenarios for CVE-2023-21219 could involve an attacker passively intercepting unencyr

What is CV CVE-2023-21214 is a vulnerability in the addGroupWithConfigInternal function of p2p\_iface.cpp withi

Which pro The product affected by CVE-2023-21214 is Android, specifically version Android-13.

What are t To exploit CVE-2023-21214, an attacker would need System execution privileges on the affected devic

What is the CVSS base score for CVE-2023-21214 is 4.4, which is categorized as MEDIUM severity.

When was CVE-2023-21214 was published on 28 June 2023.

Where can More information about CVE-2023-21214 can be found on the Android Security Bulletin website, part

Could you A possible attack scenario for CVE-2023-21214 could involve a malicious application that has already c

Are there a As an AI, I don't have access to real-world exploits and do not provide code examples for exploiting vu

What type CVE-2023-21214 is classified as an out-of-bounds read vulnerability due to unsafe deserialization in th

What is the CVE ID of the reported vulnerability is CVE-2023-21213.

Can you pr Certainly. The vulnerability described by CVE-2023-21213 exists in the 'initiateTdsTeardownInternal'

What is the CVSS base score for CVE-2023-21213 is 4.4, which is cate

When was CVE-2023-21213 was published on the 28th of June, 2023.

Where can More information about CVE-2023-21213 can be found at the Android Security Bulletin webpage: http

What prod The product affected by CVE-2023-21213 is Android, specifically versions corresponding to Android 13

What are t To exploit the vulnerability of CVE-2023-21213, an attacker would need System execution privileges o

Could you An example of an attack involving CVE-2023-21213 would be an attacker with system-level access to t

What is CV CVE-2023-21212 refers to a security vulnerability present in multiple files of Android version 13, whic

What is the impact of CVE-2023-21212 could lead to local information disclosure, where sensitive data access

What are t To exploit CVE-2023-21212, an attacker would need to possess System execution privileges on the aff

Has CVE-2( Yes, CVE-2023-21212 has been publicly disclosed. Information about the vulnerability and its fixes can

Which vers CVE-2023-21212 affects Android-13. Users of this version of Android should ensure they apply the lat

What is the base score assigned to CVE-2023-21212 is 4.4 on a scale of 0 to 10, categorized as MEDIUM severi

Can you de In a potential attack scenario for CVE-2023-21212, an attacker who has already gained System executi

What actio To mitigate CVE-2023-21212, users of affected Android-13 devices should apply the security updates

What is the CVE ID for the vulnerability is CVE-2023-21211.

Can you de CVE-2023-21211 refers to a security issue in multiple files of the Android operating system where the

What is the CVE-2023-21211 has been assigned a severity base score of 5.5, which is categorized as MEDIUM.

On which c CVE-2023-21211 was published on 28 June 2023.

Which vers Android version 13 is affected by CVE-2023-21211.

Where can You can find more information about CVE-2023-21211 in the official Android security bulletin at the fo

What kind No user interaction is required to exploit CVE-2023-21211, making it potentially easier to exploit as it

What are t The possible attack scenarios for CVE-2023-21211 include an attacker leveraging the heap buffer over

What is the CVE ID of the vulnerability is CVE-2023-21210.

Could you CVE-2023-21210 refers to a vulnerability in the initiateHs20IconQueryInternal function of sta\_iface.cp

What is the CVE-2023-21210 has been assigned a base score of 4.4, categorizing it as a MEDIUM severity level vul

When was CVE-2023-21210 was published on 28 June 2023.

What Andr The Android versions affected by CVE-2023-21210 are those based on Android version 13.

Where can More details about CVE-2023-21210 can be found in the Android security bulletin at [https://source.ar](https://source.android.com/security/bulletin/2023-06-28)

What kind To exploit the CVE-2023-21210 vulnerability, an attacker would need System execution privileges.

Is user inte No, user interaction is not required to exploit the vulnerability related to CVE-2023-21210.

What are s Potential attack scenarios for CVE-2023-21210 include an attacker with system-level access exploiting

What is CV CVE-2023-21209 refers to a security vulnerability found in multiple functions of sta\_iface.cpp, which a

How seriou The severity of CVE-2023-21209 is rated as 'MEDIUM' with a Base Score of 6.7. This indicates that it po

What versi CVE-2023-21209 affects Android version 13. This is specified in the vulnerability description, and com

What are t To exploit CVE-2023-21209, an attacker would need to have System execution privileges on the affect

Is user inte No, user interaction is not necessary for the exploitation of CVE-2023-21209. The vulnerability can be

What is the CVE-2023-21209 was published on 28 June 2023. It was included in the security bulletins and resource

Where can More information on CVE-2023-21209 can be found at the provided reference link: <https://source.android.com/security/bulletin/2023-06-01>

Can you de In a possible attack scenario for CVE-2023-21209, an attacker with System execution privileges could

What is the CVE ID for the vulnerability is CVE-2023-21208.

What type CVE-2023-21208 identifies a possible out of bounds read due to improper input validation.

In which file The vulnerability CVE-2023-21208 is found in the `sta_iface.cpp` file.

What are the Exploiting CVE-2023-21208 could lead to local information disclosure.

What level System execution privileges are needed to exploit the vulnerability CVE-2023-21208.

Is user inte No, user interaction is not needed to exploit the vulnerability mentioned in CVE-2023-21208.

On what platform The CVE-2023-21208 vulnerability exists on the Android platform.

Which versions Android version 13 is affected by the vulnerability CVE-2023-21208.

What is the Android ID associated with CVE-2023-21208 is A-262245254.

How severe CVE-2023-21208 has been assessed with a base score of 4.4, which is categorized as MEDIUM severity

Where can More details or advisories about CVE-2023-21208 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-01>

What date The vulnerability CVE-2023-21208 was published on 28 June 2023.

Can you provide An attacker with system-level execution privileges could exploit the out of bounds read issue in CVE-2

What is the CVE ID for the vulnerability is CVE-2023-21207.

What are the CVE-2023-21207 describes a vulnerability in the `initiateTdlSetupInternal` function of `sta_iface.cpp`, w

What is the CVSS base score for CVE-2023-21207 is 6.7, which is classified as MEDIUM severity.

Which product The product affected by CVE-2023-21207 is Android, particularly version Android-13.

When was CVE-2023-21207 was published on 28 June 2023.

Where can More information about CVE-2023-21207 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-06-01>

Could you provide A possible attack scenario for CVE-2023-21207 could involve a malicious application that is already ins

What level An attacker would need System execution privileges to exploit CVE-2023-21207.

What is CV CVE-2023-21206 refers to a security vulnerability identified in the `initiateVenueUrlAnqpQueryInternal`

How severe The vulnerability identified by CVE-2023-21206 has been given a Base Score of 4.4, which is categorize

What are the In order to exploit the vulnerability CVE-2023-21206, an attacker would need System execution privile

Does explo No, exploiting CVE-2023-21206 does not require user interaction. An attacker can take advantage of t

On which version CVE-2023-21206 is found in Android version 13.

Where can More information about the CVE-2023-21206 vulnerability can be found on the Android security bulletin

When was CVE-2023-21206 was published on June 28, 2023.

Can you explain A possible attack scenario for CVE-2023-21206 could involve an attacker who has already gained Syste

Are there c As CVE-2023-21206 is related to unsafe deserialization leading to an out-of-bounds read, typically no

What is the CVE ID for the vulnerability related to an out of bounds read due to unsafe deserialization in Andr

What is the CVE-2023-21205 could lead to local information disclosure as there is a possible out of bounds read di

What privil No additional execution privileges are needed to exploit CVE-2023-21205.

Is user inte No, user interaction is not needed for the exploitation of CVE-2023-21205.

Which And CVE-2023-21205 affects Android version Android-13.

What is the severity rating of CVE-2023-21205 is 5.5, which is categorized as MEDIUM.

When was CVE-2023-21205 was published on 28 June 2023.

Where can More information about CVE-2023-21205 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-01>

Can you provide A potential attack scenario for CVE-2023-21205 could involve a malicious application installed on the

What is 'sta\_iface.cpp' is the source file in Android where the vulnerability CVE-2023-21205 is located. The fla

What is CV CVE-2023-21204 refers to a security vulnerability discovered in multiple files of the Android operating

What type CVE-2023-21204 is classified as an out of bounds read vulnerability, which could potentially result in l

How severe CVE-2023-21204 has been assigned a Base Score of 4.4, which categorizes it as a MEDIUM severity vul

What version of Android version 13 is affected by the out of bounds read vulnerability identified by CVE-2023-21204.

What are the prerequisites? To exploit CVE-2023-21204, an attacker would need System execution privileges on the affected device.

Is user interaction required? No, user interaction is not needed for an attacker to exploit CVE-2023-21204.

On what date was CVE-2023-21204 published? CVE-2023-21204 was published on 28 June 2023.

Where can more detailed information regarding CVE-2023-21204 be found? More detailed information regarding CVE-2023-21204 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21204>.

Can you provide specific code examples for exploiting CVE-2023-21204? While specific code examples for exploiting CVE-2023-21204 may not be readily available due to ethical considerations, the vulnerability involves an out of bounds read in the `startWpsPbcInternal` function of `sta_if`.

What are the potential impacts? If CVE-2023-21204 is successfully exploited, an attacker could gain access to sensitive information from the device's memory.

What is the CVE ID of the vulnerability discovered in Android-13? The vulnerability discovered in Android-13 is identified as CVE-2023-21203.

What are the prerequisites for CVE-2023-21203? CVE-2023-21203 involves a potential out of bounds read in the `startWpsPbcInternal` function of `sta_if`.

What is the base score assigned to CVE-2023-21203? The base score assigned to CVE-2023-21203 is 6.7, which is categorized as MEDIUM severity.

When was CVE-2023-21203 published? CVE-2023-21203 was published on 28 June 2023.

Where can more information about CVE-2023-21203 be found? More information about CVE-2023-21203 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21203>.

What are the attack scenarios for CVE-2023-21203? Exploiting CVE-2023-21203 could lead to local escalation of privilege, potentially allowing a malicious user to gain access to sensitive information.

What are the attack scenarios for CVE-2023-21203? Attack scenarios for CVE-2023-21203 could involve an unauthorized user with access to the system partition.

What is the CVE ID of the vulnerability published on June 28, 2023? The CVE ID of the vulnerability published on June 28, 2023, is CVE-2023-21202. It involves a possible out of bounds read in the `btm_delete_stored_link_key_complete` function of `btm`.

Which file is affected by the CVE-2023-21202 vulnerability? The file affected by the CVE-2023-21202 vulnerability is `btm_delete_stored_link_key_complete` in the `btm` module.

What causes CVE-2023-21202? CVE-2023-21202 is caused by a missing bounds check in the `btm_delete_stored_link_key_complete` function.

What are the prerequisites for CVE-2023-21202? To exploit the CVE-2023-21202 vulnerability, an attacker would need to have System execution privileges on the affected device.

What is the base score assigned to CVE-2023-21202? The CVSS Base Score assigned to CVE-2023-21202 is 4.5, classified as MEDIUM severity.

Which Android version is affected by CVE-2023-21202? Android version 13 is affected by CVE-2023-21202.

How can users find more detailed information about CVE-2023-21202? Users can find more detailed information about CVE-2023-21202 by visiting the Android security bulletin page: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21202>.

What might be an attack scenario exploiting CVE-2023-21202? An attack scenario exploiting CVE-2023-21202 could involve an attacker with System execution privileges sending a crafted Bluetooth message to the device.

What is the Android ID associated with CVE-2023-21202? The Android ID associated with CVE-2023-21202 is A-260568359.

What is the CVE ID for the vulnerability associated with a potential out of bounds read in the Bluetooth implementation? The CVE ID for the vulnerability associated with a potential out of bounds read in the Bluetooth implementation is CVE-2023-21201.

Can you describe the vulnerability signified by CVE-2023-21201? The vulnerability signified by CVE-2023-21201 is related to an issue in the `'on_create_record_event'` function of `btm`.

How severe is the CVE-2023-21201 vulnerability? The CVE-2023-21201 vulnerability has been given a base score of 7.5 and is classified as HIGH in terms of severity.

When was the CVE-2023-21201 vulnerability published? The CVE-2023-21201 vulnerability was published on the 28th of June, 2023.

Which Android version is affected by the CVE-2023-21201 vulnerability? The Android version affected by the CVE-2023-21201 vulnerability is Android-13.

What kind of user interaction is required to exploit the CVE-2023-21201 vulnerability? No user interaction is required to exploit the CVE-2023-21201 vulnerability.

Are there any other details about CVE-2023-21201? Yes, more information about CVE-2023-21201 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21201>.

Can you provide a potential attack scenario for CVE-2023-21201? A potential attack scenario for CVE-2023-21201 would involve an attacker remotely sending a crafted Bluetooth message to the device.

What is the Android ID associated with CVE-2023-21201? The Android ID associated with CVE-2023-21201 is A-263545186.

What is CVE-2023-21200? CVE-2023-21200 is a security vulnerability identified in a specific version of Android. It concerns an out of bounds read in the `btu_ble_proc_ltk_req` function of `btu_hcif.cc`.

What version of Android is affected by CVE-2023-21200? The vulnerability designated as CVE-2023-21200 affects Android version 13.

What is the base score assigned to CVE-2023-21200? The Common Vulnerability Scoring System (CVSS) base score assigned to CVE-2023-21200 is 5.5, indicating MEDIUM severity.

How can CVE-2023-21200 be exploited? CVE-2023-21200 can potentially be exploited by an attacker who is able to interact with the system in a way that triggers the vulnerable code path.

Was there a patch or remediation for CVE-2023-21200? Details about a patch or remediation for CVE-2023-21200 would generally be provided in the security bulletin.

When was CVE-2023-21200 published? The vulnerability CVE-2023-21200 was published on 28 June 2023.

Could you provide a possible attack scenario for CVE-2023-21200? A possible attack scenario for CVE-2023-21200 involves an attacker who targets an Android application that interacts with the Bluetooth stack.

Where can more information about CVE-2023-21200 be found? More information about CVE-2023-21200 can be found in Android's security bulletins or advisories. The relevant bulletin is <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21200>.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21199.

Can you describe CVE-2023-21199? CVE-2023-21199 refers to a vulnerability in the `btu_ble_proc_ltk_req` function of `btu_hcif.cc`, which is part of the Bluetooth stack.

What Android version is impacted by CVE-2023-21199? The Android version impacted by CVE-2023-21199 is Android-13.

What is the base score for CVE-2023-21199? The CVSS base score for CVE-2023-21199 is 4.4, which is categorized as MEDIUM severity. This score is based on the potential for local escalation of privilege.

When was CVE-2023-21199 published? CVE-2023-21199 was published on 28 June 2023.

Where can more information about CVE-2023-21199 be found? More information about CVE-2023-21199 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21199>.

What are the prerequisites for CVE-2023-21199? To exploit CVE-2023-21199, an attacker would need System execution privileges on the affected Android device.

What type The CVE-2023-21199 vulnerability could lead to an attack where an adversary might exploit the out of

What are s Mitigation of vulnerabilities like CVE-2023-21199 can involve updating the affected systems with secu

What is CV CVE-2023-21198 is a security vulnerability found in the Android operating system, specifically in a file

How sever CVE-2023-21198 has a base score of 5.5, which categorizes it as a MEDIUM severity vulnerability. This

Which And CVE-2023-21198 affects Android version 13. Devices running this version of Android are potentially at

When was CVE-2023-21198 was published on 28 June 2023. It was disclosed through security bulletins and advis

Where can Detailed information about CVE-2023-21198 can be found in the Android Security Bulletin at the follo

Is user inte No, user interaction is not required to exploit CVE-2023-21198. An attacker could potentially exploit tl

What type To exploit CVE-2023-21198, an attacker does not require any additional execution privileges. The vuln

Can you de A possible attack scenario for CVE-2023-21198 might involve a malicious application that is installed o

Has CVE-20 The provided information does not state whether CVE-2023-21198 has been fixed. However, it is com

What is CV CVE-2023-21197 refers to a security vulnerability in the `btm_acl_process_sca_cmpl_pkt` function of bt

What versi CVE-2023-21197 affects Android-13.

What is the CVE-2023-21197 has been assigned a CVSS base score of 7.5, which is categorized as HIGH. This indica

What is the The impact of exploiting CVE-2023-21197 would be remote information disclosure. This means that an

What is ne CVE-2023-21197 can be exploited by an attacker without the need for any additional execution privile

Where can More information about CVE-2023-21197 can be found on the Android Security Bulletin page, specific

On which c CVE-2023-21197 was published on 28 June 2023.

What are t Possible attack scenarios involving CVE-2023-21197 may include an attacker sending specially crafted

What is the The CVE ID for the vulnerability is CVE-2023-21196.

Can you de CVE-2023-21196 refers to a vulnerability in the `'btm_ble_batchscan_filter_track_adv_vse_cback'` func

What privil To exploit the vulnerability CVE-2023-21196, an attacker would need System execution privileges on t

Is user inte No, user interaction is not required to exploit the vulnerability identified by CVE-2023-21196.

What is the The Base Score assigned to CVE-2023-21196 is 4.4 and its severity is classified as MEDIUM.

When was CVE-2023-21196 was published on 28 June 2023.

Which And The vulnerability CVE-2023-21196 affects Android version Android-13.

Where can Additional information regarding CVE-2023-21196 can be found through the reference: <https://source>

What type CVE-2023-21196 could lead to local information disclosure if successfully exploited.

Could you A possible attack scenario for CVE-2023-21196 would involve a malicious actor with system-level acce

What is the The CVE ID assigned to the vulnerability is CVE-2023-21195.

Can you pr CVE-2023-21195 refers to a vulnerability in the `btm_ble_periodic_adv_sync_tx_rcvd` function of btm\_

What is the The CVSS Base Score for CVE-2023-21195 is 4.5, which is categorized as MEDIUM severity.

As of its pu CVE-2023-21195 affects Android version Android-13.

Does CVE- No, user interaction is not needed for the exploitation of CVE-2023-21195.

What date The vulnerability CVE-2023-21195 was publicly disclosed on 28 June 2023.

Where can More information about CVE-2023-21195 can be found in the Android security bulletin at: <https://sou>

What privil An attacker would need System execution privileges to exploit CVE-2023-21195.

What are t The potential attack scenarios for CVE-2023-21195 include an attacker who has already compromised

What is the The CVE ID of the vulnerability discussed is CVE-2023-21194.

What is the CVE-2023-21194 addresses a possible out of bounds read in `gatt_dbg_op_name` function of `gatt_utils`

What comp The component affected by CVE-2023-21194 is the Bluetooth server on Android.

Which vers Android version 13 is impacted by CVE-2023-21194.

What are t An attacker would need System execution privileges to exploit CVE-2023-21194.

Is user inte No, user interaction is not needed for exploitation of the vulnerability described in CVE-2023-21194.

What kind CVE-2023-21194 is an information disclosure vulnerability with a severity score of 4.4, categorized as

When was CVE-2023-21194 was published on 28 June 2023.

What is the The Android ID associated with CVE-2023-21194 is A-260079141.

Can you pr For more information on CVE-2023-21194, you can visit the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21194>.

What are t A potential attack scenario for CVE-2023-21194 includes an attacker with system-level access exploiting the vulnerability to gain unauthorized access to sensitive data.

What is CV CVE-2023-21193 is a security vulnerability identified in the VideoFrame.h file of Android, specifically in the VideoFrame class.

How sever The CVE-2023-21193 vulnerability has been assessed with a Base Score of 7.5, which places it in the HIGH severity category.

Does CVE-21193 require user interaction? No, user interaction is not required to exploit the CVE-2023-21193 vulnerability. It can be exploited remotely.

When was CVE-2023-21193 published? CVE-2023-21193 was published on 28 June 2023.

Where can More details about CVE-2023-21193 can be found on the Android Security Bulletin page for Pixel devices.

Can you de In a possible attack scenario for CVE-2023-21193, an attacker could craft a malicious video file that, when played, would trigger the vulnerability.

What is the CVE ID for the reported security vulnerability in InputMethodManagerService.java? The CVE ID for the reported security vulnerability in InputMethodManagerService.java is CVE-2023-21192.

What kind CVE-2023-21192 involves a security issue in Android where there is a possible way to set up input method spoofing.

For which Android version CVE-2023-21192 has been reported for the Android platform. CVE-2023-21192 has been reported for the Android platform.

Which version of Android is affected by CVE-2023-21192? Android version Android-13 is affected by CVE-2023-21192.

What is the CVSS base score associated with CVE-2023-21192? The CVSS base score associated with CVE-2023-21192 is 7.8, which classifies it as HIGH severity.

When was CVE-2023-21192 published? CVE-2023-21192 was published on 28 June 2023.

Can you pr Detailed information regarding CVE-2023-21192 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21192>.

Does exploiting CVE-2023-21192 require user interaction for exploitation? No, exploiting CVE-2023-21192 does not require user interaction for exploitation.

What are t A possible attack scenario for CVE-2023-21192 could involve a malicious app exploiting the improper handling of input methods.

What is the Android ID associated with CVE-2023-21192? The Android ID associated with CVE-2023-21192 is A-227207653.

What is the CVE ID for the vulnerability allowing a possible bypass of notification hide preference in Android? The CVE ID for the vulnerability allowing a possible bypass of notification hide preference in Android is CVE-2023-21191.

What is the base score of CVE-2023-21191, indicating its severity, is 7.8, which is classified as HIGH.

Which file The logic error leading to the vulnerability identified by CVE-2023-21191 is located in the file named NotificationManager.java.

What are t The affected version of Android by the CVE-2023-21191 vulnerability is Android-13.

What type Due to the vulnerability CVE-2023-21191, there is a possible local escalation of privilege without the need for user interaction.

Is user interaction required to exploit the vulnerability described in CVE-2023-21191? No, user interaction is not required to exploit the vulnerability described in CVE-2023-21191.

Where can More information about CVE-2023-21191 can be found at the source Android Security Bulletin URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21191>.

What specific error responsible for CVE-2023-21191 is due to a logic error in the code within the fixNotificationManager method.

When was The CVE-2023-21191 vulnerability was published on 28 June 2023.

Can you de A possible attack scenario for exploiting the CVE-2023-21191 vulnerability could involve an attacker displaying a notification that they can click on to trigger the vulnerability.

What is the CVE ID of the vulnerability related to Android's btm\_acl\_encrypt\_change? The CVE ID of the vulnerability related to Android's btm\_acl\_encrypt\_change is CVE-2023-21190.

Can you de CVE-2023-21190 is a vulnerability in the btm\_acl\_encrypt\_change function of btm\_acl.cc in Android-13.

What is the severity of CVE-2023-21190? CVE-2023-21190 has been assigned a severity base score of 5.0, which categorizes it as MEDIUM severity.

When was The vulnerability CVE-2023-21190 was published on 28 June 2023.

Is there a remote exploit for CVE-2023-21190? Yes, further details about CVE-2023-21190 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21190>.

Does the CVE-2023-21190 require user interaction? Yes, the exploitation of CVE-2023-21190 requires user interaction.

Which version of Android is affected by the vulnerability CVE-2023-21190? Android-13 is affected by the vulnerability CVE-2023-21190.

What could A possible attack scenario could involve a malicious remote device initiating a Bluetooth connection with a victim device.

What is CV CVE-2023-21189 refers to a security vulnerability discovered in the LockTaskController.java file within the Android framework.

What version of Android is affected by CVE-2023-21189? CVE-2023-21189 affects Android 13.

What is the severity of CVE-2023-21189? CVE-2023-21189 has been assigned a base severity score of 7.3, which is categorized as HIGH according to the CVSS v3.1 scoring system.

When was CVE-2023-21189 published? CVE-2023-21189 was published on 28 June 2023.

What kind CVE-2023-21189 allows for local escalation of privilege, meaning that a person with physical access to the device can exploit the vulnerability to gain higher privileges.

Are there any references for more information about CVE-2023-21189? Yes, one of the references for more information about CVE-2023-21189 is the Android security bulletin.

What type To exploit CVE-2023-21189, user interaction is needed, although the specifics of that interaction are not detailed in the CVE description.

What is the Android ID associated with CVE-2023-21189? The Android ID associated with CVE-2023-21189 is A-213942596.

Can you pr An attack scenario for CVE-2023-21189 might involve an installed rogue application that tricks the user into interacting with it.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-21188.

Can you de CVE-2023-21188 describes a vulnerability in btm\_ble\_update\_inq\_result of btm\_ble\_gap.cc where a race condition exists.



Which product? The product affected by CVE-2023-21188 is Android, specifically version Android-13.

What level of user interaction is required? Exploiting the vulnerability described in CVE-2023-21188 requires System execution privileges.

Is user interaction needed? No, user interaction is not needed for the exploitation of the vulnerability in CVE-2023-21188.

What is the severity? The CVSS Base Score assigned to CVE-2023-21188 is 4.4, which is considered MEDIUM severity.

When was it disclosed? CVE-2023-21188 was publicly disclosed on 28 June 2023.

Where can I find more information? More information about CVE-2023-21188 can be found on the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-06#CVE-2023-21188>.

What kind of attack scenarios are possible? Possible attack scenarios for CVE-2023-21188 include an attacker with system-level access exploiting the vulnerability to perform a local escalation of privilege.

Could you provide an example? As CVE-2023-21188 details a specific vulnerability rather than a generic issue, I cannot provide an exact example.

What is the CVE ID? The CVE ID for the security vulnerability found in the UsbAccessoryUriActivity.java file of Android-13 is CVE-2023-21187.

Can you describe the vulnerability? CVE-2023-21187 is a vulnerability in the onCreate method of the UsbAccessoryUriActivity.java file in Android-13.

What is the severity? The Base Score assigned to CVE-2023-21187 is 7.8, which is categorized as HIGH.

When was it published? The CVE-2023-21187 vulnerability was published on 28 June 2023.

Is user interaction needed? No, user interaction is not needed to exploit the CVE-2023-21187 vulnerability.

What are the potential impacts? A successful exploit of CVE-2023-21187 could allow an attacker to perform a local escalation of privilege.

Are there any mitigations? Yes, details about CVE-2023-21187 can be found at the official Android Security Bulletin webpage, specifically <https://source.android.com/security/bulletin/2023-06#CVE-2023-21187>.

What Android versions are affected? CVE-2023-21187 affects devices running Android version 13.

As a developer, how can I protect my application? To protect your Android application from vulnerabilities like CVE-2023-21187, you should follow best practices for secure coding and keep your application up-to-date.

What kind of attack scenarios are possible? Possible attack scenarios using the CVE-2023-21187 vulnerability include an attacker who has physical access to the device.

What is the CVE ID? The CVE ID of the vulnerability is CVE-2023-21186.

What kind of vulnerability is it? CVE-2023-21186 identifies a vulnerability that could lead to a remote denial of service due to a possible network flood.

Which product is affected? The product affected by CVE-2023-21186 is Android, specifically version Android-13.

What is the severity? The CVSS base score of CVE-2023-21186 is 7.5, which is categorized as HIGH severity.

On what date was it published? CVE-2023-21186 was published on 28 June 2023.

Where can I find more information? More information about CVE-2023-21186 can be found in the Android security bulletin at the following link: <https://source.android.com/security/bulletin/2023-06#CVE-2023-21186>.

Does CVE-2023-21186 require user interaction? No, CVE-2023-21186 does not require user interaction for exploitation.

What kind of attack is possible? Exploitation of CVE-2023-21186 does not require any additional execution privileges.

Could you provide an example? An attacker could exploit CVE-2023-21186 by crafting malicious network traffic that triggers an out of memory condition.

Is there a mitigation or patch? The details for a mitigation or patch would typically be provided in the security bulletin or updates issued by the manufacturer.

What is the CVE ID? The CVE ID of the vulnerability reported in WifiNetworkFactory.java is CVE-2023-21185.

Can you describe the vulnerability? CVE-2023-21185 refers to a security issue in multiple functions of WifiNetworkFactory.java, where the vulnerability could lead to a denial of service.

What is the severity? The CVSS Base Score assigned to CVE-2023-21185 is 7.8, which is classified as HIGH. This signifies that the vulnerability has a high impact on the system.

On what date was it published? CVE-2023-21185 was officially published on 28 June 2023.

Which product is affected? The product affected by CVE-2023-21185 is Android, specifically Android version 13.

Where can I find more information? More information about CVE-2023-21185 can be found in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-06#CVE-2023-21185>.

What are the potential attack scenarios? Potential attack scenarios for CVE-2023-21185 could involve a guest user on an Android device exploiting the vulnerability to perform a denial of service.

What is the CVE ID? The CVE ID of the vulnerability is CVE-2023-21184.

Can you describe the vulnerability? CVE-2023-21184 involves a possible permission bypass due to a logic error in the getCurrentPrivilegedPermissions method.

What is the severity? The CVSS base score assigned to CVE-2023-21184 is 7.8, which is classified as HIGH.

Was user interaction needed? No, user interaction was not needed to exploit the vulnerability in CVE-2023-21184.

Which product is affected? The product affected by CVE-2023-21184 is Android, specifically Android version 13.

What is the source of information? The official source of information regarding CVE-2023-21184 can be found at the Android Security Bulletin: <https://source.android.com/security/bulletin/2023-06#CVE-2023-21184>.

Could you provide an example? Exploiting CVE-2023-21184 would likely involve writing a malicious application that takes advantage of the permission bypass.

Has a patch been released? The information provided does not explicitly mention a patch, but the reference to the Android security bulletin suggests that a patch or mitigation is being discussed.

What is the CVE ID? The CVE ID of the vulnerability is CVE-2023-21183.

What kind of vulnerability is it? Due to the vulnerability CVE-2023-21183, there could be a local escalation of privilege.

Does the vulnerability require user interaction? No, user interaction is not needed to exploit the vulnerability CVE-2023-21183.

Which version of Android is affected? Android version Android-13 is affected by the CVE-2023-21183 vulnerability.

What is the base score assigned to CVE-2023-21183 is 7.8, which is classified as HIGH.

When was the CVE-2023-21183 vulnerability published on 28 June 2023.

Where can more information about the CVE-2023-21183 vulnerability be found at: <https://source.android.com/security/bulletin/2023-06-01>

What is the logic error in CVE-2023-21183 pertains to a possibility where NFC tag data could be read while the device is in a locked state.

Are there any additional execution privileges needed to exploit the vulnerability CVE-2023-21183.

What potential attack scenarios from the CVE-2023-21183 vulnerability could include an attacker surreptitiously capturing NFC data.

What is the CVE ID of the vulnerability is CVE-2023-21182.

What type CVE-2023-21182 is an out of bounds read vulnerability with a base score of 4.4, which is categorized as MEDIUM severity.

What are the conditions to exploit CVE-2023-21182, an attacker would need System execution privileges. No user interaction is required.

Which Android version Android-13 is affected by CVE-2023-21182.

Where can more information about CVE-2023-21182 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-01>

On which date CVE-2023-21182 was published on 28 June 2023.

What potential attack scenarios CVE-2023-21182 could lead to local information disclosure where sensitive data could be read from memory.

What kind of privileges are needed to exploit CVE-2023-21182.

Describe a possible attack scenario for CVE-2023-21182 could involve a malicious application that has already gained System execution privileges.

What is the Android ID associated with CVE-2023-21182 is A-252764175.

What is the CVE ID for the vulnerability is CVE-2023-21181.

Which file The vulnerability CVE-2023-21181 is in the file `btm_ble_gap.cc`.

What type CVE-2023-21181 is a heap buffer overflow vulnerability that could potentially lead to local information disclosure.

What are the conditions to exploit the vulnerability CVE-2023-21181, System execution privileges are required.

Is user interaction required for exploiting CVE-2023-21181.

Which product The product affected by CVE-2023-21181 is Android, specifically version Android-13.

What is the CVSS base score for CVE-2023-21181 is 4.4, classified as MEDIUM severity.

When was the CVE-2023-21181 vulnerability published on 28 June 2023.

Where can more information about the CVE-2023-21181 vulnerability be found at the following source: <https://source.android.com/security/bulletin/2023-06-01>

Can you describe a possible attack scenario for CVE-2023-21181 could involve a malicious application that is already installed on the device.

What is the CVE ID CVE-2023-21180 is a security vulnerability identified in the `xmlParseTryOrFinish` function of the `libxml2` library.

What version CVE-2023-21180 affects Android version 13.

What is the impact of CVE-2023-21180 is significant—it is a heap buffer overflow that could lead to remote information disclosure.

How severe CVE-2023-21180 has been assigned a base score of 7.5, which is categorized as HIGH severity. This reference is from the CVE database.

What is the Android ID associated with CVE-2023-21180 is A-261365944.

On what date CVE-2023-21180 was published on 28 June 2023.

Where can more information about CVE-2023-21180 can be found in the Android security bulletin for Pixel devices: <https://source.android.com/security/bulletin/2023-06-01>

Can you provide a hypothetical example while specific code for CVE-2023-21180 is not provided, a hypothetical example might involve improper memory management.

What possible attack scenarios for CVE-2023-21180 could include an attacker crafting a malicious XML file to trigger the overflow.

What is the CVE ID of the described vulnerability is CVE-2023-21179.

What component CVE-2023-21179 affects the `XmlUtil.java` component, which is responsible for parsing security parameters from XML.

Can you explain CVE-2023-21179 is a security vulnerability that results from the improper use of cryptographic functions.

Is user interaction required for exploiting the vulnerability described in CVE-2023-21179.

What type CVE-2023-21179 could lead to local escalation of privilege on the affected Android device.

What version Android version 13 is impacted by the vulnerability referenced in CVE-2023-21179.

How severe The vulnerability detailed in CVE-2023-21179 is rated as 7.8, which is classified as HIGH severity.

What is the official resource for more details regarding CVE-2023-21179 is the Android Security Bulletin link, <https://source.android.com/security/bulletin/2023-06-01>

What kind of attack scenarios associated with CVE-2023-21179 might involve an attacker using a malicious XML file to trigger the vulnerability.

Which product CVE-2023-21179 identifies a security vulnerability in the Android operating system.

What is the CVE ID of the vulnerability in `KeyUtil.cpp` has been assigned CVE-2023-21178.

What kind of vulnerability CVE-2023-21178 refers to a race condition in `installKey` of `KeyUtil.cpp` that could lead to a failure in file operations.

What are the prerequisites? To exploit the vulnerability CVE-2023-21178, an attacker would need to have System execution privileges. Which Android version is affected? CVE-2023-21178 affects Android version 13.

What is the CVSS Base Score? The CVSS Base Score given to CVE-2023-21178 is 4.1, which is classified as MEDIUM severity.

When was it published? The information about CVE-2023-21178 was published on 28 June 2023.

Where can more details be found? More details about CVE-2023-21178 can be found at the Android Security Bulletin page, specifically at [https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21178](#). Can you provide a possible attack scenario? A possible attack scenario for CVE-2023-21178 involves an attacker with system-level access waiting for a user interaction. No, user interaction is not needed to exploit the vulnerability CVE-2023-21178.

Could you provide a code example? Since CVE-2023-21178 is specific to the Android platform and its source code, a code example of the vulnerability is not provided.

What is CVE-2023-21177? CVE-2023-21177 refers to a security vulnerability found in the WindowManagerService.java file of the Android framework.

What type of vulnerability is it? CVE-2023-21177 is classified as an information disclosure vulnerability, where a missing permission check allows an attacker to access sensitive system information.

How severe is it? CVE-2023-21177 has been given a base score of 5.5, which categorizes it as a MEDIUM severity vulnerability.

When was it published? CVE-2023-21177 was published on the 28th of June, 2023.

Which Android version is affected? Android version 13 is affected by CVE-2023-21177.

Where can more information be found? More information about CVE-2023-21177 can be found on the official Android security bulletin page at [https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21177](#).

Can you provide an attack scenario? An attacker with local access to the device could exploit CVE-2023-21177 by invoking the 'requestAppPermissions' method. Does CVE-2023-21177 require user interaction? No, CVE-2023-21177 does not require user interaction to be exploited.

What is the Android ID associated with CVE-2023-21177? The Android ID associated with CVE-2023-21177 is A-273906410.

What is CVE-2023-21176? CVE-2023-21176 refers to a security vulnerability discovered in the list\_key\_entries function of the WindowManagerService.java file.

What are the prerequisites? Exploiting CVE-2023-21176 could lead to a local denial of service on an affected Android device. This vulnerability requires System execution privileges.

What privileges are needed? To exploit CVE-2023-21176, an attacker would need System execution privileges, which is a high level of privilege.

Does it require user interaction? No, exploiting CVE-2023-21176 does not require user interaction. An attacker with the necessary System execution privileges can exploit the vulnerability.

What is the CVSS Base Score? CVE-2023-21176 has been assigned a CVSS base score of 4.4, which is categorized as medium severity.

On what date was it published? CVE-2023-21176 was published on 28 June 2023.

Which Android version is affected? CVE-2023-21176 affects the Android operating system version 13.

Where can more information be found? More information about CVE-2023-21176 can be found at the published reference, which is the Android Security Bulletin page at [https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21176](#).

What could be the impact? In a possible attack scenario exploiting CVE-2023-21176, an attacker with System execution privileges could cause a local denial of service.

What is CVE-2023-21175? CVE-2023-21175 refers to a security vulnerability that was identified in the onCreate method of DataUsageList.java.

How severe is it? CVE-2023-21175 has been assigned a base score of 7.8, which is considered HIGH severity. This means that an attacker with System execution privileges can exploit the vulnerability.

Do users need to interact? No, user interaction is not needed for the exploitation of CVE-2023-21175. This means that an attacker can exploit the vulnerability without needing a user to interact with the device.

When was it published? CVE-2023-21175 was published on 28 June 2023, signifying the date when the security community became aware of the vulnerability.

Where can additional information be found? Additional information about CVE-2023-21175 can be found on the Android security bulletin webpage at [https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21175](#).

Can you describe an attack scenario? A possible attack scenario for CVE-2023-21175 would involve a malicious guest user who has physical access to the device and can trigger the vulnerability.

What sort of impact can it have? CVE-2023-21175 allows for a local escalation of privilege, meaning that the attacker, without having a high level of privilege, can gain System execution privileges.

Is there a code example? As of this response, there is no specific code example provided for CVE-2023-21175. The vulnerability is described in the Android Security Bulletin.

What is CVE-2023-21174? CVE-2023-21174 refers to a security vulnerability in 'BillingCycleSettings.java' within Android version 13.

What type of vulnerability is it? CVE-2023-21174 is classified as a permissions bypass vulnerability that could lead to local escalation of privileges.

What is the CVSS Base Score? The base score of CVE-2023-21174 is rated as 7.8, which is considered HIGH severity.

Does it require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21174.

On what date was it published? CVE-2023-21174 was published on 28 June 2023.

What Android version is affected? CVE-2023-21174 affects Android version 13.

Are there any mitigations? Yes, you can find more information about CVE-2023-21174 on the Android security bulletin page at [https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21174](#).

Can you describe an attack scenario? An attacker with access to the guest user account on an Android device could exploit CVE-2023-21174 to gain System execution privileges.

What mitigation is recommended? To protect against CVE-2023-21174, users should update their Android devices to the latest security patch available.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21173.

Can you describe the issue? CVE-2023-21173 is a security issue where multiple methods in DataUsageList.java in Android lack a proper permission check.

What is the CVSS Base Score? CVE-2023-21173 has been assigned a CVSS Base Score of 5.5, which is categorized as MEDIUM severity.

When was it published? The vulnerability identified by CVE-2023-21173 was published on 28 June 2023.

Which version? The version of Android affected by CVE-2023-21173 is Android 13.

Where can I find more information? More information and the official advisory regarding CVE-2023-21173 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21173>.

What is the Android ID associated with CVE-2023-21173? The Android ID associated with CVE-2023-21173 is A-262741858.

Are there any potential attack scenarios? A potential attack scenario for exploiting CVE-2023-21173 could involve a malicious app installed on the device.

What are the requirements to exploit CVE-2023-21173? To exploit CVE-2023-21173, an attacker would need to leverage a missing permission check in DataUsageStats.

What is the CVE ID for the vulnerability involving a permissions bypass in WifiCallingSettings.java? The CVE ID for the vulnerability involving a permissions bypass in WifiCallingSettings.java in Android is CVE-2023-21172.

What version of Android is affected by CVE-2023-21172? The vulnerability CVE-2023-21172 affects Android version Android-13.

What is the base score assigned to CVE-2023-21172? The base score assigned to CVE-2023-21172 is 7.8, which is categorized as HIGH.

When was CVE-2023-21172 published? The CVE-2023-21172 vulnerability was published on 28 June 2023.

Is user interaction required to exploit CVE-2023-21172? No, user interaction is not required to exploit the CVE-2023-21172 vulnerability.

Can you provide the official reference? The official reference for the CVE-2023-21172 vulnerability can be found at <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21172>.

What kind of impact can CVE-2023-21172 have? CVE-2023-21172 can lead to local escalation of privilege without the need for additional execution privileges.

Describe a potential attack scenario. For CVE-2023-21172, an attacker with local access to the device could potentially exploit the permissions bypass.

What is the CVE ID for the vulnerability that involves potential click fraud? The CVE ID for the vulnerability that involves potential click fraud on Android is CVE-2023-21171.

Can you explain the vulnerability identified as CVE-2023-21171? The vulnerability identified as CVE-2023-21171 concerns a possible way to conduct click fraud through the use of a malicious app.

What is the base severity score assigned to CVE-2023-21171? CVE-2023-21171 has been assigned a base severity score of 6.7 on the CVSS scale, categorizing it as MEDIUM.

On which date was CVE-2023-21171 published? CVE-2023-21171 was published on 28 June 2023.

Where can I find more information about CVE-2023-21171? More information about the CVE-2023-21171 vulnerability can be found in the Android security bulletin.

Which version of Android is affected by CVE-2023-21171? CVE-2023-21171 affects Android version 13.

What type of privileges are needed to exploit CVE-2023-21171? An attacker would need System execution privileges to exploit the vulnerability assigned CVE-2023-21171.

Is user interaction required for CVE-2023-21171? No, user interaction is not required for the exploitation of CVE-2023-21171.

What could be a potential attack scenario for CVE-2023-21171? A potential attack scenario involving CVE-2023-21171 could involve a malicious application or process.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21170.

Can you describe the vulnerability? CVE-2023-21170 refers to a security issue in executeSetClientTarget of ComposerCommandEngine.h.

What privileges are needed to exploit CVE-2023-21170? To exploit the vulnerability CVE-2023-21170, System execution privileges are needed.

Does CVE-2023-21170 require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21170.

Which product is affected by CVE-2023-21170? The product affected by CVE-2023-21170 is Android, specifically version Android-13.

What is the severity rating for CVE-2023-21170? CVE-2023-21170 has been assigned a severity rating of 4.4 and classified as MEDIUM.

When was CVE-2023-21170 published? The vulnerability with CVE ID CVE-2023-21170 was published on 28 June 2023.

Where can I find more information about CVE-2023-21170? More information about CVE-2023-21170 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21170>.

Could you describe a potential attack scenario for CVE-2023-21170? A potential attack scenario for CVE-2023-21170 could involve a malicious app that runs on an Android device.

Is there any specific public code example for CVE-2023-21170? As CVE-2023-21170 is a recently identified vulnerability, there may not be specific public code examples available.

What is the CVE ID for the vulnerability involving a possible out of bounds read in p2p\_iface.cpp? The CVE ID for the vulnerability involving a possible out of bounds read in p2p\_iface.cpp is CVE-2023-21169.

Can you describe the vulnerability identified as CVE-2023-21169? The vulnerability identified as CVE-2023-21169 exists in the inviteInternal function of p2p\_iface.cpp, which involves a possible out of bounds read.

What is the base score assigned to CVE-2023-21169? CVE-2023-21169 has been assigned a base score of 4.4, which categorizes it as a vulnerability with MEDIUM severity.

When was CVE-2023-21169 published? CVE-2023-21169 was published on 28 June 2023.

Where can I find more information about CVE-2023-21169? More information about CVE-2023-21169 can be found at the following reference URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21169>.

What versions of Android are affected by CVE-2023-21169? The Android versions affected by CVE-2023-21169 are Android-13.

What type of privileges are needed to exploit CVE-2023-21169? An attacker would need System execution privileges to exploit CVE-2023-21169.

Does CVE-2023-21169 require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21169.

Can you provide a hypothetical attack scenario for CVE-2023-21169? In a hypothetical attack scenario involving CVE-2023-21169, an attacker with System execution privileges could exploit the vulnerability.

Are there any code examples for CVE-2023-21169? As CVE-2023-21169 involves a security issue within a system-level codebase, code examples detailing the vulnerability are not provided.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21168.

What is the base score assigned to CVE-2023-21168? The base score assigned to CVE-2023-21168 is 5.5, which is categorized as MEDIUM severity.

Which product is affected by CVE-2023-21168? The product affected by CVE-2023-21168 is Android, specifically version 13.

Can CVE-2023-21168 be exploited without user interaction? Yes, CVE-2023-21168 can be exploited without any user interaction.

What type of privileges are needed to exploit CVE-2023-21168? No additional execution privileges are needed for an attacker to exploit CVE-2023-21168.

What is the issue? The issue denoted by CVE-2023-21168 is a possible out of bounds read due to a missing bounds check. On which CVE-2023-21168 was published on 28 June 2023.

Where can more information about CVE-2023-21168 be found? More information about CVE-2023-21168 can be found at the Android Security Bulletin URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21168>.

What kind of vulnerability is CVE-2023-21168? The vulnerability CVE-2023-21168 could lead to a local information disclosure attack, where an attacker could read memory locations outside the intended bounds.

What is CVE-2023-21167? CVE-2023-21167 refers to a security vulnerability identified in the `setProfileName` function of the `DevFusion` application. Which Android version is affected? The vulnerability CVE-2023-21167 affects Android version 13.

How serious is CVE-2023-21167? CVE-2023-21167 has been rated with a base score of 5.5, categorizing it as a MEDIUM severity vulnerability. When was CVE-2023-21167 published? CVE-2023-21167 was published on 28 June 2023.

Where can more information about CVE-2023-21167 be found? More information about CVE-2023-21167 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21167>.

Does CVE-2023-21167 require user interaction? No, CVE-2023-21167 does not require user interaction for exploitation; a local attacker can exploit the vulnerability without needing to interact with the user.

What kind of attack is possible? No additional execution privileges are needed to exploit CVE-2023-21167; a local attacker can cause a denial of service by repeatedly triggering the vulnerable code.

Can you provide an attack scenario? An attack scenario for CVE-2023-21167 could involve a malicious application installed on the affected device, which repeatedly triggers the vulnerable function.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-21161.

What is the issue? CVE-2023-21161 involves a possible out of bounds write due to a missing bounds check, which could lead to a denial of service. What are the requirements for exploitation? To exploit CVE-2023-21161, an attacker would need System execution privileges, but user interaction is not required.

Which product is affected? The product affected by CVE-2023-21161 is Android, more specifically the Android kernel.

What are the affected versions? The CVE-2023-21161 affects unspecified versions of the Android kernel.

How was CVE-2023-21161 disclosed? CVE-2023-21161 was publicly disclosed on the Android Security Bulletin webpage dedicated to Pixel updates. What is the severity? CVE-2023-21161 has been assigned a base score of 6.7, which is categorized as MEDIUM severity.

When was CVE-2023-21161 published? CVE-2023-21161 was published on 28 June 2023.

Where can detailed information and updates about CVE-2023-21161 be found? Detailed information and updates about CVE-2023-21161 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21161>.

Can you describe a possible attack scenario? In a possible attack scenario for CVE-2023-21161, an attacker with system-level access on an Android device could trigger a denial of service by repeatedly writing to memory locations outside the intended bounds.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21160.

Can you describe the vulnerability? CVE-2023-21160 refers to a vulnerability in the `BuildSetTcsFci` function of `protocolmiscbuilder.cpp`, which could lead to a denial of service.

What type of attack is possible? No user interaction is needed for the exploitation of the vulnerability identified by CVE-2023-21160.

What is the severity? The base severity score assigned to CVE-2023-21160 is 5.5 and it is classified as MEDIUM.

When was CVE-2023-21160 published? CVE-2023-21160 was published on 28 June 2023.

Where can further information about CVE-2023-21160 be found? Further information about CVE-2023-21160 can be found at the following link: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21160>.

What Android versions are affected? CVE-2023-21160 affects the Android kernel, and it pertains to Android versions prior to the security updates released in June 2023.

What are the requirements for exploitation? By exploiting CVE-2023-21160, an attacker could achieve local information disclosure from the Android kernel.

How can the issue be mitigated? To mitigate or fix CVE-2023-21160, users should apply the security updates provided by the Android security team.

What kind of attack is possible? An attack scenario for CVE-2023-21160 could involve a malicious application that doesn't require any user interaction to trigger the vulnerability.

What is CVE-2023-21159? CVE-2023-21159 refers to a security vulnerability found in the `Parse` function of `simdata.cpp` within the `simdata` application.

How can CVE-2023-21159 be exploited? CVE-2023-21159 can be exploited by an attacker to cause a local escalation of privilege. This means that a user could gain higher privileges than they are entitled to.

What versions are affected? The affected versions of Android for CVE-2023-21159 are not specified in detail, but it is an issue related to the Android kernel.

Where can more detailed information about CVE-2023-21159 be found? More detailed information about CVE-2023-21159 can be found on the official Android security bulletin: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21159>.

What is the severity? The Common Vulnerability Scoring System version 3 (CVSSv3) Base Score for CVE-2023-21159 is rated as MEDIUM (5.5).

What is the most likely attack scenario? In the most likely attack scenario for CVE-2023-21159, an attacker with system-level access could take advantage of the vulnerability to escalate their privileges.

What is CVE-2023-21158? CVE-2023-21158 refers to a security vulnerability identified in the Android kernel, specifically within the `wldata` module.

What versions are affected? The CVE-2023-21158 vulnerability impacts Android kernel versions, but the exact range of versions affected is not specified.

What is the severity? CVE-2023-21158 has been given a base score of 4.4, which is considered MEDIUM severity according to the CVSSv3.

When was CVE-2023-21158 published? CVE-2023-21158 was published on June 28, 2023.

Are there any requirements for exploitation? No, there is no user interaction required to exploit CVE-2023-21158. A malicious entity with System execution privileges can trigger the vulnerability.

What type of attack is possible? To exploit CVE-2023-21158, an attacker would need System execution privileges. This level of access is typically only available to the root user.

Where can additional information about CVE-2023-21158 be found? Additional information about CVE-2023-21158 can be found in the Android security bulletin at: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21158>.

Could you provide an attack scenario? A potential attack scenario for CVE-2023-21158 could involve a malicious application or process that triggers the vulnerable code in the `wldata` module.

What is the CVE ID for the vulnerability related to a heap buffer overflow in the `encode` function of `wldata`? The CVE ID for the vulnerability related to a heap buffer overflow in the `encode` function of `wldata` is CVE-2023-21157.

Can you de CVE-2023-21157 pertains to a possible out of bounds write due to a heap buffer overflow in the encoder. What are t To exploit the vulnerability identified by CVE-2023-21157, an attacker would need System execution privileges. Is user inte No, user interaction is not needed to exploit the vulnerability described in CVE-2023-21157.

On what pl CVE-2023-21157 impacts the Android platform, specifically versions of the Android kernel as mentioned. What is the CVSS base score given to CVE-2023-21157 is 6.7, which classifies it as a MEDIUM severity vulnerability. What date CVE-2023-21157 was published on 28 June 2023.

Are there any Yes, further information on CVE-2023-21157 can be found at the Android Security Bulletin link: <https://source.android.com/security/bulletin/2023-06-28>.

What are s Possible attack scenarios for CVE-2023-21157 involve an attacker with System execution privileges write to memory.

Can you pr While the specific code details related to CVE-2023-21157 have not been provided, a general example of the exploit is as follows:

What is CV CVE-2023-21156 is a security vulnerability that has been identified in a component named BuildGetRegion.

What type CVE-2023-21156 is classified as an information disclosure vulnerability, more specifically, a possible out-of-bounds read.

What are t To exploit CVE-2023-21156, an attacker would need to have system execution privileges on the affected device.

What is the CVSS base score assigned to CVE-2023-21156 is 4.4, which categorizes it as a medium severity vulnerability.

When was CVE-2023-21156 was published on 28 June 2023.

Which versions CVE-2023-21156 affects the Android kernel version, but the specific versions impacted are not detailed.

What are t An exploit of CVE-2023-21156 can lead to local information disclosure from the modem. This means that an attacker can access sensitive information stored in the modem.

What mitig While there are no direct mitigation steps provided, users can refer to the Android security bulletin for more information.

Where can More information about CVE-2023-21156 is available on the Android security bulletin, which can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28>.

What is CV CVE-2023-21155 refers to a security vulnerability found in a component of the Android operating system.

What is the CVSS base score assigned to CVE-2023-21155 is 5.5, which indicates a medium level of severity.

What versions CVE-2023-21155 affects versions of the Android kernel. However, the exact version numbers vulnerable are not specified.

Does exploit No, exploiting CVE-2023-21155 does not require user interaction. This vulnerability can be exploited by an attacker with system execution privileges.

On what date CVE-2023-21155 was published on 28 June 2023.

Where can More information about CVE-2023-21155 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28>.

What type CVE-2023-21155 is a type of out-of-bounds read vulnerability due to a missing null check in a component.

What kind To exploit CVE-2023-21155, an attacker does not need any additional execution privileges beyond those required for system execution.

Can you pr While specific code examples for exploiting CVE-2023-21155 are not provided, an attack scenario could involve an attacker with system execution privileges.

What is the CVE ID associated with the out of bounds read vulnerability in Android is CVE-2023-21154.

Can you de CVE-2023-21154 refers to a vulnerability in the StoreAdbSerialNumber function of protocolmiscbuild.

What privileges To exploit the vulnerability described in CVE-2023-21154, System execution privileges are needed. The attacker must have the ability to execute code on the device.

Is user inte No, user interaction is not needed to exploit the vulnerability detailed in CVE-2023-21154. An attacker can exploit this vulnerability without user interaction.

What products The products affected by CVE-2023-21154 are Android devices that utilize the affected Android kernel version.

What is the CVSS base score for CVE-2023-21154 is 4.4, which is classified as MEDIUM severity. This score indicates a medium level of severity.

When was CVE-2023-21154 was published on 28 June 2023.

Are there any Yes, more information about CVE-2023-21154 can be found at the Android Security Bulletin link provided.

What could A potential attack scenario for exploiting CVE-2023-21154 could involve a malicious application that has been granted system execution privileges.

What is CV CVE-2023-21153 is a security vulnerability identified in the Do\_AIMS\_SET\_CALL\_WAITING function of the Android kernel.

How is CVE CVE-2023-21153 can be exploited by an attacker who has already gained execution privileges on the device.

What are t The successful exploitation of CVE-2023-21153 could lead to a local escalation of privileges, allowing an attacker to gain system execution privileges.

Which And CVE-2023-21153 affects unspecified versions of the Android operating system. The Android kernel is the component affected.

What is the CVSS base score for CVE-2023-21153 has been assigned a base score of 6.7, which classifies it as having MEDIUM severity.

Where can More information about CVE-2023-21153 can be found at the Android Security Bulletin page, specifically at the following URL: <https://source.android.com/security/bulletin/2023-06-28>.

When was CVE-2023-21153 was published on 28 June 2023.

Can you pr An attack scenario for CVE-2023-21153 could involve a malicious application that has already been granted system execution privileges.

What is CV CVE-2023-21152 refers to a security vulnerability identified in the FaceStatsAnalyzer::InterpolateWeights function.

How severe The vulnerability identified by CVE-2023-21152 is rated with a Base Score of 5.5, which is categorized as a MEDIUM severity vulnerability.

Was there No, user interaction is not needed for exploitation of the CVE-2023-21152 vulnerability. This increases the potential for exploitation.

On what date? The CVE-2023-21152 vulnerability was published on 28 June 2023.

Where can I find more information? More information about CVE-2023-21152 can be found at the Android Security Bulletin website, specifically at [https://source.android.com/security/bulletin/2023-06/01](#).

What privileges are required to exploit this? To exploit the CVE-2023-21152 vulnerability, an attacker would need User execution privileges. This means that the attacker must be able to execute code as a regular user.

What is a possible attack scenario? A possible attack scenario for CVE-2023-21152 could involve a malicious application that is already installed on the device. The application could attempt to exploit the vulnerability to gain higher privileges.

What products are affected? The product affected by CVE-2023-21152 is the Android kernel as it applies to the Android operating system.

What is CVE-2023-21151? CVE-2023-21151 refers to a security vulnerability discovered in the Google BMS kernel module that affects certain Android devices.

How severe is it? The severity of CVE-2023-21151 is rated as 'MEDIUM' with a base score of 6.7. This means it represents a moderate risk to the system.

What products are affected? The products affected by CVE-2023-21151 include devices running the Android operating system that use the Google BMS kernel module.

What are the requirements for exploitation? In order to exploit CVE-2023-21151, an attacker would need to have System execution privileges on the device.

Is user interaction required? No, user interaction is not required to exploit CVE-2023-21151. An attacker with the necessary privileges can exploit the vulnerability without the user's knowledge.

Where can I find more information? More information about CVE-2023-21151 can be found in the Android Security Bulletin at the following URL: [https://source.android.com/security/bulletin/2023-06/02](#).

Was a patch released? As specific patch information is not provided in the context, you should refer to the Android Security Bulletin for the latest updates.

What is the CVE ID? CVE-2023-21151 was published on 28 June 2023.

Could you describe the vulnerability? An attack scenario for CVE-2023-21151 could involve a malicious app or a rogue process that already has System execution privileges.

Are there any code examples? Vulnerability details like CVE-2023-21151 often do not come with public code examples, especially when the details are sensitive.

What is the CVE ID of the reported Android vulnerability? The CVE ID of the reported Android vulnerability is CVE-2023-21150.

Can you describe the vulnerability? CVE-2023-21150 describes a vulnerability in the `handle_set_parameters_ctrl` function of `hal_socket.c`.

What privileges are required? To exploit the vulnerability in CVE-2023-21150, an attacker would need System execution privileges.

Is user interaction required? No, user interaction is not needed to exploit the vulnerability in CVE-2023-21150.

What is the severity? CVE-2023-21150 has been given a base score of 4.4, which is classified as MEDIUM severity.

When was it published? The vulnerability with CVE ID CVE-2023-21150 was published on 28 June 2023.

Which Android versions are affected? CVE-2023-21150 affects the Android kernel, but specific version numbers are not mentioned in the provided information.

Where can I find more details? More details about CVE-2023-21150 can be referenced from the Android Security Bulletin for Pixel at [https://source.android.com/security/bulletin/2023-06/03](#).

What is a possible attack scenario? A possible attack scenario for CVE-2023-21150 could involve a malicious application that exploits the vulnerability to gain higher privileges.

What is the CVE ID for the vulnerability that enables the possible activation or deactivation of RCS service with the RCS service? The CVE ID for the vulnerability that enables the possible activation or deactivation of RCS service with the RCS service is CVE-2023-21149.

What is the severity? CVE-2023-21149 describes a security issue in the `registerGsmServiceIntentReceiver` function of `Shan`.

Can you provide the base score? Yes, the base score for CVE-2023-21149 is 7.8, and it is classified as HIGH severity.

When was it published? CVE-2023-21149 was published on 28 June 2023. More information can be found at the following URL: [https://source.android.com/security/bulletin/2023-06/04](#).

What is the affected component? The vulnerability CVE-2023-21149 affects the Android kernel. Specific affected versions have not been mentioned.

Does it require user interaction? No, exploitation of CVE-2023-21149 does not require user interaction, making it easier for an attacker to exploit.

What is a possible attack scenario? A possible attack scenario for exploiting CVE-2023-21149 could involve an unauthorized application or process that exploits the vulnerability to gain higher privileges.

Are there any code examples? Code examples illustrating the exact exploitation technique for CVE-2023-21149 are not provided, as the details are sensitive.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-21148.

What kind of vulnerability is it? CVE-2023-21148 refers to a possible out of bounds read due to a missing null check in `BuildSetConfig`.

What are the requirements for exploitation? To exploit CVE-2023-21148, an attacker would need System execution privileges on the Android device.

Can you provide an attack scenario? An attacker with System execution privileges could exploit CVE-2023-21148 by crafting a malicious input.

What is the base score? The base score of CVE-2023-21148 is 4.4, which is considered MEDIUM severity.

When was it published? CVE-2023-21148 was published on 28 June 2023.

Are there any references for fixing it? Yes, a reference for fixing CVE-2023-21148 can be found in the Android Security Bulletin for Pixel devices at [https://source.android.com/security/bulletin/2023-06/05](#).

Which Android versions are affected? CVE-2023-21148 affects versions of the Android kernel; however, the exact versions are not specified.

What type of vulnerability is it? Exploiting CVE-2023-21148 can lead to local information disclosure, potentially exposing sensitive information.

Is user interaction required? No, user interaction is not required for an attacker to exploit CVE-2023-21148.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-21147.

Where does the vulnerability exist? The vulnerability CVE-2023-21147 exists in the `lwip_i2c_device_disable` function of the `lwip_device_i2c`.

What effect does exploitation have? Exploitation of CVE-2023-21147 could lead to local escalation of privilege on the affected Android device.

Does it require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21147.

What are the affected Android versions? The specific Android versions affected by CVE-2023-21147 are not mentioned in the provided information.

What is the severity base score assigned to CVE-2023-21147 is 7.8, which is considered HIGH.

On what date CVE-2023-21147 was published on 28 June 2023.

What is a potential attack scenario for exploiting CVE-2023-21147 could involve a malicious application execution.

Are there any specific workarounds or fixes for CVE-2023-21147 are not provided in the information, user interaction is required.

What is the CVE ID of the vulnerability is CVE-2023-21146.

What type of vulnerability CVE-2023-21146 could lead to local escalation of privilege with System execution privileges.

Is user interaction needed for the exploitation of CVE-2023-21146. No, user interaction is not needed for the exploitation of CVE-2023-21146.

What is the CVSS base score assigned to CVE-2023-21146 has been assigned a CVSS base score of 6.7, which is categorized as MEDIUM severity.

On which date CVE-2023-21146 was published on 28 June 2023.

Where can you find more information about CVE-2023-21146 in the Android security bulletin at the following URL: [https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21146](#)

Can you provide an example of an attack scenario for CVE-2023-21146 might involve a malicious application installed on the device.

What is the CVE ID for the Android kernel vulnerability is CVE-2023-21066.

Can you describe CVE-2023-21066 describes a vulnerability in the cd\_CodeMsg function within the cd\_codec.c file, where a buffer overflow occurs.

What is the CVSS base score assigned to CVE-2023-21066 has been assigned a Base Score of 9.8, which classifies it as CRITICAL in severity.

When was CVE-2023-21066 was published on 28 June 2023.

Which Android versions are affected CVE-2023-21066 affects Android kernel, but the specific versions impacted by this vulnerability haven't been specified.

Is there any user interaction required to exploit the vulnerability described in CVE-2023-21066. No, user interaction is not required to exploit the vulnerability described in CVE-2023-21066.

Are there any further information regarding CVE-2023-21066 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-28#CVE-2023-21066>

What type of vulnerability CVE-2023-21066 is a heap buffer overflow vulnerability, which could potentially result in remote code execution.

What is the Android ID attributed to CVE-2023-21066 is A-250100597.

Could you provide a possible attack scenario for CVE-2023-21066 could involve an attacker crafting a malicious input that triggers the buffer overflow.

What is the CVE ID of the vulnerability related to a tapjacking/overlay attack in Android 13 is CVE-2022-20443.

Can you provide CVE-2022-20443 involves a vulnerability in the hasInputInfo function of Layer.cpp within Android OS, which allows an attacker to inject input events.

What is the CVSS Base Score assigned to CVE-2022-20443 has been assigned a CVSS Base Score of 7.8, which indicates a HIGH severity level.

On what date The vulnerability CVE-2022-20443 was published on 28 June 2023.

Where can you find Official information and references regarding CVE-2022-20443 can be found on the Android Security Bulletin: [https://source.android.com/security/bulletin/2023-06-28#CVE-2022-20443](#)

Could you describe CVE-2022-20443 could be exploited in a scenario where a malicious application creates an overlay on top of the system interface.

What could be the impact of the tapjacking/overlay attack described in CVE-2022-20443 could be significant, as it allows an attacker to capture sensitive information.

Can you show As CVE-2022-20443 relates to a tapjacking/overlay vulnerability, a theoretical code example would involve creating an overlay window.

What is the CVE ID for the vulnerability found in the laola.redbull application for Android is CVE-2023-29459.

Can you describe The vulnerability in CVE-2023-29459 pertains to the exposure of an exported activity named at.redbull.LaolaActivity.

What is the CVSS Base Score assigned to CVE-2023-29459 is 6.1, which is categorized as MEDIUM severity.

On what date CVE-2023-29459 was published on 26 June 2023.

Where can you find More information about CVE-2023-29459 can be found on the Google Play Store page for the laola.redbull application.

Could you describe A potential attack scenario for CVE-2023-29459 might involve an attacker crafting a malicious data: URI to access the activity.

Is there any user interaction While I don't provide actual code that can exploit vulnerabilities, a theoretical example for CVE-2023-29459 would involve using a data: URI.

What is the CVE ID for the directory traversal vulnerability in the Basecamp Android application is CVE-2023-36612.

What version The vulnerability CVE-2023-36612 affects the Basecamp com.basecamp.bc3 application for Android version 3.27.0.

What is the impact of the CVE-2023-36612 vulnerability on the Basecamp Android app is that it allows an attacker to access files outside the application's sandbox.

What is the base score severity rating for CVE-2023-36612 is 7.5, which classifies it as HIGH.

On what date The CVE-2023-36612 vulnerability was published on 25 June 2023.

Where can you find More details about the CVE-2023-36612 vulnerability can be found on HackerOne's report at the following URL: [https://hackerone.com/reports/544444](#)

Can you describe One potential attack scenario exploiting CVE-2023-36612 could involve an attacker crafting a malicious file path.

Has the vulnerability identified by CVE-2023-36612 in the Basecamp com.basecamp.bc3 application for Android version 3.27.0.

What is the CVE ID for the vulnerability related to hard coded credentials in the Enphase Installer Toolkit is identified by the CVE-2023-32274.

Can you describe The vulnerability in Enphase Installer Toolkit version 3.27.0 involves hard-coded credentials embedded in the application.

What is the CVSS Base Score assigned to CVE-2023-32274 has been assigned a CVSS Base Score of 7.5, which is categorized as HIGH. This indicates a significant security risk.



When was The vulnerability CVE-2023-32274 was first published on 20 June 2023.

Where can Additional information and an advisory related to CVE-2023-32274 can be found at the following URL:

What pote The hard-coded credential vulnerability CVE-2023-32274 could enable attackers to gain unauthorized

Is there a c Since CVE-2023-32274 is about a hard-coded credential in an Android application binary, there isn't a

What are s Possible attack scenarios due to CVE-2023-32274 might include an attacker decompiling the APK file c

What is CV CVE-2023-29546 is a security vulnerability present in Firefox for Android where when recording the sc

Which vers CVE-2023-29546 affects Firefox for Android versions earlier than 112. Users who are using these outd

How can tr To mitigate CVE-2023-29546, users should update their Firefox for Android and Focus for Android to v

What type CVE-2023-29546 is classified as a privacy vulnerability with a CVSS Base Score of 6.5, indicating a med

Are other c No, other operating systems are not affected by CVE-2023-29546. This vulnerability is specific to Firef

What was t The security advisory for CVE-2023-29546 was officially released on 19 June 2023, as per the advisorie

Can you pr In an attack scenario exploiting CVE-2023-29546, an attacker could trick a user into screen recording t

Where can More information about CVE-2023-29546 can be found by visiting the Mozilla security advisory page a

Is there a c Providing a code example for CVE-2023-29546 is not applicable in this context, as the issue resides in t

What is CV CVE-2023-29534 refers to a security vulnerability that was identified in Firefox and Focus for Android

What versi Firefox for Android versions prior to 112 are affected by CVE-2023-29534.

How sever The vulnerability described by CVE-2023-29534 is considered CRITICAL, with a base score of 9.1. This i

When was CVE-2023-29534 was published on 19 June 2023.

Which Mo: Other versions of Firefox, apart from Firefox for Android, are unaffected by CVE-2023-29534. This vuln

Where can More information about CVE-2023-29534 can be found through various references such as the Mozill

Can you pr An attack scenario for CVE-2023-29534 could involve an attacker creating a malicious website that use

What is CV CVE-2023-25747 is a security vulnerability identified in the libaudio component, which was found to h

Which vers CVE-2023-25747 affects Firefox for Android versions earlier than 110.1.0. Users who are running a ver

What is the The Common Vulnerability Scoring System (CVSS) Base Score for CVE-2023-25747 is 7.5, which is class

How has tr The vulnerability in CVE-2023-25747 has been mitigated by disabling the AAudio backend in the libau

What is the CVE-2023-25747 was published on 19 June 2023.

Can you pr Yes, additional information about CVE-2023-25747 can be found at the following URLs: Mozilla's Secu

What are s Possible attack scenarios for CVE-2023-25747 include an attacker exploiting the use-after-free vulnera

Are there a As CVE-2023-25747 is related to a use-after-free vulnerability, code examples demonstrating the exac

What is CV CVE-2023-25645 is a security vulnerability identified in some ZTE AndroidTV Set-Top Boxes (STBs). It i

How sever The vulnerability symbolized by CVE-2023-25645 has been assessed with a base score of 7.7 and categ

When was The CVE-2023-25645 vulnerability was publicly disclosed on the 16th of June, 2023.

Where can Detailed information about CVE-2023-25645 can be found on ZTE's official support website. The direc

What can a By exploiting CVE-2023-25645, an attacker could potentially execute privileged functions without prop

Can you pr In a potential attack scenario for CVE-2023-25645, a malicious application without proper privileges c

What is CV CVE-2023-21144 is a security vulnerability identified in the doInBackground method of NotificationCo

Which And The Android versions affected by CVE-2023-21144 include Android 11, Android 12, Android 12L (which

What is the The CVSS base score of CVE-2023-21144 is 7.5, which is classified as HIGH severity.

Was a patc Yes, a patch has been issued for CVE-2023-21144. You can find more detailed information in the relat

What is the CVE-2023-21144 was published on 15 June 2023.

Can you de An attack scenario exploiting CVE-2023-21144 might involve a malicious actor crafting a special notific

Do I need t No, exploitation of CVE-2023-21144 does not require user interaction, which increases the risk since t

Are there a Typically, code examples that demonstrate how to exploit a vulnerability like CVE-2023-21144 are not

What is CV CVE-2023-21143 refers to a security vulnerability in multiple functions of multiple files within Androi

How sever The vulnerability described by CVE-2023-21143 has been assigned a Base Score of 5.5, which is categc

What versi CVE-2023-21143 affects multiple Android versions including Android 11, Android 12, Android 12L, and

When was CVE-2023-21143 was published on 15 June 2023. It is important for users and administrators to review

Where can More information about CVE-2023-21143 can be found on the Android security bulletin page at <https://source.android.com/security/bulletin/2023-06-15>:  
Can you de A possible attack scenario exploiting CVE-2023-21143 could involve an attacker developing a malicious app that exploits the vulnerability.  
What are t To exploit CVE-2023-21143, an attacker would typically not need any additional execution privileges a  
What is th The CVE ID for the vulnerability is CVE-2023-21142.

What type CVE-2023-21142 describes a permissions bypass vulnerability that could lead to local information disclosure.  
What versi The versions of Android affected by CVE-2023-21142 include Android 11, Android 12, Android 12L, and Android 13.  
What is th The CVSS base score given to CVE-2023-21142 is 5.5, and its severity rating is MEDIUM.

Is user inte No, user interaction is not needed to exploit the vulnerability in CVE-2023-21142.

When was CVE-2023-21142 was published on 15 June 2023.

Where can More information about CVE-2023-21142 can be found at: <https://source.android.com/security/bulletin/2023-06-15>

Can you ex An attack scenario for exploiting CVE-2023-21142 could involve a malicious app installed on the device.

What is th The CVE ID for the reported security vulnerability is CVE-2023-21141.

Can you de CVE-2023-21141 refers to a permissions bypass in several functions of multiple files within Android. This vulnerability affects Android versions 11 through 13.

What is th CVE-2023-21141 has been given a Base Score of 5.5, which is classified as MEDIUM severity. This indicates a moderate risk of exploitation.

Which vers The versions of Android affected by CVE-2023-21141 include Android 11, Android 12, Android 12L, and Android 13.

What could Exploiting CVE-2023-21141 could lead to local information disclosure. An attacker with physical or privileged access to the device could exploit this vulnerability.

Was there No, CVE-2023-21141 does not require any user interaction for exploitation, making it easier for an attacker to exploit.

Where can More detailed information about CVE-2023-21141 can be found in the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-06-15>

What is th CVE-2023-21141 was published on 15 June 2023.

How can u Users can mitigate the risk associated with CVE-2023-21141 by applying security updates provided by the device manufacturer.

Are there a Due to the nature of CVE-2023-21141, a potential attack scenario involves an attacker with local access to the device.

What is CV CVE-2023-21139 refers to a security vulnerability identified in the MediaControlPanel.java component of the Android framework.

How sever The CVE-2023-21139 vulnerability has been assigned a Base Score of 7.8, which is classified as HIGH severity. This indicates a high risk of exploitation.

Was there Details about patches or updates for CVE-2023-21139 can typically be found in the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-06-15>

What versi CVE-2023-21139 affects Android version 13. Users with devices running this version are recommended to update to the latest version.

Does explo No, the exploitation of CVE-2023-21139 does not require user interaction. The vulnerability could allow an attacker to exploit the device without user interaction.

Can you gi An example attack scenario for CVE-2023-21139 could involve a malicious application installed on the device that exploits the vulnerability.

What is th The official Android ID associated with CVE-2023-21139 is A-271845008. This identifier can be used to track the vulnerability across different versions of the Android framework.

When was CVE-2023-21139 was published on 15 June 2023. After this date, vendors and developers should have been aware of the vulnerability and taken steps to address it.

What is th The CVE ID for the Android security vulnerability involving improper input validation in CallRedirectionProcessor.java is CVE-2023-21138.

Can you de CVE-2023-21138 refers to a vulnerability in the onNullBinding method of CallRedirectionProcessor.java, which is used for handling null values in the Android framework.

What versi The versions of Android affected by CVE-2023-21138 include Android 11, Android 12, Android 12L, and Android 13.

What is th The CVSS Base Score assigned to CVE-2023-21138 is 7.8, which is categorized as HIGH severity. This indicates a high risk of exploitation.

On what d CVE-2023-21138 was published on 15 June 2023.

Where can More information regarding CVE-2023-21138 can be found in the Android security bulletin at the following URL: <https://source.android.com/security/bulletin/2023-06-15>

Does CVE-2 No, CVE-2023-21138 does not require user interaction for exploitation. The vulnerability can be exploited by an attacker with physical or privileged access to the device.

What are t If an attacker successfully exploits CVE-2023-21138, they could achieve local escalation of privilege on the device.

Could you An example of an attack scenario involving CVE-2023-21138 might involve a malicious app that, once installed, exploits the vulnerability to gain elevated privileges.

What is th The CVE ID of the vulnerability is CVE-2023-21137.

Can you de CVE-2023-21137 refers to a vulnerability found in several methods of JobStore.java. Due to uncaught exceptions, an attacker could potentially cause a denial of service or other adverse effects.

Which vers The affected versions of Android by CVE-2023-21137 include Android-11, Android-12, Android-12L, and Android-13.

What is th The base score assigned to CVE-2023-21137 is 5.5, which is categorized as MEDIUM severity. This score indicates a moderate risk of exploitation.

When was CVE-2023-21137 was published on 15 June 2023.

Are there a Yes, more information about CVE-2023-21137 can be found at the following URL: <https://source.android.com/security/bulletin/2023-06-15>

What kind No user interaction is required to exploit the vulnerability described by CVE-2023-21137.

Could you A possible attack scenario for CVE-2023-21137 could involve a malicious app that triggers the uncaught exception, leading to a denial of service or other adverse effects.

What is CV CVE-2023-21136 is a security vulnerability identified in multiple functions of the JobStore.java file within the Android framework.

Which And CVE-2023-21136 affects Android versions 11, 12, 12L, and 13.

What is the The base score assigned to CVE-2023-21136 is 5.5, which is categorized as MEDIUM severity.

Does CVE- No, CVE-2023-21136 does not require user interaction for exploitation.

What type No additional execution privileges are needed to exploit CVE-2023-21136.

When was CVE-2023-21136 was published on 15 June 2023.

Where can More information about CVE-2023-21136 can be found at the Android Security Bulletin webpage: <http://s>

What is the The main impact of exploiting CVE-2023-21136 is a local denial of service, where the device can crash

Can you de A possible attack scenario for CVE-2023-21136 would involve an attacker creating or manipulating inp

What is the The CVE ID of the vulnerability found in NotificationAccessSettings.java in Android is CVE-2023-21135

Can you de CVE-2023-21135 is associated with a potential failure to persist notifications settings within Android's

What are t The Android versions affected by CVE-2023-21135 include Android 11, Android 12, Android 12L, and A

What is the The base score assigned to CVE-2023-21135 is 7.8, which is categorized as HIGH.

When was CVE-2023-21135 was published on 15 June 2023.

Where can Additional details about CVE-2023-21135 can be found at the Android Security Bulletin page: <https://s>

Is user inte No, user interaction is not required to exploit the vulnerability described by CVE-2023-21135.

What type CVE-2023-21135 could lead to a local escalation of privilege on an Android device.

Could you A possible attack scenario for CVE-2023-21135 involves an attacker exploiting the improper input vali

What is CV CVE-2023-21131 is a security vulnerability identified in the ActivityManagerService.java of Android, s

How sever CVE-2023-21131 has been rated with a base score of 7.8, which is categorized as HIGH severity. This i

On what d CVE-2023-21131 was published on 15 June 2023.

Does explo No, exploiting CVE-2023-21131 does not require user interaction. An attacker could potentially exploi

What versi The Android versions affected by CVE-2023-21131 are Android 11, Android 12, Android 12L, and Andr

Where can More information about CVE-2023-21131 can be found in the Android Security Bulletin at the followin

What is the The Android ID associated with CVE-2023-21131 is A-265015796.

What pote If successfully exploited, CVE-2023-21131 could lead to a local escalation of privilege, enabling an atta

Can you pr A possible attack scenario involving CVE-2023-21131 could involve a malicious app installed on the de

What is CV CVE-2023-21130 is a vulnerability that was found in the file btm\_ble\_periodic\_adv\_sync\_lost of btm\_

How sever CVE-2023-21130 has been given a base score of 9.8, which classifies it as CRITICAL in severity. This ind

What versi Android version 13 is affected by CVE-2023-21130.

Is user inte No, user interaction is not needed to exploit CVE-2023-21130. This means an attack could be fully rerr

Where can Additional information about CVE-2023-21130 can be found on the Android Security Bulletin at the fo

When was CVE-2023-21130 was published on 15 June 2023.

What are s Given that CVE-2023-21130 is a buffer overflow vulnerability, a potential attack scenario would involv

What is the The CVE ID for the vulnerability affecting the getFullScreenIntentDecision process in Android is CVE-20

Which vers CVE-2023-21129 impacts Android versions 11, 12, 12L, and 13.

What is the The base score of CVE-2023-21129 is rated as 7.8, which is categorized as HIGH severity.

What is the CVE-2023-21129 represents a type of vulnerability that allows for local escalation of privilege without

When was CVE-2023-21129 was published on 15 June 2023.

Does CVE- Yes, CVE-2023-21129 requires user interaction for exploitation.

Where can More information about CVE-2023-21129 can be found in the Android Security Bulletin at this URL: <ht>

Can you ex A possible attack scenario for CVE-2023-21129 could involve a malicious application waiting for a user

What is the The CVE ID of the vulnerability affecting Android's AppStandbyController.java is CVE-2023-21128.

What versi The versions of Android impacted by CVE-2023-21128 include Android 11, Android 12, Android 12L, a

What is the The CVSS base score of CVE-2023-21128 is 7.8, which is considered HIGH.

What is the CVE-2023-21128 describes a vulnerability due to a logic error in various functions of AppStandbyCont

When was CVE-2023-21128 was published on 15 June 2023.

Could you More information about CVE-2023-21128 can be found at the Android Security Bulletin page at <https;>

Are there any? Typically, specific code examples demonstrating the vulnerability are not provided publicly to prevent exploitation. What kind of vulnerability? CVE-2023-21128 allows an attacker to exploit a logic error in the AppStandbyController.java to escalate privileges. What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21127.

Can you provide details? CVE-2023-21127 is a vulnerability in the readSampleData function of NuMediaExtractor.cpp, where the attacker can trigger a local privilege escalation. What is the CVSS base score for CVE-2023-21127? The CVSS base score for CVE-2023-21127 is 8.8, and it is classified as HIGH severity.

When was it published? The CVE-2023-21127 vulnerability was published on 15 June 2023.

Which Android versions are affected? The Android versions affected by CVE-2023-21127 are Android-11, Android-12, Android-12L, and Android-13.

Where can I find more information? More information about the CVE-2023-21127 vulnerability can be found at the Android Security Bulletin.

What are the potential attack scenarios? Potential attack scenarios for CVE-2023-21127 include a remote attacker enticing a user to open a crafted application.

What is the Android ID assigned to the vulnerability? The Android ID assigned to the CVE-2023-21127 vulnerability is A-275418191.

What is CVE-2023-21126? CVE-2023-21126 refers to a security vulnerability identified in the Android operating system. Specifically, it is a local privilege escalation vulnerability. Which versions are affected? CVE-2023-21126 affects devices running Android version 13.

How severe is it? CVE-2023-21126 has been assigned a base score of 7.8, which is categorized as HIGH severity according to the CVSS.

Does it require user interaction? No, user interaction is not required for the exploitation of CVE-2023-21126, making it more dangerous.

When was it published? CVE-2023-21126 was published on 15 June 2023.

Where can I find more details? More details about CVE-2023-21126 can be found on the Android Security Bulletin webpage at <https://source.android.com/security/bulletin/2023-06>.

Can you give an attack scenario? An attack scenario for CVE-2023-21126 could involve a malicious application that is able to broadcast a crafted intent.

What are the mitigation measures? Mitigation measures for CVE-2023-21126 include applying the security updates provided by the Android manufacturer.

What is the impact? The impact of CVE-2023-21126 is a local escalation of privilege which can allow attackers to perform actions that require higher privileges.

Are there any known advisories? While specific code examples illustrating the exploit for CVE-2023-21126 are generally not released in public, there are known advisories.

What is CVE-2023-21124? CVE-2023-21124 is a security vulnerability in certain versions of the Android operating system, specifically affecting the Binder interface.

How severe is it? CVE-2023-21124 has been assigned a base score of 7.8, which is classified as HIGH severity. This indicates a significant risk of exploitation.

When was it published? CVE-2023-21124 was published on the 15th of June, 2023.

Are there any known advisories? Yes, there is at least one known advisory for CVE-2023-21124, which can be found at the Android Security Bulletin.

What type of vulnerability is it? The systems affected by CVE-2023-21124 are devices running the Android operating system, particularly those using the Binder interface.

Can you explain the attack scenario? An attacker exploiting CVE-2023-21124 could potentially execute a local privilege escalation on a victim's device.

What should users do? Users should apply any provided patches or updates from their device manufacturers or service providers as soon as they are available.

What is the CVE ID associated with the vulnerability? The CVE ID associated with the vulnerability is CVE-2023-21123.

Which Android versions are affected? The Android versions affected by CVE-2023-21123 are Android-11, Android-12, Android-12L, and Android-13.

What type of vulnerability is it? CVE-2023-21123 allows local escalation of privilege due to a missing permission check in multiple functions.

Does it require user interaction? No, user interaction is not needed to exploit the vulnerability described in CVE-2023-21123.

What is the CVSS base score of CVE-2023-21123? The CVSS base score of CVE-2023-21123 is 7.8, which is categorized as HIGH severity.

When was it published? CVE-2023-21123 was published on 15 June 2023.

Where can I find more information? More information about CVE-2023-21123 can be found on the Android Security Bulletin page, specifically under the section for June 2023.

Can you provide an attack scenario? CVE-2023-21123 can be exploited by a malicious local application. The app could attempt to trace other applications.

What are the attack scenarios? Attack scenarios for CVE-2023-21123 include a malicious app gaining unauthorized access to trace other applications.

What is CVE-2023-21122? CVE-2023-21122 is a security vulnerability identified in various functions of multiple files within the Android operating system.

What is the base score assigned to CVE-2023-21122? The base score assigned to CVE-2023-21122 is 7.8, which is categorized as HIGH severity.

Which Android versions are affected? CVE-2023-21122 affects Android versions 11, 12, 12L, and 13.

Does it require user interaction? No, user interaction is not required to exploit the vulnerability described in CVE-2023-21122.

When was it published? CVE-2023-21122 was published on 15 June 2023.

Where can I find more information? More information about CVE-2023-21122 can be found at the official Android security bulletin link: <https://source.android.com/security/bulletin/2023-06>.

What are the attack scenarios? Attack scenarios for CVE-2023-21122 could involve a malicious app or actor exploiting the missing permission check.

What kind of vulnerability is it? CVE-2023-21122 could potentially lead to a local escalation of privilege on the affected Android device.

What is the CVE ID for the vulnerability that affects the onResume method in AppManagementFragment.java? The CVE ID for the vulnerability that affects the onResume method in AppManagementFragment.java is CVE-2023-21121.

What are the affected Android versions? The Android versions affected by CVE-2023-21121 are Android 11 and Android 12.

What is the base score of CVE-2023-21121? The base score of CVE-2023-21121 is 7.8, which is rated as HIGH.

Does CVE-21 No, user interaction is not required for the exploitation of CVE-2023-21121.

What type CVE-2023-21121 could lead to local escalation of privilege on an Android device.

What addit No additional execution privileges are needed to exploit CVE-2023-21121.

On what d CVE-2023-21121 was published on 15 June 2023.

Can you pr For more information on CVE-2023-21121, you can visit the Android security bulletin page: <https://source.android.com/security/bulletin/2023-06-15>

What is the The vulnerability stated in CVE-2023-21121 is due to improper input validation in the onResume method.

What might CVE-2023-21121 might allow a malicious actor to prevent users from forgetting a previously connected device.

Can you describe A possible attack scenario for exploiting CVE-2023-21121 could involve an attacker installing a rogue app.

What is the The CVE ID for the use-after-free vulnerability affecting Android is CVE-2023-21120.

What kind CVE-2023-21120 describes a possible use-after-free vulnerability due to improper locking in multiple functions.

What could Exploiting the vulnerability CVE-2023-21120 could lead to local escalation of privilege on an affected device.

Is user interaction No, user interaction is not required for the exploitation of the vulnerability tracked as CVE-2023-21120.

What are the CVE-2023-21120 affects certain versions of Android SoC, but the specific affected versions are not detailed.

How severe The severity of the vulnerability CVE-2023-21120 is rated as 7.8, which is considered HIGH.

On what date The CVE-2023-21120 vulnerability was published on 15 June 2023.

Where can More information about CVE-2023-21120 can be found in the official Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-06-15>

Can you give An example attack scenario for CVE-2023-21120 could involve a malicious app that takes advantage of the vulnerability.

What are the To mitigate the CVE-2023-21120 vulnerability, users should apply updates provided by the Android Security Center.

What is the The CVE ID for the vulnerability related to an encryption downgrade in btm\_sec.cc is CVE-2023-21115.

What is the The base severity score of CVE-2023-21115 is 8.8, which is classified as HIGH.

Which And The Android versions affected by CVE-2023-21115 are Android-11, Android-12, and Android-12L.

Can you explain CVE-2023-21115 pertains to an issue in the btm\_sec\_encrypt\_change function within btm\_sec.cc, which allows an attacker to downgrade the encryption.

What are the Potential implications of the CVE-2023-21115 vulnerability include an attacker being able to downgrade the encryption.

What is the CVE-2023-21115 was published on 15 June 2023.

Where can More information and reference material about CVE-2023-21115 can be found on the Android Security Center.

Are there any As a security best practice, we do not share existing exploits or vulnerable code examples. However, we do share the CVE details.

Does CVE-21 No, user interaction is not needed for the exploitation of CVE-2023-21115.

How can users To protect their devices from the CVE-2023-21115 vulnerability, users should ensure that they apply updates provided by the Android Security Center.

What is the The CVE ID of the vulnerability is CVE-2023-21108.

Can you describe CVE-2023-21108 refers to a vulnerability in the sdp\_build\_uuid\_seq function of sdp\_discovery.cc within the Android SoC.

What Android The Android versions affected by CVE-2023-21108 are Android 11, Android 12, Android 12L (Android 12L), and Android 13.

What severity CVE-2023-21108 has been assigned a Base Score of 8.8, which is categorized as HIGH severity.

On what date CVE-2023-21108 was published on 15 June 2023.

Where can More information regarding CVE-2023-21108 can be found at the following reference URL: <https://source.android.com/security/bulletin/2023-06-15>

What kind No user interaction is needed for the exploitation of CVE-2023-21108. This means that an attacker can exploit the vulnerability without the need for user interaction.

Could you describe A possible attack scenario could involve an attacker who is within Bluetooth range of a vulnerable device.

What is CVE-2023-21105 is a vulnerability identified in multiple functions of ChooserActivity.java in Android, which allows an attacker to bypass the confirmation dialog.

How severe The vulnerability referenced by CVE-2023-21105 has a Base Score of 5.5, which is classified as MEDIUM severity.

Which And CVE-2023-21105 affects Android versions 11, 12, 12L, and 13. Devices running these versions are vulnerable.

What are the In potential attack scenarios for CVE-2023-21105, a malicious application could exploit the confused state of the device.

Was there Yes, there was a public disclosure for CVE-2023-21105. The details about the vulnerability were published in the Android Security Bulletin.

When was CVE-2023-21105 was publicly disclosed on 15 June 2023, as indicated by the published date in the Android Security Bulletin.

Does CVE-21 No, CVE-2023-21105 does not require user interaction for exploitation. The vulnerability can be exploited without the need for user interaction.

What is the The CVE ID for the vulnerability found in multiple functions of WVDrmPlugin.cpp is CVE-2023-21101.

What type CVE-2023-21101 refers to a possible use after free due to a race condition in multiple functions of WVDrmPlugin.cpp.

Is user interaction No, user interaction is not needed for the exploitation of the vulnerability described in CVE-2023-21101.

What product The product affected by CVE-2023-21101 is Android, specifically versions on Android SoC (System on Chip).

On which c CVE-2023-21101 was published on 15 June 2023.

What is the The base score assigned to CVE-2023-21101 is 7.0, which is categorized as HIGH.

Where can More details or the official advisory about CVE-2023-21101 can be found at the Android Security Bulletin

Can you el Potential attack scenarios for CVE-2023-21101 involve an attacker exploiting the race condition to per

What is the The CVE ID of the reported vulnerability in the Android system is CVE-2023-21095.

Can you pr CVE-2023-21095 is associated with a potential partial lockscreen bypass due to a race condition in the

What is the The severity base score of CVE-2023-21095 is rated as 4.7, which is categorized as MEDIUM.

When was CVE-2023-21095 was published on the 15th of June, 2023.

Which vers The versions of Android affected by CVE-2023-21095 are Android-12L and Android-13.

Is user inte No, user interaction is not required to exploit the vulnerability described in CVE-2023-21095.

Where can More details about the CVE-2023-21095 vulnerability can be found in the Android Security Bulletin at

What is the The Android ID associated with CVE-2023-21095 is A-242704576.

Could you A possible attack scenario for CVE-2023-21095 might involve an attacker identifying the race condition

Are there a As CVE-2023-21095 pertains to a race condition and potential security vulnerability within Android's s

What is the The CVE ID for the vulnerability is CVE-2023-2976.

What versi CVE-2023-2976 affects Google Guava versions from 1.0 to 31.1.

Which clas The class associated with the vulnerability CVE-2023-2976 is `FileBackedOutputStream`.

On which p CVE-2023-2976 manifests on Unix systems and Android Ice Cream Sandwich.

What is the The base score of CVE-2023-2976 is 7.1, which is classified as HIGH severity.

What does CVE-2023-2976 allows other users and apps with access to the default Java temporary directory to be

Has CVE-2( Yes, the security vulnerability CVE-2023-2976 has been fixed in version 32.0.0 of Google Guava.

Why is it re It is recommended to use version 32.0.1 of Google Guava because version 32.0.0 breaks some functio

When was CVE-2023-2976 was published on 14 June 2023.

Where can You can find more information about CVE-2023-2976 from the following references:- <https://github.cc>

Can you pr Certainly. Here's a simple code example that shows how CVE-2023-2976 can manifest using `FileBacke

What are p Possible attack scenarios for CVE-2023-2976 include a local malicious user or application exploiting th

What is CV CVE-2023-29501 is a security vulnerability identified in the Jiyu Kukan Toku-Toku coupon App for both

How sever The vulnerability identified by CVE-2023-29501 has been assigned a Base Score of 4.8, which categori

On which c CVE-2023-29501 was published on 13 June 2023.

Which vers The versions of the Jiyu Kukan Toku-Toku coupon App that are affected by CVE-2023-29501 are 3.5.0 ;

What could If CVE-2023-29501 is exploited, the consequences could include a man-in-the-middle attack where an

Where can More information about CVE-2023-29501 can be found in the references provided with the CVE. This i

What is on A possible attack scenario for CVE-2023-29501 involves an attacker setting up a malicious Wi-Fi hotsp

Can you pr While the specific implementation details that led to CVE-2023-29501 are not provided, a general cod

What is CV CVE-2023-29753 is a vulnerability identified in the Facemoji Emoji Keyboard application version 2.9.1.

How sever The security issue represented by CVE-2023-29753 has been categorized with a Base Score of 5.5, whi

On what d CVE-2023-29753 was published on 09 June 2023.

Which plat The platform affected by CVE-2023-29753 is the Android operating system, specifically devices runnin

Where can More detailed information about CVE-2023-29753 can be found at the following URL: <https://github.c>

What kind CVE-2023-29753 can be exploited to perform a denial of service attack. An attacker with local access t

Can you gi A possible attack scenario for CVE-2023-29753 could involve an attacker with physical access or malw

What is CV CVE-2023-29751 refers to a security vulnerability found in Yandex Navigator version 6.60 for Android,

How sever The severity of CVE-2023-29751 is rated as 'MEDIUM' with a base score of 5.5, indicating that it repre

On what d CVE-2023-29751 was published on the 09th of June, 2023.

What type CVE-2023-29751 makes it possible to carry out a persistent denial of service attack on the Yandex Nav

Are there a Yes, detailed information about CVE-2023-29751 can be found at the following URL: <https://github.co>

What could Exploiting CVE-2023-29751 could potentially allow a malicious application to cause a denial of service

Can you pr An example attack scenario for CVE-2023-29751 could involve a rogue application on an Android device.  
What is the CVE ID for the vulnerability found in CrossX v.1.15.3 for Android is CVE-2023-29767.

What type CVE-2023-29767 is a security issue that allows a local attacker to cause a persistent denial of service.  
What is the severity score of CVE-2023-29767 is rated as 5.5, which classifies it as a MEDIUM severity vulnerability.  
When was CVE-2023-29767 was published on 09 June 2023.

Where can More details about CVE-2023-29767 can be found at the following URL: <https://github.com/LianKee/SecurityAdvisories>  
Can you de A possible attack scenario for CVE-2023-29767 involves a local attacker gaining access to the device.

Could you While specific code examples related to CVE-2023-29767 are not provided in the initial information, a  
What is CV CVE-2023-29766 is a security vulnerability in CrossX v.1.15.3 for Android that allows a local attacker to  
How severe The severity of CVE-2023-29766 is rated as '7.8 HIGH' on the Common Vulnerability Scoring System (CVSS).  
When was CVE-2023-29766 was published on 09 June 2023.

Could you A possible attack scenario for CVE-2023-29766 involves a local attacker gaining unauthorized access to  
What kind CVE-2023-29766 can lead to an escalation of privileges where a local attacker might gain elevated permissions.  
Where can More information about CVE-2023-29766 can be found at the referenced URL: <https://github.com/LianKee/SecurityAdvisories>

What is CV CVE-2023-29761 refers to a security vulnerability found in Sleep v.20230303 for Android. It allows unauthorized access.  
What is the severity score of CVE-2023-29761 is rated as 5.5, which categorizes it as a MEDIUM level of severity.  
When was The CVE-2023-29761 was published on 09 June 2023.

Where can More detailed information about CVE-2023-29761 can be found at the provided URL: <https://github.com/LianKee/SecurityAdvisories>  
Could you An attacker could exploit CVE-2023-29761 by creating a malicious app that, once installed on the device,

Can you pr While exact code which is vulnerable to CVE-2023-29761 is not provided, a general example of vulnerability is provided.  
What is CV CVE-2023-29759 refers to a security vulnerability found in FlightAware version 5.8.0 for Android, which allows unauthorized access.  
What is the base score assigned to CVE-2023-29759 is 5.5, which is categorized as MEDIUM severity.  
On what date CVE-2023-29759 was published on 09 June 2023.

Where can More information about CVE-2023-29759 can be found at the following URL: <https://github.com/LianKee/SecurityAdvisories>  
What type CVE-2023-29759 is a denial of service (DoS) vulnerability.

What might An attack exploiting CVE-2023-29759 could involve an unauthorized application gaining access to FlightAware.  
What is CV CVE-2023-29758 is a vulnerability found in Blue Light Filter version 1.5.5 for Android. It allows unauthorized access.  
What type The security issue associated with CVE-2023-29758 is a persistent denial of service attack that could be triggered.  
How severe CVE-2023-29758 is rated with a base score of 5.5, which is categorized as MEDIUM severity.

On what date CVE-2023-29758 was published on 09 June 2023.

Can you pr An attacker could craft a malicious app that, once installed on the device, could manipulate the SharedPreferences.  
Where can More detailed information about CVE-2023-29758 can be found at the following URL: <https://github.com/LianKee/SecurityAdvisories>

What kind Applications on Android that use Blue Light Filter version 1.5.5 are at risk due to CVE-2023-29758. Specifically,  
Is there a code example for CVE-2023-29758 would showcase how a malicious app manipulates SharedPreferences.

What is CV CVE-2023-29757 refers to a security issue discovered in Blue Light Filter version 1.5.5 for Android. It is a denial of service.  
How severe The vulnerability identified as CVE-2023-29757 has been given a base score of 7.8, which is classified as HIGH severity.  
What is the main impact of CVE-2023-29757 on users' devices is that it allows attackers to escalate their privileges.  
When was CVE-2023-29757 was published on 09 June 2023.

Where can More information about CVE-2023-29757 can be found at the following URL: <https://github.com/LianKee/SecurityAdvisories>  
What kind Using CVE-2023-29757, an attacker could perform an escalation of privilege attack. This means the attacker can gain higher permissions.

Can you pr While a specific code example is not provided, an attack exploiting CVE-2023-29757 might involve a rogue application.  
What are SharedPreferences files in Android are used to store key-value pairs of primitive data types in a private storage.  
What steps Users should update the Blue Light Filter app to the latest version that patches the CVE-2023-29757 vulnerability.  
Should developers of other Android apps be aware of vulnerabilities like CVE-2023-29757 as it depends on the same library.  
What is the CVE ID for the vulnerability found in Twilight for Android is CVE-2023-29756.

Can you de CVE-2023-29756 describes a security issue in Twilight v.13.3 for Android where unauthorized applications can access sensitive data.  
What severity CVE-2023-29756 has been assigned a severity level of 5.5, which is categorized as 'MEDIUM'.

When was CVE-2023-29756 published on 09 June 2023.

Where can More detailed information about CVE-2023-29756 can be found at the following URL: <https://github.com>

What kind Unauthorized applications, which are not granted permission to interfere with Twilight's operation, can

What is the A successful exploitation of CVE-2023-29756 could result in a persistent denial of service, where the ir

Could you An example attack scenario using CVE-2023-29756 could involve a malicious app that is installed on th

What is CV CVE-2023-29755 is a security vulnerability found in Twilight v.13.3 for Android. This vulnerability allow

How sever The CVE-2023-29755 vulnerability has been assessed with a Base Score of 7.8, which categorizes it as

When was CVE-2023-29755 published on 09 June 2023.

What com CVE-2023-29755 affects the SharedPreferences files in Twilight v.13.3 for Android.

Where can More information regarding CVE-2023-29755 can be found on the provided reference link: <https://git>

Can you ex A possible attack scenario for CVE-2023-29755 could involve a malicious app installed on the same An

Could you While an exact code example for the exploitation of CVE-2023-29755 would depend on the specifics of

What is CV CVE-2023-29752 refers to a security issue in the Facemoji Emoji Keyboard application version 2.9.1.2 f

How sever The vulnerability CVE-2023-29752 is considered to have a 'HIGH' severity level with a Base Score of 7.8

When was CVE-2023-29752 published on 09 June 2023.

Where can More details about CVE-2023-29752 can be found at the provided GitHub link: <https://github.com/Lia>

What kind CVE-2023-29752 is an escalation of privilege vulnerability, which typically involves a lower-privileged

What are p An attacker could exploit CVE-2023-29752 by creating a malicious app that takes advantage of the vul

What is CV CVE-2023-29749 refers to a security vulnerability discovered in Yandex Navigator version 6.60 for Anc

How sever The security issue associated with CVE-2023-29749 is considered to have a severity rating of 7.8, whic

When was The vulnerability CVE-2023-29749 was published on 09 June 2023.

Where can Detailed information about the CVE-2023-29749 vulnerability can be found at the following URL: <http>

What pote Possible attack scenarios associated with CVE-2023-29749 include an unauthorized application on the

What is the The impact of CVE-2023-29749 on Android devices includes the risk of unauthorized applications esca

Can you pr While it is not ethical or legal to provide an actual code example for exploiting a vulnerability such as

What shou Users of Yandex Navigator should ensure they update the application to the latest version where this

What is the The CVE ID for the memory corruption issue in Linux Android is CVE-2022-33227.

What type CVE-2022-33227 reports a memory corruption vulnerability in Linux Android caused by a double free

How sever The vulnerability described in CVE-2022-33227 has been given a base score of 7.8, which is classified a

When was CVE-2022-33227 published on 06 June 2023.

Are there a Yes, detailed information regarding CVE-2022-33227 can be found in the Qualcomm Product Security

Can you pr As CVE-2022-33227 pertains to a double free vulnerability, I cannot provide an exact code example wit

What are s Attack scenarios for CVE-2022-33227 might include a local attacker exploiting the double free vulnera

What is the The CVE ID for the memory safety vulnerabilities identified in Firefox is CVE-2023-29551.

Can you de CVE-2023-29551 represents a set of memory safety bugs found in Firefox version 111. The bugs indica

Which Mo The CVE-2023-29551 vulnerability affects Firefox for Android versions prior to 112, Firefox desktop ve

What is the The Base Score for CVE-2023-29551 is 8.8, which is rated as HIGH severity.

When was The CVE-2023-29551 vulnerability was published on 02 June 2023.

Where can More information about CVE-2023-29551 can be found in the Mozilla Foundation Security Advisory at

What are t Possible attack scenarios for CVE-2023-29551 include an attacker crafting a malicious web page that,

Are there a Actual exploitation code examples for CVE-2023-29551 would be both unethical and dangerous to sha

What is the The CVE ID for the memory safety vulnerability in Firefox and Firefox ESR is CVE-2023-29550.

Which vers CVE-2023-29550 affects Firefox versions before 112, Firefox ESR versions before 102.10, Focus for Anc

What is the The Common Vulnerability Scoring System (CVSS) Base Score for CVE-2023-29550 is 8.8, which is rated

When was The security vulnerability CVE-2023-29550 was publicly disclosed on 02 June 2023.

What can t The memory safety bugs in CVE-2023-29550 could potentially be exploited to run arbitrary code if an

Can you pr Yes, more information on CVE-2023-29550 can be found on the following URLs: - <https://www.mozilla>



What are p Possible attack scenarios for CVE-2023-29550 include an attacker crafting malicious web content that  
What is the CVE ID of the vulnerability is CVE-2023-29549.

Can you de CVE-2023-29549 is a security vulnerability that arises under certain conditions when the 'bind' functio  
Which Mo: CVE-2023-29549 affects Firefox for Android versions earlier than 112, Firefox browser versions earlier  
What is the CVE ID of the vulnerability is CVE-2023-29549, with a base score of 6.5.

When was CVE-2023-29549 was published on 02 June 2023.

Where can More information about CVE-2023-29549 can be found in the Mozilla Foundation Security Advisory (M  
Could you Unfortunately, without details on the exact nature of the incorrect realm assignment in CVE-2023-295

What are t An attacker might exploit CVE-2023-29549 by executing code that takes advantage of the incorrect re  
What is the CVE ID for the vulnerability is CVE-2023-29548.

Which Mo: The Mozilla products affected by CVE-2023-29548 include Firefox versions prior to 112, Focus for And  
What is the CVE ID of the vulnerability is CVE-2023-29548, categorized as MEDIUM severity.

When was CVE-2023-29548 was published on 02 June 2023.

Can you pr Links to the Mozilla Security Advisories for CVE-2023-29548 include: 1. <https://www.mozilla.org/secu>  
Where can The bug report related to CVE-2023-29548 can be found at: [https://bugzilla.mozilla.org/show\\_bug.cgi](https://bugzilla.mozilla.org/show_bug.cgi)

What is the CVE ID of the vulnerability where a wrong lowering instruction in the ARM64 Ion comp  
Could you A possible attack scenario for CVE-2023-29548 would involve an attacker crafting a web page with spe  
What is the CVE ID for the vulnerability affecting Firefox and its related applications is CVE-2023-29547.

What are t CVE-2023-29547 allows for the creation of an insecure cookie when a secure cookie already exists for  
What appli CVE-2023-29547 affects Firefox for Android before version 112, Firefox before version 112, and Focus  
What is the CVE ID of the vulnerability is CVE-2023-29547, and it is rated as MEDIUM severity.

When was CVE-2023-29547 was published on 02 June 2023.

Where can More information or advisories about CVE-2023-29547 can be found at the official Mozilla Security Ac  
What are s Possible attack scenarios involving CVE-2023-29547 include an attacker leveraging the weakness to cr

Are there a Since CVE-2023-29547 involves the handling of cookies within the browser, the issue is not something  
What is CV CVE-2023-29544 is a vulnerability in Firefox for Android, Firefox, and Focus for Android where multipl  
Which vers CVE-2023-29544 affects versions of Firefox for Android prior to 112, Firefox desktop versions before 1  
What is the CVE ID of the vulnerability is CVE-2023-29544, which is categorized as MEDIUM.

When was CVE-2023-29544 was published on 02 June 2023.

Where can More information about CVE-2023-29544 can be found at the Mozilla Security Advisories webpage (ht  
What are t To mitigate CVE-2023-29544, users and administrators should update their Firefox for Android to vers

Can you de An attacker could potentially exploit CVE-2023-29544 by causing the application to handle multiple in  
Are there a As of my last update, there is no specific proof-of-concept code available to the public for CVE-2023-2

What is CV CVE-2023-29543 refers to a security vulnerability that was identified in Firefox for Android, Firefox for  
Which vers CVE-2023-29543 affects versions of Firefox for Android before 112, Firefox for desktop before version  
When was CVE-2023-29543 was published on 02 June 2023.

What base CVE-2023-29543 was assigned a base score of 8.8, which is categorized as HIGH according to the vulne  
Where can More information about CVE-2023-29543 can be found at the official Mozilla Bugzilla page at <https://>

What pote Possible attack scenarios for CVE-2023-29543 could involve an attacker crafting malicious web conten  
How can u: Users can protect themselves from CVE-2023-29543 by updating Firefox for Android, Firefox for desk

What is the CVE ID of the vulnerability affecting Firefox on certain Linux distributions concerning the handlin  
Can you de CVE-2023-29541 is a security issue in Firefox where the browser did not properly manage the downlo

Which Fire The versions of Firefox affected by CVE-2023-29541 include Firefox releases prior to version 112, Focu  
How sever The severity of CVE-2023-29541 is rated as HIGH with a base score of 8.8, indicating that it is consider

When was The CVE-2023-29541 vulnerability was published on 02 June 2023.

Where can More information about CVE-2023-29541 can be found through the following references: Mozilla's se  
What oper CVE-2023-29541 specifically affects Firefox for Linux on certain distributions; other operating systems

Explain a p A potential attack scenario exploiting CVE-2023-29541 could involve an attacker creating a malicious .

What is CV CVE-2023-29540 is a security vulnerability identified in several Mozilla products, including Firefox for .

What are t The vulnerability described by CVE-2023-29540 could allow an attacker to bypass expected navigation

How can C' To mitigate CVE-2023-29540, it is recommended to update the affected Mozilla applications to versio

Where can More detailed information about CVE-2023-29540 can be found in the official advisories and referenc

What is th The CVSS base score for CVE-2023-29540 is '6.1 MEDIUM'. This score indicates that the vulnerability is

What is an A possible attack scenario for CVE-2023-29540 involves an attacker creating a malicious website with

What is th The CVE ID for the vulnerability related to the filename directive in the Content-Disposition header is

What is th The base score assigned to CVE-2023-29539 is 8.8, which is considered HIGH severity.

Which Mo CVE-2023-29539 affects several Mozilla products including Firefox versions prior to 112, Focus for And

What is th CVE-2023-29539 is a vulnerability where the filename in the Content-Disposition header would be tru

When was CVE-2023-29539 was published on 02 June 2023.

Where can More information about CVE-2023-29539 can be found in the security advisories at the following URL

Can you pr A potential attack scenario for CVE-2023-29539 involves an attacker crafting a webpage that triggers a

Could you An attacker could exploit CVE-2023-29539 by crafting a malicious link or a script on a website that sets

What is th The CVE ID of the reported vulnerability is CVE-2023-29538.

Can you de CVE-2023-29538 is a security vulnerability where a WebExtension may incorrectly receive a `jar:file://

What versi The vulnerability CVE-2023-29538 affects versions of Firefox for Android prior to 112, Firefox browser

How sever The CVE-2023-29538 vulnerability has been assigned a Base Score of 4.3, categorizing it as a MEDIUM

When was CVE-2023-29538 was publicly disclosed on 02 June 2023.

Where can More information and details about CVE-2023-29538 can be found by visiting the references such as t

What migh A possible attack scenario for CVE-2023-29538 could involve a malicious website or a compromised W

What step To protect against the security issue referenced by CVE-2023-29538, users should update their Firefox

What is CV CVE-2023-29537 refers to a security vulnerability involving multiple race conditions in the font initiali

Which Mo The Mozilla products affected by CVE-2023-29537 include Firefox for Android prior to version 112, Fir

What is th The CVSS base score for CVE-2023-29537 is 7.5, which classifies it as HIGH severity.

When was CVE-2023-29537 was published on 02 June 2023.

Are there a Yes, more information about CVE-2023-29537 can be found at the following URLs: - <https://bugzilla.m>

Can you pr Attack scenarios for CVE-2023-29537 could involve an attacker crafting malicious web content that ex

What shou To protect against CVE-2023-29537, users should update their Mozilla Firefox, Firefox for Android, and

What is CV CVE-2023-29536 is a security vulnerability in various Mozilla products including Firefox, Firefox for An

Which Mo The products affected by CVE-2023-29536 are Firefox (versions before 112), Focus for Android (versio

What is th The base score assigned to CVE-2023-29536 is 8.8, which is categorized as HIGH severity.

When was CVE-2023-29536 was published on 02 June 2023.

Where can More information about CVE-2023-29536 can be found in the security advisories published by Mozilla

What are p A potential attack scenario for CVE-2023-29536 could involve an attacker crafting malicious content s

How can C' To mitigate CVE-2023-29536, users and administrators should update their Mozilla products to the lat

What is th The CVE ID of the vulnerability is CVE-2023-29535.

What type CVE-2023-29535 refers to a memory corruption vulnerability that can lead to a potentially exploitable

Which pro The affected products by CVE-2023-29535 include Firefox versions prior to 112, Focus for Android pri

What is th CVE-2023-29535 has been assigned a base score of 6.5, which categorizes it as a MEDIUM severity vul

On what d CVE-2023-29535 was published on 02 June 2023.

Where can More information about CVE-2023-29535 can be found through the following references:- <https://ww>

What are t The exploitation of CVE-2023-29535 could lead to memory corruption and an exploitable crash, allowi

Could you The technical cause of CVE-2023-29535 is that after a Garbage Collector compaction in the affected M

What mea To mitigate the risk posed by CVE-2023-29535, users should update the vulnerable Mozilla products to

Can you de A possible attack scenario exploiting CVE-2023-29535 might involve an attacker crafting a malicious w

What is CV CVE-2023-29533 is a security vulnerability identified in certain versions of Mozilla software including Firefox for Android. Which Mo: The Mozilla software versions affected by CVE-2023-29533 include Firefox versions prior to 112, Focus, and Firefox for iOS. What is the CVSS base score of CVE-2023-29533 is 4.3, which is classified as MEDIUM severity.

When was CVE-2023-29533 was published on 02 June 2023.

Where can More information about CVE-2023-29533 can be found on the official Mozilla security advisory pages at <https://www.mozilla.org/en-US/security/advisories/mozcve-2023-29533/>.

What could An example code that could lead to the CVE-2023-29533 vulnerability would be a malicious website using the following code snippet:

Can you describe Possible attack scenarios for CVE-2023-29533 could involve a phishing website that invites the user to download a malicious app.

What is the CVE ID for the vulnerability involving fullscreen notifications being hidden is CVE-2023-28159.

Can you describe CVE-2023-28159 describes a vulnerability in Firefox for Android where the fullscreen notification could be hidden.

What is the CVSS base score for CVE-2023-28159 is 4.3, and it is considered a MEDIUM severity vulnerability.

When was CVE-2023-28159 was published on 02 June 2023.

Are there any further information on CVE-2023-28159 can be found at the following references: '<https://www.mozilla.org/en-US/security/advisories/mozcve-2023-28159/>'.

Which versions Firefox for Android versions prior to 111 are affected by CVE-2023-28159.

Is CVE-2023-28159 specific to Firefox for Android. Other operating systems are not affected by this vulnerability.

What kind of CVE-2023-28159 could potentially lead to user confusion or spoofing attacks due to the fullscreen notification being hidden.

How can CVE-2023-28159, users should update Firefox for Android to version 111 or later, where the fullscreen notification is visible.

Describe a An attacker could exploit CVE-2023-28159 by creating malicious web content that triggers download of a malicious app.

What is CV CVE-2023-25749 is a security vulnerability identified in Firefox for Android, where Android application could be hijacked.

Which versions CVE-2023-25749 affects Firefox for Android versions that are older than Firefox 111. Other versions of Firefox for Android are not affected.

How severe The CVE-2023-25749 vulnerability has been given a base score of 4.3 and is classified as MEDIUM severity.

When was The CVE-2023-25749 vulnerability was published on June 2, 2023.

What are the potential attack scenarios associated with CVE-2023-25749 occur when an attacker convinces a user to download a malicious app.

Where can More information about CVE-2023-25749 can be found on the Mozilla Foundation Security Advisory page at <https://www.mozilla.org/en-US/security/advisories/mozcve-2023-25749/>.

What is CV CVE-2023-25748 is a security vulnerability that was identified in Firefox for Android. It involves a scenario where a user is prompted to download a malicious app.

Which versions CVE-2023-25748 affects versions of Firefox for Android that are older than version 111. This vulnerability is not present in Firefox for iOS.

What is the impact of CVE-2023-25748 can result in user confusion or spoofing attacks, as it allows a prompt to download a malicious app.

How serious CVE-2023-25748 has been given a Base Score of 4.3, which classifies it as a MEDIUM severity vulnerability.

When was The CVE-2023-25748 vulnerability was published on 02 June 2023, with the necessary details provided in the Mozilla Security Advisory page.

Are there any references available for CVE-2023-25748. The Mozilla Security Advisory page at '<https://www.mozilla.org/en-US/security/advisories/mozcve-2023-25748/>'.

Can you provide For CVE-2023-25748, a code example is not possible as it's related to the behavior of the interface in Firefox for Android.

What is CV CVE-2023-23600 is a security vulnerability identified in Mozilla Firefox for Android, where per origin network access is not properly enforced.

Which platform CVE-2023-23600 specifically affects the Firefox browser on Android. Other operating systems, such as iOS, are not affected.

What is the severity rating of CVE-2023-23600 is classified as '6.5 MEDIUM' based on its Base Score.

On which date CVE-2023-23600 was published on 2nd June 2023.

How can I find More details about CVE-2023-23600 can be found by visiting the Mozilla security advisory webpage at <https://www.mozilla.org/en-US/security/advisories/mozcve-2023-23600/>.

Which versions CVE-2023-23600 affects versions of Firefox for Android prior to version 109. Users running Firefox for Android version 109 or later are not affected.

What could A potential attack scenario for CVE-2023-23600 could involve an attacker leveraging the permission to access the network.

How can users Users can protect themselves from CVE-2023-23600 by updating their Firefox for Android browser to version 109 or later.

What is the CVE ID for the critical security vulnerability discovered in The Thaiger app for Android is CVE-2023-29746.

Can you describe CVE-2023-29746 refers to a vulnerability found in The Thaiger v.1.2 for Android, which allows unauthorized access to the app's data.

What is the CVE-2023-29746 has been assigned a CVSS base score of 9.8 which is categorized as CRITICAL. This score is based on the severity of the impact.

When was CVE-2023-29746 was published on 02 June 2023.

Where can More information and references about CVE-2023-29746 can be found on several websites including the Mozilla Security Advisory page.

What possible Possible attack scenarios for CVE-2023-29746 include unauthorized applications on the same device or network.

How might In practice, a code execution attack exploiting CVE-2023-29746 might involve the attacker creating a script to execute arbitrary code.

What is CV CVE-2023-29725 refers to a vulnerability found in the BT21 x BTS Wallpaper app version 12 for Android, which allows unauthorized access to the app's data.

What is the CVE-2023-29725 has been assigned a base score of 5.5, which is categorized as MEDIUM severity.

What type CVE-2023-29725 can lead to a persistent denial of service attack by allowing an attacker to inject mali  
On what d CVE-2023-29725 was published on 02 June 2023.

What are t An attacker could exploit CVE-2023-29725 by creating a malicious application that, once installed on t  
Are there a Yes, there are several references available for more information on CVE-2023-29725, including a blog  
What is the CVE ID of the vulnerability is CVE-2023-29724.

What type CVE-2023-29724 is an escalation of privilege vulnerability in the BT21 x BTS Wallpaper app for Android  
Can you de CVE-2023-29724 allows unauthorized apps to request permission to modify data in the app's database  
What is the CVE-2023-29724 has been given a Base Score of 7.8, which is classified as HIGH.

When was CVE-2023-29724 was published on 02 June 2023.

Where can More information about CVE-2023-29724 can be found on the following websites:- <http://bungaakpst>

What could By exploiting CVE-2023-29724, an attacker could potentially alter the data representing user preferen

What are s Possible attack scenarios for CVE-2023-29724 include an attacker developing a malicious app that req

What shou Users of the BT21 x BTS Wallpaper app should keep their app updated and follow any official advice o

How can d Developers can prevent vulnerabilities similar to CVE-2023-29724 by implementing secure coding pra

What is CV CVE-2023-29736 refers to a security vulnerability found in the Keyboard Themes app version 1.275.1.

How sever The threat posed by CVE-2023-29736 is considered to be CRITICAL, with a base score of 9.8. This indic  
On what d CVE-2023-29736 was published on 01 June 2023.

What versi The version of Keyboard Themes app affected by CVE-2023-29736 is 1.275.1.164 for Android.

Where can More technical details about CVE-2023-29736 can be found at the following URL: <https://github.com/>

What type CVE-2023-29736 is categorized as a dictionary traversal vulnerability.

Can you pr An attack scenario for CVE-2023-29736 may involve a malicious application that is installed on the sar

What is the The arbitrary code execution impact of CVE-2023-29736 allows an attacker to run malicious code of th

What is CV CVE-2023-29723 is a vulnerability in the Glitter Unicorn Wallpaper app for Android 7.0 through 8.0. It

How sever CVE-2023-29723 has been assessed with a Base Score of 7.5, which is categorized as HIGH severity.

When was The vulnerability identified as CVE-2023-29723 was published on 01 June 2023.

Which vers CVE-2023-29723 affects the Glitter Unicorn Wallpaper app for Android versions 7.0 through 8.0.

Can you pr An attack leveraging CVE-2023-29723 could involve a malicious application sending a crafted intent co

Where can More information about CVE-2023-29723 can be found on the GitHub repository at the following link:

What type CVE-2023-29723 addresses a security issue that leads to a persistent denial of service attack by allowi

What is CV CVE-2023-29722 is a vulnerability identified in the Glitter Unicorn Wallpaper app for Android versions

How sever CVE-2023-29722 has been assessed with a base score of 9.1 and classified as CRITICAL. This high sever

What versi CVE-2023-29722 affects the Glitter Unicorn Wallpaper app specifically on Android versions 7.0 throug

When was CVE-2023-29722 was published on 01 June 2023.

Where can More information about CVE-2023-29722 can be found at the provided reference link: <https://github>.

Can you de An attacker could exploit CVE-2023-29722 by creating a malicious app that requests permission to acc

Are there a Since the exact technical details and source code of the Glitter Unicorn Wallpaper app are not include

What is the CVE ID for the vulnerability found in Story Saver for Instagram - Video Downloader for Android is

Can you de CVE-2023-29748 describes a vulnerability in the Story Saver for Instagram - Video Downloader 1.0.6 fo

What is the base severity score of CVE-2023-29748 is 7.5, and it is classified as HIGH.

When was CVE-2023-29748 was published on 01 June 2023.

Where can More information about CVE-2023-29748 can be found through several sources: the app's Google Play

Could you An example attack scenario for CVE-2023-29748 would involve an attacker using the exposed method

What kind CVE-2023-29748 is a vulnerability caused by an exposed component that allows unauthorized modific

How might To prevent an attack like CVE-2023-29748, developers should ensure that components such as metho

What is the CVE ID of the vulnerability found in Story Saver for Instagram - Video Downloader for Android is C

Which vers Story Saver for Instagram - Video Downloader version 1.0.6 for Android is affected by CVE-2023-2974

What kind CVE-2023-29747 involves an exposed component vulnerability where the component allows methods

How critical CVE-2023-29747 is rated as 9.8, and it is classified as CRITICAL in terms of severity.

When was CVE-2023-29747 published on 31 May 2023.

What are some possible attack scenarios due to CVE-2023-29747 include attackers modifying data in any SharedPreferences

Where can more information about CVE-2023-29747 can be found through the following references:- <https://www>

Can you provide As CVE-2023-29747 refers to an issue with exposed methods to modify SharedPreferences files, a simple

What is the CVE ID of the vulnerability found in BestWeather v.7.3.1 for Android is CVE-2023-29745.

Can you explain CVE-2023-29745 refers to a security issue within the BestWeather version 7.3.1 application for Android

What is the CVE ID The base score assigned to CVE-2023-29745 is 7.1, which is categorized as HIGH severity.

When was CVE-2023-29745 published on 31 May 2023.

Where can more information about CVE-2023-29745 can be found through the following references:- <http://www>

What could An attacker could exploit CVE-2023-29745 by creating a malicious app that, once installed, manipulate

Is there an While providing exact exploit code for CVE-2023-29745 would not be responsible, I can describe hypo

What is the CVE ID for the vulnerability found in BestWeather app for Android is CVE-2023-29742.

Which version The affected version of BestWeather for Android by CVE-2023-29742 is v.7.3.1.

What type CVE-2023-29742 is associated with a code execution attack that can be initiated by unauthorized apps

How severe The CVE-2023-29742 vulnerability is considered to have a high severity with a base score of 7.8.

When was The CVE-2023-29742 vulnerability published on 31 May 2023.

Can you provide Yes, more information about CVE-2023-29742 can be found at the following references: - <http://www>.

What possible Possible attack scenarios for exploiting CVE-2023-29742 include unauthorized apps on the same device

Are there any As an AI, I can't directly provide code examples for exploitation. However, typically, exploiting a vulner

What can Users of BestWeather app can protect themselves from CVE-2023-29742 by ensuring they update the

Has the vendor The information provided does not specify if the vendor has already addressed CVE-2023-29742. How

What is the CVE ID The CVE ID associated with the vulnerability found in BestWeather v.7.3.1 for Android is CVE-2023-29

Can you describe CVE-2023-29743 refers to a vulnerability in BestWeather v.7.3.1 for Android that allows unauthorized

What is the CVE ID The base score assigned to CVE-2023-29743 is 7.5, which is categorized as HIGH severity.

When was CVE-2023-29743 publicly disclosed on 30 May 2023.

Where can more details about CVE-2023-29743 can be found at the following references:- <https://github.com/Lia>

Can you provide In an attack scenario for CVE-2023-29743, an unauthorized app installed on the same device could exploit

What type Due to the vulnerability identified by CVE-2023-29743, a persistent denial of service (DoS) attack is possible

What is CVE-2023-29741 references a security vulnerability found in BestWeather version 7.3.1, an Android app

How severe The vulnerability identified as CVE-2023-29741 has been given a Base Score of 9.8, which classifies it as

When was The CVE-2023-29741 vulnerability published on 30 May 2023.

What are the Possible attack scenarios for CVE-2023-29741 include unauthorized apps gaining higher privileges than

Where can more information about CVE-2023-29741 can be found through the following references: the ZMTQSI

Which app CVE-2023-29741 affects the BestWeather application for Android, specifically version 7.3.1.

What is the CVE ID The CVE ID of the vulnerability found in the Alarm Clock for Heavy Sleepers app for Android is CVE-20

What type CVE-2023-29740 describes an issue that allows unauthorized apps to cause a denial of service attack

What is the CVE ID The severity score of CVE-2023-29740 is 7.5, which is considered HIGH.

When was CVE-2023-29740 published on 30 May 2023.

Can you provide Yes, detailed information about CVE-2023-29740 can be found at the following URL: <https://github.com>

Where can The Alarm Clock for Heavy Sleepers app affected by CVE-2023-29740 can be found on the Google Play

What could A potential attack scenario for CVE-2023-29740 could involve a malicious application that is installed

What is the CVE ID The CVE ID for the critical vulnerability found in the 'Alarm Clock for Heavy Sleepers' Android app is C

Can you describe CVE-2023-29739 refers to an issue in Alarm Clock for Heavy Sleepers version 5.3.2 for Android where

What is the CVE ID The severity score assigned to CVE-2023-29739 is 9.8, which is categorized as CRITICAL.

When was The vulnerability designated as CVE-2023-29739 was publicly disclosed on the 30th of May, 2023.

Are there any Yes, more information about CVE-2023-29739 can be found at the following URLs:- <https://github.com>

**What kind** The vulnerability CVE-2023-29739 could be exploited by an unauthorized application to perform escalation of privilege.

**What are the impacts** An escalation of privilege attack exploiting CVE-2023-29739 could lead to various negative outcomes, such as unauthorized access to sensitive data, system compromise, and data loss.

**What steps should users take** Users of the 'Alarm Clock for Heavy Sleepers' app should immediately check for updates to the application from the Google Play Store.

**What is the CVE ID** The CVE ID of the vulnerability is CVE-2023-29738.

**Which versions are affected** The Wave Animated Keyboard Emoji app for Android version 1.70.7 is affected by CVE-2023-29738.

**What type of vulnerability is this** CVE-2023-29738 is a local code execution and privilege escalation vulnerability.

**What is the base score** The base score assigned to CVE-2023-29738 is 7.8, which is categorized as HIGH.

**When was it published** CVE-2023-29738 was published on 30 May 2023.

**Where can I find more information** Information on CVE-2023-29738 can be found at the following URLs: - <https://play.google.com/store/apps/details?id=com.wave.keyboard.emoji>

**Can you provide a possible attack scenario** A possible attack scenario for CVE-2023-29738 would involve a local attacker with access to the device exploiting the vulnerability to gain elevated privileges.

**Is CVE-2023-29738 a local-only vulnerability** CVE-2023-29738 is described as requiring local access for exploitation, which means it is not a vulnerability that can be exploited remotely.

**What is CVE-2023-29728** CVE-2023-29728 refers to a security flaw in the Call Blocker application version 6.6.3 for Android, where an attacker could exploit the vulnerability to perform denial of service.

**How does it affect an Android device** CVE-2023-29728 affects an Android device by enabling attackers to tamper with certain feature-related settings, potentially leading to system instability or data loss.

**When was it published** CVE-2023-29728 was published on 30 May 2023.

**What is the severity score** The severity score given to CVE-2023-29728 is 9.8, which classifies it as CRITICAL according to the Common Vulnerability Scoring System (CVSS).

**How can users protect themselves** To protect themselves from CVE-2023-29728, users should immediately update the Call Blocker application to the latest version.

**Can you provide more details** Certainly, for more details on CVE-2023-29728, you can visit the following links: 1. A detailed CVE report from CVE Details: <https://cve.mitre.org/cve/2023/29728/>

**What might be a possible attack scenario** An attack scenario involving CVE-2023-29728 could be that an attacker develops a malicious application that exploits the vulnerability to perform denial of service.

**Are there any code examples** As CVE-2023-29728 is a recently disclosed vulnerability, direct code examples detailing the specific exploit are not available.

**What is CVE-2023-29727** CVE-2023-29727 is a security vulnerability identified in the Call Blocker application version 6.6.3 for Android, where an attacker could exploit the vulnerability to perform denial of service.

**How severe is it** The severity of CVE-2023-29727 is rated as 9.8 and classified as CRITICAL according to its base score.

**When was it published** CVE-2023-29727 was published on 30 May 2023.

**Where can I find more information** More information about CVE-2023-29727 can be found on multiple sources online such as the Google Play Store, CVE Details, and GitHub.

**What are the potential attack scenarios** Potential attack scenarios for CVE-2023-29727 include unauthorized applications on the same device exploiting the vulnerability to perform denial of service.

**What should users do** Users of the Call Blocker app should check for and apply updates provided by the application's developer to mitigate the risk of CVE-2023-29727.

**Can you give an example** While specific code examples are not available, an attacker could create a malicious application that is designed to exploit the vulnerability to perform denial of service.

**What is CVE-2023-29726** CVE-2023-29726 refers to a vulnerability in Call Blocker application version 6.6.3 for Android, where an attacker could exploit the vulnerability to perform denial of service.

**How severe is it** The vulnerability described by CVE-2023-29726 has been given a Base Score of 7.5, which is classified as MEDIUM.

**When was it published** CVE-2023-29726 was published on May 30, 2023.

**Can you explain the attack** An attacker could exploit CVE-2023-29726 by injecting a large volume of data into the Call Blocker application, leading to a denial of service.

**What are the impacts** The exploitation of CVE-2023-29726 could lead to a persistent denial of service condition for the Call Blocker application.

**Are there any references** Yes, you can find more information about CVE-2023-29726 at the following references:- The GitHub repository for Call Blocker: <https://github.com/CallBlocker/CallBlocker>

**What version is affected** The version of the Call Blocker application affected by CVE-2023-29726 is 6.6.3 for Android.

**What is the CVE ID** The CVE ID for the issue found in edjing Mix for Android is CVE-2023-29735.

**What kind of vulnerability is this** CVE-2023-29735 is a denial of service (DoS) vulnerability that affects the edjing Mix application version 7.09.01.

**How can it be exploited** A local attacker can exploit the vulnerability in CVE-2023-29735 by manipulating the database files of the edjing Mix application.

**What is the CVSS Base Score** The CVSS Base Score assigned to CVE-2023-29735 is 5.5 MEDIUM. This score indicates that the vulnerability has a moderate impact.

**When was it published** The CVE-2023-29735 vulnerability was published on 30 May 2023.

**Where can I find more information** More details about CVE-2023-29735 can be found in the reference provided, which is a link to a GitHub repository.

**Can you describe a possible attack scenario** A possible attack scenario involving CVE-2023-29735 could involve an attacker with physical or shell access to the device exploiting the vulnerability to perform denial of service.

**What is CVE-2023-29734** CVE-2023-29734 refers to a security vulnerability identified in the edjing Mix application version 7.09.01 for Android, where an attacker could exploit the vulnerability to perform denial of service.

**How severe is it** The vulnerability represented by CVE-2023-29734 is considered CRITICAL with a Base Score of 9.8. This indicates a high level of severity.

**When was it published** CVE-2023-29734 was published on the 30th of May, 2023.

**Which application is affected** The CVE-2023-29734 affects the edjing Mix application, specifically version 7.09.01, on the Android platform.

**Where can I find more information** More information about CVE-2023-29734 can be found at the provided GitHub link: <https://github.com/edjingmix/edjingmix>

**What type of attack is this** An attack exploiting CVE-2023-29734 would typically involve an unauthorized application on the same device exploiting the vulnerability to perform denial of service.

**What kind of mitigation is recommended** To mitigate CVE-2023-29734, users should update the edjing Mix application to a version where the vulnerability has been patched.

What is the CVE ID for the vulnerability found in the Lock Master app for Android is CVE-2023-29733.

What version The affected version of the Lock Master app for Android by CVE-2023-29733 is version 2.2.4.

What is the base score assigned to CVE-2023-29733 is 7.8, which is categorized as HIGH severity.

What type CVE-2023-29733 in the Lock Master app for Android allows unauthorized apps to modify SharedPreferences.

When was CVE-2023-29733 was published on 30 May 2023.

Where can More information about CVE-2023-29733 can be found on GitHub at the following URL: <https://github.com>

Could you A potential attack scenario for CVE-2023-29733 could involve a malicious app installed on the same device.

What is the CVE ID for the vulnerability found in SoLive for Android is identified by the CVE ID CVE-2023-29732.

What version The versions of SoLive for Android affected by CVE-2023-29732 range from 1.6.14 through to 1.6.20.

Can you provide The vulnerability in CVE-2023-29732 resides in an exposed component of SoLive versions 1.6.14 to 1.6.20.

What is the CVSS Base Score of CVE-2023-29732 has a CVSS Base Score of 9.8, which classifies it as a CRITICAL severity vulnerability.

When was CVE-2023-29732 was published on 30 May 2023.

Where can Further details or a detailed write-up about CVE-2023-29732 can be found on GitHub at the following URL: <https://github.com>

What are the Exploitation of CVE-2023-29732 could enable an attacker to manipulate the SharedPreferences data of the app.

What is the CVSS Base Score of CVE-2023-29731 refers to a security vulnerability discovered in SoLive versions 1.6.14 through 1.6.20.

How can it be exploited An attacker can exploit CVE-2023-29731 by injecting an excessive amount of data into the SharedPreferences.

What is the impact The impact of CVE-2023-29731 is significant as it allows an attacker to create a persistent denial of service.

What is the base score of CVE-2023-29731 is 7.5, which is categorized as HIGH severity according to the rating system.

On what date CVE-2023-29731 was published on 30 May 2023.

Where can More information about CVE-2023-29731 can be found at the following URL: <https://github.com/LianKe>

Can you provide A possible attack scenario for CVE-2023-29731 is that an attacker who has the ability to execute code on the device.

What is the CVE ID of the vulnerability found in Action Launcher for Android is CVE-2022-47028.

What type CVE-2022-47028 is a denial of service vulnerability that can be caused by arbitrary data injection into the app.

In which version CVE-2022-47028 was discovered in version 50.5 of Action Launcher for Android.

How severe The severity level of CVE-2022-47028 is rated as 'MEDIUM' with a base score of 5.5.

When was CVE-2022-47028 was published on 30 May 2023.

Where can More details about CVE-2022-47028 can be found on the GitHub page at <https://github.com/LianKe>

What could A possible attack scenario for CVE-2022-47028 involves an attacker injecting arbitrary data into the 'input' field.

Could you While I do not have access to the specific codebase of Action Launcher or the exact details of the vulnerability.

What is the CVE ID of the reported security issue is CVE-2023-29737.

In which version The vulnerability CVE-2023-29737 was discovered in version 1.70.7 of the Wave Animated Keyboard for Android.

What type CVE-2023-29737 describes a security issue that allows a local attacker to cause a denial of service via the keyboard.

What is the CVSS base score assigned to CVE-2023-29737 is 5.5 with a severity rating of MEDIUM. It indicates a moderate severity level.

On what date CVE-2023-29737 was published on 30 May 2023.

Where can More information about CVE-2023-29737 can be found at the following references:- <https://play.google.com/store/apps/details?id=com.wave.keyboard>

What are some Possible attack scenarios for CVE-2023-29737 include a local attacker exploiting the vulnerability to make the keyboard unresponsive.

Are there any As CVE-2023-29737 is a recently reported vulnerability, specific code examples demonstrating the exploit are not available.

What is the CVE ID for the vulnerability found in the Kiddoware Kids Place Parental Control app is CVE-2023-28153.

How severe CVE-2023-28153 has been assigned a Base Score of 6.4, which is categorized as a MEDIUM severity level.

When was CVE-2023-28153 was published on 29 May 2023.

Where can More information about CVE-2023-28153 can be found at the following URL: <https://sec-consult.com>

Can you provide An attack scenario for CVE-2023-28153 would involve a child who has restricted access to their Android phone.

What steps To remediate CVE-2023-28153, users should update the Kiddoware Kids Place Parental Control application to the latest version.

What is the CVE ID for the vulnerability found in the Omni-notes Android app is CVE-2023-33188.

How severe The CVE-2023-33188 vulnerability has been given a Base Score of 5.5, which classifies it as MEDIUM severity.

What application The Omni-notes Android app, an open-source note-taking application for Android devices, is affected by this vulnerability.

What steps To address CVE-2023-33188, users of the Omni-notes Android app should update to the latest version of the application.

When was The CVE-2023-33188 vulnerability was published on 27 May 2023.

Where can Further information about CVE-2023-33188 is available in the security advisory published on GitHub, 1

Can you pr In an attack scenario for CVE-2023-33188, a malicious or compromised application on the same device

What is the CVE ID for the vulnerability discovered in the c-ares library is CVE-2023-31124.

Can you de CVE-2023-31124 is related to the c-ares library, which is an asynchronous resolver library. The issue a

The base score assigned to CVE-2023-31124 is 3.7, which is classified as LOW severity.

When was CVE-2023-31124 was published on 25 May 2023.

What version The vulnerability mentioned in CVE-2023-31124 was patched in c-ares version 1.19.1.

Where can You can find more information and advisories related to CVE-2023-31124 at the following sources: the

Could you | In an attack scenario related to CVE-2023-31124, a malicious actor might exploit the vulnerability by p

How does CVE-2023-31124 impacts the security of the c-ares library by compromising the randomness quality re

Is there an While a specific code example for CVE-2023-31124 is not provided in the CVE description, one could ir

What is the CVE ID is CVE-2023-2863.

Can you de CVE-2023-2863 is a vulnerability that has been identified in the Simple Design Daily Journal app versic

How serious The CVE-2023-2863 vulnerability has been given a base score of 5.5, which is categorized as MEDIUM

What type The attack vector for CVE-2023-2863 is local, meaning that the attack would have to be launched on the

When was CVE-2023-2863 was published on 24 May 2023.

Where can More information about CVE-2023-2863 can be found on various websites such as <https://vuldb.com/>

Are there a To mitigate the risk associated with CVE-2023-2863, users should update the Simple Design Daily Jour

Can you pr An attack scenario for exploiting CVE-2023-2863 could involve an attacker gaining physical access to a

What is the CVE ID of the vulnerability found in the Brother iPrint&Scan application is CVE-2023-28369.

Can you de CVE-2023-28369 is a vulnerability in Brother iPrint&Scan V6.11.2 and earlier versions, where the appli

What is the The Common Vulnerability Scoring System (CVSS) base score for CVE-2023-28369 is 3.3, which is class

When was CVE-2023-28369 was published on 18 May 2023.

Where can More information or updates about CVE-2023-28369 can be found at the following links: - <https://support.microsoft.com/en-us/topic/faq-for-cve-2023-28369>

What versi Brother iPrint&Scan V6.11.2 and earlier versions are affected by the vulnerability CVE-2023-28369.

Could you | A potential attack scenario for CVE-2023-28369 would involve an attacker creating or using a malicious

Are there any? As of the information provided, there are no specific code examples available to the public that demo

What is CV CVE-2023-2722 refers to a security vulnerability in Google Chrome on Android where there is a 'use at

How severe: The CVE-2023-2722 vulnerability has been given a severity rating of 'High' with a CVSS base score of 8

On what date CVE-2023-2722 was published on 16 May 2023.

Which vers CVE-2023-2722 affects Google Chrome on Android versions prior to 113.0.5672.126.

Are there : Yes, there are several references available for CVE-2023-2722, including an entry in the Chromium bu

What are p Potential attack scenarios for CVE-2023-2722 involve a remote attacker crafting a malicious HTML page

Can you pr Unfortunately, it would be inappropriate and potentially harmful to provide a code example that expl

Has CVE-2023-2722 been addressed by Google in Chrome version 113.0.5672.126 for Android. Us

What is the CVE ID for the vulnerability involving a heap buffer overflow in Android is CVE-2023-21118.

Which file The vulnerability CVE-2023-21118 affects the unflattenString8 function within the Sensor.cpp file.

What could The CVE-2023-21118 vulnerability could lead to local information disclosure due to the possible out of

Do users need to interact with the system to exploit the vulnerability? No, user interaction is not needed for the exploitation of the CVE-2023-21118 vulnerability.

What are t The Android versions affected by CVE-2023-21118 are Android-11, Android-12, Android-12L, and And

What is the base score severity rating assigned to CVE-2023-21118 is 5.5, which is categorized as MEDIUM.

When was The CVE-2023-21118 vulnerability was published on 15 May 2023.

Where can More information about CVE-2023-21118 can be found at the Android Security Bulletin URL: <https://source.android.com/security/bulletin/2023-08-01>

What is the Android ID associated with CVE-2023-21118 is A-269014004.

Describe a An attacker could exploit CVE-2023-21118 by crafting a malicious application that does not require an

What is CV CVE-2023-21117 is a security vulnerability identified in the Android operating system. It resides in the



How severe CVE-2023-21117 is categorized with a base score of 7.8, which is considered HIGH in severity. This means that exploiting CVE-2023-21117 does not require user interaction. An attacker could take advantage of this vulnerability to perform actions on the device without the user's knowledge.

When was CVE-2023-21117 published? CVE-2023-21117 was published on 15 May 2023. It was disclosed in official Android security bulletins.

Where can I find more details about CVE-2023-21117? More details about CVE-2023-21117 can be found in the Android Security Bulletin at the following URL: [https://source.android.com/security/bulletin/2023-05/01](#).

Can you describe an attack scenario for CVE-2023-21117? An attack scenario for CVE-2023-21117 could involve a malicious application that leverages the permissions granted to it to perform actions on the device without the user's knowledge.

Are there any mitigations for CVE-2023-21117? Due to the nature of this security vulnerability, sharing exploit code examples is not a responsible practice. However, users can update their devices to the latest version of Android to mitigate the risk.

What is CVE-2023-21116? CVE-2023-21116 refers to a security vulnerability in the Android operating system where there is a local privilege escalation (LPE) issue.

How severe is CVE-2023-21116? The CVE-2023-21116 vulnerability has been given a base score of 6.7, which is categorized as MEDIUM severity.

Which versions of Android are affected by CVE-2023-21116? CVE-2023-21116 affects Android versions 11, 12, 12L, and 13. Devices running these versions are potentially vulnerable.

Is user interaction required to exploit CVE-2023-21116? No, user interaction is not required to exploit CVE-2023-21116. This means that the vulnerability could be exploited by a malicious actor who has gained system execution privileges.

Where can I find more information and a detailed report on CVE-2023-21116? More information and a detailed report on CVE-2023-21116 can be found on the official Android security website: [https://source.android.com/security/bulletin/2023-05/02](#).

Can you describe an attack scenario for CVE-2023-21116? An attack scenario for CVE-2023-21116 could involve a malicious actor who has gained system execution privileges to exploit the vulnerability.

What are the requirements to exploit CVE-2023-21116? An attacker needs to have system execution privileges to exploit CVE-2023-21116. This level of access is typically only available to root users.

What is CVE-2023-21112? CVE-2023-21112 refers to a security vulnerability in the `AnalyzeMfcResp` of `NxpMfcReader.cc`, where there is a local information disclosure (LID) issue.

Which versions of Android are affected by CVE-2023-21112? The Android versions affected by CVE-2023-21112 are Android-11, Android-12, Android-12L, and Android-13.

What are the potential impacts of CVE-2023-21112? CVE-2023-21112 could lead to local information disclosure on affected Android devices. This means that an attacker could potentially access sensitive information stored on the device.

Is user interaction required to exploit CVE-2023-21112? No, user interaction is not required to exploit CVE-2023-21112.

What is the CVSS base score for CVE-2023-21112? The CVSS base score for CVE-2023-21112 is 5.5, which is categorized as MEDIUM severity.

When was CVE-2023-21112 published? CVE-2023-21112 was published on 15 May 2023.

Where can I find more information about CVE-2023-21112? You can find more information about CVE-2023-21112 by visiting the Android Security Bulletin at the following URL: [https://source.android.com/security/bulletin/2023-05/03](#).

What could be a possible attack scenario for CVE-2023-21112? A possible attack scenario for CVE-2023-21112 involves a malicious app that requires no special permissions to exploit the vulnerability.

What mitigation steps can be taken for CVE-2023-21112? To protect against CVE-2023-21112, users should ensure their Android devices are up to date with the latest security patches.

What is CVE-2023-21111? CVE-2023-21111 refers to a security vulnerability identified in several functions of the `PhoneAccountManager` class.

Which versions of Android are affected by CVE-2023-21111? The affected versions of Android by CVE-2023-21111 are Android 11, Android 12, Android 12L, and Android 13.

What type of vulnerability is CVE-2023-21111? CVE-2023-21111 is a security vulnerability related to improper input validation in certain functions of the `PhoneAccountManager` class.

What is the CVSS base score assigned to CVE-2023-21111? The CVSS base score assigned to CVE-2023-21111 is 5.5, indicating a medium level of severity.

How might an attacker exploit CVE-2023-21111? An attacker could exploit CVE-2023-21111 by manipulating the input in a way that is not properly validated.

Is user interaction required to exploit CVE-2023-21111? No, user interaction is not required to exploit the vulnerability described in CVE-2023-21111.

What date was CVE-2023-21111 published? CVE-2023-21111 was published on the 15th of May, 2023.

Where can I find more details about CVE-2023-21111? More details about CVE-2023-21111 can be found in the Android Security Bulletin at the following URL: [https://source.android.com/security/bulletin/2023-05/04](#).

What mitigation steps can be taken for CVE-2023-21111? Mitigation for CVE-2023-21111 would generally involve applying patches or updates provided by Android.

Can you provide an attack scenario for CVE-2023-21111? Since CVE-2023-21111 involves improper input validation and is specific to the internal workings of Android, an attack scenario is not provided.

What is CVE-2023-21110? CVE-2023-21110 is a security vulnerability identified in certain functions of the file `SnoozeHelper.java`.

What is the CVSS base score assigned to CVE-2023-21110? The base score assigned to CVE-2023-21110 is 7.8, which is classified as HIGH.

On which date was CVE-2023-21110 published? CVE-2023-21110 was published on 15 May 2023.

Which versions of Android are affected by CVE-2023-21110? The Android versions affected by CVE-2023-21110 include Android-11, Android-12, Android-12L, and Android-13.

Are there any mitigations for CVE-2023-21110? As CVE-2023-21110 is related to resource exhaustion in `SnoozeHelper.java`, specific code examples are not provided.

Can you provide an attack scenario for CVE-2023-21110? An attack scenario for CVE-2023-21110 might involve a malicious app that is designed to exhaust system resources.

Is user interaction required to exploit CVE-2023-21110? No, user interaction is not needed for exploitation of the vulnerability described by CVE-2023-21110.

Where can I find additional information or updates regarding CVE-2023-21110? Additional information or updates regarding CVE-2023-21110 can be found at the provided reference URL.

What is CVE-2023-21109? CVE-2023-21109 refers to a vulnerability in multiple places of the `AccessibilityService` on Android devices.

How severe is CVE-2023-21109? CVE-2023-21109 is considered to have a HIGH severity level with a base score of 7.8. This indicates that exploiting this vulnerability could have significant impacts.

Does exploiting CVE-2023-21109 require user interaction? No, exploiting CVE-2023-21109 does not require user interaction, which means that an attacker can potentially exploit the vulnerability without the user's knowledge.

On which date was CVE-2023-21109 published? CVE-2023-21109 was published on May 15, 2023.

What Android versions are affected by CVE-2023-21109? The Android versions affected by CVE-2023-21109 are Android 11, 12, 12L, and 13.

Are there any mitigations for CVE-2023-21109? Yes, a reference to the vulnerability CVE-2023-21109 can be found in the Android Security Bulletin at the following URL: [https://source.android.com/security/bulletin/2023-05/05](#).

Can you provide an example of an attack scenario for CVE-2023-21109? A possible attack scenario for CVE-2023-21109 could be as follows: An attacker could exploit the logic in the `AccessibilityService` to perform actions on the device without the user's knowledge.

What is CV CVE-2023-21107 refers to a security vulnerability in Android's NotificationAccessDetails.java file, where the security flaw in CVE-2023-21107 is a missing permission check that could allow an attacker to escape the sandbox. CVE-2023-21107 has been assigned a Base Score of 7.8, which is categorized as HIGH.

When was CVE-2023-21107 published on 15 May 2023.

Which Android versions affected by CVE-2023-21107 include Android 11, Android 12, Android 12L, and Android 13. Does CVE-21107 require user interaction for exploitation. No, CVE-2023-21107 does not require user interaction for exploitation.

How might an attacker exploit the vulnerability in CVE-2023-21107 by executing a malicious application that triggers the vulnerability.

Where can additional information about CVE-2023-21107 be found in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-05#CVE-2023-21107>.

Is there a known attack scenario for CVE-2023-21107? While specific details of an attack scenario for CVE-2023-21107 are not provided, a likely scenario would involve an attacker with local access to the device exploiting the vulnerability to escalate privileges.

What is the Android ID associated with CVE-2023-21107 is A-259385017.

What is the CVE ID of the vulnerability involving possible memory corruption in adreno\_set\_param of adreno\_gpu.c? The CVE ID of the vulnerability involving possible memory corruption in adreno\_set\_param of adreno\_gpu.c is CVE-2023-21106.

Can you describe CVE-2023-21106 describes a vulnerability in the adreno\_set\_param function within the adreno\_gpu.c file, which allows for local escalation of privilege.

What is the CVE-2023-21106 has been assigned a base score of 7.8, which categorizes it as a HIGH severity vulnerability.

On what date was CVE-2023-21106 published on 15 May 2023.

What versions of Android are affected by CVE-2023-21106 affects the Android kernel, but the specific versions of Android impacted by this vulnerability are not specified.

Where can more information on CVE-2023-21106 be found at <https://source.android.com/security/bulletin/2023-05#CVE-2023-21106>.

What type of attack does CVE-2023-21106 allows for local escalation of privilege, where an attacker with access to the device can exploit the vulnerability to gain higher privileges.

What kind of user interaction is required to exploit CVE-2023-21106, making it more severe as the vulnerability can be exploited without user interaction.

Could you provide a potential attack scenario for CVE-2023-21106 would involve an attacker with local access to the device exploiting the vulnerability to gain higher privileges.

Are there any specific details of the code that would exploit CVE-2023-21106, such as a proof-of-concept, are not provided.

What is CVE-2023-21104 refers to a vulnerability in the applySyncTransaction function of WindowOrganizer.java, which allows for local information disclosure.

What is the impact of CVE-2023-21104 is a local information disclosure vulnerability which means that a malicious actor can access sensitive information stored on the device.

Which Android versions are affected by CVE-2023-21104 affects Android versions Android-12L and Android-13.

What is the CVSS (Common Vulnerability Scoring System) base score for CVE-2023-21104 is 5.5, classified as MEDIUM severity.

How can a malicious actor exploit CVE-2023-21104 by creating an application that utilizes the applySyncTransaction function to access sensitive information.

Was there an advisory released for CVE-2023-21104. Yes, an advisory was released for CVE-2023-21104. Information about the vulnerability and related patches can be found in the Android Security Bulletin.

When was CVE-2023-21104 publicly disclosed on May 15, 2023.

What immediate actions should users take? Users should immediately check for and apply updates provided by Android to patch CVE-2023-21104.

Can you provide specific code patches are typically provided by the official developers (in this case, Android's security team) to address the vulnerability.

What is the CVE ID for that vulnerability is CVE-2023-21103.

Which Android versions are affected by CVE-2023-21103 affects Android-11, Android-12, Android-12L, and Android-13.

How severe is the vulnerability described by CVE-2023-21103 has been rated with a base score of 5.5, classifying it as MEDIUM severity.

What type of attack does CVE-2023-21103 concerns a local persistent denial of service that can occur due to uncaught exceptions in the kernel.

Do users need to interact with the device to exploit CVE-2023-21103. No, user interaction is not needed for the exploitation of CVE-2023-21103.

When was CVE-2023-21103 published on 15 May 2023.

What is the Android ID associated with CVE-2023-21103 is A-259064622.

Where can more information about CVE-2023-21103 be found at the Android Security Bulletin link provided: <https://source.android.com/security/bulletin/2023-05#CVE-2023-21103>.

Could you provide an attack scenario for CVE-2023-21103 by causing a malformed or specially crafted piece of user data to trigger the vulnerability.

What are the execution privileges needed to exploit CVE-2023-21103; it can be triggered under normal user permissions.

What is the CVE ID associated with the vulnerability is CVE-2023-21102.

What is the CVE-2023-21102 has been assigned a base score of 7.8, which is categorized as HIGH severity.

What is the CVE-2023-21102 describes a possible bypass of shadow stack protection due to a logic error in the code.

Does CVE-21102 require user interaction for exploitation. No, CVE-2023-21102 does not require user interaction for exploitation.

As per the vulnerable versions affected by CVE-2023-21102 are not specified other than being referred to as Android versions 11 through 13.

Where can more information about CVE-2023-21102 be found on the Android security bulletin for May 2023.

When was CVE-2023-21102 published on 15 May 2023.

Can you provide an attack scenario for CVE-2023-21102 by running a malicious application that triggers the vulnerability.

What is the Android ID associated with CVE-2023-21102 is A-260821414.

Is there an example of code that demonstrates the issue in CVE-2023-21102. Typically, the details of the issue are provided in the Android Security Bulletin.

What is the CVE ID of the vulnerability related to Android's ShortcutPackage.java is CVE-2023-20930.

Can you describe the vulnerability in the pushDynamicShortcut method of Android's ShortcutPackage.java?

What are the affected Android versions by CVE-2023-20930 are Android 11, Android 12, Android 12L, and Android 13.

What is the base score assigned to CVE-2023-20930 is 5.5, which is considered MEDIUM severity.

When was CVE-2023-20930 was publicly disclosed on 15 May 2023.

Where can more information about CVE-2023-20930 can be found at the Android Security Bulletin, with the following link: <https://source.android.com/security/bulletin/2023-05/01>.

What is the Android ID associated with CVE-2023-20930 is A-250576066.

What are the details of the vulnerability in the pushDynamicShortcut method within the ShortcutPackage.java?

Are there any code examples or PoCs available for CVE-2023-20930? Given the nature of CVEs and responsible disclosure practices, specific exploit code examples detailing the vulnerability are not typically provided.

What is CVE-2023-20914 refers to a security vulnerability found in onSetRuntimePermissionGrantStateByDeviceName method.

What versions of Android are affected by CVE-2023-20914 affects the Android-11 version.

What is the base score assigned to CVE-2023-20914 is 5.5, which is categorized as MEDIUM severity.

What kind of issue does CVE-2023-20914 represents a permissions bypass issue that results in a potential information disclosure.

On what date was CVE-2023-20914 was published on 15 May 2023.

Where can more information about CVE-2023-20914 can be found in the official Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-05/01>.

What are the potential attack scenarios for exploiting CVE-2023-20914 could include an attacker with user-level permissions.

What is the severity level of CVE-2021-0877 is 9.8, which is classified as CRITICAL.

What products are affected by CVE-2021-0877 is Android, specifically Android System on Chip (SoC) systems.

When was CVE-2021-0877 was published on 15 May 2023.

Where can more details about CVE-2021-0877 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2021-05/01>.

Could you provide an example of how the vulnerability in CVE-2021-0877 could be exploited? A possible attack scenario for exploiting CVE-2021-0877 could involve an attacker crafting a malicious application that triggers the vulnerability.

What versions of Android are affected by CVE-2021-0877 impacts certain versions of Android associated with the System on Chip (SoC) platform.

What is the CVE ID of the reported vulnerability in the Intel Retail Edge Mobile Android application is CVE-2023-25772.

Can you describe the vulnerability in the Intel Retail Edge Mobile Android application where the application fails to properly validate user input.

What is the severity level of CVE-2023-25772 has been assigned a severity level of MEDIUM with a base score of 5.5.

As of which date has the vulnerability reported by CVE-2023-25772 has been addressed in the Intel Retail Edge Mobile Android application.

When was the vulnerability with the identifier CVE-2023-25772 was published on 10 May 2023.

Where can more information about the vulnerability CVE-2023-25772 can be found at the following URL: <https://www.intel.com/content/www/us/en/security/advisories/000000000.html>.

What might be a possible attack scenario for CVE-2023-25772 is where an attacker, who is already authenticated and authorized, could exploit the vulnerability to gain unauthorized access to sensitive data.

What is the CVE ID for the vulnerability is CVE-2023-25179.

Can you describe the vulnerability in the Intel Unite android application where the application fails to properly validate user input.

What is the CVSS base score for CVE-2023-25179 is 5.5, which is classified as MEDIUM severity. This score indicates the severity of the vulnerability.

When was CVE-2023-25179 was published on 10 May 2023.

Where can detailed information regarding CVE-2023-25179 can be found in the advisory provided by Intel, which is available at the following link: <https://www.intel.com/content/www/us/en/security/advisories/000000000.html>.

How might an attacker exploit CVE-2023-25179 by creating conditions that lead to uncontrolled resource consumption.

What are the prerequisites for an attack to be successful against CVE-2023-25179, the attacker would need to have local access to the device.

Has Intel addressed the vulnerability with the release of Intel Unite android application Release 17.0?

What is CVE-2023-23573 is a security vulnerability identified in the Intel® Unite® Android application. It involves a buffer overflow in the application's handling of user input.

How severe is the security issue associated with CVE-2023-23573 is rated as 'MEDIUM' with a base score of 5.5.

When was CVE-2023-23573 was published on 10 May 2023.

Which versions of the Intel® Unite® Android application are affected by CVE-2023-23573 affects versions of the Intel® Unite® Android application that are prior to Release 17.

Where can you find more information or an advisory about CVE-2023-23573 on Intel's official website, specifically at the following link: <https://www.intel.com/content/www/us/en/security/advisories/000000000.html>.

What type of vulnerability is CVE-2023-23573 is an improper access control vulnerability. A potential attack scenario could involve an attacker exploiting the vulnerability to gain unauthorized access to sensitive data.

Can you provide an actual code example for the CVE-2023-23573 vulnerability would not be appropriate as it could reveal sensitive information.

What is the CVE ID for the vulnerability is CVE-2022-46645.

Can you pr CVE-2022-46645 is described as an issue of uncontrolled resource consumption in the Intel(R) Smart C  
How sever The CVE-2022-46645 vulnerability has been assigned a base score of 5.5, which is categorized as MED  
When was The CVE-2022-46645 vulnerability was published on 10 May 2023.

Where can More information about CVE-2022-46645 can be found on the Intel Security Center advisory page at h  
What versi CVE-2022-46645 affects versions of the Intel Smart Campus Android application prior to version 9.9.

What type To exploit the vulnerability CVE-2022-46645, an attacker would need local access and must be authen

Can you di: Potential attack scenarios for CVE-2022-46645 could involve an authenticated attacker exploiting the

Are there ε A code example for exploiting CVE-2022-46645 is not possible to provide as it would depend on specifi

What does CVE-2022-46279 refers to a vulnerability involving improper access control in the Intel(R) Retail Edge i

What is the The CVSS base score given to CVE-2022-46279 is 5.5, which is rated as MEDIUM severity.

When was CVE-2022-46279 was published on 10 May 2023.

What could A possible attack scenario for CVE-2022-46279 might involve an attacker, who already has authentical

How can th To mitigate the vulnerability identified as CVE-2022-46279, users of the Intel(R) Retail Edge android ap

Where can More information about CVE-2022-46279 can be found in the security advisory released by Intel, whic

What is CV CVE-2022-41801 refers to a security vulnerability discovered in the Intel(R) Connect M Android applica

What is the The impact of CVE-2022-41801 is medium level with a base score of 5.5. It may allow an authenticated

How can C CVE-2022-41801 can be mitigated by updating the Intel(R) Connect M Android application to version i

When was CVE-2022-41801 was published on 10 May 2023.

Where can More information about CVE-2022-41801 can be found at the official Intel security advisory page: http

What would A possible attack scenario for CVE-2022-41801 could involve an attacker who has authenticated acces

Are there ε Due to the nature of CVE-2022-41801, code examples specific to exploiting this vulnerability are not to

What is CV CVE-2022-41769 refers to a security vulnerability that was found in the Intel(R) Connect M Android ap

How sever The severity of CVE-2022-41769 is rated as HIGH with a Base Score of 7.8.

When was CVE-2022-41769 was published on 10 May 2023.

Which vers Versions of the Intel(R) Connect M Android application before version 1.82 are affected by CVE-2022-

Where can More information about CVE-2022-41769 can be found at the Intel Security Center advisory page: http

What type CVE-2022-41769 is categorized as an 'improper access control' vulnerability that could lead to escalati

What remε The recommended remediation for CVE-2022-41769 is to update the Intel(R) Connect M Android appl

Could you A potential attack scenario for CVE-2022-41769 could involve an attacker who has legitimate access to

What is the The CVE ID for the stored Cross-Site Scripting vulnerability in the WPMobile.App plugin is CVE-2023-2

Can you de CVE-2023-28932 is a stored Cross-Site Scripting (XSS) vulnerability that affects the WPMobile.App — /

What versi CVE-2023-28932 affects versions of the WPMobile.App plugin for WordPress up to and including 11.2

What is the The CVSS base score assigned to CVE-2023-28932 is 4.8, which represents a medium severity level for

When was CVE-2023-28932 was publicly disclosed on 10 May 2023.

Where can More information about CVE-2023-28932 can be found at the following URL: <https://patchstack.com/>

What is a p An attack scenario for CVE-2023-28932 could involve an admin-level user inadvertently adding malici

Can you pr Certainly! While I won't provide a malicious script, I can explain the nature of the exploit. An attacker

What is the The CVE ID of the vulnerability is CVE-2023-32060.

What is the CVE-2023-32060 has been assigned a base score of 6.5, which indicates a medium severity level.

As of when CVE-2023-32060 was published on 09 May 2023.

What versi Affected versions of DHIS2 Core are those starting in the 2.35 branch and prior to versions 2.36.13, 2.3

What issue CVE-2023-32060 addresses an issue in DHIS2 Core where the `/trackedEntityInstances` and `/events` /

Can you pr Yes, more information about CVE-2023-32060 can be found at the following reference link: <https://git>

How is the In the case of CVE-2023-32060, the events will not appear in the user interface for the web-based Tra

What is the The impact of the CVE-2023-32060 vulnerability is that users might gain access to events which they s

What are t The fixed versions for CVE-2023-32060 are 2.36.13, 2.37.8, 2.38.2, and 2.39.0 of the DHIS2 Core.

Is there a k According to the information given, no workaround is known for the CVE-2023-32060 vulnerability. Th

On what date CVE-2022-48186 was published on 01 May 2023.

Which app The Baiying Android application is affected by CVE-2022-48186.

What could Exploiting CVE-2022-48186 could potentially lead to information disclosure, which means sensitive data

Are there any Yes, more information on CVE-2022-48186 can be found at the following URL: <https://iknow.lenovo.com/en-us/knowledge-center/cve-2022-48186>

What kind Attack scenarios for CVE-2022-48186 could involve an attacker exploiting the certificate validation vulnerability

What is CVSS CVE-2023-21100 is a security vulnerability found in the inflate function of inflate.c within the Android

How severe The vulnerability described by CVE-2023-21100 is rated with a Base Score of 7.8, which is considered High

Does CVE-21097 require user interaction? No, the exploitation of CVE-2023-21100 does not require user interaction.

On what date was CVE-2023-21100 published? CVE-2023-21100 was published on 19 April 2023.

Which Android versions are affected by CVE-2023-21100? The Android versions affected by CVE-2023-21100 are Android 12, Android 12L, and Android 13.

Where can more details about CVE-2023-21100 be found? More details about CVE-2023-21100 can be found on the Android Security Bulletin web page at <https://source.android.com/security/bulletin/2023-04/01>.

What kind of vulnerability is CVE-2023-21100? CVE-2023-21100 could potentially lead to a local escalation of privilege on the affected Android device.

What are possible attack scenarios for CVE-2023-21100? Possible attack scenarios for CVE-2023-21100 include an attacker being able to exploit the heap buffer overflow.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21099.

Can you describe CVE-2023-21099? CVE-2023-21099 is a vulnerability in the PackageManagerSession.java of Android, where due to a logic error, a user can escalate their privileges.

What versions of Android are affected by CVE-2023-21099? The Android versions affected by CVE-2023-21099 include Android 11, Android 12, Android 12L, and Android 13.

How severe is CVE-2023-21099? The severity of CVE-2023-21099 is rated as 7.8, which is categorized as HIGH according to its Base Score.

When was CVE-2023-21099 published? CVE-2023-21099 was published on 19 April 2023.

Where can more information about CVE-2023-21099 be found? More information about CVE-2023-21099 can be found on the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-04/01>.

What kind of vulnerability is CVE-2023-21099? CVE-2023-21099 could lead to local escalation of privilege, which means an attacker could gain elevated privileges.

Is user interaction required for CVE-2023-21099? No, user interaction is not required for an attacker to exploit CVE-2023-21099. This makes the vulnerability more severe.

Could you provide an example of an attack scenario for CVE-2023-21099? An example of an attack scenario for CVE-2023-21099 could involve a malicious app that exploits the vulnerability to gain root access.

What is the CVE ID for the vulnerability affecting AccountManagerService.java? The CVE ID for the vulnerability affecting AccountManagerService.java in Android is CVE-2023-21098.

Can you describe CVE-2023-21098? CVE-2023-21098 describes a vulnerability in AccountManagerService.java, where there is a potential for local escalation of privilege.

What are the affected versions of Android for CVE-2023-21098? CVE-2023-21098 affects the following Android versions: Android-11, Android-12, Android-12L, and Android-13.

What is the base score assigned to CVE-2023-21098? The base score assigned to CVE-2023-21098 is 7.8, which is categorized as HIGH severity.

When was CVE-2023-21098 published? CVE-2023-21098 was published on 19 April 2023.

Where can more information about CVE-2023-21098 be found? More information about CVE-2023-21098 can be found at the following URL: <https://source.android.com/security/bulletin/2023-04/01>.

What kind of vulnerability is CVE-2023-21098? No user interaction is required to exploit the vulnerability detailed in CVE-2023-21098.

Could you provide an example of an attack scenario for CVE-2023-21098? A possible attack scenario for CVE-2023-21098 involves an attacker exploiting the confused deputy problem to gain root access.

What is CVE-2023-21097? CVE-2023-21097 is a security vulnerability identified in the Android operating system's Intent.java file.

What is the base score assigned to CVE-2023-21097? The base score assigned to CVE-2023-21097 is 7.8, which is classified as HIGH severity.

What are the affected versions of Android for CVE-2023-21097? The affected versions by CVE-2023-21097 include Android-11, Android-12, Android-12L, and Android-13.

Was user interaction required for CVE-2023-21097? No, user interaction was not required to exploit CVE-2023-21097.

What kind of vulnerability is CVE-2023-21097? Due to CVE-2023-21097, an attacker could exploit the confused deputy problem in the toUriInner method.

When was CVE-2023-21097 published? CVE-2023-21097 was published on 19 April 2023.

Where can more information about CVE-2023-21097 be found? You can find more information about CVE-2023-21097 on the Android Security Bulletin website at <https://source.android.com/security/bulletin/2023-04/01>.

What is the CVE ID of the recently discovered vulnerability affecting Android? The CVE ID of the recently discovered vulnerability affecting Android is CVE-2023-21096.

What kind of vulnerability is CVE-2023-21096? CVE-2023-21096 represents a use after free vulnerability in the OnWakelockReleased function of attribution\_processor.cc.

How severe is CVE-2023-21096? The CVE-2023-21096 vulnerability has been assigned a base score of 9.8, which classifies it as CRITICAL severity.

Does CVE-2023-21096 require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21096.

Which Android versions are affected by CVE-2023-21096? The Android versions affected by CVE-2023-21096 are Android-12, Android-12L, and Android-13.

Where can more details about the CVE-2023-21096 vulnerability be found? More details about the CVE-2023-21096 vulnerability can be found in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-04/01>.

When was the CVE-2023-21096 vulnerability published? The CVE-2023-21096 vulnerability was published on 19 April 2023.

What could happen by exploiting CVE-2023-21096? By exploiting CVE-2023-21096, attackers could perform remote code execution on the affected device.

What is the Android ID associated with CVE-2023-21096? The Android ID associated with CVE-2023-21096 is A-254774758.

Can you provide an example of an attack scenario for CVE-2023-21096? As CVE-2023-21096 is a use after free vulnerability in attribution\_processor.cc, a potential exploit could involve sending a crafted packet to the device.

Are there any mitigations for CVE-2023-21096? Mitigations for CVE-2023-21096 typically would involve applying security patches provided by Android.

What is CVE-2023-21094? CVE-2023-21094 is a security vulnerability identified in the LayerState.cpp file's sanitize function within the Android operating system.

What are the affected versions of Android for CVE-2023-21094? By exploiting CVE-2023-21094, a malicious entity could gain control over the device's screen display.

Does CVE-2023-21094 require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21094. An attacker could exploit this vulnerability to gain control over the device's screen display.

What is the base score assigned to CVE-2023-21094? CVE-2023-21094 has been assigned a CVSS base score of 7.8, classified as HIGH. This score indicates that the vulnerability is of high severity.

How can CVE-2023-21094 be mitigated? More information and updates regarding CVE-2023-21094 can be found in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-04/01>.

What versions of Android are affected by CVE-2023-21094? CVE-2023-21094 affects multiple versions of Android, specifically Android 11, Android 12, Android 12L, and Android 13.

Can you pr An example of an attack scenario involving CVE-2023-21094 could involve a malicious app that is executed on a device. What steps: If a device is identified as vulnerable to CVE-2023-21094, it is important to check for updates provided by the manufacturer. What is CV CVE-2023-21093 is associated with a vulnerability in the extractRelativePath method of FileUtils.java in the Android framework. What versi The versions of Android affected by CVE-2023-21093 are Android 11, Android 12, Android 12L, and Android 13. What is the CVE-2023-21093 has a severity rating of 'HIGH' with a base score of 7.8.

Was user i No, user interaction was not needed for the exploitation of CVE-2023-21093.

When was CVE-2023-21093 was published on 19 April 2023.

Can you pr Yes, additional details on CVE-2023-21093 can be found at the Android Security Bulletin: <https://source.android.com/security/bulletin/2023-04-19-01>.

Could you An attack scenario for CVE-2023-21093 could involve a malicious application exploiting the path traversal vulnerability.

Are there a In the case of CVE-2023-21093, a specific code example isn't provided. However, a path traversal vulnerability was identified.

What is CV CVE-2023-21092 refers to a security vulnerability identified in certain versions of Android where there was a local escalation of privilege.

Which And The Android versions affected by CVE-2023-21092 include Android 11, Android 12, Android 12L, and Android 13.

What is the CVE-2023-21092 The Base Score assigned to CVE-2023-21092 is 7.8, and it is categorized as HIGH severity.

On what date CVE-2023-21092 was published on 19 April 2023.

Is user inte No, user interaction is not required to exploit the vulnerability described in CVE-2023-21092.

Are there a Yes, more information on CVE-2023-21092 can be found in the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-04-19-01>.

Can you de In a possible attack scenario for CVE-2023-21092, a malicious app installed on the device could exploit the vulnerability to gain elevated privileges.

What type CVE-2023-21092 is a local escalation of privilege vulnerability that allows a malicious app or process to gain elevated privileges.

What mea: To mitigate CVE-2023-21092, users should apply updates provided by Google or their device manufacturer.

What is the CVE ID for the vulnerability that permits changing system app locales without a proper permission? CVE-2023-21091.

Can you de CVE-2023-21091 describes a vulnerability in the canDisplayLocalUi function of AppLocalePickerActivity in the Android framework.

Which vers Android version 13 is affected by CVE-2023-21091.

What is the CVE-2023-21091 has been assigned a base score of 5.5, which is categorized as MEDIUM severity. This indicates that the vulnerability has a moderate impact.

What is the CVE-2023-21091 was published on 19 April 2023.

Where can More information about CVE-2023-21091 can be found at the following URL: <https://source.android.com/security/bulletin/2023-04-19-01>.

What kind No user interaction is needed to exploit the vulnerability described in CVE-2023-21091; it can be exploited by a malicious app.

What pote An exploitation of CVE-2023-21091 can lead to a local denial of service on the affected Android system.

Is there a c As a responsible entity, we do not provide or condone the distribution of exploit code. However, CVE-2023-21091 is a vulnerability that can be exploited.

What kind Possible attack scenarios enabled by CVE-2023-21091 include an attacker creating a malicious application that exploits the vulnerability.

What is the CVE-2023-21091 The vulnerability involving a possible boot loop due to resource exhaustion is identified by the CVE ID CVE-2023-21091.

Can you de CVE-2023-21090 is a vulnerability in the parseUsesPermission function of ParsingPackageUtils.java in the Android framework.

What are the affected versions The affected version of Android impacted by CVE-2023-21090 is Android 13.

What is the CVE-2023-21090 has a CVSS base score of 5.0, which is classified as MEDIUM severity.

When was CVE-2023-21090 was published on 19 April 2023.

Where can More information about CVE-2023-21090 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-04-19-01>.

What code The code file associated with CVE-2023-21090 is ParsingPackageUtils.java, specifically within the parseUsesPermission method.

What kind To exploit the vulnerability identified by CVE-2023-21090, user interaction is needed; however, the exploit can be automated.

Could you Attack scenarios for CVE-2023-21090 could involve an attacker convincing a user to download and install a malicious application.

What is CV CVE-2023-21089 refers to a security vulnerability located in startInstrumentation of ActivityManagerService in the Android framework.

What is the CVE-2023-21089 The impact of CVE-2023-21089 is rated as 'HIGH' with a base score of 7.8. This indicates that the vulnerability has a high impact.

Do users n No, user interaction is not required for the exploitation of CVE-2023-21089. The vulnerability can be exploited by a malicious app.

When was CVE-2023-21089 was published on April 19, 2023.

Are there a Yes, CVE-2023-21089 is discussed in an official Android security bulletin, which is available at: <https://source.android.com/security/bulletin/2023-04-19-01>.

Can you pr An attacker could potentially exploit CVE-2023-21089 by crafting a malicious application that uses the startInstrumentation method.

What reme To address CVE-2023-21089, users should apply the security patches and updates provided by the Android manufacturer.

What is the CVE ID of the vulnerability found in LocationProviderManager.java is CVE-2023-21088.

What type CVE-2023-21088 is a vulnerability that involves a possible way to bypass background activity launch restrictions.

Does the e No, user interaction is not needed for the exploitation of CVE-2023-21088.

Which And The Android versions affected by CVE-2023-21088 are Android 12, Android 12L, and Android 13.

What is the severity rating of CVE-2023-21088 is 7.8, categorized as HIGH.

When was CVE-2023-21088 was published on 19 April 2023.

Where can More information about CVE-2023-21088 can be found at the following URL: [https://source.android.c](https://source.android.com/security/bulletin/2023-04-19-01)

Can you ex Potential attack scenarios for CVE-2023-21088 may include an attacker leveraging the logic error in th

Is there a c Unfortunately, specific code examples demonstrating the vulnerability described in CVE-2023-21088 ;

What is the CVE ID of the vulnerability is CVE-2023-21087.

Can you de CVE-2023-21087 represents a vulnerability where an uncaught exception in PreferencesHelper.java c;

Which And CVE-2023-21087 affects Android versions 11, 12, 12L, and 13.

How sever The CVE-2023-21087 vulnerability has a Base Score of 5.5 and is rated as MEDIUM severity according

When was CVE-2023-21087 was published on 19 April 2023.

Is user inte No, user interaction is not needed to exploit CVE-2023-21087.

Where can More details about CVE-2023-21087 can be found at the Android Security Bulletin page: [https://sourc](https://source.android.com/security/bulletin/2023-04-19-02)

What privil To exploit CVE-2023-21087, no additional execution privileges are needed. This vulnerability takes ad

Can you pr A possible attack scenario for CVE-2023-21087 could involve a malicious application triggering the unc

What is CV CVE-2023-21086 refers to a security vulnerability found in certain Android versions including Android

How sever CVE-2023-21086 has been assigned a base score of 7.8, which is categorized as HIGH severity. This inc

Can you gi An attack scenario for CVE-2023-21086 could involve a malicious actor with access to a Guest account

When was CVE-2023-21086 was published on 19 April 2023.

Where can More information about CVE-2023-21086 can be found at the Android Security Bulletin URL: [https://s](https://source.android.com/security/bulletin/2023-04-19-03)

What Andr Android versions affected by CVE-2023-21086 include Android 11, Android 12, Android 12L, and Andri

Is user inte No, user interaction is not required to exploit CVE-2023-21086. The vulnerability can be exploited with

What is the CVE ID for the vulnerability that involves a possible out of bounds write in nci\_snd\_set\_routing\_c

What type To exploit the vulnerability identified by CVE-2023-21085, no additional execution privileges are need

Does explo No, exploitation of CVE-2023-21085 does not require user interaction.

Which vers The versions of Android affected by CVE-2023-21085 are Android-11, Android-12, Android-12L, and Ai

What is the severity score assigned to CVE-2023-21085 is 8.8, which is classified as HIGH.

What is the official source of security information regarding CVE-2023-21085 can be found at the Android Sec

What kind CVE-2023-21085 could lead to remote code execution where the attacker could be proximal or adjace

Discuss po Potential attack scenarios for CVE-2023-21085 could involve an attacker within the radio communicat

What is CV CVE-2023-21084 is a security vulnerability identified in the 'buildPropFile' function within the 'filesyste

What is the base score assigned to CVE-2023-21084 is 6.7, which classifies it as a medium severity issue.

What versi CVE-2023-21084 affects Android version 13.

How can C CVE-2023-21084 can be exploited locally by an attacker with system execution privileges. The attacke

Where can More information about CVE-2023-21084 can be found on the Android Security Bulletin page at the fc

What are t The potential consequences of an attack exploiting CVE-2023-21084 include a local escalation of privil

Was any us No, user interaction is not needed to exploit CVE-2023-21084. This implies that the vulnerability could

When was CVE-2023-21084 was publicly disclosed on the 19th of April, 2023.

What is the CVE ID for the permissions bypass issue is CVE-2023-21083.

What is the CVE-2023-21083 has been assigned a base score of 7.8, which is categorized as HIGH severity.

Which vers The vulnerability CVE-2023-21083 affects Android versions Android-11, Android-12, Android-12L, and

Is user inte No, user interaction is not required to exploit the vulnerability described in CVE-2023-21083.

What type To exploit CVE-2023-21083, a perpetrator would need User execution privileges.

What pote Exploitation of CVE-2023-21083 could lead to a local escalation of privilege, which might allow an atta

Where can More information or the official advisory for CVE-2023-21083 can be found at [https://source.android.](https://source.android.com/security/bulletin/2023-04-19-04)

Can you pr CVE-2023-21083 is a vulnerability in the onNullBinding of CallScreeningServiceHelper.java, where the



On what date CVE-2023-21083 was published on 19 April 2023.

What are the Possible attack scenarios for CVE-2023-21083 include an attacker with user execution privileges exploiting the vulnerability.

What is the CVE ID of the vulnerability is CVE-2023-21082.

Can you describe CVE-2023-21082 involves a potential information disclosure issue in getNumberFromCallIntent of NewIntent.

What versions CVE-2023-21082 affects Android versions Android-11, Android-12, Android-12L, and Android-13.

What are the To exploit CVE-2023-21082, an attacker would need User execution privileges. No user interaction is required.

What is the CVE-2023-21082 has been assigned a CVSS Base Score of 5.5, categorizing it as MEDIUM severity.

When was CVE-2023-21082 was publicly disclosed on 19 April 2023.

Where can More information about CVE-2023-21082 can be found in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-04>

What could For CVE-2023-21082, an attacker with user execution privileges could potentially exploit the confused state of the system.

Are there any As CVE-2023-21082 involves proprietary Android code, code examples demonstrating the vulnerability are not typical.

What is the CVE ID of the discovered vulnerability is CVE-2023-21081.

What kind CVE-2023-21081 identifies a vulnerability that allows for bypassing background activity launch restrictions.

Which And The Android versions affected by CVE-2023-21081 are Android 11, Android 12, Android 12L, and Android 13.

Does exploit No, exploitation of CVE-2023-21081 does not require user interaction.

What is the The base score assigned to CVE-2023-21081 is 7.8, which is considered high.

On what date CVE-2023-21081 was published on 19 April 2023.

Where can More information about CVE-2023-21081 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-04>

What is the The Android ID associated with CVE-2023-21081 is A-230492955.

Could you In a possible attack scenario for CVE-2023-21081, a malicious app installed on the device could exploit the vulnerability.

Are there any As CVE-2023-21081 is a security vulnerability, code examples demonstrating the exploit are not typical.

What is the CVE ID of the vulnerability is CVE-2023-21080.

Which file The vulnerability CVE-2023-21080 affects the file btif\_rc.cc.

What type CVE-2023-21080 is an out of bounds read vulnerability due to a missing bounds check.

What are the The potential consequence of exploiting CVE-2023-21080 could lead to local information disclosure with user interaction.

Is user interaction No, user interaction is not required for the exploitation of CVE-2023-21080.

Which And The Android versions impacted by CVE-2023-21080 are Android-11, Android-12, Android-12L, and Android-13.

What is the The severity level assigned to CVE-2023-21080 is '5.5 MEDIUM'.

When was CVE-2023-21080 was published on 19 April 2023.

Where can More information about CVE-2023-21080 is available at the Android Security Bulletin website: <https://source.android.com/security/bulletin/2023-04>

Can you provide An attacker could potentially exploit CVE-2023-21080 by executing a crafted application that invokes the vulnerable function.

What is CV CVE-2023-20967 refers to a vulnerability in the avdt\_scb\_hdl\_pkt\_no\_frag function of the avdt\_scb\_adapter module.

What is the The Base Score assigned to CVE-2023-20967 is 7.8, which is categorized as HIGH.

On what date CVE-2023-20967 was published on 19 April 2023.

Does CVE-2023-20967 No, CVE-2023-20967 does not require user interaction for exploitation.

Can you provide CVE-2023-20967 can be exploited by a malicious app or attacker with code that triggers the vulnerable function.

What refer For further information on CVE-2023-20967, the following reference is provided: '<https://source.android.com/security/bulletin/2023-04>'

What versions The Android versions affected by CVE-2023-20967 are Android 11, Android 12, Android 12L (12.1), and Android 13.

Are there any A plausible attack scenario for CVE-2023-20967 could involve an attacker creating a malicious application that triggers the vulnerable function.

What is the CVE ID associated with the vulnerability in the AlarmManagerActivity.java file is CVE-2023-20950.

What type CVE-2023-20950 refers to a vulnerability that allows a potential bypass of background activity launch restrictions.

On what date The CVE-2023-20950 vulnerability was published on 19 April 2023.

Which And CVE-2023-20950 affects Android versions 11, 12, and 12L.

Can you provide More information about CVE-2023-20950 can be found on the Android security bulletin page: <https://source.android.com/security/bulletin/2023-04>

What are the To exploit the CVE-2023-20950 vulnerability, no additional execution privileges are needed, and no user interaction is required.

How severe CVE-2023-20950 is considered to have a severity level of HIGH, with a base score of 7.8.

What is a potential A potential attack scenario involving CVE-2023-20950 could be an attacker crafting a malicious application that triggers the vulnerable function.

What is CV CVE-2023-20941 refers to a security vulnerability identified in the 'acc\_ctrlrequest\_composite' function. What type CVE-2023-20941 is classified as an out of bounds write vulnerability that leads to a possible physical escalation of privilege. What are the details of CVE-2023-20941? Exploiting CVE-2023-20941 could allow a malicious entity to perform a physical escalation of privilege. What versions of Android are affected by CVE-2023-20941? CVE-2023-20941 affects unspecified versions of the Android kernel. Details about the specific affected versions can be found in the Android Security Bulletin. What is the CVSS base score of CVE-2023-20941? The base score of CVE-2023-20941 is 6.6, which is categorized as MEDIUM severity.

When was CVE-2023-20941 published? CVE-2023-20941 was published on 19 April 2023.

What kind of interaction is required for CVE-2023-20941? Exploitation of CVE-2023-20941 requires some form of user interaction, although the exact nature of the interaction is not specified. Where can more information about CVE-2023-20941 be found? More information about CVE-2023-20941 can be found in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-04>.

Can you describe a possible attack scenario involving CVE-2023-20941? A possible attack scenario involving CVE-2023-20941 might involve an attacker crafting a malicious application that exploits the vulnerability to perform a physical escalation of privilege.

What is CV CVE-2023-20935 is a security vulnerability identified in Android where there is a possible out of bounds read vulnerability. Which Android versions are affected by CVE-2023-20935? The Android versions affected by CVE-2023-20935 are Android 11, Android 12, Android 12L, and Android 13.

What is the CVSS base score of CVE-2023-20935? The CVSS base score of CVE-2023-20935 is 5.5, which categorizes it as a MEDIUM severity level vulnerability. When was CVE-2023-20935 published? CVE-2023-20935 was published on 19 April 2023.

What type of vulnerability is CVE-2023-20935? CVE-2023-20935 is an out of bounds read vulnerability that is a result of a missing bounds check in the `RunningTasks.java` file. Does CVE-2023-20935 require user interaction to be exploited? No, CVE-2023-20935 does not require user interaction to be exploited. An attacker can take advantage of this vulnerability without the need for user interaction. Where can more details or updates about CVE-2023-20935 be found? More details or updates about CVE-2023-20935 can be found in the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-04>.

Can you provide an example of a potential attack scenario for CVE-2023-20935? An example of a potential attack scenario for CVE-2023-20935 might involve a malicious application that exploits the vulnerability to read sensitive data from memory. Does CVE-2023-20935 require additional execution privileges to exploit? No, CVE-2023-20935 does not require additional execution privileges to exploit. This vulnerability can be exploited by any application with the `android.permission.READ_LOGS` permission.

What is the CVE ID for the vulnerability involving a missing privilege check in `RunningTasks.java` which could lead to a possible physical escalation of privilege? The CVE ID for the vulnerability involving a missing privilege check in `RunningTasks.java` which could lead to a possible physical escalation of privilege is CVE-2023-20909. Which product is affected by the vulnerability CVE-2023-20909? The product affected by the vulnerability CVE-2023-20909 is Android.

What are the affected versions of Android for CVE-2023-20909? The affected versions of Android for CVE-2023-20909 are Android-11, Android-12, Android-12L, and Android-13. What is the CVSS base score of CVE-2023-20909? CVE-2023-20909 has been given a Base Score of 5.5 and is categorized as MEDIUM severity.

When was the CVE-2023-20909 vulnerability published? The CVE-2023-20909 vulnerability was published on 19 April 2023.

Can you provide more detailed information about CVE-2023-20909? Yes, more detailed information about CVE-2023-20909 can be found on the Android security bulletin page at <https://source.android.com/security/bulletin/2023-04>. Is user interaction required for the exploitation of CVE-2023-20909? No, user interaction is not needed for the exploitation of CVE-2023-20909.

What is the CVE ID for the security issue in various functions of `RunningTasks.java` with a possible privilege escalation? The CVE ID for the security issue in various functions of `RunningTasks.java` with a possible privilege escalation is CVE-2023-20909. What kind of interaction is required for CVE-2023-20909? By exploiting the CVE-2023-20909 vulnerability, an attacker might gain unauthorized access to sensitive data without the need for user interaction.

Are there any code examples or patches to address CVE-2023-20909? To find code examples or patches to address CVE-2023-20909, one would typically look at the official Android Security Bulletin or the Android Open Source Project (AOSP) for any relevant updates. What is the CVE ID of the vulnerability found in the PowerVR kernel driver? The CVE ID of the vulnerability found in the PowerVR kernel driver is CVE-2021-0885.

What component of the PowerVR kernel driver is affected by CVE-2021-0885? CVE-2021-0885 affects the `PVRSRVBridgeSyncPrimOpTake` of the PowerVR kernel driver.

Can you explain the security issue in CVE-2021-0885? The security issue in CVE-2021-0885 pertains to a missing size check that could lead to a possible integer overflow. Is user interaction required to exploit the vulnerability described in CVE-2021-0885? No, user interaction is not required to exploit the vulnerability described in CVE-2021-0885.

What are the affected versions of Android for CVE-2021-0885? CVE-2021-0885 affects Android, specifically on systems with an Android SoC (System on Chip).

What can you tell me about the severity of CVE-2021-0885? The severity of CVE-2021-0885 is rated as 7.8, which is considered HIGH according to its Base Score.

When was the vulnerability CVE-2021-0885 first published? The vulnerability CVE-2021-0885 was first published on 19 April 2023.

Where can more information about CVE-2021-0885 be found? More information about CVE-2021-0885 can be found on the Android Security Bulletin at the URL: <https://source.android.com/security/bulletin/2023-04>. What is the Android ID associated with CVE-2021-0885? The Android ID associated with CVE-2021-0885 is A-270401914.

Can you describe a potential attack scenario exploiting CVE-2021-0885? A potential attack scenario exploiting CVE-2021-0885 would involve an attacker crafting a specific operation that triggers the integer overflow vulnerability in the `PVRSRVBridgeSyncPrimOpTake` function. What is the CVE ID for the vulnerability found in the PowerVR kernel driver? The CVE ID for the vulnerability found in the PowerVR kernel driver is CVE-2021-0884.

Can you describe the security flaw in the `PVRSRVBridgePhysemImportSparseDmaBuf` of the PowerVR kernel driver? CVE-2021-0884 is a security flaw in the `PVRSRVBridgePhysemImportSparseDmaBuf` of the PowerVR kernel driver. What is the CVSS base score for CVE-2021-0884? The CVSS base score for CVE-2021-0884 is 7.8, which is categorized as HIGH. This score indicates that the vulnerability has a high impact on the confidentiality, integrity, and availability of the system.

When was the CVE-2021-0884 vulnerability published? The CVE-2021-0884 vulnerability was published on 19 April 2023.

Which versions of Android are affected by CVE-2021-0884? CVE-2021-0884 affects certain versions of Android for SoC (System on Chip). The specific versions impacted are listed in the Android Security Bulletin. Is user interaction required to exploit the vulnerability described in CVE-2021-0884? No, user interaction is not required to exploit the vulnerability described in CVE-2021-0884.

What is the Android ID associated with CVE-2021-0884? The Android ID associated with CVE-2021-0884 is A-270393454.

Where can more information about CVE-2021-0884 be found? More information about CVE-2021-0884 can be found at the following source: <https://source.android.com/security/bulletin/2023-04>.

What are the potential attack scenarios for CVE-2021-0884? Potential attack scenarios for CVE-2021-0884 include an attacker exploiting the integer overflow vulnerability to cause a system crash or to execute arbitrary code.

What is the CVE ID for the vulnerability found in the PowerVR kernel driver? CVE-2021-0883.

Can you describe CVE-2021-0883? CVE-2021-0883 refers to a vulnerability in the PVRSRVBridgeCacheOpQueue of the PowerVR kernel driver.

What is the Base Score assigned to CVE-2021-0883? The Base Score assigned to CVE-2021-0883 is 7.8, which is categorized as HIGH severity.

On which date was CVE-2021-0883 published? CVE-2021-0883 was published on 19 April 2023.

Which devices are affected by CVE-2021-0883? CVE-2021-0883 affects devices with Android SoCs that utilize the PowerVR kernel driver. The vulnerability is user interaction not required.

Where can I find more information about CVE-2021-0883? Additional information about CVE-2021-0883 can be found on the Android security bulletin page at <https://source.android.com/security/bulletin/2023-04/01>.

What could be a possible attack scenario for CVE-2021-0883? A possible attack scenario for CVE-2021-0883 could involve a malicious local application exploiting the vulnerability.

Are there any code examples for CVE-2021-0883? As an AI, I cannot provide actual code examples for exploitations due to ethical considerations. However, I can provide a high-level overview of the vulnerability.

Has CVE-2021-0883 been patched? As of the published data provided in the security bulletin reference, CVE-2021-0883 has likely been addressed in a subsequent update.

What is CVE-2021-0882? CVE-2021-0882 refers to a specific security vulnerability identified in the PowerVR kernel driver affecting certain Android devices.

How severe is CVE-2021-0882? The CVE-2021-0882 vulnerability is considered 'HIGH' severity with a base score of 7.8, indicating that it is a significant security issue.

What are the details of CVE-2021-0882? The CVE-2021-0882 vulnerability arises from a missing size check within the PVRSRVBridgeRGXKickSync function.

When was CVE-2021-0882 published? CVE-2021-0882 was published on 19 April 2023.

Are there any code examples for CVE-2021-0882? Yes, you can find more information about CVE-2021-0882 on the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-04/01>.

What type of attack is CVE-2021-0882? To exploit the CVE-2021-0882 vulnerability, no additional execution privileges are needed. This means it is a local privilege escalation vulnerability.

Is user interaction required for CVE-2021-0882? No, user interaction is not required to exploit the CVE-2021-0882 vulnerability.

What are the possible attack scenarios for CVE-2021-0882? Possible attack scenarios for CVE-2021-0882 could include a local attacker writing a malicious application that exploits the vulnerability.

What is CVE-2021-0881? CVE-2021-0881 is a security vulnerability found in the PVRSRVBridgeRGXKickCDM of the PowerVR kernel driver.

How can CVE-2021-0881 be exploited? An attacker can exploit CVE-2021-0881 by interacting with the PVRSRVBridgeRGXKickCDM component.

Does CVE-2021-0881 require user interaction? No, CVE-2021-0881 does not require user interaction for exploitation. An attacker can exploit the vulnerability without user interaction.

On which platform is CVE-2021-0881 found? CVE-2021-0881 is found on the Android platform, specifically on Android SoC. It is a vulnerability in the kernel driver.

What is the CVSS base score for CVE-2021-0881? The CVSS base score for CVE-2021-0881 is 7.8, which is categorized as HIGH severity.

When was CVE-2021-0881 published? CVE-2021-0881 was published on 19 April 2023.

Where can I find more information about CVE-2021-0881? More information about CVE-2021-0881 can be found on the Android security bulletin at <https://source.android.com/security/bulletin/2023-04/01>.

What is CVE-2021-0880? CVE-2021-0880 is a vulnerability found in the PVRSRVBridgeRGXKickTA3D of the PowerVR kernel driver.

How severe is CVE-2021-0880? The security threat posed by CVE-2021-0880 is considered HIGH, with a base score of 7.8. This means it is a significant security issue.

What products are affected by CVE-2021-0880? CVE-2021-0880 affects certain versions of products that use the PowerVR kernel driver, specifically those with Android SoCs.

Is user interaction required for CVE-2021-0880? No, user interaction is not required to exploit CVE-2021-0880. An attacker could leverage the vulnerability without user interaction.

What type of attack is CVE-2021-0880? CVE-2021-0880 could facilitate a local attack that leads to escalation of privilege. As it allows out-of-band access, it is a remote-to-local attack.

Where can I find more information about CVE-2021-0880? More information about CVE-2021-0880 can be found on the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-04/01>.

When was CVE-2021-0880 published? CVE-2021-0880 was published on 19 April 2023.

Are there any code examples for CVE-2021-0880? As an industry standard, code examples for exploiting vulnerabilities such as CVE-2021-0880 are typically not provided due to ethical considerations.

What are the possible mitigation strategies for CVE-2021-0880? Possible mitigation strategies for CVE-2021-0880 include updating the affected PowerVR kernel driver to the latest version.

What attack scenario is associated with CVE-2021-0880? An attacker with local access to an affected system could exploit CVE-2021-0880 to perform an out-of-band attack.

What is CVE-2021-0879? CVE-2021-0879 refers to a security vulnerability identified in the PowerVR kernel driver, specifically within the PVRSRVBridgeServerSyncGetStatus function.

How severe is CVE-2021-0879? The security issue identified by CVE-2021-0879 is rated with a Base Score of 7.8, which is considered HIGH severity.

Which products are affected by CVE-2021-0879? CVE-2021-0879 affects Android products that utilize the PowerVR kernel driver. The Android System Core is affected.

What does CVE-2021-0879 allow? CVE-2021-0879 allows attackers to perform local escalation of privilege without requiring any additional user interaction.

Do users need to interact for CVE-2021-0879? No, user interaction is not needed for the exploitation of CVE-2021-0879. An attacker can exploit this vulnerability without user interaction.

When was CVE-2021-0879 publicly disclosed? CVE-2021-0879 was publicly disclosed on the 19th of April, 2023.

Where can I find more information about CVE-2021-0879? More information about CVE-2021-0879 can be found on the Android Security Bulletin website at the following link: <https://source.android.com/security/bulletin/2023-04/01>.

What are the potential attack scenarios for CVE-2021-0879? Potential attack scenarios for CVE-2021-0879 include a malicious app installed on the device that exploits the vulnerability.

What is CVE-2021-0878? CVE-2021-0878 refers to a vulnerability found in the PVRSRVBridgeServerSyncGetStatus function of the PowerVR kernel driver.

How severe is CVE-2021-0878? The security issue described by CVE-2021-0878 has been given a Base Score of 7.8, which is classified as HIGH severity.

On which platform is CVE-2021-0878 found? CVE-2021-0878 was published on 19 April 2023.

Which products are affected by CVE-2021-0878? The products affected by CVE-2021-0878 are the Android devices running on systems-on-chip (SoCs) that utilize the PowerVR kernel driver.

Are there any? Yes, more information about CVE-2021-0878 can be found at the Android Security Bulletin, accessible at <https://source.android.com/security/bulletin/2023-04/0878>.  
What type of vulnerability is CVE-2021-0878? CVE-2021-0878 is an integer overflow vulnerability that can result in out-of-bounds heap access. Consider the following: Is user interaction required to exploit the vulnerability described in CVE-2021-0878, which may lead to a local escalation of privilege? No, user interaction is not required to exploit CVE-2021-0878, which may lead to a local escalation of privilege.  
Can you provide a hypothetical attack scenario for CVE-2021-0878? A hypothetical attack scenario for CVE-2021-0878 could be a malicious application installed on an Android device. The application could trigger the vulnerability, leading to a local escalation of privilege.  
What is CVE-2021-0876? CVE-2021-0876 is a security vulnerability found in the PowerVR kernel driver. Specifically, the flaw exists in the `PVR_SRV_BridgeDeviceMemHistorySparseCache` function.  
How severe is CVE-2021-0876? The vulnerability described in CVE-2021-0876 has been given a base score of 7.8, which is rated as HIGH severity.  
Was user interaction required to exploit CVE-2021-0876? No, user interaction is not required to exploit CVE-2021-0876. The exploitation can occur without any user interaction.  
Which products are affected by CVE-2021-0876? The product affected by CVE-2021-0876 is Android, specifically versions running on systems with a System on Chips (SoCs) that use the PowerVR kernel driver.  
When was CVE-2021-0876 publicly disclosed? CVE-2021-0876 was publicly disclosed on 19 April 2023.

What is the Android ID associated with CVE-2021-0876? The Android ID associated with CVE-2021-0876 is A-270400229.

Where can I find more information about CVE-2021-0876? More information about CVE-2021-0876 can be found at the Android Security Bulletin, specifically at <https://source.android.com/security/bulletin/2023-04/0876>.  
Can you explain a potential attack scenario for exploiting CVE-2021-0876? A potential attack scenario for exploiting CVE-2021-0876 would involve a malicious application that is installed on an Android device. The application could trigger the vulnerability, leading to a local escalation of privilege.  
What is the CVE ID of the vulnerability found in the PowerVR kernel driver? The CVE ID of the vulnerability found in the PowerVR kernel driver is CVE-2021-0875.

What is the base score assigned to CVE-2021-0875? The base score assigned to CVE-2021-0875 is 7.8, which is categorized as HIGH severity.

On what date was CVE-2021-0875 published? CVE-2021-0875 was published on 19 April 2023.

What type of vulnerability is CVE-2021-0875? The products affected by CVE-2021-0875 include Android devices with the PowerVR kernel driver.

What is the CVE ID of the vulnerability involving a missing size check that could lead to an integer overflow? CVE-2021-0875 is a vulnerability involving a missing size check that could lead to an integer overflow, which may lead to a local escalation of privilege.  
What are the products affected by CVE-2021-0875? Exploiting CVE-2021-0875 could lead to local escalation of privilege without any additional execution. Does exploitation of CVE-2021-0875 require user interaction? No, exploitation of CVE-2021-0875 does not require user interaction.

Where can I find more information about CVE-2021-0875? More information about CVE-2021-0875 can be found in the Android Security Bulletin at: <https://source.android.com/security/bulletin/2023-04/0875>.

Can you describe a possible attack scenario for CVE-2021-0875? A possible attack scenario for CVE-2021-0875 includes an attacker with local access to an Android device. The attacker could trigger the vulnerability, leading to a local escalation of privilege.  
Can you provide a hypothetical attack scenario for CVE-2021-0875? Due to the nature of the CVE-2021-0875 vulnerability being a security flaw in a proprietary kernel driver, a hypothetical attack scenario is not provided.

What is CVE-2021-0874? CVE-2021-0874 is a security vulnerability identified in the `PVR_SRV_BridgeDeviceMemHistorySparseCache` function of the PowerVR kernel driver.  
How severe is CVE-2021-0874? The vulnerability CVE-2021-0874 has been rated with a base score of 7.8, which is classified as HIGH severity.  
Was user interaction required to exploit CVE-2021-0874? No, user interaction was not required to exploit CVE-2021-0874. This means an attacker could potentially exploit the vulnerability without any user interaction.  
Which products are affected by CVE-2021-0874? CVE-2021-0874 affects the Android operating system on System on Chips (SoCs) that use the PowerVR kernel driver.  
When was CVE-2021-0874 officially published? CVE-2021-0874 was officially published on 19 April 2023.

What are the products affected by CVE-2021-0874? Exploiting CVE-2021-0874 could potentially lead to a local escalation of privilege. An attacker with access to an Android device could trigger the vulnerability, leading to a local escalation of privilege.

Are there any? Yes, more information about CVE-2021-0874 can be found on the Android Security Bulletin page at <https://source.android.com/security/bulletin/2023-04/0874>.  
Can you provide an attack scenario for CVE-2021-0874? An attack scenario for CVE-2021-0874 could involve an attacker developing a malicious application that triggers the vulnerability, leading to a local escalation of privilege.

What is CVE-2021-0873? CVE-2021-0873 refers to a security vulnerability located in the PowerVR kernel driver, specifically in the `PVR_SRV_BridgeDeviceMemHistorySparseCache` function.  
How severe is CVE-2021-0873? The base score for CVE-2021-0873 is rated 7.8 out of 10, which classifies it as HIGH severity. This signifies a high level of severity.

Which products are affected by CVE-2021-0873? CVE-2021-0873 affects the Android operating system on System on Chips (SoCs) that incorporate the PowerVR kernel driver.  
Is user interaction required to exploit CVE-2021-0873? No, user interaction is not required to exploit CVE-2021-0873. An attacker could take advantage of this vulnerability without any user interaction.  
What is the impact of CVE-2021-0873? If an attacker successfully exploits CVE-2021-0873, they could achieve local escalation of privilege on the affected device.  
When was CVE-2021-0873 published? CVE-2021-0873 was published on April 19, 2023.

Where can I find more information about CVE-2021-0873? More information about CVE-2021-0873 is available in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-04/0873>.  
Could you provide a hypothetical attack scenario for CVE-2021-0873? While I do not have access to the exact proprietary code of the PowerVR kernel driver, a generic example of a hypothetical attack scenario for CVE-2021-0873 would involve a local attacker using a specially crafted application to trigger the vulnerability, leading to a local escalation of privilege.

What are the products affected by CVE-2021-0873? Potential attack scenarios for CVE-2021-0873 include a local attacker using a specially crafted application to trigger the vulnerability, leading to a local escalation of privilege.  
What is CVE-2021-0872? CVE-2021-0872 refers to a security vulnerability found in the `PVR_SRV_BridgeRGXKickVRDM` of the PowerVR kernel driver.  
How severe is CVE-2021-0872? CVE-2021-0872 is rated with a base score of 7.8, which is categorized as HIGH severity. This severity level indicates a high level of severity.

Was there user interaction required to exploit CVE-2021-0872? No, CVE-2021-0872 could be exploited without any user interaction. This increases the risk associated with the vulnerability.  
Which products are affected by CVE-2021-0872? CVE-2021-0872 affects Android devices with System on Chips (SoCs) that use the PowerVR kernel driver.

Where can I find more detailed information about CVE-2021-0872? More detailed information about CVE-2021-0872 can be found on Android's security bulletin page at <https://source.android.com/security/bulletin/2023-04/0872>.

What is the Android ID associated with CVE-2021-0872? The Android ID associated with CVE-2021-0872 is A-270401229. This identifier can be used to reference the vulnerability.

What is the impact of CVE-2021-0872? The exploitation of CVE-2021-0872 can lead to local escalation of privilege. This means that a malicious application could potentially exploit the vulnerability to gain higher privileges on the device.

What is the date when CVE-2021-0872 was published? CVE-2021-0872 was published on 19 April 2023. This is the date when the vulnerability was publicly disclosed.

What is CV CVE-2023-27647 is a security vulnerability found in DUALSPACE Lock Master version 2.2.4 that allows How does CVE-2023-27647 affects a system by allowing a local attacker to cause a disruption in the service (den When was CVE-2023-27647 was published on 14 April 2023.

What are s Additional information about CVE-2023-27647 can be found through the following references:- A GitH Where cou A detailed explanation or example code demonstrating the vulnerability CVE-2023-27647 might be fo What woul An attack scenario for CVE-2023-27647 could involve a local attacker with access to the device where What is CV CVE-2023-26980 is a vulnerability in PAX Technology's PAX A920 Pro PayDroid operating on version 8. What is the The severity score of CVE-2023-26980 is classified as '7.0 HIGH'.

Which dev The device affected by CVE-2023-26980 is the PAX A920 Pro PayDroid running on version 8.1.

What is the The vendor of PAX A920 Pro PayDroid disputes the viability of the attack described in CVE-2023-26980 On what d CVE-2023-26980 was published on 14 April 2023.

Where can More information about CVE-2023-26980 can be found in the following references:- <https://uploads.s>

What is a p In a possible attack scenario for CVE-2023-26980, an attacker might exploit the race condition during t

What is the The CVE ID for the vulnerability associated with improper intent handling in certain mobile printing ap

Which mo The mobile printing apps affected by CVE-2023-25954 are 'KYOCERA Mobile Print,' 'UTAX/TA MobileP What is the CVE-2023-25954 has a base score of 5.5 and is classified as MEDIUM severity.

When was CVE-2023-25954 was published on 13 April 2023.

What kind CVE-2023-25954 describes a vulnerability related to improper intent handling, where a malicious app

Can you pr Yes, further information on CVE-2023-25954 can be found through the following references:- <https://v>

What is the The potential impact of CVE-2023-25954 on users includes the unauthorized download of malicious fil

What are t Possible attack scenarios for CVE-2023-25954 include a malicious actor creating and distributing an Ar

Is there an I do not have a specific code example to demonstrate how CVE-2023-25954 might be exploited. Howe

What is CV CVE-2023-27703 is a security vulnerability identified in the Android version of the PikPak application,

How sever The vulnerability described in CVE-2023-27703 is rated with a Base Score of 3.3, which classifies it as L

When was CVE-2023-27703 was published on 12 April 2023.

Where can You can find more information about CVE-2023-27703 from the following sources:1. A Google Drive fo

Could you Since CVE-2023-27703 involves an information leak through the debug interface, an example of explo

What might Possible attack scenarios for CVE-2023-27703 include unauthorized users gaining access to the debug

What is CV CVE-2023-22808 is a security vulnerability discovered in the Arm Android Gralloc Module, where a no

Which gra CVE-2023-22808 impacts Bifrost (r24p0 through r41p0 before r42p0), Valhall (r24p0 through r41p0 be

How sever CVE-2023-22808 has been given a Base Score of 3.3, which classifies it as a LOW severity vulnerability

When was CVE-2023-22808 was published on 11 April 2023.

Where can Additional details about CVE-2023-22808 are available on the Arm Security Center website at the foll

Can you pr As CVE-2023-22808 refers to a vulnerability allowing reading of memory, there isn't a specific code ex

What are t For CVE-2023-22808, an attacker with access to a non-privileged user account could exploit the vulner

What shou Users of the affected Arm graphics architectures should update their Android Gralloc Module to the la

What is the The CVE ID of the vulnerability found in the 'Wolt Delivery: Food and more' Android app is CVE-2023-2

Which vers Version 4.27.2 and earlier of the 'Wolt Delivery: Food and more' Android app are affected by CVE-202

What type CVE-2023-22429 describes a vulnerability resulting from the use of hard-coded credentials in the form

What is the CVE-2023-22429 has been assigned a CVSS base score of 7.8, which is categorized as HIGH severity. Th

When was CVE-2023-22429 was published on 11 April 2023.

Can you pr Yes, more information about CVE-2023-22429 can be found at the following URLs: <https://jvn.jp/en/jj>

What coul A possible attack scenario for CVE-2023-22429 would involve a local attacker reverse-engineering the

How shoul To prevent a vulnerability like CVE-2023-22429, developers should avoid using hard-coded credential

What is CV CVE-2023-1817 refers to a security vulnerability found in Google Chrome on Android. This issue is due

What is the The CVSS base score of CVE-2023-1817 is 6.5, which is categorized as MEDIUM severity.

What versi CVE-2023-1817 affects Google Chrome on Android versions prior to 112.0.5615.49.

What is the severity? The Chromium security team has assigned a severity level of 'Medium' to CVE-2023-1817.

On what date? CVE-2023-1817 was published on 04 April 2023.

Where can I find more information? More information about CVE-2023-1817 can be found in the references provided, including an official Chromium advisory.

Are there any attack scenarios? An attack scenario for CVE-2023-1817 would involve an attacker creating a malicious HTML page specifying a particular user agent.

Can you give me specific code examples? While specific code examples of exploiting CVE-2023-1817 are not typically shared to prevent misuse, the vulnerability is related to the User-Agent string.

What is the CVE ID for the vulnerability found in Nextcloud's desktop and mobile client applications? The CVE ID for the vulnerability found in Nextcloud's desktop and mobile client applications is CVE-2023-28999.

Which versions are affected? The Nextcloud Desktop client versions affected by CVE-2023-28999 range from 3.0.0 to 3.8.0, excluding versions 3.1.0 and 3.2.0.

What is the description of the vulnerability? CVE-2023-28999 describes a vulnerability where a malicious server administrator can gain full access to the Nextcloud client applications.

How was it addressed? The CVE-2023-28999 vulnerability was addressed through updates to the Nextcloud applications. Specific details are provided in the Nextcloud security advisory.

Are there any known workarounds? No known workarounds are available for CVE-2023-28999. The only remediation is to update to the fixed versions of the Nextcloud applications.

What is the CVSS score and severity? CVE-2023-28999 has been assigned a base score of 6.4 with a severity level of MEDIUM.

On what date was it published? CVE-2023-28999 was published on the 4th of April, 2023.

Where can I find detailed technical information and a security advisory? Detailed technical information and a security advisory on CVE-2023-28999 can be found through the Nextcloud security advisory page.

Can you provide a possible attack scenario? In a possible attack scenario for CVE-2023-28999, a malicious server administrator at a Nextcloud service provider could exploit the vulnerability to gain full access to the client applications.

Could you explain the vulnerability? Since CVE-2023-28999 involves proprietary code in the Nextcloud client applications interacting with the server, the details are not publicly disclosed.

What is the CVE ID for the vulnerability found in the Nextcloud Android app? CVE-2023-28646 is a security vulnerability found in the Nextcloud Android app versions from 3.7.0 up to 3.8.0.

How severe is it? CVE-2023-28646 has been assigned a Base Score of 2.4, which is categorized as LOW severity. This indicates a low risk of exploitation.

Which versions are affected? CVE-2023-28646 affects the Nextcloud Android app versions from 3.7.0 up until the version before 3.8.0.

What does the vulnerability allow? CVE-2023-28646 allows attackers to bypass the Pin/passcode protection on the Nextcloud Android app, potentially leading to unauthorized access.

What is the recommended action? The recommended action for addressing CVE-2023-28646 is to upgrade the Nextcloud Android app to the latest version where the vulnerability has been patched.

Are there any known workarounds? There are no known workarounds for CVE-2023-28646. The only recommended solution is to upgrade the Nextcloud Android app.

Where can I find additional information? Additional information about CVE-2023-28646 can be found on the GitHub security advisories page at <https://github.com/nextcloud/android/security/advisories>.

What could be a potential attack scenario? A potential attack scenario for CVE-2023-28646 would involve an attacker gaining physical access to a device and exploiting the vulnerability to bypass security measures.

What does CVE-2023-21079 refer to? CVE-2023-21079 refers to a security vulnerability in the `rtt_unpack_xtlv_cbfn` function of `dhd_rtt.c` in the Android kernel.

Which product is affected? The product affected by CVE-2023-21079 is Android, specifically a component in the Android Kernel.

Is user interaction needed? No, user interaction is not needed to exploit the security issue described by CVE-2023-21079.

What privileges are required? An attacker needs to have System execution privileges to successfully exploit CVE-2023-21079.

What is the CVSS score and severity? CVE-2023-21079 is assigned a base CVSS score of 6.7, and it is classified as 'MEDIUM' severity.

When was it published? CVE-2023-21079 was published on 24 March 2023.

Where can I find more information? More information about CVE-2023-21079 can be found at the Android Security Bulletin URL: <https://source.android.com/security/bulletin/2023-03>.

What are the possible attack scenarios? Possible attack scenarios for CVE-2023-21079 include a malicious application or an attacker with local access exploiting the vulnerability to escalate privileges.

What is the CVE ID of the reported vulnerability? The CVE ID of the reported vulnerability is CVE-2023-21078.

What is the description of the vulnerability? CVE-2023-21078 could lead to local escalation of privilege due to a buffer overflow, allowing an attacker to execute arbitrary code.

Which file is affected? The affected file in the Android kernel by CVE-2023-21078 is `dhd_rtt.c`, specifically in the function `rtt_unpack_xtlv_cbfn`.

Do users need to interact? No, user interaction is not needed for the exploitation of CVE-2023-21078.

What are the affected versions? CVE-2023-21078 affects versions of the Android kernel as referenced in the CVE description. The specific versions are listed in the Android Security Bulletin.

What is the CVSS score and severity? The Common Vulnerability Scoring System (CVSS) base score of CVE-2023-21078 is 6.7, which is categorized as MEDIUM severity.

When was it published? CVE-2023-21078 was published on 24 March 2023.

Where can I find more details? More details about CVE-2023-21078 can be found on the Android Security Bulletin page at the following URL: <https://source.android.com/security/bulletin/2023-03>.

Are there any attack scenarios? One possible attack scenario for CVE-2023-21078 is a malicious local application exploiting the buffer overflow to escalate privileges.

What is the Android ID associated with the vulnerability? The Android ID associated with CVE-2023-21078 is A-254840211.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-21077.

Can you describe the vulnerability? CVE-2023-21077 details a possible out of bounds write due to a buffer overflow in the function `'rtt_unpack_xtlv_cbfn'` in the Android kernel.

What type of privileges are required? An attacker would need System execution privileges to exploit CVE-2023-21077.

Is user interaction required? No, user interaction is not required to exploit the vulnerability described in CVE-2023-21077.

Which product is affected? The product affected by CVE-2023-21077 is the Android kernel, which is part of the Android operating system.

What is the CVSS score and severity? The CVSS base score assigned to CVE-2023-21077 is 6.7, which is categorized as MEDIUM severity.

When was CVE-2023-21077 published on 24 March 2023.

Where can More information about CVE-2023-21077 can be found in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-03-24>

What are the possible attack scenarios for CVE-2023-21077 include a malicious application or process running with

Are there any CVE-2023-21077 is a security vulnerability, specific exploit code examples are typically not provided

What is the CVE ID of the vulnerability is CVE-2023-21076.

Can you provide CVE-2023-21076 describes a vulnerability in the createTransmitFollowupRequest function of nan.cpp

What severity CVE-2023-21076 has been assigned a Base Score of 6.7, which is categorized as MEDIUM severity.

When was CVE-2023-21076 published on 24 March 2023.

Where can More information about CVE-2023-21076 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-24>

Is there any No, user interaction is not needed to exploit the vulnerability mentioned in CVE-2023-21076.

What are the versions of Android affected by CVE-2023-21076 are not specified other than mentioning it pertains to

What type An attacker would need System execution privileges in order to exploit the vulnerability described in CVE

What is the Exploiting CVE-2023-21076 could lead to a local escalation of privilege, where an attacker could execute

Can you describe A possible attack scenario for CVE-2023-21076 could involve a malicious app or process that already has

What is CVE-2023-21075 is a security vulnerability identified in the Android operating system. It is found in the

How severe CVE-2023-21075 has been assessed with a Base Score of 6.7, which is categorized as MEDIUM severity

Was user interaction No, user interaction is not needed for the exploitation of CVE-2023-21075. This means an attacker could

What privilege To exploit CVE-2023-21075, an attacker would need System execution privileges. This implies that the

Which Android CVE-2023-21075 affects the Android kernel, but the specific versions of Android that are vulnerable are

Where can More information about CVE-2023-21075 can be found in the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-03-24>

What is the Android ID associated with CVE-2023-21075 is A-261857862. This identifier can be used to track a

What kind Possible attack scenarios for CVE-2023-21075 include a local attacker with System execution privilege

What is CVE-2023-21073 refers to a security vulnerability identified in the Android kernel where there is a possible

What type CVE-2023-21073 is classified as a buffer overflow vulnerability that can result in an out of bounds write

How severe CVE-2023-21073 has been assigned a Base Score of 6.7, which is categorized as MEDIUM severity according

What are the To exploit CVE-2023-21073, an attacker would need System execution privileges. This means the attacker

Does CVE-2023-21073 No, CVE-2023-21073 can be exploited without any user interaction.

On what date CVE-2023-21073 was published on the 24th of March, 2023.

Where can More details about CVE-2023-21073 can be found at the Android Security Bulletin, specifically on the

Can you provide A potential attack scenario for CVE-2023-21073 could involve a malicious application that has already

What is the CVE ID for this vulnerability is CVE-2023-21072.

Can you explain CVE-2023-21072 refers to a buffer overflow issue occurring in the rtt\_unpack\_xtlv\_cbfn function of

What type To exploit the vulnerability identified by CVE-2023-21072, an attacker would need System execution privileges

Does the CVE-2023-21072 No, user interaction is not required for the exploitation of CVE-2023-21072.

What is the CVE-2023-21072 has been assigned a CVSS base score of 6.7, indicating a medium severity level.

When was The CVE-2023-21072 vulnerability was published on 24 March 2023.

Which product The product affected by CVE-2023-21072 is the Android kernel, a fundamental part of the Android operating

Where can More information about CVE-2023-21072 can be found at the following reference URL: <https://source.android.com/security/bulletin/2023-03-24>

What would A potential attack scenario for CVE-2023-21072 could involve an unauthorized attacker with system-level

Are there any CVE-2023-21072 is a security vulnerability, code examples elucidating how to exploit the flaw are not

What is CVE-2023-21071 is a security vulnerability identified in a component of the Android kernel, specifically in the

How severe The CVE-2023-21071 vulnerability has been rated with a base score of 6.7, which classifies it as MEDIUM

Which versions CVE-2023-21071 affects Android kernel versions prior to the one detailed in the security bulletin released

Is user interaction No, user interaction is not needed for exploitation of CVE-2023-21071. An attacker could exploit the vulnerability

Does CVE-2023-21071 CVE-2023-21071 details a local escalation of privilege, which means an attacker would need local access

What kind An attacker would need to have System execution privileges to exploit CVE-2023-21071. This implies that the

Can you provide A potential attack scenario for exploiting CVE-2023-21071 could involve an attacker who already has

Where can Additional information about CVE-2023-21071 can be found through the published Android Security Bulletin?   
What is the Users affected by CVE-2023-21071 should apply the relevant security patches provided by device manufacturers.   
On what date CVE-2023-21071 was published on March 24, 2023.

What is the The CVE ID for the vulnerability is CVE-2023-21070.

Can you describe The vulnerability CVE-2023-21070 is an out of bounds write issue in the 'add\_roam\_cache\_list' function.   
What execution To exploit the vulnerability CVE-2023-21070, an attacker would need System execution privileges.

Is user interaction No, user interaction is not required to exploit the vulnerability identified by CVE-2023-21070.

What is the The base score assigned to CVE-2023-21070 is 6.7 and it's classified as MEDIUM severity.

On which date CVE-2023-21070 was published on 24 March 2023.

Where can More information about CVE-2023-21070 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03/01>

What versions The affected versions of Android for CVE-2023-21070 are unspecified in this context, but it would typically affect versions from Android 12 to Android 13.

What could If CVE-2023-21070 is successfully exploited, an attacker could leverage the out of bounds write to gain unauthorized access to system data.

Are there any While specific code examples and patches are not provided in the context of this response, mitigation steps may be available.

What is the The CVE ID for the vulnerability with a possible out of bounds write in wl\_cfgscan.c is CVE-2023-21069.

What kind CVE-2023-21069 describes an issue with a possible out of bounds write due to a missing bounds check.

What kind To exploit the vulnerability CVE-2023-21069, an attacker would require System execution privileges.

Do users need No, user interaction is not needed for the exploitation of CVE-2023-21069.

What is the CVE-2023-21069 has a severity score of 6.7, categorized as MEDIUM.

When was CVE-2023-21069 was published on 24 March 2023.

Where can More information about the details of CVE-2023-21069 can be found at the following reference URL: <https://source.android.com/security/bulletin/2023-03/01>

What Android CVE-2023-21069 affects versions of the Android kernel. Specific version numbers are not provided in the bulletin.

How can CVE-2023-21069 can lead to a security breach by allowing an attacker with System level privileges to access sensitive data.

Could you An out of bounds write vulnerability, like the one described in CVE-2023-21069, typically involves accessing memory outside the intended boundaries.

What is the The CVE ID for the vulnerability that permits booting with a hidden debug policy is CVE-2023-21068.

What type CVE-2023-21068 describes a vulnerability that allows for local escalation of privilege due to the possibility of booting with a hidden debug policy.

What are the CVE-2023-21068 affects Android kernel versions, but the specific versions were not provided in the details.

What is the CVE-2023-21068 has been rated with a base score of 7.8, which classifies it as HIGH severity.

When was CVE-2023-21068 was published on the 24th of March, 2023.

What is the The Android ID associated with CVE-2023-21068 is A-243433344.

Where can More information about CVE-2023-21068 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03/01>

Is user interaction No, user interaction is not required to exploit the vulnerability mentioned in CVE-2023-21068.

Can you describe A potential attack scenario for CVE-2023-21068 could involve an attacker obtaining physical access to the device.

Is a code example As CVE-2023-21068 pertains to an undisclosed method of booting with a hidden debug policy due to a vulnerability in the kernel.

What is the The CVE ID for the reported vulnerability in the Android kernel is CVE-2023-21067.

Can you provide CVE-2023-21067 refers to a vulnerability in the Android kernel, but the specific details of the vulnerability are not provided.

What is the CVE-2023-21067 has been assigned a CVSS base score of 7.5, which is classified as HIGH severity.

When was The vulnerability CVE-2023-21067 was published on 24 March 2023.

Where can More information about CVE-2023-21067 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03/01>

What versions CVE-2023-21067 affects certain versions of the Android kernel, but the specific versions are not mentioned in the bulletin.

Is there a code example As CVE-2023-21067 refers to a vulnerability in the Android kernel, code examples demonstrating the vulnerability are not provided.

What kind Possible attack scenarios for CVE-2023-21067, given the high severity score, could include gaining unauthorized access to system data.

What is CV CVE-2023-21065 refers to a security vulnerability in the fdt\_next\_tag function of fdt.c within an Android kernel.

What are the To exploit the vulnerability specified in CVE-2023-21065, the attacker would need System execution privileges.

On which platform CVE-2023-21065 occurs on the Android platform, specifically within the Android kernel.

What is the The CVSS base score assigned to CVE-2023-21065 is 6.7, which is categorized as MEDIUM severity.

Which versions CVE-2023-21065 affects unspecified versions of the Android kernel. The exact versions are not provided in the bulletin.

How can one More information about CVE-2023-21065 can be found at the Android Security Bulletin page, specifically at <https://source.android.com/security/bulletin/2023-03/01>



What type CVE-2023-21065 could potentially lead to a local escalation of privilege attack, allowing an attacker w  
What is the The impact of CVE-2023-21065 is significant as it could allow an attacker with the necessary privileges  
When was CVE-2023-21065 was published on 24 March 2023.

What is the The CVE ID of the vulnerability involving an out of bounds read in miscservice.cpp is CVE-2023-21064.  
Can you de CVE-2023-21064 is an out of bounds read vulnerability found in DoSetPinControl of miscservice.cpp o  
What is the The CVSS base score for CVE-2023-21064 is 6.7, which is considered MEDIUM severity. This score indi  
On what d CVE-2023-21064 was published on 24 March 2023.

Are there a No, user interaction is not needed to exploit CVE-2023-21064.

Which And The Android versions affected by CVE-2023-21064 are not explicitly listed in the provided information  
Where can More details about CVE-2023-21064 can be found on the Android Security Bulletin webpage specific t

What type To exploit CVE-2023-21064, System execution privileges are needed. If an attacker successfully exploi

Are there a Yes, there is at least one public reference available for CVE-2023-21064, which is the Android Security

Can you pr A potential attack scenario for CVE-2023-21064 could involve a malicious app that is installed on the A

What is the The CVE ID for the vulnerability related to an out of bounds read in simdata.cpp is CVE-2023-21063.

What are t The exploitation of CVE-2023-21063 could lead to local escalation of privilege, where a malicious app

What privil To exploit the vulnerability CVE-2023-21063, an attacker would need to already have System executio

Is user inte No, user interaction is not required to exploit CVE-2023-21063, making it more concerning. The vulne

On what d CVE-2023-21063 was published on the 24th of March, 2023.

Can you pr For more information about CVE-2023-21063, you can visit the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-03-24>

What versi CVE-2023-21063 affects certain versions of the Android kernel. However, the specific versions impact

Describe a A possible attack scenario for CVE-2023-21063 would involve a malicious app that, once installed on t

Are there a Code examples for the exploitation of CVE-2023-21063 are not provided as part of the CVE description

What is CV CVE-2023-21062 refers to a security issue found in the DoSetTempEcc function within imsservice.cpp.

How sever CVE-2023-21062 carries a Base Score of 6.7, which is categorized as MEDIUM severity according to the

On what d CVE-2023-21062 was published on 24 March 2023.

Which vers CVE-2023-21062 affects unspecified versions of the Android kernel.

Is user inte No, user interaction is not required to exploit CVE-2023-21062.

What privil Exploiting CVE-2023-21062 requires system execution privileges.

Are there a Yes, more information about CVE-2023-21062 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-24>

Can you pr In a potential attack scenario for CVE-2023-21062, a bad actor with system-level access can exploit the

Is there an As of the time the data was provided, there was no specific mention of publicly available exploit code

What is the The CVE ID for the vulnerability found in the Android kernel is CVE-2023-21061.

What are t The affected versions for CVE-2023-21061 are Android kernel versions that fall under the umbrella of

What is the CVE-2023-21061 has been assigned a base score of 7.5, which is categorized as HIGH severity.

When was CVE-2023-21061 was published on 24 March 2023.

Where can More information about CVE-2023-21061 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-24>

Can you pr As per the security standards and ethical guidelines, we do not provide code examples that exemplify

What kind While specifics of CVE-2023-21061 are not detailed, a high-severity vulnerability in the Android kerne

What is the The CVE ID for that vulnerability is CVE-2023-21060.

What is the The impact of CVE-2023-21060 is a possible out of bounds read due to a missing bounds check, which

What is the CVE-2023-21060 has been assigned a severity level of HIGH with a base score of 7.5.

When was CVE-2023-21060 was published on 24 March 2023. More information about the CVE can be found on t

Does CVE- No, CVE-2023-21060 does not require user interaction for exploitation. The vulnerability could be exp

Can you pr An attack scenario for CVE-2023-21060 could involve a malicious actor sending a specially crafted SM

What versi The affected versions of Android for CVE-2023-21060 are not explicitly mentioned in the provided info

What is the The CVE ID of the vulnerability is CVE-2023-21059.

What is the The vulnerability CVE-2023-21059 could lead to remote information disclosure due to a possible out c

What are the details? To exploit the issue described by CVE-2023-21059, an attacker does not need any additional execution privileges.  
Which product? The product affected by CVE-2023-21059 is Android, specifically the Android kernel. The exact version is not specified.  
What is the severity? The base score assigned to CVE-2023-21059 is 7.5, which is categorized as HIGH severity.

When was it published? CVE-2023-21059 was published on 24 March 2023.

Are there any references? Yes, further information on CVE-2023-21059 can be found at the following link: <https://source.android.com/security/bulletin/2023-03-01>

Can you provide a possible attack scenario? A possible attack scenario for CVE-2023-21059 could involve an attacker sending a crafted message or packet to the affected device.

What type of vulnerability is this? CVE-2023-21059 is an out of bounds read vulnerability. To mitigate such risks, developers should ensure proper bounds checking.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-21058.

Can you describe the vulnerability? CVE-2023-21058 describes a vulnerability in `lcsn_SendRrAcquiAssist` of `lcsn_bcm_assist.c`, where the base score of CVE-2023-21058 is 9.8, and it is rated as CRITICAL in terms of severity.

When was it published? CVE-2023-21058 was published on 24 March 2023.

What product is affected? The product affected by CVE-2023-21058 is the Android kernel, specifically referenced within the Android Security Bulletin.

Is user interaction required? No, user interaction is not required for the exploitation of CVE-2023-21058.

Where can I find more information? More information on CVE-2023-21058 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03-01>

Can you provide a specific code example? While a specific code example for CVE-2023-21058 exploitation is not provided, a general concept involves exploiting the lack of bounds check.

What are the potential attack scenarios? Potential attack scenarios for CVE-2023-21058 include an attacker exploiting the lack of bounds check to read memory.

What is the CVE ID for the vulnerability discovered in ProfSixDecomTcpSACKoption of RohcPacketCommon? The CVE ID for the vulnerability discovered in `ProfSixDecomTcpSACKoption` of `RohcPacketCommon` is CVE-2023-21057.

What kind of vulnerability is this? CVE-2023-21057 is classified as a possible out of bounds write due to a missing bounds check, which could lead to a denial of service.

Does CVE-2023-21057 require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21057.

On what platform does the vulnerability exist? CVE-2023-21057 exists on the Android platform, specifically on Android kernel versions. The specific versions affected are not listed.

How critical is the vulnerability? CVE-2023-21057 is rated with a base score of 9.8, which classifies it as CRITICAL in terms of severity.

When was it published? CVE-2023-21057 was published on 24 March 2023.

Where can I find more information? You can find more information about CVE-2023-21057 at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03-01>

What identifier was assigned? The Android Security Team has assigned the identifier A-244450646 to CVE-2023-21057.

Can you provide a potential attack scenario? A potential attack scenario for CVE-2023-21057 could involve a network-based attacker sending specially crafted packets.

What is the CVE ID for the vulnerability discussed? The vulnerability discussed has the CVE ID CVE-2023-21056.

What is the severity? CVE-2023-21056 has a base score of 6.7, which is categorized as MEDIUM severity.

On what date was it published? CVE-2023-21056 was published on 24 March 2023.

Can you describe the vulnerability? CVE-2023-21056 describes a memory corruption vulnerability caused by type confusion in the `lwslc_slc` function.

Are user interaction required? No, user interaction is not needed to exploit CVE-2023-21056. An attacker can take advantage of this vulnerability to corrupt memory.

What versions are affected? CVE-2023-21056 affects the Android kernel, but the specific versions impacted by this vulnerability are not listed.

What kind of privileges are required? An attacker would need System execution privileges to exploit the vulnerability CVE-2023-21056.

Where can I find more information? More information about CVE-2023-21056 can be found in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-03-01>

What are the potential attack scenarios? Potential attack scenarios for CVE-2023-21056 include a local attacker leveraging the type confusion vulnerability to corrupt memory.

Is there a workaround? Unfortunately, without the specific details about the implementation of `lwslc_slc_buffer_free` function, a workaround cannot be provided.

What is the CVE ID for the vulnerability discussed in the dit\_hal\_ioctl function of the dit.c file in the Android kernel? The CVE ID for the vulnerability discussed in `dit_hal_ioctl` function of the `dit.c` file in the Android kernel is CVE-2023-21055.

What is the severity? The CVSS base score for CVE-2023-21055 is assessed as 6.4, which classifies it as a medium severity vulnerability.

What versions are affected? CVE-2023-21055 affects the Android kernel, but the specific versions of Android impacted by this vulnerability are not listed.

When was it published? CVE-2023-21055 was published on 24 March 2023.

Is user interaction required? No, user interaction is not required to exploit CVE-2023-21055.

What kind of privileges are required? An attacker would need System execution privileges in order to exploit CVE-2023-21055.

How can I stay updated? Further information and updates about CVE-2023-21055 can be found on the Android security bulletin page: <https://source.android.com/security/bulletin/2023-03-01>

What is a possible attack scenario? One possible attack scenario for CVE-2023-21055 would involve an attacker who has already obtained System execution privileges.

What is the CVE ID for the vulnerability associated with Android, specifically a component within the EUTRAN\_LCS\_Converter? The CVE ID for the vulnerability associated with Android, specifically a component within the `EUTRAN_LCS_Converter` is CVE-2023-21054.

What type of vulnerability is this? CVE-2023-21054 describes an out of bounds write vulnerability which is caused by a logic error in the `convert` function.

What are the details? To exploit CVE-2023-21054, the attacker would need System execution privileges.

Does CVE-2023-21054 require user interaction? No, CVE-2023-21054 does not require user interaction for its exploitation.

What is the CVE-2023-21054 has been assigned a CVSS Base Score of 7.2, which is categorized as HIGH.

When was CVE-2023-21054 was published on 24 March 2023.

Where can More information about CVE-2023-21054 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-01>

Can you ex An attack scenario for CVE-2023-21054 could involve an attacker crafting a malicious input that triggers the vulnerability.

Is there a c As CVE-2023-21054 is a newly discovered vulnerability, detailed code examples that demonstrate the exploit are not provided.

What is the CVE ID for the vulnerability is CVE-2023-21053.

Can you de CVE-2023-21053 involves a possible out of bounds read due to a missing bounds check in the sms\_Ext function.

What base CVE-2023-21053 has been assigned a base score of 7.5, which is categorized as HIGH severity.

When was CVE-2023-21053 was published on 24 March 2023.

Which vers CVE-2023-21053 affects Android kernel versions up to the one specified before the Android ID A-2518.

Are there a No specific code examples for CVE-2023-21053 are provided in the information I have. Understanding the vulnerability is key.

Where can More information about CVE-2023-21053 can be found on the Android security bulletin webpage at <https://source.android.com/security/bulletin/2023-03-01>

Is user inte No, user interaction is not needed to exploit CVE-2023-21053. The vulnerability can be exploited remotely.

What are s Since CVE-2023-21053 is a vulnerability that can result in remote information disclosure without user interaction.

What is the CVE ID of the vulnerability is CVE-2023-21052.

Can you de CVE-2023-21052 refers to a vulnerability in the 'setToExternal' function of ril\_external\_client.cpp, which affects the Android kernel.

What is the CVE-2023-21052 has been assigned a base score of 6.7, which categorizes it as a MEDIUM severity vulnerability.

When was The CVE-2023-21052 vulnerability was published on 24 March 2023.

Which vers The versions of Android affected by CVE-2023-21052 are Android kernel versions specified in the Android Security Bulletin.

Where can More information about the CVE-2023-21052 vulnerability can be found in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-03-01>

Does CVE-2 No, user interaction is not required for the exploitation of CVE-2023-21052.

What privil To exploit the vulnerability identified by CVE-2023-21052, an attacker would need System execution privileges.

Could you An example attack scenario for CVE-2023-21052 could involve an attacker with access to a system process.

Are there a Without more specific details or access to the vulnerable source code, it is not possible to provide exact code examples.

What is the CVE ID of the vulnerability is CVE-2023-21051.

What is the CVE-2023-21051 is a local escalation of privilege in the kernel, where an attacker can gain system-level access.

What privil To exploit CVE-2023-21051, an attacker would need system execution privileges.

Is user inte No, user interaction is not needed to exploit the CVE-2023-21051 vulnerability.

Which prod The product affected by CVE-2023-21051 is Android, specifically the Android kernel.

What is the CVE-2023-21051 has been assigned a CVSS Base Score of 6.7, which is categorized as Medium severity.

When was The CVE-2023-21051 vulnerability was published on 24 March 2023.

Where can More information about CVE-2023-21051 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-01>

Can you pr Unfortunately, without further details on the exact nature of the out of bounds write, providing a specific attack scenario is not possible.

What might An attack scenario for CVE-2023-21051 could involve an attacker who already possesses system-level access.

What is the CVE ID of the reported vulnerability affecting the Android kernel is CVE-2023-21050.

Can you pr CVE-2023-21050 is described as a vulnerability in the load\_png\_image function of ExynosHWCHelper.

What is the CVE-2023-21050 is rated as 6.7, which falls under the 'MEDIUM' category.

When was CVE-2023-21050 was published on 24 March 2023.

Where can More information about CVE-2023-21050 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-01>

Is user inte No, user interaction is not required to exploit CVE-2023-21050.

What privil To exploit CVE-2023-21050, an attacker would need System execution privileges.

Could you An attack scenario involving CVE-2023-21050 could occur if an attacker with system-level access to an Android device.

What is CV CVE-2023-21049 refers to a security vulnerability in the 'append\_camera\_metadata' function of 'camera2\_hwcomposer.cpp'.

When was CVE-2023-21049 was published on 24 March 2023.

What is the CVE-2023-21049 has been assigned a CVSS Base Score of 4.4, which is categorized as MEDIUM severity.

Does CVE-2 No, CVE-2023-21049 does not require user interaction for exploitation. The vulnerability can be exploited remotely.

Which And CVE-2023-21049 affects the Android kernel. However, specific version numbers of Android have not been specified.

What kind An attacker would need System execution privileges to exploit CVE-2023-21049, which suggests the a

Can you pr Yes, more details on CVE-2023-21049 can be found at the following URL: <https://source.android.com/>

What are s Potential attack scenarios for CVE-2023-21049 include an attacker with System execution privileges le

What is CV CVE-2023-21048 is a security vulnerability identified in the 'handleEvent' function of 'nan.cpp' which i

What type: CVE-2023-21048 affects systems running certain versions of the Android kernel. As Android is used in

What is the severity rating of CVE-2023-21048 is '4.4 MEDIUM'. The severity rating, provided by the Common

On what date CVE-2023-21048 was publicly disclosed on 24 March 2023. Details of the vulnerability along with its a

Is there a p For CVE-2023-21048, users are advised to refer to sources provided by Android, such as the security b

Can you pr A possible attack scenario for CVE-2023-21048 could involve a malicious application that has already c

What is the CVE ID for the vulnerability in ConvertToHalMetadata of aidl\_utils.cc is CVE-2023-21047.

What type CVE-2023-21047 is a security vulnerability that involves a possible out of bounds read due to a missing

What privil To exploit CVE-2023-21047, an attacker would need System execution privileges.

Does CVE-2 No, CVE-2023-21047 does not require user interaction for exploitation.

On which p CVE-2023-21047 impacts users on the Android platform, specifically in the Android kernel.

What is the severity rating of CVE-2023-21047 has been assigned a CVSS Base Score of 4.4, indicating a medium severity level.

When was CVE-2023-21047 was publicly disclosed on 24 March 2023.

Where can More information about CVE-2023-21047 can be found at the official Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03-24#CVE-2023-21047>

What are t Possible attack scenarios for exploiting CVE-2023-21047 include an unauthorized user with System ex

Can you pr As CVE-2023-21047 is a recently disclosed vulnerability, specific code examples that demonstrate the v

What is CV CVE-2023-21046 is a security vulnerability identified in the ConvertToHalMetadata function of aidl\_ut

What is the severity rating of CVE-2023-21046 The Common Vulnerability Scoring System (CVSS) base score for CVE-2023-21046 is 4.4, which is cate

On what date CVE-2023-21046 was published on 24 March 2023.

Which vers The Android versions affected by CVE-2023-21046 are not explicitly listed in the provided information

Are there a Yes, additional information regarding CVE-2023-21046 can be found on the Android Security Bulletin <https://source.android.com/security/bulletin/2023-03-24#CVE-2023-21046>

What kind To exploit CVE-2023-21046, an attacker would need System execution privileges, which are significant

Does CVE-2 No, CVE-2023-21046 does not require user interaction for exploitation. An attacker could exploit this v

Can you de A possible attack scenario for CVE-2023-21046 could involve a malicious app or process that already h

Please pro CVE-2023-21046 refers to a vulnerability in the Android OS, more specifically in the ConvertToHalMet

What is the CVE ID of the vulnerability is CVE-2023-21045.

What is the severity rating of CVE-2023-21045 has been rated with a Base Score of 4.4, which is categorized as MEDIUM severity.

On what date CVE-2023-21045 was published on 24 March 2023.

Which prod The product affected by CVE-2023-21045 is the Android kernel, which is part of the Android operating

What versi CVE-2023-21045 impacts Android kernel versions. However, the specific versions affected are not det

What type CVE-2023-21045 is a vulnerability resulting from a possible out of bounds read due to a use after free

Is user inte No, user interaction is not needed for the exploitation of CVE-2023-21045.

Where can More information about CVE-2023-21045 can be found at the Android Security Bulletin URL: <https://source.android.com/security/bulletin/2023-03-24#CVE-2023-21045>

Can you de A possible attack scenario for CVE-2023-21045 could involve a malicious app or process running on th

What actio To mitigate or fix the vulnerability identified by CVE-2023-21045, users should apply updates provide

What is the CVE ID for this vulnerability is CVE-2023-21044.

Can you de CVE-2023-21044 refers to a vulnerability in the initialization of VendorGraphicBufferMeta, where the

What privil To exploit the vulnerability described by CVE-2023-21044, an attacker would need System execution p

What versi The affected versions of Android related to CVE-2023-21044 are specified as 'Android kernel'. This im

Does CVE-2 No, user interaction is not required for the exploitation of CVE-2023-21044.

What is the severity rating of CVE-2023-21044 has been assigned a Base Score of 4.4, which indicates a severity level of MEDIUM.

When was The vulnerability with CVE ID CVE-2023-21044 was published on 24 March 2023.

Where can Detailed information and possible mitigation for CVE-2023-21044 can be found on the Android Security Bulletin <https://source.android.com/security/bulletin/2023-03-24#CVE-2023-21044>

What coul A potential attack scenario involving CVE-2023-21044 could involve an attacker with System level acc

Is there a c Unfortunately, there is no specific code example provided for the out of bounds read vulnerability ide  
What is CV CVE-2023-21043 is a security vulnerability identified in the Android kernel where there is a potential r  
What Andr CVE-2023-21043 affects unspecified versions of the Android kernel. The exact versions have not been  
What are t To exploit CVE-2023-21043, an attacker would need system execution privileges, indicating that the a  
Is user inte No, user interaction is not required to exploit CVE-2023-21043. The vulnerability can be exploited with  
What is the CVE-2023-21043 has been assigned a base score of 6.7, which classifies it as a MEDIUM severity vulne  
When was CVE-2023-21043 was published on 24 March 2023.

Are there a Yes, a reference is available for CVE-2023-21043: the Android Security Bulletin details for Pixel device:  
What sort i Attack scenarios for CVE-2023-21043 could involve an attacker who already has system execution priv  
Can you pr As CVE-2023-21043 is described as a 'use after free' vulnerability in the Android kernel, a generic exan  
What is the CVE ID of the vulnerability that affects the Android kernel and allows for local escalation of privile  
What is the CVE-2023-21042 has been assigned a Base Score of 6.7, which is categorized as MEDIUM severity.

As of its pu As of its publishing date, the affected component and its version regarding CVE-2023-21042 are listed  
Is user inte No, user interaction is not needed to exploit the vulnerability mentioned in CVE-2023-21042.

Where can Additional information or updates about CVE-2023-21042 can be found at the provided reference URL  
What privil To exploit the vulnerability in CVE-2023-21042, an attacker would need System execution privileges o  
Can you de A possible attack scenario for CVE-2023-21042 would involve an attacker who already has access to the  
What is the The impact of CVE-2023-21042 is significant, as it could allow an attacker to perform a local escalation  
What are t The specific affected versions of Android regarding CVE-2023-21042 were not disclosed in the provide  
What is the CVE ID of the vulnerability is CVE-2023-21041.

What kind CVE-2023-21041 refers to a possible out of bounds write due to an incorrect bounds check, which can  
What are t The potential consequence of the CVE-2023-21041 vulnerability is a local escalation of privilege which  
Does explo No, exploiting the vulnerability described by CVE-2023-21041 does not require user interaction.

How was C To mitigate CVE-2023-21041, Android would have issued a security bulletin and possibly provided a p  
What is the CVE-2023-21041 has been assigned a base score of 7.8, which is classified as HIGH severity.

On which c CVE-2023-21041 was published on the 24th of March, 2023.

Which vers The CVE-2023-21041 vulnerability affects the Android kernel, but the exact versions impacted are not  
What identi The Android platform has assigned the identifier A-250123688 for tracking the CVE-2023-21041 vulne  
Could you A possible attack scenario for CVE-2023-21041 might involve a malicious application without any spec  
What is the CVE ID for this vulnerability is CVE-2023-21040.

What type CVE-2023-21040 could lead to local escalation of privilege.

Does the e No, user interaction is not needed for the exploitation of CVE-2023-21040.

What is the CVE-2023-21040 is 7.8, which is classified as HIGH.

On what d CVE-2023-21040 was published on 24 March 2023.

Which And CVE-2023-21040 affects versions of the Android kernel but does not specify exact version numbers.

What is the CVE-2023-21040 is A-238420277.

Where can More information about CVE-2023-21040 can be found at the following URL: <https://source.android.c>

Are there c There are no specific code examples provided for CVE-2023-21040 within the information given; how

What might A possible attack scenario for CVE-2023-21040 could involve an attacker who has access to the device  
What is the CVE ID for the vulnerability is CVE-2023-21039.

Can you de CVE-2023-21039 involves a possible out of bounds read due to an incorrect bounds check in the dump  
What are t To exploit the vulnerability in CVE-2023-21039, System execution privileges are needed.

Is user inte No, user interaction is not needed to exploit the vulnerability stated in CVE-2023-21039.

What versi Affected versions include the Android kernel specified in the Android Security Bulletin for the vulnera

What is the CVE-2023-21039 has been assigned a CVSS base score of 4.4, which is considered to be of medium sev  
On which c CVE-2023-21039 was published on 24 March 2023.

Where can You can find more information about CVE-2023-21039 at the provided reference URL: [https://source.:](https://source.)

What kind Exploiting CVE-2023-21039 could lead to local information disclosure. An attacker with system execution privileges could exploit this vulnerability to read sensitive data from the system memory.

What is the CVE ID of the vulnerability published on 24 March 2023 is CVE-2023-21038.

Can you describe CVE-2023-21038 is a vulnerability in the `cs40l2x_cp_trigger_queue_show` function of the `cs40l2x.c` file.

What is the severity level of CVE-2023-21038 is rated as MEDIUM, and it has been given a base score of 6.7.

Which versions The Android versions affected by CVE-2023-21038 are part of the Android kernel, but the specific versions are not detailed.

Is there a user interaction required for the exploitation of CVE-2023-21038.

Are there any references available for CVE-2023-21038 which can be found at the following link: <https://source.android.com/security/bulletin/2023-03>

Could you provide an example attack scenario for CVE-2023-21038 could involve a malicious app or an attacker with access to the system.

What kind To exploit CVE-2023-21038, an attacker would need System execution privileges.

What is the CVE ID for the vulnerability in `BitmapExport.java` affecting Android is CVE-2023-21036.

Can you describe The vulnerability identified by CVE-2023-21036 is due to a logic error in `BitmapExport.java`, which results in a denial of service.

What is the severity level of CVE-2023-21036 has been assigned a severity base score of 5.5, which is categorized as MEDIUM.

Which versions CVE-2023-21036 affects the Android kernel, but specific version numbers have not been detailed. However, it is known to affect Android versions 12 and 13.

When was CVE-2023-21036 was publicly disclosed on 24 March 2023.

Where can More information about CVE-2023-21036 can be found on the Android Security Bulletin website at <https://source.android.com/security/bulletin/2023-03>

Are there any known exploits or proof-of-concept code examples for CVE-2023-21036.

What are the possible attack scenarios associated with CVE-2023-21036 could involve an attacker leveraging the logic error to cause a denial of service.

What is the CVE ID of the vulnerability is CVE-2023-21035.

What versions Android version 13 is affected by CVE-2023-21035.

What is the severity level of CVE-2023-21035 has a base score of 7.8, which is categorized as HIGH severity.

When was CVE-2023-21035 was published on 24 March 2023.

What kind Due to CVE-2023-21035, there is a possible way for an app to get permissions previously granted to another app.

Is there a user interaction required to exploit CVE-2023-21035.

Where can More information about CVE-2023-21035 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03>

Can you describe An attack scenario for CVE-2023-21035 could involve a malicious app that is designed to use the same permissions as a legitimate app.

What products The product primarily affected by CVE-2023-21035 is Android, specifically Android 13.

What is the CVE ID CVE-2023-21034 is a security vulnerability found in the `SensorService.cpp` of the Android operating system.

How severe The CVE-2023-21034 vulnerability has been assigned a Base Score of 7.8, which is categorized as HIGH severity.

What versions CVE-2023-21034 affects Android version 13. Users running this version of the Android operating system are affected.

Is there a user interaction required to exploit CVE-2023-21034. This characteristic makes the vulnerability more severe.

What privileges In order to exploit CVE-2023-21034, an attacker would need User execution privileges. This means that the attacker needs to have access to the system.

Where can Additional information about CVE-2023-21034 can be found on the Android Security Bulletin webpage: <https://source.android.com/security/bulletin/2023-03>

What is the impact of CVE-2023-21034 on an individual's Android smartphone could include unauthorized access to sensitive data.

Can you provide An attack scenario for exploiting CVE-2023-21034 could involve a malicious app that targets the `SensorService` to cause a denial of service.

What is the identifier for the reported vulnerability in Android that can lead to a persistent denial of service is CVE-2023-21033.

Which Android versions The vulnerability CVE-2023-21033 affects Android version 13.

What is the severity level of CVE-2023-21033 is 5.5, and its severity is rated as MEDIUM.

When was The CVE-2023-21033 vulnerability was published on 24 March 2023.

Where can More information or the official advisory about CVE-2023-21033 can be found on the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03>

Does CVE-2023-21033 does not require user interaction for exploitation, and it does not require additional privileges.

What file is The file in the Android Operating System that is implicated in CVE-2023-21033 is `WifiManager.java`.

Can you provide Potential attack scenarios for CVE-2023-21033 may include a malicious application exploiting the vulnerability to cause a denial of service.

What are the mitigation steps or updates for CVE-2023-21033 would typically involve applying security patches provided by the manufacturer.

What type CVE-2023-21033 is classified as a type of denial of service (DoS) vulnerability that is caused by resource exhaustion.

What is the CVE ID for that vulnerability is CVE-2023-21032.

Can you describe CVE-2023-21032 involves a possible out of bounds read due to a heap buffer overflow in `_ufdt_output` function.

What privileges To exploit CVE-2023-21032, System execution privileges are needed.

Is user interaction needed for exploitation of CVE-2023-21032?

What is the severity? CVE-2023-21032 has been assigned a base score of 4.4, indicating a MEDIUM severity level.

When was it published? The CVE-2023-21032 vulnerability was published on 24 March 2023.

Which Android version is affected by CVE-2023-21032?

Where can I find more information about CVE-2023-21032? It can be found at the following link: <https://source.android.com/security/bulletin/2023-03-01>.

What kind of vulnerability is CVE-2023-21032 linked to? It is linked to Android ID A-248085351.

Can you provide a possible attack scenario for CVE-2023-21032? A possible attack scenario for CVE-2023-21032 could involve an attacker who already has system-level access.

What is the CVE ID of the vulnerability discovered in the HWC2.cpp component of Android? The CVE ID is CVE-2023-21031.

Can you describe the vulnerability? CVE-2023-21031 is a vulnerability involving a possible out of bounds read due to a race condition in the HWC2.cpp component.

What version of Android is affected by CVE-2023-21031? CVE-2023-21031 affects Android-13, the version of Android that was available when the vulnerability was discovered.

Does CVE-2023-21031 require user interaction for exploitation? No, CVE-2023-21031 can be exploited without any user interaction.

What is the severity of CVE-2023-21031? CVE-2023-21031 has been assigned a severity base score of 4.7, categorizing it as MEDIUM.

When was CVE-2023-21031 publicly disclosed? CVE-2023-21031 was publicly disclosed on 24 March 2023.

Where can I find more information about CVE-2023-21031? You can find more information about CVE-2023-21031 on the Android Security Bulletin webpage at <https://source.android.com/security/bulletin/2023-03-01>.

Can you provide a possible attack scenario for CVE-2023-21031? An attacker could exploit CVE-2023-21031 by triggering a race condition in the setPowerMode function.

Is there an associated Android ID for CVE-2023-21031? Yes, the Android ID associated with CVE-2023-21031 is A-242688355.

What is the CVE ID of the security vulnerability identified in the Confirmation of keystore\_cli\_v2.cpp? The CVE ID is CVE-2023-21030.

How severe is CVE-2023-21030? CVE-2023-21030 is rated with a base score of 7.8, categorizing it as HIGH severity. This rating indicates a high level of severity.

Does CVE-2023-21030 require user interaction for exploitation? No, CVE-2023-21030 does not require user interaction for exploitation, making it more critical because it can be exploited without user interaction.

On what date was CVE-2023-21030 published? CVE-2023-21030 was published on 24 March 2023.

Which Android version is affected by CVE-2023-21030? CVE-2023-21030 affects Android version 13.

Where can I find more information about CVE-2023-21030? More information about CVE-2023-21030 can be found at the Android Security Bulletin URL: <https://source.android.com/security/bulletin/2023-03-01>.

What type of vulnerability is CVE-2023-21030? CVE-2023-21030 is a type of vulnerability that results from a double free error, where a block of memory is freed twice.

What are the steps to remediate CVE-2023-21030? To remediate CVE-2023-21030, one should apply security updates and patches provided by the Android Open Source Project.

Are there any example code changes to address CVE-2023-21030? An example code change to address a double free issue might involve adding proper checks before freeing memory.

What is the CVE ID for the security vulnerability in UidObserverController.java of Android 13? The CVE ID is CVE-2023-21029.

Can you describe the vulnerability? The vulnerability associated with CVE-2023-21029 is due to a missing permission check in the registerUidObserver function.

What type of vulnerability is CVE-2023-21029? To exploit the vulnerability described in CVE-2023-21029, User execution privileges are needed. This is a privilege escalation vulnerability.

Is user interaction needed for exploitation of CVE-2023-21029? No, user interaction is not needed to exploit the vulnerability CVE-2023-21029. An attacker could potentially exploit this vulnerability without user interaction.

What is the severity or CVSS (Common Vulnerability Scoring System) base score assigned to CVE-2023-21029? The severity or CVSS (Common Vulnerability Scoring System) base score assigned to CVE-2023-21029 is 4.3, indicating a MEDIUM severity level.

When was the CVE-2023-21029 vulnerability published? The CVE-2023-21029 vulnerability was published on 24 March 2023.

Where can I find more details regarding CVE-2023-21029? More details regarding CVE-2023-21029 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-01>.

Could you provide a hypothetical attack scenario exploiting CVE-2023-21029? A hypothetical attack scenario exploiting CVE-2023-21029 could involve an attacker who has gained user-level access to the device.

What versions of Android are affected by CVE-2023-21029? The versions of Android affected by CVE-2023-21029 are Android 13.

What is the CVE ID of the vulnerability involving an out of bounds read in ipphelper.c? The CVE ID is CVE-2023-21028.

What component is affected by CVE-2023-21028? The component affected by CVE-2023-21028 is the parse\_printerAttributes function in ipphelper.c.

What is the potential impact of CVE-2023-21028? The potential impact of CVE-2023-21028 is a possible out of bounds read due to a string lacking a null terminator.

Does CVE-2023-21028 require user interaction for exploitation? No, user interaction is not required for the exploitation of CVE-2023-21028.

What are the versions of Android affected by CVE-2023-21028? The versions of Android affected by CVE-2023-21028 include Android-13.

What is the CVSS base score assigned to CVE-2023-21028? The CVSS base score assigned to CVE-2023-21028 is 7.5, which is considered HIGH.

When was CVE-2023-21028 published? CVE-2023-21028 was published on 24 March 2023.

Where can I find more information about CVE-2023-21028? More information about CVE-2023-21028 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03-01>.

Could you provide a possible attack scenario for CVE-2023-21028? A possible attack scenario for CVE-2023-21028 might involve an attacker sending a crafted IPP (Internet Printing Protocol) packet.

What mitigation steps can be taken to address CVE-2023-21028? To mitigate the risk posed by CVE-2023-21028, users should apply any patches or updates provided by the Android Open Source Project.

What is the CVE ID for the vulnerability found in the PasspointXmlUtils.java file of Android? The CVE ID is CVE-2023-21027.

What versions of Android are affected by CVE-2023-21027? The versions of Android that are affected by CVE-2023-21027 are those based on Android 13.

What is the potential impact of CVE-2023-21027? The vulnerability described in CVE-2023-21027 could lead to remote information disclosure due to an out of bounds read.

Does the e No, the exploitation of CVE-2023-21027 does not require user interaction.

What is the CVE-2023-21027 has a CVSS base score of 7.5, which is considered HIGH severity.

Where can More information about CVE-2023-21027 can be found at the following URL: [https://source.android.c](https://source.android.com/security/bulletin/2023-03-24/cve-2023-21027)

What privil An attacker does not need any additional execution privileges to exploit CVE-2023-21027.

What is the The logic error in CVE-2023-21027 is related to an authentication misconfiguration within the Passpoi

Can you pr A possible attack scenario for CVE-2023-21027 could involve a remote attacker sending specifically cr

Has CVE-20 The CVE-2023-21027 details have been made public as evidenced by the published date and reference

What is the The assigned number for the vulnerability affecting Android's WindowManagerService.java is CVE-20

Can you ex CVE-2023-21026 refers to a vulnerability in the updateInputChannel of WindowManagerService.java 1

Which And CVE-2023-21026 affects Android version 13.

What is the According to its Base Score, CVE-2023-21026 is rated as a MEDIUM severity vulnerability with a score

When was The CVE-2023-21026 vulnerability was publicly disclosed on 24 March 2023.

Where can More information about CVE-2023-21026 can be found at the Android Security Bulletin page, with the

What type CVE-2023-21026 allows an attacker to potentially cause a local denial of service due to a touchable re

Could you A possible attack scenario for CVE-2023-21026 might involve a malicious app running on the device th

What are t To exploit CVE-2023-21026, an attacker does not need any additional execution privileges. The vulner

What is CV CVE-2023-21025 refers to a security vulnerability in the ufdt\_local\_fixup\_prop function of the ufdt\_ov

What type CVE-2023-21025 is classified as an out of bounds read vulnerability due to an incorrect bounds check i

What is the The Common Vulnerability Scoring System (CVSS) base score for CVE-2023-21025 is 4.4, which is cons

Does CVE- No, CVE-2023-21025 does not require user interaction for exploitation. An attacker with System exec

What Andr CVE-2023-21025 affects Android version Android-13. It is advisable for users and administrators to rel

Where can Information regarding the fix or mitigation procedures for CVE-2023-21025 can be found at the officia

Can you pr A possible attack scenario for CVE-2023-21025 could involve a malicious application that has been gra

What is CV CVE-2023-21024 is a security vulnerability identified in Android-13 that is described as a possible dela

What is the The CVSS base score of CVE-2023-21024 is 7.8, which is classified as HIGH severity.

When was CVE-2023-21024 was published on 24 March 2023.

What versi CVE-2023-21024 affects Android version 13.

Are there a There are no specific code examples provided for CVE-2023-21024 in the given information. Detailed

Is user inte No, user interaction is not needed to exploit CVE-2023-21024.

Where can More information about CVE-2023-21024 can be found at the Android Security Bulletin page: [https://](https://source.android.com/security/bulletin/2023-03-24/cve-2023-21024)

What are s Possible attack scenarios for CVE-2023-21024 include an attacker exploiting the logic error in Fallback

What is the The CVE ID assigned is CVE-2023-21022.

What is the CVE-2023-21022 has been assigned a base score of 7.8, which is categorized as HIGH.

What type CVE-2023-21022 describes a possible out of bounds write due to memory corruption in the BufferBloc

On which CVE-2023-21022 occurs on the Android platform, specifically affecting Android version 13.

Is user inte No, user interaction is not needed to exploit CVE-2023-21022.

Where can More information about CVE-2023-21022 can be found at the following URL: [https://source.android.c](https://source.android.com/security/bulletin/2023-03-24/cve-2023-21022)

When was CVE-2023-21022 was published on 24 March 2023.

What Andr The Android ID associated with CVE-2023-21022 is A-236098131.

Can you de A possible attack scenario for CVE-2023-21022 involves a malicious application exploiting the out of b

Could you A direct code example representing this specific vulnerability CVE-2023-21022 cannot be provided wit

What is the The CVE ID of the vulnerability is CVE-2023-21021.

Can you de CVE-2023-21021 describes a vulnerability in WifiServiceImpl.java where a guest user could potentially

What is the CVE-2023-21021 has been assigned a CVSS base score of 7.8, which is categorized as HIGH severity.

When was CVE-2023-21021 was published on 24 March 2023.

Which And CVE-2023-21021 affects Android version 13.

Is there a p Information about patches or updates for CVE-2023-21021 can typically be found in the Android secur



What might A possible attack scenario for CVE-2023-21021 would involve a malicious guest user on an Android device.  
What is CVE-2023-21020 is a security vulnerability identified in a certain version of the Android operating system.  
Which version Android version 13 is affected by the security vulnerability CVE-2023-21020.

How severe The CVE-2023-21020 vulnerability has been assigned a base score of 6.7, which indicates a medium level.  
Is user interaction No, user interaction is not required to exploit the CVE-2023-21020 vulnerability. An attacker can exploit it without user interaction.  
When was CVE-2023-21020 was published on 24 March 2023.

Where can Additional information about CVE-2023-21020 can be found by visiting the security bulletin page at the Android Security Bulletin.  
What are the To exploit the vulnerability CVE-2023-21020, an attacker would need to have System execution privileges.  
Can you describe An attack scenario for exploiting CVE-2023-21020 might involve an attacker who has already obtained System execution privileges.

Are there any Specific code examples demonstrating the CVE-2023-21020 vulnerability are typically not publicly disclosed.  
What is CVE-2023-21019 refers to a security vulnerability discovered in Android 13, specifically within the 'lib2' component.  
How severe CVE-2023-21019 has been assigned a base score of 5.5, which is categorized as MEDIUM severity. This is because it requires user interaction.  
What version CVE-2023-21019 affects Android version 13. Users running this version on their devices are potentially affected.  
Is user interaction No, user interaction is not required to exploit CVE-2023-21019. The vulnerability could be exploited without user interaction.  
What could By exploiting CVE-2023-21019, an attacker could potentially gain access to sensitive information due to the lack of proper permission checks.  
Where can More information about CVE-2023-21019 can be found through security bulletins such as the Android Security Bulletin.  
What are the Possible attack scenarios for CVE-2023-21019 include a malicious app or script running on an Android device.

What is the CVE ID for the vulnerability is CVE-2023-21018.  
In which component The vulnerability CVE-2023-21018 is found in the UnwindingWorker component of the unwind.cc file.  
What type CVE-2023-21018 is a vulnerability that involves a possible out of bounds write due to a use after free.  
Is user interaction No, user interaction is not needed to exploit the CVE-2023-21018 vulnerability.  
Which version CVE-2023-21018 affects Android version Android-13.

What is the severity The severity base score assigned to CVE-2023-21018 is 6.7, which is categorized as MEDIUM.  
When was CVE-2023-21018 was publicly disclosed on 24 March 2023.  
Can you provide More information on CVE-2023-21018 can be found at the following link: <https://source.android.com>,  
Are there any As CVE-2023-21018 is a security vulnerability, specific code examples illustrating the exploit are not typically provided.  
What could A possible attack scenario for CVE-2023-21018 would involve an attacker with local access to the Android device.  
What is the CVE ID of the vulnerability is CVE-2023-21017.

In which file The vulnerability CVE-2023-21017 is found in the InstallStart.java file of the Android operating system.  
What kind Due to the CVE-2023-21017 vulnerability, local escalation of privilege is possible.  
What are the To exploit the CVE-2023-21017 vulnerability, a potential attacker would need User execution privileges.  
Is user interaction No, user interaction is not needed for exploitation of the CVE-2023-21017 vulnerability.

Which version The Android version affected by the CVE-2023-21017 vulnerability is Android-13.

What is the base score The base score assigned to CVE-2023-21017 is 7.8, which is categorized as HIGH.

When was CVE-2023-21017 was published on 24 March 2023.

Where can More information about CVE-2023-21017 can be found at the Android Security Bulletin, which can be found at the Android Security Bulletin.  
Can you describe A possible attack scenario for CVE-2023-21017 could involve a malicious application that, by exploiting the vulnerability, could escalate its privileges.  
What is the CVE ID of the vulnerability related to AccountTypePreference in Android is CVE-2023-21016.

Can you describe CVE-2023-21016 refers to a vulnerability in the AccountTypePreference component of AccountTypePreference.  
What is the CVE-2023-21016 has a CVSS base score of 5.5, which is classified as MEDIUM severity.  
Which Android version The Android version affected by CVE-2023-21016 is Android-13.

When was CVE-2023-21016 was published on 24 March 2023.

Are there any Yes, more details about CVE-2023-21016 can be found at the following URL: <https://source.android.com>.  
Is user interaction No, user interaction is not needed to exploit the vulnerability identified by CVE-2023-21016.

What kind Exploiting CVE-2023-21016 can lead to a local denial of service attack. No additional execution privileges are required.  
What are the Attack scenarios for CVE-2023-21016 could involve an attacker crafting malicious inputs that exploit the vulnerability.  
What is the CVE ID for the vulnerability related to a missing permission check in the Transcode Permission Component is CVE-2023-21015.

Can you de CVE-2023-21015 refers to a security vulnerability found in several Transcode Permission Controllers o  
What is the CVE-2023-21015 has a CVSS base score of 7.8, which is classified as HIGH. This score indicates that the  
When was CVE-2023-21015 was published on the 24th of March, 2023.

Where can More information about CVE-2023-21015 can be found at the provided reference URL: <https://source>

What are t The product affected by CVE-2023-21015 is Android, specifically versions Android-13.

What are s A potential attack scenario for CVE-2023-21015 could involve a malicious application installed on the  
Does CVE-2 No, CVE-2023-21015 does not require user interaction for exploitation. An attacker could potentially e

What is the CVE ID for the out of bounds read vulnerability in Android's p2p\_iface.cpp is CVE-2023-21014.

What are t The vulnerability referenced by CVE-2023-21014 could lead to local information disclosure, potentially

Does explo No, user interaction is not required for the exploitation of the vulnerability associated with CVE-2023-  
What Andr The Android version affected by CVE-2023-21014 is Android-13.

What is the CVE-2023-21014 has been assigned a CVSS Base Score of 4.4, which is categorized as MEDIUM severity  
When was CVE-2023-21014 was published on 24 March 2023.

Where can More details about CVE-2023-21014 can be found at the Android security bulletin webpage, specifical

What privil To exploit the vulnerability identified as CVE-2023-21014, an attacker would need System execution p

Can you pr A potential attack scenario for CVE-2023-21014 would involve an attacker who has gained System exe

Is there an As CVE-2023-21014 is a security vulnerability, specific code examples showcasing how to exploit it wo

What is CV CVE-2023-21013 is a security vulnerability identified in the forceStaDisconnection function of the host

What com The component affected by CVE-2023-21013 is the forceStaDisconnection function within the hostapc

What versi Android version 13 is impacted by CVE-2023-21013 as mentioned in the vulnerability details.

What are t To exploit CVE-2023-21013, an attacker would need System execution privileges on the affected devic

What is the CVE-2023-21013 is 4.4, which is categorized as Medium severity.

When was CVE-2023-21013 was published on 24 March 2023.

Where can More information about CVE-2023-21013 can be found at the following URL: <https://source.android.c>

What kind If successfully exploited, CVE-2023-21013 could lead to local information disclosure, potentially giving

What is the CVE ID associated with CVE-2023-21013 is A-256818945.

What could A possible attack scenario involving CVE-2023-21013 could involve a malicious application or a rogue

What is the CVE ID for the vulnerability is CVE-2023-21012.

What is the CVE-2023-21012 is a possible out of bounds read due to a missing bounds check in r

What type An attacker would need System execution privileges to exploit the vulnerability described in CVE-2023-

Do users n No, user interaction is not needed for the exploitation of CVE-2023-21012.

Which prod The product affected by CVE-2023-21012 is Android.

Which And The Android version mentioned as vulnerable in CVE-2023-21012 is Android-13.

What is the CVE-2023-21012 is 4.4 and the severity rating is MEDIUM.

When was CVE-2023-21012 was published on 24 March 2023.

Where can You can find more details about CVE-2023-21012 at the following URL: <https://source.android.com/se>

Can you su Possible attack scenarios for CVE-2023-21012 include a malicious application or process running with

What is the CVE ID for the vulnerability involving an out of bounds read in p2p\_iface.cpp is CVE-2023-21011.

Can you de CVE-2023-21011 is a vulnerability in multiple locations of p2p\_iface.cpp in Android whereby there is a

What are t To exploit the CVE-2023-21011 vulnerability, an attacker would need System execution privileges. Thi  
Is user inte No, user interaction is not required to exploit the CVE-2023-21011 vulnerability.

What is the CVE-2023-21011 is 4.4, classifying it as a MEDIUM severity vulnerabi

When was CVE-2023-21011 was published on 24 March 2023.

Which vers The versions of Android affected by CVE-2023-21011 include Android-13.

Where can More information about CVE-2023-21011 can be found on the Android Security Bulletin webpage at h

What kind The CVE-2023-21011 vulnerability could be exploited to perform local information disclosure attacks.

What is the CVE ID of the vulnerability being discussed is CVE-2023-21010.

What is the base score of CVE-2023-21010 is 4.4, which is categorized as MEDIUM severity.

In which file is the vulnerability CVE-2023-21010 found in the file 'p2p\_iface.cpp'.

What is the CVE-2023-21010 involves a possible out of bounds read due to a missing bounds check, which could lead to a denial of service.

What privileges System execution privileges are needed to exploit the vulnerability CVE-2023-21010.

Does exploitation No, user interaction is not needed for the exploitation of CVE-2023-21010.

Which product The product affected by CVE-2023-21010 is Android, specifically Android version 13.

What is the Android ID associated with CVE-2023-21010 is A-257029915.

When was CVE-2023-21010 published on 24 March 2023.

Where can Further information on CVE-2023-21010 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03>

What could A possible attack scenario for CVE-2023-21010 could involve a malicious app that is able to exploit the vulnerability to perform a denial of service.

Are there any No specific code examples detailing the CVE-2023-21010 vulnerability have been provided. As it pertains to the p2p\_iface.cpp file.

What is CVE-2023-21009 refers to a security vulnerability identified in multiple locations of the p2p\_iface.cpp file.

What is the impact score of CVE-2023-21009 is rated as 4.4 which falls under the 'MEDIUM' severity category.

When was CVE-2023-21009 published on 24 March 2023.

What versions The versions of Android affected by CVE-2023-21009 include Android-13.

What privileges To exploit the vulnerability described in CVE-2023-21009, an attacker would need System execution privileges.

Is user interaction No, user interaction is not required to exploit CVE-2023-21009.

Are there any Yes, more details on CVE-2023-21009 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03>

What are the Potential attack scenarios for CVE-2023-21009 involve an attacker with System execution privileges reading sensitive information from the p2p\_iface.cpp file.

What is the CVE ID of the vulnerability is CVE-2023-21008.

What type CVE-2023-21008 addresses a possible out of bounds read due to a missing bounds check in p2p\_iface.cpp.

Which product The product affected by CVE-2023-21008 is Android, particularly version Android-13.

What is the severity CVE-2023-21008 has a severity rating of 4.4, which is classified as MEDIUM. This suggests that it represents a moderate risk.

When was The CVE-2023-21008 vulnerability was published on 24 March 2023.

What privileges An attacker would need System execution privileges to exploit CVE-2023-21008.

Do users need No, user interaction is not needed for the exploitation of CVE-2023-21008.

What type Exploiting CVE-2023-21008 could lead to local information disclosure.

Where can More information about CVE-2023-21008 can be found at the official Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-03>

Can you provide An example of an attack scenario for CVE-2023-21008 could involve a malicious application that is installed on the device and attempts to read sensitive information from the p2p\_iface.cpp file.

What is CVE-2023-21007 refers to a security vulnerability found in the p2p\_iface.cpp of Android, specifically in the p2p\_iface.cpp file.

What component CVE-2023-21007 affects the p2p\_iface.cpp component in Android-13. It may affect multiple locations within the p2p\_iface.cpp file.

How severe The severity of CVE-2023-21007 is rated as medium, with a base score of 4.4. While it could lead to information disclosure, it is not considered critical.

When was CVE-2023-21007 published on 24 March 2023.

What are the To exploit CVE-2023-21007, an attacker would need to have System execution privileges on the affected device.

Does CVE-2023-21007 require No, CVE-2023-21007 does not require user interaction for exploitation. This means that the vulnerability can be exploited without the user's knowledge or action.

Is there any Yes, official documentation and reference for addressing CVE-2023-21007 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03>

Can you give A possible attack scenario for CVE-2023-21007 could involve a malicious app already installed on the device attempting to read sensitive information from the p2p\_iface.cpp file.

What is the Android ID associated with CVE-2023-21007 is A-257029965. This identifier can be used to reference the vulnerability in official documentation.

Are there any As a knowledge-based AI, I don't have direct access to the source code of proprietary software like Android. However, I can provide information based on publicly available data.

What is CVE-2023-21006 refers to a security vulnerability found in multiple locations of the p2p\_iface.cpp component in Android-13.

What is the CVSS base score of CVE-2023-21006 is 4.4, which classifies it as a MEDIUM severity level vulnerability.

On what date CVE-2023-21006 was published on 24 March 2023.

What product The product affected by CVE-2023-21006 is Android, specifically version 13.

What are the Exploiting CVE-2023-21006 could lead to local information disclosure, which means that sensitive information stored on the device could be accessed by an attacker.

Is user interaction No, user interaction is not required to exploit CVE-2023-21006.

Can you provide Since CVE-2023-21006 is an out of bounds read vulnerability, one possible attack scenario involves a malicious app attempting to read sensitive information from the p2p\_iface.cpp file.

Where can You can find more information and advisories about CVE-2023-21006 at the following reference URL: <https://source.android.com/security/bulletin/2023-03>

What is the CVE ID of the vulnerability related to the potential permission bypass in several Transcode Permissions? The CVE ID of the vulnerability related to the potential permission bypass in several Transcode Permissions is CVE-2023-21005.

Can you describe CVE-2023-21005? CVE-2023-21005 refers to a security vulnerability in `getAvailabilityStatus` of several Transcode Permissions.

What is the CVSS base score for CVE-2023-21005? The CVSS base score for CVE-2023-21005 is 7.8, which is rated as HIGH severity. This indicates that the vulnerability has a significant impact on the system.

When was CVE-2023-21005 published? CVE-2023-21005 was published on 24 March 2023.

What version of Android is affected by CVE-2023-21005? CVE-2023-21005 affects Android 13. Users of affected devices should update their systems to the latest version of Android 13.

Where can I find more information about CVE-2023-21005? More information about CVE-2023-21005 can be found in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-03>.

Is user interaction required to exploit CVE-2023-21005? No, user interaction is not required for an attacker to exploit the vulnerability described in CVE-2023-21005.

What type of vulnerability is CVE-2023-21005? CVE-2023-21005 is a permission bypass vulnerability that can lead to local escalation of privilege. If successful, an attacker could gain access to sensitive data and perform actions that are not intended for the application.

Can you provide an example attack scenario for CVE-2023-21005? In an attack scenario for CVE-2023-21005, a malicious application installed on the Android device could attempt to bypass the permission check in `getAvailabilityStatus` to access sensitive data.

What is CVE-2023-21004? CVE-2023-21004 refers to a security vulnerability in `getAvailabilityStatus` of several Transcode Permissions.

Which Android version is affected by CVE-2023-21004? CVE-2023-21004 affects Android version 13.

What is the impact of CVE-2023-21004? Exploiting CVE-2023-21004 could lead to a local escalation of privilege. This means that an attacker could gain access to sensitive data and perform actions that are not intended for the application.

Is user interaction required to exploit CVE-2023-21004? No, user interaction is not needed to exploit CVE-2023-21004.

What is the CVSS base score for CVE-2023-21004? The CVSS base score assigned to CVE-2023-21004 is 7.8, categorized as HIGH severity.

Where can I find more information about CVE-2023-21004? More information about CVE-2023-21004 can be found in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-03>.

What kind of vulnerability is CVE-2023-21004? An attacker does not need any additional execution privileges to exploit CVE-2023-21004. The vulnerability allows local escalation of privilege.

What is the risk associated with CVE-2023-21004? The risk associated with CVE-2023-21004 is high, as it allows attackers to escalate their privileges locally.

Describe a potential attack scenario for CVE-2023-21004. In a potential attack scenario for CVE-2023-21004, an attacker could create a malicious application that attempts to bypass the permission check in `getAvailabilityStatus` to access sensitive data.

Has CVE-2023-21004 been officially recognized and documented? Yes, CVE-2023-21004 has been officially recognized and documented in the Android Security Bulletin.

What is CVE-2023-21003? CVE-2023-21003 refers to a security vulnerability found in several Transcode Permission Controllers.

What version of Android is affected by CVE-2023-21003? The vulnerability CVE-2023-21003 affects the Android-13 version of the Android operating system.

How severe is CVE-2023-21003? CVE-2023-21003 has been classified with a base score of 7.8, which is considered HIGH severity.

What type of vulnerability is CVE-2023-21003? CVE-2023-21003 allows local escalation of privilege, meaning an attacker with access to the device could gain access to sensitive data and perform actions that are not intended for the application.

Do users need to interact with the application to exploit CVE-2023-21003? No, user interaction is not needed for exploitation of CVE-2023-21003. An attacker can exploit this vulnerability without user interaction.

When was CVE-2023-21003 published? CVE-2023-21003 was published on 24 March 2023.

Where can I find more information about CVE-2023-21003? More information about CVE-2023-21003 can be found on the Android Security Bulletin page at the following link: <https://source.android.com/security/bulletin/2023-03>.

What is a possible attack scenario for CVE-2023-21003? An attack scenario for CVE-2023-21003 could involve a malicious application installed on an Android device attempting to bypass the permission check in `getAvailabilityStatus` to access sensitive data.

What is the CVE ID of the reported vulnerability? The CVE ID of the reported vulnerability is CVE-2023-21002.

Can you describe CVE-2023-21002? CVE-2023-21002 involves a possible permission bypass due to a missing permission check in `getAvailabilityStatus`.

What version of Android is affected by CVE-2023-21002? Android version 13 is affected by CVE-2023-21002.

How severe is CVE-2023-21002? CVE-2023-21002 has been assigned a Base Score of 7.8, which is categorized as HIGH severity.

What is the publication date of CVE-2023-21002? CVE-2023-21002 was published on 24 March 2023.

Where can I find more information about CVE-2023-21002? More information about CVE-2023-21002 can be found on the Android Security Bulletin page at: <https://source.android.com/security/bulletin/2023-03>.

Does the vulnerability require user interaction? No, user interaction is not needed for the exploitation of CVE-2023-21002.

Could you provide an example attack scenario for CVE-2023-21002? An attacker with local access to an Android 13 device could exploit CVE-2023-21002 by utilizing a flaw in the permission check in `getAvailabilityStatus`.

What is the Android ID associated with CVE-2023-21002? The Android ID associated with CVE-2023-21002 is A-261193935.

What is CVE-2023-21001? CVE-2023-21001 refers to a security vulnerability in `NetworkProviderSettings.java` of the Android operating system.

How severe is CVE-2023-21001? CVE-2023-21001 has been assigned a base score of 7.8, which is categorized as HIGH severity. This indicates that the vulnerability has a significant impact on the system.

Was there a missing permission check? Yes, CVE-2023-21001 involves a missing permission check in `onContextItemSelected` of `NetworkProviderSettings.java`.

Is user interaction required to exploit CVE-2023-21001? No, user interaction is not needed to exploit CVE-2023-21001, making the vulnerability more critical as it can be exploited without user interaction.

Can you provide a possible attack scenario for CVE-2023-21001? A possible attack scenario for exploiting CVE-2023-21001 could involve a malicious application installed on the device attempting to bypass the permission check in `onContextItemSelected` to access sensitive data.

Where can I find additional details about the CVE-2023-21001 vulnerability? Additional details about the CVE-2023-21001 vulnerability can be found in the official Android Security Bulletin.

What is the CVE ID for the vulnerability related to MediaCodec.cpp in Android? The CVE ID for the vulnerability related to `MediaCodec.cpp` in Android is CVE-2023-21000.

What are the details of the vulnerability identified by CVE-2023-21000? The vulnerability identified by CVE-2023-21000 could lead to local escalation of privilege due to a possible missing permission check.

Which versions of Android are affected by CVE-2023-21000? CVE-2023-21000 affects versions of Android-13.

What is the Base Score rating for CVE-2023-21000? The Base Score rating for CVE-2023-21000 is 7.8, which is considered HIGH.

Is user interaction required to exploit CVE-2023-21000? No, user interaction is not needed for exploitation of the vulnerability mentioned in CVE-2023-21000.

On what date CVE-2023-21000 was published on 24 March 2023.

Where can More information about CVE-2023-21000 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-01>

Could you In a possible attack scenario for CVE-2023-21000, an attacker could run a malicious app that exploits the vulnerability.

What's the The Android ID associated with CVE-2023-21000 is A-194783918.

Can you provide I cannot provide a specific code example since CVE descriptions typically do not include source code. (

What is the The CVE ID associated with the vulnerability that can trigger a persistent reboot loop on Android devices is CVE-2023-21000.

Can you describe CVE-2023-20999 describes a vulnerability in Android where improper input validation at multiple locations can lead to a denial of service.

What is the CVE-2023-20999 has a CVSS base score of 5.5, which is considered a MEDIUM severity issue.

On what date The vulnerability with CVE ID CVE-2023-20999 was published on 24 March 2023.

Where can More information or the official advisory about CVE-2023-20999 can be found on the Android Security Bulletin page at: <https://source.android.com/security/bulletin/2023-03-01>

Which Android version The Android version affected by CVE-2023-20999 is Android-13.

What privileges To exploit the vulnerability outlined in CVE-2023-20999, an attacker would need User execution privileges.

Is user interaction No, user interaction is not needed for the exploitation of CVE-2023-20999.

What is the The Android ID associated with CVE-2023-20999 is A-246750467.

Could you Possible attack scenarios for CVE-2023-20999 include a malicious app exploiting the vulnerability with the help of a specially crafted APK file.

What is CV CVE-2023-20998 is a security vulnerability identified in certain versions of the Android operating system.

Which version The version of Android that is affected by CVE-2023-20998 is Android 13.

What privileges An attacker would need to have User execution privileges in order to exploit CVE-2023-20998.

What is the CVE-2023-20998 has been assigned a Base Score of 5.5, indicating a MEDIUM level of severity.

When was CVE-2023-20998 was published on March 24, 2023.

Where can Official information about CVE-2023-20998 can be found at the following link: <https://source.android.com/security/bulletin/2023-03-01>

Are code examples Code examples for CVE-2023-20998 are typically not provided in the CVE details. Instead, the reference is made to the official advisory.

What are the Potential attack scenarios for exploiting CVE-2023-20998 include a malicious app gaining User execution privileges.

What is CV CVE-2023-20997 is a security vulnerability identified in the Android operating system, specifically in Android 13.

How severe The severity of CVE-2023-20997 is rated as a 'MEDIUM' with a base score of 5.5. This means that while it is not critical, it still poses a significant risk.

What are the To exploit CVE-2023-20997, the attacker needs to have User execution privileges on the Android device.

Is user interaction No, user interaction is not required to exploit CVE-2023-20997. An attacker with the necessary privileges can trigger the vulnerability.

When was CVE-2023-20997 was published on 24 March 2023.

Where can Additional information about CVE-2023-20997 can be found on the Android Security Bulletin webpage at: <https://source.android.com/security/bulletin/2023-03-01>

Could you A possible attack scenario for exploiting CVE-2023-20997 could involve an attacker who has obtained a specially crafted APK file.

What measures To mitigate CVE-2023-20997, users and administrators are advised to apply any patches or updates promptly.

What is CV CVE-2023-20996 refers to a security vulnerability identified in Android version 13, where there is a potential for a denial of service.

What is the The severity score of CVE-2023-20996 is rated as 5.5, which is categorized as MEDIUM on the Common Vulnerability Scoring System (CVSS).

When was CVE-2023-20996 was published on the 24th of March, 2023.

Which product The product affected by CVE-2023-20996 is Android, specifically version 13.

Where can More information about CVE-2023-20996 can be found at the Android Security Bulletin page at: <https://source.android.com/security/bulletin/2023-03-01>

Are user interaction To exploit CVE-2023-20996, User execution privileges are needed, but user interaction is not necessarily required.

What is the The Android ID associated with CVE-2023-20996 is A-246749764.

Can you provide A potential attack scenario for CVE-2023-20996 could involve a malicious application that exploits the vulnerability to cause a denial of service.

What type CVE-2023-20996 is classified as a local denial of service (DoS) vulnerability caused by improper input validation.

What is CV CVE-2023-20995 is a security vulnerability found in the CustomizedSensor.cpp file of the Android operating system.

How severe The vulnerability defined by CVE-2023-20995 has been assigned a Base Score of 7.8, which is classified as HIGH severity.

Does exploitation No, exploiting CVE-2023-20995 does not require user interaction, which means it can be exploited by an attacker with the necessary privileges.

Which version CVE-2023-20995 affects the Android-13 version of the Android operating system.

When was CVE-2023-20995 was publicly disclosed on 24 March 2023.

What references Further details on CVE-2023-20995 can be found on the Android Security Bulletin page at the following link: <https://source.android.com/security/bulletin/2023-03-01>

What kind CVE-2023-20995 could allow a local escalation of privilege, giving an attacker the ability to execute arbitrary code with system-level permissions.

How might an attacker exploit CVE-2023-20995 by taking advantage of the logic error in the captureImage method? No, a specific code example for exploiting CVE-2023-20995 has not been provided here. Analyzing the vulnerability details, the issue is a logic error in the captureImage method where the buffer size is not properly validated, leading to a potential out-of-bounds write.

What Android ID is associated with CVE-2023-20995? The Android ID associated with CVE-2023-20995 is A-241910279.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-20994.

Can you describe the vulnerability? CVE-2023-20994 refers to a vulnerability where there is a possible out of bounds write due to an incorrect bounds check in the write method.

What are the requirements for exploiting CVE-2023-20994? To exploit CVE-2023-20994, an attacker would need System execution privileges.

Is there any user interaction required for exploiting CVE-2023-20994? No, user interaction is not required for exploiting CVE-2023-20994.

Which product is affected by CVE-2023-20994? The product affected by CVE-2023-20994 is Android, and the affected version is Android-13.

What is the severity rating of CVE-2023-20994? CVE-2023-20994 has a severity rating of 6.7, which is categorized as MEDIUM. This means the vulnerability is a moderate risk.

Where can more information about CVE-2023-20994 be found? More information about CVE-2023-20994 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03>.

When was CVE-2023-20994 published? CVE-2023-20994 was published on 24 March 2023.

Could you provide a possible attack scenario for CVE-2023-20994? One possible attack scenario for CVE-2023-20994 could involve a malicious application gaining System execution privileges and exploiting the out of bounds write to overwrite sensitive data.

What is the Android ID associated with CVE-2023-20994? The Android ID associated with CVE-2023-20994 is A-259062118.

What is CVE-2023-20993? CVE-2023-20993 is a security vulnerability identified in multiple functions of the SnoozeHelper.java file in the Android system.

Which Android versions are affected by CVE-2023-20993? CVE-2023-20993 affects Android versions 11, 12, 12L, and 13.

What kind of risk does CVE-2023-20993 present? CVE-2023-20993 presents a security risk of local escalation of privilege, where a potentially malicious application could gain higher privileges.

Does CVE-2023-20993 require user interaction for exploitation? No, CVE-2023-20993 does not require user interaction for exploitation.

How was CVE-2023-20993 disclosed? CVE-2023-20993 was disclosed to the public through the Android Security Bulletin, specifically in the March 2023 bulletin.

What is the CVSS base score of CVE-2023-20993? CVE-2023-20993 has a CVSS base score of 7.8, categorized as HIGH. This score reflects a high-level severity.

What is the underlying issue leading to CVE-2023-20993? The underlying issue leading to CVE-2023-20993 is a failure to persist settings within the SnoozeHelper class.

Can you provide a possible attack scenario for CVE-2023-20993? A possible attack scenario for CVE-2023-20993 could involve a malicious application exploiting the underlying issue to gain higher privileges.

What is CVE-2023-20992? CVE-2023-20992 is a security vulnerability identified in the Bluetooth component of Android, specifically in the BluetoothLeController class.

Which Android versions are affected by CVE-2023-20992? CVE-2023-20992 affects devices running Android 13.

What type of vulnerability is CVE-2023-20992? CVE-2023-20992 is classified as an information disclosure vulnerability with a base score rated as 4.5.

Is user interaction required for exploiting CVE-2023-20992? No, user interaction is not required to exploit CVE-2023-20992. The vulnerability could be exploited without user interaction.

What kind of risk does CVE-2023-20992 present? To exploit CVE-2023-20992, an attacker would need System execution privileges, indicating that the vulnerability is a high-risk issue.

When was CVE-2023-20992 published? CVE-2023-20992 was published on 24 March 2023.

Where can more information about CVE-2023-20992 be found? More information and updates regarding CVE-2023-20992 can be found by visiting the Android Security Bulletin page.

What potential impact does CVE-2023-20992 have? An attack exploiting CVE-2023-20992 could likely involve a malicious application or code execution on the device.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-20991.

Can you provide a possible attack scenario for CVE-2023-20991? CVE-2023-20991 is a security vulnerability in the btm\_ble\_process\_periodic\_adv\_sync\_lost\_evt function of the Bluetooth component.

What is the base score of the CVE-2023-20991 vulnerability? The base score of the CVE-2023-20991 vulnerability is rated as 4.4 with a severity level of MEDIUM.

When was the CVE-2023-20991 vulnerability published? The CVE-2023-20991 vulnerability was published on 24 March 2023.

Which Android version is affected by the CVE-2023-20991 vulnerability? Android version Android-13 is affected by the CVE-2023-20991 vulnerability.

Where can more information about the CVE-2023-20991 vulnerability be found? More information about the CVE-2023-20991 vulnerability can be found on the Android Security Bulletin page.

What privileges are required to exploit the CVE-2023-20991 vulnerability? To exploit the CVE-2023-20991 vulnerability, an attacker would need System execution privileges on the device.

Is user interaction required for exploiting CVE-2023-20991? No, user interaction is not needed for the exploitation of CVE-2023-20991.

What is the Android ID associated with CVE-2023-20991? The Android ID associated with CVE-2023-20991 is A-255305114.

What are the requirements for exploiting CVE-2023-20991? Given the nature of CVE-2023-20991, an attacker with System execution privileges could exploit the vulnerability to gain higher privileges.

What is CVE-2023-20990? CVE-2023-20990 is a security vulnerability located in the btm\_ble\_rand\_enc\_complete function of the Bluetooth component.

What are the requirements for exploiting CVE-2023-20990? To exploit CVE-2023-20990, an attacker would need to have System execution privileges on the affected device.

Does CVE-2023-20990 require user interaction for exploitation? No, CVE-2023-20990 does not require user interaction for exploitation, which means it could potentially be exploited without user interaction.

What is the base score assigned to CVE-2023-20990? The base score assigned to CVE-2023-20990 is 4.4, which classifies it as a vulnerability with MEDIUM severity.

When was CVE-2023-20990 published? CVE-2023-20990 was published on 24 March 2023.

Can you provide more details about CVE-2023-20990? Yes, more details about CVE-2023-20990 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03>.

What versions of Android are affected by CVE-2023-20990? CVE-2023-20990 affects Android devices running Android version Android-13.

What type of vulnerability is CVE-2023-20990? CVE-2023-20990 is classified as an out of bounds read vulnerability due to a missing bounds check, which could lead to information disclosure.

What could A potential attack scenario for exploiting CVE-2023-20990 might involve a malicious application that is

Is there a p It is not mentioned in the provided information whether a proof of concept or exploit code for CVE-20

What is the CVE ID of the vulnerability described is CVE-2023-20989.

Can you su CVE-2023-20989 refers to a vulnerability in the `btm_ble_write_adv_enable_complete` function of `bt`

What is the CVE-2023-20989 has been assigned a Base Score of 4.4 with a severity rating of MEDIUM.

Which vers CVE-2023-20989 affects Android version 13.

What privil To exploit CVE-2023-20989, an attacker would need System execution privileges.

Is user inte No, user interaction is not needed to exploit the vulnerability described in CVE-2023-20989.

Where can More information about CVE-2023-20989 can be found at the Android Security Bulletin page, with the

What date CVE-2023-20989 was published on 24 March 2023.

Could you An attack scenario for exploiting CVE-2023-20989 could involve a malicious application with system-le

Are there a As a construct for the questions and answers, we don't provide real-world code examples or Proof of

What is the CVE ID for the vulnerability is CVE-2023-20988.

Can you de The vulnerability with CVE-2023-20988 concerns an out of bounds read due to a missing bounds check

Which pro The product affected by CVE-2023-20988 is Android.

What Andr CVE-2023-20988 impacts Android version 13.

What is the CVE-2023-20988 is 4.5, which is categorized as MEDIUM severity.

When was CVE-2023-20988 was published on 24 March 2023.

Are there a No, user interaction is not needed to exploit the vulnerability CVE-2023-20988.

Where can Further details about CVE-2023-20988 can be found at <https://source.android.com/security/bulletin/>

What kind System execution privileges are needed to exploit the vulnerability CVE-2023-20988.

Can you pr An attacker with system-level access on an affected Android 13 device could potentially exploit CVE-2

What is the CVE ID for the vulnerability related to an out of bounds read in `btm_read_link_quality_complete`

Can you de CVE-2023-20987 describes a vulnerability in `btm_read_link_quality_complete` of `btm_acl.cc`, where a

What versi The versions of Android affected by CVE-2023-20987 are those based on Android 13.

What is the CVE-2023-20987 has been assigned a severity level of MEDIUM with a base score of 4.5.

Was there No, user interaction is not needed for the exploitation of CVE-2023-20987.

Where can More information about CVE-2023-20987 can be found at the following URL: [https://source.android.c](https://source.android.com/security/bulletin/)

What are t An attacker would need System execution privileges in order to exploit the vulnerability described in

When was CVE-2023-20987 was published on 24 March 2023.

Could you A potential attack scenario for CVE-2023-20987 could involve an attacker with system-level access to

Are there c As CVE-2023-20987 involves an out of bounds read vulnerability in proprietary software components,

What is CV CVE-2023-20986 is a security vulnerability identified in a component of Android, more specifically in t

What is the CVE-2023-20986 is 4.4, which is considered MEDIUM severity.

What versi CVE-2023-20986 affects Android version 13.

What privil To exploit CVE-2023-20986, an attacker would need System execution privileges.

Is user inte No, user interaction is not required to exploit CVE-2023-20986.

When was CVE-2023-20986 was published on 24 March 2023.

Could you In a plausible attack scenario for CVE-2023-20986, an attacker who has already gained System executi

Where can More information about CVE-2023-20986 can be found at the Android Security Bulletin page, specific

Are there a As CVE-2023-20986 is a vulnerability disclosed in a security bulletin, specific exploit code examples are

What is CV CVE-2023-20985 refers to a security vulnerability identified in the `BTA_GATTS_HandleValueIndication`

How sever The vulnerability identified by CVE-2023-20985 has been assigned a base score of 7.8, which is catego

Which vers CVE-2023-20985 affects Android version 13. Users running this version of Android should ensure they

What are t To exploit CVE-2023-20985, an attacker does not need any additional execution privileges. Additionall

Has CVE-2( Yes, CVE-2023-20985 has been addressed by Google. Details of the security fix can be found in the An

Can you pr While exact code examples for exploiting CVE-2023-20985 might not be publicly available due to resp

What is the implication of a local escalation of privilege due to CVE-2023-20985 is that an unauthorized user could gain access to sensitive information. Where can more information about CVE-2023-20985 be found in the reference link provided, which directs to the Android Security Bulletin. What is CVE-2023-20984 refers to a security vulnerability that has been identified in the ParseBqrLinkQualityE method of the BqrLink class. How serious is CVE-2023-20984 has been given a Base Score of 4.4, which classifies it as a MEDIUM severity vulnerability. On what date was CVE-2023-20984 published on 24 March 2023.

Does CVE-2023-20984 require user interaction for exploitation, which implies that an attacker could exploit the vulnerability without the user's knowledge. What are the potential attack scenarios for CVE-2023-20984 involve an attacker gaining System execution privileges on the affected device. Where can more information about CVE-2023-20984 be found on the Android Security Bulletin webpage. You can find the details of CVE-2023-20984 affects Android version 13.

What privilege does CVE-2023-20984, an attacker would need to have System execution privileges on the affected device. What is the CVE ID of the vulnerability is CVE-2023-20983.

Can you describe CVE-2023-20983 is a vulnerability in btm\_ble\_rand\_enc\_complete of btm\_ble.cc that involves a possible local escalation of privilege. What are the attack scenarios for CVE-2023-20983, an attacker would need System execution privileges. No user interaction is required for exploitation. What is the CVSS base score for CVE-2023-20983 is 4.4, which is classified as MEDIUM severity. On what date was CVE-2023-20983 published on 24 March 2023.

Which product is affected by CVE-2023-20983, specifically version Android-13. What source can provide additional information on CVE-2023-20983 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-24-01>.

What might be a possible attack scenario for CVE-2023-20983 could involve a malicious application already having gained System execution privileges. What is CVE-2023-20982 is a security vulnerability identified in btm\_read\_tx\_power\_complete of btm\_acl.cc. How severe is CVE-2023-20982 has been evaluated with a Base Score of 4.4, which is classified as MEDIUM severity. What version of Android is affected by CVE-2023-20982 is Android-13.

Are there any real-world scenarios where CVE-2023-20982 could be exploited. In a real-world scenario, an attacker with the ability to execute code on an Android-13 device could exploit the vulnerability without user interaction. What user interaction is needed to exploit CVE-2023-20982. An attacker can exploit the vulnerability silently.

Has CVE-2023-20982 been patched. Information about the remediation or updates for CVE-2023-20982 should be available through the Android Security Bulletin. Where can more information about CVE-2023-20982 be found in the Android Security Bulletin for Pixel devices.

What privilege does CVE-2023-20982, an attacker would need System execution privileges on the affected Android device. What is CVE-2023-20981 refers to a security vulnerability in the 'btu\_ble\_rc\_param\_req\_evt' function of the 'btu\_ble.cc' file. What are the attack scenarios for CVE-2023-20981 could result in local information disclosure, where an attacker with System execution privileges could exploit the vulnerability.

Which Android version is affected by CVE-2023-20981 affects Android version 13. It is important for users and administrators to pay attention to updates. How can CVE-2023-20981 be mitigated. As of now, the primary method for mitigating CVE-2023-20981 involves applying updates provided by Google. What is the CVSS base score for CVE-2023-20981 is 4.4, which is categorized as MEDIUM severity. Where can more information about CVE-2023-20981 be found on the Android security bulletin page, specifically for Pixel devices. What date was CVE-2023-20981 published on 24 March 2023.

What are the possible attack scenarios for CVE-2023-20981 could involve a malicious app that has been granted System execution privileges. What is CVE-2023-20980 refers to a reported vulnerability in the file 'btu\_ble\_ll\_conn\_param\_upd\_evt' of 'btu\_ble.cc'.

How severe is CVE-2023-20980 is rated as '5.5 MEDIUM' on the Common Vulnerability Scoring System. What type of vulnerability is CVE-2023-20980 could enable an attack that leads to local information disclosure. An attacker with system execution privileges could exploit the vulnerability. Are there any fixes or patches for CVE-2023-20980 can typically be found in the Android Security Bulletin.

What kind of attack scenario for CVE-2023-20980, an attacker would need to have System execution privileges on the affected device. Do users need to interact with the device to exploit CVE-2023-20980. No, user interaction is not needed for the exploitation of CVE-2023-20980. The vulnerability can be exploited silently.

When was CVE-2023-20980 published on 24 March 2023. After vulnerabilities are discovered and reported, they are typically addressed in the next Android version. What type of vulnerability is CVE-2023-20979 identifies a possible out of bounds read vulnerability in the getNextSourceDataPack method of the BqrLink class.

What are the potential consequences of the vulnerability described in CVE-2023-20979 include local information disclosure. What version of Android is affected by CVE-2023-20979. Android version 13 is affected by CVE-2023-20979.

Is user interaction required for the exploitation of CVE-2023-20979. No, user interaction is not required for the exploitation of CVE-2023-20979. What is the CVSS base score given to CVE-2023-20979 is 5.5, which is categorized as MEDIUM severity. On what date was CVE-2023-20979 published on 24 March 2023.



What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-20979.

Where can I find more details on CVE-2023-20979? More details on CVE-2023-20979 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03#CVE-2023-20979>.

Can you provide an attack scenario involving CVE-2023-20979? In an attack scenario involving CVE-2023-20979, a malicious application on the device could potentially exploit the vulnerability to read data from the system's internal storage.

What is CVE-2023-20977? CVE-2023-20977 refers to a security vulnerability found in a specific part of the Android operating system.

What type of vulnerability is CVE-2023-20977? CVE-2023-20977 is categorized as a local information disclosure vulnerability, which poses a threat in the context of a compromised device.

Which versions of Android are affected by CVE-2023-20977? CVE-2023-20977 affects Android version 13, as specified in the vulnerability details.

What is the Base Score for CVE-2023-20977? CVE-2023-20977 has been assigned a Base Score of 4.4, indicating a medium level of severity for the vulnerability.

What are the prerequisites for exploiting CVE-2023-20977? In order to exploit CVE-2023-20977, an attacker would need System execution privileges on the compromised device.

What is the impact of CVE-2023-20977? No user interaction is necessary for the exploitation of CVE-2023-20977. The vulnerability could be exploited to read data from the system's internal storage.

When was CVE-2023-20977 published? CVE-2023-20977 was published on 24 March 2023.

Where can I find more information about CVE-2023-20977? More information about CVE-2023-20977 can be found at the Android Security Bulletin webpage, specifically at <https://source.android.com/security/bulletin/2023-03#CVE-2023-20977>.

Can you describe a possible attack scenario for CVE-2023-20977? A possible attack scenario for CVE-2023-20977 may involve a malicious application that has gained System execution privileges on the device.

What are the consequences if left unpatched? If left unpatched, CVE-2023-20977 could allow an attacker with System execution privileges to read data from the system's internal storage.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2023-20976.

What is the vulnerability? CVE-2023-20976 refers to a security issue in DefaultAutofillPicker.java, where there is a possibility of an out-of-bounds read.

Which product is affected by CVE-2023-20976? The product affected by CVE-2023-20976 is Android, specifically version Android-13.

Are there any prerequisites for exploiting CVE-2023-20976? For CVE-2023-20976, no additional execution privileges are needed to exploit the vulnerability. However, the attacker must have access to the system's internal storage.

What is the Base Score for CVE-2023-20976? CVE-2023-20976 has been assigned a Base Score of 7.3, which classifies it as HIGH in severity.

When was CVE-2023-20976 published? CVE-2023-20976 was first published on 24 March 2023.

Where can I find more information about CVE-2023-20976? More information about CVE-2023-20976 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03#CVE-2023-20976>.

Can you provide an example of how CVE-2023-20976 might be exploited? An example of how CVE-2023-20976 might be exploited involves an attacker crafting a malicious input string that triggers the out-of-bounds read.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-20975.

What type of vulnerability is CVE-2023-20975? The vulnerability reported in CVE-2023-20975 relates to a permissions bypass which could lead to local information disclosure.

Which versions of Android are affected by CVE-2023-20975? Android version Android-13 is affected by CVE-2023-20975.

What is the Base Score for CVE-2023-20975? The CVSS Base Score assigned to CVE-2023-20975 is 7.8, which is classified as HIGH.

What type of interaction is needed to exploit CVE-2023-20975? No user interaction is needed to exploit the vulnerability in CVE-2023-20975.

On what date was CVE-2023-20975 published? CVE-2023-20975 was published on 24 March 2023.

Where can I find more information about the CVE-2023-20975 vulnerability? More information about the CVE-2023-20975 vulnerability can be found at the following reference URL: <https://source.android.com/security/bulletin/2023-03#CVE-2023-20975>.

Are there any code examples provided for CVE-2023-20975? As CVE-2023-20975 is a permissions bypass issue, specific code examples are typically not provided for this type of vulnerability.

Can you describe an attack scenario for CVE-2023-20975? An attack scenario for CVE-2023-20975 could involve a malicious app that exploits the permissions bypass to access sensitive data.

What is the Android ID associated with CVE-2023-20975? The Android ID associated with CVE-2023-20975 is A-250573776.

What is the CVE ID for the vulnerability involving an out of bounds read? The CVE ID for the vulnerability involving an out of bounds read is CVE-2023-20974.

What is the base score for CVE-2023-20974? The base score given to CVE-2023-20974 is 5.5, which is categorized as MEDIUM severity.

In which part of the code does CVE-2023-20974 occur? CVE-2023-20974 occurs in the btm\_ble\_add\_resolving\_list\_entry\_complete function within the btm module.

Which Android version is affected by CVE-2023-20974? CVE-2023-20974 affects Android version Android-13.

What type of interaction is needed to exploit CVE-2023-20974? An attacker would need System execution privileges to exploit CVE-2023-20974.

Is user interaction needed for CVE-2023-20974? No, user interaction is not needed for the exploitation of CVE-2023-20974.

What is the Android ID associated with CVE-2023-20974? The Android ID associated with CVE-2023-20974 is A-260078907.

Can you provide more information about CVE-2023-20974? Yes, more information about CVE-2023-20974 can be found at the following link: <https://source.android.com/security/bulletin/2023-03#CVE-2023-20974>.

What kind of vulnerability is CVE-2023-20974? CVE-2023-20974 is an out of bounds read vulnerability due to a missing bounds check. The impact could be local information disclosure.

Could you describe an attack scenario for CVE-2023-20974? An attack scenario for CVE-2023-20974 could involve a malicious application already having System execution privileges that triggers the out of bounds read.

What is the CVE ID for the vulnerability? The CVE ID for the vulnerability is CVE-2023-20973.

Can you describe CVE-2023-20973? CVE-2023-20973 refers to a vulnerability where there is a possible out of bounds read due to a missing bounds check.

What is the Base Score for CVE-2023-20973? CVE-2023-20973 has been assigned a Base Score of 5.5, which is categorized as MEDIUM severity.

When was CVE-2023-20973 published? CVE-2023-20973 was published on 24 March 2023.

Which Android versions are affected by CVE-2023-20973? The Android versions affected by CVE-2023-20973 are versions of Android-13.

What is the Android ID associated with CVE-2023-20973? The Android ID associated with CVE-2023-20973 is A-260568245.

Where can I find more information about CVE-2023-20973? More information about CVE-2023-20973 can be found at the following link: <https://source.android.com/security/bulletin/2023-03#CVE-2023-20973>.

What kind To exploit the vulnerability CVE-2023-20973, an attacker would need System execution privileges.

Is user inte No, user interaction is not required for the exploitation of CVE-2023-20973.

What are t Potential attack scenarios for CVE-2023-20973 involve an attacker with system-level access reading d

What is the CVE ID of the reported vulnerability is CVE-2023-20972.

Could you CVE-2023-20972 refers to a potential out of bounds read in the 'btm\_vendor\_specific\_evt' function of

What is the CVE-2023-20972 has been assigned a severity level of MEDIUM with a base score of 5.5.

On which c The CVE-2023-20972 vulnerability was published on 24 March 2023.

Which And The Android version impacted by CVE-2023-20972 is Android-13.

What kind To exploit the CVE-2023-20972 vulnerability, System execution privileges are required.

Is user inte No, user interaction is not needed to exploit the CVE-2023-20972 vulnerability.

Where can More information about CVE-2023-20972 can be found at the following URL: <https://source.android.c>

What Andr The Android ID associated with CVE-2023-20972 is A-255304665.

Can you pr In a potential attack scenario for CVE-2023-20972, a malicious application with system-level execution

What is the CVE ID allocated to this vulnerability is CVE-2023-20971.

Can you de CVE-2023-20971 is a vulnerability in the updatePermissionTreeSourcePackage of PermissionManager

What versi Android version 13 is affected by the vulnerability CVE-2023-20971.

What is the CVE-2023-20971 is 7.8, which is categorized as HIGH.

When was CVE-2023-20971 was published on 24 March 2023.

What is the CVE-2023-20971 is the Android Security Bulletin, which

What are s Potential attack scenarios for CVE-2023-20971 include a malicious app exploiting the logic error in the

Is there a r No, user interaction is not needed for an attacker to exploit CVE-2023-20971.

Are there e No code examples are provided for vulnerability CVE-2023-20971 within this context. Generally, speci

What is the CVE ID of the vulnerability is CVE-2023-20970.

In which pr The file p2p\_iface.cpp is a C++ file, as indicated by the '.cpp' file extension.

What type Exploitation of CVE-2023-20970 requires System execution privileges.

Is user inte No, user interaction is not required to exploit the vulnerability described in CVE-2023-20970.

What is the CVE-2023-20970 is local information disclosure, potentially allowing an attacker to gain

What is the CVE-2023-20970 is 4.4, which is considered MEDIUM.

On which c CVE-2023-20970 was published on 24 March 2023.

Which And Android version 13 is affected by CVE-2023-20970.

Where can More information about CVE-2023-20970 can be found on the Android Security Bulletin at <https://sou>

What might While the actual code affected by CVE-2023-20970 is not supplied in the vulnerability details, an exam

Can you de An attacker with System execution privileges might leverage the out of bounds read vulnerability in p

What is the CVE ID for the vulnerability involving a possible out of bounds read in multiple locations of p2p\_if

What type CVE-2023-20969 addresses an issue that leads to a possible out of bounds read due to a missing boun

Which vers CVE-2023-20969 affects Android version 13.

What are t To exploit the vulnerability described in CVE-2023-20969, an attacker would require System execution

What is the CVE-2023-20969 has been assigned a CVSS base score of 4.4, indicating a severity level of MEDIUM.

When was CVE-2023-20969 was published on 24 March 2023.

Where can More information about CVE-2023-20969 can be found at the following URL: <https://source.android.c>

Can you ex A possible attack scenario for CVE-2023-20969 would involve an attacker with System execution privil

What is the CVE ID for the vulnerability involving a possible out-of-bounds read in p2p\_iface.cpp is CVE-2023-

What kind CVE-2023-20968 identifies a vulnerability that allows for a possible out-of-bounds read due to a missi

Which pro The product affected by CVE-2023-20968 is Android.

What are t CVE-2023-20968 impacts Android version Android-13.

What is the CVE-2023-20968 is MEDIUM with a base score of 4.4.

When was CVE-2023-20968 was published on 24 March 2023.

Is there a r Yes, more details about CVE-2023-20968 can be found at the following link: [https://source.android.co](https://source.android.com/security/bulletin/2023-03-01)

What type To exploit the vulnerability detailed in CVE-2023-20968, System execution privileges are needed.

Does explo No, user interaction is not needed for the exploitation of CVE-2023-20968.

What is the The potential impact of CVE-2023-20968 on a user's device could lead to local information disclosure.

How could An attacker could potentially exploit CVE-2023-20968 by triggering a scenario where the vulnerability

What is the The CVE ID for the vulnerability affecting the inflate function in inflate.c is CVE-2023-20966.

Can you pr CVE-2023-20966 is a security vulnerability that resides in the inflate function of inflate.c. It involves a

What is the The severity base score for CVE-2023-20966 is 7.8, which is classified as HIGH.

When was CVE-2023-20966 was published on 24 March 2023.

Which And The Android versions affected by CVE-2023-20966 include Android 11, Android 12, Android 12L, and A

Are there a Yes, more information about CVE-2023-20966 can be found at the Android Security Bulletin, available

What are t If CVE-2023-20966 is successfully exploited, it could lead to a local escalation of privilege. This means

What are s Potential attack scenarios for CVE-2023-20966 might include an attacker crafting a malicious input the

What is the The CVE ID associated with this vulnerability is CVE-2023-20964.

Can you de CVE-2023-20964 involves a potential intent rebroadcast issue in multiple functions of MediaSessionRe

What versi The Android versions affected by CVE-2023-20964 are Android 12, Android 12L (12.1), and Android 13

What is the The severity level of CVE-2023-20964 is rated as HIGH with a base score of 7.8.

Where can More detailed information about CVE-2023-20964 can be found in the Android Security Bulletin at the

Is user inte No, user interaction is not needed to exploit the vulnerability CVE-2023-20964.

Could you In an attack scenario exploiting CVE-2023-20964, a malicious app on the device could send or intercept

What is the The CVE ID for the vulnerability found in WorkSource affecting Android is CVE-2023-20963.

Can you de CVE-2023-20963 is described as a possible parcel mismatch in WorkSource, which could lead to local e

What is the The base score assigned to CVE-2023-20963 is 7.8, categorized as HIGH. This signifies that the vulnera

Which And The Android versions affected by CVE-2023-20963 are Android 11, Android 12, Android 12L, and Andr

Where can Official information regarding CVE-2023-20963 can be found on the Android Security Bulletin page at

What kind Due to CVE-2023-20963, there is a potential for local escalation of privilege, meaning a lower privileg

Is user inte No, user interaction is not needed to exploit the vulnerability described by CVE-2023-20963.

What are p Potential attack scenarios for CVE-2023-20963 could include a malicious app running on the device th

What is the The CVE ID of the vulnerability is CVE-2023-20962.

What type CVE-2023-20962 describes a potential exploit that could lead to local information disclosure due to th

On which p The CVE-2023-20962 vulnerability is found on the Android platform, specifically on Android version 11

What is the The CVSS base score given to CVE-2023-20962 is 5.5 MEDIUM, which signifies that the vulnerability pr

Was user i No, user interaction is not needed for the exploitation of the vulnerability identified by CVE-2023-209

Where can More information about CVE-2023-20962 can be found on the Android Security Bulletin page, at this l

When was CVE-2023-20962 was published on 24 March 2023.

What are t One possible attack scenario of the CVE-2023-20962 vulnerability would involve a malicious applicati

Can you pr While I don't have specific code demonstrating the CVE-2023-20962 vulnerability, the issue arises from

What is CV CVE-2023-20960 refers to a security vulnerability in the SettingsHomepageActivity.java file of the And

How sever CVE-2023-20960 has been assigned a base score of 8.8, which is categorized as HIGH severity. This inc

Does explo No, user interaction is not needed for the exploitation of CVE-2023-20960. An attacker can exploit the

When was CVE-2023-20960 was published on 24 March 2023.

What Andr The Android versions affected by CVE-2023-20960 are Android-12L and Android-13.

Where can More information about CVE-2023-20960 can be found on the Android Security Bulletin website at <https://source.android.com/security/bulletin/2023-03-01>

Can you de A possible attack scenario for CVE-2023-20960 could involve a malicious application installed on the u

What type To exploit CVE-2023-20960, an attacker would need user execution privileges. Although it does not re

What is the The CVE ID for the vulnerability that affects AddSupervisedUserActivity in Android is CVE-2023-20959

Can you de CVE-2023-20959 is a security vulnerability where guest users can initiate AddSupervisedUserActivity v

What is the CVE-2023-20959 has a CVSS base score of 7.8, which categorizes it as a HIGH severity vulnerability.

When was CVE-2023-20959 published on 24 March 2023.

Which And CVE-2023-20959 affects Android version Android-13.

Are there a Yes, detailed information about CVE-2023-20959 can be found at the following reference link: <https://source.android.com/security/bulletin/2023-03-01>

What kind CVE-2023-20959 allows for local escalation of privilege. This means that a guest user could potentially

What are t In an attack scenario exploiting CVE-2023-20959, a malicious guest user could trigger the AddSupervis

What is the The CVE ID of the vulnerability is CVE-2023-20958.

Can you de CVE-2023-20958 is a security vulnerability that involves a possible out-of-bounds read due to a heap k

What is the The CVSS Base Score given to CVE-2023-20958 is 7.1, which is categorized as HIGH. This indicates that

When was CVE-2023-20958 published on 24 March 2023.

What versi CVE-2023-20958 affects Android version Android-13.

Where can More information or technical details about CVE-2023-20958 can be found in the Android Security Bul

What type An attacker could exploit CVE-2023-20958 by crafting a malicious input that triggers a heap buffer ove

What is the The Android ID associated with CVE-2023-20958 is A-254803162.

What is the The CVE ID for the vulnerability involving Android's SettingsPreferenceFragment reported in Spring 20

Can you de CVE-2023-20957 involves a potential bypass of Factory Reset Protections in the onAttach method of S

What is the The severity base score assigned to CVE-2023-20957 is 7.8, which is categorized as HIGH.

Which vers The Android versions affected by the vulnerability CVE-2023-20957 include Android 11, Android 12, an

Does CVE- No, user interaction is not needed for exploiting CVE-2023-20957.

Where can Additional details and updates related to CVE-2023-20957 can be found on the official Android Secur

What pote The exploitation of CVE-2023-20957 could lead to a local attacker gaining escalated privileges on a de

On what d CVE-2023-20957 was published on the 24th of March, 2023.

Could you In the context of CVE-2023-20957, a confused deputy problem refers to a situation where an Android

What mea To protect their Android devices from the vulnerability CVE-2023-20957, users should ensure that the

What is the The CVE ID associated with the out of bounds write in C2SurfaceSyncObj.cpp is CVE-2023-20956.

What kind CVE-2023-20956 describes a vulnerability that could lead to a possible out of bounds write due to a m

Which vers CVE-2023-20956 affects Android versions Android-12, Android-12L, and Android-13.

What are t To exploit CVE-2023-20956, an attacker would need to have System execution privileges on the device

Is user inte No, user interaction is not required to exploit the vulnerability identified by CVE-2023-20956.

How sever The vulnerability CVE-2023-20956 has a Base Score of 4.4, categorizing it as MEDIUM severity.

When was CVE-2023-20956 published on 24 March 2023.

Where can More information about CVE-2023-20956 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-01>

What coul A possible attack scenario for CVE-2023-20956 could involve an attacker with system-level access on a

Can you pr While specific code examples related to CVE-2023-20956 are not provided as this would depend on th

What is the The vulnerability that allows bypassing admin restrictions to uninstall applications for all users is iden

Can you de CVE-2023-20955 is a security issue found in the onPrepareOptionsMenu method of AppInfoDashboar

What is the The impact scope of CVE-2023-20955 affects multiple Android versions including Android 11, Android

How sever CVE-2023-20955 has been assigned a base score of 7.8, which is categorized as HIGH severity. This ref

When was CVE-2023-20955 published on 24 March 2023.

Where can You can find more information or the official advisory about CVE-2023-20955 at the Android Security |

What are t One possible attack scenario for CVE-2023-20955 involves an attacker exploiting the missing permissi

What is CV CVE-2023-20954 refers to a security vulnerability that was identified in the SDP\_AddAttribute functio

Which And CVE-2023-20954 impacts multiple Android versions including Android 11, Android 12, Android 12L, an

How sever The severity of the CVE-2023-20954 vulnerability is considered 'CRITICAL', with a CVSS (Common Vuln

When was The CVE-2023-20954 vulnerability was published on 24 March 2023.

Does the C No, CVE-2023-20954 can be exploited without any user interaction, making it especially dangerous sir

Where can More detailed information about CVE-2023-20954 can be found in the Android Security Bulletin at the

Can you ex A potential attack scenario for CVE-2023-20954 could involve a malicious actor sending specially craft  
What are t The consequences of an attack exploiting CVE-2023-20954 can range from unauthorized data access a  
What is CV CVE-2023-20953 refers to a security vulnerability found in Android's ClipboardListener.java, where the  
Which And CVE-2023-20953 affects Android version 13.

What type CVE-2023-20953 could lead to a local escalation of privilege where the attacker could gain elevated ac  
Is user inte Yes, user interaction is required to exploit the vulnerability described in CVE-2023-20953.

How sever The severity of CVE-2023-20953 is rated as 7.8, which is categorized as HIGH according to its Base Sco  
When was CVE-2023-20953 was publicly disclosed on 24 March 2023.

Where can More information about CVE-2023-20953 can be found at the Android Security Bulletin webpage at ht

Can you pr An attacker could exploit CVE-2023-20953 by tricking a user into performing certain actions that inter

What is the CVE ID for the vulnerability identified in the A2DP codec header building process is CVE-2023-209

Can you de CVE-2023-20952 is a vulnerability in the A2DP\_BuildCodecHeaderSbc function of a2dp\_sbc.cc. The iss

Which vers CVE-2023-20952 affects multiple versions of Android including Android 11, Android 12, Android 12L, a

What is the CVE-2023-20952 has been assigned a base score of 5.5, which classifies it as a MEDIUM severity vulne

On what d CVE-2023-20952 was published on the 24th of March, 2023.

Where can Official information or an advisory about CVE-2023-20952 can be found at the following URL: [https://](https://source.android.com/security/bulletin/2023-03-24/cve-2023-20952)

Could you An attacker could exploit CVE-2023-20952 by tricking the Android system into processing a specially c

What is the CVE ID for the vulnerability in the gatt\_process\_prep\_write\_rsp function in Android is CVE-2023-2

What versi The versions of Android affected by CVE-2023-20951 are Android 11, Android 12, Android 12L, and An

What type CVE-2023-20951 represents a possible out of bounds write due to a missing bounds check in the gatt\_

Is user inte No, user interaction is not required to exploit the vulnerability described by CVE-2023-20951.

What is the CVE-2023-20951 is 9.8, which is classified as CRITICAL in terms of severity.

On what d CVE-2023-20951 was published on 24 March 2023.

Where can More information regarding CVE-2023-20951 can be found at the Android Security Bulletin URL: [https://](https://source.android.com/security/bulletin/2023-03-24/cve-2023-20951)

What are t Possible attack scenarios for CVE-2023-20951 include a remote attacker sending a specially crafted Bl

How might An attacker could exploit CVE-2023-20951 by crafting a malicious Bluetooth message that, when proc

What is the CVE ID of the security vulnerability is CVE-2023-20947.

What versi The versions of Android affected by CVE-2023-20947 are Android-12, Android-12L, and Android-13.

What is the CVE-2023-20947 has been assigned a CVSS Base Score of 7.8, which is categorized as HIGH.

On what d CVE-2023-20947 was published on 24 March 2023.

What is the CVE-2023-20947 could lead to local escalation of privilege on affected Android systems without needi

Is user inte No, user interaction is not needed to exploit CVE-2023-20947.

Can you pr Yes, you can find more information and patches related to CVE-2023-20947 at the Android Security Bu

What kind CVE-2023-20947 relates to a possible way to keep a one-time permission granted, which constitutes a

What is the CVE-2023-20947 is that an attacker could maintain acce

Are there While specific attack scenarios for CVE-2023-20947 are not detailed, an attacker could exploit this vul

What is CV CVE-2023-20936 refers to a security vulnerability that was identified in the 'bta\_av\_rc\_disc\_done' fun

What is the CVE-2023-20936 is rated as 7.8, which is classified as 'HIGH' according to the stan

Did CVE-20 No, CVE-2023-20936 does not require user interaction for exploitation, which indicates that the vulne

On what d CVE-2023-20936 was publicly disclosed on 24 March 2023.

Which vers CVE-2023-20936 affects Android versions 11, 12, 12L, and 13.

Where can More information about CVE-2023-20936 can be found in the Android Security Bulletin at [https://sou](https://source.android.com/security/bulletin/2023-03-24/cve-2023-20936)

Can you pr In an attack scenario involving CVE-2023-20936, a malicious application installed on the device could

What is the CVE ID associated with CVE-2023-20936 is A-226927612.

What is the CVE ID of the vulnerability published on March 24, 2023, is CVE-2023-20931.

Can you de CVE-2023-20931 describes a vulnerability due to a possible out of bounds write caused by a heap buff

What versi CVE-2023-20931 affects the following versions of Android: Android-11, Android-12, Android-12L, and

Does the vulnerability require user interaction? No, the exploitation of the vulnerability identified by CVE-2023-20931 does not require user interaction.

What is the severity? The severity Base Score assigned to CVE-2023-20931 is 7.8, which is categorized as HIGH.

Where can I find more detailed information? More detailed information about CVE-2023-20931 can be found at the Android Security Bulletin page, [https://source.android.com/security/bulletin/2023-03/01](#).

What is the impact? Exploiting the CVE-2023-20931 vulnerability could lead to local escalation of privilege on the device. An attacker could potentially gain access to sensitive data and perform actions that the user is not authorized to perform.

Is there a proof of concept? While specific code examples demonstrating CVE-2023-20931 are not typically provided publicly to avoid exploitation, a possible attack scenario for CVE-2023-20931 involves an attacker writing a malicious app that triggers the vulnerability.

What is the vulnerability related to? The vulnerability related to the possible disclosure of nearby Bluetooth MAC addresses on Android is CVE-2023-20929.

What class does CVE-2023-20929 refer to? CVE-2023-20929 refers to a local information disclosure security vulnerability in the `BluetoothLeScanner` class.

Which Android version is affected? The Android version affected by CVE-2023-20929 is Android 13.

What is the severity? The severity base score of CVE-2023-20929 is rated as 5.5, which is categorized as MEDIUM severity.

What is the component affected? The component affected by CVE-2023-20929 is the `sendHalfSheetCancelBroadcast` method within the `BluetoothLeScanner` class.

Do users need to interact? No, user interaction is not required for the exploitation of CVE-2023-20929.

Where can I find detailed information? Detailed information about CVE-2023-20929 can be found in the Android Security Bulletin at the following URL: [https://source.android.com/security/bulletin/2023-03/02](#).

What kind of impact? To exploit the vulnerability CVE-2023-20929, no additional execution privileges are needed. An attacker could potentially gain access to sensitive data and perform actions that the user is not authorized to perform.

What is the CVE ID? CVE-2023-20929 was published on 24 March 2023.

What are possible attack scenarios? Possible attack scenarios for CVE-2023-20929 include an attacker leveraging the unrestricted broadcast permissions to trigger the vulnerability and gain access to sensitive data.

What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2023-20926.

In which file was the vulnerability located? The vulnerability CVE-2023-20926 was located in the file `HeaderPrivacyConsController.kt`.

What are the affected Android versions? The Android versions affected by CVE-2023-20926 are Android-12, Android-12L, and Android-13.

What type of impact? CVE-2023-20926 could lead to a local escalation of privilege on a device. An attacker could potentially gain access to sensitive data and perform actions that the user is not authorized to perform.

Does exploitation require user interaction? No, exploiting CVE-2023-20926 does not require user interaction or additional execution privileges.

What is the CVSS base score? The CVSS base score of CVE-2023-20926 is 6.8, and its severity level is MEDIUM.

On what date was the vulnerability published? The vulnerability CVE-2023-20926 was published on 24 March 2023.

Where can I find more details? More details about CVE-2023-20926 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03/03>.

What is the Android ID associated with the vulnerability? The Android ID associated with CVE-2023-20926 is A-253043058.

Can an attacker exploit the vulnerability? An attacker could potentially exploit CVE-2023-20926 by gaining physical access to a device that has been updated to an affected Android version.

What is the CVE ID for the vulnerability related to a logic error in ResolverActivity.java? The CVE ID for the vulnerability related to a logic error in ResolverActivity.java in certain Android versions is CVE-2023-20917.

Which versions of Android are affected? The versions of Android affected by CVE-2023-20917 include Android 11, Android 12, Android 12L, and Android 13.

What is the severity rating? CVE-2023-20917 has been assigned a severity rating of HIGH with a Base Score of 7.8.

Is user interaction required? No, user interaction is not required to exploit the vulnerability described in CVE-2023-20917.

What type of impact? CVE-2023-20917 could lead to a local escalation of privilege on the affected Android devices. An attacker could potentially gain access to sensitive data and perform actions that the user is not authorized to perform.

Are there any additional execution privileges needed? No additional execution privileges are needed to exploit CVE-2023-20917.

Can you provide official details or patch information? Yes, official details or patch information for CVE-2023-20917 can be found on the Android Security Bulletin page, [https://source.android.com/security/bulletin/2023-03/04](#).

What is the logic error in ResolverActivity.java specified by CVE-2023-20917? The logic error in ResolverActivity.java specified by CVE-2023-20917 could potentially lead to sharing sensitive information with nearby devices.

When was CVE-2023-20917 published? CVE-2023-20917 was published on 24 March 2023.

Could you provide a possible attack scenario? In a possible attack scenario involving CVE-2023-20917, a malicious app could exploit the logic error in ResolverActivity.java to share sensitive information with nearby devices.

What is CVE-2023-20911? CVE-2023-20911 is a security vulnerability identified in Android's `PermissionManagerServiceImpl.java`.

How severe is the vulnerability? The CVE-2023-20911 vulnerability is rated as 7.8 HIGH on the CVSS (Common Vulnerability Scoring System).

Does CVE-2023-20911 require user interaction? No, CVE-2023-20911 does not require user interaction for exploitation. An attacker can exploit the vulnerability without the user's knowledge or consent.

When was CVE-2023-20911 publicly disclosed? CVE-2023-20911 was publicly disclosed on 24 March 2023.

Which Android versions are affected? CVE-2023-20911 affects Android versions 11, 12, 12L, and 13.

What is the official source for security patch information? The official source for security patch information regarding CVE-2023-20911 is the Android Security Bulletin, [https://source.android.com/security/bulletin/2023-03/05](#).

What potential impact could CVE-2023-20911 have? CVE-2023-20911 could lead to a local escalation of privilege on an affected Android device. This means an attacker could potentially gain access to sensitive data and perform actions that the user is not authorized to perform.

Is there a code example for exploiting CVE-2023-20911? An explicit code example for exploiting CVE-2023-20911 is not provided here. Security practices discourage the public release of exploit code.

What are some possible attack scenarios? Possible attack scenarios for CVE-2023-20911 could include a malicious application exploiting the vulnerability to gain access to sensitive data and perform actions that the user is not authorized to perform.

What is CVE-2023-20906? CVE-2023-20906 refers to a vulnerability in the `PermissionManagerService.java` of the Android operating system.

Which versions of Android are affected? CVE-2023-20906 affects Android versions Android-11, Android-12, Android-12L, and Android-13.

What is the severity? The security issue described in CVE-2023-20906 is a permissions bypass vulnerability that could potentially allow an attacker to access sensitive information. CVE-2023-20906 has a CVSS base score of 7.8, which is categorized as HIGH severity, indicating that there is a high level of risk. How can an attacker exploit the vulnerability? An attacker could exploit the vulnerability in CVE-2023-20906 by creating a malicious app that targets the vulnerability. Was any user interaction required? No, user interaction is not needed to exploit the vulnerability in CVE-2023-20906. An attacker could leverage the vulnerability without the need for user interaction. When was CVE-2023-20906 published? CVE-2023-20906 was published on 24 March 2023.

Can you provide more details? For more detailed information about CVE-2023-20906, one can refer to the following link from the Android Security Bulletin: <https://source.android.com/security/bulletin/2023-03-01#CVE-2023-20906>. What is the CVE ID for the described vulnerability? The CVE ID for the described vulnerability is CVE-2022-42528.

Can you explain the details? CVE-2022-42528 details a vulnerability in the `ffa_mrd_prot` function of `shared_mem.c`. This is attributed to a buffer overflow. What is the severity? The CVSS Base Score for CVE-2022-42528 is 5.5, which classifies it as a MEDIUM severity vulnerability. On what date was CVE-2022-42528 published? CVE-2022-42528 was published on 24 March 2023.

What versions are affected? The affected versions of Android by CVE-2022-42528 are documented within the Android kernel, but the specific versions are not listed. What type of interaction is required? No user interaction is required to exploit the vulnerability CVE-2022-42528.

Where can more information be found? More information about CVE-2022-42528 can be found on the Android security bulletin page at <https://source.android.com/security/bulletin/2023-03-01#CVE-2022-42528>.

What are potential attack scenarios? Potential attack scenarios for CVE-2022-42528 might involve a malicious local application exploiting the vulnerability. What is the CVE ID of the vulnerability? The CVE ID of the vulnerability is CVE-2022-42500.

Can you describe the vulnerability? CVE-2022-42500 refers to a vulnerability in the `OEM_OnRequest` function of `scdd.cpp` within the Android kernel. What is the severity? The severity base score for CVE-2022-42500 is rated as 6.7, which is categorized as MEDIUM severity. When was CVE-2022-42500 published? CVE-2022-42500 was published on 24 March 2023.

What versions are affected? CVE-2022-42500 affects Android kernel versions that were not specified in the provided details. Users should check their device's kernel version for more information.

Where can more information be found? You can find more information or references about CVE-2022-42500 on the Android security bulletin page at <https://source.android.com/security/bulletin/2023-03-01#CVE-2022-42500>.

What are the required privileges? To exploit the vulnerability designated by CVE-2022-42500, an attacker would require System execution permissions. Is user interaction required? No, user interaction is not required for the exploitation of CVE-2022-42500.

What are potential attack scenarios? Potential attack scenarios involving CVE-2022-42500 could include an attacker with system-level access exploiting the vulnerability to gain unauthorized access to system resources.

What is the CVE ID for the vulnerability involving a possible out of bounds write in `sms_SendMmCpErrMsg`? The CVE ID for the vulnerability involving a possible out of bounds write in `sms_SendMmCpErrMsg` is CVE-2022-42499.

Can you provide details? CVE-2022-42499 is a security vulnerability where there is a possible heap buffer overflow in `sms_SendMmCpErrMsg`. What is the severity? CVE-2022-42499 has been assigned a Base Score of 9.8, and its severity level is considered CRITICAL. When was CVE-2022-42499 published? CVE-2022-42499 was published on 24 March 2023.

What product is affected? The product affected by CVE-2022-42499 is Android, specifically the Android kernel. The versions impacted are not specified. Where can more information be found? More information about the CVE-2022-42499 vulnerability can be found at the following link: <https://source.android.com/security/bulletin/2023-03-01#CVE-2022-42499>.

Does CVE-2022-42499 require user interaction for exploitation? No, CVE-2022-42499 does not require user interaction for exploitation.

Could you provide a potential attack scenario? A potential attack scenario for CVE-2022-42499 involves a malicious actor sending a specially crafted message to the device.

What is the CVE ID for the vulnerability found in Pixel cellular firmware? The CVE ID for the vulnerability found in Pixel cellular firmware is CVE-2022-42498.

Can you describe the vulnerability? CVE-2022-42498 is a vulnerability in Pixel cellular firmware that involves a possible out of bounds write.

What type of interaction is required? User interaction is not needed for the exploitation of CVE-2022-42498, which makes the vulnerability more severe.

What is the severity score? The severity score of CVE-2022-42498 is 9.8, which classifies it as CRITICAL according to the common CVSS scoring system.

When was CVE-2022-42498 publicly disclosed? CVE-2022-42498 was publicly disclosed on 24 March 2023.

Where can more details and advisories be found? More details and advisories about CVE-2022-42498 can be found at the following URL: <https://source.android.com/security/bulletin/2023-03-01#CVE-2022-42498>.

What product is affected? The product affected by CVE-2022-42498 is the Android operating system, specifically the Android kernel.

Could you provide a potential attack scenario? A potential attack scenario for CVE-2022-42498 could involve a remote attacker sending specially crafted data to the device.

Are there code examples for exploitation? I am unable to provide code examples for the exploitation of CVE-2022-42498 due to ethical reasons and the complexity of the vulnerability.

What is the CVE ID for the vulnerability in the `parseParamsBlob` function of `types.cpp`? The CVE ID for the vulnerability in the `parseParamsBlob` function of `types.cpp` is CVE-2022-20542.

How severe is the vulnerability? The security vulnerability represented by CVE-2022-20542 is considered 'HIGH' with a base score of 7.0.

Was the vulnerability addressed? Yes, the CVE-2022-20542 vulnerability was addressed by Google. Information about the fix can be found in the Android Security Bulletin.

As an Android developer, where can more information be found? As an Android developer, you can find more information about CVE-2022-20542 in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-03-01#CVE-2022-20542>.

Does the exploitation require user interaction? No, the exploitation of CVE-2022-20542 does not require user interaction. This makes the vulnerability more severe.

Could you provide a possible attack scenario? A possible attack scenario exploiting CVE-2022-20542 would involve a malicious application or malware attempting to exploit the vulnerability.

What is the CVE ID associated with the vulnerability in `MPEG4Extractor.cpp`? The CVE ID associated with the vulnerability in `MPEG4Extractor.cpp` is CVE-2022-20532.

What is the CVE-2022-20532 is described as a vulnerability in the `parseTrackFragmentRun()` function of `MPEG4Extractor`. How severe The severity of CVE-2022-20532 is considered 'CRITICAL', with a base score of 9.8.

On what date CVE-2022-20532 was published on 24th March 2023.

Which Android version The vulnerability CVE-2022-20532 affects Android version 13.

What is the Android ID associated with CVE-2022-20532 is A-232242894.

Can you provide For more information about CVE-2022-20532, you can refer to the Android Security Bulletin at: <https://source.android.com/security/bulletin/2023-02>.

What are the possible exploits An attacker could exploit CVE-2022-20532 by crafting a malicious media file and convincing a user to open it.

Are there any code examples As CVE-2022-20532 is a security vulnerability, code examples are generally not published to avoid facilitating further exploitation.

What is CVE-2022-20499 refers to a security vulnerability found in the `PasspointConfiguration.java` file of Android.

How severe The CVE-2022-20499 vulnerability is rated with a base score of 5.5 which is considered MEDIUM severity.

Which versions The Android versions affected by CVE-2022-20499 are Android 12, Android 12L, and Android 13.

Does CVE-2022-20499 require user interaction No, user interaction is not required for the exploitation of CVE-2022-20499.

When was CVE-2022-20499 published on 24 March 2023.

Where can More information about CVE-2022-20499 can be found at the Android Security Bulletin URL: <https://source.android.com/security/bulletin/2023-02>.

How could An attacker could exploit CVE-2022-20499 by crafting a maliciously configured Passpoint configuration.

Are there any code examples Since CVE-2022-20499 involves a security issue with the parsing mechanism in the Android OS, it is not recommended to publish code examples.

What is CVE-2022-20467 refers to a security vulnerability in the Android operating system. Specifically, it occurs when parsing a malformed XML file.

How severe CVE-2022-20467 has a Base Score of 5.5, which is classified as MEDIUM severity according to the Common Vulnerability Scoring System (CVSS).

Which versions The versions of Android affected by CVE-2022-20467 are Android 11, Android 12, Android 12L, and Android 13.

What kind To exploit CVE-2022-20467, a malicious actor would require some form of user interaction, though the exact details are not disclosed.

When was CVE-2022-20467 was publicly disclosed on 24 March 2023.

Where can Additional details about CVE-2022-20467 can be found in the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-02>.

What is a possible attack scenario An attack scenario for CVE-2022-20467 could involve a malicious application that takes advantage of the parsing vulnerability.

Can you provide Providing a specific code example for CVE-2022-20467 would require detailed knowledge of the affected system's internal components.

What is CVE-2023-22702 is a Cross-Site Scripting (XSS) vulnerability found in the `WPMobile.App` plugin for WordPress.

How severe CVE-2023-22702 has been given a Base Score of 5.4, which is categorized as MEDIUM severity. This means it could allow an attacker to execute arbitrary scripts in the context of the application.

What versions CVE-2023-22702 affects the `WPMobile.App` — Android and iOS Mobile Application plugin for WordPress.

When was CVE-2023-22702 was published on 23 March 2023.

Where can More information about CVE-2023-22702 can be found on the Patchstack database at the following URL: <https://patchstack.com/database/vulnerability/cve-2023-22702>.

What is the A Cross-Site Scripting (XSS) vulnerability, like CVE-2023-22702, allows an attacker to inject and execute arbitrary scripts within the application.

Can you provide One possible attack scenario for CVE-2023-22702 could involve an attacker who has contributor-level access to the application.

What is CVE-2022-45634 is a security vulnerability that was discovered in the `MEGA FEIS BOFEI DBD+ Application`.

How severe The vulnerability CVE-2022-45634 has been given a base score of 4.3, which corresponds to a MEDIUM severity.

When was CVE-2022-45634 was published on 22 March 2023.

What versions The vulnerability CVE-2022-45634 affects the `MEGA FEIS BOFEI DBD+ Application` for iOS & Android versions.

Where can More information about CVE-2022-45634 can be found in the advisories on the following websites: <https://www.megafeis.com/advisories/cve-2022-45634>.

Could you provide In a possible attack scenario for CVE-2022-45634, an attacker who has already authenticated with the application could exploit the vulnerability.

Are there any code examples As CVE-2022-45634 pertains to a sensitive information disclosure vulnerability, it's not common practice to publish code examples.

What is CVE-2022-45636 refers to a security vulnerability found in the `MEGA FEIS, BOFEI DBD+ Application` for iOS & Android.

How severe The vulnerability identified by CVE-2022-45636 has been given a base score of 8.1, which is categorized as CRITICAL severity.

When was CVE-2022-45636 was published on 21 March 2023.

Where can More details about CVE-2022-45636 can be found on the following references:- WithSecure Labs advisory: <https://www.withsecure.com/advisories/cve-2022-45636>.

Can you provide An attack scenario involving CVE-2022-45636 would include an attacker identifying the API endpoint that is vulnerable.

What kind CVE-2022-45636 can allow an unauthorized attacker to unlock models within the `MEGA FEIS, BOFEI DBD+ Application`.

What is the CVE ID for the insecure password reset issue in the `MEGA FEIS, BOFEI DBD+ Application` for iOS & Android is CVE-2022-45637.

What is the CVE-2022-45637 describes an insecure password reset issue within `MEGA FEIS, BOFEI DBD+ Application`.

How severe The issue CVE-2022-45637 has been assigned a base score of 9.8, which categorizes it as CRITICAL in severity.



When was The issue identified by CVE-2022-45637 was published on 21 March 2023.

Where can Additional information regarding CVE-2022-45637 can be found on GitHub at the following URL: <https://github.com/MEGAFEIS/BOFEI-DBD-plus>

What versi CVE-2022-45637 affects the MEGAFEIS, BOFEI DBD+ Application for IOS & Android version 1.4.4.

Can you de A possible attack scenario for CVE-2022-45637 would involve an attacker exploiting the insecure expiration date

What type: To mitigate the issue described in CVE-2022-45637, the application developers should ensure that password is not stored in plain text

What is CV CVE-2022-45635 refers to a security vulnerability found in the MEGAFEIS BOFEI DBD+ Application for IOS & Android version 1.4.4.

When was CVE-2022-45635 was published on 21 March 2023.

What is the CVE-2022-45635 has been assigned a CVSS base score of 7.5, indicating it is considered a HIGH severity vulnerability

Where can More information about CVE-2022-45635 can be found on the GitHub page: <https://github.com/MEGAFEIS/BOFEI-DBD-plus>

Can you pr An example exploitation scenario for CVE-2022-45635 involves an attacker making use of the insecure expiration date

What mea: To mitigate the impact of CVE-2022-45635, it is recommended that users of MEGAFEIS BOFEI DBD+ Application for IOS & Android version 1.4.4.

What is the CVE ID associated with the vulnerability in the Samsung Calendar is CVE-2023-21464.

What type CVE-2023-21464 refers to an improper access control issue in the Samsung Calendar.

Which vers Samsung Calendar versions prior to 12.4.02.9000 in Android 13 and 12.3.08.2000 in Android 12 are affected

What is the CVE-2023-21464 has been assigned a CVSS base score of 3.3, which categorizes it as a LOW level vulnerability

When was CVE-2023-21464 was published on 16 March 2023.

Where can More information or official advisories about CVE-2023-21464 can be found at the following URL: <https://www.samsung.com/global/mobility/updates/20230316>

What kind An attack exploiting CVE-2023-21464 would involve a local attacker manipulating the Samsung Calendar app

What is CV CVE-2023-21463 refers to an improper access control vulnerability found in the MyFiles application on Samsung devices

How sever The vulnerability described by CVE-2023-21463 is rated with a base score of 3.3, which classifies it as a LOW severity vulnerability

On what date CVE-2023-21463 was published on 16 March 2023.

Are there a Yes, more information on CVE-2023-21463 can be found at the Samsung Mobile Security website through the following URL: <https://www.samsung.com/global/mobility/updates/20230316>

What versi CVE-2023-21463 affects devices running Android versions prior to 12.2.09.0 in Android 11, 13.1.03.50 in Android 13

Can you pr A possible attack scenario for CVE-2023-21463 may involve a malicious local application or a user with physical access to the device

Are there a As CVE-2023-21463 is a vulnerability related to improper access control and not a specific code execution vulnerability

What is CV CVE-2023-21462 refers to a sensitive information exposure vulnerability found in Quick Share Agent on Samsung devices

How sever The vulnerability described by CVE-2023-21462 has a Base Score of 3.3, which categorizes it as a LOW severity vulnerability

When was CVE-2023-21462 was published on 16 March 2023.

Where can Additional information about CVE-2023-21462 can be found at the Samsung Mobile Security webpage through the following URL: <https://www.samsung.com/global/mobility/updates/20230316>

What are p Attack scenarios for CVE-2023-21462 include a malicious app or an attacker with physical access to the device

What is CV CVE-2023-24923 refers to a security vulnerability identified in Microsoft OneDrive for Android that involves an information disclosure

What type CVE-2023-24923 is an information disclosure vulnerability. This type of vulnerability means that it may allow an attacker to access sensitive information

When was CVE-2023-24923 was published on 14 March 2023.

How sever The CVE-2023-24923 vulnerability has a severity rating of 5.5, which is classified as MEDIUM according to the CVSS

Where can More details about CVE-2023-24923 can be found on the Microsoft Security Response Center (MSRC) website through the following URL: <https://msrc.microsoft.com/updateguides/index>

What are t While specific details of an attack scenario for CVE-2023-24923 are not provided, generally, an information disclosure vulnerability allows an attacker to access sensitive information

What is the CVE ID for the Microsoft OneDrive for Android information disclosure vulnerability is CVE-2023-24923

What type CVE-2023-24882 is related to an information disclosure vulnerability in Microsoft OneDrive for Android

What is the CVE ID for the Microsoft OneDrive for Android information disclosure vulnerability is CVE-2023-24882

On what date CVE-2023-24882 was published on 14 March 2023.

Where can More information about CVE-2023-24882 can be found on the Microsoft Security Response Center (MSRC) website through the following URL: <https://msrc.microsoft.com/updateguides/index>

Can you pr While the specific details of the CVE-2023-24882 vulnerability are not provided, a possible attack scenario involves an attacker accessing sensitive information

What mea: To mitigate the impact of CVE-2023-24882, users should ensure that their Microsoft OneDrive for Android app is updated to the latest version

What is the CVE ID for the Office for Android Spoofing Vulnerability is CVE-2023-23391.

How has the Office for Android Spoofing Vulnerability has been classified with a Base Score of 5.5, which is categorized as a MEDIUM severity vulnerability

When was CVE-2023-23391 was published on 14 March 2023.

Where can More information about CVE-2023-23391 can be found at the Microsoft Security Response Center (MSRC) website through the following URL: <https://msrc.microsoft.com/updateguides/index>

What kind CVE-2023-23391 is associated with a spoofing vulnerability in Office for Android.

Could you | An example of a possible attack scenario for CVE-2023-23391 could involve an attacker creating a mal

What is CV CVE-2023-27895 refers to a security vulnerability found in SAP Authenticator for Android, specifically

How does | The impact of CVE-2023-27895 on the security of SAP Authenticator for Android is that it permits an a

What is the CVSS Base Score assigned to CVE-2023-27895 is 6.5, which categorizes it as a medium-severity vu

When was CVE-2023-27895 was published on 14 March 2023.

Where can More information about CVE-2023-27895 can be found in the SAP Knowledge Base Article at <https://l>

Can you pr A potential attack scenario for CVE-2023-27895 could involve an attacker first persuading a user to ins

Could you | Due to CVE-2023-27895, sensitive information at risk includes the OTP (one-time password) being cur

What is the CVE ID associated with the memory corruption vulnerability in Automotive Android OS is CVE-202

Can you de The vulnerability identified by CVE-2022-40539 involves memory corruption in the Automotive Andro

What is the CVE-2022-40539 has a Base Score of 7.8, which is categorized as HIGH severity. This implies that the v

When was The CVE-2022-40539 vulnerability was published on 10 March 2023.

Where can More information about the CVE-2022-40539 vulnerability can be found on Qualcomm's website, spe

What might A possible attack scenario exploiting CVE-2022-40539 would involve an attacker crafting a malicious in

What is the CVE ID for the vulnerability affecting Google Chrome on Android is CVE-2023-1234.

Can you de CVE-2023-1234 refers to an inappropriate implementation in the Intents mechanism of Google Chrom

What versi CVE-2023-1234 affects versions of Google Chrome on Android prior to 111.0.5563.64.

How sever The security issue identified by CVE-2023-1234 has been assigned a severity rating of 'Low' by the Chr

When was CVE-2023-1234 was published on 07 March 2023.

Where can Further information about CVE-2023-1234 can be found at the following references: - The Chromium i

What is a p A possible attack scenario for the vulnerability in CVE-2023-1234 involves a remote attacker creating ;

What is the CVE ID for the vulnerability associated with Google Chrome's Autofill feature on Android is CVE-2

What kind CVE-2023-1231 describes an inappropriate implementation in the Autofill feature of Google Chrome c

What is the severity level given to CVE-2023-1231 by the Chromium security team is Medium, and it has a bas

Can you pr Yes, more information on CVE-2023-1231 can be found at the following references:- <https://chromerele>

Which vers The issue described by CVE-2023-1231 was addressed in Google Chrome for Android version 111.0.55

On what d CVE-2023-1231 was published on 07 March 2023.

What pote By exploiting the vulnerability in CVE-2023-1231, a remote attacker could potentially spoof the conter

Could you | In the context of Google Chrome, the 'omnibox' is the browser's address bar which also serves as a se

Can you pr An attacker would need to craft a malicious HTML page that takes advantage of the improper implem

What is CV CVE-2023-1230 is a security vulnerability identified in Google Chrome on Android. It is related to an ir

What versi Google Chrome on Android versions prior to 111.0.5563.64 are affected by CVE-2023-1230.

What is the CVE-2023-1230 has been assigned a severity level of 'Medium' with a base score of 4.3 according to th

On which c CVE-2023-1230 was published on 07 March 2023.

Where can More information about CVE-2023-1230 can be found at the following references:- <https://crbug.com>,

Can you pr A possible attack scenario for CVE-2023-1230 would involve an attacker creating a malicious website ;

What shou To mitigate the risk posed by CVE-2023-1230, users should update their Google Chrome browser on A

What is CV CVE-2023-1228 is a security vulnerability identified in Google Chrome for Android prior to version 111

What is the severity level of CVE-2023-1228 is classified as 'Medium', with a base score of 4.3.

How can a A remote attacker can exploit CVE-2023-1228 by creating a malicious HTML page that, when visited b

When was CVE-2023-1228 was published on 07 March 2023.

Are there e Yes, more information about CVE-2023-1228 can be found on the following references:- Google Chron

Has CVE-2( Yes, Google has addressed CVE-2023-1228 in version 111.0.5563.64 of Google Chrome on Android. Us

Can you pr In an attack scenario involving CVE-2023-1228, a user on an affected Android device could visit a web;

What is the In the context of CVE-2023-1228, 'Intents' refer to a messaging component in the Android operating s

What is CV CVE-2023-1223 is a security vulnerability identified in the Autofill feature of Google Chrome on Andro

What component does CVE-2023-1223 affect in Google Chrome on Android devices?

What is the severity level of CVE-2023-1223 is rated as 'Medium' with a base score of 4.3.

Has CVE-2023-1223 been addressed in an update for Google Chrome. Users should update to version 115.0.5941.106 or later.

When was CVE-2023-1223 was publicly disclosed on 07 March 2023.

Where can More information about CVE-2023-1223 can be found in the references provided, such as the Google

What are s In an attack scenario exploiting CVE-2023-1223, a remote attacker could create a malicious HTML pag

Could you | As a responsible entity, I cannot provide specific code examples that facilitate the exploitation of vuln

What is the CVE ID for the reported vulnerability in the hgzojer Vocab Trainer application is CVE-2017-2018:

Which cor The critical vulnerability referenced by CVE-2017-20181 is found in the Vocab Trainer application up

What is the CVE-2017-20181 describes a security flaw that leads to path traversal. This means that an attacker could

How can the vulnerability CVE-2017-20181 be mitigated by upgrading the hgzoj Vocab Trainer application?

Is there an Yes, there is a patch for CVE-2017-20181. It is identified by the commit hash accf6838078f8eb105cfc7:

What is the CVE-2017-20181 has been assigned a CVSS base score of 5.5, which categorizes it as MEDIUM severity

What are t To exploit the vulnerability described by CVE-2017-20181, an attacker must have local access to the d

Can you gain? Since CVE-2017-20181 allows for path traversal, an attacker with local access could manipulate file paths to access sensitive data or execute arbitrary code.

When was The vulnerability with CVE ID CVE-2017-20181 was published on 07 March 2023.

Where can You can find the patch for CVE-2017-20181 in the GitHub commit referenced in the CVE details (comm

What is the CVE ID for the vulnerability involving a heap buffer overflow in AAVCAssembler.cpp is CVE-2023-2

What are the primary consequences of the CVE-2023-20948 vulnerability? The primary consequence of the CVE-2023-20948 vulnerability is a possible out of bounds read due to

Which vers The Android versions affected by CVE-2023-20948 are Android 12, Android 12L, and Android 13.

Is user inte No, user interaction is not required to exploit the CVE-2023-20948 vulnerability.

What is the CVE-2023-20948 has been assigned a CVSS Base Score of 7.5, which is categorized as HIGH.

When was CVE-2023-20948 was publicly disclosed on 28 February 2023.

Where can More information about CVE-2023-20948 can be found on the Android Security Bulletin page at <https://source.android.com/security/bulletin/2023-09>

What is the Android ID associated with CVE-2023-20948 is A-230630526.

Could you | An example of a potential attack scenario involving CVE-2023-20948 could involve a malicious actor cr

**What steps:** To mitigate the impact of CVE-2023-20948, users should ensure their Android devices are updated with the latest security patches.

What is CV CVE-2023-20946 refers to a security vulnerability found in the BluetoothSwitchPreferenceController.j

How severe CVE-2023-20946 has been given a base score of 9.8, which classifies it as CRITICAL in severity. Such a l

Which vers CVE-2023-20946 affects Android versions 11, 12, 12L, and 13. Devices running these versions of Andro

Does CVE-2 No, CVE-2023-20946 can be exploited without requiring any user interaction. This increases the poten

Where can More details about CVE-2023-20946 can be found in the Android Security Bulletin at the following UR

What are the Attack scenarios for CVE-2023-20946 could include an attacker leveraging the permission bypass vuln

When was CVE-2023-20946 was published on 28 February 2023, when it was included in an Android security bull

What is the CVE ID for this vulnerability is CVE-2023-20945.

Can you ex CVE-2023-20945 refers to a vulnerability in the phNciNfc\_MfCreateXchgDataHdr function of phNxpEx

What type The vulnerability reported in CVE-2023-20945 is a type of out of bounds write, which could lead to loc

What kind Through the exploitation of CVE-2023-20945, local escalation of privilege is possible, meaning a local :

Is user inte No, user interaction is not required to exploit the vulnerability mentioned in CVE-2023-20945.

Which And CVE-2023-20945 affects Android version 10.

How severe: The vulnerability CVE-2023-20945 has been given a Base Score of 7.8, which classifies it as HIGH in severity.

When was CVE-2023-20945 was published on 28 February 2023.

Where can More information about CVE-2023-20945 can be found in the Android Security Bulletin linked here: [https://source.android.com/security/bulletin/2023-09-01](#)

Can you de An attack scenario involving CVE-2023-20945 would typically involve a malicious application executed

What is the CVE ID for the vulnerability involving unsafe deserialization in ChooseTypeAndAccountActivity.java?

What type CVE-2023-20944 identifies a possible escalation of privilege vulnerability due to unsafe deserialization

Which vers The vulnerability CVE-2023-20944 affects the following versions of Android: Android 10, Android 11, /

What is the CVSS base score assigned to CVE-2023-20944 is 7.8 HIGH. This implies that the vulnerability poses a high risk to the system. To exploit the vulnerability CVE-2023-20944, an attacker does not need any additional execution privileges. More information about the CVE-2023-20944 vulnerability can be found on the Android Security Bulletin. CVE-2023-20944 can lead to a local escalation of privilege on an Android device. An attacker could exploit this vulnerability to gain root access to the device. An attack scenario exploiting CVE-2023-20944 could involve an attacker creating a malicious application that triggers the vulnerability. Yes, the specific Android ID associated with CVE-2023-20944 is A-244154558.

To mitigate the risk posed by CVE-2023-20944, users should ensure that their device's firmware is up to date. The CVE ID for the vulnerability involving a path traversal error in ActivityManagerService.java is CVE-2023-20943. The Android versions affected by CVE-2023-20943 include Android-10, Android-11, Android-12, and Android-13. CVE-2023-20943 is a vulnerability in the clearApplicationUserData function of ActivityManagerService. The CVSS base score for CVE-2023-20943 is 7.8, which is categorized as HIGH severity. CVE-2023-20943 was published on 28 February 2023.

More information about CVE-2023-20943 can be found in the Android Security Bulletin at the following link: [https://source.android.com/security/bulletin/2023-02/03](#). To exploit CVE-2023-20943, an attacker would need User execution privileges. No, user interaction is not needed to exploit CVE-2023-20943.

In an attack scenario for CVE-2023-20943, an attacker with execution privileges on an affected Android device could trigger the vulnerability by providing a specially crafted input path. To prevent vulnerabilities like CVE-2023-20943, developers should validate and sanitize all input paths. CVE-2023-20940 refers to a vulnerability in the Android operating system where there is a potential for local escalation of privilege. Android version 13 is affected by CVE-2023-20940.

CVE-2023-20940 can lead to local escalation of privilege on the Android operating system. CVE-2023-20940 has been rated with a base score of 7.8, which is classified as HIGH severity. The official source of information for CVE-2023-20940 can be found at the Android security bulletin website: [https://source.android.com/security/bulletin/2023-02/04](#). CVE-2023-20940 was published on 28 February 2023. No, user interaction is not required to exploit CVE-2023-20940.

An example attack scenario for CVE-2023-20940 could involve an attacker with physical access to an Android device exploiting the vulnerability to gain root access. To mitigate CVE-2023-20940, it is recommended that users update their Android devices to the latest version. CVE-2023-20939 refers to a security vulnerability found in an Android component named looper\_back. The vulnerability CVE-2023-20939 has been assessed with a Base Score of 7.8, categorizing it as HIGH severity. No, user interaction is not needed for exploitation of CVE-2023-20939, which means the vulnerability can be exploited without user interaction. CVE-2023-20939 affects Android versions Android-12, Android-12L, and Android-13. Devices running these versions are vulnerable. By exploiting CVE-2023-20939, an attacker has the potential to corrupt memory due to improper locking. CVE-2023-20939 was published on 28 February 2023.

Yes, you can learn more about CVE-2023-20939 by visiting the provided reference in the Android Security Bulletin: [https://source.android.com/security/bulletin/2023-02/05](#). An attack scenario could involve a malicious app installed on the Android device that exploits the vulnerability. CVE-2023-20938 is a security vulnerability that exists in the binder\_transaction\_buffer\_release function. CVE-2023-20938 has a base score of 7.8, which is classified as HIGH severity. This suggests that the vulnerability is high severity. No, exploiting CVE-2023-20938 does not require user interaction. An attacker could potentially leverage this vulnerability to cause a local escalation of privilege. CVE-2023-20938 was published on 28 February 2023.

CVE-2023-20938 affects the Android kernel, but the specific versions impacted by this vulnerability are not specified. More information about CVE-2023-20938 can be found in the Android Security Bulletin at the following link: [https://source.android.com/security/bulletin/2023-02/06](#). As I don't have access to the specific code, I can't provide an exact code snippet. However, a user could trigger the vulnerability by exploiting CVE-2023-20938, an attacker could potentially cause a local escalation of privilege on the device. The Android ID associated with CVE-2023-20938 is A-257685302.

CVE-2023-20937 is a security vulnerability identified in several functions of the Android Linux kernel. CVE-2023-20937 has been assigned a base score of 7.8, which is classified as HIGH severity. This indicates a high risk to the system. No, CVE-2023-20937 does not require user interaction for exploitation, which means it can be exploited without user interaction. CVE-2023-20937 affects the Android operating system, specifically the versions that include vulnerable kernel versions.

What are t The exploitation of CVE-2023-20937 does not require additional execution privileges. This means that

What is the Exploiting CVE-2023-20937 could lead to local escalation of privilege on the affected system. An attack

Where can Additional information about CVE-2023-20937 can be found on the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-02-01>.

Are there c As CVE-2023-20937 involves a use-after-free vulnerability, specific code examples demonstrating the

What are s A possible attack scenario for CVE-2023-20937 could involve an attacker who already has access to the

Has CVE-2023-20937 Yes, CVE-2023-20937 has been referenced in the Android Security Bulletin published on February 28, 2023.

What is the The CVE ID is CVE-2023-20934, which refers to a permissions bypass vulnerability that can disable the

Which vers CVE-2023-20934 affects the following Android versions: Android 12, Android 12L, and Android 13.

What is the CVE-2023-20934 has been assigned a Base Score of 7.8 and classified as HIGH in severity.

What type CVE-2023-20934 is classified as a permissions bypass vulnerability that could potentially result in local

When was CVE-2023-20934 was published on 28 February 2023.

Where can More details or the official advisory about CVE-2023-20934 can be found at the Android Security Bulletin

Does explo No, exploiting CVE-2023-20934 does not require any user interaction, which increases the vulnerability

Can you pr An attack scenario for CVE-2023-20934 could involve a malicious app that exploits the vulnerability to

Are there c Code examples for specific vulnerabilities like CVE-2023-20934 are often not published to avoid enabling

What is the The CVE ID of the vulnerability is CVE-2023-20933.

Can you de CVE-2023-20933 refers to a vulnerability in various functions of MediaCodec.cpp where there is a potential

Which And The affected Android versions by CVE-2023-20933 include Android-10, Android-11, Android-12, and Android-13.

According No, user interaction is not required to exploit the vulnerability described in CVE-2023-20933.

What is the CVE-2023-20933 has been rated with a base score of 7.8 and classified as HIGH severity.

When was CVE-2023-20933 was published on 28 February 2023.

Where can Detailed information about CVE-2023-20933 can be found on the Android Security Bulletin website at

What type: An attacker could exploit CVE-2023-20933 to cause memory corruption through a use-after-free error

Are there c It would be considered unethical and potentially illegal to provide or distribute exploit code for CVE-2023-20933.

What is the The CVE ID of the vulnerability is CVE-2023-20932.

What is the CVE-2023-20932 has been given a base score of 3.3, which classifies it as a LOW severity vulnerability.

In which file The vulnerability CVE-2023-20932 was discovered in the EditInfoFragment.java file.

Which vers The Android versions affected by CVE-2023-20932 are Android-10, Android-11, Android-12, Android-12L, and Android-13.

What kind CVE-2023-20932 is a vulnerability due to improper input validation which could lead to local information disclosure.

Is user inte No, user interaction is not required to exploit the vulnerability described in CVE-2023-20932.

What addit No additional execution privileges are needed to exploit the vulnerability identified by CVE-2023-20932.

Can you pr CVE-2023-20932 was published on 28 February 2023.

Where can More information regarding CVE-2023-20932 can be found at: <https://source.android.com/security/bulletin/2023-02-01>.

What are s Possible attack scenarios for exploiting CVE-2023-20932 involve an attacker using a specially crafted audio file.

What is CV CVE-2022-20551 refers to a vulnerability found in the AudioFlinger.cpp component of Android, where

Which And The Android versions affected by CVE-2022-20551 include Android-12, Android-12L, and Android-13.

What is the The base score assigned to CVE-2022-20551 is 6.7, which is categorized as MEDIUM severity.

When was CVE-2022-20551 was published on 28 February 2023.

Where can More information about CVE-2022-20551 can be found on the Android Security Bulletin page at <https://source.android.com/security/bulletin/2022-02-01>.

What privil To exploit the vulnerability identified as CVE-2022-20551, an attacker would require System execution privileges.

Do users n No, user interaction is not needed for the exploitation of CVE-2022-20551.

What pote The impact of CVE-2022-20551 on an Android device could include unauthorized audio recording with the microphone.

Can you pr An example attack scenario for CVE-2022-20551 could involve a malicious app that has been granted the RECORD\_AUDIO permission.

Has CVE-2022-20551 Yes, CVE-2022-20551 has been addressed in the security updates provided by Android, as mentioned in the Android Security Bulletin.

What is CV CVE-2022-20481 is a security vulnerability identified in multiple versions of Android where there is a potential for local escalation of privilege.

Which And CVE-2022-20481 affects Android 10, Android 11, Android 12, Android 12L, and Android 13.

What is the The CVSS base score for CVE-2022-20481 is 5.5, and it is classified as MEDIUM severity.

How can CVE-2022-20481 can be exploited by a local attacker who takes advantage of the residual data left after the application is uninstalled. On what date CVE-2022-20481 was published on 28 February 2023.

Where can Additional information about CVE-2022-20481 can be found in the Android security bulletin at the following link: <https://source.android.com/security/bulletin/2023-02-01>

What is the impact The impact of CVE-2022-20481 is local information disclosure. An attacker with physical access to a device can retrieve residual data left after the application is uninstalled.

Are there any code examples As CVE-2022-20481 relates to residual data preservation, code examples would not be appropriate. It is recommended to review the security bulletin for more details.

Could you provide a possible attack scenario for CVE-2022-20481 involves an attacker gaining physical access to a device and retrieving residual data left after the application is uninstalled.

What is the CVE ID for the vulnerability related to ZenModeHelper.java is CVE-2022-20455.

What is the impact The vulnerability mentioned in CVE-2022-20455 could lead to a persistent local denial of service due to a buffer overflow in the ZenModeHelper.java file.

Which Android versions affected by CVE-2022-20455 are Android 10, Android 11, Android 12, Android 13.

Does the exploit require user interaction No, the exploitation of CVE-2022-20455 does not require user interaction.

What is the severity CVE-2022-20455 has been assigned a base score of 5.5 and is considered to be of MEDIUM severity level.

When was CVE-2022-20455 was published on 28 February 2023.

Where can you find more information about CVE-2022-20455 on the Android Security Bulletin website at <https://source.android.com/security/bulletin/2023-02-01>

What additional execution privileges are needed to exploit the vulnerability mentioned in CVE-2022-20455 No additional execution privileges are needed to exploit the vulnerability mentioned in CVE-2022-20455.

Can you provide an attack scenario for CVE-2022-20455 might involve a malicious application that exploits the vulnerability to cause a persistent local denial of service.

Are there any code examples As CVE-2022-20455 is a vulnerability found in ZenModeHelper.java, a part of the Android operating system, code examples would not be appropriate.

What is the issue associated with the need to use HTTPS to protect network information is identified by the CVE-2022-32906.

In which Apple version The vulnerability identified as CVE-2022-32906 was addressed in Apple Music version 3.9.10 for Android.

What is the severity CVE-2022-32906 has been assigned a base score of 5.3, which classifies it as MEDIUM severity.

When was The information about CVE-2022-32906 was officially published on 27 February 2023.

What could happen Due to CVE-2022-32906, a user in a privileged network position could potentially intercept SSL/TLS communications.

Where can More information or the official advisory about CVE-2022-32906 can be found on Apple's support page: <https://support.apple.com/en-us/HT213111>

What type An attack that could exploit CVE-2022-32906 is known as a man-in-the-middle (MITM) attack, where the attacker intercepts and potentially alters the communication between the user and the service.

What mitigation To mitigate CVE-2022-32906, Apple implemented the use of HTTPS, which is a secure communication protocol.

What is the CVE ID of the vulnerability is CVE-2022-32846.

What type CVE-2022-32846 is a logic issue that was addressed with improved state management in Apple Music.

What is the severity The base score severity of CVE-2022-32846 is rated as 7.5, which is considered HIGH.

As of which Apple Music version The vulnerability CVE-2022-32846 is fixed as of Apple Music version 3.9.10 for Android.

What kind Due to CVE-2022-32846, an app might be able to access user-sensitive data.

What might happen Possible attack scenarios for CVE-2022-32846 could include a malicious app exploiting the logic issue to access sensitive data.

Where can More information about CVE-2022-32846 can be found at the official Apple support page: <https://support.apple.com/en-us/HT213111>

When was CVE-2022-32846 was publicly disclosed on 27 February 2023.

What is CVE-2022-32836 refers to a vulnerability that was identified in the Apple Music application version 3.9.10 for Android.

How severe The CVE-2022-32836 vulnerability was given a base score of 7.5, which is categorized as HIGH severity.

When was The CVE-2022-32836 vulnerability was published on 27 February 2023.

Has the vulnerability been addressed Yes, the CVE-2022-32836 vulnerability was addressed and fixed by improving state management in the application.

Where can You can find more information about CVE-2022-32836 through the advisory note provided by Apple at <https://support.apple.com/en-us/HT213111>

What kind An attack scenario for CVE-2022-32836 might include a malicious app or an attacker exploiting the vulnerability to access sensitive data.

What action Users should update their Apple Music application to version 3.9.10 for Android or later as soon as possible to mitigate the risk.

What is the CVE ID of the vulnerability is CVE-2021-46841.

In which Apple version Apple addressed the vulnerability in Apple Music 3.5.0 for Android.

What is the CVSS base score of CVE-2021-46841 is 5.9, which classifies it as a MEDIUM severity vulnerability.

What is the impact The vulnerability CVE-2021-46841 could allow an attacker in a privileged network position to track a user's location.

What date The vulnerability CVE-2021-46841 was published on 27 February 2023.

Where can Official information and support documentation for CVE-2021-46841 can be found at Apple's support page: <https://support.apple.com/en-us/HT213111>

Can you provide a possible attack scenario for CVE-2021-46841 involves an attacker who has the capability to monitor the location of a user's device.

How did Apple mitigate the risk associated with CVE-2021-46841 by updating the Apple Music app to use HTTPS for location tracking.



What is the CVE ID of the vulnerability involving a permissions bypass in the AndroidManifest.xml is CVE-2023-20927.  
What is the vulnerability described in CVE-2023-20927 is a permissions bypass that could lead to local escalation of privilege.  
Which version of Android is affected by CVE-2023-20927 affects Android 13.

How is CVE-2023-20927 can be exploited by an attacker to grant themselves signature permissions without providing a valid signature.  
What is the CVE-2023-20927 has been assigned a CVSS Base Score of 7.8, which is classified as HIGH severity.

What type of vulnerability Exploiting CVE-2023-20927 can result in a local escalation of privilege, allowing an attacker to gain high privileges.  
When was CVE-2023-20927 was published on 15 February 2023.

Where can Details regarding CVE-2023-20927 can be found in the Android Security Bulletin at the following link: [https://source.android.com/security/bulletin/2023-02](#).  
What is the Android ID associated with CVE-2023-20927 is A-244216503.

Can you describe a possible attack scenario for CVE-2023-20927 could involve a malicious app that is installed on the device and then the user grants permissions to the app.

What is CVE-2023-24804 refers to a security vulnerability in the ownCloud Android app that remained after a security update.  
Which app The application affected by CVE-2023-24804 is the ownCloud Android app, which is designed to allow users to sync their files and folders to the cloud.  
What type of vulnerability CVE-2023-24804 is a path traversal vulnerability with two bypass methods that can lead to information disclosure.

What versions Versions of the ownCloud Android app prior to 3.0 are affected by the vulnerability CVE-2023-24804.  
How was CVE-2023-24804 was mitigated by releasing version 3.0 of the ownCloud Android app which contains a security update.

What is the CVSS base score for CVE-2023-24804 is given as 4.4, categorizing it as a MEDIUM severity vulnerability.  
Can you provide potential attack scenarios involving CVE-2023-24804, an attacker might exploit the path traversal vulnerability to access sensitive files.

Where can More information regarding CVE-2023-24804 can be found on several platforms, including the report on the ownCloud website.  
What is CVE-2023-23948 is a security vulnerability identified in the ownCloud Android app, specifically in version 2.21.1.  
Which app CVE-2023-23948 is associated with the ownCloud Android app version 2.21.1.

What are the affected databases In CVE-2023-23948, there are two affected databases: 'filelist' and 'owncloud\_database'.  
Has CVE-2023-23948 been mitigated As of version 3.0 of the ownCloud Android app, the 'filelist' database, which was affected by CVE-2023-23948, has been mitigated.

What is the SQL injection vulnerability identified by CVE-2023-23948 can result in information disclosure, which could lead to unauthorized access to sensitive data.  
Can you provide an attack scenario An attacker could exploit CVE-2023-23948 by crafting malicious input that, when processed by the ownCloud Android app, could lead to information disclosure.  
What is the CVE-2023-23948 has a CVSS base score of 5.5, which is classified as MEDIUM severity.

Where can More information about CVE-2023-23948 can be found in the advisory published by GitHub Security Lab.  
What is the CVE ID for the vulnerability found in the Ichiran App for both iOS and Android is CVE-2023-22367.

What type of vulnerability CVE-2023-22367 describes a security flaw where the Ichiran App for iOS and Android versions prior to 1.0.0 are affected by the vulnerability.  
What is the CVSS base score for CVE-2023-22367 is 5.9, which is categorized as MEDIUM severity. This score is based on the CVSS 3.1 methodology.  
When was The CVE-2023-22367 was published on 13 February 2023.

Where can You can find more information about the CVE-2023-22367 vulnerability in the following resources: the GitHub Security Lab advisory.  
Can you provide an attack scenario An attack scenario exploiting CVE-2023-22367 could involve an attacker performing a man-in-the-middle attack to intercept and modify network traffic.

What versions The Ichiran App versions for both iOS and Android that are affected by CVE-2023-22367 are those prior to version 1.0.0.  
What is CVE-2023-22362 refers to a security vulnerability in the SUSHIRO App for Android where the app outputs sensitive information to the logcat.  
Which versions The versions of the SUSHIRO app affected by CVE-2023-22362 include SUSHIRO Ver.4.0.31 (unspecified version).  
What is the CVE-2023-22362 has been given a CVSS base score of 7.5, which is considered HIGH severity.

When was CVE-2023-22362 was published on 13 February 2023.

Where can More details about CVE-2023-22362 can be found on various URLs such as the Google Play links provided in the advisory.  
Can you describe an attack scenario In an attack scenario involving CVE-2023-22362, an attacker might exploit the vulnerability by accessing the logcat output.

Are there any mitigations While specific code examples for CVE-2023-22362 are not provided, the vulnerability suggests that users should update the app as soon as an update is available to prevent further exploitation.  
What steps Users of the affected SUSHIRO app versions should update the app as soon as an update is available to prevent further exploitation.

How can developers To prevent vulnerabilities like CVE-2023-22362, developers should avoid writing sensitive information to the logcat.  
What is CVE-2023-21451 refers to a security vulnerability identified in SECRIL (Samsung's secure RIL) prior to version 1.0.0.

What type of vulnerability CVE-2023-21451 is classified as a Stack-based buffer overflow vulnerability. Buffer overflows occur when a program writes more data to a buffer than it can hold.  
What is the CVE-2023-21451 has been assigned a severity rating of 7.8, which is categorized as HIGH according to the CVSS 3.1 methodology.  
When was CVE-2023-21451 was published on 09 February 2023.

Where can Further details about CVE-2023-21451 can be found on the Samsung Mobile Security website, specifically in the February 2023 bulletin.



Can you describe an attack scenario for CVE-2023-21451 might involve a malicious application or actor sending crafted messages to the application?

What is the CVE ID of the vulnerability is CVE-2023-21446.

Can you provide details on CVE-2023-21446 is a security vulnerability due to improper input validation in the MyFiles app. It affects Samsung devices running Android R(11) prior to version 12.2.09.

What is the severity of CVE-2023-21446 has been assigned a base score of 5.5, which is classified as MEDIUM severity.

When was CVE-2023-21446 was publicly disclosed on 09 February 2023.

Where can More information about CVE-2023-21446 can be found on the Samsung Mobile security website at the link provided.

What versions CVE-2023-21446 affects the MyFiles app on Android R(11) prior to version 12.2.09, Android S(12) prior to version 13.1.03.501.

Could you describe a potential attack scenario for CVE-2023-21446? Exploiting CVE-2023-21446 would involve a local attacker leveraging the improper input validation vulnerability to gain unauthorized access to the application's data.

Are there any mitigations for CVE-2023-21446? As CVE-2023-21446 is a specific vulnerability related to improper input validation in a proprietary app, users are advised to update the app to the latest version.

What is the CVE ID of the vulnerability identified in the MyFiles application on Android devices. It affects Samsung devices running Android R(11) prior to version 12.2.09.

Which versions CVE-2023-21445 affects MyFiles on Android before versions 12.2.09 on Android R (11), 13.1.03.501 or 13.1.03.501 on Android S(12) prior to version 13.1.03.501.

What is the severity of CVE-2023-21445 The CVSS base score for CVE-2023-21445 is 7.8, which is categorized as HIGH. This indicates that the vulnerability has a significant impact on the confidentiality, integrity, and availability of the system.

When was CVE-2023-21445 was published on 09 February 2023.

Where can More information and updates about CVE-2023-21445 can be found on the Samsung Mobile Security website at the link provided.

Can CVE-2023-21445 be exploited remotely? No, CVE-2023-21445 cannot be exploited remotely. It requires local access to the device as the attack vector is local.

Describe a potential attack scenario for CVE-2023-21445? One potential attack scenario for CVE-2023-21445 would involve an attacker with physical access to the device exploiting the vulnerability to gain unauthorized access to the application's data.

What is the CVE ID of the vulnerability is CVE-2023-21443.

Which app Samsung Flow for Android is affected by CVE-2023-21443.

What is the severity of CVE-2023-21443 CVE-2023-21443 describes an improper cryptographic implementation that could allow adjacent attackers to potentially decrypt encrypted messages or inject malicious data.

What could be the impact of CVE-2023-21443? By exploiting CVE-2023-21443, adjacent attackers could potentially decrypt encrypted messages or inject malicious data into the application.

What is the severity of CVE-2023-21443 The CVSS base score assigned to CVE-2023-21443 is 8.8, which is categorized as HIGH.

What versions of Samsung Flow for Android prior to 4.9.04 are affected by CVE-2023-21443.

When was CVE-2023-21443 was published on 09 February 2023.

Where can More information about CVE-2023-21443 can be found on the Samsung Mobile Security website at the link provided.

Could you describe a possible attack scenario for CVE-2023-21443? A possible attack scenario for CVE-2023-21443 could involve an attacker who is physically close to the device exploiting the vulnerability to gain unauthorized access to the application's data.

Has a fix been issued for CVE-2023-21443? Yes, a fix has been issued for CVE-2023-21443. Users are advised to update their Samsung Flow application to the latest version.

What is the CVE ID of the vulnerability is CVE-2023-21442.

Can you describe the vulnerability referenced by CVE-2023-21442 is an improper access control issue in the Runestone application.

What versions CVE-2023-21442 affects the Runestone application prior to version 2.9.09.003 on Android R (11) and prior to version 3.1.21.10 in Android A(12) prior to version 3.1.21.10.

What is the severity of CVE-2023-21442 The CVSS base score assigned to CVE-2023-21442 is 5.5, which is categorized as MEDIUM severity. This indicates that the vulnerability has a moderate impact on the confidentiality, integrity, and availability of the system.

When was CVE-2023-21442 was published on 09 February 2023.

Where can Additional information or advisories regarding CVE-2023-21442 can be found at <https://security.samsungmobile.com>.

Could you describe a local attack scenario for CVE-2023-21442? A local attacker could exploit CVE-2023-21442 by manipulating the Runestone application's access control to gain unauthorized access to the application's data.

What mitigation steps can be taken to address CVE-2023-21442? To mitigate the risk posed by CVE-2023-21442, users should update their Runestone application to the latest version.

What is the CVE ID of the vulnerability titled 'Insufficient Verification of Data Authenticity' in the Routine application.

What is the severity of CVE-2023-21441 CVE-2023-21441 allows a local attacker to exploit insufficient data authenticity checks to access protected data.

How severe is CVE-2023-21441 CVE-2023-21441 has been assigned a base score of 5.5, which is categorized as MEDIUM severity. This indicates that the vulnerability has a moderate impact on the confidentiality, integrity, and availability of the system.

On what date was CVE-2023-21441 was published on 09 February 2023, at which time details about the vulnerability became available.

Where can More information about CVE-2023-21441 can be found on the Samsung Mobile Security website, via the link provided.

What versions CVE-2023-21441 affects the Routine feature prior to versions 2.6.30.6 in Android Q(10), 3.1.21.10 in Android A(12) prior to version 3.1.21.10.

What might be the impact of CVE-2023-21441? An attack scenario for CVE-2023-21441 could involve a local attacker with access to a vulnerable device exploiting the vulnerability to gain unauthorized access to the application's data.

Is there a fix available for CVE-2023-21441? Unfortunately, as the details surrounding CVE-2023-21441 focus on the description of the vulnerability, a fix has not been issued yet.

What is the CVE ID for the vulnerability concerning inappropriate implementation in full-screen mode in Google Chrome.

What versions CVE-2023-0697 affects Google Chrome on Android prior to version 110.0.5481.77.

How did CVE-2023-0697 allowed a remote attacker to spoof the contents of the security UI in Google Chrome.

What is the severity of CVE-2023-0697 The severity level assigned to CVE-2023-0697 is 'High' according to the Chromium security team.

What is the severity of CVE-2023-0697 CVE-2023-0697 has been assigned a CVSS base score of 6.5, categorizing it as 'Medium'.

Can you pr CVE-2023-0697 was published on 07 February 2023.

What are s More information about CVE-2023-0697 can be found at the following references: - <https://chromerel>

Could you | A possible attack scenario for CVE-2023-0697 involves a remote attacker creating a specially crafted H

What is CV CVE-2022-39380 refers to a security vulnerability in the Wire web-app where certain Markdown form

Which vers Versions of the Wire web-app prior to the 2022-11-02 update are affected by the CVE-2022-39380 vul

How can C' To mitigate CVE-2022-39380, Wire managed services have already deployed a fix in version 2022-11-0

What is the Base Score for CVE-2022-39380 is 5.3, which is categorized as MEDIUM severity.

When was CVE-2022-39380 was published on 27 January 2023.

Where can More information about CVE-2022-39380 can be found in the GitHub security advisory at the followin

Can you gi An attack scenario for CVE-2022-39380 could involve an attacker deliberately sending a message with

What is the CVE ID of the vulnerability is CVE-2023-20928.

What are t CVE-2023-20928 leads to a possible use after free due to improper locking, which could result in a loc

What is the base score assigned to CVE-2023-20928 is 7.8, which is considered HIGH.

When was CVE-2023-20928 was published on 26 January 2023.

Do users n No, user interaction is not needed to exploit the vulnerability described in CVE-2023-20928.

Which vers The versions of Android affected by CVE-2023-20928 are not specified beyond being related to the An

Can you pr Yes, more information about CVE-2023-20928 is available in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-01>

What are p Possible attack scenarios for CVE-2023-20928 include an attacker exploiting the use after free vulnera

What is the CVE ID for the vulnerability is CVE-2023-20925.

Can you de CVE-2023-20925 refers to a use after free vulnerability found in setUclampMinLocked of PowerSessio

What versi The vulnerability with CVE-2023-20925 affects Android kernel versions, however, the exact affected v

What is the CVE-2023-20925 has been assigned a base score of 7.8, which is classified as HIGH severity.

When was CVE-2023-20925 was published on 26 January 2023.

Where can More information about CVE-2023-20925 can be found at the following URL: <https://source.android.com/security/bulletin/2023-01>

Are there a No, user interaction is not required to exploit the vulnerability CVE-2023-20925.

What could By exploiting CVE-2023-20925, an attacker could corrupt memory and potentially achieve local escala

What is the Android ID associated with CVE-2023-20925 is A-236674672.

Could you | A possible attack scenario for CVE-2023-20925 involves an attacker writing a malicious app or code th

What is the CVE ID for the vulnerability that involves potential lockscreen bypass due to Biometric Auth Failur

Can you pr CVE-2023-20924 is a security vulnerability that exists in the Android operating system where there is

When was CVE-2023-20924 was published on 26 January 2023.

What is the base score assigned to CVE-2023-20924 is 6.8, categorizing it as MEDIUM severity.

Which vers The vulnerability CVE-2023-20924 affects the Android kernel, but the specific versions impacted by th

How can t The vulnerability identified by CVE-2023-20924 can be exploited by an attacker with physical access to

Are there a Details about specific software updates that address CVE-2023-20924 are not provided within the give

What type No user interaction is required to exploit the vulnerability CVE-2023-20924. An attacker can potentiall

Is there an The provided information does not indicate the availability of public exploit code for CVE-2023-20924

What are p Possible attack scenarios for CVE-2023-20924 include an attacker obtaining physical access to an Andr

What is the CVE ID of the vulnerability involving a permissions bypass in ShannonRcs exported content provid

Can you ex The vulnerability CVE-2023-20923 describes a scenario where there is a permissions bypass in exporte

What is the CVE-2023-20923 has a CVSS base score of 5.5, which is rated as MEDIUM severity.

When was The vulnerability CVE-2023-20923 was published on 26 January 2023.

Is user inte No, user interaction is not required to exploit the CVE-2023-20923 vulnerability.

Which vers The affected versions of Android due to the CVE-2023-20923 vulnerability are specified as 'Android ke

Where can More information about CVE-2023-20923 can be found at the following URL: <https://source.android.com/security/bulletin/2023-01>

Could you | An attack scenario for CVE-2023-20923 could involve a malicious app that exploits the permissions by

What Andr The Android ID associated with CVE-2023-20923 is A-246933910.

What is the CVE ID for the reported vulnerability causing a possible crash loop due to resource exhaustion in `setMimeGroup` function of `PackageManagerService.java`? The CVE ID for the reported vulnerability is CVE-2023-20922.

Can you describe CVE-2023-20922? CVE-2023-20922 refers to a vulnerability in the `setMimeGroup` function of `PackageManagerService.java`.

What versions of Android are affected by CVE-2023-20922? The versions of Android affected by CVE-2023-20922 include Android-11, Android-12, Android-12L, and Android-13.

What is the severity of CVE-2023-20922? CVE-2023-20922 has been assigned a Base Score of 5.5, which is categorized as MEDIUM severity.

When was CVE-2023-20922 published? The vulnerability with CVE ID CVE-2023-20922 was published on 26 January 2023.

Where can I find more information about CVE-2023-20922? More detailed information about CVE-2023-20922 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-01>.

Are there any specific examples or proof-of-concept codes given for CVE-2023-20922? As of the provided information, there are no specific examples or proof-of-concept codes given for CVE-2023-20922.

What are the potential attack scenarios for CVE-2023-20922? For CVE-2023-20922, a potential attack scenario involves a malicious application or script exploiting the `setMimeGroup` function to cause a crash loop.

What is the CVE ID of the vulnerability in `AccessibilityManagerService.java`? The CVE ID of the vulnerability is CVE-2023-20921.

What is the severity of CVE-2023-20921? The base score of CVE-2023-20921 is 7.3, which is categorized as HIGH.

On what date was CVE-2023-20921 published? CVE-2023-20921 was published on 26 January 2023.

What versions of Android are affected by CVE-2023-20921? The versions of Android affected by CVE-2023-20921 include Android 10, Android 11, Android 12, and Android 13.

Can you provide more information about CVE-2023-20921? More information about CVE-2023-20921 can be found at the Android Security Bulletin: <https://source.android.com/security/bulletin/2023-01>.

What type of vulnerability is CVE-2023-20921? CVE-2023-20921 is an escalation of privilege vulnerability located in `AccessibilityManagerService.java`.

Is user interaction needed to exploit CVE-2023-20921? Yes, user interaction is needed to exploit CVE-2023-20921.

What is the Android ID associated with CVE-2023-20921? The Android ID associated with CVE-2023-20921 is A-243378132.

Explain a potential attack scenario for CVE-2023-20921. An attack scenario for CVE-2023-20921 could involve a malicious application that tricks the user into granting accessibility permissions, which can then be used to escalate privileges.

What is the CVE ID for the vulnerability in `AccessibilityManagerService.java`? The CVE ID for this vulnerability is CVE-2023-20920.

What versions of Android are affected by CVE-2023-20920? Android versions affected by CVE-2023-20920 are Android-10, Android-11, Android-12, Android-12L, and Android-13.

What is the severity of CVE-2023-20920? CVE-2023-20920 has been assigned a base score of 7.8, which is categorized as HIGH.

What type of vulnerability is CVE-2023-20920? CVE-2023-20920 describes a possible memory corruption issue due to a use after free in the queue of `AccessibilityManagerService.java`.

Is user interaction needed for the exploitation of CVE-2023-20920? No, user interaction is not needed for the exploitation of CVE-2023-20920.

When was CVE-2023-20920 published? CVE-2023-20920 was published on 26 January 2023.

Where can I find more information about CVE-2023-20920? More information about CVE-2023-20920 can be found at the following URL: <https://source.android.com/security/bulletin/2023-01>.

Could you describe a potential attack scenario for CVE-2023-20920? An attack scenario for CVE-2023-20920 could involve a malicious application exploiting the use after free vulnerability to cause a crash.

What steps can be taken to mitigate the risks associated with CVE-2023-20920? To mitigate the risks associated with CVE-2023-20920, users should apply patches and updates provided by the manufacturer.

What is the Android ID assigned to CVE-2023-20920? The Android ID assigned to CVE-2023-20920 is A-204584366.

What is CVE-2023-20919? CVE-2023-20919 refers to a security vulnerability affecting the `getStringsForPrefix` method of `Settings`.

What are the potential impacts of CVE-2023-20919? Exploiting the CVE-2023-20919 vulnerability can lead to local escalation of privilege, which means that a malicious app could gain higher permissions than it was intended to have.

What is the severity of CVE-2023-20919? The CVSS base score assigned to CVE-2023-20919 is 7.8, which is classified as HIGH severity.

Does CVE-2023-20919 require user interaction for exploitation? No, CVE-2023-20919 does not require user interaction for exploitation, making it more critical as it can be exploited automatically.

When was CVE-2023-20919 published? CVE-2023-20919 was published on 26 January 2023.

Where can I find more information about CVE-2023-20919? More information about CVE-2023-20919 can be found in the Android Security Bulletin published at <https://source.android.com/security/bulletin/2023-01>.

Could you describe a potential attack scenario for CVE-2023-20919? A potential attack scenario for CVE-2023-20919 might involve a malicious app or script that exploits the `getStringsForPrefix` method to escalate privileges.

Which products are affected by CVE-2023-20919? The product affected by CVE-2023-20919 is Android, specifically version Android-13.

What is the CVE ID for the vulnerability concerning activity launch restrictions in Android? The CVE ID for the vulnerability is CVE-2023-20916.

What kind of vulnerability is CVE-2023-20916? CVE-2023-20916 describes a security vulnerability where there is a possible way to bypass restrictions on activity launch.

What is the severity base score assigned to CVE-2023-20916? The severity base score assigned to CVE-2023-20916 is 7.8, categorized as HIGH.

Which versions of Android are affected by CVE-2023-20916? The versions of Android affected by CVE-2023-20916 are Android 12 and Android 12L.

Does the exploitation of CVE-2023-20916 require user interaction? No, the exploitation of CVE-2023-20916 does not require user interaction.

On what date was CVE-2023-20916 published? CVE-2023-20916 was published on 26 January 2023.

What is the official reference for further information on CVE-2023-20916? The official reference for further information on CVE-2023-20916 is the Android Security Bulletin, which can be found at <https://source.android.com/security/bulletin/2023-01>.

What type of vulnerability is CVE-2023-20916? Due to CVE-2023-20916, there is a potential for local escalation of privilege.

What is the Android ID associated with CVE-2023-20916? The Android ID associated with CVE-2023-20916 is A-229256049.

Can you provide an example of how CVE-2023-20916 could be exploited? An attacker could exploit CVE-2023-20916 by crafting a malicious app that does not require any user interaction to launch activities.

What is the CVE ID of the vulnerability in `addOrReplacePhoneAccount` method of `PhoneAccountManager`? The CVE ID of the vulnerability is CVE-2023-20915.

Can you provide more information about CVE-2023-20915? CVE-2023-20915 refers to a vulnerability in the `addOrReplacePhoneAccount` method of `PhoneAccountManager`.

What is the base score assigned to CVE-2023-20915 is 7.8, indicating it is a high-severity issue.

When was CVE-2023-20915 published on 26 January 2023.

Which Android versions affected by CVE-2023-20915 are Android 10, Android 11, Android 12, Android 13.

Where can more information about CVE-2023-20915 be found in the Android Security Bulletin at the following link: [https://source.android.com/security/bulletin/2023-01-26#CVE-2023-20915](#)

Does CVE-2023-20915 require user interaction for exploitation. No, CVE-2023-20915 does not require user interaction for exploitation.

What Android ID assigned to CVE-2023-20915 is A-246930197.

What type of attack CVE-2023-20915 could potentially lead to a local escalation of privilege attack, allowing an attacker to

Can you explain a possible attack scenario for CVE-2023-20915 could involve an attacker crafting a malicious application that

What is the CVE ID for the vulnerability associated with a tapjacking/overlay attack in Android is CVE-2023-20913.

Which file The primary file affected by the CVE-2023-20913 vulnerability is PhoneAccountSettingsActivity.java at

What type of attack CVE-2023-20913 allows an attacker to perform a tapjacking/overlay attack, which could mislead users

What versions of Android affected by CVE-2023-20913 include Android 10, Android 11, Android 12, and Android 13.

What privileges To exploit CVE-2023-20913, an attacker would need to execute with User execution privileges, indicating

What is the CVSS base score assigned to CVE-2023-20913 is 7.8, which is categorized as HIGH severity.

When was CVE-2023-20913 published on 26 January 2023.

Where can more information about CVE-2023-20913 be found on the Android Security Bulletin webpage at the following link: [https://source.android.com/security/bulletin/2023-01-26#CVE-2023-20913](#)

What are the potential consequences of an exploit against CVE-2023-20913 include the local escalation of privilege

Can you describe a possible attack scenario for CVE-2023-20913 involves an attacker crafting a malicious application that

What is the CVE ID for the vulnerability found in onActivityResult of AvatarPickerActivity.java is CVE-2023-20912.

Can you describe CVE-2023-20912 refers to a security vulnerability in the onActivityResult method of AvatarPickerActivity.java

What versions of Android affected by CVE-2023-20912 are versions up to including Android 13.

How severe The vulnerability described by CVE-2023-20912 is rated as 7.8, which is considered HIGH severity.

What is the CVE ID CVE-2023-20912 was published on 26 January 2023.

Where can more information about CVE-2023-20912 be found in the Android security bulletin at the following link: [https://source.android.com/security/bulletin/2023-01-26#CVE-2023-20912](#)

Is user interaction No, user interaction is not required to exploit CVE-2023-20912.

Could you describe An attacker could exploit CVE-2023-20912 by writing a malicious app which invokes the onActivityResult

What exact Regarding CVE-2023-20912, the onActivityResult method is vulnerable because it lacks a necessary permission

What remediation To remediate CVE-2023-20912, users should apply updates provided by the Android security bulletin,

What is the CVE ID of the vulnerability is CVE-2023-20908.

Which versions of Android affected by CVE-2023-20908 include Android 10, Android 11, Android 12, and Android 13.

What type of attack CVE-2023-20908 is a local denial of service vulnerability due to a possible system crash loop caused by

What impact CVE-2023-20908 can lead to a local denial of service on the affected Android devices where the system

Are there any CVE-2023-20908 does not require any user interaction.

What are the No additional execution privileges are needed to exploit CVE-2023-20908.

When was CVE-2023-20908 published on 26 January 2023.

Where can more information about CVE-2023-20908 be found at <https://source.android.com/security/bulletin/2023-01-26#CVE-2023-20908>

What is the CVSS The Common Vulnerability Scoring System (CVSS) base score for CVE-2023-20908 is 5.5, which is class

Could you describe An attack scenario for CVE-2023-20908 might involve a malicious application or script that repeatedly

What is CVE-2023-20905 is a security vulnerability identified in the Mfc\_Transceive function of the phNxpExtN

How severe The severity of CVE-2023-20905 is rated as 'HIGH' with a base score of 7.8, indicating a significant impact

Does CVE-2023-20905 require user interaction for exploitation, which makes it more severe because

What are the CVE-2023-20905 affects Android version 10. Devices running this version of Android could be vulnerable

When was The CVE-2023-20905 vulnerability was published on 26 January 2023.

Can you provide Yes, more details on CVE-2023-20905 can be found at the Android Security Bulletin website, specifically

What might A possible attack scenario for CVE-2023-20905 could be when a malicious application on the device in

What is 'out of bounds write' in the context of CVE-2023-20905 refers to writing data past the intended buffer

What mitigation To mitigate CVE-2023-20905, users should apply patches and updates provided by Android that address

What is the CVE ID of the security vulnerability? The CVE ID of the security vulnerability is CVE-2023-20904.

Can you describe CVE-2023-20904? CVE-2023-20904 refers to a security issue in 'getTrampolineIntent' of SettingsActivity.java on the Android platform.

What is the severity score assigned to CVE-2023-20904? The severity score assigned to CVE-2023-20904 is 7.8, which is categorized as HIGH.

Which versions of Android are affected by CVE-2023-20904? CVE-2023-20904 affects Android versions 12L and 13.

What are the steps to exploit the vulnerability described in CVE-2023-20904? To exploit the vulnerability described in CVE-2023-20904, an attacker does not need any additional execution privileges.

On what date was CVE-2023-20904 made public? CVE-2023-20904 was made public on 26 January 2023.

Where can more information about CVE-2023-20904 be found? More information about CVE-2023-20904 can be found at the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-01>.

What is the Android ID associated with CVE-2023-20904? The Android ID associated with CVE-2023-20904 is A-246300272.

What type of vulnerability is CVE-2023-20904? CVE-2023-20904 is a type of vulnerability that allows an attacker to cause a local escalation of privileges.

Could you provide a specific code example for exploiting CVE-2023-20904? I'm sorry, but I cannot provide a specific code example for exploiting CVE-2023-20904 as it would be in violation of the Android Security Bulletin's policy.

What is CVE-2022-20494? CVE-2022-20494 refers to a security vulnerability found in Android's AutomaticZenRule.java. It is a possible local escalation of privilege.

Which Android versions are affected by CVE-2022-20494? CVE-2022-20494 affects multiple Android versions including Android 10, Android 11, Android 12, and Android 12L.

What is the severity rating of CVE-2022-20494? CVE-2022-20494 has a severity rating of 5.5, which is deemed MEDIUM according to the CVSS (Common Vulnerability Scoring System).

When was CVE-2022-20494 publicly disclosed? CVE-2022-20494 was publicly disclosed on 26 January 2023.

Where can additional information about CVE-2022-20494 be found? Additional information about CVE-2022-20494 can be found in the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-01>.

How can CVE-2022-20494 be exploited? Exploitation of CVE-2022-20494 would involve an attacker leveraging the vulnerability to cause resource exhaustion.

Are there any code examples or proof of concept exploits for CVE-2022-20494? As of my knowledge up to the point of this writing, specific code examples or proof of concept exploits for CVE-2022-20494 are not available.

What kind of attack scenarios are associated with CVE-2022-20494? Attack scenarios associated with CVE-2022-20494 could involve a malicious application or script executing a denial of service attack.

What is the CVE ID for the vulnerability involving improper input validation in Condition.java? The CVE ID for the vulnerability involving improper input validation in Condition.java is CVE-2022-20493.

Can you describe CVE-2022-20493? CVE-2022-20493 is a security vulnerability present in Condition.java that allows for a possible local escalation of privilege.

Which versions of Android are affected by CVE-2022-20493? The versions of Android affected by CVE-2022-20493 include Android 10, Android 11, Android 12, and Android 12L.

What is the severity base score assigned to CVE-2022-20493? The severity base score assigned to CVE-2022-20493 is 7.8, which is categorized as HIGH.

When was the CVE-2022-20493 vulnerability published? The CVE-2022-20493 vulnerability was published on 26 January 2023.

What type of vulnerability is CVE-2022-20493? To exploit the vulnerability described in CVE-2022-20493, user interaction is needed.

Are there any code examples or proof of concept exploits for CVE-2022-20493? Yes, more details on CVE-2022-20493 can be found on the Android Security Bulletin, which is available at <https://source.android.com/security/bulletin/2023-01>.

What additional execution privileges are needed to exploit CVE-2022-20493? No additional execution privileges are needed to exploit CVE-2022-20493.

Could you provide a possible attack scenario involving CVE-2022-20493? A possible attack scenario involving CVE-2022-20493 might involve a malicious application tricking the user into interacting with it.

What is the CVE ID for the vulnerability involving CVE-2022-20492? The CVE ID for the vulnerability is CVE-2022-20492.

What is the severity level of CVE-2022-20492? CVE-2022-20492 has a severity level of HIGH with a base score of 7.8.

Which versions of Android are affected by CVE-2022-20492? CVE-2022-20492 affects Android versions Android-10, Android-11, Android-12, Android-12L, and Android-13.

What kind of vulnerability is CVE-2022-20492? CVE-2022-20492 can potentially lead to a local escalation of privilege on the affected Android device.

Is user interaction needed for the exploitation of CVE-2022-20492? No, user interaction is not needed for the exploitation of CVE-2022-20492.

Where can official details about CVE-2022-20492 be found? Official details about CVE-2022-20492 can be found on the Android Security Bulletin page at <https://source.android.com/security/bulletin/2023-01>.

Can you provide an attack scenario for CVE-2022-20492? An attacker could exploit CVE-2022-20492 by creating an app or script that intentionally exhausts system resources.

What is the Android ID associated with CVE-2022-20492? The Android ID associated with CVE-2022-20492 is A-242704043.

What is CVE-2022-20490? CVE-2022-20490 is a security vulnerability identified in multiple functions of AutomaticZenRule.java in the Android framework.

Which Android versions are affected by CVE-2022-20490? CVE-2022-20490 affects Android versions 10, 11, 12, 12L, and 13.

What is the severity score assigned to CVE-2022-20490? CVE-2022-20490 has been assigned a base score of 7.8, indicating a HIGH severity level.

When was CVE-2022-20490 published? CVE-2022-20490 was published on 26 January 2023.

Where can more information about CVE-2022-20490 be found? More information about CVE-2022-20490 can be found on the Android Security Bulletin at the following link: <https://source.android.com/security/bulletin/2023-01>.

What are the steps to exploit CVE-2022-20490? If CVE-2022-20490 is exploited, an attacker could achieve local escalation of privilege on the affected device.

Could you provide an attack scenario for CVE-2022-20490? An attack scenario for CVE-2022-20490 might involve a malicious app that exploits the vulnerability to cause a denial of service attack.

Are there any code examples or proof of concept exploits for CVE-2022-20490? Since CVE-2022-20490 is a security vulnerability, code examples that demonstrate the exploit are generally not provided.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2022-20489.

Which Android versions are affected by CVE-2022-20489? The vulnerability CVE-2022-20489 affects Android versions Android-10, Android-11, Android-12, and Android-12L.

What type of vulnerability is CVE-2022-20489? CVE-2022-20489 represents a vulnerability where there is a possible failure to persist permissions set in the system.

Is user interaction needed to exploit the vulnerability CVE-2022-20489? No, user interaction is not needed to exploit the vulnerability CVE-2022-20489.

What is the CVSS base score assigned to CVE-2022-20489 is 7.8, which is categorized as HIGH severity.

When was CVE-2022-20489 published on 26 January 2023.

Where can more detailed information about CVE-2022-20489 be found at the following URL: <https://source.android.com/security/bulletin/2023-01>.

What may a possible attack scenario for exploiting CVE-2022-20489 involve a malicious application that is able to access the system's internal storage.

What is CVE-2022-20461 refers to a security vulnerability discovered within the Bluetooth service code of Android.

Which versions CVE-2022-20461 affects several versions of the Android operating system, including Android 10, Android 11, and Android 12.

What is the severity level of CVE-2022-20461 is rated as 'HIGH' with a base score of 7.8.

Was there any user interaction required for the exploitation of CVE-2022-20461.

Where can more information about CVE-2022-20461 be found in the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2023-01>.

On what date was CVE-2022-20461 published on 26 January 2023.

What are possible attack scenarios for CVE-2022-20461 involve an attacker leveraging the type confusion error in the Bluetooth Low Energy (BLE) stack.

What does CVE-2022-20461 allow an attacker to achieve local escalation of privilege concerning Bluetooth Low Energy (BLE) communication.

What is the CVE ID of the vulnerability is CVE-2022-20458.

What is the base score assigned to CVE-2022-20458 is 5.5, which is considered MEDIUM severity.

Can you explain CVE-2022-20458 is a security vulnerability in which the StatusBarNotification.getKey() method, which is used to retrieve the key of a notification, is vulnerable to a null pointer exception.

When was CVE-2022-20458 published on 26 January 2023.

Which versions of Android version Android-12L is affected by CVE-2022-20458.

Could you provide further information about CVE-2022-20458 can be found at the Android Security Bulletin: <https://source.android.com/security/bulletin/2023-01>.

What might a possible attack scenario for CVE-2022-20458 would involve an attacker gaining access to the system's internal storage.

Is there a code example provided, the vulnerability described in CVE-2022-20458 would involve a null pointer exception.

What is the CVE ID of the reported vulnerability is CVE-2022-20456.

Which versions CVE-2022-20456 affects Android versions Android-10, Android-11, Android-12, Android-12L, and Android-13.

What is the base score assigned to CVE-2022-20456 is 7.8, which is categorized as HIGH.

What is CVE-2022-20456 is related to a possible failure to persist permissions settings due to resource exhaustion.

Is user interaction required for the exploitation of CVE-2022-20456.

When was CVE-2022-20456 published on 26 January 2023.

Where can more information about CVE-2022-20456 be found at the following reference: <https://source.android.com/security/bulletin/2023-01>.

Can you describe a possible attack scenario for CVE-2022-20456 involves a malicious app exploiting the resource exhaustion vulnerability.

What kind of impact CVE-2022-20456 could lead to local escalation of privilege.

What is the CVE ID associated with this vulnerability is CVE-2022-20235.

Can you describe CVE-2022-20235 refers to a vulnerability in the PowerVR GPU kernel driver where its 'Information Page' is vulnerable to a null pointer exception.

What is the CVSS Base Score for CVE-2022-20235 is 5.5, which is categorized as MEDIUM severity. It signifies a moderate level of risk.

On which date was CVE-2022-20235 published on 26 January 2023.

Which products are affected by CVE-2022-20235 are any Android systems on a chip (SoC) that implement the PowerVR GPU kernel driver.

What are possible attack scenarios for CVE-2022-20235 include a malicious application running on the Android system.

Could you provide further details on CVE-2022-20235 can be found at the following reference URL: <https://source.android.com/security/bulletin/2023-01>.

What action to address CVE-2022-20235, the vulnerability was patched in DDK version 1.18 of the PowerVR GPU kernel driver.

What is CVE-2022-20215 refers to a security vulnerability found in Android 10, Android 11, and Android 12. The vulnerability is related to the system's internal storage.

How severe is CVE-2022-20215 has been rated with a base score of 5.5 and classified as MEDIUM severity according to the CVSS.

When was CVE-2022-20215 published on 26 January 2023.

What versions CVE-2022-20215 affects Android versions 10, 11, and 12.

What type of attack exploiting CVE-2022-20215 requires user interaction, as a tapjacking or overlay attack necessitates the user's interaction.

Are there any further details about CVE-2022-20215 can be found at the Android Security Bulletin, specifically at <https://source.android.com/security/bulletin/2023-01>.

Can you provide an attack scenario for CVE-2022-20215 could involve a malicious application that creates an overlay on top of the system's internal storage.

What permissions To exploit CVE-2022-20215, the attacker does not need any additional execution privileges besides the system's internal storage.

What is CVE-2022-20214 refers to a security vulnerability in the Car Settings app on Android devices where the app is vulnerable to a tapjacking attack.

What type of attack CVE-2022-20214 is vulnerable to a tapjacking attack, which involves tricking a user into unknowingly granting permissions.

What is the severity score assigned to CVE-2022-20214 is 4.7, which is categorized as MEDIUM severity.

When was CVE-2022-20214 published on January 26, 2023.

Which Android versions affected by CVE-2022-20214 are Android 10, Android 11, and Android 12.

Where can you find more information about CVE-2022-20214 in the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-01>

Can you provide an example of an attack scenario for CVE-2022-20214 could be an attacker creating a malicious app that

How might an attacker exploit CVE-2022-20214 by crafting a malicious application that uses an overlay window

What is CVE-2022-20213 refers to a security vulnerability found in the ApplicationsDetailsActivity of Android

What kind of CVE-2022-20213 is a DoS vulnerability due to a tapjacking/overlay attack. It primarily impacts the user

Which Android versions affected by CVE-2022-20213 are Android 10, Android 11, and Android 12.

What is the severity score assigned to CVE-2022-20213 has been assigned a CVSS base score of 5.5, which is classified as MEDIUM severity.

What are the possible attack scenarios for CVE-2022-20213 include an attacker exploiting the tapjacking/overlay vulnerability

Where can more information on CVE-2022-20213 can be found in the Android Security Bulletin at the following link

When was CVE-2022-20213 published on January 26, 2023.

What is CVE-2022-4457 refers to a security vulnerability found in the WARP client for Android, which is due to

What type of CVE-2022-4457 allows for a task hijacking attack. In this type of attack, a malicious mobile application

What is the severity score assigned to CVE-2022-4457 is 5.5, and it is categorized as MEDIUM severity according to

When was CVE-2022-4457 published on 11 January 2023.

Where can additional information about CVE-2022-4457 can be found in the security advisory published by Cloud

Can you provide an attack scenario for CVE-2022-4457 could involve an attacker creating a malicious mobile application

Are there any code examples for exploiting vulnerabilities like CVE-2022-4457 as it could help in understanding the

What is the CVE ID of the vulnerability affecting the Fullscreen API in Google Chrome on Android is CVE-2023-0136

Can you describe the vulnerability CVE-2023-0136 refers to an inappropriate implementation in the Fullscreen API in Google

How serious is the CVE-2023-0136 vulnerability has been given a base score of 8.8, which is classified as HIGH severity

When was the CVE-2023-0136 vulnerability publicly disclosed on 10 January 2023.

Are there any references for CVE-2023-0136 vulnerability can be found through various references

What kind of attack could exploiting the CVE-2023-0136 vulnerability potentially allow an attacker to create a crafted HTML

What versions of CVE-2023-0136 affects Google Chrome on Android versions prior to 109.0.5414.74. Users running versions

What is the CVE ID for this vulnerability is CVE-2023-0133.

Can you describe CVE-2023-0133 refers to a vulnerability in Google Chrome on Android prior to version 109.0.5414.74,

What is the severity score assigned to CVE-2023-0133 has been assigned a Base Score of 6.5, which is categorized as MEDIUM severity by the

On what date was CVE-2023-0133 published on 10 January 2023.

Could you provide more details on CVE-2023-0133, you can refer to the following resources:- The Chromium

What possible attack scenarios for CVE-2023-0133 could involve a malicious actor creating a specially crafted HTML

Has CVE-2023-0133 been addressed by Google. Users should update their Google Chrome app on Android

Is there any code example that demonstrates how to exploit CVE-2023-0133 would not be helpful in understanding the

What is CVE-2023-0130 is a security vulnerability identified in the Fullscreen API of Google Chrome on Android

What is the severity rating of CVE-2023-0130 according to Chromium security is 'Medium'.

When was CVE-2023-0130 published on 10 January 2023.

What versions of Google Chrome on Android versions prior to 109.0.5414.74 are affected by CVE-2023-0130.

How could an attacker exploit CVE-2023-0130 by creating a crafted HTML page that, when visited by a user

Can you provide more information on CVE-2023-0130 can be found at the following URLs: <https://crbug.com/1370130>

What is the CVSS (Common Vulnerability Scoring System) Base Score assigned to CVE-2023-0130 is 6.5, indicating

What is the potential impact of an attack using CVE-2023-0130 is URL spoofing, where an attacker may deceive

How can users protect themselves from CVE-2023-0130 by updating their Google Chrome app on Android

What is CVE-2022-36928 is a security vulnerability identified in the Zoom application for Android, specifically in

How severe is CVE-2022-36928 is considered to have a HIGH severity level with a base score of 7.1. This indicates that

What type of CVE-2022-36928 is a path traversal vulnerability. This type of vulnerability occurs when a user input

What version? CVE-2022-36928 affects all versions of the Zoom Android client before version 5.13.0. Users with this version are affected. Has CVE-2022-36928 been addressed by Zoom? Yes, CVE-2022-36928 has been addressed by Zoom. The vulnerability was fixed in version 5.13.0 of the Zoom Android client. Where can I find additional details about CVE-2022-36928? Additional details about CVE-2022-36928 can be found in the security bulletin posted on Zoom's official website. What are the possible attack scenarios for CVE-2022-36928? In possible attack scenarios for CVE-2022-36928, a malicious third party application installed on the same device as the Zoom client could exploit the vulnerability. Could a code example be provided for the exploitation of CVE-2022-36928? Providing a specific code example for the exploitation of CVE-2022-36928 would be unethical and potentially harmful. What is the CVE ID for the vulnerability found in the Nextcloud Talk Android app? The CVE ID for the vulnerability found in the Nextcloud Talk Android app is CVE-2023-22473.

What are the details of the vulnerability described by CVE-2023-22473? The vulnerability described by CVE-2023-22473 allows an attacker with physical access to the target's device to bypass the passcode lock. What is the base score given to CVE-2023-22473? The base score given to CVE-2023-22473 is 2.1, which is categorized as LOW severity. When was CVE-2023-22473 published? CVE-2023-22473 was published on 09 January 2023.

Is there a workaround available for the vulnerability described by CVE-2023-22473? No, there are currently no known workarounds available for the vulnerability described by CVE-2023-22473. What is the recommended solution to resolve the vulnerability identified by CVE-2023-22473? The recommended solution to resolve the vulnerability identified by CVE-2023-22473 is to upgrade the Nextcloud Talk Android app to version 5.13.0 or later. Which resources provide additional information regarding CVE-2023-22473? Additional information regarding CVE-2023-22473 can be found at the following references: the GitHub repository for Nextcloud Talk, the Nextcloud website, and the Nextcloud security bulletin. What type of attack is associated with CVE-2023-22473? To exploit CVE-2023-22473, an attacker needs to have physical access to the target's Android device. Can you provide an example attack scenario for CVE-2023-22473? An example attack scenario for CVE-2023-22473 would be as follows: An attacker gains physical access to a device running the Nextcloud Talk Android app. The attacker enters the correct passcode to bypass the lock screen. The attacker then opens the Nextcloud Talk app and initiates a call to a contact. The call is successfully established, allowing the attacker to eavesdrop on the conversation. Is there any code example illustrating the vulnerability? As CVE-2023-22473 deals with a passcode bypass, concrete code examples illustrating the vulnerability are not provided. What is CVE-2022-33300? CVE-2022-33300 refers to a vulnerability that was identified in Automotive Android OS. It involves memory corruption in the Android core due to improper validation of user input. How severe is CVE-2022-33300? The CVE-2022-33300 vulnerability has been assigned a Base Score of 7.8, which is rated as HIGH. This indicates a significant risk to the system. When was CVE-2022-33300 published? The CVE-2022-33300 vulnerability was published on 09 January 2023.

Where can I find more information about CVE-2022-33300? More information about the CVE-2022-33300 vulnerability can be found on Qualcomm's website in the security bulletin. What kind of attack scenarios are associated with CVE-2022-33300? Potential attack scenarios that could exploit the CVE-2022-33300 vulnerability include scenarios where an attacker gains physical access to a device running Automotive Android OS and attempts to trigger the memory corruption. Can you provide code examples related to CVE-2022-33300? While specific code examples related to CVE-2022-33300 are not publicly provided due to security reasons, the vulnerability involves memory corruption in the Android core. What measures can be taken to mitigate the risks associated with CVE-2022-33300? To mitigate the risks associated with CVE-2022-33300, it's recommended to apply any patches or updates provided by Qualcomm. What is the CVE ID of the vulnerability involving memory corruption in the Android core? The CVE ID of the vulnerability involving memory corruption in the Android core due to improper validation of user input is CVE-2022-33274. What is the base score assigned to CVE-2022-33274? The base score assigned to CVE-2022-33274 is 7.8, which is categorized as HIGH severity. When was CVE-2022-33274 published? CVE-2022-33274 was published on 09 January 2023.

Can you provide further information on CVE-2022-33274? Yes, further information on CVE-2022-33274 can be found in the Qualcomm product security bulletin. What is the impact of CVE-2022-33274? The impact of CVE-2022-33274 is memory corruption within the Android core due to improper validation of user input. Could you provide a possible attack scenario for CVE-2022-33274? A possible attack scenario for CVE-2022-33274 would involve an attacker exploiting the improper validation of user input to trigger memory corruption in the Android core. What is CVE-2022-42979? CVE-2022-42979 refers to a security vulnerability in the RYDE application versions 5.8.43 for Android and 5.8.43 for iOS. How severe is CVE-2022-42979? The CVE-2022-42979 vulnerability has been given a base score of 8.8, which is categorized as HIGH severity. When was CVE-2022-42979 published? CVE-2022-42979 was published on 06 January 2023.

Where can I find more information on CVE-2022-42979? More information on CVE-2022-42979 can be found at the following URL: <https://medium.com/%40ja>. Can you provide an attack scenario for CVE-2022-42979? An attack scenario for CVE-2022-42979 might involve an attacker crafting a malicious deep link that exploits the vulnerability. What can be done to mitigate vulnerabilities like CVE-2022-42979? To mitigate vulnerabilities like CVE-2022-42979, developers should implement proper hostname validation. What is the CVE ID for the vulnerability found in the Opera Mini application for Android? The CVE ID for the vulnerability found in the Opera Mini application for Android is CVE-2018-16135. Can you describe the vulnerability with CVE-2018-16135? The vulnerability with CVE-2018-16135 allows remote attackers to spoof the Location Permission dialog. What is the base score assigned to CVE-2018-16135? CVE-2018-16135 has been assigned a CVSS (Common Vulnerability Scoring System) base score of 6.5, which is categorized as MEDIUM severity. On what date was CVE-2018-16135 published? CVE-2018-16135 was published on the 26th of December, 2022.

Where can I find more information about CVE-2018-16135? More information about CVE-2018-16135 can be found at the following URL: <https://payatu.com/advisories/cve-2018-16135>. Can you provide an example attack scenario for CVE-2018-16135? A possible attack scenario exploiting CVE-2018-16135 might involve an attacker creating a malicious vulnerability report. What is the CVE ID of the vulnerability affecting the F-Secure SAFE Browser on Android? The CVE ID of the vulnerability affecting the F-Secure SAFE Browser on Android is CVE-2022-47524. Can you describe the vulnerability with CVE-2022-47524? CVE-2022-47524 refers to a security vulnerability in the F-Secure SAFE Browser version 19.1 for Android. What is the base severity score assigned to CVE-2022-47524? CVE-2022-47524 has been assigned a base severity score of 5.4, which classifies it as a MEDIUM risk vulnerability. When was CVE-2022-47524 published? CVE-2022-47524 was publicly disclosed on 23 December 2022.

What is the recommended source to consult for more detailed information about CVE-2022-47524? For more detailed information about CVE-2022-47524, the recommended source to consult is the F-Secure website. Which version of the F-Secure SAFE Browser is affected by CVE-2022-47524? The issue identified in CVE-2022-47524 was resolved with the release of F-Secure SAFE Browser version 19.2. What does an IDN homograph attack involve? An IDN homograph attack, such as the one seen in CVE-2022-47524, typically involves the use of characters that look similar to legitimate domain names to trick users into visiting a malicious website.



Could you describe an example of a potential attack scenario for CVE-2022-47524? An example of a potential attack scenario for CVE-2022-47524 could involve the attacker registering a malicious domain and hosting a malicious website that exploits the vulnerability.

What is the CVE ID for the vulnerability involving the misuse of the 'S.browser\_fallback\_url' parameter in Firefox for Android? The CVE ID for the vulnerability involving the misuse of the 'S.browser\_fallback\_url' parameter in Firefox for Android is CVE-2022-45413.

Can you describe CVE-2022-45413? CVE-2022-45413 describes a vulnerability in Firefox for Android where an attacker could use the 'S.browser\_fallback\_url' parameter to redirect users to a malicious website.

Which versions of Firefox for Android are affected by CVE-2022-45413? CVE-2022-45413 affects versions of Firefox for Android prior to version 107. Users with Firefox for Android version 107 or later are not affected.

What is the CVSS Base Score assigned to CVE-2022-45413? The CVSS Base Score assigned to CVE-2022-45413 is 6.1, which is considered as 'MEDIUM' severity.

When was CVE-2022-45413 publicly disclosed? CVE-2022-45413 was publicly disclosed on 22 December 2022.

What operating systems are affected by CVE-2022-45413? Only the Android operating system is affected by CVE-2022-45413. This vulnerability is specific to Firefox for Android.

Where can I find more information about CVE-2022-45413? More information about CVE-2022-45413 can be found on the Mozilla Foundation Security Advisory page at <https://www.mozilla.org/en-US/security/advisories/mozc-2022-45413/>.

Can you suggest a potential attack scenario exploiting CVE-2022-45413? A potential attack scenario exploiting CVE-2022-45413 could involve an attacker crafting a web page that uses the 'S.browser\_fallback\_url' parameter to redirect users to a malicious website.

What is CVE-2022-40961? CVE-2022-40961 is a vulnerability identified in Firefox for Android, where a graphics driver with an untrusted pointer dereference could lead to a denial of service.

Which Firefox versions are affected by CVE-2022-40961? Firefox versions prior to 105 are affected by the CVE-2022-40961 vulnerability.

What is the impact severity of CVE-2022-40961? The impact severity of CVE-2022-40961 is rated as 6.5, which is considered MEDIUM according to the CVSS Base Score.

On which operating systems does CVE-2022-40961 exclusively have an impact? CVE-2022-40961 exclusively has an impact on the Android operating system. Other operating systems are not affected.

When was CVE-2022-40961 published? The CVE-2022-40961 vulnerability was published on 22 December 2022.

Where can I find more information regarding CVE-2022-40961? More information regarding CVE-2022-40961 can be found at Mozilla's security advisories webpage (<https://www.mozilla.org/en-US/security/advisories/mozc-2022-40961/>).

Can you provide an attack scenario for CVE-2022-40961? An attack scenario for CVE-2022-40961 might involve an attacker crafting a malicious graphics driver that exploits the untrusted pointer dereference to cause a denial of service.

Are there any code examples for exploiting CVE-2022-40961? Code examples for exploiting CVE-2022-40961 are generally not publicly provided to prevent facilitating further exploitation.

What is CVE-2022-38474? CVE-2022-38474 is a security vulnerability found in Firefox for Android. It involves an issue where a website can cause a denial of service by sending a large number of requests.

Which versions of Firefox are affected by CVE-2022-38474? CVE-2022-38474 affects Firefox versions before 104, specifically on the Android platform.

How severe is CVE-2022-38474? CVE-2022-38474 is rated as a Medium severity issue with a Base Score of 4.3 on the Common Vulnerability Scoring System (CVSS).

Was CVE-2022-38474 identified only in Firefox for Android? Yes, CVE-2022-38474 was identified only in Firefox for Android. Other operating systems where Firefox is installed are not affected.

How can users protect themselves from CVE-2022-38474? Users can protect themselves from the CVE-2022-38474 vulnerability by updating Firefox for Android to version 104 or later.

Can you provide a possible attack scenario for CVE-2022-38474? In a possible attack scenario for CVE-2022-38474, a malicious website that has previously obtained the user's IP address could cause a denial of service by sending a large number of requests.

Where can I find additional information about CVE-2022-38474? Additional information about CVE-2022-38474 can be found in the official advisories and the bug report at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784444](https://bugzilla.mozilla.org/show_bug.cgi?id=1784444).

When was CVE-2022-38474 publicly disclosed? CVE-2022-38474 was publicly disclosed on 22 December 2022, as indicated in the published advisory.

What is the CVE ID of the vulnerability affecting Firefox for Android with overly long URLs? The CVE ID of the vulnerability affecting Firefox for Android with overly long URLs is CVE-2022-36317.

What are the symptoms of the vulnerability CVE-2022-36317? The symptoms of the vulnerability CVE-2022-36317 include the user interface starting to hang when visiting a website with an overly long URL.

Which versions of Firefox for Android are affected by CVE-2022-36317? CVE-2022-36317 affects Firefox for Android versions prior to 103.

What is the CVSS Base Score for CVE-2022-36317? The CVSS Base Score for CVE-2022-36317 is 6.5, which is classified as MEDIUM severity. It indicates a partial denial of service.

When was CVE-2022-36317 published? The CVE-2022-36317 vulnerability was published on 22 December 2022.

What type of vulnerability is CVE-2022-36317? All operating systems other than Android are unaffected by CVE-2022-36317.

Where can I find more information about CVE-2022-36317? More information about CVE-2022-36317 can be found at the following references: - [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784444](https://bugzilla.mozilla.org/show_bug.cgi?id=1784444)

What could be a possible attack scenario for CVE-2022-36317? A possible attack scenario for CVE-2022-36317 could involve an attacker crafting a website with an extremely long URL to cause a denial of service.

What is the CVE ID for this vulnerability? The CVE ID for this vulnerability is CVE-2022-34469.

Can you describe CVE-2022-34469? CVE-2022-34469 describes a vulnerability where, on Firefox for Android, when a TLS Certificate error occurs, the user interface hangs.

Which versions of Firefox for Android are affected by CVE-2022-34469? CVE-2022-34469 affects versions of Firefox for Android earlier than 102.

What is the severity of CVE-2022-34469? CVE-2022-34469 has been assigned a base score of 8.1, which categorizes it as HIGH severity.

When was CVE-2022-34469 published? CVE-2022-34469 was published on 22 December 2022.

Where can I find more information and advisories about CVE-2022-34469? More information and advisories about CVE-2022-34469 can be found at the following URLs: <https://www.mozilla.org/en-US/security/advisories/mozc-2022-34469/> and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784444](https://bugzilla.mozilla.org/show_bug.cgi?id=1784444)

What operating systems are affected by CVE-2022-34469? CVE-2022-34469 only affects Firefox for Android. Other operating systems are not affected by this particular vulnerability.

Can you describe an attack scenario for CVE-2022-34469? An attacker could exploit CVE-2022-34469 by intercepting the communication between the user and a website, causing a denial of service.

What is the CVE ID for the vulnerability found in Firefox for Android? The CVE ID for the vulnerability found in Firefox for Android is CVE-2022-29910.

Can you describe CVE-2022-29910? CVE-2022-29910 refers to a vulnerability where Firefox for Android did not properly record and persist the state of the user interface, leading to a denial of service.

What is the CVSS Base Score for CVE-2022-29910? The Common Vulnerability Scoring System (CVSS) base score for CVE-2022-29910 is 6.1, which is categorized as MEDIUM severity.

On what date was CVE-2022-29910 publicly disclosed? CVE-2022-29910 was publicly disclosed on 22 December 2022.

Which versions of Firefox for Android are affected by CVE-2022-29910? CVE-2022-29910 affects versions of Firefox for Android that are older than (and excluding) version 100.

Where can I find more details about CVE-2022-29910? More details about CVE-2022-29910 can be found on the Mozilla security advisory page at <https://www.mozilla.org/en-US/security/advisories/mozc-2022-29910/>

Are there any? Since CVE-2022-29910 is related to the internal HSTS settings handling of the Firefox browser, and not the browser's HSTS settings, it is not a browser vulnerability.

What are the potential attack scenarios for CVE-2022-29910? Potential attack scenarios for CVE-2022-29910 could involve an attacker exploiting the lack of HSTS settings to perform a man-in-the-middle attack.

What is the CVE ID of the vulnerability affecting Mozilla's WebGPU IPC framework? The CVE ID of the vulnerability affecting Mozilla's WebGPU IPC framework is CVE-2022-26486.

Which Mozilla products are affected by CVE-2022-26486? CVE-2022-26486 impacts Firefox versions prior to 97.0.2, Firefox ESR versions prior to 91.6.1, Firefox for Android versions prior to 97.0.2, and Firefox for iOS versions prior to 97.0.2.

What is the severity rating for CVE-2022-26486? The severity rating for CVE-2022-26486 is 9.6, which is classified as CRITICAL.

Has CVE-2022-26486 been exploited in the wild? Yes, there have been reports of attacks in the wild exploiting CVE-2022-26486.

What type of vulnerability is CVE-2022-26486? CVE-2022-26486 is a use-after-free vulnerability in the WebGPU IPC framework, which could lead to a denial of service or information disclosure.

What is the main attack vector associated with CVE-2022-26486? The main attack vector associated with CVE-2022-26486 involves sending an unexpected message through the WebGPU IPC framework.

When was CVE-2022-26486 published? CVE-2022-26486 was published on 22 December 2022.

Where can more information on CVE-2022-26486 be found? More information on CVE-2022-26486 can be found at the following URLs:- [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1764861](https://bugzilla.mozilla.org/show_bug.cgi?id=1764861) and <https://www.mozilla.org/en-US/security/advisories/mozcvs-2022-02/>.

What is the CVE ID for the vulnerability affecting Mozilla's mobile products? The CVE ID for the vulnerability affecting Mozilla's mobile products is CVE-2022-26485.

Which versions of Firefox are affected by CVE-2022-26485? The versions of Firefox affected by CVE-2022-26485 are Firefox versions earlier than 97.0.2.

What is the severity rating for CVE-2022-26485? CVE-2022-26485 is a vulnerability that leads to an exploitable use-after-free condition in Firefox ESR, which is classified as CRITICAL.

How does CVE-2022-26485 affect Mozilla's mobile products? CVE-2022-26485 affects Mozilla's mobile products such as Firefox for Android and Focus, with vulnerabilities that could lead to information disclosure or denial of service.

Can you explain the attack scenario for CVE-2022-26485? CVE-2022-26485 has been assigned a base score of 8.8, which is categorized as HIGH severity. This indicates a significant risk of exploitation.

What type of vulnerability is CVE-2022-26485? CVE-2022-26485 addresses a use-after-free vulnerability, which is a type of memory corruption issue that can lead to crashes or information disclosure.

On which operating systems is CVE-2022-26485 exploitable? CVE-2022-26485 was published on 22 December 2022.

Where can you find more information and the official advisory about CVE-2022-26485? You can find more information and the official advisory about CVE-2022-26485 from the following sources: <https://www.mozilla.org/en-US/security/advisories/mozcvs-2022-02/> and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1764851](https://bugzilla.mozilla.org/show_bug.cgi?id=1764851).

Are there any direct code examples for an exploit like CVE-2022-26485? Providing direct code examples for an exploit like CVE-2022-26485 is typically not recommended due to the potential for misuse.

What are the possible attack scenarios associated with CVE-2022-26485? Possible attack scenarios associated with CVE-2022-26485 include remote attackers exploiting the use-after-free condition to perform a denial of service or information disclosure.

What is the CVE ID for the vulnerability affecting Mozilla's mobile products? The CVE ID for the vulnerability affecting Mozilla's mobile products is CVE-2022-22762.

What versions of Firefox are affected by CVE-2022-22762? CVE-2022-22762 affects Firefox versions prior to 97 on Android.

What kind of vulnerability is CVE-2022-22762? CVE-2022-22762 addresses an issue where a JavaScript alert or prompt could be displayed over another application's UI.

What is the severity rating for CVE-2022-22762? CVE-2022-22762 has a CVSS base score of 4.3 and is rated as MEDIUM severity.

Which operating systems are affected by CVE-2022-22762? Operating systems other than Android are not affected by CVE-2022-22762.

Can you provide more details about CVE-2022-22762? Yes, for more detailed information on CVE-2022-22762, you can refer to the Mozilla Security Advisory <https://www.mozilla.org/en-US/security/advisories/mozcvs-2022-02/>.

On what date was CVE-2022-22762 published? CVE-2022-22762 was published on 22 December 2022.

How can an attacker exploit CVE-2022-22762? An attacker could exploit CVE-2022-22762 by convincing a user to visit a malicious website that triggers the vulnerability.

What is the CVE ID for the vulnerability affecting Mozilla's mobile products? The CVE ID for the vulnerability affecting Mozilla's mobile products is CVE-2022-22758.

What is the severity rating for CVE-2022-22758? The risk severity base score of CVE-2022-22758 is 8.8, which is considered HIGH.

Can you describe the attack scenario for CVE-2022-22758? CVE-2022-22758 is a vulnerability in Firefox for Android where, upon clicking a tel: link, USSD codes for mobile services could be triggered.

Which versions of Firefox are affected by CVE-2022-22758? CVE-2022-22758 affects versions of Firefox for Android that are older than Firefox 97.

Are there any code examples for CVE-2022-22758? While specific code examples exploiting CVE-2022-22758 are not provided, an attack scenario could involve an attacker crafting a malicious web page.

Could you explain the attack scenario for CVE-2022-22758? A possible attack scenario for CVE-2022-22758 might involve an attacker crafting a malicious web page that triggers the vulnerability.

What should users do to mitigate CVE-2022-22758? Users of Firefox for Android should ensure their browser is updated to version 97 or higher to mitigate the vulnerability.

Where can more information about CVE-2022-22758 be found? More information about CVE-2022-22758 can be found on Mozilla's security advisory page at <https://www.mozilla.org/en-US/security/advisories/mozcvs-2022-02/>.

What is the CVE ID for the vulnerability associated with the scanning of QR codes? The CVE ID for the vulnerability associated with the scanning of QR codes in Firefox for Android is CVE-2022-22749.

Can you describe the attack scenario for CVE-2022-22749? CVE-2022-22749 is a vulnerability in Firefox for Android which, when scanning QR codes, allowed the attacker to perform a man-in-the-middle attack.

What is the severity rating for CVE-2022-22749? The base score for CVE-2022-22749 is 4.3, which is categorized as MEDIUM severity.

On what date was CVE-2022-22749 published? CVE-2022-22749 was published on 22 December 2022.

Are there any further details about CVE-2022-22749? Yes, further details about CVE-2022-22749 can be found at the following official references: Mozilla's security advisory <https://www.mozilla.org/en-US/security/advisories/mozcvs-2022-02/> and the bug report [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1764491](https://bugzilla.mozilla.org/show_bug.cgi?id=1764491).

Which versions of Firefox are affected by CVE-2022-22749? CVE-2022-22749 affects versions of Firefox for Android that are earlier than version 96.

What are the attack scenarios for CVE-2022-22749? An attack scenario for CVE-2022-22749 could involve a malicious actor creating a QR code that, when scanned, triggers the vulnerability.

What is the CVE ID for the vulnerability affecting Mozilla's software products? The CVE ID for the vulnerability affecting Mozilla's software products, including Firefox, Firefox for Android, and Firefox for iOS, is CVE-2022-1802.

What type of vulnerability is CVE-2022-1802? The software affected by CVE-2022-1802 includes Firefox ESR (versions before 91.9.1), Firefox (versions before 97.0.2), Firefox for Android (versions before 97.0.2), and Firefox for iOS (versions before 97.0.2).

What is the severity rating for CVE-2022-1802? The base score assigned to CVE-2022-1802 is 8.8, which is classified as HIGH.

When was CVE-2022-1802 published? CVE-2022-1802 was published on 22 December 2022.

Where can More information about CVE-2022-1802 can be found at the following URLs: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784441](https://bugzilla.mozilla.org/show_bug.cgi?id=1784441)

Can you pr A possible attack scenario for CVE-2022-1802 could involve a malicious actor crafting a web page or sc

What are t The recommended mitigation steps for CVE-2022-1802 include updating to the latest versions of Firef

What is the CVE ID for the vulnerability is CVE-2022-1529.

What is the Base Score assigned to CVE-2022-42543 is 4.4, which is categorized as a MEDIUM severity level.

When was CVE-2022-42543 published on 16 December 2022.

Where can More information about CVE-2022-42543 can be found at the following URL: <https://source.android.com/security/bulletin/2022-12-16-01>

What are the Possible attack scenarios for CVE-2022-42543 include an attacker with system-level privileges reading

What is the CVE-2022-42542 is a vulnerability that could lead to a local escalation of privilege on an Android device

What are the CVE-2022-42542 affects Android version 13.

What is the Base Score assigned to CVE-2022-42542 is 6.7, which is considered as MEDIUM severity.

Is user interaction required to exploit CVE-2022-42542.

What is the CVE-2022-42542 has been assigned a severity level of MEDIUM.

When was CVE-2022-42542 published on 16 December 2022.

Where can More information about CVE-2022-42542 can be found at the Android Security Bulletin page, at the following URL: <https://source.android.com/security/bulletin/2022-12-16-01>

Could you exploit CVE-2022-42542? A possible attack scenario for CVE-2022-42542 could involve a malicious application that is already running on the device.

What type CVE-2022-42542 is classified as a local privilege escalation vulnerability due to a possible out of bounds read.

What is the CVE ID of the vulnerability concerning SQL injection in MmsSmsProvider.java is CVE-2022-42535.

Could you exploit CVE-2022-42535? The CVE-2022-42535 vulnerability is described as a potential SQL injection in a query within MmsSmsProvider.java.

What is the CVE-2022-42535 has been assigned a severity level of MEDIUM with a base score of 5.5.

When was CVE-2022-42535 published on 16 December 2022.

Where can More information about CVE-2022-42535 can be found at the following URL: <https://source.android.com/security/bulletin/2022-12-16-01>

What are the To exploit CVE-2022-42535, an attacker would need to have User execution privileges on the device.

Can you exploit CVE-2022-42535? An attacker could exploit CVE-2022-42535 by crafting a malicious application that contains specifically crafted SQL queries.

What is the CVE ID for this vulnerability is CVE-2022-42534.

What is the CVE-2022-42534 could lead to local escalation of privilege due to improper input validation.

What are the To exploit the CVE-2022-42534 vulnerability, no additional execution privileges are needed, and no user interaction is required.

On what platform CVE-2022-42534 affects the Android operating system specifically on the Android kernel. The exact version affected is Android 13.

What is the CVSS base score for CVE-2022-42534 is 7.8 and it is classified as HIGH in terms of severity.

When was CVE-2022-42534 publicly disclosed on 16 December 2022.

Are there any details regarding CVE-2022-42534 can be found in the Pixel Update Bulletin at the following URL: <https://source.android.com/security/bulletin/2022-12-16-01>

Could you exploit CVE-2022-42534? In a possible attack scenario involving CVE-2022-42534, an attacker who already has access to the local system can exploit the vulnerability.

Is there any publicly shared code example detailing the exploit for CVE-2022-42534?

What is CVE-2022-42532 is a security vulnerability identified in the Pixel firmware that could allow an attacker to access sensitive information.

How serious CVE-2022-42532 has been assigned a Base Score of 4.4 and is classified as MEDIUM severity. This suggests that the vulnerability is not critical.

Were there any details on CVE-2022-42532 can be found at the Android Security Bulletin for Pixel devices.

What privileges To exploit CVE-2022-42532, an attacker would need System execution privileges. This means that the attacker must have root access.

Is user interaction required for an attacker to exploit CVE-2022-42532. This means the vulnerability can be exploited without user interaction.

What could happen If an attacker is able to successfully exploit CVE-2022-42532, they could potentially access sensitive information stored in memory.

Can you provide An example attack scenario for CVE-2022-42532 would involve a malicious application or a compromised system.

Was an update or patch provided for CVE-2022-42532. However, as of the information available, there is no publicly shared code example detailing the exploit for CVE-2022-42532.

What is the CVE ID of the vulnerability involving a possible mitigation bypass due to Permissive Memory Allocation is CVE-2022-42531.

Can you describe The vulnerability identified by CVE-2022-42531 exists in the mmu\_map\_for\_fw function within the gnu linker.

What is the Base Score assigned to CVE-2022-42531 is 7.8, which is categorized as HIGH. This score suggests that the vulnerability is critical.

When was The vulnerability with CVE ID CVE-2022-42531 was published on 16 December 2022.

Are there any details about CVE-2022-42531 can be found on the Android Open Source Project (AOSP) security page.

Which products The products affected by CVE-2022-42531 are Android kernel versions that are part of the Android operating system.

Does CVE-2022-42531 require user interaction for exploitation. No, user interaction is not needed for the exploitation of CVE-2022-42531. This means that the vulnerability can be exploited without user interaction.

What type As a result of CVE-2022-42531, local escalation of privilege can occur. This means an attacker with access to the kernel can escalate their privileges.

Could you provide A possible attack scenario for CVE-2022-42531 might involve a malicious application that is already installed on the device.

What is the CVE ID for the vulnerability in Pixel firmware related to an out of bounds read is CVE-2022-42530.

Can you pr CVE-2022-42530 involves a possible out of bounds read in Pixel firmware due to a missing bounds che  
What is the The base score of CVE-2022-42530 is rated as 4.4, which is categorized as MEDIUM severity.  
When was CVE-2022-42530 was published on 16 December 2022.  
Which pro The product affected by CVE-2022-42530 is Android, specifically the Android kernel within Pixel firmw  
Do users n No, user interaction is not needed for exploitation of CVE-2022-42530.  
Are there c The details provided do not include specific code examples for exploiting CVE-2022-42530. As exploit  
What privil Exploiting CVE-2022-42530 requires System execution privileges.  
What are t Potential attack scenarios for CVE-2022-42530 include an attacker with System execution privileges e  
Where can Further information on CVE-2022-42530 can be found at the Android Security Bulletin's Pixel updates  
What is CV CVE-2022-42529 is a security vulnerability identified in the Android kernel which affects various versio  
How sever The threat posed by CVE-2022-42529 is considered to be CRITICAL with a base score of 9.8. This implic  
What Andr CVE-2022-42529 affects Android kernel versions without specific version details provided. Users are n  
Where can More information about CVE-2022-42529 can be found at the Android Security Bulletin for Pixel devic  
What kind An exploit of CVE-2022-42529 could allow an attacker to execute arbitrary code with kernel privileges  
Are there a Specific code examples for CVE-2022-42529 have not been provided in the given information, and ger  
What are p Attack scenarios for CVE-2022-42529 could vary but typically involve an attacker crafting a malicious a  
What is the The CVE ID for this vulnerability is CVE-2022-42527.  
What is the The vulnerability identified by CVE-2022-42527 could lead to remote denial of service due to a possibl  
Does explo No, the exploitation of CVE-2022-42527 does not require additional execution privileges or user intera  
What is the CVE-2022-42527 has been assigned a CVSS base score of 7.5, which is categorized as HIGH.  
What versi The versions of Android affected by CVE-2022-42527 are specified as 'Android kernel'. Exact version n  
What is the The Android ID associated with CVE-2022-42527 is A-244448906.  
Where can More details and reference material about CVE-2022-42527 can be found at the Android security bull  
Was there No code example was provided in the details for CVE-2022-42527. However, it is mentioned that the v  
When was CVE-2022-42527 was first published on 16 December 2022.  
Can you de A possible attack scenario for CVE-2022-42527 involves a remote attacker sending a maliciously craft  
What is the The CVE ID for the vulnerability is CVE-2022-42526.  
Can you de CVE-2022-42526 is a security flaw in the ConvertUtf8ToUcs2 function of radio\_hal\_utils.cpp, where th  
What is the CVE-2022-42526 has a severity rating of MEDIUM with a base score of 6.7.  
On which c CVE-2022-42526 was published on 16 December 2022.  
Which pro The products affected by CVE-2022-42526 are devices running the Android kernel, as specified in the v  
What kind To exploit CVE-2022-42526, an attacker would need System execution privileges.  
Is user inte User interaction is not needed to exploit the vulnerability described in CVE-2022-42526.  
Where can More information about CVE-2022-42526 can be found at the Android security bulletin page, specifica  
Could you An attack scenario for CVE-2022-42526 could involve a malicious application that uses the vulnerabilit  
What is the The Android ID associated with CVE-2022-42526 is A-243509880.  
What is the The CVE ID for the vulnerability is CVE-2022-42525.  
What is the The vulnerability described in CVE-2022-42525 could lead to a local escalation of privilege, allowing a  
What is the CVE-2022-42525 has been assigned a base CVSS score of 6.7, categorized as MEDIUM severity.  
What versi CVE-2022-42525 affects the Android kernel, but the specific versions of Android impacted by this vuln  
Are user in No, user interaction is not necessary for the exploitation of CVE-2022-42525.  
When was The CVE-2022-42525 vulnerability was published on 16 December 2022.  
Where can More information about CVE-2022-42525 can be found on the Android security bulletin webpage at: h  
Can you pr As CVE-2022-42525 is an out of bounds write vulnerability, a possible attack scenario would involve a  
What privil To exploit the vulnerability CVE-2022-42525, the attacker would need to have System execution privil  
What is CV CVE-2022-42524 is a vulnerability in the sms\_GetTpUdIle function of the sms\_PduCodec.c file, which i  
What is the The severity level of CVE-2022-42524 is rated as 'HIGH' with a base score of 7.5 according to its entry i

Which versions CVE-2022-42524 affects the Android operating system kernel, but specific version numbers are not provided.

How can CVE-2022-42524 be exploited? CVE-2022-42524 can potentially be exploited by a remote attacker who can send a specially crafted SIP message.

Has CVE-2022-42524 been addressed by Android in their security bulletins? Yes, CVE-2022-42524 has been addressed by Android in their security bulletins. The vulnerability was fixed in Android 12L.

Where can more information about CVE-2022-42524 be found? More information about CVE-2022-42524 can be found in the Android security bulletin for Pixel devices.

What are the potential consequences of an exploitation of CVE-2022-42524? The potential consequences of an exploitation of CVE-2022-42524 include unauthorized remote information disclosure.

What is CVE-2022-42523? CVE-2022-42523 refers to a security vulnerability identified in a specific part of the Android operating system kernel.

How severe is CVE-2022-42523? The severity of CVE-2022-42523 is rated as 'MEDIUM' with a base score of 6.7. This indicates that the vulnerability is of moderate severity.

What kind of attack is CVE-2022-42523? To exploit CVE-2022-42523, an attacker would need to have System execution privileges on the affected device.

Is user interaction required for CVE-2022-42523? No, user interaction is not required to exploit the vulnerability described in CVE-2022-42523. The flaw exists in the kernel.

On which Android versions does CVE-2022-42523 exist? CVE-2022-42523 exists on the Android platform, and the vulnerability is present in the Android kernel versions 30 and 31.

Where can more information about CVE-2022-42523 be found? More information about CVE-2022-42523 can be found on the Android Security Bulletin page, specifically for Pixel devices.

What is the impact of CVE-2022-42523? The impact of CVE-2022-42523 on an Android device could potentially allow an attacker to execute code with System execution privileges.

When was CVE-2022-42523 made public? CVE-2022-42523 was made public on the 16th of December, 2022, as indicated by the published date in the security bulletin.

What is the CVE ID for the vulnerability discovered in DoSetCarrierConfig of miscservice.cpp? The CVE ID for the vulnerability discovered in DoSetCarrierConfig of miscservice.cpp is CVE-2022-42522.

Can you briefly describe CVE-2022-42522? CVE-2022-42522 is associated with a possible out of bounds read in DoSetCarrierConfig of the miscservice.cpp file.

What is the severity of CVE-2022-42522? CVE-2022-42522 has been assigned a Base Score of 4.4, which is considered MEDIUM severity.

When was CVE-2022-42522 published? The CVE-2022-42522 vulnerability was published on 16 December 2022.

Where can more information about CVE-2022-42522 be found? More information about CVE-2022-42522 can be found on the official Android Security Bulletin page.

What versions of Android are affected by CVE-2022-42522? CVE-2022-42522 affects Android kernel versions as noted in the provided CVE description. For precise details, refer to the security bulletin.

What are the attack requirements for CVE-2022-42522? To exploit the vulnerability CVE-2022-42522, an attacker would need to have System execution privileges.

Can you illustrate an attack scenario for CVE-2022-42522? An attack scenario for CVE-2022-42522 might involve a malicious application that has somehow obtained System execution privileges.

What is CVE-2022-42521? CVE-2022-42521 is a security vulnerability found in the encode function of wlandata.cpp. It is a possible information disclosure vulnerability.

How severe is CVE-2022-42521? CVE-2022-42521 has been assigned a Base Score of 6.7, which is classified as MEDIUM severity. This score is based on the CVSS (Common Vulnerability Scoring System).

Was there a patch for CVE-2022-42521? Information regarding a specific patch for CVE-2022-42521 wasn't directly provided, but it's referenced in the security bulletin.

When was CVE-2022-42521 publicly disclosed? CVE-2022-42521 was publicly disclosed on 16 December 2022.

What kind of attack is CVE-2022-42521? An attacker would need system execution privileges to exploit CVE-2022-42521, implying that a certain level of access is required.

What are the attack requirements for CVE-2022-42521? An attack scenario for CVE-2022-42521 could involve a malicious application that is already running on the device.

Where can additional details about CVE-2022-42521 be found? Additional details about CVE-2022-42521 can be found in the Android Security Bulletin dated December 16, 2022.

What is the CVE ID of the vulnerability affecting the Android kernel in serviceinterface.cpp? The CVE ID of the vulnerability affecting the Android kernel in serviceinterface.cpp is CVE-2022-42520.

Can you describe CVE-2022-42520? CVE-2022-42520 describes a vulnerability in ServiceInterface::HandleRequest of serviceinterface.cpp, which could lead to a denial of service.

What type of attack is CVE-2022-42520? An attacker would need system execution privileges to exploit the vulnerability described by CVE-2022-42520.

Is user interaction required for CVE-2022-42520? No, user interaction is not required for exploiting the vulnerability referenced in CVE-2022-42520.

What is the CVSS (Common Vulnerability Scoring System) Base Score assigned to CVE-2022-42520? The CVSS (Common Vulnerability Scoring System) Base Score assigned to CVE-2022-42520 is rated as 6.7.

When was CVE-2022-42520 published? CVE-2022-42520 was published on 16 December 2022.

Where can more information regarding the CVE-2022-42520 vulnerability be found? More information regarding the CVE-2022-42520 vulnerability can be found at the following source: [https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42520](#).

What versions of Android are affected by CVE-2022-42520? The affected versions of Android related to CVE-2022-42520 are specified as Android kernel versions 30 and 31.

What could be a possible attack scenario for CVE-2022-42520? A possible attack scenario for CVE-2022-42520 could involve a malicious app or process already having System execution privileges.

What is CVE-2022-42519? CVE-2022-42519 is a security vulnerability identified in the CdmaBroadcastSmsConfigsRequestData::encode function.

How severe is CVE-2022-42519? The vulnerability CVE-2022-42519 has been assigned a base score of 6.7, which is categorized as MEDIUM severity.

Does CVE-2022-42519 require user interaction? No, user interaction is not needed to exploit the vulnerability denoted by CVE-2022-42519. This makes it a remote attack.

What privileges are required for CVE-2022-42519? To exploit the vulnerability CVE-2022-42519, an attacker would need System execution privileges. This is a high-privilege requirement.

Which versions of Android are affected by CVE-2022-42519? CVE-2022-42519 affects versions of the Android kernel without specifying a particular version range. It is a kernel-wide vulnerability.

Where can official information about CVE-2022-42519 be found? Official information about CVE-2022-42519 can be found in the Android security bulletin located at [https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42519](#).

What could be the impact of CVE-2022-42519? By exploiting CVE-2022-42519, an attacker could cause memory corruption through a stack clash, leading to a denial of service.

When was CVE-2022-42519 published? CVE-2022-42519 was published on 16 December 2022.

What is a potential attack scenario for CVE-2022-42519? A potential attack scenario involving CVE-2022-42519 could start with an attacker who has already gained System execution privileges.

What is the CVE ID for the vulnerability allowing a possible out of bounds write in 'smsdata.cpp'? The CVE ID for the vulnerability allowing a possible out of bounds write in 'smsdata.cpp' is CVE-2022-42518.

What type CVE-2022-42518 specifies a vulnerability that involves a possible out of bounds write due to a missing Which prod The product affected by CVE-2022-42518 is Android, specifically the Android kernel.

What are t To exploit CVE-2022-42518, System execution privileges are needed, and the potential impact include Does CVE-2 No, exploitation of CVE-2022-42518 does not require user interaction.

What is the The CVSS Base Score assigned to CVE-2022-42518 is 6.7, which is categorized as MEDIUM severity.

When was CVE-2022-42518 was publicly disclosed on 16 December 2022.

Where can More details about CVE-2022-42518 can be found at the Android security bulletin webpage: <https://source.android.com/security/bulletin/2022-12-16-01>

Can you de A possible attack scenario for CVE-2022-42518 could involve a malicious application that is already pr

Are there e As of my current knowledge, specific code examples or proof of concept for exploiting CVE-2022-4251

What is CV CVE-2022-42517 is a security vulnerability found in MiscService::DoOemSetTcsFci of miscservice.cpp c

What is the The severity level of CVE-2022-42517 is rated as 'MEDIUM' with a base score of 4.4.

When was CVE-2022-42517 was published on 16 December 2022.

Does CVE-2 No, CVE-2022-42517 does not require user interaction for exploitation.

What are t An attacker needs System execution privileges to exploit CVE-2022-42517.

Are there e The specific details of the code vulnerability for CVE-2022-42517 are not mentioned, and an example

Where can Additional information about CVE-2022-42517 can be found in the Android Security Bulletin for Pixel c

What attac Attack scenarios for CVE-2022-42517 would likely involve a malicious application or actor with System

What is the The CVE ID of the vulnerability is CVE-2022-42516.

Which file The file associated with CVE-2022-42516 in the Android operating system is protocolsimbuilderlegacy

What are t The potential consequences of the CVE-2022-42516 vulnerability include local information disclosure.

What privil To exploit the vulnerability described in CVE-2022-42516, System execution privileges are needed.

Does the e No, user interaction is not needed for the exploitation of CVE-2022-42516.

How is CVE CVE-2022-42516 is rated as having a severity level of 4.4, which is categorized as MEDIUM.

When was CVE-2022-42516 was published on 16 December 2022.

Where can More information or a detailed security bulletin about CVE-2022-42516 can be found at the following

Which vers Affected versions include the Android kernel versions specified in the Android security bulletin, but th

Can you de A potential attack scenario for exploiting CVE-2022-42516 could involve a malicious application with s

What is CV CVE-2022-42515 refers to a security vulnerability found in MiscService::DoOemSetRtpPktlossThreshol

How sever CVE-2022-42515 has been assessed with a base score of 4.4, categorizing it as a MEDIUM severity vuln

What coul By exploiting CVE-2022-42515, an attacker could potentially cause an out-of-bounds read, which may

What versi CVE-2022-42515 affects versions of the Android kernel without specific version details provided. User

Was CVE-2 Yes, CVE-2022-42515 was the result of a missing bounds check in the MiscService::DoOemSetRtpPktl

Are there e Official CVE descriptions typically do not include specific code examples that demonstrate vulnerabilit

Does explo No, exploiting CVE-2022-42515 does not require user interaction.

What is the The best source for updates and patches regarding CVE-2022-42515 is the Android Security Bulletin, p

What are s Potential attack scenarios for CVE-2022-42515 could involve a malicious application already having sy

What is the The CVE ID number for the vulnerability is CVE-2022-42514.

Can you de CVE-2022-42514 corresponds to a vulnerability in the ProtocolmsBuilder::BuildSetConfig method of p

What type To exploit the vulnerability denoted by CVE-2022-42514, an attacker would need System execution pr

Is user inte No, user interaction is not required to exploit the vulnerability described in CVE-2022-42514.

What is the The CVSS Base Score for CVE-2022-42514 is 4.4, which indicates a medium severity level for this vulne

On what d The CVE-2022-42514 vulnerability was published on 16 December 2022.

Where can A detailed report and more information about CVE-2022-42514 can be found at the Android Security l

What versi CVE-2022-42514 affects versions of the Android kernel; however, the specific versions impacted are n

Could you An attacker with system-level access on a vulnerable Android device could exploit the out-of-bounds i

What is the The CVE ID for the vulnerability is CVE-2022-42513.

Can you de CVE-2022-42513 describes a vulnerability where there is a possible out of bounds write in the file 'pro

What are the requirements to exploit the vulnerability CVE-2022-42513, System execution privileges are required.

Is user interaction needed? No, user interaction is not needed to exploit the vulnerability CVE-2022-42513.

How has the CVE-2022-42513 vulnerability been rated with a base score of 6.7, which falls into the MEDIUM severity category.

On which date was the CVE-2022-42513 vulnerability published on 16 December 2022.

Which versions of the Android kernel are affected by CVE-2022-42513 are not explicitly mentioned, but the vulnerability affects versions 30 and 31.

Are there any additional details on CVE-2022-42513 can be found at the Android Security Bulletin webpage: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42513>

What could an attacker successfully exploit the CVE-2022-42513 vulnerability, they could achieve local escalation of privileges.

Are there any other CVEs related to CVE-2022-42513 concerns a specific out of bounds write due to a missing bounds check in the Protobuf library.

What possible attack scenarios for CVE-2022-42513 include an attacker with system-level privileges manipulating the system's boot process.

What is the CVE ID associated with this vulnerability is CVE-2022-42512.

Can you describe CVE-2022-42512 describes a vulnerability in the `VsimOperationDataExt::encode` function, specifically in the `encode` method.

What is the severity score given to CVE-2022-42512 has a CVSS base score of 4.4, which categorizes it as a vulnerability with medium severity.

When was CVE-2022-42512 was first published on 16 December 2022.

What versions of the Android kernel are affected by CVE-2022-42512 affects the Android kernel. However, the specific versions of Android impacted by this vulnerability are not explicitly mentioned.

Is user interaction needed? No, user interaction is not needed for the exploitation of CVE-2022-42512.

What privileges are required to exploit the vulnerability detailed in CVE-2022-42512, an attacker would need to have System execution privileges.

Are there any additional details on CVE-2022-42512 can be found in the Android Security Bulletin for Pixel devices: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42512>

What are the possible attack scenarios for CVE-2022-42512 include an attacker with System execution privileges manipulating the system's boot process.

What is the CVE ID of the vulnerability is CVE-2022-42511.

Can you explain CVE-2022-42511 refers to a security issue in `EmbmsSessionData::encode` of `embmsdata.cpp` where there is a possible out of bounds read.

What are the requirements to exploit the vulnerability with CVE-2022-42511, an attacker would need to have System execution privileges.

Was there any user interaction needed? No, user interaction is not required for exploitation of the CVE-2022-42511 vulnerability.

Which product is affected by CVE-2022-42511 is Android, specifically the Android kernel. The exact version affected is not specified.

What is the severity score given to CVE-2022-42511 is 6.7, which is classified as MEDIUM according to risk assessment.

When was the CVE-2022-42511 vulnerability published on 16 December 2022.

Where can more details regarding the CVE-2022-42511 vulnerability can be found at the following link: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42511>

Could you provide an attack scenario utilizing CVE-2022-42511 could involve a malicious application that has already gained System execution privileges.

Are there any code examples for vulnerabilities like CVE-2022-42511 are not typically made public to prevent misinterpretation.

What is the CVE ID for the vulnerability involving an out of bounds read in `StringsRequestData::encode` is CVE-2022-42510.

Can you describe CVE-2022-42510 is described as a possible out of bounds read due to improper input validation in `StringsRequestData::encode`.

What are the requirements to exploit CVE-2022-42510, an attacker would need System execution privileges on the affected Android device.

Is user interaction needed? No, user interaction is not needed for the exploitation of CVE-2022-42510.

What versions of the Android kernel are affected by CVE-2022-42510, but specific affected versions are not provided in the details.

What is the severity score assigned to CVE-2022-42510 is 6.7, which is categorized as MEDIUM severity.

When was CVE-2022-42510 was first published on 16 December 2022.

Where can more information about CVE-2022-42510 can be found at the following link: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42510>

Are there any public references provided in the CVE details is the Android Security Bulletin for Pixel devices: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42510>

What might a possible attack scenario for CVE-2022-42510 would involve an attacker with System execution privileges manipulating the system's boot process.

What is the CVE ID of the vulnerability involving a possible out of bounds write in `CallDialReqData::encode` is CVE-2022-42509.

Can you describe CVE-2022-42509 describes an issue where there is a possible out of bounds write in the `CallDialReqData::encode` function.

What is the severity score assigned to CVE-2022-42509 is 6.7, which is categorized as MEDIUM severity.

On what date was CVE-2022-42509 was published on 16 December 2022.

Are there any official references provided for CVE-2022-42509 is available at the Android Security Bulletin: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42509>

What versions of Android are affected by CVE-2022-42509 are specified as Android kernel versions without specific version numbers.

What type of attack is required to exploit CVE-2022-42509, meaning they must have System execution privileges.

Is user interaction needed? No, user interaction is not needed to exploit the vulnerability outlined in CVE-2022-42509.

What could a possible attack scenario for exploiting CVE-2022-42509 involves an attacker who has already obtained System execution privileges.



What is CV CVE-2022-42508 refers to a security vulnerability found in the ProtocolCallBuilder::BuildSendUsd me  
How sever The vulnerability CVE-2022-42508 has been given a Base Score of 6.7, categorizing it as a MEDIUM sev  
What are t To exploit CVE-2022-42508, an attacker would need System execution privileges on the affected Andr  
Is user inte No, user interaction is not needed for an attack exploiting CVE-2022-42508 to be successful. This incre  
On which c CVE-2022-42508 was publicly disclosed on 16 December 2022.

Where can Detailed information about CVE-2022-42508 can be found through the provided reference link: [https:](https://)  
Can you pr A potential attack scenario involving CVE-2022-42508 could be where a malicious application with Sys  
What is CV CVE-2022-42507 is a security vulnerability found in 'ProtocolSimBuilder::BuildSimUpdatePb3gEntry' o  
How sever The vulnerability identified by CVE-2022-42507 has a Base Score of 6.7, which is rated as MEDIUM sev  
Does CVE-2 No, CVE-2022-42507 does not require user interaction for exploitation, making it particularly concerni  
What privil An attacker would need System execution privileges to successfully exploit CVE-2022-42507. Despite t  
When was CVE-2022-42507 was published on 16 December 2022.

Where can Further details about CVE-2022-42507 can be found at the Android Security Bulletin page, specifically  
Can you de A potential attack scenario for CVE-2022-42507 could be as follows: an attacker with System execution  
What are t CVE-2022-42507 affects the Android kernel. Specific Android versions impacted by this vulnerability h  
What is CV CVE-2022-42506 is a security vulnerability found in the SimUpdatePbEntry::encode function of simdat  
How sever The severity of CVE-2022-42506 is rated with a base score of 6.7, which is classified as MEDIUM accor  
What versi CVE-2022-42506 affects versions of the Android kernel, but the specific versions impacted are not det  
Is there a r No, user interaction is not required to exploit CVE-2022-42506. The vulnerability can be exploited with  
What privil An attacker would need System execution privileges to exploit CVE-2022-42506.  
When was CVE-2022-42506 was published on 16 December 2022.

Are there ε The provided information does not include any specific code examples for CVE-2022-42506. Exploitati  
Where can Official information about CVE-2022-42506 can be found on the Android Security Bulletin page, partic  
What are s Potential attack scenarios for CVE-2022-42506 include a malicious application gaining System executi  
What is the CVE ID for the vulnerability is CVE-2022-42505.

What type CVE-2022-42505 could lead to local escalation of privilege with System execution privileges needed.  
What are t The affected versions for CVE-2022-42505 are Android versions that include the Android kernel involv  
Does CVE-2 No, user interaction is not needed for exploitation of CVE-2022-42505.  
Where can More information about CVE-2022-42505 can be found at the Android Security Bulletin page: <https://>  
What is the CVE ID for the vulnerability is CVE-2022-42505.  
What is the CVSS base score for CVE-2022-42505 is 6.7, which signifies that it is rated as a medium severity vu  
What is the impact of CVE-2022-42505 on an Android device could be significant as it potentially allows an att  
What mea: To mitigate the risk associated with CVE-2022-42505, users should apply the security updates provide  
Could you An attack scenario for CVE-2022-42505 could involve a malicious application specifically crafted to exp

What is CV CVE-2022-42504 is a security vulnerability identified in the CallDialReqData::encodeCallNumber funct  
What is the base score of CVE-2022-42504 is 6.7, which is categorized as MEDIUM severity.  
When was CVE-2022-42504 was published on the 16th of December, 2022.  
Does explo No, user interaction is not needed for exploitation of CVE-2022-42504.

What privil To exploit CVE-2022-42504, an attacker would need System execution privileges.  
Can you pr Yes, further information on CVE-2022-42504 can be found at the Android Security Bulletin page: [https:](https://)  
What versi CVE-2022-42504 affects Android kernel versions with reference to the Android ID A-241232209, but s  
What coul An attacker with access to the device and system execution privileges might exploit the CVE-2022-425  
Are there ε Since CVE-2022-42504 is a security vulnerability, responsible disclosure practices do not include publi  
What is the CVE ID of the vulnerability is CVE-2022-42503.

Can you pr CVE-2022-42503 involves a possible out of bounds write in ProtocolMiscBuilder::BuildSetLinkCapaRep  
What are t An attacker would need System execution privileges to exploit the vulnerability associated with CVE-2  
Was user i No, user interaction is not required to exploit the vulnerability detailed in CVE-2022-42503.  
What is the CVE ID for the vulnerability is CVE-2022-42503.  
What is the CVSS Base Score of CVE-2022-42503 has been assigned a CVSS Base Score of 6.7, which categorizes it as a MEDIUM sever

On what date CVE-2022-42503 was published on 16 December 2022.

Could you provide further information on CVE-2022-42503 can be found at the following URL: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42503>

Which product The product affected by CVE-2022-42503 is Android, specifically the Android kernel.

What potential Given that CVE-2022-42503 allows for an out of bounds write due to a missing bounds check, a local attacker could potentially exploit this vulnerability to gain system execution privileges.

What is the CVSS score CVE-2022-42502 identifies a security issue in FacilityLock::Parse of simdata.cpp, which is part of the Android kernel.

What is the severity level The severity level of CVE-2022-42502 is rated as '6.7 MEDIUM' on the Common Vulnerability Scoring System (CVSS).

When was CVE-2022-42502 was published on the 16th of December, 2022.

Which Android versions CVE-2022-42502 affects the Android kernel, but the specific versions impacted by the vulnerability are not specified.

What privileges An attacker would need System execution privileges to exploit CVE-2022-42502.

Is user interaction No, user interaction is not required to exploit the vulnerability described in CVE-2022-42502.

Where can Additional information about CVE-2022-42502 can be found in the Android Security Bulletin at <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42502>

Can you provide A possible attack scenario involving CVE-2022-42502 could be that a malicious application with System execution privileges could exploit this vulnerability to gain system execution privileges.

What type CVE-2022-42502 describes an out of bounds write vulnerability, which occurs when a program writes to a memory location outside its allocated bounds.

What is the CVE ID The CVE ID for the vulnerability in HexString2Value of util.cpp that may lead to an out of bounds write is CVE-2022-42501.

Can you describe CVE-2022-42501 is a vulnerability in HexString2Value of util.cpp, where there is a missing bounds check.

What privileges To exploit CVE-2022-42501, the attacker would need to have System execution privileges on the affected device.

What versions CVE-2022-42501 affects the Android kernel. However, the specific versions of Android affected by CVE-2022-42501 are not specified.

What is the CVSS score CVE-2022-42501 has been assigned a CVSS base score of 6.7, which is categorized as MEDIUM severity.

When was CVE-2022-42501 was published on 16 December 2022.

Where can More detailed information about CVE-2022-42501 can be found at the Android Security Bulletin page, <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-42501>

What are the attack scenarios A possible attack scenario for CVE-2022-42501 involves an attacker with system level access to the Android kernel.

What is the CVE ID CVE-2022-20610 refers to a vulnerability found in cellular modem firmware where there is a potential for remote code execution.

How severe The vulnerability CVE-2022-20610 has a base score of 8.8, which categorizes it as HIGH severity according to the CVSS.

On what date The CVE-2022-20610 was published on 16 December 2022.

What versions The affected versions of Android by CVE-2022-20610 are specified as Android kernel, but the exact versions are not specified.

Can you provide More information on CVE-2022-20610 can be found on the Android Security Bulletin at the following URL: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-20610>

Does CVE-2022-20610 require user interaction No, CVE-2022-20610 does not require user interaction for exploitation. The vulnerability can be exploited remotely.

What kind Due to CVE-2022-20610, there is a possible remote code execution attack. Since it involves an out of bounds write, it could lead to arbitrary code execution.

What are the attack scenarios Attack scenarios for CVE-2022-20610 could involve an attacker using specialized techniques to authenticate and exploit the vulnerability.

What is the CVE ID The CVE ID of the vulnerability associated with Pixel cellular firmware is CVE-2022-20609.

Can you describe CVE-2022-20609 refers to a vulnerability in Pixel cellular firmware where there's a possible out of bounds read error.

What is the CVSS score The CVSS Base Score of CVE-2022-20609 is 5.5, categorizing it as a MEDIUM severity issue.

When was CVE-2022-20609 was published on 16 December 2022.

Where can More information about CVE-2022-20609 can be found at the following URL: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-20609>

What Android versions The Android versions affected by CVE-2022-20609 are those that include the vulnerable Android kernel.

What kind CVE-2022-20609 makes it possible to perform a local information disclosure attack by exploiting the out of bounds read error.

Is there a code example Unfortunately, no code example is provided for CVE-2022-20609 in the information given, and without a code example, it is difficult to understand the exact details of the vulnerability.

Could CVE-2022-20609 be exploited remotely CVE-2022-20609 is described as an issue that could lead to local information disclosure. There is no mention of remote exploitation.

Does CVE-2022-20609 require elevated privileges No, exploiting CVE-2022-20609 does not require elevated privileges according to the information provided.

What is the CVE ID The CVE ID for the Pixel cellular firmware vulnerability disclosed in December 2022 is CVE-2022-20608.

Can you describe CVE-2022-20608 refers to a vulnerability in Pixel cellular firmware, where there is a possible out of bounds read error.

How severe The CVE-2022-20608 vulnerability has been given a Base Score rating of 5.5, which classifies it as MEDIUM severity.

Is user interaction No, user interaction is not required to exploit the CVE-2022-20608 vulnerability.

Which versions CVE-2022-20608 affects the Android kernel, but the exact versions impacted are not specified in the provided information.

Where can Official details regarding CVE-2022-20608 can be found on the Android Security Bulletin webpage at <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-20608>

What type The vulnerability in CVE-2022-20608 is a type of out of bounds read error that occurs due to an incorrect bounds check.

Could you provide While a specific code example for CVE-2022-20608 is not provided, a hypothetical scenario might involve an attacker exploiting the out of bounds read error to disclose sensitive information.

What is the CVE ID of the vulnerability in the Pixel cellular firmware is CVE-2022-20607.

Can you describe CVE-2022-20607 refers to a possible out of bounds write vulnerability in the Pixel cellular firmware, which occurs on the Android platform, specifically within the Android kernel.

What is the severity of CVE-2022-20607 has a severity base score of 8.8, which is classified as HIGH.

When was CVE-2022-20607 published on 16 December 2022.

Are there any more information about CVE-2022-20607 can be found at the following URL: <https://source.android.com/security/bulletin/2022-12-16-07>.

What is the Android ID associated with CVE-2022-20607 is A-238914868.

What might happen To exploit the vulnerability CVE-2022-20607, an attacker would need to have LTE authentication credentials.

Could you provide an example An attacker who has obtained LTE authentication credentials could potentially send a specially crafted request to the network.

Is there a code example There is no specific code example provided for CVE-2022-20607, as the details of the vulnerability are not fully disclosed.

What is CVE-2022-20606 refers to a security vulnerability discovered in the SAEMM\_MiningCodecTableWithMetadata.

What is the severity of CVE-2022-20606 has been assigned a Base Score of 4.9, which categorizes it as MEDIUM severity according to CVSS.

On what date was CVE-2022-20606 published on 16 December 2022.

Which versions The Android kernel versions affected by CVE-2022-20606 are not specified in the given information. The vulnerability affects the Android kernel.

What kind of interaction No user interaction is required to exploit CVE-2022-20606. The vulnerability can be exploited remotely.

Can you provide more information More information regarding CVE-2022-20606 can be found at the following link: <https://source.android.com/security/bulletin/2022-12-16-06>.

Are there any attack scenarios While the specific attack scenarios for CVE-2022-20606 are not detailed, a possible scenario could involve an attacker exploiting the vulnerability to gain unauthorized access to system resources.

What are the system execution privileges mentioned in the context of CVE-2022-20606 imply that for an attack, the attacker would need system execution privileges.

What is CVE-2022-20605 is a security vulnerability identified in SAECOMM\_CopyBufferBytes of SAECOMM\_Util.

What versions CVE-2022-20605 affects the Android kernel, but the specific impacted versions are not detailed in the provided information.

How severe The CVE-2022-20605 vulnerability is rated with a base score of 7.5, which is categorized as HIGH severity according to CVSS.

What type of vulnerability CVE-2022-20605 represents an out-of-bounds read vulnerability, which can lead to remote information disclosure.

Is there a patch While the provided information does not specify patch details for CVE-2022-20605, the reference to the vulnerability indicates that a patch is available.

What kind of attack Attack scenarios for CVE-2022-20605 may include a remote attacker exploiting the out of bounds read vulnerability to disclose sensitive information.

What is the CVE ID associated with this vulnerability is CVE-2022-20604.

What kind of vulnerability CVE-2022-20604 describes a vulnerability that could lead to remote information disclosure as a result of a buffer overflow.

Which product The product affected by CVE-2022-20604 is Android, specifically the Android kernel.

What are the versions The exact versions of Android affected by CVE-2022-20604 are not specified in the provided information.

How critical CVE-2022-20604 has been assigned a base score of 5.5, which categorizes it as a MEDIUM severity vulnerability.

When was CVE-2022-20604 published on 16 December 2022.

What is required To exploit CVE-2022-20604, an attacker would not need any additional execution privileges or user interaction.

Where can I find more information More information regarding CVE-2022-20604 can be found at the reference link: <https://source.android.com/security/bulletin/2022-12-16-04>.

Is there an example The provided information does not include a specific code example for the vulnerability CVE-2022-20604.

What are the potential attack scenarios Potential attack scenarios for CVE-2022-20604 could include an attacker sending specially crafted requests to the system.

What is the CVE ID associated with this vulnerability is CVE-2022-20603.

What is the severity of CVE-2022-20603 has been assigned a base score of 7.2, which is categorized as HIGH severity.

On what date was CVE-2022-20603 published on the 16th of December, 2022.

Which system CVE-2022-20603 affects the SetDecompContextDb in the file RohcDeCompContextOfRbld.cpp, which is part of the Android kernel.

What kind of interaction To exploit CVE-2022-20603, an attacker would need System execution privileges.

Does CVE-2022-20603 require user interaction No, user interaction is not necessary for the exploitation of CVE-2022-20603.

Can you provide more information Additional details about CVE-2022-20603 can be obtained from the following link: <https://source.android.com/security/bulletin/2022-12-16-03>.

What are the version details mentioned for CVE-2022-20603 are simply referred to as 'Android kernel,' without specifying a particular version.

Is there an Android ID Yes, the specific Android ID related to CVE-2022-20603 is A-219265339.

What could happen CVE-2022-20603 could lead to remote code execution with System execution privileges, possibly allowing an attacker to gain control of the device.

What is a possible attack scenario A possible attack scenario for CVE-2022-20603 could involve an attacker sending a specially crafted packet to the system.

What is the CVE ID for the vulnerability discovered in the Android kernel is CVE-2022-20602.

What is the severity of CVE-2022-20602 has been assigned a base score of 7.5, which is categorized as HIGH severity.

Which product? The product affected by CVE-2022-20602 is the Android kernel, which is a part of the Android operating system.

Can you provide a reference? Yes, the URL for the reference associated with CVE-2022-20602 is: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-20602>.

What is the associated Android ID given for CVE-2022-20602? The associated Android ID given for CVE-2022-20602 is A-211081867.

When was CVE-2022-20602 published? CVE-2022-20602 was published on 16 December 2022.

What might be the possible attack scenarios for CVE-2022-20602? Possible attack scenarios for CVE-2022-20602 could involve an attacker exploiting the vulnerability in the Android kernel to gain unauthorized access to system resources.

Are there any code examples that demonstrate the exploit? Since CVE-2022-20602 is a security vulnerability, code examples that demonstrate the exploit are typically not provided.

Has CVE-2022-20602 been referenced in an Android security bulletin? Yes, CVE-2022-20602 has been referenced in an Android security bulletin. The details can be found in the bulletin mentioned in the reference URL.

What steps should a user or system administrator take in response to CVE-2022-20602? In response to CVE-2022-20602, a user or system administrator should review the associated security bulletin and consider updating the Android system to a version that addresses the vulnerability.

What is the CVE ID for the reported Android kernel vulnerability? The CVE ID for the reported Android kernel vulnerability is CVE-2022-20601.

How severe is the security issue denoted by CVE-2022-20601? The security issue denoted by CVE-2022-20601 has been assigned a Base Score of 7.5, which is categorized as HIGH severity.

On what date was CVE-2022-20601 publicly disclosed? CVE-2022-20601 was publicly disclosed on 16 December 2022.

Can you provide more information about CVE-2022-20601? Yes, more information about CVE-2022-20601 can be found at the following URL: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-20601>.

What versions of Android are impacted by this vulnerability? CVE-2022-20601 affects the Android kernel, but the specific versions of Android impacted by this vulnerability have not been detailed.

What is the Android ID associated with CVE-2022-20601? The Android ID associated with CVE-2022-20601 is A-204541506.

Are there any specific code examples for CVE-2022-20601? The information provided does not include any specific code examples for CVE-2022-20601. To obtain code examples, one would need to refer to the associated security bulletin or other official sources.

What potential attack scenarios are possible for CVE-2022-20601? While the specific attack scenarios for CVE-2022-20601 are not provided, a vulnerability with a base score of 7.5 suggests a potential for local escalation of privilege.

What is the CVE ID for the vulnerability involving a possible out-of-bounds write due to memory corruption? The CVE ID for the vulnerability involving a possible out-of-bounds write due to memory corruption is CVE-2022-20600.

What type of vulnerability is reported in CVE-2022-20600? CVE-2022-20600 reports a vulnerability that could result in an out-of-bounds write due to memory corruption in the Android kernel.

What are the potential impacts of CVE-2022-20600? The vulnerability CVE-2022-20600 could lead to local escalation of privilege with execution privileges.

What versions of Android are impacted by CVE-2022-20600? CVE-2022-20600 affects the Android kernel, but the specific versions impacted have not been detailed.

What is the recommended action to exploit CVE-2022-20600? To exploit the vulnerability CVE-2022-20600, an attacker would need to have system execution privileges.

What is the severity rating for CVE-2022-20600? CVE-2022-20600 has been rated with a base score of 7.8, which categorizes it as HIGH severity.

When was CVE-2022-20600 published? CVE-2022-20600 was published on 16 December 2022.

Where can additional information about CVE-2022-20600 be found? Additional information about CVE-2022-20600 can be found at the following URL: <https://source.android.com/security/bulletin/2022-12-16#CVE-2022-20600>.

Can you provide more details about CVE-2022-20600? Since CVE-2022-20600 pertains to an out-of-bounds write due to memory corruption, a vulnerable component in the Android kernel is likely involved.

What could be a possible attack scenario for CVE-2022-20600? A possible attack scenario for CVE-2022-20600 might involve an attacker who already has system execution privileges exploiting the vulnerability to gain further access or control.

o 13.83 for Android. This security issue allows children to bypass restrictions by booting the device in Safe Mode, integrity. This means that the vulnerability has a major impact on the confidentiality, integrity, or availability of the affected

re post at <https://seclists.org/fulldisclosure/2023/Jul/12>, a security blog post at <https://sec-consult.com/blog/detail/the-hidden-costs-of-parental-control-apps/> e Mode on their device to remove or circumvent parental restrictions. This can undermine the purpose of the app ability booting their Android device into Safe Mode. In Safe Mode, the Boomerang Parental Control application's r specifically versions before 13.83 for Android. This issue stems from the omission of the `android:allowBackup="false"` d application's functionality. If an attacker obtains the API token through a backup, they could potentially authenticate that the vulnerability presents a moderate level of risk. While it's not as severe as high or critical vulnerabilities, it

13.83.

l update the app to version 13.83 or later, which presumably contains the fix for this vulnerability. Developers should <https://seclists.org/fulldisclosure/2023/Jul/12>, 'https://sec-consult.com/blog/detail/the-hidden-costs-of-parental-control-apps/' e access to a device with the vulnerable Boomerang Parental Control app installed. The attacker could initiate a backup @ for iOS version 1.0.0. The applications are susceptible to a Man-in-the-Middle (MITM) attack because they permit s that the vulnerability poses a moderate risk.

communication between the vulnerable applications and the server. This could lead to information disclosure, session hijacking <https://jvn.jp/en/jp/JVN22546110/> - LINE Official Blog (Japanese): <http://official-blog.line.me/ja/archives/36495925>. ves as a man-in-the-middle between the victim's device running the vulnerable LINE@ application and the application. LINE has addressed the vulnerability by enforcing SSL/TLS for all communications. Developers should release an update discovered in LINE for Android version 5.0.2 and earlier, as well as LINE for iOS version 5.0.0 and earlier. The vulnerability (Common Vulnerability Scoring System) Base Score represents the inherent qualities of a vulnerability. A score of 5.9 falls within the medium severity

communications between the LINE application and its servers, such as being on the same unprotected Wi-Fi network. For more information, see the official LINE blog at <http://official-blog.line.me/ja/archives/24809761.html> and the Japan Vulnerability Notes (JVN) at <https://jvn.jp/en/jp/JVN22546110/> as LINE for iOS version 5.0.0 and earlier.

can compromise the communication between two parties and potentially gain access to or manipulate the information. This security flaw existed in versions starting from 0.6.1 up to but not including version 0.7.0. Due to a verification bypass, an attacker can obtain root privileges on a device with KernelSU installed. If exploited, the attacker could perform privileged actions, alter system files, and forge the digital signature of an APK file. Rather than obtaining the first signature block with an ID of `0x7109871a` (which aligns with the official signature block to ensure that the correct signature block is obtained during the rooting process, and addresses the issues with the official signature block on your device and being cautious not to install unknown or untrusted apps. These workarounds aim to reduce the risk of exploitation. This score indicates that while the vulnerability presents a notable risk, it is not considered to be as severe as a high or critical vulnerability in an application with a specifically designed signing block. The attacker makes the APK available through a third-party repository. This vulnerability is listed in the Android security advisories section, specifically at the link '<https://github.com/tiann/KernelSU/security/advisories/GHSA-492v-336p-982p>' for Android devices, where a paired Bluetooth device could gain access to a long-term identifier of the Android device through Bluetooth.

Common Vulnerability Scoring System.

at the following link: <https://source.android.com/docs/security/bulletin/android-14>. This vulnerability is a

ons bypass, and typically, code examples for such vulnerabilities are not disclosed publicly in order to prevent misinterpretation. An attacker might be able to exploit the permissions bypass to access a long-term identifier of the targeted Android device. This vulnerability is located within `android.view.InputDevice.cpp`, where there exists a potential for executing arbitrary code due to a use-after-free

- all execution privileges.

Open Source Project repository at <https://android.googlesource.com/platform/frameworks/base/+2d88a5c481> android device. This application could exploit the use-after-free vulnerability in android\_view\_InputDevice\_create as provided by the Android Security Bulletin for the respective month. Users should keep their devices' software up to date on the AppHub server, utilizes MQTT protocol to manage device remotely. The protocol uses a hard-coded DES symmetric key (see the Vulnerability Scoring System). This high rating indicates that the vulnerability poses a significant risk and should be

es. The vulnerability stems from the use of a hard-coded DES symmetric encryption key that can be extracted by r  
following URL: <https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>.

network subnet as the HMI device. The attacker sets up a malicious MQTT broker and utilizes the known hard-coded credentials to connect to the broker. Since the broker is not provided. However, a conceptual example would involve intercepting MQTT messages and using the hard-coded credentials to connect to the broker. The attacker would then be able to communicate with the AppHub server. This vulnerability concerns a lack of server authentication when connecting to an MQTT broker. It is rated as HIGH in severity. This indicates that the vulnerability poses a significant risk and should be addressed promptly.

rogue MQTT broker. Once connected, the attacker could send forged messages to the HMI device, which could re  
<https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>

to facilitate message passing. MQTT stands for Message Queuing Telemetry Transport, a lightweight messaging network manipulation techniques like DNS spoofing to reroute the Android Client's broker connection request to the server authentication when connecting to MQTT brokers. This could involve verifying the broker's certificate or device in Kiosk mode to install an arbitrary Android application. This installation can be exploited to access critical 'JM' severity level risk. This highlights the vulnerability as a significant concern that should not be ignored, but it n

he following URL: <https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>.

ce locked in Kiosk mode. The attacker could exploit the vulnerability to install a malicious Android application wit or software developer in the provided security advisory. This may include applying any available patches or upda specific code example that would universally apply. However, exploiting this type of vulnerability generally involve sive information is CVE-2023-45321.

ment. When the application is enrolled using method 1, which involves the user manually entering the server IP address, the user's lack of awareness of the application's vulnerability poses a significant risk; it is relatively easy to exploit and could result in considerable damage or unauthorized access to sensitive information. This vulnerability can be found in the security advisory posted at <https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>.

ITM) attack to intercept network traffic between the vulnerable Android Client application and the server. Since the risk associated with CVE-2023-45321 would include: updating the Android Client application to a fixed version that uses the secured HTTPS protocol instead of HTTP when enrolled with a particular method where the

red to connect to a remote MQTT broker entity.

protocol (HTTP) to retrieve sensitive information, potentially allowing an attacker to intercept this data. Since this is a security issue, it is assigned a severity of High (CVSS 7.5), as the feature that uses HTTP is not configurable by the user.

ch can be found at <https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>.

in attacker could intercept the unencrypted HTTP traffic to gain unauthorized access to sensitive information such as a sample of vulnerable code:

```
public class MqttConnectionManager {
    private String serverIp;
    private String username;
    // ...
}
```

The vulnerability could be exploited by an attacker to unsafely interact with a content-provider exposed by the Android Agent application. This could potentially allow attackers to modify sensitive settings within the application, potentially allowing the latter to modify sensitive settings within the application. While the vulnerability does pose a security risk, its impact may not be as significant compared to higher severity vulnerabilities.

ety Systems at the following URL: <https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>  
oid Client application by interacting with the exposed content-provider of the Android Agent application. This could allow an attacker to bypass the application's permissions. If the Android Agent application has exposed a content provider without proper security controls, this vulnerability allows an unprivileged third-party application to modify the server settings of the affected Android Client application. This high severity rating indicates that the vulnerability represents a significant risk and could lead to severe consequences.

curity and Safety Systems at the following URL: <https://psirt.bosch.com/security-advisories/BOSCH-SA-175607.html>  
ed on the same device as the vulnerable Android Client application. The attacker could exploit the vulnerability by sending a crafted request to the content-provider. An illustration of such a vulnerability could involve an Android broadcast receiver that doesn't properly validate the data received from the content-provider.

19.111.

attacker redirecting users to arbitrary or malicious locations, leading to phishing attacks, malware distribution, and data theft. The attacker could send a crafted request with a modified HTTP Host header to the '/sisqualIdentityServer/core/' endpoint. The modified Host header could be used to redirect the user to a malicious website.

pository link: [https://github.com/omershaik0/Handmade\\_Exploits/tree/main/SISQUALWFM-Host-Header-Injection](https://github.com/omershaik0/Handmade_Exploits/tree/main/SISQUALWFM-Host-Header-Injection)  
ffects versions up to 2023.8.2. The vulnerability allows for arbitrary URL loading within a WebView, which could lead to actions like modifying the page content. This vulnerability represents a significant risk and implies that the vulnerability can have considerable negative impact on the confidentiality, integrity, and availability of the system. Users with these versions are vulnerable to the described security issues.

nion for Android app to version 2023.9.2 or later, where the vulnerability has been patched. There are no known workarounds for this vulnerability. The vulnerability allows an attacker to execute arbitrary JavaScript within the WebView context, which could lead to actions like modifying the page content. The CVE identifier is: 'https://github.com/home-assistant/core/security/advisories/GHSA-jvpm-q3f8-2023-142'. This identifier is used by the GitHub Security Lab to track the vulnerability and its details within the CVE database. The vulnerability is susceptible to an escalation of privileges and/or information disclosure, potentially allowing an attacker to gain unauthorized access to the system's data. This strongly suggests that it could have a significant impact on the affected system's security.

URL: [https://support.hp.com/us-en/document/ish\\_9393937-9393961-16/hpsbgn03870](https://support.hp.com/us-en/document/ish_9393937-9393961-16/hpsbgn03870)

running the Android operating system.

scenario for CVE-2023-5365 could involve an attacker leveraging a flaw in the application's permission handling logic to bypass the application's permissions. The flaw is located in the 'com.lge.lms.things.ui.notification.NotificationManager.java' file. It involves an intent redirection issue that allows an attacker to redirect the application's intent to a malicious application. This vulnerability represents a significant risk and implies that the vulnerability can have considerable negative impact on the confidentiality, integrity, and availability of the system.

the action 'com.lge.lms.things.notification.ACTION', which could result in the unauthorized access to not exported data. The attacker could send a crafted broadcast intent targeting the vulnerable LG ThinQ Service. Due to the intent redirection issue in the LG ThinQ Service, the application could be redirected to a malicious application. This means that exploiting the vulnerability could lead to actions like modifying the page content. The CVE identifier is: 'https://lgsecurity.lge.com/bulletins/mobile#updateDetails'. This identifier is used by the LG Security Lab to track the vulnerability and its details within the CVE database. The vulnerability is susceptible to an escalation of privileges and/or information disclosure, potentially allowing an attacker to gain unauthorized access to the system's data. This strongly suggests that it could have a significant impact on the affected system's security.

activities on the LG device. This could lead to privacy breaches, data theft, or further compromise of the device's security.

nponent is susceptible to a specific type of attack. A malicious application on the same device can capture the input data and send it to a remote server. This vulnerability represents a significant risk and implies that the vulnerability can have considerable negative impact on the confidentiality, integrity, and availability of the system.

nerability.

ervice (DoS), and the potential for code execution.

://nvidia.custhelp.com/app/answers/detail/a\_id/5476.

ed on the same Android device as NVIDIA GeForce Now. The malicious app would process the implicit intent that the application has to access the device's camera. This vulnerability represents a significant risk and implies that the vulnerability can have considerable negative impact on the confidentiality, integrity, and availability of the system.

It has not been made publicly available. Such details are typically reserved for responsible disclosure to avoid widespread disclosure as provided by NVIDIA for GeForce Now for Android. It is also recommended to only install applications from trusted sources.

for Android where a remote attacker can trigger phone calls without user consent. This happens because the DialerActivity component in the com.full.dialer.top.secure.encrypted.activities package is vulnerable to a crafted intent.

The GitHub repository <https://github.com/actuator/com.cutestudio.colordialer/blob/main/dial.gif>, <https://github.com/actuator/cve/blob/main/CVE-2023-42469>, and <https://github.com/actuator/com.full.dialer.top.secure.encrypted.activities/blob/main/CVE-2023-42469> demonstrate how a remote attacker can trigger phone calls without the necessary permissions to make phone calls. This application could construct an Intent object targeting the com.full.dialer.top.secure.encrypted.activities.DialerActivity component with specific data to trigger a phone call with the number 023-42469.

This vulnerability was patched in version 1.0.1.

The attacker can trigger a crafted intent to the com.full.dialer.top.secure.encrypted.activities.DialerActivity component, allowing it to perform actions that are not intended.

For more information, see the GitHub repository <https://github.com/actuator/cve/blob/main/CVE-2023-42469>, <https://github.com/actuator/com.full.dialer.top.secure.encrypted.activities/blob/main/CVE-2023-42469>, and <https://github.com/actuator/com.cutestudio.colordialer/blob/main/dial.gif> at sends an Intent to the com.full.dialer.top.secure.encrypted.activities.DialerActivity component with specific data to trigger a phone call with the number 023-42469.

This vulnerability was patched in version 1.0.1. This flaw allowed a remote attacker to obfuscate security UI by using a crafted intent to trigger a phone call with the number 023-42469, classifying it as 'MEDIUM' severity.

For more information, see the Chrome Releases blog at [https://chromereleases.googleblog.com/2023/09/stable-channel-update-for-desktop\\_12.html](https://chromereleases.googleblog.com/2023/09/stable-channel-update-for-desktop_12.html). The attacker could use a malicious HTML page designed to manipulate the Intents mechanism in Google Chrome on Android. The attacker could showcase how the attack is performed are typically not provided by reputable sources to avoid facilitating exploitation. Users should ensure their Chrome browser is updated to this version or later to protect against this vulnerability.

This issue allowed a remote attacker to spoof security UI via a crafted HTML page.

By using the security UI, making the user believe they are on a secure, legitimate page when they are actually on a malicious page.

Users should ensure their Chrome browser is updated to this version or later to protect against this vulnerability.

For more information, see the Chrome Releases blog at [https://chromereleases.googleblog.com/2023/09/stable-channel-update-for-desktop\\_12.html](https://chromereleases.googleblog.com/2023/09/stable-channel-update-for-desktop_12.html) and <https://crbug.com/1446709>. The attacker could use a malicious HTML page that mimics the look and feel of Chrome's Custom Mobile Tabs security UI. When a user accesses this page via Custom Tabs in Google Chrome on Android. Specifically, it allowed a remote attacker to obscure a permission prompt using a special intent. Users older than 117.0.5938.62 are at risk of this vulnerability and should update their browser to mitigate the issue.

When a user visits this page via Custom Tabs in Google Chrome on Android, they might not see the permission prompt. When a user visits this page via Custom Tabs in Google Chrome on Android, they might not see the permission prompt.

For more information, see the Chrome Releases blog, the Chromium bug tracker, and security advisories from projects such as Fedora and links to victims via email or messaging platforms. These links lead to the crafted HTML pages designed to obfuscate the security UI. Attackers may construct HTML/CSS/JS code that manipulates the visual layout to obfuscate permission prompts. Users should ensure their Chrome browser is updated to this version or later to protect against this vulnerability.

This could potentially allow for a number of malicious actions including stealing of sensitive information and activity that is responsible for displaying web content.

For more information, see the Chrome Releases blog at [https://chromereleases.googleblog.com/2023/09/stable-channel-update-for-desktop\\_12.html](https://chromereleases.googleblog.com/2023/09/stable-channel-update-for-desktop_12.html) and <https://crbug.com/1446709>. Users should ensure their browser is updated to this version or later to protect against this vulnerability.



Following GitHub links: '<https://github.com/actuator/wave.ai.browser/blob/main/poc.apk>', '<https://github.com/actuator/wave.ai.browser/blob/main/poc.apk>'. The malicious app could construct a crafted intent containing Java code. This typically means that the vulnerability can be exploited remotely without requiring user interaction or authentication through version 6.8.0 for Android. This vulnerability allows for Remote Code Execution (RCE) by sending a crafted intent. CRITICAL. This indicates that the vulnerability poses a severe threat, as it could allow attackers to execute arbitrary code on the victim's device.

The exploit is sent to an exported component of the Imou Life application. Since the WebView in the app enables JavaScript execution, these can be found at the following URLs:- <https://github.com/actuator/imou/blob/main/imou-life-6.8.0.md>. The vulnerability exists in versions up to and including 6.8.0. The exploit is sent within the WebView component of the Imou Life application and the direct loading of web content. Combined with the direct loading of web content, this could lead to a remote code execution (RCE) vulnerability.

The exploit is sent to the `com.cordovaplugincamerapreview.CameraActivity` component in certain situations. The exploit is sent with the application's manifest that lacks runtime permission checks for camera access. The exploit is sent to devices. Since it relies on versions of Android without runtime permission checks, an exploit would involve initiating a camera activity. The vulnerability applies only to a specific older version of the Android operating system (Lollipop).

The exploit is sent to the `com.cordovaplugincamerapreview.CameraActivity` component in certain situations. The exploit is sent with the application's manifest that lacks runtime permission checks for camera access. The exploit is sent to devices. Since it relies on versions of Android without runtime permission checks, an exploit would involve initiating a camera activity. The vulnerability applies only to a specific older version of the Android operating system (Lollipop).

The exploit is sent to the `com.cordovaplugincamerapreview.CameraActivity` component in certain situations. The exploit is sent with the application's manifest that lacks runtime permission checks for camera access. The exploit is sent to devices. Since it relies on versions of Android without runtime permission checks, an exploit would involve initiating a camera activity. The vulnerability applies only to a specific older version of the Android operating system (Lollipop).

The exploit is sent to the `com.cordovaplugincamerapreview.CameraActivity` component in certain situations. The exploit is sent with the application's manifest that lacks runtime permission checks for camera access. The exploit is sent to devices. Since it relies on versions of Android without runtime permission checks, an exploit would involve initiating a camera activity. The vulnerability applies only to a specific older version of the Android operating system (Lollipop).

The exploit is sent to the `com.cordovaplugincamerapreview.CameraActivity` component in certain situations. The exploit is sent with the application's manifest that lacks runtime permission checks for camera access. The exploit is sent to devices. Since it relies on versions of Android without runtime permission checks, an exploit would involve initiating a camera activity. The vulnerability applies only to a specific older version of the Android operating system (Lollipop).

The exploit is sent to the `com.cordovaplugincamerapreview.CameraActivity` component in certain situations. The exploit is sent with the application's manifest that lacks runtime permission checks for camera access. The exploit is sent to devices. Since it relies on versions of Android without runtime permission checks, an exploit would involve initiating a camera activity. The vulnerability applies only to a specific older version of the Android operating system (Lollipop).

Cloudflare on GitHub at the URL <https://github.com/cloudflare/advisories/security/advisories/GHSA-5r97-pqv6->

WARP Mobile Client for Android. This flaw allows a malicious app installed on the same device to exploit an Android vulnerability. While this vulnerability poses a considerable risk and should be addressed, but it's not as critical as higher-rated vulnerabilities.

<https://github.com/cloudflare/advisories/security/advisories/GHSA-23rx-f69w-g75c> - <https://developers.cloudflare.com/warp/client-app/> mentions a "malicious app" that is actually malicious. This app can exploit this vulnerability to manipulate the WARP Mobile Client's behavior. Creating an exploit that directly exploits the vulnerability would be unethical and possibly illegal. No publicly available exploit code is available.

the 'Skylark' app for Android and iOS.

rlier.

✓ severity.

Google Play Store page for the Skylark app (<https://play.google.com/store/apps/details?id=jp.co.skylark.app.gusto>), website through another application installed on the user's device. This could happen due to improper authorization or a vulnerability's exploitation, a hypothetical example of exploiting CVE-2023-40530 might involve an attacker sending

rikunabi NEXT' App for Android.

affected app to access an arbitrary website. This could potentially lead to unwanted information disclosure, phishing, and other security vulnerabilities.

en/jp/JVN84820712/

Inerable 'Rikunabi NEXT' app's custom URL scheme handler, leads the app to open an arbitrary website. This could

ure of Google Chrome on Android. Versions prior to 116.0.5845.96 allowed a remote attacker to spoof the conter

[illegible]

On Google Chrome on Android. Specifically, it allowed a remote attacker to bypass the restrictions of the Autofill feature, as a base score of 5.3.

5845.96.

with crafted HTML content designed to exploit the Autofill vulnerability. When a victim visits this website on a vulnerable version of Google Chrome, the browser's Autofill feature is triggered, revealing sensitive information stored in the browser's autofill database. This information can include names, addresses, phone numbers, and even payment details, which are then displayed on the website's form fields. The exploit is demonstrated in a video titled "Exploiting Chrome's Autofill Vulnerability" (https://www.youtube.com/watch?v=8jKd8jKd8jK), which shows the process of crafting the HTML content and the resulting information leakage. The official Google Chrome releases blog post (https://chromereleases.googleblog.com/2023/08/stable-channel-updates.html) provides details on the vulnerability and the steps taken to address it.

ome on Android that allowed a remote attacker to potentially spoof the contents of the Omnibox (URL bar) through Android.

releases.googleblog.com/2023/08/stable-channel-update-for-desktop\_15.html- <https://crbug.com/1454817>- <https://www.bleepingcomputer.com/news/google-chrome/android-exploit-allowing-remote-code-execution/> designed to take advantage of the inappropriate implementation in Chrome's Fullscreen mode on Android. If a u

0 are not typically shared publicly to prevent malicious use. The details of the vulnerability are normally sufficient; a 'use after free' vulnerability within the Offline feature of the browser. This issue, discovered prior to Chrome v...ing that it presents a significant risk and should be addressed promptly.

ability.

sed the renderer process can potentially exploit heap corruption. This could be achieved using a maliciously crafted official Chrome Releases blog post (<https://chromereleases.googleblog.com/2023/08/stable-channel-update-for-d>...iously crafted HTML content. The attacker's goal would be to trigger the use after free vulnerability in the Offline to version 4.2.3504, there was an issue with improper access control that could potentially allow an authenticated...ere the application does not correctly restrict access to a resource from users who are not supposed to have acce:

ion to version 4.2.3504 or later, which contains the necessary fixes to address the vulnerability.

tion. This could happen through gaining access to parts of the app that should be restricted, potentially leading to

: level of authorized access to the vulnerable Intel Unite® application before they can attempt to exploit the impro

following URL: <http://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00932.html>

Intel Unite® Android application to version 4.2.3504 or above to ensure they are protected from this vulnerability

user to potentially enable information disclosure via local access.

s of version v23.02.07.

rw.intel.com/content/www/us/en/security-center/advisory/intel-sa-00862.html.

administrative or physical access to the device, exploiting the incorrect default permissions to access sensitive info...udio calls via Nextcloud on Android devices.

party apps to trick the Talk Android app into writing files outside of its intended cache directory.

atch for this specific security issue.

tcloud Talk Android to version 17.0.0, which contains the necessary patch.

tHub pull request (<https://github.com/nextcloud/talk-android/pull/3064>), a HackerOne report (<https://hackeron>...same device as the vulnerable Nextcloud Talk Android app. This malicious app could exploit the unprotected inter

oid. It allowed a remote attacker to leak cross-origin data via a crafted HTML page due to the failure to properly e

Releases blog post, the Chromium bug tracker, and the Fedora project's mailing list archives, as indicated in the re...age designed to exploit the vulnerability in Custom Tabs to leak sensitive information from a different origin. Sinc...r Google Chrome app on Android to version 115.0.5790.98 or later, which includes the necessary security fixes for...potentially harmful. Responsible disclosure guidelines suggest that specific details of an exploit should not be pul...should update their browser to this version or later to mitigate the risk from this vulnerability.

pecifically, it affects versions up to and including 2.74.58 of the application. This issue allows a remote attacker to...lnerability Scoring System) scale. This indicates that the vulnerability poses a significant risk if exploited.

The links for more information are: '<https://github.com/actuator/cve/blob/main/CVE-2023-36351>' and '<https://git>

intent that exploits the WebViewActivity component in the ViHealth app. This could result in the execution of arbitrary code on the device, potentially leading to data theft or device compromise. App developers should release an update to the latest version, which presumably contains patches for the vulnerability. App developers should release an

element in Intents allowed a remote attacker to bypass the same origin policy using a crafted HTML page. This security

230- [https://chromereleases.googleblog.com/2023/01/stable-channel-update-for-desktop\\_24.html](https://chromereleases.googleblog.com/2023/01/stable-channel-update-for-desktop_24.html)- <https://lists.ubuntu.com/archives/ubuntu-security/2023-01/msg00009.html> for Android.

HTML page that takes advantage of the intent handling in Chrome on Android. The attacker could trick a user into visiting a malicious website that exploits vulnerabilities like CVE-2022-4926 are usually not provided. Such examples would be risky as they could aid a remote attacker in compromising the security of Google Chrome.

on Android. Specifically, a remote attacker could obscure the full-screen notification by using a crafted HTML page that takes advantage of the intent handling in Chrome on Android. By doing this, the attacker could potentially

Chrome Releases blog at [https://chromereleases.googleblog.com/2022/06/stable-channel-update-for-desktop\\_24.html](https://chromereleases.googleblog.com/2022/06/stable-channel-update-for-desktop_24.html).

network interception of network traffic.

network segment, could potentially exploit the vulnerability described in CVE-2023-32427 by intercepting unencrypted network traffic.

[support.apple.com/en-us/HT213833](https://support.apple.com/en-us/HT213833)

Idle (MitM) attack on an unsecured network, such as public Wi-Fi. The attacker could intercept network traffic from a user's device, such as Apple Music, to a compromised Wi-Fi network, an attacker could intercept the traffic using tools like Wireshark

user's contacts due to inadequate permission checks.

issue in terms of potential impact.

at the following URL: <https://support.apple.com/en-us/HT213833>

installed on a user's device, could bypass the necessary permission checks and illegitimately gain access to the user's data. This is not advisable. Moreover, since the issue has been fixed, the code example would unlikely be applicable to the product.

023-33743.

the Android Debug Bridge (adb) service is improperly accessible.

[security.com/files/173764/RoomCast-TA-2400-Cleartext-Private-Key-Improper-Access-Control.html](https://security.com/files/173764/RoomCast-TA-2400-Cleartext-Private-Key-Improper-Access-Control.html).

Bridge (adb) to gain unauthorized access to the device, potentially allowing for data extraction, device manipulation, and other malicious activities. An attacker connecting to the TeleAdapt RoomCast's Android Debug Bridge (adb) service over the network using the following code example could exploit this as a spoofing vulnerability. It has a base score of 4.3, indicating a medium level of severity.

to impersonate another entity within the system or create a falsified user interface element.

website at <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-38173>.

r interface element in the browser that appears legitimate to the user. An attacker might use this to trick the user  
 rsion 1.4. This vulnerability allows for an Authentication Bypass where the lock's opening mechanism, which uses  
 nication between the lock and the controlling device and capture the commands. These commands could later be

ured by the MojoBox Digital Lockbox. Using a BLE sniffing tool, the attacker captures the BLE signals sent from the  
urity advisories and researchers. These include a technical write-up available at '<https://packetstormsecurity.com/>

are update patching the vulnerability. Users should apply the update as soon as it becomes available. Until then, Oracle Fusion Middleware. Specifically, it affects the component known as the Android Mobile Authenticator App. Via the physical communication segment attached to the hardware where the Oracle Mobile Security Suite is running, CVE-2015-3535 indicates that the vulnerability presents a moderate level of risk. The score is based on the fact that the vulnera

access to all Oracle Mobile Security Suite accessible data. This includes the potential leakage of sensitive information to the latest version that addresses this vulnerability, which is version 11.1.2.3.1 or later. Additionally, limiting the exploit method have not been publicly disclosed, as is common with security issues to prevent further exploitation of network infrastructure, such as a LAN port in the same network segment where the vulnerable Oracle Mobile Security Suite is installed. It has been classified as a Tampering Vulnerability.

CVSS v2 Base Score: 7.5 (High) - This vulnerability is rated as a High level threat according to the scoring system used to rate the severity of vulnerabilities.

(MSRC) website at this URL: <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-36888>  
 unauthorized modifications to system files or configurations of the browser, potentially leading to altered behavior

manipulation of sensitive information, a compromised user session, or unauthorized actions executed in the browser on Android devices. Specifically, this vulnerability is a potential zip path traversal issue existing in versions of the library according to the scoring system used to rate the severity of security issues.

Mobile Security: <https://security.samsungmobile.com/serviceWeb.smsb?year=2023&month=07>.

p file that, when processed by the vulnerable Calendar application, exploits the zip path traversal issue to extract paths within a zip archive to escape the intended directory structure upon extraction. By including relative path CVE-2023-28364.

curred prior to version 1.52.117. This vulnerability was due to the browser's built-in QR scanner, which automatic  
version 1.52.117, the behavior of the built-in QR scanner has changed. Now, after scanning a QR code, the Brave f

is available at the following URL: <https://hackerone.com/reports/1946534>

and generate a QR code that redirects to a malicious website impersonating a legitimate service. Before the fix, if a user is on an Android (versions 10.4.5 and earlier) and iOS (versions 10.4.2 and earlier). The issue is that the application uses hard-coded credentials. This suggests that the vulnerability represents a significant risk, but it may not be as critical as high or critical.

ates (JVN) webpage (<https://jvn.jp/en/jp/JVN32739265/>), and details about the affected applications can be viewed where the NewsPicks app is installed. By extracting the hard-coded credentials within the app, the attacker could an

heck of shared memory.

s before 23.02.

relevant GitHub release page for Arm NN Android-NN-Driver version 23.02. The URLs are as follows:- <https://developer.arm.com/nn-driver/android/23.02>. The vulnerability might gain access to sensitive data or cause memory corruption. For example, an application using the driver might be able to provide exploit code that could facilitate attacks. However, developers of the driver would likely expect this, and the driver version 23.02, which addresses the reported out-of-bounds read and write issue.

an LED Message Cup. An unauthenticated attacker within Bluetooth Low Energy (BLE) proximity can remotely control the cup.

actor/7-Eleven-Bluetooth-Smart-Cup-Jailbreak- <https://github.com/actuator/cve/blob/main/CVE-2023-34761>. The attacker can connect to the cup without authentication and send messages to the cup's display that fixes the BLE vulnerability, requiring authentication for connections and enforcing stronger message validation.

sufficient UI in Android's applyRemoteView method. This vulnerability could lead to local information disclosure.

specifically at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

a vulnerability in the applyRemoteView method of NotificationContentInflater.java to hide a foreground service notification.

associated with CVE-2023-21236.

the missing bounds check can lead to a possible out-of-bounds write. This security flaw could lead to a local escalation of privileges. The severity of this vulnerability is moderate, indicating that the vulnerability poses a moderate threat level, and while it may not be as critical as high severity vulnerabilities, it still warrants attention.

[e.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01)

affected versions of Android are not explicitly mentioned but it is associated with Android 13.

discussed in CVE-2023-21236.

An attacker could potentially execute arbitrary code with System privileges, which may lead to taking full control of the device.

out-of-bounds check. It occurs in the SAEMM\_RetrieveTailList function of the SAEMM\_ContextManagement.c file, leading to a local information disclosure. It would be advised to check specific vendor announcements or the Android Security Bulletin for more details on this vulnerability and that it poses a significant risk.

CVE-2023-21226.

This vulnerability is more critical as it could be exploited remotely without user knowledge.

bulletin page for Pixel devices, available at <https://source.android.com/security/bulletin/pixel/2023-06-01>.

request to the vulnerable component within the Android kernel. Since it allows for remote information disclosure, this is a significant security issue. The vulnerability is associated with CVE-2023-21225.

lock screen on Android due to a failure to lock the display power. This vulnerability could lead to a local escalation of privileges.

which can be accessed at <https://source.android.com/security/bulletin/pixel/2023-06-01>.

security issue within the Android kernel, indicating that multiple versions could be vulnerable pending further clarity from the Android team.

erable Android device. The attacker could utilize a malicious application or a sequence of actions that exploit the  
Android kernel.  
24.

in/pixel/2023-06-01.

oad that when processed by the affected component in the Android kernel, would trigger an out of bounds read d  
rovide an accurate code example. But a heap buffer overflow typically occurs when data is written to a buffer and

nds check in LPP\_ConvertGNSS\_DataBitAssistance of LPP\_CommonUtil.c. This vulnerability could lead to remote

.android.com/security/bulletin/pixel/2023-06-01.

ck in the function LPP\_ConvertGNSS\_DataBitAssistance of file LPP\_CommonUtil.c that leads to an out of bounds r  
sensitive information from memory beyond the intended bounds. Since user interaction isn't required for exploita  
ossible out-of-bounds write due to a missing bounds check. This vulnerability can lead to local escalation of privi

ocated at <https://source.android.com/security/bulletin/pixel/2023-06-01>.

entifier, Android ID: A-266977723.

ileges at the System level. This means the attacker could potentially gain control over the affected device and ex  
he device. This means the vulnerability is not remotely exploitable but could be used as part of a chained attack a  
; vulnerability without requiring any action from the user.

e a function in storage.c that fails to check if a buffer index is within proper bounds before writing to it. For instan  
ss to the device. They could write to protected areas of memory, leading to corruption of critical data structures, k  
il use of an unencrypted transport over cellular networks due to an insecure default value. This weakness could le  
the security fix was applied. It is essential for users and administrators to refer to the official Android security bul  
gorized as HIGH severity.

ecurity/bulletin/pixel/2023-06-01 This reference is part of the official Android security bulletin and provides spec  
l exploit the weakness to intercept unencrypted data sent over cellular networks, gaining access to sensitive infor  
traffic from an affected device. As the vulnerability stems from the use of an insecure default value for transport  
nformation transmitted over cellular networks. This could involve the interception and disclosure of personal dat  
organizations are encouraged to review the details and apply necessary updates or mitigations to protect against i

er cellular networks due to an insecure default value in the Android kernel. This vulnerability can lead to remote i

219.

lletin/pixel/2023-06-01.

een disclosed in the provided information.

rypted cellular network traffic from a compromised Android device. Since the data is not encrypted, the attacker can read the data on an Android, where there is a possible out of bounds read due to unsafe deserialization. This could lead to local information disclosure. No user interaction is required to carry out the attack.

Available at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>  
An attacker can obtain System privileges on an Android device running version 13. The application might exploit this vulnerability to access sensitive information. The responsible disclosure of CVE-2023-21214 should not include exploit code, as it aims to inform about a vulnerability in the p2p\_iface.cpp file of Android-13. It affects the system by potentially disclosing local information to an attacker.

The vulnerability is located in the function of 'sta\_iface.cpp', where a possible out of bounds read could occur due to a missing bounds check. This flaw is categorized as MEDIUM severity.

Available at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

This vulnerability is located in the file p2p\_iface.cpp. An attacker can exploit this vulnerability on the affected Android device. No user interaction is necessary for the exploitation. An attacker with System privileges on the Android device, perhaps having achieved this through a previous exploit. The attacker could then exploit the vulnerability to read sensitive information. This allows for a possible out of bounds read due to a missing bounds check. This could potentially result in local information disclosure. Sensitive information to the wifi server could be read by an unauthorized user. The vulnerability is of a medium severity with a base score of 5.5. This implies that the attacker requires a high level of access, such as root access or control over the device. The vulnerability can be found in the Android security bulletin located at <https://source.android.com/security/bulletin/pixel/2023-06-01>. The bulletin provides the latest security updates as released and detailed by the Android security bulletin to protect against this vulnerability. The score indicates a moderate level of risk associated with the vulnerability, suggesting that while it has potential for exploitation, it is not among the most critical vulnerabilities. An attacker with System privileges on an Android-13 device could exploit this vulnerability to perform an out of bounds read in the wifi module. This is a possibility of an out of bounds read because of a heap buffer overflow. This vulnerability could potentially lead to local information disclosure. Keeping the device up to date with these patches will help protect against this vulnerability.

There is a possibility of an out of bounds read because of a heap buffer overflow. This vulnerability could potentially lead to local information disclosure.

Available at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

The vulnerability can be done silently.

An attacker can exploit this vulnerability to read sensitive information from a device running Android 13. Since no user interaction is needed, this is a high severity vulnerability.

The vulnerability is located in the file p2p\_iface.cpp, where there is a potential out of bounds read caused by improper input validation. This vulnerability could lead to local information disclosure.

Available at the following URL: [source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

An attacker can exploit the out of bounds read vulnerability to access sensitive information from the system memory without any user interaction. This affects Android version 13. This vulnerability involves a possible out of bounds read due to unsafe deserialization, which poses a significant risk and should be addressed, but it is not among the most critical vulnerabilities that would have a high impact. All compatible devices running this version of Android should be updated to mitigate the risk of exploitation. An attacker with System privileges on an Android device. This means that the vulnerability cannot be exploited by a standard user or remotely without System privileges.



exploited by an attacker with System execution privileges without requiring any action from the user.

as about vulnerabilities disclosed around that time.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01). This link takes you to the Android security bulletin page where details a vulnerability in the `sta_iface.cpp` file that allows an attacker to write a malicious application or script that takes advantage of the out of bounds read in the `sta_iface.cpp` function.

/.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

CVE-2023-21208 by sending a specially crafted input to the `setCountryCodeInternal` function. This might lead to reading

here there is a possible out of bounds read because of a missing bounds check. This could potentially lead to local

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

installed on the device exploiting the out of bounds read vulnerability in the `initiateTdsSetupInternal` function within

the `sta_iface.cpp` file, which is part of the Android operating system. This vulnerability is due to unsafe deserialization and is rated as MEDIUM severity according to the Common Vulnerability Scoring System (CVSS).

pages on the device. This means that the attacker would need to have a level of access that is typically reserved for root access to exploit the vulnerability without any input or action from the user.

bulletin webpage. The relevant link for the June 2023 bulletin is: <https://source.android.com/security/bulletin/pixel/2023-06-01>

System execution privileges on the device, either through another exploit or by leveraging system-level applications. The bulletin provides specific code examples for such vulnerabilities. The best practice is to check the official Android Open Source Project (AOSP) for updates. The relevant CVE is CVE-2023-21205.

due to unsafe deserialization.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01)

device exploiting the unsafe deserialization vulnerability in '`sta_iface.cpp`' without requiring any additional privileges. The vulnerability is specifically in the function '`startWpsPinDisplayInternal`', which leads to a possible out of bounds read.

the system where there is a potential out of bounds read due to a missing bounds check. If exploited, this could lead to local information disclosure within the wifi server of an Android device.

Vulnerability.

ce. This means the attacker would need an enhanced level of access or control over the system.

source.android.com/security/bulletin/pixel/2023-06-01

cal concerns, an attacker with System execution privileges could theoretically exploit this out of bounds read vulnerability on the wifi server, possibly including network credentials, session tokens, or data that should be restricted. This could

ace.cpp due to improper input validation. This could permit a local escalation of privilege with System execution p

source.android.com/security/bulletin/pixel/2023-06-01

entity with System execution privileges to execute arbitrary code without any user interaction.

potentially exploiting the vulnerability to read out-of-bounds memory, which could lead to extracting sensitive information out of bounds read in the file btm\_devctl.cc of the Android operating system.

btm\_devctl.cc in the Android operating system.

function of btm\_devctl.cc, leading to a potential out of bounds read. This vulnerability could result in local information leakage on the vulnerable Android device.

bulletin webpage at <https://source.android.com/security/bulletin/pixel/2023-06-01>.

leakages on an Android device, utilizing the vulnerability in btm\_delete\_stored\_link\_key\_complete of btm\_devctl.cc to

documentation is CVE-2023-21201.

function within 'btif\_sdp\_server.cc' file of Android's Bluetooth implementation. There is a missing null check that causes a crash of severity.

source.android.com/security/bulletin/pixel/2023-06-01'.

Bluetooth packets to the target device running Android-13. These packets may be designed to trigger the vulnerability

out of bounds read issue in the on\_remove\_iso\_data\_path function within the btm\_iso\_impl.h file, which could potentially

resulting in a medium level of severity.

in a way that triggers the vulnerable code in on\_remove\_iso\_data\_path. Since the vulnerability stems from improper input validation, it is not a zero-day vulnerability. For this specific vulnerability, information related to the fix would be available in the bulletin or advisories released by the vendor. For this specific vulnerability, information related to the fix would be

available in the bulletin or advisories released by the vendor. For this specific vulnerability, information related to the fix would be available in the bulletin or advisories released by the vendor. The provided reference link is a good starting point: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

a missing bounds check that can lead to a possible out of bounds read. This vulnerability could result in local information leakage on the vulnerable Android device.

indicates that the vulnerability presents a moderate level of risk.

source.android.com/security/bulletin/pixel/2023-06-01.

Android device. User interaction is not required for the exploitation of this vulnerability.

if bounds read to disclose sensitive information from the system memory without any user interaction. Priority patches provided by the vendor, applying proper input validation to prevent out of bounds reads, and restrict named btif\_sdp\_server.cc. It is classified as an out-of-bounds read issue that arises from a missing bounds check. The score indicates that the flaw presents a notable risk, though it may not be as critical as higher-scored vulnerabilities. The risk until a security patch is applied that addresses the vulnerability.

ories to inform users and enable vendors to issue updates and patches.

wing URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>. This bulletin typically contains description of the vulnerability without the need for the user to take any specific action, making it more critical for affected systems. The vulnerability can be exploited by an unprivileged process, making it easier for an attacker to leverage this flaw for information on an Android 13 device. This app could leverage the vulnerability in the Bluetooth stack (btif\_sdp\_server.cc) to perform a common practice for vulnerabilities like CVE-2023-21198 to be addressed in subsequent security updates following the btif\_acl.cc in Android-13. The flaw involves a potential out of bounds read due to an improper bounds check, which

states that the vulnerability is considered to pose a significant threat, requiring urgency in patching or remediating. An attacker could potentially access sensitive information from a device without requiring additional privileges or user interaction and does not require any user interaction, making the vulnerability particularly concerning.

ally at the provided reference URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

Bluetooth messages to an Android-13 device. Due to the out of bounds read, the attacker could potentially gain access to sensitive information within the 'btm\_ble\_batchscan.cc' file of the Android Bluetooth server. It is a possible out of bounds read caused by the Android device.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01)

process on an Android device crafting a series of Bluetooth packets to trigger an out of bounds read in the affected Bluetooth stack (btm\_ble\_gap.cc). It is caused by an incorrect bounds check, which could potentially lead to an out of bounds read. This

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

the system firmware exploiting the vulnerability to perform an out of bounds read. This could lead to the disclosure of sensitive information. The vulnerability is located in the btif\_sdp\_server.cc due to a missing bounds check.

MEDIUM.

[e.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

ng the vulnerability to read memory contents they should not be able to, leading to local information disclosure. It is within the VideoFrame component. It involves a potential abort triggered by an integer overflow that could lead to a HIGH severity category. This means that it represents a significant risk that should be addressed promptly. It can be exploited remotely by an attacker without any additional actions from the user.

ces at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

hen processed by the VideoFrame component in an Android device running version 13, triggers an integer overflow (CVE-2023-21192).

thods that are not enabled due to improper input validation. This vulnerability could lead to a local escalation of privileges.

[droid.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

input validation vulnerability to enable an input method that is not usually permitted. Since the vulnerability allows a local user to enable an input method that is not usually permitted, it is assigned a local severity of HIGH (CVSS 3.1: Local Score = 7.5). This vulnerability is tracked as CVE-2023-21191.

NotificationManagerService.java within the Android operating system.

need for additional execution privileges.

<https://source.android.com/security/bulletin/pixel/2023-06-01>.

fication function of NotificationManagerService.java.

veloping a malicious application that sends notifications with specially crafted data intended to bypass the user's lock screen.

3. There exists a possibility for a remote device to disable encryption without causing the connection to terminate. This vulnerability is assigned a remote severity of MEDIUM (CVSS 3.1: Remote Score = 5.4).

[com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

with the affected Android device and then exploiting CVE-2023-21190 to disable encryption. With encryption turned off, an attacker could potentially access data stored on the device. This vulnerability is assigned a local severity of MEDIUM (CVSS 3.1: Local Score = 5.4).

ng to CVSS (Common Vulnerability Scoring System) ratings.

the device or an application already installed on the device could exploit this vulnerability to gain elevated privileges. For more information, see the full details on this vulnerability on the Android-13 page: <https://source.android.com/security/bulletin/pixel/2023-06-01>. The details are not detailed in the summary of the vulnerability.

er into performing certain actions that lead to the bypassing of lock task mode, allowing the application to gain elevated privileges.

ossible out of bounds read could occur due to a heap buffer overflow. This security flaw could potentially lead to a local escalation of privileges.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

the heap buffer overflow to read sensitive information from beyond the allocated heap buffers. This could be used as a proof of concept code example without access to the proprietary source code of Android. However, a general example of a heap buffer overflow is CVE-2023-21187.

Android-13, where due to a logic error, there's a potential way to bypass the Setup Wizard. This vulnerability could

be exploited to gain root access, potentially gaining access to additional system resources or sensitive data, and performing unauthorized actions. For more information, see the Android security bulletin for June 2023, specifically at <https://source.android.com/security/bulletin/pixel/2023-06-01>.

Other vulnerabilities in secure coding, such as validating external inputs, employing proper error handling, and not exposing sensitive information, could allow an attacker to gain root access or temporary access to a device being able to bypass the setup wizard, potentially pre-configuring the device for

an out of bounds read caused by a missing bounds check in the LogResponse of Dns.cpp.

Original URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

an out of bounds read in the LogResponse function of Dns.cpp on an Android device running version Android-13. Because this is a security issue, it is not disclosed by the vendor. For CVE-2023-21186, users should refer to the Android security bulletin for June 2023, and apply the fix.

There is a missing permission check. This weakness could potentially be exploited to cause a local escalation of privileges. The vulnerability is considered to have significant impact and severity, making it a serious concern that should be addressed.

Original URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

Exploiting the missing permission check to gain elevated privileges on the system. This might allow the attacker to perform actions that require elevated privileges.

The `getPackagesForAllUsers` method within the `CarrierPrivilegesTracker.java` file. This error could potentially allow a local attacker to gain elevated privileges.

Android security bulletin page: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

Exploiting the logic error in the `getCurrentPrivilegedPackagesForAllUsers` method to bypass permission checks and gain access to sensitive information. The Android security bulletin suggests that the issue has been addressed by Android in a security update. Users should refer to the Android security bulletin for more information.

<https://source.android.com/security/bulletin/pixel/2023-06-01>.

The app is still in the background.

It is possible to read NFC tag data from an Android device while the corresponding app is not actively in use by the user.

The severity is MEDIUM.

No user interaction is required for exploitation.

<https://source.android.com/security/bulletin/pixel/2023-06-01>

Out of bounds memory.

The application could trigger the out of bounds read vulnerability in the Exy

Information disclosure.

<https://source.android.com/security/bulletin/pixel/2023-06-01>.

The vulnerability is triggered by a rogue system process attempting to exploit the heap buffer overflow to read out of bounds memory, which relates to the Android operating system. It involves a possible out-of-bounds read due to a heap buffer

overflow leading to information disclosure. The vulnerability does not require user interaction for exploitation, meaning that an attacker can exploit the potential impact of the vulnerability and the ease with which it could potentially be exploited.

For more details, see the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>

The vulnerability is caused by a lack of proper validation of user input or buffer sizes when parsing XML data, leading to an out-of-bounds read or heap buffer overflow that, when processed by an application using the vulnerable XML parsing function, triggers a heap buffer overflow

and leads to information disclosure from XML files in Android.

The vulnerability occurs when parsing security parameters for WiFi encryption protocols in XmlUtil.java. Due to this flaw, there is a potential

for information disclosure, which is detailed in <https://source.android.com/security/bulletin/pixel/2023-06-01>.

An attacker can use an application or script to exploit the improper cryptographic functions used in XmlUtil.java to bypass WiFi encryption

and potentially result in local information disclosure.

ages on the affected device; no user interaction is required for exploitation.

Source: <https://source.android.com/security/bulletin/pixel/2023-06-01>

This vulnerability allows an attacker to trigger the vulnerable code path in KeyUtil.cpp. As the race condition occurs, the attacker exploits it

This vulnerability isn't readily available. The issue resides in the installKey function within KeyUtil.cpp, where a race condition exists in the Android operating system. This vulnerability allows for the potential disclosure of information about the app and user data. A race condition in the Android system could lead to local information disclosure without the necessity of extra execution privileges.

Source: <https://source.android.com/security/bulletin/pixel/2023-06-01>

This vulnerability allows an attacker to bypass the KeyboardShortcuts function of WindowManagerService.java without the appropriate permissions. This could allow

This vulnerability impacts the utils.rs file, impacting Android version 13. It involves potential resource exhaustion, leading to a local denial of service. This could mean that legitimate users may be unable to use their device effectively due to resource exhaustion caused by the vulnerability. This vulnerability exists within the Android operating system, typically reserved for critical system components and applications. This vulnerability could trigger the vulnerability without any input from the user.

This score reflects the level of impact and ease of exploitation associated with the vulnerability.

Android Security Bulletin: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

This vulnerability could maliciously exhaust system resources by repeatedly calling the affected function in the utils.rs file, leading to a denial of service. This vulnerability exists in the DataUsageSummary.java of the Android operating system. Specifically, this vulnerability allows a guest user to bypass a permissions check that the vulnerability poses a significant threat if exploited and should be addressed by applying patches or mitigations. An attacker could potentially exploit the vulnerability without requiring the device user to perform any action, making it more difficult to detect. If this vulnerability were formally notified about the details of the vulnerability.

Source: <https://source.android.com/security/bulletin/pixel/2023-06-01>. This page provides details of the vulnerability. This vulnerability allows an attacker to bypass a permissions check to access a device running Android-13. The attacker could exploit the permissions bypass in the DataUsageSummary.java of the Android operating system. Specifically, this vulnerability allows a guest user to bypass a permissions check that the vulnerability poses a significant threat if exploited and should be addressed by applying patches or mitigations. An attacker could potentially exploit the vulnerability without requiring the device user to perform any action, making it more difficult to detect. If this vulnerability were formally notified about the details of the vulnerability.

Source: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

This vulnerability could be exploited by bypassing permission checks in the `BillingCycleSettings.java` file. This could be done programmatically by creating a malicious app that addresses this vulnerability. Device administrators should also restrict the ability of guest accounts to install apps.

This vulnerability bypasses a permissions check. This flaw makes it possible for an attacker to potentially learn about an admin user's network activity.

URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

he device which takes advantage of the missing permission check in DataUsageList.java to access and disclose info ageList.java, which can be done through a malicious application. This vulnerability can be exploited without any a s CVE-2023-21172.

m/security/bulletin/pixel/2023-06-01.

ileges.

ons bypass in the WifiCallingSettings.java functions to change the calling preferences for the admin user. Since nc

side channel information disclosure found in the verifyInputEvent function of InputDispatcher.cpp on Android. T  
EDIUM.

in at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

L171.

running on the Android device that leverages the side channel information disclosure vulnerability in InputDispa

It is a possible out of bounds read caused by a missing bounds check. This vulnerability could potentially lead to lo

om/security/bulletin/pixel/2023-06-01.

device with System execution privileges. The app could exploit the out of bounds read vulnerability in Composer  
les available. Generally, out of bounds read vulnerabilities occur when a program attempts to read data past the e  
21169.

where a missing bounds check could lead to an out of bounds read. This security flaw could result in local informat  
EDIUM severity.

source.android.com/security/bulletin/pixel/2023-06-01.

ges could craft a malicious application or use an existing vulnerability to execute code within the context of a priv  
its exploitation are typically not released to the public due to responsible disclosure practices. To understand the



: in the convertCbYCrY function of ColorConverter.cpp.

source.android.com/security/bulletin/pixel/2023-06-01.

er might be able to read data from out of bounds memory locations that they should not normally have access to. icePolicyManagerService.java file in Android. This vulnerability could lead to a denial of service attack on the Syst

ability.

<https://www.pwnagotchi.com/security/bulletin/pixel/2023-06-01>

the vulnerability without any user input.

denial of service without elevated access.

Android device. Given that no user interaction or additional privileges are required, the application could program

potentially lead to local escalation of privilege with System execution privileges needed.

is not needed for exploitation.

updates for June 2023.

source.android.com/security/bulletin/pixel/2023-06-01.

device could exploit the out of bounds write vulnerability in the Parse function of simdata.cpp. By manipulating the

rich relates to a possible out of bounds read due to a heap buffer overflow. This issue could result in local informa

[.com/security/bulletin/pixel/2023-06-01](https://www.mcafee.com/security/bulletin/pixel/2023-06-01)

update that addresses this vulnerability.

id device without needing any additional privileges.

security bulletin, which should include patches for the vulnerability.

permissions to operate on the device, leveraging the heap buffer overflow to read sensitive information from the

the Android kernel. This vulnerability could lead to a possible out-of-bounds write due to a missing bounds check and

that the attacker would need to have System execution privileges in the first place, but no user interaction is required.

ed to the Android kernel. Users should check the Android security bulletins and updates regarding kernel-related

in page, particularly the one for Pixel devices, at the following URL: <https://source.android.com/security/bulletin>.

as 6.7, which categorizes it as a medium-level threat.

the advantage of the missing bounds check in the `simdata.cpp` `Parse` function to write data out-of-bounds. This could

he 'encode' function of 'miscdata.cpp'. It involves a heap buffer overflow that could potentially lead to an out-of-b

affected is not specified in the provided information. For the most accurate and up-to-date information on affected

to the Common Vulnerability Scoring System (CVSS).

Execution privileges can exploit the vulnerability without any user interaction.

suggests that the attacker would have to have high-level permissions, which are typically reserved for system app

[/source.android.com/security/bulletin/pixel/2023-06-01.](https://source.android.com/security/bulletin/pixel/2023-06-01)

has already gained System execution privileges on an Android device. The attacker could leverage the heap buffer

cpp is CVE-2023-21157.

de function of wlanata.cpp. This vulnerability could potentially result in local escalation of privilege, where the a  
privileges.

ed in the CVE.  
bility.

/source.android.com/security/bulletin/pixel/2023-06-01.

iting data out of bounds on the heap through the encode function of wlanata.cpp. This could lead to memory co  
of a heap buffer overflow might look like this: ``cpp#include <cstring>int main() { char \*buffer = new char[10]; //  
radioNode within the file protocolmiscbulider.cpp. It involves a possible out-of-bounds read due to improper input  
ut-of-bounds read caused by improper input validation of certain operations in the Android OS.

ed device. User interaction is not required for this vulnerability to be exploited.  
nerability.

d in the provided information.

hat sensitive data processed by the modem could potentially be accessed by an unauthorized entity without the u  
r patches or updates that address CVE-2023-21156. They should ensure that their systems are updated to the late  
ccessed at <https://source.android.com/security/bulletin/pixel/2023-06-01>. This resource typically provides detail  
em. Specifically, in BuildSetRadioNode of protocolmiscbuilder.cpp, there is a possible out of bounds read due to a

ple to this exploit have not been specified in the given information.  
y an attacker without the need for any user involvement.

om/security/bulletin/pixel/2023-06-01

ent of Android's operating system. The impact of this vulnerability includes potential local information disclosure,  
se already present on the local system where the affected Android kernel is running.  
ld involve an attacker leveraging the out-of-bounds read vulnerability by crafting a malicious input to the BuildSet

er.cpp within the Android kernel. It is a possible out of bounds read due to a missing bounds check which could lea  
is means an attacker would need to have a high level of access, potentially equivalent to that of the operating sys  
r could potentially exploit this vulnerability without any active participation from the user.

l. The specific versions impacted by this vulnerability are noted in the CVE details but are not listed in the provide  
icates that the vulnerability represents a moderate risk, and while it shouldn't be ignored, it may not be as critical

ded in the references: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

as gained system execution privileges on an affected Android device. The application could exploit the out of bou  
the imsservice.cpp file, which is a part of the Android operating system. It is an out of bounds read issue caused k  
Android system. Due to the absence of a proper bounds check in the vulnerable function, an attacker can cause an  
an attacker with system execution privileges to potentially gain higher-level access and control over the affected /  
nvolved in the issue, but the exact version numbers vulnerable to this bug have not been provided in the CVE des  
according to the score provided.

ally at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>

anted system execution privileges on an Android device. The application could then invoke the vulnerable functio  
ghtList function of the file face\_stats\_analyzer.cc. This vulnerability is an out of bounds read issue caused by a mis  
as MEDIUM severity. It poses a considerable risk that should be addressed in a timely manner to prevent potenti  
s the ease with which an attacker could exploit the flaw, potentially without the knowledge of the device user.

specifically in their announcement for Pixel updates on June 1, 2023. The reference URL is: <https://source.android.com> means that the attacker would need the ability to execute code on the user's device.

installed on the user's device, or a rogue process that somehow obtained user-level execution privileges. Without root access to the system. Specific Android versions that include the vulnerable kernel code are impacted by this issue.

This vulnerability affects the Android platform. It involves a possible out of bounds write caused by a heap buffer overflow, which mitigates a moderate risk, and while it's not the most critical rating, it's significant enough to warrant attention and remediation to use the Android kernel. The vulnerability specifically pertains to the Google BMS kernel module within these devices on the affected device. This level of access would typically require previous exploitation of the system or insider access. Privileged processes can trigger the vulnerability without any input or action from the user.

For more information, visit the reference URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>. This bulletin provides details about the vulnerability. For more information, visit the bulletin link given in the CVE description or check with the device manufacturer for patch availability and implementation.

The process has System execution privileges on the device. This process could manipulate the Google BMS kernel module in situations when they are recently disclosed, to avoid enabling potential attackers. The specifics of the vulnerable code are typically not disclosed.

The vulnerability is located in the `android.hardware.health` package, where there is a possible out of bounds read caused by an incorrect bounds check. This vulnerability could potentially be exploited to read sensitive data from memory without proper authorization.

Additional information:

<https://source.android.com/security/bulletin/pixel/2023-06-01>.

The vulnerability is an out of bounds read in the HAL layer (`hal_socket.c`) to read sensitive data from memory without proper authorization. Without appropriate permission checks, this is CVE-2023-21149.

The vulnerability is located in `nonRcsService.java` within the Android operating system. The vulnerability arises because of a missing permission check.

For more information, visit the reference URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

The information provided, but users should refer to the Android Security Bulletin for the most up-to-date version information.

The goal is to compromise the affected system.

The vulnerability could be triggered by a malicious actor triggering the vulnerability to activate or deactivate the RCS (Rich Communication Services) with the device. Responsible disclosure practices aim to avoid aiding potential attackers. The Android Security Bulletin or official patch notes provide more details.

The vulnerability is located in `of protocolimplbuilder.cpp`, which is part of the Android operating system.

Exploitation: No user interaction is needed for the exploitation.

The vulnerability is located in `BuildSetConfig` function, which could read data beyond the bounds of an allocated memory buffer.

The vulnerability was disclosed, specifically in the June 01, 2023 update. The link is: <https://source.android.com/security/bulletin/pixel/2023-06-01> in the provided information.

The vulnerability provides information to an attacker with system-level access.

The vulnerability is located in the `!c.c` file within the Android kernel.

Exploitation:

Generally, vulnerabilities are often mentioned in the context of the Android kernel without specifying a particular version or device.

ting on the device without requiring any special permissions. The application could then exploit the Use-After-Free vulnerability. Users should refer to the Android security bulletin or updates from their device manufacturer for patches. The referer

y.

3 URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

on the device or existing vulnerabilities in other software. The attacker could exploit the use after free vulnerability

where there is a potential heap buffer overflow that could result in an out of bounds write. This issue may lead to remote

It has been detailed in the provided information.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01)

code execution. This means an attacker could exploit the vulnerability to gain control over an affected device and execute

code that triggers the heap buffer overflow in the `cd_CodeMsg` function of the `cd_codec.c` file. Since no user interaction is required

where a tapjacking or overlay attack could potentially occur. This vulnerability can lead to a local escalation of privilege

Android security bulletin webpage, specifically at <https://source.android.com/security/bulletin/android-13>.

display of legitimate UI elements on an Android device. When a user interacts with what they believe to be a genuine UI element, it allows for a local escalation of privilege. This means that a malicious application could gain elevated privileges on the device by

involving a malicious application creating an invisible overlay on top of system settings or permission dialogs. The code is located in

laola.redbull application (<https://play.google.com/store/apps/details?id=laola.redbull>) and on the Packet Storm Security Advisory ID 29459 that, when processed by the vulnerable activity in the laola.redbull application, loads harmful content into the application. CVE-2023-29459 would be an intent crafted to start the `SplashActivity` with a malicious data URI. An `android.content.Intent` object with ID 36612.

versions before 4.2.1.

an attacker to write arbitrary files in the application's private directory and redirect the server's responses to third-party

Linking URL: <https://hackerone.com/reports/1710541>.

is an intention with a custom deeplink scheme that, when processed by the vulnerable Basecamp Android app, could lead to a local escalation of privilege. This issue has been resolved in version 4.2.1 and later.

CVE ID CVE-2023-32274.

located within the binary code of the Android application. This flaw could allow an attacker to exploit these credentials to gain access to sensitive data. The vulnerability is considered to pose a significant risk and should be addressed promptly.

: <https://www.cisa.gov/news-events/ics-advisories/icsa-23-171-02>.

access to the Enphase Installer Toolkit application, potentially allowing them to access sensitive information or provide a specific code snippet that can be provided. Typically, such a vulnerability might involve the presence of a user name or password in the Enphase Installer Toolkit to extract hard-coded credentials. With these credentials, they could attempt to log in. When the screen is in Private Browsing mode, the address bar and keyboard are not hidden. This defect can potentially be exploited. Affected versions are at risk of having sensitive data exposed if they use the screen recording feature during Private Browsing mode. This update will fix the issue where the address bar and keyboard were not properly hidden during Private Browsing mode. This score reflects the potential impact of the vulnerability, which could lead to the exposure of sensitive information. Users on platforms such as Windows, macOS, or Linux are not at risk from this particular issue. The severity is provided by Mozilla.

During their browsing session while in Private Browsing mode on an affected version of Firefox for Android. Since the address bar is visible at <https://www.mozilla.org/security/advisories/mfsa2023-13/> and by examining the detailed bug report at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1744444](https://bugzilla.mozilla.org/show_bug.cgi?id=1744444), the internal behavior of Firefox for Android regarding screen recording during Private Browsing sessions. The vulnerability is where different techniques could be used to obscure the fullscreen notification. This flaw could lead to user confusion.

This indicates a significant risk that requires urgent attention and remediation by users of the affected software.

The vulnerability is specific to Firefox and Focus for Android.

See the Security Advisories page at '<https://www.mozilla.org/security/advisories/mfsa2023-13/>' and the related bug report for details on the techniques to obscure the fullscreen notification in Firefox or Focus for Android. When the user visits this website, there is a potential use-after-free issue. It was addressed by disabling the AAudio backend for instances running on Android versions of Firefox for Android below this specified version are at risk and are advised to update to at least version 115.0. The severity is classified as HIGH severity.

The audio component when running the affected application on Android API versions lower than 30. By doing so, the user can avoid the vulnerability.

See the Security Advisory page (<https://www.mozilla.org/security/advisories/mfsa2023-08/>) and the corresponding Bugzilla entry for details on the vulnerability by crafting a malicious web page or audio content that, when processed by the libaudio component using the affected vulnerability are typically not published in order to prevent malicious use. However, the issue was resolved with a patch. This involves an improper permission and access control issue which allows non-privileged applications to carry out actions that are categorized as 'HIGH' severity. This indicates that it can have a significant negative impact on the confidentiality, integrity, and availability of the system.

The link to the advisory page is <https://support.zte.com.cn/support/news/LoopholeInfoDetail.aspx?newsId=103146>. This vulnerability allows an attacker, without proper authorization, leading to the erasure of personal data and applications. This exploitation could essentially render the device unusable. An attacker could exploit this vulnerability to perform actions that should require elevated permissions. For instance, such an attacker could use the NotificationContentInflater.java in Android. It's a potential temporary denial of service due to long-running operations which could cause the device to hang (stands for Large screens), and Android 13.

See the Android security bulletin at <https://source.android.com/security/bulletin/2023-06-01>.

This notification that triggers the vulnerability in the NotificationContentInflater.java when it performs long-running operations. The vulnerability could be exploited without the device owner's knowledge.

This vulnerability was publicly disclosed to prevent misuse. However, the remediation in the Android open-source code would detail the affected versions of the operating system including Android 11, Android 12, Android 12L, and Android 13. The vulnerability is categorized as MEDIUM severity by the Common Vulnerability Scoring System (CVSS). This indicates that the vulnerability is not a critical issue for Android 13. Devices running these versions of the Android operating system could be at risk if they have not applied the latest patches. Users should check the Android security bulletins regularly to stay informed about the latest vulnerabilities and patches.

<https://source.android.com/security/bulletin/2023-06-01>. It is recommended to refer to the official security bulletin for the app or script that makes use of the improper input validation vulnerability. Since user interaction is not needed, the vulnerability stems from improper input validation. The exploitation does not require user interaction, so an information disclosure without additional execution privileges needed.   
d Android 13.

tin/2023-06-01.  
e, which takes advantage of the permissions bypass to access and disclose sensitive information from the device's  
his vulnerability allows for a possible way to access developer mode traces, leading to local information disclosure. It indicates that the vulnerability presents a moderate level of risk, and while it's not as critical as higher-rated vulnerabilities on Android 13.  
Programmatic access to a device could exploit the permissions bypass to gain unauthorized access to developer mode traces. An attacker to exploit the vulnerability without alerting the device owner.  
The following URL: <https://source.android.com/security/bulletin/2023-06-01>.

the device manufacturers or by Google for their Android devices. It's essential to keep devices updated with the latest version of the Android device utilizing the permissions bypass to access developer mode traces. This could lead to the exposure of sensitive data on the device, specifically affecting Android version 13. It involves the possibility of launching an arbitrary activity without user interaction. This indicates that the vulnerability poses a significant risk and should be addressed promptly by users and device manufacturers. As noted in the references, you can check the Android Security Bulletin for June 2023 at the following URL: <https://source.android.com/security/bulletin/2023-06-01>.  
It is recommended to update their system to the latest version to mitigate the vulnerability.  
An attacker can escalate privileges locally without any interaction from the user.  
An attacker can exploit the vulnerability in the MediaControlPanel.java component to launch an arbitrary privileged activity. For more information, please refer to the specific issue within the Android ecosystem.  
Google has started working on patches and users should be on the lookout for security updates regarding this issue.  
The vulnerability in the CallRedirectionProcessor.java is CVE-2023-21138.  
This is a vulnerability within the Android operating system. This issue is caused by improper input validation, which could result in a local denial of service (DoS) on Android 13.  
It indicates that the vulnerability presents a significant risk that should be addressed promptly.

Following URL: <https://source.android.com/security/bulletin/2023-06-01>  
It is recommended to update the device with user execution privileges.  
An attacker can exploit the vulnerability on the device, allowing them to launch background activities or perform other malicious actions without the knowledge of the user. An attacker installed, exploits the vulnerability in the CallRedirectionProcessor.java's onNullBinding method by sending carefully crafted data. Due to exceptions in job map parsing, this flaw could result in a local persistent denial of service on Android devices. The vulnerability is present in Android-13.  
It indicates that the vulnerability has a moderate impact and is a significant concern that should be addressed, but it is not a critical issue. The following URL: [source.android.com/security/bulletin/2023-06-01](https://source.android.com/security/bulletin/2023-06-01).

It is recommended to update the device with user execution privileges.  
An attacker can exploit the vulnerability on the device, allowing them to launch background activities or perform other malicious actions without the knowledge of the user. An attacker installed, exploits the vulnerability in the CallRedirectionProcessor.java's onNullBinding method by sending carefully crafted data. Due to exceptions in job map parsing, this flaw could result in a local persistent denial of service on Android devices. The vulnerability is present in Android-13.  
It indicates that the vulnerability has a moderate impact and is a significant concern that should be addressed, but it is not a critical issue. The following URL: [source.android.com/security/bulletin/2023-06-01](https://source.android.com/security/bulletin/2023-06-01).

<https://source.android.com/security/bulletin/2023-06-01>.

on startup.

but data that is processed incorrectly by JobStore.java functions. This could lead to the Android system crashing or

.

onCreate function of NotificationAccessSettings.java due to improper input validation. This could enable an attacker to bypass the notification access settings on Android 13.

[source.android.com/security/bulletin/2023-06-01](https://source.android.com/security/bulletin/2023-06-01)

dation in the onCreate method of NotificationAccessSettings.java. Since user interaction is not required, a malicious user can trigger the vulnerability specifically within the checkKeyIntentParceledCorrectly() method. This vulnerability involves a logic error that could allow an attacker to bypass the intended security checks. The analysis indicates that the vulnerability presents significant risks and could result in elevated privileges for an attacker if exploited.

t the vulnerability without any action from the user.

oid 13.

rg URL: <https://source.android.com/security/bulletin/2023-06-01>

icker to launch arbitrary activities within the Android settings interface, potentially compromising the device and service. The app would exploit the logic error in the `checkKeyIntentParceledCorrectly()` function to circumvent security checks in `ble_gap.cc`, which is part of Android. It is a buffer overflow issue that could potentially allow a remote attacker to compromise the confidentiality, integrity, and availability of the affected device.

note and without the need for the user to take any action.

llowing URL: <https://source.android.com/security/bulletin/2023-06-01>.

• a remote attacker sending a specially crafted packet or sequence of packets that could overflow the buffer with 023-21129.

additional execution privileges due to a possible activity launch while the app is in the background, which is a Br

[tps://source.android.com/security/bulletin/2023-06-01.](https://source.android.com/security/bulletin/2023-06-01)

- to perform a specific action that triggers its background process. Due to the BAL bypass, this malicious app could

nd Android 13.

roller.java on Android. This logic error can lead to a local escalation of privilege without needing additional execut

[//source.android.com/security/bulletin/2023-06-01.](https://source.android.com/security/bulletin/2023-06-01)

exploitation. However, details about the nature of the logic error could be found in Android's code repository or the privileges locally. This could allow the attacker to execute privileged actions without user consent or awareness.

There is a potential out of bounds write due to uninitialized data. This issue could lead to remote code execution with

Android-13.

Full page: <https://source.android.com/security/bulletin/2023-06-01>

affected media file, which could trigger the out of bounds write issue, potentially leading to remote code execution on

ally, it is related to a potential privilege escalation due to unsafe intent handling within the MediaControlPanel.java

ing to the CVSS (Common Vulnerability Scoring System).

as the attack can be carried out without the knowledge of the device user.

<https://source.android.com/security/bulletin/2023-06-01>.

an implicit Intent targeting the MediaControlPanel.java component. Since the component does not properly validate the intent, it could allow an attacker to execute code on the device with higher system permissions than they are granted, potentially compromising the security of the device. In order to prevent misuse, the vulnerability stems from improper handling of input intents in the `bindOutputSwitch` method. This vulnerability affects Android 11, Android 12, Android 12L, and Android 13. It involves a possible escalation of privilege due to unsafe intent handling. This is considered a significant risk and should be addressed promptly by affected users and systems.

Security Bulletin: <https://source.android.com/security/bulletin/2023-06-01>.

affected versions Android 11, Android 12, Android 12L, and Android 13.

on the device. Since the vulnerability is due to unsafe deserialization, the attacker might execute code on the device that addresses CVE-2023-21124. Staying informed by regularly checking security advisories such as the Android

Android-13.

actions of multiple files.

Full page: <https://source.android.com/security/bulletin/2023-06-01>.

er processes' execution without proper permission, due to the bug in the permission check mechanism. Since special processes are running, potentially allowing the app to capture sensitive information, modify the behavior of other processes, and potentially bypass the DISALLOW\_DEBUGGING\_FEATURES restriction on the operating system. It pertains to a potential method for bypassing the DISALLOW\_DEBUGGING\_FEATURES restriction

<https://source.android.com/security/bulletin/2023-06-01>

permission check to bypass the DISALLOW\_DEBUGGING\_FEATURES restrictions and perform unauthorized tracing operations.

This vulnerability on Android is CVE-2023-21121.



source.android.com/security/bulletin/2023-06-01

Method of AppManagementFragment.java on Android.

Android VPN, which can potentially lead to information leak and unauthorized access to a user's network traffic.

An application that exploits this vulnerability to prevent disconnection of a compromised VPN configuration. This could

functions of cdm\_engine.cpp.

Android device.

0.

ailed in the provided information.

URL: <https://source.android.com/security/bulletin/2023-06-01>

of the use-after-free vulnerability in cdm\_engine.cpp. Since the exploitation does not require user interaction, the Android Security Bulletin, specifically designed to address this and other issues. Device manufacturers would release patches

where there is a possible way to downgrade the link key type due to improperly used cryptography. This vulnerability could reduce the cryptographic strength of the link key between paired devices. This could result in the attacker escalating t

Android Security Bulletin page at the following URL: <https://source.android.com/security/bulletin/2023-06-01>.

Developers and security professionals can review the details provided in the Android Security Bulletin and patch n

all security updates provided by the Android operating system. Staying updated with the latest security patches is

within the Android platform. The issue is a possible out of bounds write due to a use after free, which could potentially affect Android 12 Long-Term Support), and Android 13.

source.android.com/security/bulletin/2023-06-01.

An attacker could potentially exploit the vulnerability remotely over Bluetooth without any actions from the user, as long as Bluetooth is enabled on the device. The attacker would first need to scan for devices with HFP support enabled. Once identifying such a device, the attacker could then attempt to exploit the vulnerability where there is a potential for a confused deputy problem that could lead to cross-user media read, resulting in local media access (LMA) severity. This means it represents a moderate risk that should be addressed in a timely manner to prevent potential escalation to the described security issue.

The confused deputy issue in ChooserActivity.java to read media files from the storage of another user without proper permissions. This vulnerability is detailed on the Android security bulletin website, as referenced by the URL: <https://source.android.com/security/bulletin/2023-06-01>.

The vulnerability could be exploited by a malicious application without any action from the user, which increases the risk and potential for a successful

local escalation of privilege without requiring additional execution. This vulnerability is detailed in the Android Security Bulletin (Android 13).

Chip).

form a use after free in WVDrmPlugin.cpp. An attacker with local access to the device could execute code in the c  
: 'canStartSystemGesture' function of the 'RecentsAnimationDeviceState.java' file. This vulnerability could lead to

n in the 'canStartSystemGesture' method and exploiting it to implement a partial lockscreen bypass. Since this vul  
system internals, specific code examples are typically not provided to the public to prevent misuse and exploitatio

ality under Windows, despite having the security fix for CVE-2023-2976.

and earlier for both iOS and Android platforms.

attacker could eavesdrop on the encrypted communication between the user's app and the server. This could potentially include links to the Google Play Store, the developer's official website, the Apple App Store, and the Japan Vulnerability Exchange. An attacker could also position themselves in a man-in-the-middle position. The attacker could then present a fraudulent example of improper server certificate verification in an app might look like the following pseudo-code snippet:

```
2 for Android. This issue allows a local attacker to perform a denial of service attack by exploiting the SharedPreferences class. This issue is considered MEDIUM in severity.
```

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29753/CVE%20detailed.md>

to the device can manipulate the SharedPreference files used by the Facemoji Emoji Keyboard application to disrupt its operation. An attacker with permissions on the targeted device modifying the SharedPreference files of the Facemoji Emoji Keyboard application, which allows unauthorized applications to cause a persistent denial of service (DoS) by tampering with the SharedPreference files, presents a moderate threat level, potentially having a noticeable impact on the availability of the affected application.

condition persistently within the Yandex Navigator app. This means the application could crash or become unres

ce taking advantage of the vulnerability to modify the SharedPreferences files of the Yandex Navigator app. This could be achieved through the manipulation of database files.

[LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29767/CVE%20detailed.md](#)

Running CrossX v.1.15.3. The attacker could then manipulate the application's database files, potentially corrupting the general type of code vulnerability that could lead to this kind of issue might involve improper handling of file permissions to perform an escalation of privileges by exploiting the database files. (CVSS). This indicates that the impact of the vulnerability is considered high.

to the database files of the CrossX application on an Android device. The attacker could potentially modify or leverage permissions beyond what is initially granted. This could allow them to perform actions or access resources on the Android device. [LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29766/CVE%20detailed.md](#) authorized applications to cause a persistent denial of service by manipulating SharedPreferences files.

[LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29761/CVE%20detailed.md](#)

ce, gains access to the SharedPreferences files used by the Sleep app. By manipulating these files, the attacker can cause a denial of service. The vulnerable code might involve insecure handling of SharedPreferences files in Android, like so: `javaSharedPreferences.putBoolean("sleep", true);` which allows unauthorized applications to cause a persistent denial of service by manipulating the app's database file.

[LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29759/CVE%20detailed.md](#)

FlightAware's local database files on an Android device and manipulating them in such a way that the FlightAware app could be used to cause a persistent denial of service (DoS) by manipulating the SharedPreferences files on the device triggered by unauthorized applications.

SharedPreferences files used by the Blue Light Filter app. By altering these preferences, the attacker could cause the Blue Light Filter app to malfunction. [LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29758/CVE%20detailed.md](#)

Specifically, if an application relies on SharedPreferences files which can be manipulated by unauthorized apps, it could cause files to disrupt the Blue Light Filter app. Since exact code may vary based on the application's implementation, this is a vulnerability that allows unauthorized applications to perform escalation of privilege attacks by manipulating the files as HIGH severity. This indicates that it is a serious issue that should be addressed promptly.

legues without proper authorization. This could potentially give them access to sensitive information or the ability to perform actions that should be restricted.

[LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29757/CVE%20detailed.md](#)

attacker could gain higher privileges than initially granted, potentially allowing them to access or modify sensitive data. A malicious app that, once installed on a device, uses a flaw in the Blue Light Filter app to manipulate the SharedPreferences file mode that can be accessed only by the creating app. However, with CVE-2023-29757, a vulnerability was found in the Blue Light Filter app. Additionally, users should be cautious about the apps they install and grant permissions to, and developers should demonstrate a security oversight in how SharedPreferences files are handled. Developers should ensure their apps

ions can manipulate SharedPreferences files to cause a persistent denial of service.

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29756/CVE%20detailed.md>

an exploit CVE-2023-29756.

affected application might continuously crash or become unresponsive.

the same Android device as Twilight. This malicious app could use an exploit to gain access to the SharedPreferences via unauthorized applications to perform escalation of privilege attacks by manipulating the SharedPreferences file. This vulnerability is classified as HIGH severity.

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29755/CVE%20detailed.md>

Android device as Twilight v.13.3, which exploits the vulnerability by gaining unauthorized access to the SharedPreferences of the Twilight application and the Android device, it might look something like this in a simplified form:// Example for Android that allows unauthorized apps to escalate privileges by manipulating the component. This could potentially be exploited, indicating that it poses a significant risk to affected systems and should be addressed promptly.

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29752/CVE%20detailed.md>.

user or application gaining higher-level access than originally intended, bypassing the security mechanisms in place. This vulnerability in Facemoji Emoji Keyboard. Upon installation, the malicious app could execute commands or make changes to the system. This vulnerability allows unauthorized applications to carry out escalation of privilege attacks by tampering with the system. This vulnerability is classified as HIGH according to its base score.

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29749/CVE%20detailed.md>

the same device exploiting this vulnerability to elevate its privileges. It could do so by modifying the SharedPreferences files of the application, allowing them to perform unauthorized actions without user consent. This can potentially lead to data theft, manipulation of app behavior, etc. CVE-2023-29749, this type of vulnerability typically involves manipulating the SharedPreferences files of an Android application. This vulnerability has likely been addressed. Additionally, users should be cautious about granting permissions to other applications.

error during the unregister provider process after a register call.

as HIGH in terms of severity.

Bulletin, specifically in their June 2023 bulletin at the following URL: <https://www.qualcomm.com/company/product-security/bulletins/june-2023>. Without specific details of the affected codebase. Generally, a double free issue might look like this in C code: ``c#inc. This vulnerability by triggering the faulty code path, which results in memory corruption. Memory corruption vulnerabilities can be exploited to execute arbitrary code or cause a denial of service.

the issues of memory corruption that, with sufficient effort, could potentially be exploited by attackers to execute arbitrary code or cause a denial of service prior to 112, and Focus for Android versions prior to 112.

: <https://www.mozilla.org/security/advisories/mfsa2023-13/> and associated bug reports listed at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1788811](https://bugzilla.mozilla.org/show_bug.cgi?id=1788811). When visited by a user with a vulnerable version of Firefox or Focus for Android, could exploit the memory safety issue, as they could facilitate malicious activities. Responsible disclosure usually involves not publishing specific details of the vulnerability.

Android before 112, Firefox for Android before 112, and Thunderbird before 102.10.

as HIGH severity.

attacker puts in enough effort, as some of the bugs showed evidence of memory corruption.

<https://www.mozilla.org/security/advisories/mfsa2023-14/> - <https://www.mozilla.org/security/advisories/mfsa2023-13/> - [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1788811](https://bugzilla.mozilla.org/show_bug.cgi?id=1788811)

can exploit the memory safety bugs when processed by a vulnerable version of the browser or email client. This can be exploited in Firefox and results in the incorrect assignment of realm. This problem can compromise JavaScript-internal data. This problem affects Firefox versions earlier than 112, and Focus for Android versions earlier than 112.

[https://www.mozilla.org/security/advisories/mfsa2023-14/2](#). [https://www.mozilla.org/security/advisories/mfsa2023-13/3](#). [https://www.mozilla.org/en-US/security/advisories/mfsa2023-14/2?id=1822754](#)

The compiler led to an incorrect optimization result. This could potentially lead to incorrect program behavior or other exploitable side effects, such as the execution of specially designed content that takes advantage of the incorrect optimization in the ARM64 Ion compiler. When processing cookies from the same domain, this could potentially desynchronize the expected results when reading from the secure cookie store.

This issue was fixed in Android 11L (API level 33) and earlier versions of Android before version 11.2.

<https://www.mozilla.org/security/advisories/mfsa2023-13/>) and the corresponding Bugzilla report ([https://bugzilla.mozilla.org/show\\_bug.cgi?id=1849544](https://bugzilla.mozilla.org/show_bug.cgi?id=1849544)). This issue affects Firefox desktop to version 112 or later, and Focus for Android to version 112 or newer. Instances of resource exhaustion simultaneously, which could trigger improper garbage collector operation leading to a crash (CVE-2023-1129). Attackers typically devise their own exploitation techniques based on the vulnerability details disclosed in a public advisory. This issue affects Firefox desktop, and Focus for Android. This issue could allow an attacker to cause memory corruption and exploit a user of Firefox desktop to version 112 or later, and Focus for Android before version 112. Users of these products are advised to update to the latest version of Firefox desktop to version 112 or later, and Focus for Android to version 112 or newer.

curity advisories (mfsa2023-14, mfsa2023-13, mfsa2023-15) and the associated Bugzilla entry (bug 1810191). The , such as Windows and macOS, are unaffected by this particular vulnerability.

desktop file and tricking the user into downloading it via a compromised website, phishing email, or masquerading as an Android before version 112, Firefox before version 112, and Focus for Android before version 112. The flaw involves bypassing of same-origin restrictions in sandboxed iframes. By exploiting the flaw, an attacker could redirect a user to malicious or unexpected content on versions 112 or newer. Mozilla has addressed the issue in these versions, closing the vulnerability window. Users should update to the latest versions provided by Mozilla. These include the Mozilla Foundation Security Advisory (MFSA) at <https://www.mozilla.org/security/advisories/mfsa2023-14/> which is considered to have a medium level of severity. The score reflects a balanced consideration of aspects like the complexity of the exploit, the impact of a sandboxed iframe. The attacker then exploits the vulnerability by embedding a redirect in `sourceMappingUrls` in a JavaScript file. CVE-2023-29539.

Firefox for Android versions prior to 112, Firefox ESR versions prior to 102.10, Firefox for Android versions prior to 112, and Thunderbird versions prior to 102.10. The vulnerability is indicated if the filename contained a NULL character. This could facilitate reflected file download attacks that mislead users.

For more information, see: <https://www.mozilla.org/security/advisories/mfsa2023-14/>, <https://www.mozilla.org/security/advisories/mfsa2023-13/>, and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1685403](https://bugzilla.mozilla.org/show_bug.cgi?id=1685403). The vulnerability involves a file download with a specially designed filename containing a NULL character. When a user visits the page, the browser sends the Content-Disposition header of a file to be downloaded as follows: Content-Disposition: attachment; filename="

example.txt" instead of the expected `moz-extension://` URI during a load request. This can result in the leakage of local file paths in versions before 112, and Focus for Android versions before 112. The severity issue.

For more information, see the Mozilla Bugzilla entry at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1685403](https://bugzilla.mozilla.org/show_bug.cgi?id=1685403) and the Mozilla Foundation Security Advisory at <https://www.mozilla.org/security/advisories/mfsa2023-14/>. The vulnerability involves a file download that triggers a request to load resources, causing the browser to inadvertently leak local file paths in versions 112 or later, which contains the fixes for this vulnerability. Regularly update to the latest versions of certain Mozilla products, which if exploited, could lead to memory corruption and execution of arbitrary code in Firefox before version 112, and Focus for Android older than version 112.

For more information, see: [https://www.mozilla.org/show\\_bug.cgi?id=1824200](https://www.mozilla.org/show_bug.cgi?id=1824200), <https://www.mozilla.org/security/advisories/mfsa2023-13/>, and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1824200](https://bugzilla.mozilla.org/show_bug.cgi?id=1824200). The vulnerability exploits the race conditions during font initialization in affected Mozilla products. When victim users interact with the affected products, the browser sends the Content-Disposition header of a file to be downloaded as follows: Content-Disposition: attachment; filename="example.txt" instead of the expected `moz-extension://` URI during a load request. This can result in the leakage of local file paths in versions 112 or later, which contains the necessary patches to mitigate the vulnerability. The vulnerability is due to an issue within the memory manager in versions before 112), Firefox ESR (versions before 102.10), Firefox for Android (versions before 112), and Thunderbird (versions before 102.10).

For more information, see the associated Bugzilla entry. Here are some references: <https://www.mozilla.org/security/advisories/mfsa2023-14/>, <https://www.mozilla.org/security/advisories/mfsa2023-13/>, and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1685403](https://bugzilla.mozilla.org/show_bug.cgi?id=1685403). The vulnerability involves a file download that triggers a request to load resources, causing the browser to inadvertently leak local file paths in versions 112 or later, which contains the necessary patches to mitigate the vulnerability. Regularly update to the latest versions. Applying the updates will patch the vulnerabilities and prevent attackers from exploiting them. The severity issue.

For more information, see the associated Bugzilla entry. Here are some references: [https://www.mozilla.org/show\\_bug.cgi?id=1824200](https://www.mozilla.org/show_bug.cgi?id=1824200), <https://www.mozilla.org/security/advisories/mfsa2023-13/>, and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1824200](https://bugzilla.mozilla.org/show_bug.cgi?id=1824200). The vulnerability exploits the race conditions during font initialization in affected Mozilla products. When victim users interact with the affected products, the browser sends the Content-Disposition header of a file to be downloaded as follows: Content-Disposition: attachment; filename="example.txt" instead of the expected `moz-extension://` URI during a load request. This can result in the leakage of local file paths in versions 112 or later, which contains the necessary patches to mitigate the vulnerability. The vulnerability is due to an issue within the memory manager in versions before 112), Firefox ESR (versions before 102.10), Firefox for Android (versions before 112), and Thunderbird (versions before 102.10).

For more information, see the associated Bugzilla entry. Here are some references: [https://www.mozilla.org/show\\_bug.cgi?id=1824200](https://www.mozilla.org/show_bug.cgi?id=1824200), <https://www.mozilla.org/security/advisories/mfsa2023-14/>, <https://www.mozilla.org/security/advisories/mfsa2023-13/>, and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1685403](https://bugzilla.mozilla.org/show_bug.cgi?id=1685403). The vulnerability involves a file download that triggers a request to load resources, causing the browser to inadvertently leak local file paths in versions 112 or later, which contains the necessary patches to mitigate the vulnerability. Regularly update to the latest versions where the issue has been addressed. This includes updating Firefox to version 112 or later, Firefox ESR to 102.10, Firefox for Android to 112, and Thunderbird to 102.10. The vulnerability is indicated if the filename contained a NULL character. This could facilitate reflected file download attacks that mislead users.

Firefox, Focus for Android, Firefox ESR, Firefox for Android, and Thunderbird. The issue arises from the potential for Firefox for Android before 112, Firefox ESR before 102.10, Firefox for Android before 112, and Thunderbird prior to 102.10.0 to be vulnerable to this attack.

[illegible]

d be hidden by exploiting download popups. This could lead to user confusion or allow for spoofing attacks.

mozilla.org/security/advisories/mfsa2023-09/' and 'https://bugzilla.mozilla.org/show\_bug.cgi?id=1783561'.

is vulnerability.

ification being hidden by download popups.

he issue has been resolved.

popups in such a way that they hide the fullscreen notification in Firefox for Android. This could lead to a user being tricked into installing updates. Updates for Firefox for Android with unpatched vulnerabilities could be launched from a browser using Intents. This flaw potentially exposed updates for Firefox for Android. Updates for other Firefox products, such as those for desktop operating systems, are not affected by this vulnerability.

erity.

ser to visit a malicious web page that can trigger the launch of an Android application with unpatched vulnerability at <https://www.mozilla.org/security/advisories/mfsa2023-09/> and the related Bugzilla entry at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1861400](https://bugzilla.mozilla.org/show_bug.cgi?id=1861400) where a prompt with a long description could hide the fullscreen notification, which might lead to user confusion. This vulnerability is specific to Firefox for Android, and other operating systems running Firefox are not affected.

with a long description to hide the fullscreen notification. Users might be misled into thinking they are interacting with a different application. While it is not the most critical issue, it still poses a significant risk that should be addressed to ensure user safety and to help affected users identify and remediate the problem.

<https://www.mozilla.org/security/advisories/mfsa2023-09/> provides details about the vulnerability and the affected versions of Firefox for Android and how prompts and notifications are displayed. An exploitation scenario could involve an attacker exploiting the fact that notification permissions were incorrectly stored without considering the browsing context. As a result, notification permissions on Windows, macOS, or Linux, are not affected by this particular vulnerability.

t <https://www.mozilla.org/security/advisories/mfsa2023-01/> and the related Bugzilla entry at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1862000](https://bugzilla.mozilla.org/show_bug.cgi?id=1862000).  
Android version 109 and newer are not susceptible to this vulnerability.

handling flaw to display unwanted notifications to a user in different browsing sessions. This could lead to various |  
version 109 or higher, which contains the necessary fixes to address this vulnerability.

3-29746.

authorized apps to perform a code execution attack by tampering with the SharedPreference files.

ore reflects the severity of the vulnerability, indicating that it poses a serious risk and may have a high impact on t

the official The Thaiger website (<https://thethaiger.com/>), the APKSOS page for story saver/downloader app (<http://www.apksos.com/>) exploiting the vulnerability to modify The Thaiger app's SharedPreferences files. These modifications could lead to a specifically crafted Android application. When installed on the same device as The Thaiger app, the malicious app could read the app's SharedPreferences files and insert data into the application's database that stores user data. The vulnerability allows unauthorized applications to insert data into the application's database that stores user data.

cious data into the app's database, causing it to load malicious image URLs, which can result in the application triggering the same device as the vulnerable BT21 x BTS Wallpaper app, requests and gains the permission to insert data into the post (<http://bungaakpstudio007.com>), a detailed examination of the CVE on GitHub (<https://github.com/LianKee>

d.  
e that contains information about the user's personal preferences. This data is then loaded into memory when the

udio007.com- <https://apkpure.com/cn/bt21-x-bts-wallpaper-hd-4k/com.bungaakp007.bt21wallpaperoffline13092>  
ces in the BT21 x BTS Wallpaper app's database to perform an escalation of privilege attack. This could lead to un  
uests permission to access the vulnerable database and then alters data to gain elevated privileges. Another scen  
r security patches released by the app's developers to mitigate the risk of being affected by CVE-2023-29724. The  
ctices, such as validating all input data, limiting the permissions granted to apps only to those absolutely necessa  
164 for Android. It is a dictionary traversal flaw that allows unauthorized applications to overwrite arbitrary files i  
ates that the vulnerability can have a severe impact on the affected systems, potentially leading to complete conf

LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29736/CVE%20detail.md.

ne device as the vulnerable Keyboard Themes app. The attacker's app could use the dictionary traversal vulnerabi  
their choice on the affected device. This can compromise the security of the device by providing the attacker with t  
allows unauthorized applications to insert data into the database that keeps user personal preferences, leading to

maintaining large amounts of data to the Glitter Unicorn Wallpaper app. This data could include links to malicious im  
: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29723/CVE%20detail.md>.

ng unauthorized applications to interfere with how the Glitter Unicorn Wallpaper app manages and loads person  
; 7.0 through 8.0. This security flaw allows unauthorized applications to request permission to modify data in the  
rity indicates that the vulnerability can have a significant impact on the confidentiality, integrity, or availability of  
h 8.0.

com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29722/CVE%20detail.md

cess the Glitter Unicorn Wallpaper app's database. Once the permission is granted, the attacker's app could modif  
d in the CVE description, no specific code examples can be given. However, the vulnerability's nature suggests it v  
CVE-2023-29748.

or Android, where an exposed component allows an attacker to modify the SharedPreferences file. The attacker ca

y store page, a user's Instagram profile, a third-party APK hosting website, and a detailed write-up on a GitHub re  
l available in the application. They might craft a payload that, when executed, inserts a substantial amount of data  
ation of a SharedPreferences file in an Android application. This vulnerability can be exploited to perform a denial-  
ds that can modify SharedPreferences files are not publicly exposed or are protected with proper security controls  
CVE-2023-29747.

7.

; to modify the SharedPreferences file in the app.



reference file which, when loaded into memory as the application opens, could lead to various consequences such as [w.instagram.com/nihans\\_macrame/](https://w.instagram.com/nihans_macrame/)- <https://apksos.com/app/story.saver.downloader.photo.video.repost.byrk-> h  
lified Android code example that might represent a similar vulnerability could be as follows: ``java// Vulnerable n

id. This vulnerability could allow unauthorized apps to create a persistent denial of service (DoS) condition by tam

[v.zmtqsh.com/](https://v.zmtqsh.com/)- <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29745/CVE%20detail.md>- Detai  
as the underlying database of the BestWeather application. By doing so, they could corrupt or alter data to render  
thetically how such an exploit might work. An attacker would create an Android app that, once granted certain pe

; through manipulation of the database.

[v.zmtqsh.com/](https://v.zmtqsh.com/) - <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29742/CVE%20detail.md> - <https://v.zmtqsh.com/>  
e intercepting or manipulating the database of the BestWeather app to execute malicious code. This could lead to  
rability like CVE-2023-29742 might involve using methods to access or modify the app's database without proper  
app to the latest version provided by the developers, as these updates usually contain patches for such vulnerabi  
ever, users can check for updates and additional information on the Google Play Store page for the BestWeather  
743.

apps to execute a persistent denial of service attack by manipulating the database. This could prevent legitimate

[v.zmtqsh.com/](https://v.zmtqsh.com/) - <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29743/CVE%20detail.md>- <http://www.zmtqsh.com/>- <https://play.google.com/store/apps/details?id=com.zmtqsh.saver.photo.video.repost.byrk>  
exploit the vulnerability by sending crafted intents or using other methods to interact with the database of the Best  
ossible. This type of attack can disrupt the app's service by making it unusable for legitimate users, typically by ove  
application. The issue allows unauthorized apps to perform an escalation of privileges attack by tampering with the  
is CRITICAL in severity.

n intended, allowing malicious actors to access or alter sensitive data within the BestWeather application. An attac  
H website at <http://www.zmtqsh.com/>, the Google Play Store listing at <https://play.google.com/store/apps/details?id=com.zmtqsh.saver.photo.video.repost.byrk>

23-29740.

by manipulating the database in version 5.3.2 of the Alarm Clock for Heavy Sleepers for Android.

[v.zmtqsh.com/](https://v.zmtqsh.com/) - <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29740/CVE%20detail.md>  
r Store at this link: <https://play.google.com/store/apps/details?id=com.amdroidalarmclock.amdroid>  
on the same device as the Alarm Clock for Heavy Sleepers app. This malicious app might exploit the vulnerability t  
VE-2023-29739.

unauthorized applications can exploit the component to perform escalation of privilege attacks.

[v.zmtqsh.com/](https://v.zmtqsh.com/) - <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29739/CVE%20detail.md>- <http://amdroidapp.com/>- <https://play.google.com/store/apps/details?id=com.amdroidalarmclock.amdroid>

ation of privilege attacks. This means that the attacking app could gain elevated permissions beyond what it was such as compromise of the user's personal data, unauthorized access to system features, disabling of security me ation, as the developer may have released a patch to fix the vulnerability associated with CVE-2023-29739. It is al

apps/details?id=com.wave.keyboard - <https://play.google.com/store/apps/details?id=com.amdroidalarmclock.am> e where the Wave Animated Keyboard Emoji app is installed. The attacker could exploit the vulnerability by manip ibility that can be exploited remotely. An attacker would need physical access to the device, or at least a local pre: ere attackers can exploit the application to manipulate feature-related data, leading to a severe elevation of priviled data in the Call Blocker app. This tampering can grant them elevated privileges, potentially allowing them to pe

Common Vulnerability Scoring System (CVSS).

ication to the latest version that addresses this vulnerability, as provided by the developers. Users can also check i ort on GitHub: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29728/CVE%20detail.md2>. The C on or a piece of code that exploits the vulnerability in the Call Blocker app. When this code is executed, it might al ploit might not be readily available to the public to prevent malicious use. However, developers and security rese ndroid. It allows unauthorized applications to use exposed components to delete data stored in its database, inclu

Play store link for the application (<https://play.google.com/store/apps/details?id=com.cuiet.blockCalls>), a GitHub exploiting the exposed components of the Call Blocker app to delete sensitive user data from its database. This ca pers that address CVE-2023-29727. It's also wise to review the app's permissions and consider using security softw : installed on the same device as the vulnerable Call Blocker app. This malicious app could then invoke exposed Br he application incorrectly opens a key component that can be exploited by an attacker to inject large amounts of as HIGH severity. This indicates that it poses a significant threat to the affected application's availability.

application's database. This could be done by finding a vulnerability within the application that does not properly res 3 blocker application. Users would be unable to use the app, as it would continually crash upon startup due to mem etailed description: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29726/CVE%20detail.md>- Th

n 7.09.01 for Android.

edjing Mix to cause a denial of service. This might involve inserting, modifying, or corrupting data within the data ability presents a moderate level of risk to the affected system.

ib repository containing a detailed markdown file: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-20> ccess to the Android device running the vulnerable version of edjing Mix. The attacker could then tamper with th 01 for Android. This flaw allows unauthorized applications to perform escalation of privilege attacks by manipulat s high score indicates that the flaw poses a significant risk and can have a severe impact if exploited.

atform.

m/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29734/CVE%20detail.md

e device manipulating the database of the edjing Mix application. The attacker could elevate their privileges withi issue is resolved. Developers of the application must issue a patch to address the vulnerability by securing the data

rence files, which can lead to security issues such as functionality manipulation and result in a severe escalation o

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29733/CVE%20detail.md>.

evice as the Lock Master app. The malicious app could exploit the vulnerability by modifying SharedPreference file

5.20 for Android. This flaw allows an unauthenticated attacker to modify the SharedPreference file due to the cor

URL: <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29732/CVE%20detail.md>

the SoLive application for Android. Depending on the usage of the altered data, possible attack scenarios include for Android. The vulnerability exposes a component of the SoLive application that allows modification of the SharedPreference file of the SoLive app. The exposed component in the app provides a method to modify this file, and once the device condition. By crashing the SoLive app at startup due to an OOM error, it prevents the legitimate user from using the provided.

[Kee/SO-CVEs/blob/main/CVEs/CVE-2023-29731/CVE%20detail.md](https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29731/CVE%20detail.md)

on the victim's device can use the exposed method to programatically inject a voluminous amount of corrupt or a

a function named 'insert'.

[/SO-CVEs/blob/main/CVEs/CVE-2022-47028/CVE%20detailed.md](https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2022-47028/CVE%20detailed.md).

'insert' function of Action Launcher for Android. This could lead to the application crashing or becoming unresponsiveness, a general example of such an exploit might involve the attacker crafting a malicious input that, when pro

Emoji application for Android.

the database files in the Wave Animated Keyboard Emoji application.

that the vulnerability poses a moderate level of risk.

[le.com/store/apps/details?id=com.wave.keyboard-](https://www.wavekeyboard.com/) <http://www.wavekeyboard.com/>- <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-33188/CVE%20detail.md>  
manipulate or delete database files, leading to the application crashing or behaving unpredictably, thus causing a d  
exploit may not be readily available, and it would be irresponsible to provide such examples as they could facilitate r  
plication versions before 3.8.50 for Android. This vulnerability allows a child to bypass parental restrictions tempc  
vel according to the CVSS (Common Vulnerability Scoring System) scale.

[/vulnerability-lab/advisory/multiple-vulnerabilities-in-kiddoware-kids-place-parental-control-android-app/](https://github.com/LianKee/vulnerability-lab/advisory/multiple-vulnerabilities-in-kiddoware-kids-place-parental-control-android-app/).

id device using the Kiddoware Kids Place Parental Control application. The child could bypass these restrictions by  
ation to version 3.8.50 or later, as this version includes fixes for the vulnerability. Additionally, parents should mo  
on vulnerability found in the Omni-notes Android app. This vulnerability could allow other applications on the sar  
severity according to the Common Vulnerability Scoring System (CVSS). This implies the vulnerability is of moderat  
by CVE-2023-33188.

), which at the time of the vulnerability disclosure was version 6.2.7. Updating to this version fixes the insufficient

which can be accessed at the following URL: <https://github.com/federicoiosue/Omni-Notes/security/advisories/CVE-2023-28369> as the Omni-notes app could send an intent to Omni-notes with a specific path. Due to the insufficient path validation, a path traversal attack can be performed, leading to the disclosure of sensitive information. This vulnerability arises when cross-compiling c-ares using the autotools build system, particularly when cross compiling for an architecture.

See the official GitHub security advisory at '<https://github.com/c-ares/c-ares/security/advisories/GHSA-54xr-f67r-4pc4>', which describes the issue with the 'rand()' function due to its lack of entropy when used as a fallback for 'CARES\_RANDOM'. This is a critical issue as it is required for certain security-related functions. The fallback to a non-cryptographically secure 'rand()' function due to the lack of entropy can lead to predictable outputs, which can be exploited by an attacker to generate DNS query IDs. Here is a link to the advisory: <https://github.com/c-ares/c-ares/security/advisories/GHSA-54xr-f67r-4pc4>.

CVSS 3.1 Base Score: 7.5 (High). This vulnerability is present in versions 1.012.GP.B for Android. The flaw is associated with the SQLite Database component of the app and involves the use of a non-cryptographically secure random number generator. The severity is High. The same host where the Simple Design Daily Journal app is installed.

For more information, see CVE-2023-28369, and there's even a YouTube video about it at <https://www.youtube.com/watch?v=V0u9C5RVSi8>. Users should update the Simple Design Daily Journal app to the latest version as soon as a fix is available. Developers should ensure that sensitive information is encrypted and not stored in plain text. This vulnerability could be exploited by an attacker to extract sensitive information from an unlocked device or using malware to exploit the cleartext storage vulnerability. The attacker could then extract the information and use it for malicious purposes.

The Simple Design Daily Journal app contains an improper access control issue. This could be exploited by another app installed on the same Android device. The severity is classified as LOW severity.

For more information, see [port.brother.com/g/b/link.aspx?prod=group2&faqid=faq00100794\\_000-](https://port.brother.com/g/b/link.aspx?prod=group2&faqid=faq00100794_000-) <https://faq.brother.co.jp/app/answers/d>

This is an application that is installed on the same Android device as the vulnerable Brother iPrint&Scan app. This malicious app can be used to demonstrate exploitation of CVE-2023-28369. The vulnerability is described as an improper access control issue within the 'After free' issue in the Autofill UI. This vulnerability could allow a remote attacker to potentially exploit heap corruption, which indicates that the impact of this security flaw is considered to be high.

For more information, see the CVE tracker at <https://cve.org/CVERecord?id=CVE-2023-2722>, a post on the Chrome Releases blog at <https://chromereleases.googleblog.com/2023/05/stable-channel-update.html>, and a blog post designed to trigger a use after free condition in the Autofill UI component of Google Chrome on Android. When this vulnerability is exploited, it demonstrates how to exploit CVE-2023-2722. Typically, such exploits target memory management vulnerabilities. Users should update their browser to this version or later to mitigate the risk associated with this vulnerability.

The vulnerability is a buffer overflow caused by a heap buffer overflow.

Android-13.

Source: [source.android.com/security/bulletin/2023-05-01](https://source.android.com/security/bulletin/2023-05-01).

The app has the necessary permissions to operate on the device. This app could leverage the heap buffer overflow vulnerability in the 'unregisterReceiver' method of the 'ActivityManagerService.java' file and allows an isolated process to register a receiver.

ans that the vulnerability poses a significant risk and should be addressed promptly to prevent potential exploits of the vulnerability without the need for any action from the user, which makes the vulnerability more critical since which are intended to notify users about vulnerabilities and encourage them to apply patches or updates to mitigate. L: <https://source.android.com/security/bulletin/2023-05-01>. This bulletin typically includes detailed information on the issue, the severity, and the steps to be taken to address the issue. In this case, the issue is a permissions bypass to register a broadcast receiver without proper authorization. Once registered, this receiver could potentially be used to perform malicious actions. However, the issue results from a flaw in the 'registerReceiverWithFeature' method in 'ActivityManagerService.java' and a logic error in 'verifyReplacingVersionCode' of 'InstallPackageHelper.java'. This error could potentially allow attackers to bypass the security checks and register a receiver without the user's knowledge. The severity of the issue is high, as it could potentially allow attackers to perform malicious actions on the device. It suggests that while the vulnerability presents a significant risk, it may not be as critical as those with a direct path to the user's data.

and potentially be exploited without any action from the user, increasing the risk of an attack being successful. For more information, see the Android Security Bulletin webpage at <https://source.android.com/security/bulletin/2023-05-01>.

ion privileges on a device by other means, perhaps through another vulnerability or insider access. They could exploit the vulnerability to gain access to sensitive data or perform other malicious actions. This implies that the attacker has already bypassed certain security measures and has a high level of control over the device. There is a possible out of bounds read due to a missing bounds check. This issue affects various Android versions and is a known issue in Android-13.

that a malicious application could potentially exploit this vulnerability to access sensitive data without needing any

following URL: <https://source.android.com/security/bulletin/2023-05-01>.

missions. The app could exploit the vulnerability by triggering the out of bounds read in the AnalyzeMfcResp function. The issue is a permissions bypass to register a broadcast receiver without proper authorization. The latest security patches provided by their device manufacturer or network provider. Regularly updating the Android operating system. This issue involves improper input validation that could potentially lead to a denial of service condition. This issue affects various Android versions and is a known issue in Android 13.

PhoneAccountRegistrar.java, which could lead to a local denial of service.

dated by PhoneAccountRegistrar.java. This could lead to a denial of service condition, preventing the use of emergency services.

L: <https://source.android.com/security/bulletin/2023-05-01>

Android in response to the vulnerability. Users should ensure their devices are updated with the latest security patches. This issue is a permissions bypass to register a broadcast receiver without proper authorization. In the Android OS, a direct code example may not be applicable or available outside of the Android development team. The issue is a logic error in the code, there is a possibility to hide the application from the user. Such an issue could potentially be exploited to perform malicious actions on the device. It allows for a local escalation of privilege without needing additional execution permissions.

Android-13.

is not provided in the CVE entry. However, such vulnerabilities typically involve improper handling of system resources. The app could be programmed to crash the device or perform other malicious actions. The app could be programmed to crash the device or perform other malicious actions. The app could be programmed to crash the device or perform other malicious actions.

: <https://source.android.com/security/bulletin/2023-05-01>. This link points to the Android Security Bulletin, which provides detailed information on the issue, the severity, and the steps to be taken to address the issue. Due to a logic error in the code, there is a possibility to hide the application from the user. Such an issue could potentially be exploited to perform malicious actions on the device. The issue is a permissions bypass to register a broadcast receiver without proper authorization. The issue could have a significant impact on affected systems if successfully exploited, although it does not directly exploit the vulnerability without the knowledge or participation of the device owner.

the following URL: <https://source.android.com/security/bulletin/2023-05-01>.

error in AccessibilityService to make their malicious app appear invisible or hidden to the user. Since user interaction is required to trigger the issue, the risk of exploitation is reduced. The app could be programmed to crash the device or perform other malicious actions. The app could be programmed to crash the device or perform other malicious actions.

re a missing permission check could lead to local escalation of privilege without any additional execution privilege. This could allow an attacker to elevate privileges locally across user boundaries on affected Android devices.

oid 13.

It takes advantage of the missing permission check to perform unauthorized actions or access sensitive data, leading to a local escalation of privilege. For more details, see the following URL: <https://source.android.com/security/bulletin/2023-05-01>

This could involve an attacker developing a rogue application that, when installed, exploits the missing permission check

\_\_gpu.c is CVE-2023-21106.

This is a double free error, where a double free error could potentially occur. This error can lead to memory corruption and consequent system instability.

The specific details of the vulnerability are not detailed in the given information. Typically, the Android Security Bulletin would provide specific details for CVE-2023-05-01, which is the Android Security Bulletin for May 2023.

An attacker could elevate their privileges to gain more control over the system.

This vulnerability could be exploited without any action from the user.

The exploit, perhaps through a malicious app, exploits the double free vulnerability in the `adreno_set_param` function.

The details provided in the information given. Generally, the details of how to exploit such vulnerabilities are not disclosed

within Android, where there is a missing permission check that could lead to local information disclosure without user interaction. A malicious application could potentially access sensitive information without needing user interaction or additional permissions.

**MEDIUM** severity. This score signifies that the vulnerability presents a moderate level of risk, though it is not as critical as a high severity issue. The vulnerability involves an unsafe transaction function in an improper manner to access information without the necessary permissions. Since the vulnerability details can be found in the Android Security Bulletin at the provided reference URL, <https://source.android.com/security/bulletin/2023-05-01>.

Staying up to date with the latest security bulletins and patches is crucial in protecting devices from known vulnerabilities. For more information, see the Android Security Bulletin (Android Security team), mitigating CVE-2023-21104 would generally involve adding the appropriate permission check within the application.

This is **MEDIUM** severity.

The issue is related to parsing persisted user data in the `registerPhoneAccount` function.

For more details, see the following reference: <https://source.android.com/security/bulletin/2023-05-01>.

The data to be parsed by the affected function (`registerPhoneAccount`) in such a way that it triggers an uncaught exception under normal application execution conditions.

The issue is located in `__efi_rt_asm_wrapper` of `efi-rt-wrapper.S`. This issue could lead to local escalation of privilege with a specially crafted exploit.

This is a kernel vulnerability. It is recommended to consult the Android security bulletin for specific version details.

For more details, see the provided link: <https://source.android.com/security/bulletin/2023-05-01>.

A specially crafted application designed to trigger the logic error in the `__efi_rt_asm_wrapper` of `efi-rt-wrapper.S`. If the exploit is successful, it could lead to local escalation of privilege.

Is of the vulnerable code would be available in the patch or the commit that fixes the issue in the Android kernel :  
ackage.java file, where there is a potential for causing a device to enter a boot loop as a result of resource exhausti  
oid 13.

wing URL: <https://source.android.com/security/bulletin/2023-05-01>

ortcutPackage.java to consume excessive system resources, potentially leading to a device entering a boot loop.  
g how to take advantage of CVE-2023-20930 are not typically shared publicly to prevent misuse. However, develo  
ceAdmin of AdminRestrictedPermissionsUtils.java, an internal component of Android. It is a permissions bypass i

ire vulnerability.

following URL: <https://source.android.com/security/bulletin/2023-05-01>  
vileges executing a malicious application on the affected device. Since no user interaction is needed, the applicati

..

.android.com/security/bulletin/2023-05-01.

application that exploits a vulnerability in the Android System on Chip (SoC) to gain unauthorized execution of co  
n, but the specific versions are not mentioned in the provided information. Usually, the Android Security Bulletin  
23-25772.

there is improper input validation. An authenticated user could potentially cause a denial of service by exploiting t

ndroid application starting from version 3.0.301126-RELEASE.

'[www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00847.html](http://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00847.html)

l has local access to the application, crafts malicious input data that the application fails to validate properly. This

ication before Release 17. This vulnerability could potentially allow an authenticated user to cause a denial of ser  
icates that the vulnerability should be taken seriously, but it does not have the same critical impact level as highe

is available at: <https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00825.html>

onsumption in the vulnerable Intel Unite android application. This may be achieved by sending crafted requests o  
ind be authenticated with the Intel Unite android application. Without meeting these prerequisites, exploiting the  
.7. Users of the application should update to this version or later to mitigate the issue.

res improper access control mechanisms in versions of the app before Release 17, which could potentially allow a  
core of 4.4 on the Common Vulnerability Scoring System (CVSS).

..

ically at the following URL: <https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00825>.  
a user with elevated privileges on a device that has the vulnerable Intel® Unite® Android app installed. They migh  
: might aid in exploiting the issue. However, typically, improper access control vulnerabilities occur when an applic

Campus Android application versions prior to 9.9, which could allow an authenticated user to potentially enable a IUM severity.

<https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00815.html>

ticated.

uncontrolled resource consumption vulnerability to initiate a denial of service attack. By flooding the application with application logic and usage patterns. However, generally, an attacker might write a script or use an existing tool to exploit android application versions before 3.0.301126-RELEASE. This weakness may permit an authenticated user to pot

tion credentials for the Intel(R) Retail Edge android application. They might exploit the improper access control issue. The application should update to version 3.0.301126-RELEASE or later. This update contains the necessary patches to address this issue which is available at <https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00847.html>. Application versions prior to 1.82. It is characterized by uncontrolled resource consumption that could potentially allow an authenticated user to exploit uncontrolled resource consumption to enable a denial of service attack on the device running the application version 1.82 or later, as this version contains fixes for the vulnerability.

<https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00779.html>.

to the local device, deliberately consuming system resources by repeatedly performing certain actions within the application. This vulnerability was typically made public in order to prevent misuse. Details of the exploit may involve interaction with the application. Application versions prior to 1.82. It involves improper access control that could potentially allow an authenticated

41769.

<https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00779.html>.

ion of privilege.

lication to version 1.82 or later, which contains a fix for the security flaw.

to the Intel(R) Connect M Android application on a compromised device. They could exploit the improper access control issue. CVE-2018-8932.

Android and iOS Mobile Application plugin for WordPress. This vulnerability allows attackers to inject and store malicious code into the application. CVE-2018-1000000.

the vulnerability.

database/vulnerability/wpappninja/wordpress-wpmobile-app-plugin-11-20-cross-site-scripting-xss-vulnerability?utm\_source=wpcom&utm\_medium=security&utm\_campaign=vulnerability. This vulnerability allows attackers to inject malicious JavaScript to the WPMobile.App plugin's configuration settings. When another admin or user accesses those settings, the attacker could use a script similar to the following example and input it into a vulnerable field within the plugin settings or

37.8, 2.38.2, and 2.39.0.

API endpoints may include all events regardless of the sharing settings applied to the category option combination. For more information, see <https://github.com/dhis2/dhis2-core/security/advisories/GHSA-7pwm-6rh2-2388>.

cker Capture or Capture applications. However, if the Android Capture App is used, the events that should be restricted to the user should not be able to see based on the sharing settings of the category options. This could lead to unauthorized access to sensitive data.

The recommended action is to update DHIS2 Core to one of the fixed versions.



ng API calls to the `/trackedEntityInstances` and `/events` endpoints and gaining access to data that should be res

Cloud Home iOS and Android Mobile Apps, SanDisk ibi iOS and Android Mobile Apps, My Cloud OS 5 Web App, My  
leading to a remote attacker being able to bypass CORS policy and authentication requirements to obtain device i

server and issue a cross-site request. The permissive CORS policy and lack of authentication for private IPs could th  
page: <https://www.westerndigital.com/support/product-security/wdc-23004-western-digital-my-cloud-os-5-my-c>  
plication called ImagePreviewActivity in Call Settings. This vulnerability is related to the improper export of applic  
ore of 4.6. Although it is not considered critical, it is still a significant security concern as it could lead to unauthori

owing URL: <https://security.samsungmobile.com/securityUpdate.smsb?year=2023&month=05>

it has the vulnerable application installed. The attacker could then manipulate the improperly exported ImagePre  
public code examples that demonstrate the actual vulnerable code are not typically available due to security and  
n the VideoPreviewActivity within Call Settings. This issue affects certain Android devices and was addressed in th  
A severity. While it is not the most critical vulnerability, it nonetheless poses a tangible security risk that should b

on of certain Android devices.

urity website, specifically at: <https://security.samsungmobile.com/securityUpdate.smsb?year=2023&month=05>  
access to the vulnerable device.

evice exploiting the improper export vulnerability in the VideoPreviewActivity component. By doing this, they co  
their devices have been updated with this release to mitigate the risk associated with this vulnerability.

it is an inappropriate implementation in prompts that could allow a remote attacker to bypass permissions restri  
urity classification. However, the CVSS (Common Vulnerability Scoring System) base score for this vulnerability is r

creates a specifically crafted HTML page, and prompts the user to interact with it. Upon interaction, the vulnerabi  
le Chrome Releases blog, the Debian security advisory, and Fedora Project package announcements. Relevant link  
affected versions should update to this version or later to mitigate the risk associated with this vulnerability.

reen Mode of Google Chrome on Android devices. Specifically, before version 113.0.5672.63, a remote attacker c

dered Medium severity.

ease blog post, a Chromium issue tracker link, Debian security advisory, and multiple Fedora project announceme  
eices to version 113.0.5672.63 or later, which contains the necessary patches to address the vulnerability.

in discuss a hypothetical attack scenario. An attacker could design a malicious web page which, when viewed in fu  
vulnerability. It refers to a certificate validation vulnerability in the Baiying Android application, which could poter

ita could be exposed to unauthorized entities.

om.cn/detail/dc\_206093.html

nerability to intercept and decrypt secured traffic between the Baiying Android application and the server, therek  
operating system. It involves a possible heap buffer overflow that could lead to a local escalation of privilege with  
HIGH in severity.

[//source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

es.

r overflow vulnerability by running a malicious application on the device. The application could then potentially w

: error, there's a possible way for an attacker to start foreground services from the background. This flaw could lea  
Android 13.

re.

[source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

ed access to resources that are normally protected from an application or user, potentially leading to unauthorize  
ibility more severe since it can be exploited silently.

logic error in PackageManagerSession.java to start a foreground service from the background without the user's kn  
d is CVE-2023-21098.

for loading arbitrary code into the System Settings app due to a confused deputy problem. This issue could enable  
droid-13.

[om/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01)

oblem in AccountManagerService.java to load arbitrary malicious code into the System Settings app. As no user ir  
, where there is a flaw that could potentially allow an attacker to launch an arbitrary activity through a confused c

13.

thod of Intent.java to launch an arbitrary activity without additional privileges, leading to a local escalation of priv

following URL: <https://source.android.com/security/bulletin/2023-04-01>

tribution\_processor.cc, which could lead to remote code execution.

L in severity.

the following URL: <https://source.android.com/security/bulletin/2023-04-01>.

: without needing additional execution privileges.

ld involve an attacker sending a crafted message or file to the victim's device. When processed by the vulnerable  
d in their regular security updates. Device manufacturers and carriers would then need to push these updates to e  
in the Android operating system. It allows for a possible screen takeover and swapping of display content due to a  
nd could swap the displayed content. This could lead to local escalation of privilege, enabling the attacker to perf  
he vulnerability silently without the user's awareness.

that the vulnerability poses a significant risk and its impact is considered high in terms of confidentiality, integrity,  
in at the following URL: <https://source.android.com/security/bulletin/2023-04-01>. It is advisable to frequently ch  
L, and Android 13.

uted on an Android device without requiring any special permissions. The app could utilize the vulnerability in the `FileUtils` class provided by the device manufacturer or software maintainer and apply the available security patches promptly. If no patch is available within the Android operating system. It is related to a path traversal error that could potentially allow an attacker to access files on Android 13.

[source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

rsal vulnerability in `FileUtils.java`. The application could craft file operations that manipulate the file system to access files that are not accessible. This vulnerability usually occurs when user input is not properly sanitized, allowing for directory navigation characters (such as `../`). There is a flaw ('`retrieveServiceLocked`' method in '`ActiveServices.java`') that allows a way to dynamically register a `BrokerService` on Android 13.

ing URL: <https://source.android.com/security/bulletin/2023-04-01>.

t the vulnerability in the '`retrieveServiceLocked`' function of '`ActiveServices.java`' to dynamically register itself as a `BrokerService` to acquire elevated system rights that it should not normally have. Specifically, it exploits permissions related to the `android.permission.ACCESS_SUPERUSER` as soon as they become available. Android Security Bulletins typically include patches for such vulnerabilities. The CVE check is CVE-2023-21091.

y.java on Android. It stems from a missing permission check that could allow a malicious entity to change system

s signifies that the vulnerability presents a moderate level of risk.

[source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01)

ited silently.

n. The vulnerability can be exploited to change system app locales, which might disrupt the normal operation of the system. CVE-2023-21091 involves the lack of a proper permission check in the `canDisplayLocalUi` function in `AppLocalePickerActivity` that exploits this vulnerability to change the system locale of an Android device without the owner's consent. The CVE is labeled 'CVE-2023-21090'.

Android. It can lead to a local denial of service by causing a boot loop due to resource exhaustion. The problem d

[source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

eUsesPermission function.

act nature of the required interaction isn't specified in the CVE description.

tall a malicious application that uses the `parseUsesPermission` function in `ParsingPackageUtils.java` improperly, leading to a denial of service. This vulnerability allows a possible way to keep a foreground service alive even if the service is not used. This vulnerability can have significant consequences on the confidentiality, integrity, or availability of the affected systems. The vulnerability can be exploited without any action from the user, increasing the severity and ease by which it can be taken advantage of.

'[source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01)

:startInstrumentation component of the `ActivityManagerService` to keep its foreground service running even when the device is in a low-power state. The Android security team for affected versions. These updates will include fixes to prevent the exploitation of the vulne

estrictions due to a logic error in the code, which could lead to local escalation of privilege.

source.android.com/security/bulletin/2023-04-01.

malicious code to perform unauthorized actions in the background without the user's knowledge. For instance, an attacker could use this vulnerability to record audio without showing a privacy indicator, thus compromising user privacy. The description indicates it is a

can cause the device to enter a boot loop, resulting in a local persistent denial of service. This means the affected device is unusable until the device is restarted. This vulnerability is assigned a CVSS score of 7.5 (High).

source.android.com/security/bulletin/2023-04-01

advantage of an existing flaw in the system.

caught exception in PreferencesHelper.java without the need for any user interaction or additional privileges. On Android 11, Android 12, Android 12L, and Android 13. The issue resides in isToggleable functions of SecureNfcEnabler.java. The description indicates that the vulnerability is considered to be significant and should be addressed promptly to prevent potential denial of service on the affected Android device. By exploiting this vulnerability, the attacker could bypass permission checks and

source.android.com/security/bulletin/2023-04-01. This reference will provide details about the vulnerability, affected devices, and mitigation steps.

without any additional execution privileges, making it more critical to address.

Android ID of nci\_hmsgs.cc is CVE-2023-21085.

fixed.

Android-13.

Security Bulletin page: <https://source.android.com/security/bulletin/2023-04-01>

proximity to the target device, implying that the attacker might be in the vicinity of the device, such as in the same local area network or within the same range of the victim's device, such as someone in the same room or nearby. The attacker could send malformed network packets to the device, causing it to crash or become unresponsive. This vulnerability is assigned a CVSS score of 7.5 (High).

An attacker could leverage this vulnerability to execute code with elevated system privileges, potentially leading to a denial of service or other malicious actions. The following URL: <https://source.android.com/security/bulletin/2023-04-01>.

privilege. An attacker could leverage this vulnerability to execute code with elevated system privileges, potentially leading to a denial of service or other malicious actions. This vulnerability can be exploited programmatically or through automated means without requiring the user to perform any specific actions.

Android-13.

attacker to record audio without showing a privacy indicator, thus compromising user privacy.

source.android.com/security/bulletin/2023-04-01.

This is a bypass of permissions that could allow recording of audio without showing the appropriate privacy indicator.

iting the permissions bypass to record audio without triggering any privacy indicators. This might be done covertly via `android.media.MediaRecorder` within Android. There exists a confused deputy problem that could allow for exploitation without the need for additional execution privileges.

ing URL: <https://source.android.com/security/bulletin/2023-04-01>.

A confused deputy issue in the phone call handling component of Android to enumerate contact phone numbers of other users. These numbers are not typically made public due to security concerns. Android platform developers and security researchers were notified of this issue.

Exploitation is possible due to a logic error, potentially leading to local escalation of privilege with no additional execution privileges required. This issue was identified in Android 13.

[source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

The logic error within `PackageInstallerService.java` to perform activities in the background without adhering to the standard Android security model. The details are provided to developers and security professionals.

Without the need for additional execution privileges.

Android-13.

[/source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01)

the affected function in `btif_rc.cc` without proper validation, leading to information disclosure from memory region `btif_rc.cc` file, found within the Android operating system. This vulnerability is due to an incorrect bounds check which

the function `avdt_scb_hdl_pkt_no_frag`, resulting in an out of bounds write because of an incorrect bounds check. See [source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01). This is likely a link to an Android security bulletin that details the vulnerability in Android 13.

Exploitation that takes advantage of this vulnerability to perform an out of bounds write on a device running one of the affected versions of Android. The details are provided to developers and security professionals. Exploitation of this vulnerability requires bypassing Android OS restrictions through a `PendingIntent`, leading to a local escalation of privilege without requiring additional execution privileges.

[source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

Further interaction is required for exploitation.

Exploitation that exploits the `PendingIntent` flaw to bypass Android OS restrictions on launching activities from the background.

on within the 'f\_accessory.c' file of the Android kernel. The issue is an out of bounds write caused by a missing bounds check, leading to a local escalation of privilege.

On a vulnerable Android device, potentially compromising the security and privacy of the device's data and user. Affected versions should be checked in the Android security bulletins or the advisories published by device manufacturer.

The interaction needed has not been specified in the provided data.

Source: [source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

Application or a physical device interaction that triggers the out of bounds write in the 'acc\_ctrlrequest\_composite' file. This is due to a missing bounds check during the deserialization process of multiple files. This flaw could potentially lead to a local escalation of privilege on Android 13.

Impact:

Local escalation of privilege during the deserialization process of multiple files in Android.

Exploitation of this vulnerability without any interaction from the user.

Following link: <https://source.android.com/security/bulletin/2023-04-01>.

An application that is able to access the affected deserialization method in Android without proper bounds checking. This application could be exploited by an attacker with normal user privileges.

The CVE leading to a privilege escalation is CVE-2023-20909.

Android-13.

Source page: <https://source.android.com/security/bulletin/2023-04-01>.

Local escalation due to a missing privilege check. This could lead to local information disclosure without the need for active information from the system, leading to an information disclosure. Since it's a local vulnerability, the attacker can exploit it on a vulnerable device. Check the Android security bulletin or the Android Open Source Project (AOSP) repositories for updates. As of now, specific details are not provided.

Integer overflow, resulting in out-of-bounds heap access. This vulnerability could potentially allow for a local escalation of privilege.

Source: <https://source.android.com/security/bulletin/2023-04-01>.

Interaction that triggers the integer overflow when calling PVRSRVBridgeSyncPrimOpTake. As no user interaction is required, this is a local vulnerability.

Kernel driver, where a missing size check could lead to an integer overflow. This overflow may enable out-of-bounds heap access. The vulnerability presents a significant risk and impact on the affected systems.

Details of the interaction are not detailed in the provided information.

Source: [source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

Ability to achieve out-of-bounds heap access. They could utilize this to gain elevated privileges on a victim's device.

river. The issue stems from a missing size check that can result in a possible integer overflow, potentially leading to

ibility has been acknowledged in specific versions of Android.

dvantage of it without any user involvement.

<https://source.android.com/security/bulletin/2023-04-01>.

an integer overflow to achieve out-of-bounds heap access. By doing so, the application could potentially modify critical data, typically such a vulnerability would be exploited by crafting a specific input that triggers the integer overflow. This vulnerability is addressed in a security update. It is recommended to refer to the provided source Android security bulletin link and to update all running Android devices. It is associated with a missing size check in the PVRSRVBridgeRGXKickSync function which could be exploited to pose a significant risk if exploited.

nc function of the PowerVR kernel driver. This omission can lead to an integer overflow, which in turn may allow an attacker to

wing URL: <https://source.android.com/security/bulletin/2023-04-01>.

s a local attacker could potentially exploit the vulnerability without high-level access privileges.

tion that takes advantage of the missing size check in the PVRSRVBridgeRGXKickSync function to perform an integer overflow in the PowerVR kernel driver, where a missing size check could result in an integer overflow, leading to out-of-bounds heap access. This vulnerability exists in the PowerVR kernel driver in a way that triggers the integer overflow. Since the vulnerability allows out-of-bounds heap access without any input from the user, which makes the vulnerability particularly dangerous as it could be exploited by a local attacker on a PowerVR kernel driver component, which affects certain versions of Android.

[ce.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

er. It is an integer overflow issue that could allow out-of-bounds heap access. This flaw can potentially lead to a local attacker gaining access to sensitive data that it can have significant adverse effects on the confidentiality, integrity, or availability of the system's resources. This vulnerability is targeted towards Android SoCs (System on Chips).

ibility without the need for any user action.

ounds heap access due to an integer overflow, an attacker with local access to the affected device could exploit this vulnerability to gain access to sensitive data.

g reference URL: <https://source.android.com/security/bulletin/2023-04-01>

ally not shared publicly to prevent misuse and are instead handled discreetly by security professionals. The focus is on providing a path to a version that includes a patch for this specific vulnerability. Users and system administrators should apply this patch as soon as it is available to prevent out-of-bounds heap access. This could lead to unauthorized reading or writing to memory locations, allowing the attacker to gain access to sensitive data within the PVRSRVBridgeRGXTDMSubmitTransfer function. This vulnerability arises due to a missing size check that could result in an integer overflow, which is rated as HIGH in severity. This indicates that the vulnerability poses a significant risk of harm if exploited.

on a Chip (SoC) that include this driver is particularly impacted by this security flaw.

nal execution privileges on the affected Android devices. This means that an attacker with local access to the device could exploit this vulnerability without any input or action from the device user, making it particularly dangerous.

following URL: <https://source.android.com/security/bulletin/2023-04-01>

oints the vulnerability to gain higher privileges and access sensitive system resources. An attacker could also use this vulnerability to gain access to sensitive data within the PowerVR kernel driver, where a missing size check could potentially lead to an integer overflow, allowing out-of-bounds heap access. This vulnerability is rated as HIGH severity. This indicates that the vulnerability poses a significant risk and should be addressed promptly.

that utilize the PowerVR kernel driver.

via the following link: <https://source.android.com/security/bulletin/2023-04-01>.

Consequences of exploiting this vulnerability could include local escalation of privilege, allowing an attacker to gain higher privileges. This makes it particularly dangerous as it can potentially be exploited without the user's awareness.

An Android device that utilizes the vulnerability in the PowerVR kernel driver. The application could perform an integer overflow in the PVRSRVBridgePhyMemNewRamBackedLockedPMR function, where a missing size check can lead to an integer overflow of high severity. This indicates that if exploited, the vulnerability could have a significant impact on the affected system without user involvement, which increases the potential risk of the vulnerability since an attack could be carried out silently on a system-on-chip (SoC) that uses the PowerVR kernel driver.

For more information, see the following URL: <https://source.android.com/security/bulletin/2023-04-01>.

An application running on an affected Android device. This application could use the vulnerability in the PowerVR kernel driver to cause an integer overflow.

allowing out-of-bounds heap access in the PVRSRVBridgeChangeSparseMem of the PowerVR kernel driver. This could lead to a local escalation of privileges, meaning an attacker could gain higher-level privileges on the affected device.

For more information, see [source.android.com/security/bulletin/2023-04-01](https://source.android.com/security/bulletin/2023-04-01).

An application exploiting the vulnerability in the PowerVR kernel driver. By taking advantage of the missing size check and causing an integer overflow, a specific code example demonstrating the vulnerability is not generally available in the public domain. However, a code snippet for the PVRSRVBridgeChangeSparseMem function of the PowerVR kernel driver. It involves a missing size check that could cause an integer overflow, leading to a local escalation of privilege. This indicates that the vulnerability presents a significant risk and should be addressed promptly. An attacker could potentially exploit the vulnerability without any action from the user.

The vulnerability affects the PowerVR kernel driver. The specific versions affected were not mentioned in the provided information, but the product is affected.

An attacker with access to the system could leverage the vulnerability to gain higher-level privileges, which could then be used to perform actions that require higher privileges. For more information, see the following URL: <https://source.android.com/security/bulletin/2023-04-01>.

An application that does not require any special permissions. When executed on an affected Android device, this app could exploit the vulnerability in the PVRSRVBridgeRGXKickRS function. This vulnerability arises from a missing size check that could result in an integer overflow. This indicates that the vulnerability presents a significant risk and should be addressed promptly to mitigate potential security issues. The vulnerability affects the PowerVR kernel driver. Any device running an affected Android version with the vulnerable driver is at risk.

An application exploiting the vulnerability without relying on the victim to perform any specific actions.

An attacker with access to the targeted device. This would potentially allow them to execute arbitrary code with higher privileges, potentially leading to a local escalation of privilege.

For more information, see the following URL: <https://source.android.com/security/bulletin/2023-04-01>.

An example of a similar integer overflow leading to out-of-bounds heap access could be something like this: ``cvoid vulnerable\_function() { // ... }`. An attacker could trigger the integer overflow and gain elevated privileges on the device. The attacker could then manipulate the system to perform actions that require higher privileges. An integer overflow due to a missing size check could allow out-of-bounds heap access, potentially leading to a local escalation of privilege. This indicates that the vulnerability is considered to have significant impact and poses a serious risk that needs to be addressed promptly.

The vulnerability affects the PowerVR kernel driver. The specific versions of Android that are affected were not disclosed in the provided information.

For more information, see <https://source.android.com/security/bulletin/2023-04-01>.

The vulnerability is discussed in the context of Android security bulletins and patches.

An attacker with access to a vulnerable system could exploit this weakness to gain higher privileges and potentially cause a local escalation of privilege. The vulnerability was not disclosed, and information regarding it was made available to the public.



a local attacker to cause a denial of service or gain sensitive information by exploiting the insert method of the SharedStorage class (denial of service) or potentially access sensitive information that should otherwise be restricted. This could occur through

the repository detailing the CVE, located at <https://github.com/LianKee/SODA/blob/main/CVEs/CVE-2023-27647/> and in the CVE's details published on GitHub. Specifically, the repository at <https://github.com/LianKee/SODA/blob/main/CVEs/CVE-2023-27647/> the DUALSPACE Lock Master app is installed. The attacker could exploit the vulnerability by interacting with the SharedStorage class. It is identified as a Race Condition vulnerability, which could potentially allow attackers to bypass the payment

3. They argue that the attack is not feasible because the home launcher will be loaded before any user application

[trikinglycdn.com/files/f1d54bf4-3803-480c-b4d3-0943f7dac76e/A920\\_EN\\_20200605.pdf?id=237392-](https://trikinglycdn.com/files/f1d54bf4-3803-480c-b4d3-0943f7dac76e/A920_EN_20200605.pdf?id=237392-) [https://drive.google.com/file/d/1d54bf4-3803-480c-b4d3-0943f7dac76e/A920\\_EN\\_20200605.pdf?id=237392-](https://drive.google.com/file/d/1d54bf4-3803-480c-b4d3-0943f7dac76e/A920_EN_20200605.pdf?id=237392-) the boot process of the PAX A920 Pro PayDroid. By manipulating the timing of operations, the attacker could intercept the boot process. CVE-2023-25954.

Print,' and 'Olivetti Mobile Print,' specifically versions v3.2.0.230119 and earlier.

installed on an Android device can send an intent to the affected apps, allowing it to download malicious files or applications. This vulnerability is detailed in a document from Kyocera Document Solutions at [www.kyoceradocumentsolutions.com/en/our-business/security/information/2023-04-11.html](https://www.kyoceradocumentsolutions.com/en/our-business/security/information/2023-04-11.html) and a Japanese document at <https://jvn.jp/en/entry/20230411000001/>. The vulnerability allows an attacker to send intents to mobile print apps, which could lead to malware infection, data breaches, or further exploitation. The affected app that, once installed, can send specially crafted intents to the affected mobile print apps. These intents are sent to the print apps, which, generally, exploiting intent handling issues in Android typically involves crafting an Intent object in a malicious application. The vulnerability is identified as CVE-2023-25954. It involves an information leak that can occur via the debug interface of the app.

OW severity according to the severity rating system.

holder with relevant details: <https://drive.google.com/drive/folders/1Szu9pjivVtG93ceECvnoAjeSABVyfDES?usp=sharing>. The vulnerability could be an attacker accessing the debug interface without proper authorization. By doing so, they might be able to access the debug interface of the PikPak application. They could exploit this access to extract sensitive information that might include API keys or other confidential data. A non-privileged user can read a small portion of the allocator process memory. It affects specific versions of Arm's Mali GPUs before r42p0, and Avalon (r41p0 before r42p0) graphic architectures in the Arm Android Gralloc Module.

Following URL: <https://developer.arm.com/Arm%20Security%20Center/Mali%20GPU%20Driver%20Vulnerabilities>. The vulnerability is a sample showcasing this issue. It is more about exploiting improper access control or lack of validation while a non-privileged user can read a portion of the allocator process memory, which may lead to information disclosure. The exposed vulnerability is identified as CVE-2023-22429. It affects specific versions of Arm's Mali GPUs before r42p0, and Avalon (r41p0 before r42p0) graphic architectures in the Arm Android Gralloc Module.

3-22429.

1 of an API key for an external service within the 'Wolt Delivery: Food and more' Android app.

This indicates that the vulnerability is considered to have significant impact and represents a serious security risk.

CVSS score: 7.5 (CVSS:3.1/AV:L/AC:L/AT:P/AU:N/C:HI/I:HI/O:HI) and [https://play.google.com/store/apps/details?id=com.wolt.android&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.wolt.android&hl=en_US&gl=US).

The vulnerability is located in the application binary of 'Wolt Delivery: Food and more' to extract the hard-coded API key. With this key, the attacker can bypass navigation restrictions within their application code. Instead, they should implement secure methods of storing and accessing sensitive data. The vulnerability is identified as CVE-2023-22429. It is caused by insufficient policy enforcement in Intents that could allow a remote attacker to bypass navigation restrictions

announcement on the Google Chrome Releases blog, Chromium bug tracker, Fedora Project mailing lists, a Debian security advisory, and a CVE. The exploit was specifically crafted to exploit the vulnerability in Chrome's Intents policy enforcement. When a user on an affected version of Chrome is on a website that has a hypothetical example could involve an attacker crafting a malicious HTML page with embedded JavaScript that triggers the vulnerability. CVE-2023-28999.

Regarding the fixed version 3.8.0 itself.

Nextcloud Desktop client was fixed in version 3.8.0, Nextcloud Android app in version 3.25.0, and Nextcloud Web in version 3.8.0. The vulnerability was fixed in all affected versions of the affected Nextcloud applications.

Following references:- <https://ethz.ch/content/dam/ethz/special-interest/infk/inst-infsec/appliedcrypto/education/nextcloud-exploit.pdf>. The exploit could exploit the vulnerability to gain unauthorized access to end-to-end encrypted folders stored by users or servers, I can't provide the exact code without access to the source code at the point of the vulnerability. However, the vulnerability was fixed in version 3.24.1. It allows an attacker with access to an unlocked physical device to bypass the Pin/passcode protection. The vulnerability is relatively less critical compared to higher-scoring vulnerabilities.

3.24.1.

When they have access to the unlocked physical device. This vulnerability could lead to unauthorized access to data stored in the device. The vulnerability was fixed in version 3.24.1 or later, which contains a fix for the vulnerability.

Regarding the affected Nextcloud Android app to the fixed version.

Regarding the GitHub repository: <https://github.com/nextcloud/security-advisories/security/advisories/GHSA-c3rf-94h6-vj8v>, as well as the pull request. The vulnerability was fixed in version 3.24.1 or later, which contains a fix for the vulnerability. By exploiting this vulnerability on an unlocked device that has the vulnerable version of the Nextcloud Android app installed. By exploiting this vulnerability, an attacker could potentially gain access to a potential heap buffer overflow that could allow a local user with System execution privileges to perform unauthorized actions.

Source: [source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

The vulnerability allows an attacker to gain access to the device exploiting the heap buffer overflow vulnerability to execute arbitrary code with elevated privileges.

The vulnerability allows an attacker to gain System execution privileges on the affected device.

The vulnerability is located in the `unpack_xtlv_cbf` function.

The affected versions are not listed in the provided data.

The vulnerability is categorized as 'MEDIUM' severity.

Original URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

The vulnerability allows an attacker to gain access to the device exploiting the heap buffer overflow vulnerability to execute arbitrary code with system privileges without requiring user interaction.

The vulnerability is located in the `unpack_xtlv_cbf` function of the file `dhd_rtt.c`. This could potentially allow an attacker to cause a local escalation of privileges.

The vulnerability is located in the `unpack_xtlv_cbf` function of the file `dhd_rtt.c`. This could potentially allow an attacker to cause a local escalation of privileges to the system.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

with system privileges, which could exploit the buffer overflow vulnerability to execute arbitrary code with escalation to the general public to prevent misuse. However, developers and security researchers use the vulnerability det

where a heap buffer overflow can occur. Exploitation of this vulnerability could lead to local escalation of privileg

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01)

ins to the Android kernel. Users should check with the Android security bulletin or device manufacturer for the sp CVE-2023-21076.

ite code with elevated privileges on the affected system.

as System execution privileges. This app or process could exploit the heap buffer overflow vulnerability in the cre e get\_svc\_hash function of nan.cpp, where there exists a possible out of bounds write due to a heap buffer overfl y. This indicates that the vulnerability presents a significant risk, but may not be as critical as those with higher seve ld potentially exploit the vulnerability without requiring any actions from the user.

: attacker must already have a high level of access to the affected system, making the vulnerability more difficult t re not mentioned in the provided information. Users should refer to Android's security bulletins or directly contac ing link: <https://source.android.com/security/bulletin/pixel/2023-03-01>. This will include details on the vulnerabili nd reference the specific issue within the Android ecosystem and its related documentation.

s writing arbitrary data outside the bounds of a heap-allocated buffer, leading to a heap buffer overflow. This cou sible out of bounds write due to a buffer overflow in 'rtt\_unpack\_xtlv\_cbfn' of 'dhd\_rtt.c'. This issue could lead to e in memory.

ording to the Common Vulnerability Scoring System (CVSS).

cker would require a certain level of access or control within the system to conduct the exploit.

page for the March 2023 Pixel Update Bulletin: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

obtained System execution privileges. The app could leverage the buffer overflow vulnerability by sending specia

id\_rtt.c. The vulnerability allows a possible out of bounds write, which could potentially be exploited to achieve lo privileges on the affected device.

erating system.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01)

level access exploiting the buffer overflow vulnerability in the dhd\_rtt.c file to perform an out of bounds write. Thi typically not shared publicly in order to prevent misuse. However, developers and security teams will analyze the y within the dhd\_prot\_ioctlcmplt\_process function of the dhd\_msgbuf.c file. It involves a possible out-of-bounds v JM in terms of severity according to the scale used by the Common Vulnerability Scoring System (CVSS).

ised on March 24, 2023. Users must review the Android security bulletin and associated patches for the exact affe vulnerability without the need for a user to perform any specific action.

ess to the system to exploit the vulnerability rather than being able to conduct the attack remotely.

that the vulnerability would likely be part of a larger attack chain, requiring initial access to be obtained by the att compromised a user's device through a separate vulnerability to gain system-level privileges. Once the device is c

bulletin, specifically the Pixel update bulletin for March 2023. The reference link is <https://source.android.com/security/bulletin/pixel/2023-03-01>. Users must refer to official Android security bulletins for detailed version information. To determine the vulnerable Android device and then using the debug policy flaw to gain higher privileges on the device without a missing user warning, no specific code example is provided. The vulnerability has been reported without detailed description.

on of `wl_roam.c` within the Android kernel. It occurs due to a missing bounds check which could potentially allow a

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

cally refer to various versions of the Android kernel prior to the security update that addressed this issue. To determine elevated privileges on the device. This could allow the attacker to execute arbitrary code with system privileges. For CVE-2023-21070 would involve applying the security update from the manufacturer that addresses the bounds check.

in the function `wl_update_hidden_ap_ie` of `wl_cfgscan.c`.

<https://source.android.com/security/bulletin/pixel/2023-03-01>.

the CVE description.

perform an out of bounds write operation. This could potentially result in a local escalation of privilege, allowing the attacker to bypass an array or memory buffer without proper bounds checking. While the specific code for CVE-2023-21069 is not

ability of booting with a hidden debug policy without showing any warning to the user.

description.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

a vulnerable Android device and then using the debug policy flaw to gain higher privileges on the device without a missing user warning, no specific code example is provided. The vulnerability has been reported without detailed

ability are not provided in the description. It impacts certain versions of Android.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

mentioned in the provided data. Users must refer to official Android security bulletins for detailed version information. To determine the vulnerable Android device and then using the debug policy flaw to gain higher privileges on the device without a missing user warning, no specific code example is provided. The vulnerability has been reported without detailed description. Exploitation will be specific to the nature of the vulnerability which is not disclosed in the provided information. To determine authorized access to a device, escalating privileges to gain control of the system, remote code execution, or system boot kernel, which could lead to a possible out of bounds write due to an integer overflow. This flaw can result in local privileges on the affected Android device. User interaction is not required to carry out the attack.

ed in the vulnerability details.

ally the Pixel update bulletin for March 2023, available at <https://source.android.com/security/bulletin/pixel/2023-03-01>.

with system-level execution privileges to perform out of bounds write operations, and potentially gain unauthorized access to perform out of bounds write operations due to an integer overflow. This can lead to the compromise of the Android system.

on Android. This issue stems from a missing bounds check that could potentially allow a local malicious application to perform out of bounds write operations due to an integer overflow. This can lead to the compromise of the Android system.

However, it is mentioned that the product affected is the Android kernel. It would be important to check the associated Pixel updates at: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

Due to this vulnerability, they could achieve local escalation of privilege which might allow them to execute arbitrary code on the device. Bulletin at: <https://source.android.com/security/bulletin/pixel/2023-03-01>. Additional references might be available for this vulnerability on an Android device. Since no user interaction is required, this app could exploit the vulnerability in the DoSetPinControl function.

The attacker could execute code as the System user without requiring user interaction.

The vulnerability exists in the DoSetPinControl function on the affected Android device.

The vulnerability has a base severity score of 6.7, which is classified as MEDIUM.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

The details of the vulnerability are not provided in the CVE description.

On the device, could exploit the out of bounds read vulnerability found in simdata.cpp without any user interaction. The details of the vulnerability are not provided in the CVE description. Moreover, sharing such code examples could be irresponsible as it could aid potential attackers.

This vulnerability is characterized by a possible out-of-bounds read resulting from an incorrect bounds check, which is a common issue in the standards for rating CVE impacts.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01)

The vulnerability is an out-of-bounds read vulnerability in the DoSetTempEcc function. Since no user interaction is required, the attack could be performed remotely for CVE-2023-21062 within the provided information, as it only references the vulnerability details and its impact.

Android ID A-229255400.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

How to exploit vulnerabilities like CVE-2023-21061.

The vulnerability could potentially allow an attacker to perform actions such as unauthorized access or denial of service. An attack could be performed remotely.

The vulnerability could lead to remote information disclosure without requiring additional execution privileges. The software affected is the Android system.

The details of the vulnerability are not provided in the CVE description. The vulnerability is referenced on the Android security bulletin page, at the following URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>. The vulnerability could be exploited remotely without any interaction from the user.

The vulnerability could be used to send an SMS message to a target device. The malicious SMS would contain data that, when processed by the sms\_GetTpPile function, could lead to the disclosure of sensitive information. However, because the issue is patched and referenced in a specific Android Security Bulletin, it is likely that the vulnerability is no longer present in the affected versions.

The vulnerability is an out of bounds read caused by a missing bounds check.

an privileges, and user interaction is not required for the exploit to work.  
Devices of Android impacted by this vulnerability were not specified in the given information.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

This vulnerability involves an attacker crafting a malicious input that is not properly bounds-checked in the EUTRAN\_LCS\_DecodeFacility function, leading to a potential out-of-bounds write issue due to a missing bounds check. This flaw can lead to remote code execution on the affected Android platform.

The affected versions have not been detailed in the provided information.

Android platform.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01)

This vulnerability involves an attacker crafting a malicious input that is not properly bounds-checked by lcsml\_SendRrAcquiAssist function, leading to a potential out-of-bounds write issue due to a missing bounds check. This flaw can lead to remote code execution on the affected Android platform. CVE-2023-21057.

This vulnerability could potentially lead to remote code execution with no additional execution privileges required.

The affected versions have not been detailed in the provided information.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

This vulnerability involves an attacker crafting malicious TCP/IP packets to an Android device with the vulnerable kernel version. By exploiting the out of bounds write issue, the attacker could potentially lead to remote code execution with no additional execution privileges required.

This vulnerability involves a buffer overflow in the lcsml\_buffer\_free function of lcsml\_device\_slc.c. This issue has the potential to allow a local attacker to escalate their privileges to System execution privileges without requiring any action from the user.

The affected versions have not been detailed in the information provided.

Link URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

This vulnerability involves a buffer overflow in the lcsml\_device\_slc.c to cause memory corruption and execute arbitrary code with System privileges. This vulnerability is due to a possible use after free because of a race condition, which could potentially lead to remote code execution on the affected Android kernel. This vulnerability is due to a possible use after free because of a race condition, which could potentially lead to remote code execution on the affected Android kernel. This vulnerability is due to a possible use after free because of a race condition, which could potentially lead to remote code execution on the affected Android kernel. This vulnerability is due to a possible use after free because of a race condition, which could potentially lead to remote code execution on the affected Android kernel.

The affected versions have not been detailed in the provided information.

This page, specifically at this URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>

This vulnerability involves an attacker crafting malicious TCP/IP packets to an Android device. They could exploit the race condition vulnerability in the lcsml\_device\_slc.c file found in the Android Kernel. This vulnerability is due to a possible use after free because of a race condition, which could potentially lead to remote code execution on the affected Android kernel. This vulnerability is due to a possible use after free because of a race condition, which could potentially lead to remote code execution on the affected Android kernel.

om/security/bulletin/pixel/2023-03-01.

rs the out of bounds write logic error in the LPP\_CommonUtil.c function. Since no user interaction is required and issue are not readily available to the public in order to prevent misuse. However, the issue lies within the EUTRA

ractCbLanguage function of sms\_CellBroadcast.c. This vulnerability could lead to remote information disclosure v

305610 was assigned. However, the exact versions prior to this ID are not mentioned in the provided details.

; and fixing the vulnerability would involve reviewing the implementation of sms\_ExtractCbLanguage in the Andrc  
https://source.android.com/security/bulletin/pixel/2023-03-01.

otely without any user interaction.

interaction, possible attack scenarios could include a malicious actor sending a crafted cell broadcast message to

ere there is a potential out of bounds write due to a missing bounds check. This flaw could lead to a local escalatic  
nerability.

roid ID: A-259063189. Specific version numbers are not provided in the vulnerability details.

tin located at: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

privileges.

ccess writing data out of bounds in the 'setToExternal' function of the ril\_external\_client.cpp file. Since there's no  
ict code examples for CVE-2023-21052. Code examples would typically detail how the out of bounds write occurs

ker could write out of bounds due to an incorrect bounds check, leading possibly to system execution privileges.

om/security/bulletin/pixel/2023-03-01

cific code example would be speculative. Generally, the vulnerability is likely in the logic that performs bounds ch  
execution privileges on an Android device, exploiting the out of bounds write vulnerability to further escalate the

cpp, where there is a possible out-of-bounds write due to improper input validation. This issue could lead to a loc

om/security/bulletin/pixel/2023-03-01

Android device uses the vulnerability in the load\_png\_image function to perform an out-of-bounds write operati  
era\_metadata.c` within the Android operating system. This issue presents a possible out-of-bounds read due to a

y.

ited by an attacker who has system execution privileges without any action from the user.

een disclosed in the provided information.







ion privileges could potentially read sensitive data from outside the bounds of intended memory structures within the kernel, which is part of the Android kernel. It involves a possible out of bounds write due to a use after free, which could lead to a denial of service. Devices susceptible to this vulnerability have not been detailed in the information provided.

<https://source.android.com/security/bulletin/pixel/2023-03-01>

Access to the device taking advantage of the use after free vulnerability in the `cs40l2x_cp_trigger_queue_show` function.

It results in a possible failure to truncate images within the Android operating system.

However, it is implied that this issue pertains to versions of the Android operating system prior to the security bulletin.

<https://source.android.com/security/bulletin/pixel/2023-03-01>.

For CVE-2023-21036. It's advisable to keep an eye on credible sources and security bulletins for any updates.

A logic error in `BitmapExport.java` to create or modify images in a way that they are not properly truncated as expected.

Another app with the same package name, leading to a permissions bypass. This could allow for local escalation of privileges.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

An app with the same package name as a legitimate app that has already been granted certain permissions. The malicious app could potentially exploit these permissions.

It involves a possible access to accurate sensor data due to a permissions bypass, which could lead to local escalation of privileges. This reflects a significant impact that the exploitation of this vulnerability could have on an affected system. Users are advised to update their software to mitigate the risk associated with this vulnerability.

It is more dangerous as it can be exploited without any actions from the user, potentially leading to unauthorized access to sensor data while the vulnerability is classified as a local escalation of privilege, the attacker would first need access to use the device, specifically on the page for the Pixel update of March 2023. The link is as follows: <https://source.android.com/security/bulletin/pixel/2023-03-01>. Access to sensor data, which might contain sensitive information such as GPS data or activity recognition. This exploit uses `UserService.cpp` to bypass permission checks. Even without requesting the necessary permissions from the user, the vulnerability leads to resource exhaustion is CVE-2023-21033.

For more information, visit the security bulletin website at the following URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

Execution privileges to be exploited.

Vulnerability to cause a denial of service on the device by exhaustively creating network configurations. This can be mitigated by the device manufacturer or the Android Open Source Project (AOSP). Users should ensure their devices are up to date to avoid resource exhaustion.

A logic error in `t_node_to_fdt` of `ufdt_convert.c`, leading to a potential local information disclosure.

oid.com/security/bulletin/pixel/2023-03-01.

l privileges on the device. They might exploit the heap buffer overflow in the `ufdt_convert.c` by manipulating the ..  
ie `setPowerMode` function of `HWC2.cpp`. This issue could lead to local information disclosure without the need for a patch was identified.

the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

in of Android's `HWC2.cpp` file potentially leading to out of bounds memory access. This could result in unauthorized

on Android. It involves a potential memory corruption issue due to a double free, leading to local escalation of privileges that it poses a serious risk and should be addressed promptly to prevent potential exploitation. The issue could be exploited without the user's knowledge.

source.android.com/security/bulletin/pixel/2023-03-01.

memory is freed twice. Exploitation of this vulnerability could lead to memory corruption, which an attacker could leverage to gain root access. This issue was reported in the Android Security Bulletin or device manufacturers. Ensuring that the affected Android devices are running on the latest version of the operating system is the best way to ensure that the memory address has not already been freed. Additionally, setting the pointer to `0` before freeing it is a good practice.

function of `UidObserverController.java`. This flaw could lead to a local information disclosure of app usage. Exploitation of this vulnerability means that the attacker would need to have access to execute commands or run applications as a regular user on the device. The issue potentially takes advantage of the vulnerability without requiring any action from the user. The issue is rated as 5.5, which is classified as MEDIUM.

m/security/bulletin/pixel/2023-03-01

user-level access to an Android device. The attacker could write a malicious application that accesses the `UidObserver` class.

!1028.

-terminator, leading to remote information disclosure without the need for additional execution privileges.

source.android.com/security/bulletin/pixel/2023-03-01.

et Printing Protocol) packet to the vulnerable device. Since the `parse_printerAttributes` function doesn't expect a `0`, the device will crash. The issue was reported in the Android Security Bulletin for their devices. This may involve updating the device's software to the latest version of the operating system.

authentication misconfiguration stemming from a logic error in the code.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

ntXmlUtils.java file's multiple functions, compromising the security of the affected Android system.

affected network information to a vulnerable Android device. Due to the logic error and misconfiguration, the attack uses the URL provided, but there is no specific information here indicating whether it has already been exploited in the wild. CVE-2023-21026.

for Android. It involves a logic error that allows setting a touchable region beyond its own SurfaceControl, potentially leading to a local escalation of privilege. The severity of this vulnerability is rated as 5.5.

For more information, see the URL <https://source.android.com/security/bulletin/pixel/2023-03-01>.

This vulnerability involves a region being set beyond its own SurfaceControl. User interaction is not required for exploitation.

The attack exploits the vulnerability in the WindowManagerService to create an overlay or invisible touchable region. This vulnerability can be exploited locally without any user interaction.

The exploit is implemented in the overlay.c file, which is part of the Android operating system. Specifically, this vulnerability is related to an out of bounds write in the ufdt\_local\_fixup\_prop function of ufdt\_overlay.c. This type of vulnerability can lead to local information disclosure. The severity of this vulnerability is rated as a MEDIUM severity level. This score indicates that the vulnerability represents a moderate risk.

Local execution privileges could potentially exploit the vulnerability without any action from the user.

For more information, refer to Android's security bulletins or updates for detailed information regarding patches or mitigation measures.

For more information, see the full Android security bulletin page, specifically the entry for the March 2023 Pixel update. The URL is <https://source.android.com/security/bulletin/pixel/2023-03-01>. The attack exploits the out of bounds write vulnerability to gain unintended System execution privileges on an Android-13 device. The malicious app could leverage the out of bounds write vulnerability of the lockdown screen in FallbackHome.java due to a logic error. This flaw could lead to local escalation of privilege.

Exploitation methods and code are typically not disclosed publicly in order to prevent the vulnerability from being widely exploited.

For more information, see the URL [source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

The attack exploits the FallbackHome.java to delay the activation of the lockdown screen. This could give the attacker time to perform unauthorized actions.

The exploit is implemented in Suballocation.cpp, which could lead to local escalation of privilege with no additional execution privileges needed.

For more information, see the URL [source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

The out of bounds write vulnerability does not require any user interaction. The application can leverage this flaw to corrupt data without access to the specific source code of Suballocation.cpp. The issue lies within 'BufferBlock' where out of bounds writes occur.

The attack can be used to change the network settings of an admin user due to a missing permission check. This could result in a local escalation of privilege.

For more information, refer to security bulletins or other references provided by the vendor. Users should check the provided reference, such as the URL <https://source.android.com/security/bulletin/pixel/2023-03-01>.



on Android, specifically the Android-13 version. The issue is a possible permission bypass due to the absence of a r  
e vulnerability presents a significant risk and impact, suggesting that it is important for affected systems to be pat

.android.com/security/bulletin/pixel/2023-03-01. It is advisable to review official security bulletins and document

Android device. Since the vulnerability does not require user interaction for exploitation, the app could silently ex  
exploit the vulnerability silently, making it more severe as users may not be aware that their device's security has

y allowing an attacker with System execution privileges to access sensitive data.  
21014.

y.

ly at the URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.  
privileges.

execution privileges on the device, intentionally triggering an out of bounds read in the p2p\_iface.cpp code to access  
ould be inappropriate and potentially harmful. Generally, security researchers and the manufacturers provide info  
:apd.cpp file, related to Android's network management. This flaw is a possible out-of-bounds read due to a missi  
d.cpp file, which is part of the network management functionalities in Android operating systems.

se. No user interaction is required for exploitation.

om/security/bulletin/pixel/2023-03-01  
; an attacker access to sensitive data on the affected device.

process with System level permissions reading unauthorized memory regions within the hostapd service to glean

multiple locations of p2p\_iface.cpp.  
3-21012.

ecurity/bulletin/pixel/2023-03-01.  
System privileges reading beyond the bounds of allocated memory in p2p\_iface.cpp, possibly leading to the discl

a possible out of bounds read due to a missing bounds check. This could potentially lead to local information discl  
s means the attacker must already have a high level of control over the system, which significantly reduces the ris

lity.

<https://source.android.com/security/bulletin/pixel/2023-03-01>.  
An attacker with system execution privileges could read data from memory they are not supposed to have access

lead to local information disclosure.

<https://www.mcafee.com/security/bulletin/pixel/2023-03-01>.

out of bounds read vulnerability in 'p2p\_iface.cpp' without requiring user interaction. Since the attack requires s  
ains to a missing bounds check, developers should ensure they perform proper bounds checking whenever accessi  
code component within Android, specifically version Android-13. It is characterized by a possible out of bounds re

privileges.

'security/bulletin/pixel/2023-03-01

reading data out of bounds, which might lead to the disclosure of sensitive information that could be used to further

.cpp.

presents a moderate risk.

<https://source.android.com/security/bulletin/pixel/2023-03-01>.

talled on the device and has the necessary System execution privileges. This application could exploit the out of bounds read issue in the `getSystemProperty` method of the `SystemProperties` class in the `android.os` package in the `android-13` version of the Android OS. This vulnerability is an out-of-bounds read issue that occurs due to a missing bounds check, within this file where proper bounds checking is not implemented, potentially leading to an out-of-bounds read. Exploitation of this vulnerability requires system execution privileges, which adds a layer of

ed Android device. This high level of access suggests that the attacker would need to have already bypassed other security mechanisms. The vulnerability could be exploited without the knowledge or input of the user, as long as the attacker has the necessary system-level privileges. For more information, see the following link: <https://source.android.com/security/bulletin/pixel/2023-03-01>. Users and system administrators are encouraged to update their devices to the latest version of Android to mitigate this vulnerability. This vulnerability affects Android devices with system-level execution privileges. The app could attempt to read beyond the expected bounds of memory, potentially leading to an out of bounds read due to a missing bounds check. This issue affects Android versions 12 and 13. For more information, see the specific issue within the Android ecosystem and can be helpful when looking up the vulnerability in Android Security Center. I cannot provide actual code examples that demonstrate the specific vulnerability of CVE-2023-21007. I am not a security expert, and my analysis is based on the information provided. The component, potentially leading to an out of bounds read due to a missing bounds check. This issue affects Android versions 12 and 13. For more information, see the specific issue within the Android ecosystem and can be helpful when looking up the vulnerability in Android Security Center. I cannot provide actual code examples that demonstrate the specific vulnerability of CVE-2023-21007. I am not a security expert, and my analysis is based on the information provided.

Information stored on the device could be accessed by an attacker with system execution privileges.

malicious app running on the Android device with system privileges. The app could manipulate the process that in 'https://source.android.com/security/bulletin/pixel/2023-03-01'.

ssion Controllers on Android is CVE-2023-21005.

ssion Controllers on Android. It involves a missing permission check, which can lead to a local escalation of privilege. This vulnerability is considered to be a significant risk.

st version to mitigate this issue.

ing URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

21005, which increases the ease of exploitation.

ccessfully exploited, an attacker could potentially gain unauthorized access to additional system resources or perform privileged activities without the need for additional execution privileges. This issue allows for a possible permission bypass due to a missing permission check in the Transcode Permission Controllers on the Android platform.

ould gain elevated permissions on a device without the need for additional execution privileges.

ing URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

ability can be exploited with the current privileges of the attacker's code on the device.

lly on an Android device, potentially leading to full control over the device and access to sensitive data or system resources. This vulnerability targets the Transcode Permission Controllers in Android 13. Once installed on a victim's device (without the need for additional execution privileges), it could lead to local escalation of privilege without requiring any action from the user. Android Security Bulletin, indicating that Google has acknowledged the issue. Users should refer to Android security updates within Android. It involves a missing permission check that could potentially allow a local attacker to escalate privilege.

ould elevate their permissions to gain additional control without requiring further execution privileges.

ulnerability without requiring any action from the user.

Following URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>

device, exploiting the missing permission check to perform actions or access data that it should not have the privilege to.

availabilityStatus of several Transcode Permission Controllers. It could lead to local escalation of privilege without requiring any action from the user.

<https://source.android.com/security/bulletin/pixel/2023-03-01>.

in the getAvailabilityStatus function of the Transcode Permission Controllers. The missing permission check allows an attacker to gain elevated permissions on a device without the need for additional execution privileges.

rating system, specifically version Android-13. It is a local escalation of privilege issue that arises from a missing permission check. This vulnerability poses a significant risk and should be addressed promptly.

derSettings.java, which could allow unauthorized changes to Wi-Fi settings by non-privileged users.

as it could be exploited silently without the user's knowledge.

ed on the device, which uses the vulnerability to programmatically change the Wi-Fi settings without the user's consent. Android Security Bulletin, specifically at the URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

sible use after free caused by improper locking. An attacker could exploit this vulnerability to gain higher privilege.



source.android.com/security/bulletin/pixel/2023-03-01.

the use after free vulnerability in MediaCodec.cpp. The improper locking issue could allow the attacker's app to ex

CVE-2023-21000 involves a use after free vulnerability which may happen, in a general sense, when a piece of me  
ces is CVE-2023-20999.

itions could potentially trigger a persistent reboot loop. This defect could lead to a local denial of service since Us

y Bulletin webpage at <https://source.android.com/security/bulletin/pixel/2023-03-01>.

eges.

out requiring user interaction to cause the device to enter a persistent reboot loop, thus rendering the device un  
em. The vulnerability stems from improper input validation in multiple locations, which can be exploited to trigge

source.android.com/security/bulletin/pixel/2023-03-01.

ces and advisories, such as the Android Security Bulletin, offer technical insights and mitigation guidance without  
on privileges and triggering the vulnerability to cause a persistent reboot loop, intentionally causing a local denia  
ndroid version 13. The issue is caused by improper input validation in multiple locations, which could potentially i  
e the vulnerability can have a significant impact on the availability of the system, it has a lower impact on its confi  
ce. This suggests that the attacker must either have physical access to the device or have already compromised th  
ges can trigger the vulnerability without any input from the user, which makes the vulnerability more concerning.

e at the following URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

User execution privileges on an Android device running Android 13. The attacker could exploit the improper input  
provided by Google for their Android devices. The Android Security Bulletin, where this vulnerability is mentioned,  
ssible way to trigger a persistent reboot loop due to improper input validation. This issue could result in local den  
n Vulnerability Scoring System (CVSS).

://source.android.com/security/bulletin/pixel/2023-03-01

y for exploitation.

improper input validation vulnerability without requiring interaction from the user. Once the application is execu  
validation that can lead to a persistent reboot loop.

rating system, specifically in the captureImage function. It involves a logic error that allows for the possibility of b  
d as HIGH severity.

an attacker without needing any action from the victim.

g URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>.

tions or access resources that are normally protected and not available to a standard user or application.

function of CustomizedSensor.cpp without needing any additional privileges. Since no user interaction is required, vulnerabilities like CVE-2023-20995 would usually require access to the source code or detailed understanding of the

direct bounds check in the function \_ufdt\_output\_property\_to\_fdt of ufdt\_convert.c. This could potentially allow a

ibility presents a moderate level of risk.  
[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01)

an execution privileges on a device running Android-13. The application could use the vulnerability in the \_ufdt\_output in the Android operating system. This vulnerability could allow for a local escalation of privilege without the need for an entity could gain elevated privileges on a device without requiring additional execution rights.

release on the 1st of May 2023, as detailed in the provided reference link to the source.android.com security bulletin. The severity of the vulnerability, implying that it poses a significant threat to systems if exploited. The vulnerability is located in the SnoozeHelper.java module due to an uncaught exception. This failure allows for an exploitation path leading to local privilege escalation. The vulnerability is caused by a missing bounds check in the on\_iso\_link\_quality\_read function within btm\_iso\_impl.h. This vulnerability is caused by a missing bounds

MEDIUM'.  
without any user engagement.  
The vulnerability could potentially be exploited by someone with a high level of access to the system.

Security Bulletin link provided in the reference: <https://source.android.com/security/bulletin/pixel/2023-06-01>.  
This vulnerability is located in the ble\_scanner\_hci\_interface.cc within the Android Bluetooth system. It is caused by a missing bounds check that

Security Bulletin page at: <https://source.android.com/security/bulletin/pixel/2023-06-01>  
This vulnerability could be exploited on the compromised device.

This vulnerability allows an attacker to perform an out of bounds read, potentially leading to access to sensitive information stored in memory. The vulnerability is located in the n\_ble.cc in Android systems. It involves a possible out of bounds read due to a missing bounds check which could be exploited on an Android device.  
This vulnerability can be exploited without the user's awareness.  
The severity of this vulnerability is

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01)

This vulnerability can lead to local information disclosure.

s already installed on the device or has gained system privileges through some other means. The application could exploit CVE-2023-20990 is available publicly. Typically, details of how to exploit specific vulnerabilities are not shared widely to

btm\_ble\_gap.cc within Android where there is a potential out of bounds read caused by a missing bounds check. This

reference link being: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

level privileges reading out-of-bounds memory in the btm\_ble\_write\_adv\_enable\_complete function. Since user information is not typically provided to prevent misuse. However, the vulnerability is due to a missing bounds check in the btm\_ble\_write\_adv\_enable function of the bta\_gatts\_api.cc file in Android. It involves a possible out of bounds write issue due to improper initialization as HIGH severity. This indicates that it poses a significant risk and should be addressed promptly.

in the btm\_read\_rssi\_complete function of btm\_acl.cc. This issue could potentially lead to local information disclosure.

[pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

CVE-2023-20988 by executing crafted code that triggers an out of bounds read in the Bluetooth server component. This vulnerability is available publicly at <https://source.android.com/security/bulletin/pixel/2023-06-01>. The out of bounds read occurs due to a missing bounds check. This issue could potentially result in local information disclosure.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01)  
CVE-2023-20987.

an Android device exploiting the out of bounds read vulnerability in the Bluetooth stack to access sensitive information. There are no publicly available code examples detailing the specific nature of the exploit. Such information is typically not provided to prevent misuse. However, the vulnerability is due to a missing bounds check in the btm\_ble\_clear\_resolving\_list\_complete function of btm\_ble\_privacy.cc. It involves a possible out of bounds read issue due to improper initialization as HIGH severity. This indicates that it poses a significant risk and should be addressed promptly.

ion privileges on a device running Android 13 could exploit the out of bounds read vulnerability in the btm\_ble\_clear\_resolving\_list\_complete function of the btm\_ble\_privacy.cc file in Android. It involves a possible out of bounds write issue due to improper initialization as HIGH severity. This indicates that it poses a significant risk and should be addressed promptly.

apply updates provided by Google to mitigate the vulnerability. Additionally, no user interaction is required for the exploitation, making it more dangerous as the attack can be carried out silently. The vulnerability is listed in the references, specifically the Pixel update bulletin for June 2023 available at <https://source.android.com/security/bulletin/pixel/2023-06-01>.

apply updates provided by Google to mitigate the vulnerability.

ly, no user interaction is required for the exploitation, making it more dangerous as the attack can be carried out silently. The vulnerability is listed in the references, specifically the Pixel update bulletin for June 2023 available at <https://source.android.com/security/bulletin/pixel/2023-06-01>. Possible disclosure practices, a hypothetical attack scenario could involve a malicious application sending malformed data to the Bluetooth stack.

or application could gain higher privileges on the device than intended. This could enable them to access sensitive data on the official Android security bulletin page specific to the Pixel update for June 2023. It is advisable to refer to the `btif_btcc` function of the `btif_btcc.cc` file in Android, specifically Android version 13. The issue is a possible out of bounds write. This means it poses a considerable risk, but it is less critical than those with higher severity ratings, such as CVE-2023-20982.

An attacker could potentially exploit the vulnerability without the need for the user to take any action.

Access to an affected device and then reading memory out of bounds in the affected function. This could lead to information disclosure. You can refer to the details provided at <https://source.android.com/security/bulletin/pixel/2023-06-01>, which is the official Android security bulletin for June 2023.

On an affected Android device. This level of access is typically higher than what ordinary apps are granted and would require the user to grant permissions to the application.

The vulnerability is a possible out of bounds read due to a missing bounds check. This flaw could potentially lead to local information disclosure. Exploitation of this vulnerability requires that the attacker has access to the device and is needed for exploitation.

[source.android.com/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

The vulnerability involves a possible out-of-bounds read due to a missing bounds check, leading to potential local information disclosure. The application could then exploit the out of bounds read vulnerability to read sensitive data from the device. It involves a possible out-of-bounds read due to a missing bounds check, leading to potential local information disclosure. The application categorizes it as having MEDIUM severity according to the Common Vulnerability Scoring System (CVSS).

An attacker could exploit CVE-2023-20982 to read data out-of-bounds. This information disclosure could potentially reveal system memory contents, provided they have System execution privileges.

The vulnerability is detailed in the Android Security Bulletin, although the specific detail regarding a fix is not given in the provided information. Typical references are at the following URL: <https://source.android.com/security/bulletin/pixel/2023-06-01>. This source often provides detailed information about the vulnerability on the affected device.

The vulnerability is located in the `btif_btcc.cc` file, which is part of the Bluetooth stack in Android. It involves a possible out-of-bounds read due to a missing bounds check. An attacker with System execution privileges could potentially access sensitive data from the device without any user interaction. The data exposure is limited to the application's access to security bulletins and update their systems to protect against vulnerabilities like this.

The vulnerability is not easily exploitable by an attacker without requiring any action from the device user, which increases the potential risk if an attacker has the ability to interact with the device. Google or device manufacturers. Google periodically releases security updates to address vulnerabilities like this. The vulnerability is categorized as a MEDIUM severity vulnerability.

For more information, refer to the Pixel security bulletin for June 2023. The URL for direct reference is <https://source.android.com/security/bulletin/pixel/2023-06-01>.

The vulnerability involves a possible out-of-bounds read due to a missing bounds check, leading to potential local information disclosure. An attacker with System execution privileges, or an attacker who has already attained such privileges through some other exploit. Given the severity of the vulnerability, it is not considered as critical but should be treated as a MEDIUM severity vulnerability. An attacker with System execution privileges could exploit the vulnerability to read memory contents that they should not be able to access. The vulnerability is detailed in the Android Security Bulletins. According to the reference provided, users should consult the security bulletin at <https://source.android.com/security/bulletin/pixel/2023-06-01> for more information about the vulnerability on the affected Android device. This means that the vulnerability requires a higher level of access and is not easily exploitable by an attacker without requiring any action from the device user, which increases the potential risk if an attacker has the ability to interact with the device. They are typically published in databases like the National Vulnerability Database (NVD) so that users and administrators can be aware of the vulnerability and take appropriate action. The vulnerability affects the `bta_av_co.cc` function of the Bluetooth stack, which affects the Android operating system.

The vulnerability involves a possible out of bounds read due to a missing bounds check, leading to potential local information disclosure, where an attacker could access sensitive data on the device without needing any additional privileges.

irity/bulletin/pixel/2023-06-01

ly trigger the out of bounds read vulnerability in the getNextSourceDataPacket function without needing any permission, particularly the btm\_ble\_read\_remote\_features\_complete function located in btm\_ble\_gap.cc. This vulnerability allows an attacker to gain unauthorized access to confidential information stored on the device by exploiting the flaw in the Android

security risk it poses.

romised firmware. This indicates the need for a fairly high level of access, making it less likely for an average user to be exploited autonomously by malicious software that has already established the required privileges on the device.

cifically at: <https://source.android.com/security/bulletin/pixel/2023-06-01>

stem execution privileges, either through exploiting another vulnerability or through user-installed firmware/software. This could lead to the disclosure of sensitive information that could be used for further attacks beyond the intended bounds, leading to the disclosure of sensitive information that could be used for further

misleading the user to select the default autofill application due to improper input validation, potentially leading to a

ver, user interaction is required for exploitation.

om/security/bulletin/pixel/2023-06-01.

t to the DefaultAutofillPicker component, which is not properly validated by the application. If they can then trick

al escalation of privilege.

RL: <https://source.android.com/security/bulletin/pixel/2023-06-01>.

or security reasons. It involves an error in the getAvailabilityStatus method in EnableContentCapturePreferenceController. This could lead to a pass vulnerability to ignore the DISALLOW\_CONTENT\_CAPTURE setting. Without any need for user interaction, th

\_complete is CVE-2023-20974.

ble\_privacy.cc file in the Android source code.

oid.com/security/bulletin/pixel/2023-06-01.

ould lead to local information disclosure on the affected Android device.

privileges exploiting this vulnerability to access potentially sensitive data from memory that it should not have access to.

g bounds check in btm\_create\_conn\_cancel\_complete of btm\_sec.cc. This could potentially lead to local information disclosure.

om/security/bulletin/pixel/2023-06-01.

ata out of the bounds of an expected memory buffer, leading to information disclosure. Given system-level privilege  
the 'btm\_devctl.cc' file. This vulnerability is due to a missing bounds check that could lead to local information di

om/security/bulletin/pixel/2023-06-01.

privileges might read memory out of bounds within the 'btm\_vendor\_specific\_evt' function. Despite not requiri  
ServiceImpl.java in Android. It involves a logic error that allows the acquisition of dangerous permissions without

can be found at <https://source.android.com/security/bulletin/pixel/2023-06-01>.

PermissionManagerServiceImpl.java to grant itself dangerous permissions without the user's knowledge or cons  
ific exploitation code is not shared to prevent misuse; rather, vendors release patches or updates to mitigate the i

access to sensitive information.

source.android.com/security/bulletin/pixel/2023-03-01.

ple issue in C++ could look like this: accessing an array without checking if the index is within the bounds of the a  
2p\_iface.cpp by manipulating inputs or the state of an Android system without checking array bounds. Due to ins  
ace.cpp is CVE-2023-20969.

ds check in p2p\_iface.cpp, which could result in local information disclosure.

privileges, and no user interaction is needed for exploitation.

om/security/bulletin/pixel/2023-03-01.

eges exploiting the out of bounds read vulnerability in p2p\_iface.cpp. The attacker could potentially access sensit  
20968.

ng bounds check in multiple functions of p2p\_iface.cpp.

[m/security/bulletin/pixel/2023-06-01](https://source.android.com/security/bulletin/pixel/2023-06-01).

is exposed by the absence of a necessary bounds check in p2p\_iface.cpp functions. This could result in an out-of-bounds write caused by a heap buffer overflow. This issue could be exploited by an attacker to act

Android 13.

at this URL: <https://source.android.com/security/bulletin/2023-03-01>.

an attacker could potentially gain higher-level permissions on a device without requiring additional execution privileges. This vulnerability has a high impact on the compromised system, indicating a serious level of severity.

Record.java, leading to a confused deputy problem. This vulnerability could result in local denial of service or privilege escalation.

the following link: <https://source.android.com/security/bulletin/2023-03-01>.

It intends that the MediaSessionRecord is not expecting to handle. Since the vulnerability involves a confused deputy problem, it has a high impact on the compromised system, indicating a serious level of severity.

Android 13.

the following URL: <https://source.android.com/security/bulletin/2023-03-01>.

and user or process could gain higher privileges on the device.

It exploits the parcel mismatch to gain elevated privileges. Since no user interaction is required, the attack could

be possible of starting a foreground activity from the background because of an unsafe PendingIntent.

3.

represents a moderate level of risk.

62.

JURL: <https://source.android.com/security/bulletin/2023-03-01>

When exploiting the unsafe PendingIntent to start a sensitive or private foreground activity without the user's knowledge, the use of a PendingIntent that is not properly secured. For example, using a PendingIntent like below without permission on the Android operating system. Specifically, it involves a potential method to launch arbitrary activities due to a lack of proper validation. This vulnerability could have significant impact if exploited, leading to a substantial level of privilege gain without any action from the user.

<https://source.android.com/security/bulletin/2023-03-01>.

user's device. This application, without requiring any user interaction, could exploit the vulnerability to launch an activity without requiring elevated privileges to start with, exploiting this vulnerability could result in an attacker gaining elevated privileges without proper permission checks, leading to a local escalation of privilege without needing any additional execution privileges.

[/source.android.com/security/bulletin/2023-03-01](https://source.android.com/security/bulletin/2023-03-01).

gain unauthorized access or permissions on the device without needing to execute additional code.

edUserActivity without needing to interact with another user or requiring additional execution rights. This could

uffer overflow in the read\_paint function of ttcolr.c. This issue may lead to local information disclosure without r  
the vulnerability is considered to be a significant risk and should be prioritized for remediation efforts due to its p

lletin at the following URL: <https://source.android.com/security/bulletin/2023-03-01>.

erflow when processed by the read\_paint function of ttcolr.c. The impact could be local information disclosure, pc

023 is CVE-2023-20957.

ettingsPreferenceFragment.java, due to a confused deputy problem. This could allow a local attacker to escalate

nd Android 12L.

ty Bulletin webpage with the URL: <https://source.android.com/security/bulletin/2023-03-01>.

vice, enabling them to bypass Factory Reset Protections and potentially access or modify sensitive device settings

system component (the 'deputy') is tricked by malicious software into misusing its privileges to perform operation  
y install all the latest security updates and patches released by manufacturers for their specific device models. Sir

issing bounds check, resulting in local information disclosure.

2.

om/security/bulletin/2023-03-01.

an Android device exploiting the vulnerability to conduct an out of bounds write operation. The attacker could lev  
e proprietary source code of the affected component, a generic example of an out of bounds write in C++ might be  
tified by the CVE ID CVE-2023-20955.

dFragment.java in Android. The vulnerability is due to a missing permission check that could potentially allow an  
12, Android 12L, and Android 13. As it could lead to a local escalation of privilege, an attacker could exploit this v  
ilects the potential serious impact of the vulnerability if successfully exploited.

Bulletin page: <https://source.android.com/security/bulletin/2023-03-01>.

on check to uninstall applications arbitrarily without the device owner's consent. Since the vulnerability doesn't re  
n of sdp\_db.cc, which is part of the Android operating system. This vulnerability is an out of bounds write issue ca  
id Android 13.

erability Scoring System) base score of 9.8. This indicates that the vulnerability poses a serious risk to affected sys

nce a successful attack could occur without the user's knowledge.

2 following URL: <https://source.android.com/security/bulletin/2023-03-01>



ed data packets to an Android device, exploiting the out of bounds write vulnerability in the SDP\_AddAttribute fu  
ind theft to the installation of additional malicious software on affected devices. Remote code execution capabilit  
ere is a potential method to bypass the factory reset protection. This is due to incorrect user interface elements b  
ccess to the device's resources without the need for additional execution privileges.

<https://source.android.com/security/bulletin/2023-03-01>.

ue arises from a missing bounds check which could result in a possible out of bounds write. This vulnerability may  
and Android 13.

[source.android.com/security/bulletin/2023-03-01](https://source.android.com/security/bulletin/2023-03-01).

Android 13.

[://source.android.com/security/bulletin/2023-03-01.](https://source.android.com/security/bulletin/2023-03-01)

ing additional execution privileges.

- permissions bypass.

rability could be exploited programmatically by a malicious entity without any active participation from the user.

exploit the vulnerability by executing a specific code sequence that triggers the out-of-bounds write in 'bta\_av\_rc

er overflow in avdt\_scb\_hdl\_write\_req of avdt\_scb\_act.cc, which is part of the Android operating system. This iss Android-13.

on.

, located at the following URL: <https://source.android.com/security/bulletin/2023-03-01>.

An attacker gaining higher privileges could perform actions that are normally restricted, potentially compromising the device. To avoid facilitating exploitation, the issue itself is due to a heap buffer overflow. Generally, CVE descriptions avoid including the heap buffer overflow vulnerability without requiring any permissions or user interaction. By doing so, the vulnerability is identified by the CVE ID CVE-2023-20929.

HalfSheetActivity.java` in Android.

Following URL: <https://source.android.com/security/bulletin/2023-03-01>.

With this intent to clandestinely discover Bluetooth MAC addresses of nearby devices. With this information, an attacker

[security/bulletin/2023-03-01](https://source.android.com/security/bulletin/2023-03-01).

After a factory reset. Without needing any additional execution privileges or user interaction, the attacker could bypass these permissions is CVE-2023-20917.

on Android 13.

bulletin website at <https://source.android.com/security/bulletin/2023-03-01>.

to a wrong file, which may result in local escalation of privilege on the device.

in ResolverActivity.java to share a file with the wrong recipient without the user's knowledge or consent. Since no user interaction is required. It involves a possible failure to persist permission settings due to resource exhaustion, potentially leading to local escalation of privilege (e.g., root access). This indicates that it is a significant security concern that should be addressed promptly.

Vulnerability without the need for any action from the user.

bulletin, and the specific URL for the bulletin addressing this vulnerability is <https://source.android.com/security/bulletin/2023-03-01>. The bulletin states that a malicious application or actor could gain elevated permissions beyond those initially granted, potentially abuse the sharing of exploit code for vulnerabilities to prevent misuse. However, developers and security professionals are urged to be vigilant. The vulnerability to gain unauthorized access to additional permissions without the user's consent or knowledge. This could lead to a local escalation of privilege on the device. Specifically, it describes a potential method for silently granting permissions to an application after it

initially allow silent granting of permissions. This issue could result in a local escalation of privilege for a malicious app. This vulnerability represents a significant threat to the affected systems.

An app running on an older SDK initially and then updates to a higher Target SDK. Upon the update, due to the permissions bypass vulnerability, the app can leverage the permissions bypass flaw to silently escalate privileges after updating an app to a higher Target SDK.

Android Security Bulletin: <https://source.android.com/security/bulletin/2023-03-01>

Due to a logic error in the code which could potentially lead to local information disclosure. An attacker would not

the specific version numbers were not provided in the original information.

<https://source.android.com/security/bulletin/pixel/2023-03-01>.

Due to the logic error to access sensitive data from the device's shared memory that it should not have access to. Since the

Android kernel. This vulnerability arises from improper input validation, which could potentially allow an attacker to e

Users are advised to check the Android security bulletin or updates from the device manufacturer for the precise affected devices. For more information, visit the following webpage, specifically at <https://source.android.com/security/bulletin/pixel/2023-03-01>. This vulnerability affects permissions on the device.

Users using this vulnerability to execute arbitrary shell commands without user interaction. This could result in unauthorized access to sensitive data (CVE-2022-42499).

AndroidManifest.xml of sms\_MmConManagement.c. This could lead to remote code execution without the need for an

affected devices are not explicitly mentioned in the provided information.

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

A message or packet to a vulnerable Android device. The message would trigger a heap buffer overflow in the sms\_

due to a missing bounds check. This flaw could potentially lead to remote code execution with no additional exploit, particularly concerning as it could be exploited remotely without the victim's knowledge.

vulnerability scoring system (CVSS).

[source.android.com/security/bulletin/pixel/2023-03-01](https://source.android.com/security/bulletin/pixel/2023-03-01).

kernel as used in Pixel devices.

Received network packets to the vulnerable Pixel device. Since no user interaction is required, this could happen while the device is in use. This is a serious issue due to the importance of prioritizing security. However, the vulnerability is related to a missing bounds check which exists in the Android operating system. The issue is a possible out-of-bounds (OOB) write caused by a missing bounds check in the Android operating system. Its severity is due to the possibility of local privilege escalation without needing user interaction or additional exploit. For more information, visit the Android Security Bulletin published online, more specifically at the URL: <https://source.android.com/security/bulletin/pixel/2023-03-01>. The Android Security Bulletin provided by Google at the following link: <https://source.android.com/security/bulletin/pixel/2023-03-01>. This vulnerability is more critical as it could be exploited quietly without alerting the user.

The exploit leverages the vulnerability within the parseParamsBlob of types.cpp to perform an out-of-bounds write of

tractor.cpp, where there is a possible out of bounds read due to an integer overflow. This issue could potentially lead to a denial of service.

<https://source.android.com/security/bulletin/pixel/2023-03-01>

open it on a vulnerable Android device. Due to the integer overflow in `parseTrackFragmentRun()` of `MPEG4Extractor.java`, it facilitates exploitation. Code examples provided by official sources, such as a security advisory or patch, are meant to help developers identify and fix the issue on Android where uncaught errors during the parsing of stored configs could lead to local persistent denial of service with high severity.

[source.android.com/security/bulletin/2023-03-01](https://source.android.com/security/bulletin/2023-03-01)

It is noted that when parsed by `PasspointConfiguration.java`, results in uncaught errors causing a persistent denial of service. It is not appropriate to share specific code examples that could facilitate the exploitation of the vulnerability. Generally, the issue occurs in the `isBluetoothShareUri` function of `BluetoothOppUtility.java`, where a confused deputy problem could lead to a local denial of service. The Common Vulnerability Scoring System (CVSS) score is 7.5, indicating that the impact of the vulnerability is significant but not critical for Android 13.

The exact nature of the interaction is not detailed. Generally, this could involve tricking the user into performing specific actions.

Original URL: <https://source.android.com/security/bulletin/2023-03-01>

The confused deputy problem in `BluetoothOppUtility.java` allows the application to read files that it should not have access to. For instance, the code in `BluetoothOppUtility.java`, which may not be publicly disclosed or easy to replicate without proprietary information from WordPress, specifically in versions up to and including 11.13. This type of vulnerability could allow an attacker with access to the application to read sensitive data. The severity is rated as moderate and should be addressed in a timely manner to prevent potential exploitation. The issue exists up to and including version 11.13.

Original URL: <https://patchstack.com/database/vulnerability/wpappninja/wordpress-wpmobile-app-android-and-ios-mobile-app-vulnerability>  
The vulnerability involves the execution of malicious scripts in the context of the web page from the victim's browser. This could lead to various attack scenarios, such as unauthorized access to the WordPress site using the vulnerable plugin. The attacker could craft a post with a malicious script execution for iOS & Android version 1.4.4. It allows an authenticated attacker to gain access to sensitive account information. The severity level is rated as HIGH according to the CVSS scoring system.

Version 1.4.4.

WithSecure Lab's advisory page at <https://labs.withsecure.com/advisories/username-disclosure-vulnerability-in-dl-megafeis-bofei-dbd-app> describes a vulnerability in the MEGAFEIS BOFEI DBD+ application that could be exploited to gain access to sensitive information such as usernames. The advisory provides specific code examples that demonstrate the exploitation of such vulnerabilities due to ethical concerns. The issue exists in iOS & Android version 1.4.4, which allows an attacker to unlock model(s) without proper authorization by sending a crafted request. The severity is rated as HIGH, indicating that it represents a significant security risk if exploited.

Advisory at <https://labs.withsecure.com/advisories/insecure-authorization-scheme-for-api-requests-in-dbd--mobile-cc> describes a vulnerability in the MEGAFEIS, BOFEI DBD+ app to unlock models. The attacker then crafts unauthorized API requests with a crafted payload to the 3D+ application. This breach of security could lead to unauthorized access and control over features or data that should be restricted. The issue exists on Android and is CVE-2022-45637.

The vulnerability exists for iOS & Android, which pertains to an insecure expiry mechanism used in the service's password reset process. The severity is rated as HIGH.

<https://github.com/WithSecureLabs/megafeis-palm/tree/main/CVE-2022-45637>

ry mechanism in the password reset process. The attacker could intercept or reuse password reset tokens due to password reset tokens have a strict expiration time and are invalidated immediately upon use or after a short, predefined time. This issue allows an attacker to gain access to sensitive account information due to a

ry vulnerability.

SecureLabs/megafeis-palm/tree/main/CVE-2022-45635.

password policy in the BOFEI DBD+ Application v1.4.4. By exploiting this weakness, the attacker might be able to bypass the password policy. The application should be updated to a version where this vulnerability has been patched. Developers of the application should im

ected by the CVE-2023-21464 vulnerability.

erability.

<https://security.samsungmobile.com/serviceWeb.smsb?year=2023&month=03>.

lar app due to improper access control, which could allow the attacker to configure an improper status. This might affect various versions of Android which affects devices prior to versions 12.2.09.0 in Android 11, 13.1.03.501 in Android 12, and 14.1.03.0 in Android 13. The severity is LOW in terms of severity according to the rating provided.

ough the following link: <https://security.samsungmobile.com/serviceWeb.smsb?year=2023&month=03>.

1 in Android 12, and 14.1.03.0 in Android 13.

physical access to the device exploiting the improper access control in the MyFiles application to extract sensitive information. There are typically no code examples for it. Security best practices would involve reviewing access control implementation on Android devices. Specifically, it affects versions of the software prior to 3.5.14.18 on Android 12, and versions prior to 14.1.03.0 in Android 13. The severity is LOW. This score indicates that the vulnerability poses a lesser risk compared to higher scored vulnerabilities.

<https://security.samsungmobile.com/serviceWeb.smsb?year=2023&month=03>.

device exploiting the vulnerability to retrieve the MAC address without the required permissions. This could lead to an information disclosure issue. It has been classified with a base score of 5.5, indicating that it is a medium severity vulnerability. It allows unauthorized access or exposure of sensitive information to an attacker under certain conditions.

g to the Common Vulnerability Scoring System (CVSS).

website at the following URL: <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-24923>

ation disclosure vulnerability like this one could allow an attacker to gain unauthorized access to sensitive data stored on the device. The severity is MEDIUM. This score indicates that the vulnerability poses a moderate risk compared to higher scored vulnerabilities.

id.

ulnerability.

ISRC) website at the following URL: <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-24882>

scenario could involve an attacker exploiting the information disclosure vulnerability in Microsoft OneDrive for Android app is updated to the latest version that includes the security fix for this vulnerability. Users can also be cautioned to update the app to the latest version.

considered MEDIUM severity.

SR) webpage: <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-23391>.

icious document and convincing the user to open it on their Android device running the vulnerable version of Office in version 1.3.0. This vulnerability allows an attacker to capture the screen if they install a malicious application or an authorized attacker to capture the screen through a malicious app. This could lead to the extraction of sensitive data and the vulnerability.

aunchpad.support.sap.com/#/notes/3302710 and in SAP's official documents at <https://www.sap.com/document>. To exploit this vulnerability, an attacker would install a malicious application on their Android device that has the SAP Authenticator installed. This malicious app would then be viewed by the user on SAP Authenticator and any secret OTP alphanumeric tokens that are displayed on the screen would be captured. CVE-2023-40539.

Android OS due to improper validation of an array index. This could mean, for example, that a function within the OS incorrectly handles an array index, leading to a buffer overflow or other memory corruption. This vulnerability poses a significant risk and can result in considerable damage or impact to the security of the system.

Specifically at the following URL: <https://www.qualcomm.com/company/product-security/bulletins/march-2023-bulletin-003>. The bulletin describes a vulnerability in the Qualcomm Snapdragon processor that takes advantage of the improper array index validation. This could lead to out-of-bounds memory access and potentially allow an attacker to execute arbitrary code.

on Android. It allows a remote attacker to perform domain spoofing through a crafted HTML page.

Qualcomm security team, with a base score of 4.3, categorized as MEDIUM.

Issue tracker: <https://crbug.com/1404621> - The official Google Chrome Releases blog entry detailing the stable channel update for desktop. The entry describes a vulnerability in Google Chrome on Android that could potentially allow a remote attacker to spoof the contents of the omnibox via a crafted HTML page. CVE-2023-1231.

on Android that could potentially allow a remote attacker to spoof the contents of the omnibox via a crafted HTML page. The vulnerability is rated as MEDIUM with a base score of 4.3.

Google Releases: <https://googleblog.com/2023/03/stable-channel-update-for-desktop.html> - <https://crbug.com/1274887> - CVE-2023-1231.

parts of the omnibox in Google Chrome on Android, which could be used to deceive users into thinking they are visiting a legitimate website. Users type in URLs or search queries, and the omnibox offers search suggestions, autocompletes queries, and provides suggestions in the Autofill feature of Google Chrome on Android. When a user visits this crafted page, the attacker could potentially capture sensitive information. This vulnerability is related to an inappropriate implementation in WebApp Installs that allowed an attacker who convinced a user to install a malicious app to perform domain spoofing.

the Common Vulnerability Scoring System (CVSS).

CVE-2023-1404230- <https://chromereleases.googleblog.com/2023/03/stable-channel-update-for-desktop.html>

with a crafted HTML page designed to mimic the appearance of a legitimate PWA installer prompt. The attacker tricked users into installing a malicious app. The vulnerability is related to insufficient policy enforcement in Intents, which could allow a remote attacker to bypass navigation restrictions imposed by Google Chrome. The vulnerability is rated as MEDIUM with a base score of 4.3. It is related to insufficient policy enforcement in Intents, which could allow a remote attacker to bypass navigation restrictions imposed by Google Chrome.

y a user, abuses insufficient policy enforcement in Intents to escape navigation restrictions imposed by Google Chrome.

Google Releases blog post: <https://chromereleases.googleblog.com/2023/03/stable-channel-update-for-desktop.html>. Users are advised to update their browser to this version or later to be protected against the vulnerability.

The vulnerability is related to insufficient policy enforcement in Intents, which could allow a remote attacker to leak cross-origin data using a specially crafted HTML page designed by an attacker. This page could exploit the vulnerability by using the Intent system used to request an action from another app component. The vulnerability arises from insufficient policy enforcement in Intents, where insufficient policy enforcement allowed a remote attacker to leak cross-origin data using a specially crafted HTML page.

sion 111.0.5563.64 or later to fix the vulnerability.

Chrome Release Blog (<https://chromereleases.googleblog.com/2023/03/stable-channel-update-for-desktop.html>) that exploits the vulnerability in the Autofill feature. When a user navigates to this page using a vulnerable version of Chrome, it triggers the exploit. Exploiting this vulnerability could lead to various security issues, such as unauthorized access to sensitive information or the execution of arbitrary code on the user's device. Exploiting vulnerabilities such as CVE-2023-1223. Doing so would be unethical and could potentially aid attackers. However, generating a proof of concept (PoC) for a vulnerability is a common practice in security research to demonstrate the existence and impact of the vulnerability. It is important to use this information responsibly and only for legitimate security research purposes.

to version 1.3.0 on Android, specifically affecting unknown code within the file `src/at/hgz/vocabtrainer/Vocab` could potentially access directories and files stored outside of the web server's root directory.

tion to version 1.3.1, which contains a patch fixing the issue. The name of the patch is accf6838078f8eb105cfc786!865aba5c705fb68426 and is available in the version 1.3.1 release of the hgzojer Vocab Trainer application.

evice. This means that the attacker needs to execute the attack from within the device itself, rather than remotely  
ths to access files outside of the intended directories. For example, the attacker might use special characters like

it hash accf6838078f8eb105cfc7865aba5c705fb68426) or in the release notes for version 1.3.1 of the hgzojer Vo  
20948.

o a heap buffer overflow, which could lead to remote information disclosure without the need for additional execution

[://source.android.com/security/bulletin/2023-02-01.](https://source.android.com/security/bulletin/2023-02-01)

rafting a video file with carefully structured frames that, when processed by the vulnerable component in Android, can cause the device to crash or execute arbitrary code. This vulnerability is not fixed in the latest security patches provided by the manufacturer or service provider. For affected versions (Android 12 and earlier), users are advised to update their devices to the latest available version of the operating system. Specifically, there is a confused deputy problem that could allow for a permission bypass. A high score indicates that the vulnerability presents a significant threat and should be remediated as soon as possible. Devices are potentially vulnerable to attacks exploiting this security issue.

critical for an attacker to exploit the vulnerability remotely and without detection.

L: <https://source.android.com/security/bulletin/2023-02-01>.

erability to remotely manipulate Bluetooth settings on an affected device. Since the vulnerability could allow escape from the device's secure environment, the vendor was notified of the vulnerability and made publicly available for awareness and remediation.

tns\_MifareStd.cpp, where there is an out of bounds write possibility due to a missing bounds check. This could potentially lead to a denial of service or a local escalation of privilege.

attacker could gain higher privileges on the system.

verity.

[https://source.android.com/security/bulletin/2023-02-01.](https://source.android.com/security/bulletin/2023-02-01)

l on an Android device. Without requiring any user interaction, this application could exploit the out of bounds write vulnerability in the `write` method of the `FileWriter` class. This vulnerability is CVE-2023-20944.

1 in ChooseTypeAndAccountActivity.java within the Android operating system.

Android 12, Android 12L, and Android 13.

is a significant risk and could potentially lead to a serious impact if exploited.

ileges, and no user interaction is required for exploitation.

etain page at the following URL: <https://source.android.com/security/bulletin/2023-02-01>.

exploit this vulnerability to gain elevated privileges on the device, potentially leading to unauthorized access and control on that targets the vulnerability in the ChooseTypeAndAccountActivity.java file. Once installed on a user's device,

dated to the latest version as provided by the device manufacturer or carrier. This will typically include patches for Android-12L, and Android-13.

Android-12L, and Android-13.

ChooseTypeAndAccountActivity.java on the Android platform. Due to a path traversal error, there exists a possible way to remove or manipulate

ing link: <https://source.android.com/security/bulletin/2023-02-01>.

device could exploit the path traversal vulnerability in ActivityManagerService.java. By crafting a malicious request to prevent path traversal attacks, implement proper permission checks, and follow secure coding best practices. A way to replace the boot partition due to improperly used cryptography. This can lead to local escalation of privilege

website: <https://source.android.com/security/bulletin/2023-02-01>.

Android device or a malicious application that manages to run code on the device without the need for user interaction. This version includes the security patch that addresses this vulnerability. Manufacturers often release updates to address this vulnerability in looped\_event\_loop.cpp. This vulnerability is due to improper locking in multiple functions, which could potentially lead to a local escalation of privilege in severity. This indicates that the vulnerability represents a significant risk that could compromise the security of the device. This can be exploited without any action from the user of the affected Android device.

these versions are susceptible to being exploited through this vulnerability.

ing mechanisms. This corruption can lead to a local escalation of privilege, which means the attacker could gain higher

urity Bulletin at <https://source.android.com/security/bulletin/2023-02-01>. The bulletin offers detailed information on the vulnerability in looped\_event\_loop.cpp. The app could employ improper locking to manipulate memory and control flow within the binder.c file of the Android kernel. It involves a possible use after free due to improper input validation. This vulnerability could have significant impact if exploited.

ge the vulnerability without user participation.

is not detailed in the provided information. Typically, the Android Security Bulletin would provide more detail on the vulnerability. ing URL: <https://source.android.com/security/bulletin/2023-02-01>

free vulnerability typically occurs when memory is accessed after it has been freed, possibly leading to a program crash or a local escalation of privilege on the affected system. This means they could gain higher level permissions than they should have, which might allow

where there is a memory corruption issue due to a use-after-free error. This vulnerability could allow a local attacker to exploit the vulnerability represents a significant threat and should be addressed promptly.

exploited by an attacker without tricking the user into performing any actions.

the Android kernel code. However, the exact version numbers are not provided in the information provided.



an attacker with basic user-level access could potentially exploit the vulnerability. An attacker could leverage this to gain higher-level permissions and potentially take control of the system, access sensitive data, and more. For more information, see the Android Security Bulletin at <https://source.android.com/security/bulletin/2023-02-01> and on Packet Storm Security at <http://packetstormsecurity.com>. Details of the vulnerability were not provided in the information shared. Use-after-free vulnerabilities are typically discovered on the device, executing a malicious application or script that triggers the use-after-free error in the kernel, leading to a crash or other issues. This vulnerability could potentially lead to local escalation of privilege on affected Android versions.

Local escalation of privilege without requiring additional execution privileges or user interaction, meaning that an attacker could potentially exploit the vulnerability without requiring additional execution privileges or user interaction.

For more information, see the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-02-01>.

The severity of this vulnerability is high as it could potentially be exploited silently.

An attacker could disable the microphone privacy indicator. This app could record audio from the device without the user's knowledge. This vulnerability could potentially be exploited by malicious applications, posing a risk to user privacy and security. However, Android's source code, patches, and related commit messages may contain fixes to address this issue.

This vulnerability could potentially lead to a local escalation of privilege on affected Android versions, including Android-12L and Android-13.

For more information, see the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-02-01>.

This vulnerability is located in `MediaCodec.cpp`. Since no user interaction is required, an attacker could program a malicious app or script that triggers the vulnerability. The responsible disclosure usually involves reporting the vulnerability details to the vendor.

This vulnerability affects Android-12L and Android-13.

This vulnerability could potentially lead to a local escalation of privilege on affected Android versions, allowing for reading contacts belonging to other users.

32.

For more information, see the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-02-01>.

An attacker could program a malicious app or script that bypasses normal input validation checks and accesses contacts data of other users on the device. Due to a logic error, there is a potential method for recording audio without displaying a privacy indicator, posing a risk to user privacy and security.

For more information, see the Android Security Bulletin page: <https://source.android.com/security/bulletin/2023-02-01>.

This vulnerability could potentially lead to a local escalation of privilege on affected Android versions, allowing for reading contacts belonging to other users.

This vulnerability could potentially lead to a local escalation of privilege, compromising the privacy and security of the system execution privileges. Due to the vulnerability, the app might be able to start recording audio without alerting the user. This vulnerability was disclosed in their Security Bulletin. Users should ensure their devices are updated with the latest security patches to mitigate this risk. This vulnerability could potentially lead to a local escalation of privilege on affected Android versions, allowing for reading contacts belonging to other users. This flaw could lead to local information disclosure.

er a device reset in order to access preserved WiFi settings, potentially leading to information disclosure. Exploita

lowing URL: <https://source.android.com/security/bulletin/2023-02-01>.

evice can potentially access sensitive WiFi credentials and network details even after the device has been reset, w  
instead, the issue would likely pertain to how the Android OS handles data sanitation during the reset process, whi  
usly reset Android device. The attacker could exploit the vulnerability to retrieve WiFi settings that were not corr

to resource exhaustion on Android devices.

2L, and Android 13.

vel.

ps://source.android.com/security/bulletin/2023-02-01.

155; it can be exploited by a local attacker.

ibility in ZenModeHelper.java to create a persistent denial of service condition on the device by causing resource  
/stem, a specific code exploit example is not provided. However, the vulnerability pertains to the addAutomaticZe  
VE ID CVE-2022-32906.

oid.

nnctions, thereby compromising the confidentiality and integrity of the information being transmitted over the i  
e at the following URL: <https://support.apple.com/en-us/HT213473>.

he attacker intercepts and potentially alters the communication between two parties without their knowledge.  
protocol that encrypts information sent over the network, in Apple Music 3.9.10 for Android. This helps prevent i

3.9.10 for Android.

to bypass application state checks and gain unauthorized access to sensitive user data, such as personal informati  
[port.apple.com/en-us/HT213473](https://support.apple.com/en-us/HT213473)

9.10 for Android. The vulnerability was related to suboptimal state management in the application that could pot  
/. This indicates that the vulnerability presents a significant risk and should be addressed promptly by users and a

e Apple Music application. Users should ensure they have updated to version 3.9.10 for Android or later to mitiga  
t the following link: <https://support.apple.com/en-us/HT213473>

lnerability to gain unauthorized access to user-sensitive data stored by the Apple Music application on Android de  
ossible. Regularly checking for updates and installing them protects against known vulnerabilities, including CVE-2

user's activity.

page: <https://support.apple.com/en-us/HT213472>

network traffic, such as on an unsecured Wi-Fi network. Before the fix, when the Apple Music app on Android ser  
TPS when sending information over the network, which encrypts the data in transit and prevents an attacker from



3-20927.

ion of privilege within Android applications. It occurs due to improper handling of signature permissions in the Ar

proper authorization. This exploitation does not require any additional execution privileges or user interaction.

Other privileges within the Android operating system that are typically protected by signature-level permissions.

<https://source.android.com/security/bulletin/aaos/2023-02-01>.

evice, which exploits this vulnerability to bypass certain security checks that are intended to restrict access to sen  
n incomplete fix for a path traversal issue. The vulnerability could lead to information disclosure and arbitrary file  
users to access, share, and edit files and folders on the ownCloud platform.

n disclosure when uploading the app's internal files, and to arbitrary file write when uploading plain text files, the

a fix for the reported path traversal bypasses.

bility.

vulnerability by crafting a malicious request to access or write to sensitive files within the app's internal storage. Fo  
by GitHub's Security Lab at '[https://securitylab.github.com/advisories/GHSL-2022-059\\_GHSL-2022-060\\_Owncloud](https://securitylab.github.com/advisories/GHSL-2022-059_GHSL-2022-060_Owncloud)  
ion 2.21.1. It involves an SQL injection vulnerability in the `FileContentProvider.kt` file. This vulnerability can resu

3-23948, has been deprecated. However, the SQL injection vulnerability remains a concern for the `owncloud\_dat  
ch might allow an attacker to access sensitive data from the affected databases.

nCloud Android app's vulnerable component (`FileContentProvider.kt`), could manipulate SQL queries. This could

ab, accessible via the following link: [https://securitylab.github.com/advisories/GHSL-2022-059\\_GHSL-2022-060\\_](https://securitylab.github.com/advisories/GHSL-2022-059_GHSL-2022-060_) (

3.1.0 improperly verify server certificates, potentially allowing a man-in-the-middle attack for eavesdropping on  
ndicates that the vulnerability has a moderate impact on the confidentiality, integrity, or availability of the affecte

Google Play Store page for the app, the Apple App Store page, and the Japan Vulnerability Notes database (JVN)  
Idle (MITM) attack. In such a scenario, the attacker could position themselves between the user's device and the s  
or to 3.1.0. Users of these versions are advised to update to version 3.1.0 or later to mitigate the vulnerability.

puts sensitive information to the log file. This could potentially allow an attacker to obtain credential information I  
:d location), Thailand SUSHIRO Ver.1.0.0, Hong Kong SUSHIRO Ver.3.0.2, Singapore SUSHIRO Ver.2.0.0, and Taiwa

ded for each regional version of the SUSHIRO app, and on the Japan Vulnerability Notes (JVN) website with the JV  
ng the log files of an affected SUSHIRO Android app. Since the app outputs sensitive information such as credentia  
thin the SUSHIRO app's code, there might be instances where sensitive information is written to the log indiscrimi  
hat addresses CVE-2023-22362. It's important to regularly check for software updates and install them to mitigate  
to log files. They can implement proper logging practices that filter out sensitive data, use robust logging librarie  
Android 12 (S) version. Specifically, it is a Stack-based buffer overflow vulnerability located in the `LpcRxEmbsSes`  
ren data exceeds the allocated buffer size, in this case, the stack, potentially resulting in execution of malicious co  
the Common Vulnerability Scoring System (CVSS).

ally at the following URL: <https://security.samsungmobile.com/serviceWeb.smsb?year=2022&month=04>





PackageManagerService.java is CVE-2023-20922.

va on Android platforms. It can cause a crash loop due to resource exhaustion, resulting in local denial of service. id Android-13.

, specifically at the URL: <https://source.android.com/security/bulletin/2023-01-01>.

'E-2023-20922; such information is not typically included in the CVE description or the Android Security Bulletin.

re vulnerability in the setMimeTypeGroup function of PackageManagerService.java to create a crash loop on the affect

lroid 12L, and Android 13.

e.android.com/security/bulletin/2023-01-01.

of the Android OS.

performing certain actions that leverage the logic error in the code of AccessibilityManagerService.java. This could

and Android-13.

UsbRequest.java, which could lead to local escalation of privilege.

om/security/bulletin/2023-01-01.

ree condition in the queue of UsbRequest.java without requiring any additional permissions or user interaction. C ed by the Android Security Bulletin or device manufacturers as soon as they become available. Developers should

.java in Android-13. It involves a logical error that could potentially prevent package uninstallation, leading to loc t an attacker could gain elevated access to the system without requiring user interaction or additional execution p

n be exploited silently.

he following URL: <https://source.android.com/security/bulletin/2023-01-01>.

re logic error in the getStringsForPrefix method of Settings.java. The attacker could manipulate the settings to pre

s on starting activities from the background due to a missing permission check. This could lead to a local escalation

ch can be found at '<https://source.android.com/security/bulletin/2023-01-01>'.

nteraction to execute. The application would attempt to start an activity from the background without the necess

tRegistrar.java within Android. It is a logic error that could potentially allow an attacker to enable a phone account

Android 12L, and Android 13.

Original URL: <https://source.android.com/security/bulletin/2023-01-01>.

to gain elevated privileges on the device.

an application that exploits the logic error in the `addOrReplacePhoneAccount` function to enable a phone account without the need for user interaction.

along with related files in the Android operating system.

leading to enabling a malicious phone account.

Android 12L, and Android 13.

indicating that some form of user interaction is necessary for the attack to be successful.

See the following URL: <https://source.android.com/security/bulletin/2023-01-01>.

an application on the affected Android device, which could result in unauthorized actions or access.

The application employs a tapjacking/overlay attack. This application overlays a transparent window over legitimate UI elements, allowing an adversary to interact with the application without the user's knowledge.

The application accesses `Image` objects in the Android operating system. Specifically, it allows for potential access to images that belong to other applications.

Original link: <https://source.android.com/security/bulletin/2023-01-01>

The application calls the `onActivityResult` method of `AvatarPickerActivity.java` without the proper permissions. The vulnerabilities might allow the application to bypass the permission check, allowing an adversary to potentially access images belonging to other users and leading to local file access, which typically include patches to fix the missing permission check vulnerability in the `onActivityResult` method of `AvatarPickerActivity.java`.

Android 12L, and Android 13.

may lead to resource exhaustion.

The application may enter a crash loop due to resource exhaustion.

in/2023-01-01.

classified as MEDIUM severity.

The application triggers the vulnerable code in `SettingsState.java`. Since no additional execution privileges or user interaction is required, an adversary could exploit this vulnerability to access the `is_MifareStd.cpp` file. It involves a possible out of bounds write due to a missing bounds check which could lead to a crash on the affected systems should the vulnerability be exploited.

Because it could be exploited without the user's knowledge.

It is recommended that device manufacturers release security patches to address this security issue.

See this URL: <https://source.android.com/security/bulletin/2023-01-01>.

The application invokes the vulnerable function in `phNxpExtns_MifareStd.cpp` without proper bounds checking. This could allow the application to write to an out-of-bounds memory location, leading to a buffer overflow or memory corruption. This can corrupt data, crash the program, or open up opportunities for an attacker to exploit this vulnerability. Device manufacturers will typically release firmware updates that include security patches for this vulnerability.



roid platform, where there is a possible exploit allowing the launch of an arbitrary activity because of an Intent m

ecution privileges, and user interaction is not necessary for exploitation.

source.android.com/security/bulletin/2023-01-01.

ge by launching an arbitrary activity due to an Intent mismatch in Android's SettingsActivity.java code. Exploitation  
responsible and potentially assist in malicious activities. However, the vulnerability generally involves modifying  
ssible persistent denial of service (DoS) due to resource exhaustion that could lead to a local denial of service with  
roid 12L, and Android 13.

on Vulnerability Scoring System).

lowing link: <https://source.android.com/security/bulletin/2023-01-01>.

rc exhaustion, leading to a persistent local denial of service. This could occur without the need for any user inter  
s for CVE-2022-20494 are not provided publicly. The actual exploitation details and code would depend on the spe  
rting on an Android device and using this vulnerability to cause a denial of service. This could lead to the affected  
93.

calation of privilege due to improper input validation. This could potentially enable unauthorized parties to grant  
roid 12L, and Android 13.

at the URL: <https://source.android.com/security/bulletin/2023-01-01>.

ge user into performing specific actions that validate improper inputs. This can compromise the device's security by

roid-13.

source.android.com/security/bulletin/2023-01-01'.

em resources, preventing Android's AutomaticZenRule from persisting permission settings correctly. Once the se

Android. It involves a possible failure to persist permissions settings due to resource exhaustion, potentially lea

ng URL: <https://source.android.com/security/bulletin/2023-01-01>.

Android device without needing any additional execution privileges or user interaction, potentially gaining unautl  
bypass Android's permissions checks due to the failed persistence of permissions settings. As a result, the attack  
erally not provided to the public to prevent misuse. Instead, the Android Open Source Project (AOSP) will typically

oid-12L, and Android-13.

tings due to resource exhaustion in several functions of AutomaticZenRule.java. This could lead to local escalator

[android.com/security/bulletin/2023-01-01](https://source.android.com/security/bulletin/2023-01-01).

crafted to exhaust system resources intentionally. When resources are depleted, the application could exploit the Android devices. Specifically, it is found in the `pinReplyNative` function of the `com_android_bluetooth_btservice_A`roid 11, Android 12, Android 12L, and Android 13.

ing URL: <https://source.android.com/security/bulletin/2023-01-01>.

to perform an out of bounds read in the Bluetooth stack. This could be used for local privilege escalation without Energy (BLE) operations. This means that an attacker could potentially execute undesired actions or access privilege

may contain sensitive information such as personally identifiable information (PII) or hardware identifiers, is being

[rce.android.com/security/bulletin/aaos/2023-01-01](https://source.android.com/security/bulletin/aaos/2023-01-01).

l logs of an affected 'user' build Android device. By examining these logs, the attacker could extract sensitive information. This involves a code snippet from `CarNotificationListener.java` where the `StatusBarNotification.getKey()` is logged. An example is provided in the [bulletin](https://source.android.com/security/bulletin/aaos/2023-01-01).  
roid-13.

tion, which could lead to local escalation of privilege without the need for additional execution privileges.

[roid.com/security/bulletin/2023-01-01](https://source.android.com/security/bulletin/2023-01-01).

stion issue to change permissions settings, thereby gaining elevated privileges without the user's knowledge. Such

ge' used by the cache subsystem was writable by a user-space program prior to DDK version 1.18. This could lead to a situation where the vulnerability has a moderate level of impact on the confidentiality, integrity, or availability of the affected

the PowerVR GPU kernel driver prior to DDK version 1.18.

device could exploit this vulnerability by writing arbitrary data to the Information Page. This could lead to memory corruption. For more details, see <https://source.android.com/security/bulletin/2023-01-01>.

kernel driver. Users are advised to update their SoC's GPU drivers to this version or later to mitigate the associated issue. The issue arises from the `onCreate` method of `MasterClearConfirmFragment.java`, which could be exploited through a buffer overflow. For more details, see <https://source.android.com/security/bulletin/2023-01-01> for its impact and how it can be exploited.

to a user to inadvertently tap on a malicious overlay.

it <https://source.android.com/security/bulletin/aaos/2023-01-01>.

in top of legitimate UI elements. When the user interacts with what they perceive to be harmless UI components, the application can create an overlay on the UI, which can be achieved with regular app permissions. No special system or kernel permissions are required. The toggle button in 'Modify system settings' is susceptible to a tapjacking attack. This flaw allows attackers to overlay their own UI on top of the system settings, giving permissions or performing actions that benefit the attacker. This is done by overlaying a transparent or dec

[ce.android.com/security/bulletin/aaos/2023-01-01](https://source.android.com/security/bulletin/aaos/2023-01-01).

that displays an overlay with a convincing interface, like a game or a survey, on top of the 'Modify system settings' dialog to cover the 'Modify system settings' toggle button. When a user interacts with the seemingly benign overlay, it manifests a malicious intent. This is a type of attack that can cause a possible denial of service (DoS) due to tapjacking/overlay. This is a user interface aspect of applications by allowing potential attackers to disrupt the normal behavior of an application,

This means that the vulnerability presents a moderate level of risk and should be taken seriously but is not as critical as a Denial of Service (DoS) vulnerability by creating a malicious overlay on top of legitimate application screens. By doing so, an attacker might be able to disrupt the normal behavior of an application. JURL: <https://source.android.com/security/bulletin/aaos/2023-01-01>

is a misconfiguration in the manifest file. This flaw allows attackers to perform task hijacking attacks, potentially leading to a denial of service (DoS) or a data breach. The vulnerability can be engineered to hijack tasks from a legitimate application, allowing the attacker to intercept and steal sensitive information sent to the provided information.

Cloudflare, which is available at the following URL: <https://github.com/cloudflare/advisories/security/advisories/GHSA-4w3t-4p3p-4p3p>. This is a vulnerability in the WARP client's manifest file for Android. Once a victim installs this malicious application, it can facilitate malicious activity. However, developers can learn about the nature of task hijacking vulnerabilities and vulnerabilities in the WARP client's manifest file for Android. CVE-2023-0136.

Google Chrome on Android versions before 109.0.5414.74, which allowed a remote attacker to execute incorrect security checks. This indicates that the vulnerability poses a significant risk and should be addressed promptly.

including the Google Chrome release blog and security advisories from Gentoo. The references are:- <https://cve.circl.lu/vulnerability/CVE-2023-0136> and <https://cve.circl.lu/vulnerability/CVE-2023-0136>. This is a vulnerability in the WARP client's manifest file for Android. Once a victim installs this malicious application, it can facilitate malicious activity. However, developers can learn about the nature of task hijacking vulnerabilities and vulnerabilities in the WARP client's manifest file for Android. CVE-2023-0136.

where there was an inappropriate implementation in permission prompts. This issue allowed a remote attacker to execute arbitrary code on the device. The vulnerability is categorized as a path traversal vulnerability, which means it could potentially lead to a data breach. The vulnerability is categorized as a path traversal vulnerability, which means it could potentially lead to a data breach. The vulnerability is categorized as a path traversal vulnerability, which means it could potentially lead to a data breach.

on issue tracker: <https://crbug.com/1375132>- The official Google Chrome Releases blog post: <https://chromereleases.googleblog.com/2023/01/stable-channel-update-for-desktop.html>. This is a vulnerability in the WARP client's manifest file for Android. Once a victim installs this malicious application, it can facilitate malicious activity. However, developers can learn about the nature of task hijacking vulnerabilities and vulnerabilities in the WARP client's manifest file for Android. CVE-2023-0136.

it, employs the Fullscreen API's inappropriate implementation to spoof the contents of the Omnibox (URL bar) maliciously. This is a vulnerability in the WARP client's manifest file for Android. Once a victim installs this malicious application, it can facilitate malicious activity. However, developers can learn about the nature of task hijacking vulnerabilities and vulnerabilities in the WARP client's manifest file for Android. CVE-2023-0136.

to version 109.0.5414.74 or later, which contains the necessary fixes for the vulnerability. This is a vulnerability in the WARP client's manifest file for Android. Once a victim installs this malicious application, it can facilitate malicious activity. However, developers can learn about the nature of task hijacking vulnerabilities and vulnerabilities in the WARP client's manifest file for Android. CVE-2023-0136.

in versions before 5.13.0. This vulnerability is categorized as a path traversal vulnerability, which means it could potentially lead to a data breach. The vulnerability is categorized as a path traversal vulnerability, which means it could potentially lead to a data breach. The vulnerability is categorized as a path traversal vulnerability, which means it could potentially lead to a data breach.

contains dot sequences like './' which can lead the application to access files or directories that are outside the intended scope. This is a vulnerability in the WARP client's manifest file for Android. Once a victim installs this malicious application, it can facilitate malicious activity. However, developers can learn about the nature of task hijacking vulnerabilities and vulnerabilities in the WARP client's manifest file for Android. CVE-2023-0136.

These versions are vulnerable to this security issue.

e Zoom Android client. For protection against this security issue, users should ensure that they have updated their mobile website. You can refer to the provided reference: '<https://explore.zoom.us/en/trust/security/security-bulletin/>', as the vulnerable Zoom Android client could exploit this path traversal vulnerability to access sensitive information potentially malicious. Discussions around exploitation code for specific vulnerabilities are generally not done in public.

device to bypass the passcode and gain access to the user's Nextcloud files and view conversations within the Ne

22473.

Update the Nextcloud Talk Android app to version 15.0.2.

job pull request at <https://github.com/nextcloud/talk-android/pull/2598>, the HackerOne report at <https://hackerone.com/nextcloud> in which the vulnerable Nextcloud Talk app is installed.

s to a user's Android device that has the compromised version of the Nextcloud Talk app installed. Because of the by are typically not provided to prevent misuse. The fix for the vulnerability can be reviewed in the GitHub pull memory corruption resulting from improper input validation. If exploited, this vulnerability could allow an attacker t suggests that the vulnerability poses a significant risk and should be addressed promptly to mitigate potential ex

their product security bulletins. Specifically, details were published in the January 2023 bulletin at the following URL: [https://www.qualcomm.com/security/bulletins/2023-01-04-01](#). In this exploit, an attacker sends specially crafted input to the Automotive Android OS. This malicious input could lead to memory corruption, which could lead to a denial of service or, in some cases, an exploit would generally involve crafting input that takes advantage of the improper input validation to crash the device. System administrators should review the Qualcomm product security bulletins as soon as possible. System administrators should review the Qualcomm product security bulletin for details on this vulnerability. The CVE identifier for this vulnerability is CVE-2022-33274.

at: <https://www.qualcomm.com/company/product-security/bulletins/january-2023-bulletin>.

tion of an array index. This could potentially lead to unauthorized actions such as information disclosure, denial of service, or escalation of privileges on the affected device. The attacker might craft a malicious input or use a specific sequence of inputs to trigger the vulnerability. This issue allows for information disclosure due to insecure hostname validation, which can lead to unauthorized access to sensitive data. This reflects a significant risk associated with the exploitable nature of the issue, which can result in an attacker gaining unauthorized access to sensitive information.

alee0606/how-i-found-my-first-one-click-account-takeover-via-deeplink-in-ryde-5406010c36d8.

exploits the insecure hostname validation in the RYDE application. The attacker could then distribute this link via platforms that perform no validation checks in their applications. This may include using secure methods to verify the authenticity of URLs, employing

og on the Opera Mini application version 47.1.2249.129326 for Android by directing a user to a crafted website de which is labeled as 'MEDIUM'. This score reflects a moderate impact that the vulnerability poses to the affected s

isory/opera-mini-location-permission-spoof- This link leads to an advisory page that provides detailed information about the vulnerability. The advisory page is hosted on a website that looks like a legitimate service requiring location access. When the Opera Mini user navigates to this v

id, which allows an attacker to perform an IDN homograph attack. IDN homograph attacks exploit similarities in c  
ulnerability.

secure security advisory page, which can be found at <https://www.f-secure.com/en/home/support/security-advisories> on 19.2 for Android.

acters that appear visually similar to characters in the intended domain name, but are actually different character

domain name that visually appears identical to a legitimate domain due to the use of similar-looking characters for Firefox for Android is CVE-2022-45413.

'browser\_fallback\_url' parameter to redirect a user to a URL that could cause SameSite=Strict cookies to be sent. This vulnerability affects Firefox for Android version 107 and above and are not affected by this vulnerability.

Firefox for Android, and other operating systems are not affected.

page at <https://www.mozilla.org/security/advisories/mfsa2022-47/> and on the Mozilla Bugzilla page for the specific vulnerability or a link that uses the 'S.browser\_fallback\_url' parameter to redirect the user to a malicious site. This site might be expected name during startup could lead to a stack-buffer overflow. This overflow has the potential to cause a crash.

scoring system for vulnerabilities.

are not affected by this Firefox vulnerability.

<https://www.mozilla.org/security/advisories/mfsa2022-40/>) and the associated Bugzilla report ([https://bugzilla.mozilla.org/show\\_bug.cgi?id=171951](https://bugzilla.mozilla.org/show_bug.cgi?id=171951)) with a name that is not anticipated by the browser's startup process. When Firefox for Android attempts to load the application, it might trigger a buffer overflow attack. The responsible disclosure process usually involves reporting the vulnerability details without disclosing the details of the vulnerability to the public website, after being granted permission to use the microphone, can record audio without displaying the ongoing audio recording.

ability Scoring System (CVSS).

Firefox for Android versions prior to version 104 are not impacted by this vulnerability.

to version 104 or later, which contains the necessary patches to mitigate this issue.

microphone access from a user could start recording audio without triggering the audio notification icon. As a result, the user might be unaware of the recording. These sources include Mozilla's security advisory (mfsa2022-33) and the associated Bugzilla entry (bug 171951) and resources regarding the vulnerability.

visiting a website with an overly long URL, which could lead to a permanent Denial of Service due to session restoration.

moderate impact of the vulnerability that could potentially lead to a denial of service.

[www.mozilla.org/show\\_bug.cgi?id=1759951](https://www.mozilla.org/show_bug.cgi?id=1759951) - <https://www.mozilla.org/security/advisories/mfsa2022-28/>

cessively long URL and luring a Firefox for Android user to visit it. The long URL would cause the Firefox user interface to crash.

occurs on a domain protected by the HSTS header, the browser improperly allows the user to bypass the certificate pinning.

[www.mozilla.org/security/advisories/mfsa2022-24/](https://www.mozilla.org/security/advisories/mfsa2022-24/) and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1721220](https://bugzilla.mozilla.org/show_bug.cgi?id=1721220). This is a critical vulnerability.

a website and presenting a forged or invalid certificate. Because Firefox for Android allowed users to bypass the certificate pinning.

at HTTP Strict Transport Security (HSTS) settings when the browser was closed or sent to the background. This failure is categorized as MEDIUM severity. This suggests that the vulnerability presents a significant risk, but is not at the highest level.

0.

[www.mozilla.org/security/advisories/mfsa2022-16/](https://www.mozilla.org/security/advisories/mfsa2022-16/) and in the associated Bugzilla entry at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=171951](https://bugzilla.mozilla.org/show_bug.cgi?id=171951).

t a typical software development issue where code examples would be applicable or illustrative, providing a code snippets persistence by intercepting and manipulating unencrypted HTTP traffic. For instance, if a user accesses a site

for Android versions prior to 97.3.0, Thunderbird versions prior to 91.6.2, and Focus versions prior to 97.3.0.

in exploitable sandbox escape.

ough the WebGPU IPC framework that triggers a use-after-free condition, allowing an attacker to potentially execute

g/show\_bug.cgi?id=1758070- <https://www.mozilla.org/security/advisories/mfsa2022-09/>

with versions prior to 91.6.1 being vulnerable to this flaw.

ible versions including Firefox for Android earlier than 97.3.0, and Focus earlier than 97.3.0.

icates that the vulnerability is significant and can pose serious threats if exploited.

that can lead to arbitrary code execution or application crashes when exploited.

rces:- Mozilla's Security Advisories page at <https://www.mozilla.org/security/advisories/mfsa2022-09/>- The Mozilla to ethical considerations and the potential for misuse. To understand how to address the vulnerability, one should use-after-free vulnerability by crafting a malicious web page with specific XSLT parameters. When this page is loaded

er website, potentially deceiving the user into thinking the message comes from the site they are viewing.

at <https://www.mozilla.org/security/advisories/mfsa2022-04/> and the corresponding Bugzilla entry at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1758070](https://bugzilla.mozilla.org/show_bug.cgi?id=1758070)

rs a JavaScript alert or prompt overlaying a legitimate website in the background. The user might then be misled into

allowing the '\*' character would be incorporated in the dialed phone number. On certain phones or through certain

involve a malicious webpage providing a tel: link, such as <a href='tel:\*12345#'>Call Support</a>. If a user clicks the link with a tel: link that contains hidden USSD codes. When the user clicks on the link, thinking they are just going to call, they are exposed to the risk of the CVE-2022-22758 vulnerability.

[www.mozilla.org/security/advisories/mfsa2022-04/](https://www.mozilla.org/security/advisories/mfsa2022-04/) and the associated Bugzilla entry at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1758070](https://bugzilla.mozilla.org/show_bug.cgi?id=1758070) and CVE-2022-22749.

browser to navigate to some URLs that do not point to web content, potentially exposing the user to unintended

security advisories page at <https://www.mozilla.org/security/advisories/mfsa2022-01/> and the corresponding Bugzilla entry at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1758070](https://bugzilla.mozilla.org/show_bug.cgi?id=1758070)

scanned by a vulnerable version of Firefox for Android, redirects the user to a deceptive or dangerous URL that does not exist. This vulnerability affects Firefox for Android, Firefox ESR (Extended Support Release), and Thunderbird. It describes a situation where an attacker could redirect users to a malicious website before 100.0.2, Firefox for Android (versions before 100.3.0), and Thunderbird (versions before 91.9.1).

org/show\_bug.cgi?id=1770137 and <https://www.mozilla.org/security/advisories/mfsa2022-19/>.  
Script that exploits the prototype pollution vulnerability in JavaScript. The attacker would aim to manipulate the proxy, Firefox for Android, Firefox ESR, and Thunderbird that have addressed the vulnerability. Specifically, users should

JavaScript executing in the privileged parent process.  
Firefox for Android before 100.3.0.

.org/show\_bug.cgi?id=1770048 and <https://www.mozilla.org/security/advisories/mfsa2022-19/>.  
When processed by the parent process, results in double-indexing into a JavaScript object. This manipulation can lead to (L) characters is CVE-2021-4221.  
Character, it would cause the domain to be displayed to the right of the path. This rendering issue can lead to user confusion

mozilla.org/show\_bug.cgi?id=1704422- <https://www.mozilla.org/security/advisories/mfsa2021-38/>

Impacted by this vulnerability.  
When creating a link with an embedded RTL character. By doing so, the attacker could make the URL appear to point to

VPN products. An attacker could exploit this by crafting a custom login URL and convincing a VPN user to login via this URL.  
Build 929, Mozilla VPN Windows versions before 1.2.2, and Mozilla VPN Android 1.1.0 before build 1360.

For more advisories: <https://www.mozilla.org/security/advisories/mfsa2020-48/> - GitHub commits for the respective Mozilla products.  
An attacker could craft a malicious login URL and convince the victim to use it to login to their VPN account. If the attacker and victim are on the same network, the attacker could also intercept the login request.  
Several +Message Apps used by major Japanese carriers KDDI, NTT DOCOMO, and SoftBank. This vulnerability arises from the use of Unicode control characters to an affected +Message App. The app displays the text unprocessed based on Unicode code points.

For more information, see the advisories for KDDI +Message App for Android prior to version 54.4, NTT DOCOMO +Message App for Android prior to version 54.4, and SoftBank +Message App for Android prior to version 54.4.  
Users should update to the latest versions of the apps as recommended by the respective service providers: KDDI, NTT DOCOMO, and SoftBank. The new versions address the issue by including Unicode control characters to reorder the text display. This could make a malicious link appear as a normal link.

For more advisories and service information from KDDI, NTT DOCOMO, and SoftBank. The references include the following:  
CVE-2022-42544.

This flaw could potentially be exploited to mislead users into providing sensitive information.

For more information, see the Android Security Bulletin for Pixel devices on December 1, 2022. For more in-depth information, one can refer to the Android Security Bulletin for Pixel devices on December 1, 2022. This app could exploit the vulnerability within the getView method of the class. The vulnerability was first reported in the Android Security Bulletin for Pixel devices on December 1, 2022. Typically, due to the sensitive nature of these vulnerabilities, specifics of an exploit are not disclosed.

This issue can potentially be exploited to mislead users into providing sensitive information.

The reference to 'Android kernel' implies that various versions could be affected, but specific version details are not provided in the advisory.

<https://www.paloaltonetworks.com/security/bulletin/pixel/2022-12-01>

data out of bounds in the kernel's memory space. This could result in information disclosure of sensitive data or a crash. A missing bounds check in the `phNxpNciHal_core_initialized` function of `phNxpNciHal.cc` might lead to an out of

Following URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

running with system execution privileges. The application might attempt to exploit the vulnerability in the NFC (Near Field Communications) HAL component of Android.

Provider.java that may allow access to restricted tables. This security flaw could lead to information disclosure on

<https://www.microsoft.com/security/bulletin/pixel/2022-12-01>.

No user interaction is required for the exploitation to succeed.

crafted SQL queries targeting MmsSmsProvider.java. Since the vulnerability allows for SQL injection, the attacker

idation. An attacker could exploit this vulnerability to gain higher privileges on the system without needing additional interaction. An attacker could exploit it through a local access to the system.

versions impacted are not specified, but this vulnerability is referenced in the context of Android.

3 URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

al system could exploit improper input validation in the `trusty_ffa_mem_reclaim` function within `shared-mem-sm` -2022-42534. Typically, vendors and security platforms do not publish exploit code for security vulnerabilities to perform an out of bounds read due to a missing bounds check. This security flaw could lead to local information leakage. The security team at the vendor has assessed that the vulnerability presents a moderate level of risk but is not considered critical.

at the following link: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

vulnerability cannot be exploited by a standard user; elevated privileges are required.

ability could potentially be exploited without the device user's knowledge or intervention.

information stored on the device by performing an out of bounds read. This information disclosure could pose a privileged system process with System execution privileges reading beyond the bounds of allocated memory. Such an Android security vulnerabilities like this are typically addressed in subsequent Android Security Bulletins with firmware update. The CVE identifier for this vulnerability is CVE-2022-42531.

\_ldfw\_load.c file. It is a security flaw that could allow a bypass of mitigation mechanisms due to Permissive Memory Access. The severity of this vulnerability is rated as High, indicating that the vulnerability has a significant impact on the affected systems, implying that it poses a serious risk, and requires immediate attention.

security bulletin at this URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

erating system. However, specific version details have not been provided for this vulnerability.

ability could be exploited autonomously, potentially allowing an attacker to escalate privileges without any action on the user's part. An attacker with access to the device could potentially escalate their privileges within the system to gain access to resources that are stalled or running on an affected Android device. The application could exploit the vulnerability in the `mmu_map`



ck. This issue could lead to local information disclosure, and exploitation requires System execution privileges. No  
rare.

ng such vulnerabilities could lead to unethical practices, it's important to ensure that any discussion of code is do

xecuting code that leads to an out of bounds read in the Pixel firmware. This could result in information disclosure  
page: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

ons of Android. It has been assigned a base score of 9.8, classifying it as CRITICAL in severity. This vulnerability wa  
es that the vulnerability could have a significant impact on the confidentiality, integrity, or availability of an affect  
ecomended to check for updates and security bulletins related to their specific Android version and device mod  
es, available through the following link: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

on an affected Android device. This could result in the attacker gaining control over the device, accessing sensitiv  
erally, responsible disclosures of vulnerabilities do not include exploit code. Researchers and developers should  
application or using a compromised application to exploit the kernel vulnerability. Upon successful exploitation, th

le crash caused by a missing null check.  
action.

umbers are not detailed in the provided information.

etin webpage: <https://source.android.com/security/bulletin/pixel/2022-12-01>.  
vulnerability is present in cd\_SsParseMsg of cd\_SsCodec.c file.

ed message to an affected Android device. Due to the missing null check in the cd\_SsParseMsg of cd\_SsCodec.c, th  
ere is a potential out of bounds write due to the absence of a necessary bounds check. This could result in local e:

vulnerability information.

ally at <https://source.android.com/security/bulletin/pixel/2022-12-01>.  
y in ConvertUtf8ToUcs2 of radio\_hal\_utils.cpp to perform an out of bounds write. This could potentially overwrit

n attacker to execute code with system-level privileges.

erability have not been explicitly listed in the provided information.

<https://source.android.com/security/bulletin/pixel/2022-12-01>  
malicious application or process that is already running on the device with system execution privileges. The attac  
eges.

is related to SMS handling in Android devices. It involves a possible out of bounds read due to a missing bounds cl  
in the Common Vulnerabilities and Exposures (CVE) database. This means that it represents a significant risk that

provided in the CVE description. Users should refer to Android security bulletins and updates from their device manufacturer to the target device. Since the vulnerability lies in the handling of SMS messages, the attacker could exploit this vulnerability published on the 16th of December 2022, and users are advised to apply the security updates provided by the manufacturer at the URL provided in the references: <https://source.android.com/security/bulletin/pixel/2022-12-01>. Users should be aware of information disclosure. An attacker could exploit this vulnerability to read sensitive data from the device's memory with system privileges, particularly in the `ril_service_1_6.cpp` file, which is part of the radio interface layer. The flaw is an out of bounds read vulnerability presents a significant risk but is not the most critical level of threat.

on an Android device. This typically means that the attacker would need to have already achieved a high level of control over the device, which can be exploited without any direct input from the user, which makes it more dangerous as it can be exploited silently. Specific versions affected by this vulnerability were not provided in the CVE description.

For more information, see the Pixel update bulletin at <https://source.android.com/security/bulletin/pixel/2022-12-01>.

The vulnerability allows an attacker to gain access to system privileges, leading to a local escalation of privilege. This means that the security and control of the device are compromised as provided in the CVE report.

22.

The vulnerability is located in the `ril_service_1_6.cpp` file due to an incorrect bounds check. This issue could lead to local information disclosure, and it requires

For more information, see the Pixel update bulletin at <https://source.android.com/security/bulletin/pixel/2022-12-01>.

For more information, see the Pixel update bulletin at <https://source.android.com/security/bulletin/pixel/2022-12-01>. For version numbers, you should consult the Android Security Bulletin or related vendor announcements for the particular versions on the vulnerable Android device. No user interaction is required to leverage the vulnerability for information disclosure. The vulnerability allows an attacker to gain access to system execution privileges on an Android device. The application could exploit the out of bounds read vulnerability or the out of bounds write issue caused by improper input validation, which could potentially allow for local escalation of privilege. The CVE description indicates that it is a significant threat that could pose serious consequences if exploited but may not be as critical as other vulnerabilities disclosed in the Android Security Bulletin dated December 1, 2022. Typically, these bulletins include details about the vulnerability and the affected versions.

A high level of access to the system is necessary. This means that the vulnerability is not easily exploitable by a casual user on an Android device with system privileges. The application could leverage this vulnerability to perform an out of bounds read or write operation. For more information, see the Pixel update bulletin at this URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>. The bulletin is available at this URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

The vulnerability is a potential use after free. This bug could lead to a local escalation of privilege with no need for user interaction. The CVE ID is CVE-2022-42520.

6.7, categorized as MEDIUM severity.

For more information, see the Pixel update bulletin at <https://source.android.com/security/bulletin/pixel/2022-12-01>.

The vulnerability is a use after free, but the exact range of versions is not provided in the provided information.

The vulnerability allows an attacker to gain access to system-level privileges exploiting the use after free vulnerability to escalate privileges even further without user interaction. The vulnerability is located in the `cdmasmsdata.cpp` file, which is part of the Android operating system. It involves a possible information disclosure. The CVE description indicates that it is a significant threat that could pose serious consequences if exploited but may not be as critical as other vulnerabilities disclosed in the Android Security Bulletin dated December 1, 2022. Typically, these bulletins include details about the vulnerability and the affected versions.

The vulnerability is more critical as it could potentially be exploited without the user's knowledge.

This means that the attacker would need to have access to system-level privileges beforehand or exploit another vulnerability to gain access to system privileges.

Users should refer to the Android security bulletin for the specific versions patched for this vulnerability.

For more information, see the Pixel update bulletin at <https://source.android.com/security/bulletin/pixel/2022-12-01>.

The vulnerability allows an attacker to gain access to system privileges, leading to a local escalation of privilege. This means the attacker could execute arbitrary code with System privileges, potentially leading to a local escalation of privilege.

The vulnerability allows an attacker to gain a foothold on the target device with limited privileges. The attacker could then exploit CVE-2022-42519 to try to escalate privileges to system level. The CVE ID is CVE-2022-42518.

; bounds check, which could lead to local escalation of privilege.

is local escalation of privilege.

source.android.com/security/bulletin/pixel/2022-12-01.

esent on the device or a crafted payload that, when run, exploits the out of bounds write vulnerability in 'smsdata  
L8 haven't been disclosed publicly. Typically, detailed exploit code would be found in security research publication  
on Android, where there is a possible out of bounds read due to a missing bounds check. This could lead to local ir

code snippet demonstrating the vulnerability is not provided in the information. Generally, the issue arises from ;  
devices at: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

1 execution privileges performing actions that exploit the out of bounds read vulnerability. This could lead to unau

l.cpp.

URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>

ie precise affected versions are not detailed in the given CVE-2022-42516 description.

system-level permissions reading out-of-bounds memory in the affected ProtocolSimBuilderLegacy::BuildSimGetC  
d of the miscservice.cpp file within the Android operating system. This issue could lead to a possible out-of-boun  
nerability according to its CVSS rating.

lead to local information disclosure. However, exploitation would require system-level execution privileges.

s should refer to Android security bulletins or patches for information regarding the impacted versions.

ssThreshold function.

ies, like CVE-2022-42515, due to security reasons. The description mentions where the vulnerability is within the

particularly the page dedicated to security updates for Pixel devices, as listed in the references included with the C  
stem-level privileges due to a separate exploit or a compromised device. The application could then leverage the

protocolmsbuilder.cpp within the Android kernel. This issue involves a possible out of bounds read because of a n  
ivileges on the affected device.

erability.

Bulletin page: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

not listed in the provided details.

read issue in ProtocolLmsBuilder::BuildSetConfig to potentially access sensitive information stored in memory. Th

protocolembmsbuilder.cpp', specifically in the function ProtocolEmbmsBuilder::BuildSetSession. This issue arises du

M severity category.

ability is associated with the Android operating system, and would pertain to relevant versions active before the vulnerability was discovered. For more information, see the Android security bulletin at [source.android.com/security/bulletin/pixel/2022-12-01](https://source.android.com/security/bulletin/pixel/2022-12-01).

on of privilege, which might enable them to execute arbitrary code with system-level privileges, potentially leading to a local escalation of privilege. In the case of the `ProtocolEmbsBuilder::BuildSetSession` function, there are often no public code examples available due to the sensitivity of the data being processed. In the case of the `ProtocolEmbsBuilder::BuildSetSession` function, there are often no public code examples available due to the sensitivity of the data being processed.

within the `vsimdata.cpp` file. This issue can lead to a possible out of bounds read due to a missing bounds check. The severity of this issue is not explicitly mentioned in the provided information.

is vulnerability are not explicitly mentioned in the provided information.

ation privileges on the affected device.

ixel devices at the following URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>

is reading data outside the bounds of an intended buffer due to the missing bounds check. This could lead to unauthorized access to sensitive data.

ere's a potential out of bounds write due to a missing bounds check. This vulnerability could allow a local attacker to escalate their privileges on the affected device.

is of Android affected were not specified in the given information.

essment metrics.

source.android.com/security/bulletin/pixel/2022-12-01.

ned System execution privileges on an Android device. This app could exploit the lack of a bounds check in the `encode` function of the `StringsRequestData::encode` function. In the case of the `StringsRequestData::encode` function, there are no code examples included here. It is advised that developers look at the patches and updates provided for CVE-2022-42510.

ngsRequestData::encode of requestdata.cpp. This vulnerability could lead to a local escalation of privilege with System execution privileges on an Android device.

he CVE description. It is typically best to refer to the Android security bulletin for detailed version information.

om/security/bulletin/pixel/2022-12-01

, which can be accessed here: <https://source.android.com/security/bulletin/pixel/2022-12-01>

eges crafting malicious input data to trigger an out of bounds read in the `StringsRequestData::encode` function of the `StringsRequestData::encode` function. In the case of the `StringsRequestData::encode` function, there are no code examples included here. It is advised that developers look at the patches and updates provided for CVE-2022-42509.

ata::encode function within the `callreqdata.cpp` file. This vulnerability is due to a missing bounds check and could lead to a local escalation of privilege.

<https://source.android.com/security/bulletin/pixel/2022-12-01>.

: providing a specific range or version numbers.

Already have a high level of access on the system.

ed System execution privileges manipulating the `CallDialReqData::encode` function to perform an out of bounds write.

method within the `protocolcallbuilder.cpp` file of the Android operating system. It involves a possible out of bounds memory access, which is a high severity issue according to the scoring system used to assess the impact of security vulnerabilities.

oid device. This means that the vulnerability could be exploited by an attacker who already has a high level of control over the device. This increases the ease with which the vulnerability can potentially be leveraged by an attacker since no action from the user is required.

[//source.android.com/security/bulletin/pixel/2022-12-01.](https://source.android.com/security/bulletin/pixel/2022-12-01)

Item execution privileges writes data out of bounds while using the BuildSendUssd method in the ProtocolCallBuilder of 'protocolsimbuilder.cpp'. It pertains to a possible out-of-bounds write due to a missing bounds check. This flaw could lead to a denial of service (DoS) or a crash, depending on the system's configuration and the severity of the write.

ing as an attacker could theoretically exploit the vulnerability without any active participation from the victim.

this requirement, the fact that user interaction is not needed makes the vulnerability still concerning.

at the following URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>

n privileges could manipulate the Android kernel by exploiting the missing bounds check in 'ProtocolSimBuilder::Fave not been listed in the provided description, but users should check their device's compatibility with security C.a.cpp within Android. It refers to a possible out of bounds write due to a missing bounds check. If exploited, it could lead to a denial of service. According to the Common Vulnerability Scoring System (CVSS).

ailed in the provided information.

hout any action from the user.

on details of this nature are usually not disclosed to prevent misuse. However, the vulnerability is linked with an   
 ularly the Pixel update bulletin for December 2022, available at: <https://source.android.com/security/bulletin/pixel>   
 on privileges and carrying out actions without user knowledge or consent, possibly leading to data corruption, un-

red in the vulnerability.

[source.android.com/security/bulletin/pixel/2022-12-01.](https://source.android.com/security/bulletin/pixel/2022-12-01)

### ulnerability.

acker to execute code with System privileges, which can lead to taking control of the device and compromising the data stored by the device manufacturer or Google for their Android devices once they are available, to ensure the vulnerability can be exploited. The out of bounds write vulnerability in ProtocolMiscBuilder::BuildSetSignalReportCriteria. The malicious application within the callreqdata.cpp file of the Android operating system. It involves a possible out-of-bounds write due to

<https://source.android.com/security/bulletin/pixel/2022-12-01>.

specific version numbers are not mentioned. Users are advised to check the Android Security Bulletin for detailed information on this vulnerability by manipulating the `CallDialReqData::encodeCallNumber` function to perform an out-of-bounds write. The exploit code examples. To protect users, details that could potentially aid attackers are typically not released.

portCriteria of protocolmiscbuilder.cpp, due to a missing bounds check. This issue could potentially allow a local a  
2022-42503.

ity vulnerability.

<https://www.mcafee.com/security/bulletin/pixel/2022-12-01>.

attacker with system execution privileges could potentially write arbitrary code to a location in memory that they could execute. This is a high severity issue in the Android operating system. It describes a possible out of bounds write due to a missing bounds check, which could lead to a denial of service or a system crash (CVSS).

are not explicitly mentioned in the CVE description.

[/source.android.com/security/bulletin/pixel/2022-12-01.](https://source.android.com/security/bulletin/pixel/2022-12-01)

an execution privileges could exploit the out of bounds write vulnerability in the simdata.cpp FacilityLock::Parse function to write data outside the memory boundaries that have been allocated for that data, potentially overwriting other data and causing a crash. This is CVE-2022-42501.

ck that could potentially lead to an out of bounds write. This flaw could allow a local attacker to execute code with elevated privileges on an affected Android device.

E-2022-42501 were not detailed in the provided information.

y. This score indicates that the vulnerability poses a significant risk, but it's not considered critical.

, specifically at <https://source.android.com/security/bulletin/pixel/2022-12-01>.

Android kernel exploiting the missing bounds check in the HexString2Value function of util.cpp. By doing so, they could perform an out of bounds read due to a missing bounds check. This could allow for remote code execution with LTE authenticating to the scoring system used to assess the impact of security issues.

ersion range is not provided. Users are advised to check the Android security bulletins or with their device manufa  
URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

ited remotely with LTE authentication.

ounds read, it might allow an attacker who possesses the necessary LTE authentication capabilities to execute co  
ticate to the LTE network and then sending crafted requests to the cellular modem. By exploiting the out of bound

ends read due to a missing bounds check. The issue could lead to local information disclosure without needing ad

<https://www.paloaltonetworks.com/security/bulletin/pixel/2022-12-01>

el. The specific kernel version is not mentioned in the provided details.

out of bounds read vulnerability present in Pixel cellular firmware. An attacker with access to the device could exploit its access to the vulnerable cellular firmware or related source code, it is not feasible to provide a specific code example of remote exploitation in the provided information, indicating that the vulnerability would require local access. The vulnerability could lead to information disclosure without needing additional execution privileges.

3. Bounds read caused by an incorrect bounds check. This flaw could result in local information disclosure without requiring a CME exploit. **CVSS 3.1 Base Score: 5.9** (Medium severity).

provided information. It refers to 'AndroidVersions' generally, which suggests that multiple versions could be affected. For more information, see the following URL: <https://source.android.com/security/bulletin/pixel/2022-12-01>.

ect bounds check in Pixel cellular firmware. An attacker could potentially exploit this by accessing memory outside the bounds of the buffer, potentially leading to a crash or a malicious application running on the device that takes advantage of the firmware's bounds check vulnerability.

which is caused by a missing bounds check. This vulnerability could potentially lead to remote code execution with

[android.com/security/bulletin/pixel/2022-12-01](https://android.com/security/bulletin/pixel/2022-12-01)

entials; however, no user interaction would be required to carry out the attack.

A packet to the vulnerable device over the LTE network. This packet, when processed by the device's cellular firmware, is likely sensitive and not published to prevent misuse. Typically, vendors do not provide such code examples for security. This is a CVE in the SAEMM\_RadioMessageCodec.c file. It involves a potential out of bounds read due to the absence of a bounds check according to the standards set by the Common Vulnerability Scoring System (CVSS).

The vulnerability is described to exist in the Android operating system, but detailed version numbers are not provided by an attacker without the need for the victim to take any action.

[android.com/security/bulletin/pixel/2022-12-01](https://android.com/security/bulletin/pixel/2022-12-01)

To solve an attacker crafting a malicious message designed to trigger the out of bounds read vulnerability in SAEMM\_RadioMessageCodec.c to successfully exploit this vulnerability, they would need to have privileges that allow them to execute system firmware within the Android operating system. It involves a possible out of bounds read due to an improperly conducted out of bounds read. Generally, it is advisable to refer to official Android security bulletins for details on affected versions. This indicates that the vulnerability poses a significant risk and should be addressed with priority.

on disclosure due to incorrect bounds checking in the code.

The Android security bulletin suggests that details about the patch or update for this vulnerability could be found at [android.com/security/bulletin/pixel/2022-12-01](https://android.com/security/bulletin/pixel/2022-12-01) to access sensitive information in the system's memory without the need for any additional permissions or user

of a possible out of bounds read due to a missing bounds check in SAECOMM\_SetDcnIdForPlmn function of SAECOMM

on, but it is referred to as 'Android kernel'.

nerability.

interaction. The vulnerability can be exploited remotely.

[android.com/security/bulletin/pixel/2022-12-01](https://android.com/security/bulletin/pixel/2022-12-01).

504. However, it likely involves improper validation or a lack of boundary checks within the 'SAECOMM\_SetDcnIdForPlmn' function. Requests to the affected function, which may lead to unauthorized disclosure of information stored in out-of-bounds

is part of the Android operating system.

[android.com/security/bulletin/pixel/2022-12-01](https://android.com/security/bulletin/pixel/2022-12-01)

: specifying any particular version number.

ving a remote attacker to execute arbitrary code and take control of the affected device.

A packet to an affected Android device that triggers the out of bounds write condition within SetDecompContextDb fi

ing system.  
ty/bulletin/pixel/2022-12-01.

the Android kernel to execute unauthorized code or commands, potentially gaining elevated privileges on the device. This information was not provided to the public to prevent malicious use. Instead, the information provided in security bulletins and the Pixel update bulletin at: <https://source.android.com/security/bulletin/pixel/2022-12-01>.  
The bulletin provided by the Android Security Team, apply any patches or updates released for their specific device, are rated as HIGH severity.

[source.android.com/security/bulletin/pixel/2022-12-01](https://source.android.com/security/bulletin/pixel/2022-12-01).  
The severity were not detailed in the provided information.

For details about the vulnerability and potential code examples, further investigation into the vulnerability details are required. A score of 7.5 HIGH typically indicates that an attacker could exploit it to gain unauthorized access or privileges, possibly through the Android kernel is CVE-2022-20600.  
The vulnerability is a Denial of Service (DoS) disruption.  
The attack is performed at the system level.  
The details are not provided in the provided information.  
No user interaction is needed for the exploitation.

[source.android.com/security/bulletin/pixel/2022-12-01](https://source.android.com/security/bulletin/pixel/2022-12-01)  
The example might involve incorrect handling of buffer sizes. However, without specific details about the function, it is difficult to determine the exact impact. The attacker could execute malicious code that manipulates memory in an unsafe way, potentially leading to a Denial of Service (DoS) or other system-level issues.





df8986dbba2e02c5bf82f105b36243 and on the Android Security Bulletin for October 2023 at <https://source.android.com/security/bulletin/2023-10-01>. This vulnerability allows an attacker to execute arbitrary code with the privileges of the application. Since no user interaction is required, the app should be updated to date and avoid installing applications from untrusted sources. Additionally, app developers and OEMs should use a unique symmetric key for encryption, which can be obtained by reverse-engineering the Android Client and server-side web applications. This issue should be addressed with urgency.

By reverse engineering the client and server applications, any attacker with knowledge of this key can intercept or modify messages sent over the MQTT connection.

An attacker could use a compromised DES key to create and send malicious messages. The compromised MQTT connection permits the attacker to decrypt messages using the compromised DES key to decrypt and potentially modify the message payloads. Upon re-encryption and forwarding the messages to the MQTT broker, potentially allowing an attacker to coerce the application to establish a connection with a malicious MQTT broker.

This vulnerability could result in misinformation or unauthorized actions on the HMI device, potentially leading to disruptions in service, operational downtime, or data loss.

MQTT is a lightweight messaging protocol widely used in the Internet of Things (IoT) for device-to-device communication. The vulnerability allows an attacker to intercept and modify messages sent between the client and the broker. Once the client connects to the attacker's broker, the attacker could send crafted messages to the client, potentially causing the client to perform unauthorized actions. Using other secure authentication methods to ensure that the client is connecting to a legitimate server. Furthermore, the severity of this vulnerability may vary depending on the device settings, including device power management and potentially the device's secure settings, like ADB debugging. Vulnerabilities may not be as critical as those with 'HIGH' or 'CRITICAL' severity ratings.

Without proper authorization, this app could be used to gain unauthorized access to various device functionalities like camera, microphone, location, etc. that address the security flaw. Other general recommendations include regularly reviewing and restricting user permissions, updating the app to the latest version, and avoiding the abuse of existing functionality or security weaknesses in the Kiosk mode software, rather than conducting a thorough security audit.

When the app is accessed over an unencrypted HTTP address, it retrieves sensitive information like IP address and MQTT broker credentials using the unencrypted HTTP. This could lead to unauthorized access to the affected system if exploited.

Since the app uses HTTP, the attacker could capture sensitive information like IP addresses and MQTT broker credentials. To mitigate this risk, the app should use HTTPS for all sensitive communications, ensuring all communication is encrypted in transit, and monitor for suspicious activity. Additionally, the user should be prompted to manually enter the server IP address. This could lead to the exposure of sensitive information such as IP addresses and MQTT broker credentials.

Since this feature is not user-configurable, users cannot easily mitigate the risk by switching to a secure protocol like HTTPS.

This vulnerability could allow an attacker to retrieve sensitive information like MQTT broker credentials and server IP addresses. This could allow the attacker to eavesdrop on the communication between the client and the broker. The attacker could also use this information to modify sensitive settings in the Android Client application. The following code snippet shows the vulnerable code using HTTP to retrieve MQTT broker information:

```
private void retrieveBrokerInfo() {
```

```
} // Vulnerable code using HTTP to retrieve MQTT broker info
```

could lead to data leakage, unauthorized setting changes, or other unforeseen impacts depending on what settings a third-party app could potentially send crafted intents or use content resolver calls to interact with the content application, potentially causing it to connect to a server under the control of an attacker. The issue arises due to various consequences if successfully exploited, such as allowing an attacker to steer the client application's network

mal

by constructing a fake broadcast intent containing server configuration data encrypted with the known hardcoded key where the intent is coming from or whether the source is trustworthy. An attacker could craft a broadcast intent lik

unauthorized access to sensitive resources.

host header could mislead the application into generating incorrect links, which could redirect a user to an attacker

on-CVE-2023-36085

potentially enable attacks such as JavaScript execution, native code execution, and credential theft. This vulnerability, or availability of the affected system.

workarounds for this issue, so updating the app is crucial for maintaining security.

web content, stealing user credentials, or performing actions on behalf of the user within the application. Attackers could exploit this vulnerability to execute arbitrary code on the device, potentially leading to data theft or unauthorized access to sensitive resources. This advisory will contain detailed information about the vulnerability, affected versions, and the remediation steps. The vulnerability implies that malicious URLs could be loaded into a WebView component, bypassing typical security checks and potentially leading to data leakage or unauthorized access to sensitive resources.

unauthorized access to data or elevated system permissions.

confidentiality, integrity, or availability.

to escalate their privileges. For instance, the attacker could exploit the vulnerability to execute code with elevated privileges, potentially leading to data theft or unauthorized access to sensitive resources. This vulnerability can be exploited by third-party applications to gain access to and potentially misuse not exported activities of a

activities of any installed application.

For example, if the malicious app is installed on a device and the user grants it the necessary permissions, the malicious broadcast would result in the malicious app being able to execute arbitrary code on the device, potentially leading to data theft or unauthorized access to sensitive resources. The vulnerability could potentially grant the attacker access with system-level privileges, which could lead to a range of fu

security. The malicious app might access sensitive components of other applications or perform actions on behalf of the user, potentially leading to data theft or unauthorized access to sensitive resources.

malicious intent intended for the streamer component, leading to limited information disclosure, denial of service, and

was intended for the streamer component of GeForce Now, potentially leading to information disclosure, service

exploitation before affected users have a chance to apply necessary updates or mitigations.  
ed sources and to be wary of any apps that request permissions not relevant to their intended purpose.

lerActivity component is improperly exported, allowing a third-party app without permissions to craft an Intent t

ain/CVE-2023-42468, <https://github.com/actuator/com.cutestudio.colordialer/blob/main/CWE-284.md>, and <http://top.secure.encrypted.com/poc.apk>, <https://github.com/actuator/com.full.dialer.top.secure.encrypted/blob/main/CWE-284.md>, and <https://github.com/actuator/com.full.dialer.top.secure.encrypted/blob/main/CWE-284.md> for triggering the DialerActivity component of the vulnerable com.cutestudio.colordialer application. By sending this inte

lace phone calls without user interaction.

er.top.secure.encrypted/blob/main/poc.apk, <https://github.com/actuator/com.full.dialer.top.secure.encrypted/blob/main/CWE-284.md>, and <https://github.com/actuator/com.full.dialer.top.secure.encrypted/blob/main/CWE-284.md> for triggering the dialer to initiate a phone call. As this requires no permissions, the user might not notice the ca

| HTML page.

\_12.html, a bug tracker at <https://crbug.com/1462104>, and security advisories from Debian and Fedora at <https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/6T655QF7CQ3D> and then trick a user into visiting this page. Doing so, the attacker could obfuscate or alter the security UI elements c  
oitation of the vulnerability. The detailed technical aspects would generally be found in the vulnerability reports, p  
against possible exploitation of this vulnerability.

icious page designed to steal their information.

<https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/6T655QF7CQ3D>  
ge, they might be tricked into inputting sensitive information such as login credentials, thinking the page is secure  
lly crafted HTML page. The severity of this vulnerability is rated as Medium.  
ie.

ie permission prompt clearly, which can lead to inadvertently granting permissions that the attacker wants.

and Debian. The original references for this CVE include links to those resources for detailed information.  
iscate permission prompts. Unsuspecting victims might then grant permissions thinking they are accepting somet  
s. As a responsible disclosure, detailed exploitation techniques are typically not publicly shared to prevent misuse  
r to protect against this vulnerability.

formation, session hijacking, and other client-side attacks.

i passed in an intent by a third-party application without any permissions.

uator/wave.ai.browser/blob/main/CWE-94.md', and 'https://github.com/actuator/cve/blob/main/CVE-2023-424'. The JavaScript code meant to be executed within the browser's WebView. When this intent is sent to the exported Splash screen, it grants advanced privileges and could facilitate the compromise of the affected system leading to significant impacts such as unauthorized access to sensitive data. The crafted intent is sent to an exported component, namely the com.mm.android.easy4ip.MainActivity activity. This security vulnerability allows an attacker to execute arbitrary code remotely on a victim's device without the need for any interaction from the user.

When the application is executed and loads web content directly, an attacker could leverage this to execute arbitrary code on the device. The vulnerability is located in the file https://github.com/actuator/imou/blob/main/poc.apk- https://github.com/actuator/cve/blob/main/CVE-2023-4

With the fact that an exported component (com.mm.android.easy4ip.MainActivity) can receive a crafted intent, the

attacker can access the camera without the user's explicit consent, potentially accessing the camera feed without permission.

The vulnerability was activated without the user's knowledge. The attack could potentially be part of a phishing scheme where the attacker uses the Camera API in Android SDK 5.1.1 API 22 and below, where explicit runtime permission checks are not performed.

Android 13.

The vulnerable Camera app. When a user unknowingly interacts with the malicious application, the attacker could intercept the data and exploit the vulnerability that allows code execution on the device.

The attacker might obtain sensitive information from the accessed files.

This vulnerability is categorized as an 'Improper export of android application components vulnerability'. This allows a local attacker to access sensitive data and functionalities.

The vulnerability was discovered by the device manufacturer, in this case, Samsung.

The application does not function without the user's consent.

The application runs on the device, which could be exploited to change settings that should otherwise be protected.

This application could invoke the improperly exported component using an 'Intent' to activate or deactivate the Accessibility service. The application patches to protect against this vulnerability.

The attack. In a tapjacking attack, a malicious application could masquerade as a legitimate app, deceiving the user into interacting with the client's interface. This would deceive the user into believing they are interacting with the WARP client when they make taps and gestures intended for the WARP client and redirect them to the attacker's application. The attacker's application is not harmful in specific scenarios, its overall impact is considered to be low, likely because it requires user interaction to be successful.

Tapjacking attacks. Users should also be cautious about installing applications from unknown sources, as this is a common method for the WARP Mobile Client to be opened and then overlay its own interface on top of the client without the user's knowledge.

xpx7. Additionally, the WARP Mobile Client's official documentation and updates are available at [oid function under specific conditions, enabling it to influence the task behavior of the vulnerable WARP app.](https://develop</a></p></div><div data-bbox=)

n/warp-client/

s operation on the user's Android device. Since this requires the malicious app to be installed first, the exploitation should be used or distributed for this purpose. The emphasis should be on understanding the vulnerability, mitig

, the Apple App Store page for the Skylark app (<https://apps.apple.com/jp/app/%E3%81%99%E3%81%8B%E3%81> n in the custom URL scheme handler in the Skylark app, for example, if the app does not properly validate incoming a specially crafted intent to the Skylark app on Android or a URL to the app on iOS. This intent or URL would ap

hing attacks, or exposure to malicious content.

d be leveraged to redirect a user to a phishing website that mimics a legitimate service, encouraging the user to e  
nts of a dialog URL through a specifically crafted HTML page.

w.debian.org/security/2023/dsa-5479- Lists of Fedora Project package announcements.

g URL presented by the WebShare feature in Chrome on Android.

tracts with this page and uses the WebShare feature, they could be misled into thinking that they are sharing info  
mended to keep browsers and other software up to date with the latest security patches.

eature by using a crafted HTML page. This could lead to information disclosure or potentially more serious attacks

erable version of Google Chrome on Android, the attacker could potentially retrieve sensitive information filled l  
ite-for-desktop\_15.html), the Chromium bug tracker (<https://crbug.com/1465230>), Debian security announcemer

gh a specially crafted HTML page.

tps://www.debian.org/security/2023/dsa-5479- <https://lists.fedoraproject.org/archives/list/package-announce@> user were to visit this malicious webpage, the attacker could potentially spoof the contents of the Omnibox, decei

t to allow developers to understand and patch the affected systems without revealing exploit methods.  
ersion 116.0.5845.96, could allow a remote attacker with control over the renderer process to potentially cause h

ed HTML page that takes advantage of the use after free vulnerability in the Offline feature of Google Chrome on .  
esktop\_15.html), the related Chromium bug report (<https://crbug.com/1448548>), the Debian security announcen  
e feature of Chrome by manipulating memory in the browser after it has been freed. This could corrupt memory a  
d user to gain unauthorized access to certain information through local access.  
ss to that resource.

o data leaks and violations of user privacy.

oper access control vulnerability.

y.

ormation that should be restricted. This could lead to unauthorized disclosure of personal data or confidential syst

e.com/reports/1997029), and a Nextcloud security advisory (<https://github.com/nextcloud/security-advisories/se>  
nt by sending a crafted intent that causes the Nextcloud Talk app to erroneously write files to a directory outside c

enforce security policies.

ferences provided.

e this attack relies on the mishandling of cross-origin data, it could potentially lead to privacy violations or unauth  
r this issue.

blicly shared to avoid facilitating potential attacks. Still, a general understanding is that this exploit would involve

o execute arbitrary code through the app's 'com.viatom.baselib.mvvm.webWebViewActivity' component.

thub.com/actuator/cve/blob/main/vihealth.md'.

bitrary code on the user's device without their knowledge, potentially leading to data theft, app manipulation, or update that addresses the issue and instruct users to install it immediately. Users can also ensure that their device

urity flaw is classified with a medium severity level.

[.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/PQKT7EGDD2P3L7S3NXEDC](https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/PQKT7EGDD2P3L7S3NXEDC)

isiting the page which could then bypass same origin policy restrictions, potentially allowing access to sensitive data attackers. However, developers and security professionals perform testing in controlled environments to understand

, potentially leading to insufficient notification content visibility.

ld trick the user into interacting with the notification in an unintended manner, which might lead to further exploitation

21.html, the Chromium bug tracker at <https://crbug.com/1311683>, and Fedora's mailing list archive, with details on

rypted network traffic.

m an affected version of Apple Music for Android to capture sensitive information due to the lack of HTTPS encryption or ARP spoofing. The attacker could capture unencrypted data transmitted by the app, potentially exposing user c

ser's contact list. This information could then be used for various malicious purposes such as spamming, phishing, attached version, and it is not productive to provide such examples that could aid in malicious activities.

ion, or further exploitation of connected systems. An attacker could conveniently use adb commands to gain shell following command: `adb connect <device\_ip>`. Once connected, they could issue various adb commands like `adb c



into entering sensitive information like login credentials or personal data. Another scenario could be to mislead Bluetooth Low Energy (BLE), is susceptible to replay attacks. This means that a malicious user could intercept BLE signals and replay them to unlock the MojoBox Digital Lockbox. Second, if the attacker gains physical access to the Android device,

the owner's smartphone when the lock is opened. The attacker then replays these signals at a later time, bypassing the lock. For more information, see CVE-2020-2307-exploits/mojobox14-replay.txt, a blog post at 'https://www.whid.ninja/blog/mojobox-yet-another-not-so-secure'.

Users could minimize the risk by disabling Bluetooth when not in use, or by using additional physical security measures. Versions of the software prior to 11.1.2.3.1 are vulnerable to this issue, which can be exploited without authentication. This access may allow the attacker to bypass security measures without needing user interaction or system privileges. This vulnerability can cause significant confidentiality impact, without requiring user privileges or interaction, through local access.

The impact of this vulnerability includes the possibility of the compromised system being used as a platform for further attacks against the network. Physical access to the communication segment connected to the Oracle Mobile Security Suite can also reduce the effectiveness of the suite. It is primarily detailed as a vulnerability that allows unauthorized access to the Oracle Mobile Security Suite when the suite is operating. The attacker could exploit the vulnerability by intercepting traffic or issuing unauthorized commands.

This vulnerability could lead to unauthorized functions, or information disclosure.

The impact of this vulnerability is context dependent on the nature of the tampering.

This vulnerability affects the Calendar application prior to 12.4.07.15 on Android 13, which could allow attackers to write arbitrary files to a device.

Attackers could write arbitrary files to arbitrary locations on the device. This could allow the attacker to overwrite sensitive files or plant malicious code. For example, if an attacker places files outside the intended extraction directory, possibly overwriting sensitive data.

Brave Browser automatically navigated to scanned URLs without displaying the URL to the user first. This could allow a malicious actor to redirect the user to a malicious website. Brave Browser requires the user to manually navigate to the URL, thereby preventing automatic redirection to potential malicious websites.

If a user scanned this QR code with Brave Browser's QR scanner, they would be automatically redirected to the malicious website. This could allow a local attacker to analyze the app's data and obtain an API key for an external service. This vulnerability is of critical severity.

For more information, see the following links: Google Play for Android (https://play.google.com/store/apps/details?id=com.news) and the application. Furthermore, they might use the discovered hard-coded API key to access sensitive data within the application.

loper.arm.com/Arm%20Security%20Center- <https://github.com/ARM-software/android-nn-driver/releases/tag/v1.0.0>  
The affected driver could be tricked into reading or writing memory outside the bounds of allocated structures, potentially allowing an attacker to execute arbitrary code that performs length checks on shared memory and ensure that bounds are properly verified to prevent further exploitation.

connect to the cup and bypass the application's client-side chat censor filter.

4. The attacker could exploit this to display inappropriate or offensive messages by bypassing the client-side censor filter. This could lead to the disclosure of sensitive information on both the client and server sides. Users should apply the update as soon as it is available. Additionally, users should be aware of the potential for abuse of this vulnerability.

without needing any additional execution privileges.

notification. This could deceive the user into believing there is no active service when there is, leading to the disclosure of sensitive information.

5. The attacker could exploit this to gain elevation of privilege with the attacker needing System execution privileges. The exploitation does not require user interaction. However, due to the nature of the vulnerabilities, it still requires attention and remediation to prevent potential exploitation.

6. The attacker could exploit this to gain access to the affected system, compromising data security, and potentially spreading the attack to other systems.

7. The attacker could exploit this to gain access to sensitive information without requiring additional execution privileges. Users should be aware of the potential for abuse of this vulnerability and should refer to detailed version information.

8. The attacker could exploit this to gain access to sensitive information without additional privileges and does not require user interaction, an attacker might leverage this vulnerability to gain elevation of privilege without the need for additional execution privileges. However, user interaction is required to successfully exploit this vulnerability.

9. The attacker could exploit this to gain access to sensitive information from additional sources or advisories.

: failure to lock display power, bypassing the confirmation screen that is meant to secure sensitive operations, the

ue to a heap buffer overflow. Without the need for user interaction or additional privileges, this could allow the a  
exceeds the buffer's storage capacity, which can be caused by programming errors such as incorrect buffer size ca  
information disclosure without requiring additional execution privileges.

ead. The scenario would typically involve accessing an array or buffer beyond its allocated size, potentially due to  
ition, an attacker may craft malicious packets or utilize specific protocol requests to remotely trigger the out of bc  
ege, requiring System execution privileges to exploit. It affects the Android kernel, and no user interaction is nece

ecute arbitrary code.  
after the attacker has gained a certain level of access to the system.

ce: ``cvoid load\_dt\_data(char \*data, size\_t data\_size) { char buffer[128]; if (data\_size > sizeof(buffer)) { // CVE-2  
bypassing security mechanisms, installing rootkits, or even taking full control of the system. The attack could be fu  
ad to remote information disclosure without requiring any additional execution privileges or user interaction.  
letin for detailed version information and ensure their systems are updated to the patched version.

ific information about the vulnerability, including affected versions and resolutions.  
mation without needing any additional privileges or user interaction.  
encryption, no special execution privileges or user interaction are required.  
a, credentials, or other confidential information, potentially leading to privacy breaches, identity theft, or seconda  
potential exploitation.

nformation disclosure without requiring additional execution privileges or user interaction.



ails of the vulnerability and potential fixes are discussed.

is. Since the deserialization is unsafe, they could manipulate data structures to read memory they should not have

g memory out of bounds, potentially disclosing sensitive information stored in memory to the attacker, without re

escalation of privilege with the need for System execution privileges. User interaction is not required for exploitat

out any user interaction. Since this vulnerability can lead to local escalation of privilege, the malicious app could p

fe deserialization that can lead to an out-of-bounds read, potentially resulting in local information disclosure. To e

system-level processes or applications.

!023-06-01

The attacker could then trigger the unsafe deserialization vulnerability in `initiateVenueUrlAnqpQueryInternal` to r  
n Source Project (AOSP) repositories for commits that fix the issue, as they may contain detailed code changes th

ges or user interaction. This could allow the attacker to read sensitive information out of the bounds expected by

to local information disclosure with the attacker requiring System execution privileges. This vulnerability impacts

erability by running a malicious application or script that targets the wifi server component to access sensitive data could lead to further attacks such as network intrusion, account takeovers, or privacy breaches.

privileges required. User interaction is not required to exploit this flaw.

rmation or executing unauthorized code with escalated privileges.

tion disclosure over Bluetooth, and exploitation does not require user interaction.

o perform an out of bounds read. This could potentially lead to the disclosure of sensitive information without the

could lead to an out-of-bounds read, potentially causing a remote denial of service without requiring any additional

ility in the 'on\_create\_record\_event' function by exploiting the missing null check, which can result in an out-of-b

entially result in local information disclosure without requiring additional execution privileges. The affected funct

er input validation, an attacker may be able to read out of bounds memory, leading to information disclosure. No  
be included in Android's security bulletin, possibly found at the provided reference link ([https://source.android.co](https://source.android.com)

e attacker might craft malicious inputs or commands that interact directly or indirectly with this function, leading to  
1. This should give you detailed insights into the vulnerability, affected versions, and the status of any patches or

rmation disclosure with the need for System execution privileges. No user interaction is required for exploitation.

ting system privileges to only trusted applications and processes to minimize the impact of such vulnerabilities. within the function `remove_sdp_record`. This vulnerability could potentially lead to local information disclosure v  
ies.

tions of the issue, impacted versions, and any mitigations or patches that have been released.  
ms to receive a timely security update.

ormation disclosure.

perform an out of bounds read operation. Since no extra privileges or user interaction are required, the app could s  
eir disclosure. Users should check the most recent Android security bulletins and updates for their devices to det  
h could result in remote information disclosure without needing additional execution privileges. Exploitation doe

user interaction.

access to memory contents that could include sensitive information like keys, credentials, or personal user data. S

used by a missing bounds check that might lead to local information disclosure.

etooth function. By doing so, the attacker could potentially access and disclose sensitive information within the d

vulnerability could result in local information disclosure over Bluetooth, but exploitation would require comprom

ure of sensitive information over Bluetooth without the need for any user interaction. The attacker would leverag

This might allow the attacker to access sensitive data or gain further leverage on the system in a multi-stage attack or remote information disclosure without requiring any additional execution privileges. This issue affects Android versions 4.0.3 to 4.2.2.

This could lead to the application unexpectedly aborting and potentially leaking sensitive information to the attacker without needing additional execution privileges.

As for local escalation of privilege, the attacker could potentially gain elevated system permissions on the victim's device.

Since user interaction is not required, the attacker's application could silently trigger a notification hide preferences, due to an unusual root cause. This flaw could enable local information disclosure without the need for additional execution privileges.

When the connection is turned off, the attacker could potentially eavesdrop on the Bluetooth connection, resulting in the disclosure of sensitive information. This could lead to local escalation of privilege without the need for additional execution privileges. User interaction is required for this exploit.

Privileges.

Elevated privileges on the device. Since there's no code example required or given for this kind of logic error exploit, the attacker could potentially gain elevated privileges on the device. This could lead to local information disclosure.



used to gather information for further attacks or to bypass security mechanisms, potentially leading to further exploitation. A buffer overflow occurs when data is copied into a buffer without proper bounds checking, potentially overwriting memory and leading to local escalation of privilege without requiring additional execution privileges.

operations on the device.

sensitive activities that can be invoked without proper authorization. In addition, staying updated with Android's security updates is crucial to prevent malicious purposes, installing unauthorized applications, or extracting sensitive information. No code examples are provided.

If no user interaction or additional privileges are needed, an attacker could potentially cause a denial of service remotely by denying any updates or patches provided for their devices to address the issue.

privilege from the guest user without requiring any additional execution privileges. No user interaction would be needed to exploit this vulnerability, and the issue should be addressed promptly.

perform unauthorized actions such as accessing sensitive information, enabling or disabling system functionality, or causing a denial of service. This could enable an attacker to escalate privileges without needing additional execution privileges.

access to privileged API calls or system functionalities without the proper permissions. This could enable the attacker to perform unauthorized actions. For more details, see the Android security bulletin for the exact patch details and update their devices as recommended by the manufacturer.

This could potentially lead to information disclosure or further exploitation if sensitive data is obtained from the I

/nos\_parsing\_user\_data\_registered\_itu\_t\_t35 function of VendorVideoAPI.cpp without requiring additional user i

nds memory. The attacker would leverage the overflow to potentially access sensitive data or system properties v  
ffer overflow, potentially leading to remote information disclosure without requiring any additional execution pri

er could potentially access sensitive information without any action from the user.

er overflow. Developers should ensure they correctly handle buffer lengths and protect against potential overflow v  
to read sensitive information from memory. Since no user interaction is required, this could be exploited remote

ential for local privilege escalation without the need for any additional execution privileges.

on protocols. This could allow the attacker to gain elevated privileges on the device without requiring any further

: to interfere with the file encryption process, leading to information disclosure without the victim's knowledge. S

ndition might occur, but without access to the proprietary or internal source code, we can't provide a specific code. The user is interacting with due to a missing permission check. It affects Android version 13 and does not require user i  
privileges.

ow the attacker to learn which application the user is currently interacting with, potentially divulging sensitive info

ice without requiring user interaction. To exploit the vulnerability, an attacker would need System execution privi  
d by the attacker, potentially disabling user credentials.

S.

to a denial of service. This could render the device unusable for the legitimate user without directly stealing data  
permissions and enable or disable mobile data, which could lead to local escalation of privilege without needing a  
gations as soon as possible to prevent potential abuse.

are dangerous and easier to exploit.

y, as well as any updates or patch information provided by the vendor.

ary onCreate method to enable or disable mobile data without authorization. This could disrupt connectivity for l  
arded by permissions. In this case, it's specifically the ability to toggle mobile data on or off.

aking place to prevent unauthorized access to mobile data settings. However, developers and security researcher  
of privilege without needing additional execution privileges.

nstructing a malicious application or script that interacts with the settings to alter data limits, leading to privilege e  
stall or run unauthorized applications, and monitor for unusual activity indicating privilege escalation attempts.

work activities, leading to local information disclosure without requiring any additional execution privileges. No us

ormation about the network activities of an admin user on the device, without requiring further permissions or user interaction and does not require additional execution privileges and does not require user interaction.

If user interaction is needed, the attacker could silently alter the settings to reroute calls or manipulate call behavior.

This vulnerability could lead to local escalation of privilege, and it requires System execution privileges to exploit. It

allows an attacker to perform unauthorized click events. This could enable click fraud or other malicious activities without user interaction and local information disclosure if exploited.

CommandEngine.h to gain access to sensitive information without the user's knowledge. This vulnerability involves a buffer overflow in the processing of an allocated buffer. Fixing such vulnerabilities often involves adding proper bounds checks to ensure the proper handling of input data. Information disclosure, and to exploit it, an attacker would need System execution privileges. User interaction is not required.

privileged process. By invoking the inviteInternal function of p2p\_iface.cpp without proper bounds checking, they could trigger a buffer overflow. To address this vulnerability, one would typically analyze the affected component's source code and review the patch that fixes it.

, potentially exposing sensitive information.

SystemUI menu due to missing bounds checks, allowing a local attacker to crash the SystemUI without additional exe

manically invoke the vulnerable setProfileName method with improper bounds, causing the SystemUI to crash. "

he data parsing without proper bounds checking, the attacker could write data outside the expected memory bou

ation disclosure without the need for additional execution privileges.

system's memory without the user's knowledge.

nd has been rated with a medium severity score of 6.7.

ed to exploit the vulnerability.

issues for detailed affected version information.

/pixel/2023-06-01

d result in corruption of data, crash the system, or in more severe cases, the attacker could execute arbitrary code  
bounds read. This issue could be exploited to disclose local information and requires System execution privileges  
versions, one would need to refer to official Android security bulletins or the vendor's advisory.

lications or the OS itself.

overflow vulnerability in the 'encode' function of 'miscdata.cpp' to read out-of-bounds memory locations. This cc

ttacker would need System execution privileges. User interaction is not necessary for the exploitation of this flaw

rruption, which the attacker might exploit to execute arbitrary code with elevated privileges on the affected device.  
/ A simulated overflow where we write 15 'A' characters into a buffer that has space for only 10 characters. mems  
: validation that could lead to local information disclosure from the modem, requiring system execution privileges

user's knowledge.

est version containing the fix for this vulnerability.

s on the vulnerability, affected versions, and the corresponding fixes.

i missing null check. This issue could potentially lead to local information disclosure without requiring any additional

, meaning that sensitive information from the affected device could be unintentionally accessed or leaked to an attacker.

:RadioNode function. Since there's a missing null check, this input could trick the system into reading data past the

ad to local information disclosure.

tem itself, to exploit the issue.

d information.

l as higher-scoring vulnerabilities.

inds read vulnerability to access sensitive data in memory that it's not supposed to, leading to information disclosure  
by a missing bounds check that could potentially allow for local escalation of privilege with System execution privileges.  
i out of bounds read to occur, which may be used to escalate privileges without any need for user interaction.  
Android device. This could compromise the integrity and confidentiality of the system and lead to unauthorized access  
cription.

n in imsservice.cpp without the correct bounds check, leading to out of bounds memory access. This could be harmful  
sing bounds check that could potentially lead to local information disclosure, requiring User execution privileges for  
al exploitation.

[r/security/bulletin/pixel/2023-06-01](#).

eeding any further interaction from the user, the malicious entity could exploit the out of bounds read vulnerabili

ght result in a local escalation of privilege. An attacker with System execution privileges could potentially exploit  
mediation efforts from affected parties.

ices.

is.

ulnerability and information on other Android security issues as well.

entation instructions for CVE-2023-21151.

uch a way as to cause a heap buffer overflow, writing data beyond the boundaries of the buffer. This could corrup  
ically shared with device manufacturers and developers directly to help them patch the issue before wider exploit

ntially lead to local information disclosure.

ion. Since no user interaction is required, the application could silently gather information and possibly transmit it

1 check, which could allow for a local escalation of privilege without requiring additional execution privileges. The

out the rightful user's knowledge or permission. This could lead to unauthorized access or control of communicat  
atch notes might offer technical insights into the vulnerability without revealing exploitable code.

ocated structure. This could lead to information disclosure. An example code detailing this vulnerability is not prov

-06-01

ticular version, as they may impact multiple versions. Users should check Android's security bulletins or updates f

use (UAF) vulnerability in the `lwis_i2c_device_disable` function to destabilize or take control of the kernel, leading to a successful exploit. The linked URL, <https://source.android.com/security/bulletin/pixel/2023-06-01>, should provide more details on the security issue.

exploit in the Android kernel to corrupt memory and potentially gain System execution privileges without requiring an additional privilege. This exploit allows for arbitrary code execution without the need for additional privileges, and user interaction is not necessary for exploitation.

execute arbitrary code without requiring any additional privileges or user interaction.

As required, this input could be delivered through various channels such as a malicious app download, a compromised device, or a network-based attack. This exploit allows for arbitrary code execution without requiring additional execution privileges, and it does not necessitate user interaction for exploitation.

the user interface, they might inadvertently grant permissions or execute actions that lead to privilege escalation for the application. This exploit allows for arbitrary code execution without needing any additional execution rights, potentially leading to unauthorized access to sensitive data. The exploit would detect user input and simulate clicks on the underlying UI, tricking the user into granting unintended permissions.

Android. This activity accepts a `data: URI`, which can be used to load arbitrary content into the application's webview.

website (<http://packetstormsecurity.com/files/172701/FC-Red-Bull-Salzburg-App-5.1.9-R-Improper-Authorization>) to load arbitrary content into the application's webview. This content could potentially execute scripts or access sensitive data within the context of the application. The exploit object would be created with action `'android.intent.action.VIEW'` and data set to the URI containing the malicious content.

malicious applications using a custom-crafted deeplink scheme. This could lead to sensitive information disclosure.

the application to redirect server responses to a third-party application controlled by the attacker. This could potentially expose sensitive information to the attacker.

and gain unauthorized access to sensitive information.



perform actions with the same privileges as a legitimate user of the application, potentially leading to data breaches or password within the code, something like 'String password = "hardcodedpassword";' Hard-coded credentials in to any systems or services where these credentials might provide access. This could lead to unauthorized data and to the leakage of sensitive information. The vulnerability is present in Firefox for Android versions prior to 112 browsing sessions.

During screen recording sessions in Private Browsing mode, ensuring that sensitive information is not inadvertently leaked during screen recording in Private Browsing mode on affected versions of Firefox for Android.

Address bar and keyboard are not hidden during the recording, the attacker could gain access to the recording file and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1780842](https://bugzilla.mozilla.org/show_bug.cgi?id=1780842).

The vulnerability was a flaw in the browser's functionality rather than something that could be exploited through code by session and potential spoofing attacks, where an attacker might mimic the interface or prompts of trusted sites or a

reports on Mozilla's Bugzilla website, such as '[https://bugzilla.mozilla.org/show\\_bug.cgi?id=1816059](https://bugzilla.mozilla.org/show_bug.cgi?id=1816059)'.

When the app requests to go fullscreen, the notification could be made less visible or hidden, tricking the user into believing the app is in fullscreen. The issue only affects Firefox for Android, with other versions of Firefox being unaffected. The issue affects Android API levels below version 30. The issue only affects Firefox for Android, with other versions of Firefox being unaffected. The issue affects Android API levels below version 30. The issue only affects Firefox for Android, with other versions of Firefox being unaffected.

The after-free condition is prevented, thereby eliminating the associated risk.

entry ([https://bugzilla.mozilla.org/show\\_bug.cgi?id=1815801](https://bugzilla.mozilla.org/show_bug.cgi?id=1815801)).

The AACAudio backend, could lead to arbitrary code execution. This could result in the compromise of the Firefox for Android. The AACAudio component of Firefox for Android, and those interested in understanding the technical details can find more information in the libaudio component of Firefox for Android, and those interested in understanding the technical details can find more information in the libaudio component of Firefox for Android. This could potentially lead to a denial of service, or availability of the affected systems.

4, which should provide further details and guidance.

Under the device unusable or compromise user data and disrupt device functionality.

The app could wipe sensitive personal information or uninstall critical applications without the user's consent, resulting in a denial of service. This could lead to remote denial of service without any additional privileges needed, and exploitation does not require a

actions in the doInBackground method. This can lead to a denial of service condition on the victim's device by overw

The changes made to prevent the vulnerability.

The issue was caused by improper input validation which could potentially allow a local attacker to cause a denial of service (DoS). The issue presents a significant risk, but may not be as critical as vulnerabilities with higher scores.

The issue was addressed by the relevant security updates.

or detailed information and guidance on remediation.

, the malicious code could automatically trigger the denial of service condition, causing the device to become unresponsive. An attacker could potentially automate the attack process. However, the specifics of the attack vector are not detailed.

It traces without needing user interaction or additional privileges. Since the vulnerability is in developer mode, an

exploit without any additional execution privileges. No user interaction is required to exploit this flaw. Given the potential consequences, it still warrants attention and timely remediation.

The traces, potentially exposing sensitive or proprietary information stored on the device.

Latest security patches to protect against known vulnerabilities.

Exposure of sensitive information without the user's knowledge, such as debugging data that may include personally identifiable information in the System UI due to unsafe handling of Intents, which can lead to local escalation of privilege without need for administrator privileges.

For more information on the patch: <https://source.android.com/security/bulletin/2023-06-01>.

Activity without user consent. Since the exploit does not require additional privileges or user interaction, it could

long-lived connection. This vulnerability has the potential to lead to local escalation of privilege and could allow bypassing

Edge or consent of the device owner. This could compromise the security and privacy of the user's data and allow an attacker to use specially crafted inputs that bypass improper validation checks. This could allow the app to maintain a long-lived connection

This vulnerability affects the system without requiring any additional execution privileges and user interaction isn't necessary.

While it does not necessarily represent an immediate critical threat to most systems.

Since it could be exploited without any additional execution privileges or user interaction, the app could cause the device to become unresponsive on startup due to improper input validation. This vulnerability affects Android versions 11, 12, 12L, and 13.

on startup, therefore causing a local denial of service. For example, an attacker might introduce malformed data into the system, causing a crash or a denial of service. This could be achieved by an attacker to achieve local escalation of privilege without requiring additional execution privileges.

This app could be designed to trigger the vulnerability in the background without the user's knowledge, resulting in a denial of service. This could potentially allow an attacker to bypass Parcel Mismatch mitigations, leading to local escalation of privilege and exploitation.

the data it contains. This could be achieved by an attacker to achieve local escalation of privilege without requiring additional execution privileges. Since the exploitation does not require additional permissions, it could potentially allow an attacker to bypass Parcel Mismatch mitigations, leading to local escalation of privilege and exploitation. The vulnerable systems.

in the affected component on a device running Android 13. This overflow could then be leveraged to execute arbitrary code on the device.

roadcast Action Listener (BAL) bypass.

then potentially launch an activity in the foreground, gaining visibility or interaction that it would not typically be able to.

tion privileges and does not require user interaction for exploitation.

patch notes related to the CVE-2023-21128.

s, potentially leading to data theft, unauthorized changes to system settings, installing persistent malware, or con

ith no additional execution privileges needed. However, exploitation requires user interaction.

n the user's device without requiring additional privileges. The exploitation relies on user interaction, such as the

/a component of the System UI. An adversary could exploit this vulnerability to launch arbitrary activities without

late or limit the incoming Intent, the malicious application can specify an arbitrary component to be started. As a  
e latest security patches to protect against exploitation of this vulnerability.

e and the privacy of its data.

cherAndBroadcastButton` method within `MediaControlPanel.java`. A code fix would involve validating the inten  
safe deserialization, which could lead to local escalation of privilege without requiring additional execution privile

by sending a crafted input that, when processed by the Android system, leads to deserialization of untrusted data  
d Security Bulletins can help users know when and how to update their systems to mitigate vulnerabilities.

pecific code examples of vulnerabilities are sensitive and vary widely, it is best practice not to share them publicly to  
apps, or perform other unauthorized actions on a device without requiring any user interaction. This could lead to  
riction when tracing due to an omitted permission check. This could lead to local escalation of privilege without n

operations. Without needing to elevate their execution privileges or require user interaction, an attacker could leve

ld allow the attacker to maintain access to the user's network traffic or redirect the user through a harmful proxy

app could escalate privileges in the background without the user's knowledge, potentially allowing the attacker to  
s based on the bulletin, and users should ensure their devices are up-to-date with the latest security updates.

y could lead to an escalation of privilege on a paired device without requiring additional execution privileges. No  
their privileges on the targeted device, potentially leading to unauthorized access or control over the device.

otes to understand the nature of the vulnerability and apply necessary updates or mitigations.

critical for safeguarding devices against such vulnerabilities.

illy lead to remote code execution (RCE) over Bluetooth. This vulnerability is especially critical as it doesn't require

both HFP support is enabled on the device.

hey could then proceed to send specially crafted Bluetooth packets to trigger the use after free vulnerability in the  
al information disclosure without needing additional execution privileges. No user interaction is required for the  
ential information disclosure.

on. Since no user interaction is required, the malicious app could silently access sensitive information, leading to a  
lettin/2023-06-01

essful attack.

n privileges.

context of a higher-privileged process, thereby escalating their privileges on the system. This could be used to gain a local escalation of privilege without requiring additional execution privileges, and does not necessitate user interaction.

Vulnerability can be exploited without user interaction and does not require additional privileges, an attacker with persistence. The exact details of such vulnerabilities are usually managed discreetly between security researchers and vendors.

public static void main(String[] args) throws Exception { // Creating a FileBackedOutputStream with default temporary directory. An attacker with the right permissions could read, modify, or even delete the content of these files, as improper server certificate verification. This weakness could potentially allow an attacker to perform a man-in-the-middle attack.

potentially lead to the interception and disclosure of sensitive information transmitted during the app's operation. Vulnerability Notes.

Identified security certificate that the vulnerable app fails to validate properly. If the app accepts the fraudulent certificate, it may lead to the interception and disclosure of sensitive information transmitted during the app's operation. ServiceConnection conn = new ServiceConnection() { ... public void onServiceConnected(ComponentName name, Intent service) { ... } } Preference files.

Impact its functionality.

1. This could lead to the app crashing or becoming unresponsive, thereby denying service to the legitimate user. 2. It may lead to the disclosure of sensitive information stored in the app's Preference files of the app.

3. It may lead to the disclosure of sensitive information stored in the app's Preference files of the app.

4. It may lead to the disclosure of sensitive information stored in the app's Preference files of the app, affecting the usability of the application for the user until the issue is resolved.

could be done by the rogue app injecting corrupt or malicious data into the SharedPreferences files, which may cause

them or causing them to become unreadable. As a result, the application would be unable to access the necessary permissions or insecure file operations. For instance, if the app has a function that does not correctly validate user input,

change these files to elevate their privileges within the system. For example, they could change database entries to grant an Android device that requires higher privileges.

change settings or corrupt the files in a way that makes the Sleep app unusable, leading to a persistent denial of service. The following code snippet shows how to edit the SharedPreferences file for the Sleep app:

```
SharedPreferences prefs = getSharedPreferences("com.example.sleep", MODE_PRIVATE);
SharedPreferences.Editor editor = prefs.edit();
editor.putString("setting", "value");
editor.apply();
```

The app can no longer function properly, leading to a persistent denial of service for the user, preventing them from using the device.

The Light Filter app to fail or become unusable, leading to a persistent denial of service condition on the device.

The app becomes vulnerable to this denial of service attack.

Therefore, no specific code can be provided without more details on how the SharedPreferences files are used in the Blue Light Filter app.

The app can change system settings, leading to a compromised device.

The app can install applications, or manipulate system settings.

SharedPreferences files. For instance, it might alter preferences to grant itself higher permissions or disable security checks, thus allowing these files to be manipulated by unauthorized apps, allowing them to perform an escalation of privileges. Developers should ensure their apps correctly implement security best practices for handling SharedPreferences data to prevent unauthorized access and manipulation of these files, and stay updated with the latest security measures.

e files of Twilight, and then make unauthorized changes to those files. As a result, Twilight might experience persi  
s.

erence files. Since SharedPreferences is a common way to store private data, such as user settings and credentials,  
: Malicious code attempting to modify SharedPreferences context = victimAppContext;SharedPreferences |  
ntially lead to a compromised security posture on the affected device.

ce.

anges at a higher privilege level without user consent, potentially leading to data theft, account compromise, or fi  
with the SharedPreferences files of the app.

ce files of Yandex Navigator, potentially gaining access to sensitive user data or system functionality that should b  
or unauthorized access to restricted parts of the device. Users running Yandex Navigator version 6.60 on their An  
id application. The exploit code would use Android APIs to access and modify the preferences of Yandex Navigato  
er apps and install applications from trusted sources only to prevent exploitation of such vulnerabilities.

duct-security/bulletins/june-2023-bulletin.

```
#include <stdlib.h>void unregister_provider(char *provider) { // Incorrectly freeing memory twice free(provider); //
```

an lead to outcomes such as denial of service (DoS) through system crashes or potentially allow the attacker to e

arbitrary code on the affected system.

.mozilla.org/buglist.cgi?bug\_id=1763625%2C1814314%2C1815798%2C1815890%2C1819239%2C1819465%2C181  
bugs to cause memory corruption. This corruption could then potentially be leveraged to execute arbitrary code c  
ails or code that could directly be used to exploit a vulnerability. Instead, security advisories and bug reports will c

illa.mozilla.org/buglist.cgi?bug\_id=1720594%2C1812498%2C1814217%2C1818357%2C1751945%2C1818762%2C



could lead to memory corruption and potentially allow the attacker to execute arbitrary code on the victim's machine. This vulnerability was discovered in Firefox versions 102.0 and older than 102.10.

[bugzilla.mozilla.org/show\\_bug.cgi?id=1823042](https://bugzilla.mozilla.org/show_bug.cgi?id=1823042).

When a script is bound to a certain context, under some conditions the context (or 'realm') may not be set as intended, potentially allowing an attacker to execute arbitrary code within the context of another realm, leading to information disclosure.

older than 102.10.

[www.mozilla.org/security/advisories/mfsa2023-15/](https://www.mozilla.org/security/advisories/mfsa2023-15/)

exploitable conditions.

When processed by a vulnerable version of Firefox or other affected Mozilla products, it could cause the application to behave unpredictably, affecting the confidentiality and integrity of cookie-based information.

e, affecting the confidentiality and integrity of cookie-based information.

[bugzilla.mozilla.org/show\\_bug.cgi?id=1783536](https://bugzilla.mozilla.org/show_bug.cgi?id=1783536).

This vulnerability involves insecure cookies. For instance, an attacker might use insecure cookies to perform session hijacking, track a user's activity, or manipulate cookie storage and security checks. No specific code example is available as it involves the browser's internal state and a potentially exploitable crash.

[www.mozilla.org/show\\_bug.cgi?id=1818781](https://www.mozilla.org/show_bug.cgi?id=1818781)).

This vulnerability could lead to memory corruption. This scenario could be facilitated through crafted web content that, when processed by the browser, triggers a memory corruption. Users are advised to apply updates as soon as possible to prevent exploitation.

This vulnerability was discovered in Firefox versions 102.0 and older than 102.10. It was assigned a high severity base score of 8.8 due to the potential for information disclosure and the ease of exploitation. Mozilla is working on a patch to mitigate the risk associated with this vulnerability.

[www.mozilla.org/security/advisories/mfsa2023-13/](https://www.mozilla.org/security/advisories/mfsa2023-13/).

This vulnerability could potentially allow the attacker to gain arbitrary code execution within the context of the browser. Users are advised to apply updates as soon as possible to address known security vulnerabilities.

This vulnerability could be exploited to execute arbitrary code. This is a concern because .desktop files in Linux can specify an application to run, and if an attacker can execute arbitrary code, they could potentially execute any application. This vulnerability was discovered in Firefox versions 102.0 and older than 102.10, and Thunderbird releases before 102.10.

The provided URLs will take you to the respective pages which provide detailed information about the vulnerability.

ing the file as a legitimate download. Once downloaded, if the Firefox browser on the vulnerable Linux system executes the misuse of `sourceMappingUrls` that could lead to navigation to external protocol links in sandboxed iframes, it would load external protocol links without the user's consent or knowledge, which could be used in phishing attacks. Users should ensure that their Firefox for Android, Firefox, and Focus for Android are updated to a secure version to prevent exploitation. For more information, see the security advisory at <https://www.mozilla.org/security/advisories/mfsa2023-13/> and the corresponding Bugzilla report at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784348](https://bugzilla.mozilla.org/show_bug.cgi?id=1784348). Due to the complexity of the attack, the potential impact on confidentiality, integrity, and availability, and the need for user intervention to navigate the iframe to an external protocol link, such as a custom scheme handler that triggers unwanted or harmful behavior.

Thunderbird versions prior to 102.10.  
This could lead to users being misled into installing malware.

See <https://www.mozilla.org/security/advisories/mfsa2023-13/>, [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784348](https://bugzilla.mozilla.org/show_bug.cgi?id=1784348), and <https://www.mozilla.org/security/advisories/mfsa2023-13/>. A browser would incorrectly process the filename due to the vulnerability, potentially causing the file to have a different name, such as "safe\_document.pdf\x00.exe". When a user tries to download the 'safe\_document.pdf', the browser affected by this vulnerability would attempt to save the file to the directory paths on the user's machine.

See the Security Advisory at <https://www.mozilla.org/security/advisories/mfsa2023-13/>. This vulnerability affects the form of `jar:file:///` URIs. An attacker could capture these paths and gain information about the directory structure of the system. Keeping software up to the latest versions is a best practice for maintaining security. This vulnerability could be used to execute attacker-controlled code.

See [https://www.mozilla.org/show\\_bug.cgi?id=1823365](https://www.mozilla.org/show_bug.cgi?id=1823365) and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1825569](https://bugzilla.mozilla.org/show_bug.cgi?id=1825569). If a user views the content using vulnerable versions of these applications, the attacker could potentially trigger memory corruption. This could lead to a crash or a memory manager which could allow an attacker to cause incorrect freeing of a pointer that addresses attacker-controlled memory. This vulnerability affects versions before 102.10).

See <https://www.mozilla.org/security/advisories/mfsa2023-14/> and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1823365](https://bugzilla.mozilla.org/show_bug.cgi?id=1823365). Through an affected Mozilla product, the attacker could manipulate the memory in a way that improperly frees a memory address. This vulnerability affects fixed versions are Firefox 112, Focus for Android 112, Firefox ESR 102.10, Firefox for Android 112, and Thunderbird 115.0.2.

See <https://www.mozilla.org/security/advisories/mfsa2023-15/> and [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1820543](https://bugzilla.mozilla.org/show_bug.cgi?id=1820543). This vulnerability affects versions before 115.0.2.

Weak maps leads to memory corruption. If a user visits this page using a vulnerable browser version, the attacker could exploit the memory corruption to execute arbitrary code.

or a website to obscure the fullscreen notification, utilizing a combination of `window.open`, fullscreen requests, and the `confirm` method. This could lead to users being unaware of the fullscreen mode, potentially leading to phishing attacks or exposure to malicious content without their awareness that the context has changed, thus compromising the confidentiality, integrity, or availability of the affected system.

https://www.mozilla.org/security/advisories/mfsa2023-14/- https://www.mozilla.org/security/advisories/mfsa2023-14/- val` to frequently modify the `window.name` property, which could trick the user into not realizing they are in fullscreen mode, due to the fullscreen notification being obscured. In fullscreen mode, the website could then use the `confirm` method to prompt the user for confirmation before performing any actions, potentially leading to phishing attacks or exposure to malicious content without their awareness that the context has changed, thus compromising the confidentiality, integrity, or availability of the affected system.

ing unaware they are in fullscreen mode, which the attacker could then use to emulate the browser UI or show developers to these vulnerabilities. Firefox introduced a fix in which the browser will now confirm with users before launching a new window or tab.

ies through an Intent. Since the affected Firefox versions would not have prompted the user for confirmation before displaying a new page or element, this could lead to unintended actions or compromise security. This issue affected versions of Firefox for Android prior to version 111.

https://bugzilla.mozilla.org/show\_bug.cgi?id=1810705.

with a different page or element than they actually are, which could lead to unintended actions or compromise security. This issue affected versions of Firefox for Android prior to version 111.

versions. Additionally, the bug report on Mozilla's Bugzilla tracker can be found at 'https://bugzilla.mozilla.org/show\_bug.cgi?id=1787034'. An attacker designing a web page that triggers a fullscreen notification followed by a prompt with a long description. This could be displayed during different browsing sessions, potentially leading to privacy and security issues.

https://bugzilla.mozilla.org/show\_bug.cgi?id=1787034.

phishing attacks or expose the user to malicious content without their awareness that the context has changed, thus compromising the confidentiality, integrity, or availability of the affected system.

the confidentiality, integrity, or availability of the affected system.

https://apksos.com/app/story.saver.downloader.photo.video.repost.byrk), a detailed CVE report on GitHub (https://github.com/0x00sec/cve-2023-38411), and a remote code execution, allowing the attacker to perform actions as if they were the user or to gain unauthorized access to the target app's SharedPreference files. Since SharedPreference is a common Android API for storing application preferences. This injected data can force the application to load malicious image URLs, display them in the UI, and potentially lead to phishing attacks or exposure to malicious content without their awareness that the context has changed, thus compromising the confidentiality, integrity, or availability of the affected system.

triggering an OOM error and crashing.

to the app's database. The attacker would then inject large amounts of data, including malicious image URLs, which can be found in the repository <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29725/CVE%20detail.md>, the Google Play Store listing of a related app (<https://play.google.com/store/apps/details?id=com.liankee.keyboardthemes>).

When the app is opened, an attacker could exploit this to tamper with the data and potentially perform an escalation of privileges.

For example, an attacker could inject a large amount of data into the database, which could cause the app to run out of memory and crash. This could lead to unauthorized actions being performed on the device with more privileges than the attacker would typically have access to. An attacker might be using a compromised app to inject malicious code or commands into the database data, which could lead to a denial of service attack. Developers should also be cautious of granting permissions to apps and avoid installing apps from unknown or untrusted sources. Additionally, developers should use content providers with proper access controls, and employing runtime permission checks. Additionally, developers should be cautious of storing sensitive data in the app's internal storage, which can lead to arbitrary code execution. This could lead to a compromise of confidentiality, integrity, and availability with little to no user interaction.

The ability to navigate the file system and overwrite sensitive files within the Keyboard Themes app's internal storage. Because of the ability to perform unauthorized actions, such as data exfiltration, installation of additional malware, or compromise of the system, the possibility of loading malicious image URLs in the UI, which can cause an Out-Of-Memory (OOM) error and a crash.

Images that, when loaded by the Glitter Unicorn Wallpaper app, cause it to run out of memory and crash. For example, the app's database contains a large amount of data, which can cause the app to run out of memory and crash.

When the app is opened, it would load the data that stores users' personal preferences. When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences. When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences.

The database that records information about a user's personal preferences, which is supposed to be loaded into memory. The database that records information about a user's personal preferences, which is supposed to be loaded into memory. The database that records information about a user's personal preferences, which is supposed to be loaded into memory.

When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences. When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences. When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences.

When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences. When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences. When the Glitter Unicorn Wallpaper app is opened, it would load the data that stores users' personal preferences.

Repository dedicated to the vulnerability. The links can be found in the references of the CVE.

The repository dedicated to the vulnerability. The links can be found in the references of the CVE. The repository dedicated to the vulnerability. The links can be found in the references of the CVE. The repository dedicated to the vulnerability. The links can be found in the references of the CVE. The repository dedicated to the vulnerability. The links can be found in the references of the CVE.

exceptions in ad display or other unexpected behavior depending on how the modified data is used.

<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29747/CVE%20detail.md>

method in an Activity or other component public void updateSharedPreferences(String key, String value) { SharedPreferences  
interfering with the application's database.

Is about related apps can be found on their respective Google Play Store pages: - <https://play.google.com/store/apps/details?id=com.bestweather.android>  
- the BestWeather app unusable, effectively conducting a persistent denial of service attack, as the app would crash. The malicious app, by requesting permissions, accesses the data storage location of the BestWeather app. The malicious app might use SQL injection

<https://play.google.com/store/apps/details?id=com.icoolme.android.weather>

to compromising the integrity of the app, leaking sensitive information, or providing a gateway for further attacks or unauthorized access. Developers should look to the app's database access functions and ensure they validate and sanitize user input. Additionally, users should be cautious about downloading and granting permissions to other apps that could potentially access the app's data. For more information, see the app's details page at <https://play.google.com/store/apps/details?id=com.icoolme.android.weather> or reach out to the app developer.

users from accessing the functionality provided by the application.

<https://play.google.com/store/apps/details?id=com.icoolme.android.weather>

Weather app. This could involve inserting, modifying, or deleting records in such a way that the app fails to function properly, overwhelming the app's resources or corrupting data necessary for its operation. This could potentially compromise the application's database.

A hacker could manipulate the database to gain access to functionality reserved for higher-privileged users, potentially leading to unauthorized access to sensitive data. For more information, see the app's details page at <https://play.google.com/store/apps/details?id=com.icoolme.android.weather>, and a detailed breakdown on GitHub at <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29747/CVE%20detail.md>

by sending specially crafted intents or data to the Alarm Clock app, which could potentially corrupt or manipulate the app's data.

<https://play.google.com/store/apps/details?id=com.android.alarmclock>

originally granted, potentially allowing it to access sensitive user data, alter system settings, or interfere with the mechanisms, and potentially even taking full control of the affected device.  
It is so advisable to review app permissions and avoid installing apps from untrusted sources.

Android - <https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29738/CVE%20detail.md> - <http://www.vulnerability-lab.com/vulnerability.php?ID=VLAB-0144>  
manipulating the database files of the app to execute arbitrary code with elevated privileges. This could potentially allow an attacker to gain access to sensitive data on the system, such as via an already compromised application on the device, to exploit the vulnerability. This is a privilege attack. This vulnerability has been assessed with a base score of 9.8, indicating it's critical in terms of severity and could be used to perform unauthorized actions on the device, compromise user privacy, or interfere with the normal functioning of the device.

If there are any security advisories released for the app or contact the developers directly for guidance on mitigation. Visit the Call Blocker app's page on Google Play Store: <https://play.google.com/store/apps/details?id=com.cuie.callblocker>  
After the data associated with the app's features, such as changing the rules for blocking calls, altering permissions, etc. Researchers often analyze such vulnerabilities and may publish proof of concept codes in security advisories or repositories. This could lead to unauthorized access to user privacy settings, leading to potential escalation of privilege attacks.

Visit the repository with details about the CVE (<https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29727/CVE%20detail.md>)  
This vulnerability could be used to compromise user privacy settings and disrupt the app's functionality. In a targeted attack, an adversary could leverage this vulnerability to gain access to sensitive data stored in the application's database. This could lead to unauthorized access to user privacy settings, leading to potential escalation of privilege attacks.  
Broadcast Receivers or Content Providers of Call Blocker without the proper permissions, using intents or queries to access sensitive data stored in the application's database. This leads to an Out of Memory (OOM) error and crashes the app, causing a denial of service (DoS) attack.

Unauthorized access to its database or a key component, and using this access point to insert malicious data. As the app launches, it may cause a memory overload from the injected data.  
Visit the Google Play Store link for the Call Blocker application: <https://play.google.com/store/apps/details?id=com.cuie.callblocker>

base to disrupt normal application functioning.

[23-29735/CVE%20detail.md](https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29735/CVE%20detail.md).

This vulnerability could be used to compromise the application's database files by injecting malformed data or by causing the files to become unusable. This manipulation of the application's database, potentially leading to unauthorized actions within the device's operating system could be used to gain access to sensitive data stored in the application's database.

In the system by conducting this exploitation, thereby gaining access to functionalities or data that are normally protected. This could lead to unauthorized access to sensitive data stored in the application's database, potentially leading to unauthorized actions within the device's operating system or a denial of service (DoS) attack against unauthorized access or manipulation. It is also essential for users to regularly review app permissions and avoid installing apps from untrusted sources.

f privilege attack.

es of Lock Master without proper authorization. By altering certain values in these files, the attacker could manip

ponent's method that is made accessible. When the application is launched, the compromised data from these fi

unauthorized ad modifications leading to revenue loss, presentation of malicious advertisements that could dece  
edPreference file. An attacker can exploit this to inject a large amount of data, leading to an Out of Memory (OOM  
too much data is inserted, the application will encounter an OOM error upon opening, leading to a crash. This for  
sing the app until the problem is resolved, which might necessitate clearing the application data or even reinstalli

bitrary data into the SharedPreferences file of the SoLive application. When the user attempts to open the SoLive

ve, thereby inducing a denial of service condition for users of the affected application version.

ccessed by the 'insert' function, triggers an exception or resource exhaustion, resulting in the application's denial

3/SO-CVEs/blob/main/CVEs/CVE-2023-29737/CVE%20detail.md

denial of service for the user. The attacker would need to have access to the device where the app is installed, eith  
malicious activities. However, developers should examine the handling of their database files and ensure that pro  
orarily by rebooting the device into Android Safe Mode and disabling the 'Display over other apps' permission, wit

/ rebooting the device into Safe Mode. Safe Mode disables many third-party apps and services, including Kids Plac  
nitor their device security settings and consider using additional measures to prevent children from booting into !  
ne device to force Omni-notes to move files from its internal storage to its external storage directory, making the  
e concern and should be addressed, but it is not as critical as more severe vulnerabilities.

path validation vulnerability.

iHSA-g38r-4cf6-3v32.

lation, Omni-notes might proceed to copy files from its internal storage to an external storage directory. Once in t

i64 Android environment. In such cases, the 'CARES\_RANDOM\_FILE' is not set, leading the system to fall back on t

the c-ares release page at '[https://github.com/c-ares/c-ares/releases/tag/cares-1\\_19\\_1](https://github.com/c-ares/c-ares/releases/tag/cares-1_19_1)', Fedora's package anno  
DOM\_FILE'. This could be exploited in various ways depending on the use of the library in an application. For insta  
to the unavailability of 'CARES\_RANDOM\_FILE' can make the system vulnerable to attacks where the predictabilit  
e's a simplified and hypothetical example: ``c// Hypothetical insecure c-ares code using rand() instead of CSPRNG (

e improper storage of sensitive information in cleartext, either within a file or on disk.

ncrypted before being stored and use parameterized queries to interact with the database securely.

sensitive information stored insecurely in the SQLite Database used by the Simple Design Daily Journal app. This i

ndroid device to display settings and log information of the Brother iPrint&Scan app as a print preview without pr

etail/a\_id/13468- <https://play.google.com/store/apps/details?id=com.brother.mfc.brprint>- <https://jvn.jp/en/vu/J>

us app could exploit the improper access control vulnerability to display settings or log information from the Brot  
the Brother iPrint&Scan app, but the exact details and method of exploitation are not disclosed to prevent furthe  
ption through a specially crafted HTML page. It affects versions of Chrome prior to 113.0.5672.126.

og.com/2023/05/stable-channel-update-for-desktop\_16.html, and advisories from various Linux distributions like  
n a user visits this page, the attacker might exploit heap corruption to execute arbitrary code or cause a denial of s  
ilities, often involving manipulating the lifecycle of objects so that an attacker's code interacts with a freed memo

attenString8 function of Sensor.cpp to perform an out of bounds read. Since no additional execution privileges ar  
register a broadcast receiver due to a permissions bypass. This issue could lead to a local escalation of privilege v



that could compromise the security of the affected system.

It can be exploited passively.

mitigate the risks associated with these issues.

on the vulnerability, affected versions, and any available patches or workarounds.

potentially intercept sensitive system broadcasts or act with escalated privileges, which it should not normally have. 'IceServiceManager.java' where certain permission checks are inadequately enforced. Ideally, the code fix would involve tightening checks to downgrade system apps below the system image version, leading to a local escalation of privilege with system apps receiving higher severity scores.

exploit this specific vulnerability to downgrade a system application to an older, more vulnerable version without the need for user interaction. With these privileges, the attacker can exploit the logic error in `InstallPackageHelper.java` to downgrade a system app, which could lead to local information disclosure without requiring additional execution privileges.

No additional privileges or user interaction.

on and access information that should normally be restricted. This can result in the disclosure of sensitive data on the device. This vulnerability in the Android operating system is crucial to mitigate the risks associated with this and other vulnerabilities.

It could potentially allow an attacker to create a local denial of service condition, specifically preventing access to emergency services.

Emergency services on the affected device. No additional execution privileges are required, and user interaction is not required.

available for their version of Android.

The exact details of the vulnerability are kept confidential to prevent exploitation, and the fix would be implemented through a security update. The vulnerability affects notification rights through a possible exploitation of notifications access due to resource exhaustion. It affects Android versions 6.0.1 through 7.1.2.

Resources which can be inadvertently consumed by an attacker, leading to a denial of service or privilege escalation. For example, an attacker could create an excessive number of notifications or invoke particular methods in `SnoozeHelper.java` to trigger the resource exhaustion.

This section includes details about the issue and potentially any remedies or patches that have been released.

It could lead to local escalation of privilege without requiring any additional execution privileges. The vulnerability affects system apps and does not require complex or sophisticated methods for exploitation.

User interaction is not needed, the attacker might be able to perform actions with elevated permissions without the user's consent.

as needed. This issue affects Android versions 11, 12, 12L, and 13.

ing to privilege escalation on the device without user interaction.

to gain higher privileges on the device. This could potentially allow the attacker to access or modify protected da

ily local escalation of privilege without any additional execution privileges required. User interaction is not necess

version numbers that are affected.

The attacker could corrupt memory and escalate their privileges without needing additional execution privileges publicly to prevent their misuse. However, researchers and developers can review the changes made in the upst out requiring additional execution privileges. The affected versions are Android-12L and Android-13. missions.

itical as higher-scoring vulnerabilities.

nerability does not require user interaction for exploitation, the attack could happen without the user's knowledg ecurity/bulletin/2023-05-01.

erabilities.

1 the applySyncTransaction function of WindowOrganizer.java. For example, one would need to ensure that the fu

tion. This could effectively disrupt the normal operation of the phone account registration process, leading to a pr

no additional execution privileges needed.

s successful, the attacker could bypass the shadow stack protection mechanism, leading to privilege escalation wit

source. Reviewing the changes in the relevant files in the kernel source or the documentation in Android security  
tion. This could result in a local denial of service without requiring any additional execution privileges. The exploit

This could render the device unusable and lead to a local denial of service. Attack scenarios could include a malici  
pers and security professionals should refer to the official Android Security Bulletin for mitigation techniques and  
ssue that could allow the work profile to read SMS messages, potentially leading to local information disclosure. "

on could silently exploit the permissions bypass to read SMS messages from the work profile. This would lead to u

ide at a highly privileged context, such as the kernel level. Due to the critical severity and the potential to bypass s  
would detail the exact versions affected.

this flaw through local access.

could result in the application crashing or becoming unresponsive, leading to a denial of service condition. An exa  
vice through local access.  
r-scoring vulnerabilities.

or performing actions that the application does not handle efficiently, leading to depletion of system resources and  
e vulnerability is not possible.

privileged user to enable information disclosure via local access.

.html

it exploit this vulnerability to gain unauthorized access to sensitive information which should normally be restrict  
ation fails to properly restrict access to a feature or data. In pseudocode, the issue might look like the following: /

denial of service via local access.

with excessive requests or performing actions that result in resource exhaustion, the attacker could render the application to send numerous requests to consume resources like CPU, memory, or network bandwidth, causing the application to potentially enable information disclosure through local access.

issue to gain access to information that should be restricted, potentially leading to unauthorized information disclosure. Address the improper access control, preventing the potential for information disclosure by authenticated users.

an authenticated user to cause a denial of service (DoS) through local access.  
a vulnerable application.

the Intel(R) Connect M Android application, leading to excessive memory or CPU usage. This resource exhaustion occurs in a way that causes the uncontrolled resource consumption, but these specifics are usually reserved for security researchers to escalate their privileges via local access.

control vulnerability to gain elevated privileges on the device which could be leveraged to perform malicious actions. Malicious scripts in the web application that can be executed in the browser of any user who accesses the compromised device.

?\_s\_id=cve  
settings or a page where those settings are utilized, the malicious script executes in their browser, potentially leading to information disclosure. Content areas: `

ns. This can occur when Category Option Combination Sharing settings are used to control access to specific tracking information. Restricted content will still be displayed to the user. Access to sensitive data.



write beyond the bounds of allocated heap buffers, allowing the attacker to execute unauthorized code or operations and to a local escalation of privilege without needing additional execution privileges, and no user interaction is required.

and actions.

knowledge. This service could then perform unauthorized actions, such as accessing sensitive data or interfacing with other services, leading to a local escalation of privilege without the need for additional execution privileges. No user interaction is needed.

If user interaction is required, an attacker could remotely execute this exploit to gain elevated privileges on the victim's device, leading to a local escalation of privilege without the need for additional execution privileges. This is a local escalation of privilege without the need for additional execution privileges. This is a local escalation of privilege without the need for additional execution privileges.

privilege on the device.

On the service, the use after free condition could be triggered, allowing the attacker's code to be executed. A code examination of the service revealed a missing permission check, resulting in potential local escalation of privilege without needing additional execution privileges, potentially accessing sensitive information or compromising the service's availability.

and availability.

Check this link for the latest updates and possible patches or workarounds.

the LayerState.cpp sanitize function to take control of the device's screen display. If this exploit is available, users should monitor the device for unusual activity and contact the manufacturer for support. It's important to access files in directories of other applications and lead to local escalation of privilege without requiring additional

access or modify files and directories which are normally protected and belong to other applications. Since no additional input (such as '..') is to be passed through. This could, for example, allow a malicious application to construct a string that registers a BroadcastReceiver using the permissions of a System App due to improper input validation. This can lead to local escalation

of a BroadcastReceiver with the permissions typically reserved for a System App. Since the input is improperly validated, it allows the ability to register and receive broadcasts intended for system applications, which can be abused to perform privileged actions. Device manufacturers may distribute these updates over-the-air (OTA). It is also important to only install trusted

app locales, leading to a local denial of service without the need for additional execution privileges. This can happen

on the device or affect user experience across different user accounts.

activity.java which might allow an unauthorized app to change the system locale settings without user consent. This could be done without user knowledge. This could disrupt device functionality, cause confusion by changing the language unexpectedly, or

does not require any additional execution privileges to exploit, but it does need user interaction for exploitation.

leading to resource exhaustion and a boot loop. The attack could be engineered to trigger upon device reboot, causing a denial of service even when the app is running in the background, leading to potential local escalation of privilege without needing further user interaction if successfully exploited.

When the app is not actively being used. This could go unnoticed by the user and would allow the attacker to maintain persistence and should be applied as soon as they become available to ensure security is maintained.

an attacker could create a malicious app that exploits this vulnerability to start background processes or services, bypassing a logic error within the LocationProviderManager.java file, but without access to the vulnerable code or the security manager. The device could continually restart and fail to complete the boot process, rendering it unusable.

Once triggered, the device could fall into a boot loop, persistently denying the user access to their device. The vulnerability exists in the SecureNfcPreferenceController.java file, where there exists a potential approach for the NFC feature to be enabled without the need for additional privileges or user interaction. Once NFC is enabled, the attacker could potentially exploit the device's boot process, leading to a boot loop. This vulnerability affects certain versions of Android, and mitigation or patching instructions are available for those versions.

The vulnerability exists in the NFC interface, which is used for communication between devices. An attacker could send specially crafted packets to the device's Near Field Communication (NFC) interface, triggering the out of bounds write vulnerability. This could potentially allow a malicious actor to gain local escalation of privilege with system execution rights.

Since no user interaction is required, the attacker could automate the exploitation process, allowing them to potentially exploit multiple devices simultaneously. This could lead to unauthorized access, data manipulation, or disruption of normal system operations. The severity of this vulnerability is high, as it allows for local escalation of privilege without user interaction.

or, potentially leading to local escalation of privilege.



y to eavesdrop on the user, potentially gathering sensitive information without the user's knowledge or consent.

r the enumeration of other user's contact phone numbers.

ers. This could be achieved without requiring any user interaction, making it a silent attack that leads to the disclosure of sensitive information. With access to Android source code may be able to review the exact nature and context of the vulnerability within the application.

s needed.

ne required restrictions. This could give the app unauthorized privileges and access to system resources, potentially leading to a local escalation of privilege without need for user interaction. However, the underlying issue lies within the Android framework.

ons that the application should not normally be able to access.

could lead to a possible out of bounds write. Such a flaw could result in a local escalation of privilege without need for user interaction.

since user interaction is not required, the exploit could be carried out silently, potentially leading to unauthorized access to sensitive information and possibly provides guidance on mitigation or patches.

affected versions of Android. This could lead to the malicious app gaining elevated privileges on the device without need for user interaction.

ground. This could enable the attacker's app to perform unauthorized actions without the user's knowledge, leading to a local escalation of privilege.

bounds check, which could potentially allow an attacker to escalate privileges without requiring any additional execution privileges.

function. This exploit could be used to gain elevated privileges on the system without the need for additional permissions, potentially leading to local information disclosure without requiring additional execution privileges. User interaction is not required.

The exploit could then read data out of bounds, potentially accessing sensitive information from the device's memory and escalating privileges.

Additional execution privileges. The exploit would need access to the device or be running a malicious app on it. Code examples for the fix are not provided in the provided information, but they may exist in the source code link.

Without the need for additional execution privileges.

If required, the attacker could execute this operation on a compromised or vulnerable Android device to achieve out-of-bounds heap access, potentially resulting in local escalation of privilege without requiring additional execution privileges.

This exploit, allowing them to execute unauthorized actions without the need for further privileges or user interaction. The

to out-of-bounds heap access. This vulnerability could allow a local user to escalate their privileges on the system

tical data or code within the heap, leading to escalation of privilege on the device. Since no user interaction is required when passed to the vulnerable function within the kernel driver, subsequently leading to heap corruption and potential look for updates pertaining to this vulnerability, as this would contain details on patches or mitigation measures could result in an integer overflow, potentially allowing out-of-bounds heap access. This could lead to local escalation

an attacker to perform out-of-bounds heap access. Successful exploitation could result in a local escalation of priv

er overflow. This could lead to unauthorized out-of-bounds heap access, allowing the attacker to potentially modify. This issue could allow for local escalation of privilege without requiring additional execution privileges. The problem with out-of-bounds heap access, it can lead to local privilege escalation. This can be done without requiring any additional execution privileges and without the user's knowledge.

local escalation of privileges without the need for additional execution permissions and does not require user interaction and data if exploited.

is to gain higher privileges within the system.

is on providing patches and mitigation strategies rather than publicizing exploit details.

relevant security updates provided by device manufacturers or by the Android Security Bulletin.

er to execute arbitrary code or access sensitive information. Escalation of privilege could occur, granting the attacker could lead to an integer overflow and potential out-of-bounds heap access. Exploiting this issue could result in local

ce can exploit this vulnerability to gain elevated privileges, which could lead to a compromise of the device's integrity

his vulnerability to escalate privileges and bypass security mechanisms, allowing for greater control over the device. This could result in local escalation of privilege without the need for additional execution permissions

gher levels of access on the device, potentially leading to unauthorized actions.

overflow in the PVRSRVBridgeServerSyncGetStatus function to gain access to out-of-bounds memory areas on the device. This integer overflow has the potential to permit out-of-bounds heap access, which could be exploited to perform unauthorized actions.

use an integer overflow when accessing heap memory. By carefully crafting the overflow, an attacker could gain access to sensitive data.

Using an integer overflow, the attacker could perform an out-of-bounds heap access to modify memory, potentially leading to unauthorized actions. Generally, the vulnerability could occur in a scenario where an integer variable responsible for size calculation allows out-of-bounds heap access. This flaw could potentially lead to a local escalation of privilege without requiring additional execution privileges.

Android SoC.

perform unauthorized actions such as accessing restricted data, modifying system settings, or installing malicious software.

The integer overflow vulnerability in the PowerVR kernel driver to perform out-of-bounds heap access. Successful exploitation of this vulnerability could lead to out-of-bounds heap access. An attacker could exploit this flaw to achieve local escalation of privilege.

leading to full device compromise and access to sensitive data.

```
able_function(size_t size) { size_t allocation_size = size * sizeof(int); int* buffer = (int*)malloc(allocation_size);
```

the system, access or modify sensitive files, and potentially take over the compromised device for further attack. This vulnerability could lead to local escalation of privilege without the need for additional execution privileges. This issue should be addressed promptly.

compromise the system, accessing or modifying data and functionality that should be restricted.



an security advisory, and a Gentoo security advisory.

rsion of Google Chrome on Android visits this malicious page, the attacker could bypass intended navigation restr  
makes use of the bug in Chrome's Intents handling on Android. Such a page might mimic a legitimate website bu

oud iOS app in version 4.8.0.

l/theses/report\_DanieleCoppola.pdf- <https://github.com/nextcloud/security-advisories/security/advisories/GHSA>  
n the server. They could potentially decrypt the files, view or alter the content, and even plant new files, thereby c  
r, typically, these kinds of vulnerabilities could stem from insufficient validation of server-side encryption proced  
tection of the app, potentially exposing meta information like sharer, sharees, and file activity.

meta information related to files, such as information about sharers, sharees, and file activities.

request that addressed the issue at <https://github.com/nextcloud/android/pull/11242>.

rability through a third-party app, the attacker could bypass the security pin or passcode of the Nextcloud app an  
n an out of bounds write operation, potentially leading to local escalation of privilege.

privileges without user interaction. This could result in the attacker gaining greater control over the device or access

page.

ted privileges without the user's knowledge.

tails to create patches and security measures to mitigate the risk of exploitation.

e with the attacker needing System execution privileges. No user interaction is required for exploitation. The affe

ecific affected versions.

ateTransmitFollowupRequest function of nan.cpp to execute arbitrary code with escalated privileges, potentially  
ow. This vulnerability could allow a local attacker to perform escalation of privilege to System execution privilege  
verity ratings.

to exploit compared to those that can be exploited with lower privileged accounts.

at the device manufacturer for detailed version information.

ty and any available patches or mitigation strategies.

ild potentially allow the attacker to execute arbitrary code, manipulate memory or processes, or gain elevated pri  
, local escalation of privilege with System execution privileges needed and does not require user interaction for e)

illy crafted data to the 'rtt\_unpack\_xtlv\_cbfn' function in 'dhd\_rtt.c'. By doing so, it might corrupt memory adjace

ocal escalation of privilege.

s could then lead to the execution of arbitrary code with high privileges, allowing the attacker to gain control over  
vulnerable code in order to create patches and workarounds to mitigate the risk.

write due to improper input validation, which can lead to local escalation of privilege with System execution privi

ected versions.

acker through some other means.

ompromised, the attacker could leverage the out-of-bounds write vulnerability to execute arbitrary code with sys

curity/bulletin/pixel/2023-03-01.

ing system up to date is essential in ensuring protection against known vulnerabilities like CVE-2023-21071.

an attacker to perform a local escalation of privilege.

etermine the exact affected versions, one would need to check the Android security bulletin or related vendor advis without requiring any user interaction, potentially leading to a compromised device.

d check issue in the 'add\_roam\_cache\_list' function. Users are advised to update their devices with the latest avai

he attacker to execute arbitrary code or interfere with the normal operation of the system.

not provided, a general example of such a vulnerability might look like the following: ``#define BUFFER\_SIZE 256c

alerting the user. The attacker could hide the debug warning and pass the phone to a new user or keep it for ther d technical specifics or proof of concept code.

typically, such code examples would not be readily available to the public for security reasons.

n denial of service. However, without specific details on the nature of the vulnerability, it's not possible to give a p cal escalation of privilege with System execution privileges required. No user interaction is needed to exploit this

3-03-01.



d access or control over the device.

ndroid system's integrity and availability, and possibly to the unauthorized disclosure of information.

to escalate privileges without any user interaction, gaining System execution privileges.

d still lead to a significant impact if leveraged by an attacker.

d security bulletin or updates from the Android Security Bulletin for specific version details.

code with System privileges and potentially gain control over the device.

able through the Common Vulnerabilities and Exposures (CVE) database or other security advisory platforms.

ol of miscservice.cpp silently. By bypassing the missing bounds check, it might read sensitive information out of bo

he app uses the flaw to achieve escalated privileges, effectively gaining System-level access. This could then allow  
ers. To understand the vulnerability, one would typically look into the Android Open Source Project (AOSP) repos  
ich could lead to a local escalation of privilege with the need for System execution privileges. It affects Android's l

er could potentially execute code with elevated privileges discreetly, leading to a compromised device and unaut  
.. To know if exploit code has been published since, thorough research and examination of security forums, reposi

cker might leverage such a vulnerability to gain elevated privileges, execute arbitrary code, or cause a system crash

ted is the Android operating system, specifically the Android kernel.

;-03-01.

function in sms\_PduCodec.c, triggers an out of bounds read. This could allow the attacker to read sensitive inform  
that multiple versions prior to the patch date are vulnerable. Users should refer to the Android Security Bulletin c



If system execution privileges are needed, the attacker could potentially exploit this vulnerability by compromising the `N_LCS_ConvertLCS_MOLRRReq` function of `LPP_CommonUtil.c`, where an out of bounds write due to a logic error can occur without requiring additional execution privileges, and user interaction is not required for exploitation.

to the kernel and ensuring the function properly checks bounds before reading data.

the affected device. Upon processing the message, information from beyond the expected bounds may be read and used to escalate privilege with System execution privileges required. No user interaction is needed to exploit this vulnerability.

If user interaction is required and the attacker would already need to have System privileges, they could use this vulnerability in the `'setToExternal'` function due to the missing bounds check, but such details have not been made public.

checking before writing data, which could be in the `dwc3_exynos_clk_get` function within the `dwc3-exynos.c` file in the kernel. This could be achieved by running a malicious program that takes advantage of the vulnerability to escalate privilege, and System execution privileges are required to exploit it. No user interaction is needed.

Since no user interaction is required, the exploitation could happen without the user's knowledge, potentially leading to local information disclosure with the requirement of having System execution privileges.

!l access.

he intended data structure. This could lead to the disclosure of personal information, configuration data, or other , which could potentially lead to local information disclosure. To exploit this flaw, an attacker would need System d potentially be at risk if they are running a vulnerable version of the Android kernel.

M' rating typically means that the vulnerability could have notable consequences if exploited, such as potential di their systems.

l/2023-03-01) may contain details on any available patches or mitigations. As always, users should apply the latest vulnerability in the 'handleEvent' function of the 'nan.cpp' file to access memory it should not be able to read. Thi

mory, potentially compromising the confidentiality of the system. Since no user interaction is required, a malicious tadata function in aidl\_utils.cc, where a proper bounds check is missing. An example of a bounds check that might :check that could potentially result in local information disclosure if exploited. To exploit the vulnerability, an attac

Bulletin or updates from the specific Android device manufacturers for detailed version information.

utils.cc. This could lead to the disclosure of sensitive information that might be residing in the adjacent memory, : he issue has a CVSS base score of 4.4 (MEDIUM severity) and was published on 24 March 2023. Exploitation grant

; required to exploit the vulnerability.

use after free vulnerability to read out of bounds memory, potentially leading to disclosure of sensitive informatio tem updated with the latest security patches is the primary way to protect against known vulnerabilities.

ation disclosure.

emory they shouldn't have access to. Because the attack doesn't require user interaction, it could be carried out si

the boundary of a buffer but fails to check whether the read operation exceeds the buffer's limit.  
execution privileges required for exploiting it. This vulnerability does not require user interaction for exploitation

authorized code execution with escalated privileges. Since no user interaction is required, the exploitation could be  
e;};int main() { struct data \*ptr = (struct data \*)malloc(sizeof(struct data)); ptr->sensitive = 42; free(ptr); // 'pt

gation.

loit the use after free vulnerability to corrupt memory, which could allow them to execute arbitrary code or compromise  
ity and confidentiality by executing unauthorized actions or accessing restricted data.  
by Android, but exact version numbers would need to be obtained from further updates or security bulletins.

curity/bulletin/pixel/2023-03-01.

o, the application could potentially write data outside the intended memory buffer, which could allow the attacker

f bounds write. The exact code would depend on the Android kernel source code which is not detailed in the provided  
the buildCommand of bluetooth\_ccc.cc to perform an out of bounds write, potentially allowing them to execute

n the device.

d lead to a local escalation of privilege with System execution privileges required. No user interaction is needed f

tion. By triggering the bug, they could write data out of bounds, potentially allowing them to execute arbitrary code

in date.

ed. This could lead to unauthorized access to sensitive information or the ability to trigger behavior that deviates f

privilege without needing additional execution privileges.

otentially exploit the permissions bypass issue in multiple functions of BackupHelper.java to gain the same permis

escalation of privilege with the requirement of User execution privileges. This vulnerability affects Android versio  
item.

d privilege escalation.

the system as a non-privileged user.

ecurity/bulletin/pixel/2023-03-01

it could lead to privacy breaches and, potentially, could be used to gain elevated privileges on the device without  
app could access sensor data, leveraging the vulnerability to silently collect sensitive information or perform acti

ctively prevent legitimate network operations, rendering the Wi-Fi functionality unusable until a fix is applied or th  
are updated to the latest version of Android that includes fixes for this security issue. Manufacturers often releas

data structures involved, potentially leading to local information disclosure without the need for user interaction. No additional execution privileges.

ed information disclosure. Due to the lack of need for additional privileges or user interaction, an attacker could leverage this issue in an unprivileged process without needing additional execution privileges. This issue affects Android versions 6.0.1 through 7.1.2.

age to gain escalated privileges on an affected device. A possible attack scenario is that a malicious application could exploit this issue to gain root access. The availability of a patched firmware version is crucial to protect them from known vulnerabilities like this one. The issue is fixed in the latest version of the firmware. A NULL after freeing the memory can prevent subsequent free attempts. However, specific code fixes for CVE-2021-1472 are not available.

the device.

verController class without the proper permission. This application could then monitor and disclose sensitive information.

string missing a null-terminator, it might read out of bounds memory. The attacker could potentially leverage this issue to gain access to sensitive information. The issue is fixed in the latest version of the firmware, which includes a fix for the vulnerability.

er might be able to retrieve sensitive data from the device without needing any additional permissions or user interaction.

ally leading to a local denial of service without the need for elevated privileges or user interaction.

s overlay could interfere with the user's ability to interact with other apps or the system UI, effectively resulting in

unds read caused by an incorrect bounds check. This issue could potentially lead to local information disclosure, a disclosure.

.android.com/security/bulletin/pixel/2023-03-01.

ead vulnerability within the `ufdt_local_fixup_prop` function, allowing it to read sensitive information from memory illegally without needing additional execution privileges, and it does not require user interaction to be exploited.

g exploited maliciously.

zed actions on the device without the user's knowledge or consent, potentially leading to a local escalation of privileges.

eded.

memory and potentially escalate its privileges on the device. This could allow the attacker to gain access to system memory and potentially write to memory. Generally, such vulnerabilities stem from writing beyond

ilation of privilege without requiring any additional execution privileges, and it does not require user interaction for

Android Security Bulletin at <https://source.android.com/security/bulletin/pixel/2023-03-01>, for details on the security



er without needing any additional privileges or user interaction. This could grant the attacker unauthorized access  
y code execution vulnerability due to a use-after-free. This issue could potentially allow a local attacker to achieve

exploit the use-after-free vulnerability in the registerSignalHandlers function to execute arbitrary code without an  
visible disclosure processes mean that such information is usually only available to security researchers and vendo  
o buffer overflow which could lead to local information disclosure without requiring any additional privileges. No  
-scoring vulnerabilities.

on privileges.

o vulnerability and provides relevant security patch information.

nation disclosure. This could allow the attacker to access sensitive data that is not intended to be accessible, such

rimarily for the use of developers and security professionals to understand and mitigate the issue within their sys  
could be done by inducing the system to access or write to an out-of-bounds memory location after the memory h

2.

anting it elevated privileges. Since the exploitation does not require user interaction, the attack could be carried c

o about the accounts installed on their device. This can result in a local denial of service without requiring any add

or status of accounts on their Android device, potentially causing app crashes or unusable account management in

required permission check. This weakness could potentially allow a local attacker to escalate privileges without needing to be patched or mitigated as soon as possible to prevent potential abuses.

For full details and any possible mitigation strategies or patches.

Exploit the permission bypass in the Transcode Permission Controllers to gain escalated privileges. With these privileges, the attacker can compromise the system.

Access sensitive information that is beyond the intended bounds of an array or buffer. This information could then potentially be used to gain information on the nature of the vulnerability to allow users and developers to understand and patch the problem without needing a bounds check, which could lead to local information disclosure with System execution privileges needed. User interaction is not required.

Access sensitive information, potentially without the user being aware. This information could be used to further compromise the system.

Disclosure of sensitive information or even executing arbitrary code. Since no user interaction is required, if an attacker can exploit this vulnerability, they can compromise the system.

Disclosure if exploited.

Risk of exploitation by unprivileged attackers.

Access to, potentially exposing sensitive information without the user's knowledge.

system execution privileges, it might need to leverage another vulnerability to escalate privileges first. If successfully reading arrays or buffer data to prevent out of bounds access. A common fix might involve validating the array index check that stems from a missing bounds check. This flaw could lead to local information disclosure with System execution

or compromise the affected system. This could be done without any user interaction, possibly running silently in the

bounds read issue in p2p\_iface.cpp to read sensitive information from a location in memory that it should not have access to and it could possibly lead to local information disclosure. To exploit it, an attacker would require System execution

difficulty for potential attackers.

for security measures or be running a process with elevated privileges.

on execution privileges.

is encouraged to check this source for detailed information on patches and mitigation steps.

exists within the p2p\_iface.cpp component, potentially accessing sensitive memory information that should be restricted. For more information, see Microsoft's security bulletin or databases.

However, generally speaking, an out-of-bounds read vulnerability would involve accessing an array or buffer index that is out of bounds. This flaw could lead to local information disclosure. System execution privileges are required to exploit this vulnerability.

interacts with p2p\_iface.cpp to trigger the out of bounds read. Successfully doing so could allow the malicious application to

e. The flaw allows an attacker to gain elevated privileges on the device without needing any additional execution

form actions with elevated privileges on the affected device.

without being granted the necessary permissions. As no user interaction is needed, the exploit could occur without a permission check, which could lead to a local escalation of privilege without the need for additional execution privileg

functionality without the user's knowledge or consent.

ed for any special permissions), the application could exploit the missing permission check to quietly elevate its pr  
y updates and apply the necessary patches or updates provided by the device manufacturers to mitigate the risk  
leges on a device without requiring any additional execution privileges.

eges to do so. This can lead to unauthorized access or control over the device's resources and data.

airing additional execution privileges.

vs the attacker to potentially escalate their privileges without requiring additional execution rights. For instance, a

permission check, allowing users to change Wi-Fi settings for other users without requiring any additional executio

nsent or knowledge. This could result in unauthorized connections to unsecured networks, data leakage, or furth

as on the affected device.

execute arbitrary code with higher privileges without any user interaction, leading to a local escalation of privilege. Memory is freed and later accessed by a mechanism that was not properly locked during multi-threaded operations. User execution privileges are needed to exploit it, and user interaction is not required for exploitation.

usable and achieving a local denial of service. Since an attacker requires User execution privileges, it implies that a persistent reboot loop. Such a loop could lead to a local denial of service on the affected device. To exploit this

exposing the exact code that might be used for exploitation. It is a common practice not to disclose exact code exploit of service on a device. Another scenario could involve a user downloading and executing a malicious piece of software allow an attacker to trigger a persistent reboot loop, leading to a local denial of service. Confidentiality and integrity. In the user space of the targeted Android device.

that validation vulnerability to create a situation where the device enters a reboot loop. This loop could be persistent. Mitigation will typically include any available fixes or provide instructions for mitigating the risk. Additionally, limiting User execution of service with the requirement of User execution privileges. User interaction is not needed to exploit this vulnerability.

With the necessary user privileges, it could repeatedly trigger a device reboot, resulting in a persistent denial of service. Bypassing the fingerprint unlock feature, potentially leading to a local escalation of privilege without the need for a

d, a malicious app could trigger the flaw silently and bypass the fingerprint authentication to gain elevated privileges. The exploit uses a flawed logic in order to construct a code example that demonstrates the exploit.

an attacker to execute code with elevated privileges.

tput\_property\_to\_fdt function to perform an out of bounds write, potentially leading to arbitrary code execution and for additional execution privileges. It affects Android versions 11, 12, 12L, and 13 and is categorized by a 7.8 HI

atin.

escalation.

it user consent or awareness, potentially leading to unauthorized access to sensitive data or functions on the command check that can lead to an out-of-bounds read, potentially resulting in local information disclosure from the Blu

result in information disclosure from the Bluetooth server. Since the vulnerability allows for information leakage that can result in a possible out of bounds read, which could lead to local information disclosure. Exploitation of th

nory. Since user interaction is not required, an attacker could remotely trigger this exploit if they already have a t lead to local information disclosure requiring System execution privileges for exploitation. It affects Android vers

d then execute code within the context of the Bluetooth module and read out of bounds memory, potentially allowing an attacker to prevent abuse. The Android security bulletin may provide more information about the issue and any remediation.

This flaw could lead to local information disclosure if exploited, requiring system execution privileges. No user interaction is required.

If user interaction is not required, the application could silently gather sensitive information without the user's knowledge. Developers and security researchers can reference the Android Open Source Project (AOSP) to understand the nature of the issue.

Local information disclosure within the Bluetooth server, requiring System execution privileges for exploitation.

This could result in unauthorized disclosure of sensitive information from the memory of the Bluetooth service process.

Local information disclosure over Bluetooth, where an attacker with System execution privileges could exploit the vulnerability without user interaction.

Since no user interaction is required, an attacker could silently extract information from the impacted device. While the vulnerability is restricted to prevent widespread abuse, but the details of the vulnerability may be used by developers and security researchers to understand the nature of the issue. The vulnerability was caused due to a missing bounds check, which could potentially lead to local information disclosure. To exploit this vulnerability, an attacker would need to have System execution privileges.

The vulnerability is located in the `ear_resolving_list_complete` function. By doing so, they might access sensitive information beyond the intended bounds.

The vulnerability is located in the `ble_clear_resolving_list_complete` function within `btm_ble_privacy.cc`. Developers should ensure bounds are properly validated during input validation, which could lead to a local escalation of privilege without requiring additional execution privileges.

silently.

For more information, see <https://source.android.com/security/bulletin/pixel/2023-06-01>.

The vulnerability allows an attacker to write arbitrary data to the `BTA_GATT_HANDLE_VALUE_INDICATION` function. Since the vulnerability allows an out of bounds write, it could lead to local information disclosure.

data, modify system settings, install additional malicious software, or perform other unauthorized actions that could be exploited. For more details on the vulnerability and associated patches, see the [vulnerability page](#).

The vulnerability is caused by a missing bounds check, which could potentially lead to local information disclosure. For successful exploitation, the attacker must have system execution privileges (HIGH or CRITICAL).

Information disclosure, such as the retrieval of sensitive data from the device's memory. The exact details of how an attacker can exploit this vulnerability are not provided in the official reference listed for this particular CVE.

The attacker must have already compromised the system to some degree.

Disclosure.

The vulnerability allows an attacker to access and disclose sensitive information from the device without the user's knowledge or interaction. This is a local information disclosure in the Bluetooth server on Android systems with Android-13. System execution privileges are required to exploit this vulnerability.

Memory contents, which might include sensitive information. Since user interaction is not necessary for exploitation, this is a local information disclosure.

To address the vulnerability, users should update the affected system to a version of the software that includes detailed descriptions of vulnerabilities and instructions for mitigation.

The vulnerability is caused by a missing bounds check. Exploitation of this vulnerability could lead to local information disclosure and requires system execution privileges. The exact details of how an attacker can exploit this vulnerability are not provided in the official reference listed for this particular CVE.

Users should ensure their Android devices are updated with the latest security patches, particularly the one for this vulnerability.

For more details, see the [vulnerability page](#).

When that no user interaction is required, the malicious app or attacker could leverage the vulnerability to read memory contents, which could potentially lead to local information disclosure on the Bluetooth server. The vulnerability will still be addressed promptly by users and administrators.

To access, potentially exposing sensitive information handled by the Bluetooth service on an affected Android device. For more details, see the [vulnerability page](#) for details on the available updates and security patches to address this vulnerability.

System execution privileges are required to exploit this vulnerability.

Administrators are informed and can take necessary actions to protect their systems.

es.



missions. This could be done by the app invoking specific calls that interact with the function, leading to the disclosure of sensitive information. This vulnerability relates to a potential out of bounds read that could occur due to improper input validation. If exploited, it could lead to a local system compromise.

The device could be affected through casual or opportunistic exploitation.

ware. Once it has this level of access, the application could attempt to read data outside the bounds of allocated memory, leading to crashes or data breaches. This underscores the importance of applying security updates to address such vulnerabilities.

to local escalation of privilege.

If the user is tricked into selecting this as their default autofill service, it could result in the attacker escalating their privileges.

controller.java. The details are likely to be encapsulated in proprietary Google source code.

This app could gain escalated privileges on the device, allowing it to capture user content that should be restricted, such as messages or location data.

Access to. This could be done without the user's knowledge, leading to information disclosure.

Information disclosure, and exploitation would require System execution privileges. No user interaction is needed to exploit this vulnerability.

ges, an attacker could potentially leverage this vulnerability to access restricted information without the need for disclosure. The flaw requires system execution privileges to be exploited and does not require user interaction for exploitation.

Without user interaction, the attacker would need to circumvent Android's security barriers to gain the necessary privileges. This could lead to potential local escalation of privilege without any additional execution privileges needed.

This could enable the app to perform actions that compromise the user's data or privacy, or affect the integrity of the system. The risk associated with the CVE is considered high.

For example, by using 'array[index]' where 'index' could exceed the array size, leading to undefined behavior. Without sufficient validation of certain parameters, this could result in the attacker reading memory they shouldn't have access to.

This could lead to the disclosure of sensitive information in memory that they are not authorized to read, leading to information disclosure. The absence of proper validation is a common issue in many applications.

bounds read which might lead to access to sensitive information. Since the vulnerability can be exploited without  
need for local escalation of privilege without needing any additional execution privileges, and it doesn't require any u

privileges, enabling them to perform unauthorized actions such as accessing or modifying sensitive data, installing ap  
as required, the exploitation could happen without the user's knowledge. The attacker could then leverage this vul  
age escalation without requiring additional permissions. It affects certain Android versions and does not need use

uity, the malicious app could effectively rebroadcast these intents, causing denial of service by disrupting media s  
at any user interaction.

be conducted silently and without the user's knowledge, potentially allowing the attacker to access sensitive info

edge. This could result in unauthorized access to protected system features or user data. An attacker could use thi  
improperly setting the immutable flag or restricting the access to it can lead to such vulnerabilities: Intent intent = ne  
oper input validation, which could lead to a local escalation of privilege with the need for user execution privilege  
ained by an attacker.

arbitrary Android activity which is not intended to be accessible to third-party applications. Through this exploitati  
privileges on the affected device.

ion privileges. This vulnerability presents a serious risk because user interaction is not required for its exploitation

allow the attacker to gain elevated privileges on the device, enabling them to perform operations that are normally requiring additional execution privileges. No user interaction is necessary for exploitation, making it a more severe potential impact.

potentially allowing the attacker to gain access to sensitive data stored in memory without requiring any user interaction.

their privileges without requiring any additional execution permissions.

for data without authorization.

ns that benefit the attacker. In this case, it is implied that due to incorrect permission checks or logic flaws, the 'device the affected versions include Android 11, 12, and 12L, it's essential to check for updates regularly and keep the

erage this to disclose sensitive information or potentially gain further access or control over the device.

ook like this: ``int array[10]; // An out of bounds write if index is >= 10 or index < 0 void vulnerableFunction(int index)

attacker to bypass administrator restrictions and uninstall applications for all users. This issue could lead to a local vulnerability to gain elevated permissions on a device without user interaction.

require user interaction, it could be exploited silently or via another malicious app or script running on the device, used by an incorrect bounds check, and it has the potential for remote code execution without requiring additional

steps.

nction. Since no user interaction is required, the attacker could execute arbitrary code remotely on the affected device, giving attackers the potential to compromise system integrity, confidentiality, and availability, which can lead to data being displayed before the setup process is completed, potentially leading to a local escalation of privilege without

the attacker can be able to perform unauthorized actions or access sensitive information on the device that is typically protected

the attacker can lead to local information disclosure without the need for any additional execution privileges. Furthermore, user

check, the attacker could manipulate the codec header to write data out of bounds, potentially leading to information

no execution privileges needed.

any code without requiring user interaction or additional privileges.  
attacker to execute arbitrary code on the device remotely.

functionality, thereby escalating privileges locally.

the attacker can perform unauthorized actions without the user's knowledge or consent, leading to a local escalation of privilege.

due to a missing bounds check, potentially leading to local escalation of privilege without needing additional execution

'\_disc\_done'. Since the vulnerability allows for local privilege escalation, the malicious app could potentially gain full

this issue could potentially allow a local attacker to escalate privileges on the affected device without requiring any additional

the device's security and user's data.

cluding explicit code examples to prevent malicious use. You may refer to the Android Open Source Project (AOSP) [Security Bulletin/2023-03-01](#) for more details. An attacker could gain unauthorized access to system resources and perform actions that could compromise the device and

could potentially track users or perform further attacks that may require knowing a device's MAC address.

pass the factory reset protection mechanisms to gain unauthorized access and escalate their privileges on the com

additional execution privileges or user interaction is required, this flaw could be exploited silently, leading to unauthorized escalation of privilege without the need for additional execution privileges. It affects Android versions 11, 12, 13

bulletin/2023-03-01.

compromising the security and privacy of the device.

ionals should review the affected module, `PermissionManagerServiceImpl.java`, and implement the provided security patches. The patches should allow the application to perform actions it would typically not be allowed to do, such as accessing sensitive data. This flaw was discovered in the Android Target SDK has been updated, which could lead to a local escalation of privilege without requiring additional execution

application, especially after an app is updated to target a higher SDK version.

If a vulnerability exists, the app could be granted permissions silently, without user knowledge, potentially leading to local escalation of privilege.

Since no additional execution privileges are required, the malicious entity could execute this attack upon installation.

Since the vulnerability does not require additional privileges or user interaction, any app running on the device could potentially execute shell commands and lead to local escalation of privilege with System execution privileges required. Exploited versions.

Exploited versions.

Unauthorized actions such as installation of a rootkit, data exfiltration, or further compromise of the device. Given the high severity of this vulnerability, it is recommended that users update their devices to the latest version as soon as possible. This vulnerability does not require additional execution privileges and does not require user interaction to be exploited.

The vulnerability exists in the `SendMmCpErrMsg` function, which could result in remote code execution on the device. Since the vulnerability can be exploited without requiring any additional execution privileges, it is considered a high severity issue.

If the device is simply connected to a mobile network. The malicious packets could exploit the out of bounds write vulnerability and result in an out of bounds write. Developers and security researchers will need to review the cellular firmware to determine if this vulnerability exists, which could lead to local privilege escalation without requiring any additional execution privileges. The vulnerability exists in the `SendMmCpErrMsg` function, which could result in remote code execution on the device. Since the vulnerability can be exploited without requiring any additional execution privileges, it is considered a high severity issue.

Security/bulletin/pixel/2023-03-01.

01. It's recommended to review this and follow any guidance or patch instructions provided.

operation. Since no additional privileges are required, the malicious entity could execute this attack upon installation.

lead to remote escalation of privilege without requiring any additional execution privileges, and no user interaction.

tor.cpp, the attacker could trigger an out of bounds read, which may lead to remote code execution with escalate for educational or mitigation purposes and typically include remediation rather than exploitation details. Without the need for additional execution privileges. The affected versions are Android 12, Android 12L, and Android

on the device. Since the vulnerability can be exploited without user interaction, it could happen without the user speaking, it involves manipulating the stored configurations that are handled by PasspointConfiguration.java in a to improper file read operations. This vulnerability may result in local information disclosure without requiring any critical.

ic actions or opening a crafted file or message.

ice, the attacker may craft a scenario where the user is tricked into sharing a file via Bluetooth which inadvertently y information. The vulnerability stems from a confused deputy problem which generally means the legitimate pro ontributor-level access or higher to inject malicious scripts into web pages viewed by other users. on.

le-application-plugin-11-13-cross-site-scripting-xss-vulnerability?\_s\_id=cve.

narios such as stealing session tokens or other sensitive information, defacing web pages, or redirecting users to r mbedded within it. When this post is viewed by other users, the script would execute in their browsers, potentiall ation.

bd--application-used-by-me' and the associated GitHub repository 'https://github.com/WithSecureLabs/megafeis users' account details. This could be done by manipulating inputs or exploiting specific functions of the application cerns. However, the generic nature of the attack could involve making unauthorized API calls or tampering with re g arbitrary API requests.

y- GitHub repository at https://github.com/WithSecureLabs/megafeis-palm/tree/main/CVE-2022-45636 h possibly guessed or brute-forced parameters to unlock a model without needing to authenticate properly. For it should be protected and only accessible to authenticated users.

s.



a lack of proper expiration handling. This might allow the attacker to reset the password of a user's account within a defined period. Additionally, implementing monitoring for multiple password reset requests and limiting reset tokens in insecure password policy.

to guess or brute-force user passwords more easily, thereby gaining unauthorized access to sensitive account information. To prevent this, implement a strong password policy that adheres to industry best practices, including password complexity requirements.

it be achievable through unauthorized actions within the app without the user's knowledge.

oid 12, and 14.1.03.0 in Android 13. This vulnerability allows a local attacker to access sensitive information relate

the information from the secret mode of the Samsung Internet application. This could happen if the attacker is able to exploit vulnerabilities in the software and ensuring that proper permissions and authentication checks are in place to prevent such access. CVE-2023-24497 is a local privilege escalation vulnerability in Samsung Internet version 3.5.16.20 on Android 13. It allows a local attacker to access the device's MAC address without the necessary permissions.

d to further attacks that exploit the gathered information, such as tracking the device or circumventing MAC address-severity vulnerability.

ored within the affected application. This could be exploited by intercepting data transmissions, accessing unsecu

might be able to gain unauthorized access to sensitive information stored on the victim's OneDrive account. This might happen if the attacker is able to install applications on the victim's device without their knowledge. To prevent this, users should be cautious of the applications they install on their devices and ensure that they only grant permissions to trusted apps. ,

ice for Android. The document could be crafted in such a way that it appears legitimate, but is designed to deceive on the mobile device, potentially exposing sensitive information such as one-time passwords (OTPs) and secret alpha, such as the OTP currently being viewed and the secret OTP alphanumeric token during the token setup process.

ts/2022/02/fa865ea4-167e-0010-bca6-c68f7e60039b.html.

may have the capability to capture the device's screen. When the user opens SAP Authenticator and accesses a one-time password screen during the setup process. This information could then be used by an attacker to gain unauthorized access to the system.

properly checks the bounds of an array when accessed, leading to potential overflow or underflow conditions that can be exploited by an attacker.

letín

s, allowing the attacker to corrupt memory, potentially leading to arbitrary code execution, data leakage, or crash

annel update: <https://chromereleases.googleblog.com/2023/03/stable-channel-update-for-desktop.html>

age, the attacker could exploit the inappropriate implementation of Intents to spoof the domain displayed in the l

L page.

ting a legitimate site when in fact they are on a malicious page. This could lead to phishing attacks or other forms s, and displays the current URL. It's an important security feature because it shows the security status (such as HT could manipulate the contents of the omnibox to display a false URL or search query, effectively spoofing the omni ous WebApp to spoof the contents of the PWA (Progressive Web App) installer using a specially crafted HTML page

hen convinces a user to visit this website, and when the user attempts to install what they believe to be a legitimate application, they are exploiting the user's lack of knowledge of the vulnerability.

lass navigation restrictions through a specially crafted HTML page.

rome on Android.

- Chromium bug tracker: <https://crbug.com/1365100>

malformed Intents to bypass Chrome's navigation restrictions, potentially leading to unauthorized actions or exposure of sensitive information. This vulnerability could be exploited to bypass navigation restrictions enforced in handling these Intents within Google Chrome, which could be exploited to bypass navigation restrictions on a restricted HTML page.

l) and the Chromium bug tracker (<https://crbug.com/1398579>).

ion of Google Chrome on Android, the attacker could leverage the vulnerability to leak sensitive information that. Generally speaking, exploitation of this type of vulnerability might involve a crafted HTML page that triggers the Autc

leTrainerProvider.java.

5aba5c705fb68426.

/.  
'../' to move up the file hierarchy and read or manipulate files to which the application should not have access. Th

cable Trainer found on its GitHub releases page.

ution privileges.

d (AAVCAsembler.cpp), triggers a heap buffer overflow. This could then lead to out-of-bounds reading of memor  
, Android 12L, and Android 13), installing the security updates that address this specific vulnerability is crucial. Ad  
pass, potentially leading to a remote escalation of privilege within Bluetooth settings without requiring additional  
ble.

alation of privileges, this could be used to enable Bluetooth without the user's knowledge, pair with unauthorized

ntentially lead to local escalation of privilege without the need for additional execution privileges.

ite vulnerability to perform unauthorized actions or gain higher privileges than intended, potentially compromisi

control over the device, access to sensitive data, or the execution of arbitrary code without the user's knowledge. Without requiring any special execution privileges or user interaction, the app could exploit the unsafe deserialization known vulnerabilities. Additionally, users should only install applications from trusted sources and pay attention to system files. This could lead to local escalation of privilege without requiring any user interaction for exploitation.

Next to the `clearApplicationUserData` functionality, the attacker could traverse file directories and potentially delete files. Regular security reviews and updates to the latest Android patches are also important measures to protect against this vulnerability without requiring additional execution privileges, and user interaction is not needed to exploit this vulnerability.

Conclusion. The attacker or the application could exploit the vulnerability to replace the boot partition, gaining higher privileges shortly after vulnerabilities like this one are disclosed. Users should also avoid downloading applications from untrusted sources to avoid memory corruption. Such corruption can result in a local escalation of privilege without the need for any additional user interaction on the affected systems.

Higher access rights on the affected device, potentially taking over the system or accessing sensitive information without user interaction about the vulnerability and steps for mitigation. Without requiring additional execution privileges, the attacker could elevate its privileges without the user's knowledge. With escalated privileges, the attacker could access or alter system files, which could lead to local escalation of privilege without the need for additional execution privileges.

affected versions.

crash or code execution. Here's a generic example: `void *buffer = malloc(sizeof(data)); free(buffer);` // Buffer is freed, allowing the attacker to execute commands or access resources that are normally restricted.

Attacker to escalate privileges without needing additional execution privileges.

e information, or manipulate system functions.

com/files/171239/Android-GKI-Kernels-Contain-Broken-Non-Upstream-Speculative-Page-Faults-MM-Code.html. through code auditing or dynamic analysis, and exploits would vary depending on the specific implementation de memory corruption. This could be used to escalate privileges and perform unauthorized actions on the device wit

icker could gain elevated privileges on a device.

idge, compromising their privacy. Since the vulnerability allows local escalation of privilege, the attacker might fu is that indirectly demonstrate what the issue was, and security researchers may analyze these to understand how

r privileges on the system without needing any additional execution privileges.

it triggers the vulnerability, leading to unauthorized local escalation of privilege. Since the vulnerability does not r ils to the vendor and allowing time for them to patch the issue before any specifics of the exploit are made public.

. Since user interaction isn't required, the attack could be performed silently without the user's knowledge, leadir g a risk of local escalation of privilege. For exploitation, System execution privileges are required, and user interac

y of the device user.

ing the user through the typical privacy indicator, thereby secretly capturing sensitive information without consei te the vulnerability.

1 disclosure without requiring additional execution privileges.

tion of this vulnerability requires user interaction.

which poses a risk to user privacy and security.

ch would be abstracted within the system's source code and not directly exposed in the form of code snippets. ectly cleared during the reset process. This could expose WiFi network names, passwords, and other sensitive det

exhaustion. Since no user interaction is required, the malicious app could trigger this condition in the background enRule method, and exploiting it would typically involve maliciously crafted code that calls this method repeatedl

network.

nterception and tampering by unauthorized parties.

on or credentials. An attacker could leverage this vulnerability to steal information or act on behalf of the user wi

entially allow an application to access user-sensitive data. dministrators.

ate the risk of this vulnerability.

ervices. This could include personal information, credentials, or media files which could be exploited for further att 022-32836.

nt information over the network using HTTP, the attacker could intercept this traffic. This would allow the attacke easily eavesdropping on the communication.



AndroidManifest.xml file.

sensitive functionalities. The app could do this without requiring any user interaction, deceitfully obtaining signature-write, although the latter is limited to files with a .txt extension. This issue was addressed in version 3.0 of the app

though it's limited by the .txt file extension.

For instance, they could possibly upload internal configuration or user-specific data files from the app, leading to in d\_Android\_app/', ownCloud's official security advisory at '<https://owncloud.com/security-advisories/oc-sa-2023-> It is an information disclosure since it affects two databases, 'filelist' and 'owncloud\_database'.

database'.

It allows the attacker to read sensitive information from the 'owncloud\_database' or affect the integrity of the data

Owncloud\_Android\_app/

encrypted communication.  
ed system.

l.  
server. Due to the app's improper certificate validation, the attacker could present a forged certificate to the app,

by accessing the log files.  
n SUSHIRO Ver.2.0.1.

'N identifier JVN84642320.

als into the log files, the attacker could potentially extract and use those credentials to gain unauthorized access to nately. For example, a generic code snippet that could lead to such a vulnerability might look like this: // WARNING: e vulnerabilities. If the developer provides any specific instructions or recommendations, users should follow those is that offer security features, and perform regular code audits to check for any inadvertent logging of confidentialia sionList component, which allows an attacker to cause memory corruption.  
ode or system crashes.





[curity.gentoo.org/glsa/202309-17](https://security.gentoo.org/glsa/202309-17)

interface in full-screen mode. This could lead to phishing attacks, where the attacker might trick the user into disclosing history display impossible. Other conversations would not be affected by this error.

30 or wire-server 2022-11-03 (chart/4.26.0). As a temporary workaround, affected messages can be deleted from

error when converted to the HTML representation in the Wire web-app, leading to the affected conversation's cl

malicious application designed to trigger the flaw in the `binder_vma_close` function in `binder.c`, possibly taking control  
lead to local escalation of privilege without needing additional execution privileges. No user interaction is required

Since no user interaction is required, the malicious code could silently run in the background, corrupt memory, and c

1. This flaw could allow an attacker with physical access to the device to escalate privileges locally without needing

pass the lockscreen, gaining unauthorized access to the device and potentially escalating their privileges without ne  
patches, which are often included in monthly security bulletins published by Android, such as the one referenced

affected users have applied the necessary security patches.

bility. This could lead to unauthorized access to the device's data and the ability to perform actions with escalated

could lead to local information disclosure without requiring additional execution privileges.

consult additional Android security bulletins or updates for detailed version impacts.

access protected content and potentially harvest sensitive user data from the device without alerting the user, the

No additional execution privileges are required for exploitation, and user interaction is not necessary.

ted Android device. This could lead to a resource exhaustion condition, causing the device to become unresponsive.

It results in the malicious app being granted accessibility services without proper permissions, allowing for local escalation of privilege.

Once the memory has been corrupted, the attacker could potentially execute arbitrary code with elevated privileges. We ensure they integrate the latest security updates into their Android applications and firmware.

Local escalation of privilege without requiring additional execution privileges. This could compromise the system's security and data integrity.

prevent the uninstallation of a malicious package. Since no additional execution privileges are required, the attacker can perform a local escalation of privilege.

Local escalation of privilege.

any permissions, taking advantage of the missing permission check in `getMainActivityLaunchIntent` of `LauncherApp`. This results in the malicious app being granted accessibility services without any user interaction, leading to a local escalation of privilege without requiring any additional execution privileges.

a user's knowledge. Once the phone account is enabled, the attacker might leverage it to intercept or place calls, |

nts, tricking the user into enabling a malicious phone account without their knowledge. This can lead to further ab  
users due to a missing permission check, potentially leading to a local escalation of privilege without needing adc

o retrieve photos from other user profiles stored on the device, leading to unauthorized access to private images  
privilege escalation.  
if AvatarPickerActivity.java. It's important for Android device users and administrators to regularly install these se

eeded, the attacker could remotely trigger the vulnerability and cause the device to crash and enter a loop of cra  
o a local escalation of privilege without the need for additional execution privileges.

e malicious app to perform an out of bounds write operation, potentially leading to code execution or modificati  
acker to execute arbitrary code or escalate their privileges on the system.  
om Android, so keeping devices up-to-date is crucial. Additionally, it's good practice to only install trusted applica

ismatch in the code. This issue could lead to a local escalation of privilege without requiring additional execution

1 of this vulnerability can occur without user interaction and does not require additional execution privileges. An attacker can use an intent within an Android application to redirect the execution flow to unauthorized activities or settings. Development of this exploit does not require the need for any additional execution privileges. No user interaction is required for exploitation.

action.

Specific mechanisms of the vulnerability and typically are not disclosed by security bulletins to prevent further abuse. The device becomes unresponsive or unable to perform tasks until the issue is resolved. Since no additional privileges are required for exploitation, notification access without the need for additional execution privileges. However, user interaction is required to exploit this vulnerability.

notification access without the need for additional execution privileges. However, user interaction is required to exploit this vulnerability.

by granting notification access that could be used to observe or tamper with notifications, potentially leading to data loss or unauthorized access to sensitive information.

Settings fail to persist, the attacker could gain privileges they would not normally have access to. As no user interaction is required for exploitation, this vulnerability could lead to a local escalation of privilege without any additional execution privileges needed, and with no user interaction required.

Unauthorized access or control over the device functions related to permissions settings.

An attacker could escalate privileges locally, gaining access to protected resources or the ability to perform actions that would otherwise be restricted. The vendor is working to fix the vulnerability, and details will be provided for developers and vendors to patch the affected systems and devices.

1 of privilege with no additional execution privileges needed, affecting the device's security by potentially allowing an attacker to gain unauthorized access to sensitive information.

e failure to persist permission settings and subsequently achieve unauthorized access or elevated privileges on the `adapterService.cpp` file. It involves a possible out of bounds read due to type confusion, which could potentially le

requiring any user interaction or additional privileges. An attacker could exploit this to gain unauthorized access to locked features within the Bluetooth service of an affected device, without needing any special execution privileges

ing logged in the 'user' build type of Android. Specifically, CarNotificationListener.java inappropriately logs the return

information such as the user's account name or other PII, leading to privacy breaches or further attacks such as identity theft. An example might look like this: ``public void onNotificationPosted(StatusBarNotification sbn) { Log.d("CarNotificationLi

which an attack could compromise the security of the device and allow the attacker to access or modify sensitive data

to a security issue where arbitrary data could be written to the page, resulting in memory corruption. This could lead to a system crash or other unintended behavior.

memory corruption, potentially resulting in unauthorized access to data, disruption of normal operations, or in some cases, a denial of service.

and risks.

in a tapjacking or overlay attack, potentially leading to a local denial of service by triggering a factory reset without the user's consent.

they unknowingly initiate a factory reset of the device due to the overlay concealing the actual interface that requires root permissions are required.

overlay the toggle button and enable apps to modify system settings without the user's consent. It affects Android versions 7.0 and above, where the overlay layer is used to create a deceptive layer over the actual user interface.

screen. When the user interacts with what they believe to be the game or survey, they could actually be toggling y, the malicious app can intercept this interaction and toggle the 'Modify system settings' switch without the user vulnerability could lead to a local DoS with no additional execution privileges required. Some level of user interaction causing a local denial of service.

cal as higher-scored vulnerabilities.

: trick a user into performing unintended actions, or they may block a user from interacting with the application, e

adding to the theft of sensitive information from a legitimate application when a malicious app is installed on the s  
sive information from the victim's device.

A-35f7-fqrc-4hhj.

cious app alongside the legitimate WARP client, the attacker's app could hijack tasks that were intended for the V  
erify the manifest configuration of their apps to protect against similar issues. Developers should ensure that all a

ecurity UI through a crafted HTML page. This could potentially lead to misleading the user into performing uninte

ug.com/1356987- <https://chromereleases.googleblog.com/2023/01/stable-channel-update-for-desktop.html>- htt  
cting with legitimate and secure UI elements in fullscreen mode, when in fact they could be compromised, enabli

to bypass the main origin permission delegation by using a specially crafted HTML page.

ses.googleblog.com/2023/01/stable-channel-update-for-desktop.html- Gentoo Linux Security Advisories: <https://www.gentoo.org/glsa/202305-10>, and <https://security.gentoo.org/glsa/202311-11>.  
Google Chrome for Android, the attacker could potentially bypass security measures that delegate permissions to  
sue.

; specifics that could lead to misuse or malicious activities. However, researchers and developers can refer to the i  
the contents of the Omnibox (the URL bar) through a malicious HTML page.

king it appear as if the user is on a different website than they actually are.

/gentoo.org/glsa/202305-10, and <https://security.gentoo.org/glsa/202311-11>.

earing attacks. The attacker could capture sensitive information or credentials by tricking the user into interacting

otentially allow a third party application to read or write to the Zoom application's data directory without proper

ended directories.

Zoom Android client to version 5.13.0 or later.

information stored within Zoom's application data directory. This could include read access to potentially sensitive forums to prevent facilitating attacks on users who may not yet be protected. Responsible disclosure and handling

Nextcloud Talk Android app.

Nextcloud.com/reports/1784645, and the Nextcloud security advisories page at <https://github.com/nextcloud/security-advisories>

Due to this vulnerability, the attacker is able to bypass the app's passcode protection mechanism and thereby access the user's data. A proof of concept request found at the provided reference link.

This vulnerability could be used to perform unauthorized actions or access sensitive information.

Exploits.

Link: <https://www.qualcomm.com/company/product-security/bulletins/january-2023-bulletin>.

This vulnerability could lead to data corruption, giving the attacker the ability to execute arbitrary code, gain elevated privileges, or cause a denial of service. For instance, by submitting data that the application does not expect or cannot handle properly (e.g., a large array). This is a security bulletin, and follow any remediation advice provided. As a best practice, regular security audits and input validation are recommended.

Denial of service, or execution of arbitrary code after license authentication.

An attacker could perform operations to trigger the array index issue after a license authentication process. This could result in the corruption of data, allowing attackers being able to take over user accounts via a specially crafted deep link.

An attacker gaining unauthorized access to user accounts.

Phishing emails, messages, or social media. When an unsuspecting user clicks on the link, the attacker could potentially exploit the vulnerability by using certificate pinning, and ensuring that all deep link handling is done with careful validation of the intent and

assigned to mimic the dialog. This can trick the user into thinking they are providing location permissions to the browser. The score indicates that the vulnerability should be taken seriously, but it is not as critical as those with high

Information about the vulnerability.

On the Nextcloud website, the spoofed Location Permission dialog appears. If the user accepts thinking they're granting permission to

The character appearance to deceive users about the web address they are visiting, potentially leading to phishing and

Nextcloud.com/advisories/cve-2022-47524.

Characters from various scripts, like Cyrillic or Greek alphabets. This can trick a user into thinking they are visiting a legitimate



from different scripts. The attacker then creates a phishing site at the deceptive domain. When a user of the F-Sec  
s security issue could lead to cookie-based data leakage and might be exploited to bypass intended security restri

ic bug at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1791201](https://bugzilla.mozilla.org/show_bug.cgi?id=1791201).

e designed to look like a legitimate service that the user trusts. If the user interacts with it, the SameSite=Strict co  
ash that could be exploited by an attacker.

ozilla.org/show\_bug.cgi?id=1784588).

his driver, it could trigger a stack-buffer overflow that leads to a crash. An attacker might leverage such a crash to  
ire of exploit code in the interest of security. However, the technical details of the vulnerability suggest it involves  
udio notification. This flaw does not permit the website to bypass initial permission prompts; it strictly affects the

t, the user might be unaware that their conversations or surrounding sounds are being recorded, leading to poten  
l1).

e.

face to hang, and if the browser's session restore feature is enabled, it would constantly try to reopen the proble  
te error. The user could explicitly ignore the error and proceed to the site, which is against the intended security r

ertificate error even on HSTS-protected domains, the user might ignore the warning and continue to the website.

ure to record HSTS settings could potentially expose the user to certain attacks, such as man-in-the-middle (MiTM  
st level of concern on the CVSS scale.

rg/show\_bug.cgi?id=1757138.

example for this vulnerability isn't possible or relevant. The issue would need to be addressed by the Firefox dev  
te that sends an HSTS header to the browser, indicating that future requests should be sent over HTTPS, and the b

ecute arbitrary code.

lla Foundation Security Advisory (MFSA) document, which details the vulnerability.  
d review the patches provided by the developers and follow best practices for secure coding. Code analysis can al  
d by a user with a vulnerable version of the software, the attacker could execute arbitrary code with the privilege:

mozilla.org/show\_bug.cgi?id=1743931.

into thinking that the alert or prompt is coming from the underlying legitimate website, possibly divulging sensitiv

in carriers, if the number containing the USSD code was dialed, it could lead to unintended actions performed on t

ie link, the browser may dial the USSD code \*12345#, causing potential unintended actions on the user's mobile a  
o make a phone call, the USSD code is automatically dialed, which could then execute commands that manipulate

org/show\_bug.cgi?id=1728742.

and possibly malicious web activity.

ozilla report at [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1705094](https://bugzilla.mozilla.org/show_bug.cgi?id=1705094).

oes not point to actual web content. This could potentially be used to trigger protocol handlers or initiate actions i  
ld corrupt the methods of a JavaScript Array object through prototype pollution, leading to the execution of attac

prototype of an Array object, injecting malicious code that could be executed when the array methods are called. If you should update to Firefox ESR version 91.9.1, Firefox version 100.0.2, Firefox for Android version 100.3.0, or Thunderbird

due to prototype pollution, which in turn, could allow the attacker to execute arbitrary JavaScript code with the privileges of the user, leading to information disclosure and enable spoofing attacks on Firefox for Android.

When a user clicks on a legitimate site when, in fact, it directs users to a malicious site. For instance, the attacker might use this technique to spoof a URL, thereby obtaining authenticated access as that user. This vulnerability is contingent upon the attacker and the victim.

Mozilla VPN products indicate the fixes: - iOS: <https://github.com/mozilla-mobile/guardian-vpn-ios/commit/4309f5c> - Android: <https://github.com/mozilla-mobile/guardian-vpn-android/commit/4309f5c>. If the attacker and the victim share the same source IP, the attacker could gain authenticated access as the victim, potentially viewing the victim's data. This is due to improper handling of Unicode control characters, which could result in the display of misleading web links. The attacker could also exploit the Unicode control character's specifications, allowing a malicious actor to make a URL in the text appear as if it's pointing to a legitimate site.

Android prior to version 12.0.0 and iOS prior to version 3.9.4, and SoftBank +Message App for Android prior to version 12.9.5 and iOS prior to version 3.9.4. The attacker could exploit this vulnerability to spoof a URL, thereby obtaining authenticated access as the victim, potentially viewing the victim's data. To address the issue by handling Unicode control characters appropriately to prevent spoofing and phishing attacks. The attacker could also exploit the Unicode control character's specifications, allowing a malicious actor to make a URL in the text appear as if it's pointing to a legitimate site.

Example URLs: <https://jvn.jp/en/jp/JVN43561812/index.html>, <https://www.au.com/mobile/service/plus-message/information>

and lead to local information disclosure, leading to a local escalation of privilege without the need for additional execution.

For more information, refer to the official security bulletin at <https://source.android.com/security/bulletin/pixel/2022-12-01>. The vulnerability is located in the `AddAppNetworksFragment.java` file for misleading the system about network add requests. As a result, it could be exploited to prevent malicious use until a majority of users have applied the necessary patches and updates.

This vulnerability may lead to local information disclosure.

For more information, refer to the CVE description.

kernel memory contents. Since user interaction is not needed, a malicious application with the necessary permissions could perform an out of bounds write operation. Exploitation of this issue would not require user interaction and could allow an attacker

to write to the (Android Framework) hardware abstraction layer (HAL) by inputting data that triggers the out of bounds write.

the local device and requires User execution privileges to be exploited, although no user interaction is necessary

an application could execute unauthorized SQL commands without the user's knowledge, leading to access of restricted data. This issue affects devices running Android versions 4.4.2 through 5.0.2. Exploitation of this issue requires System execution privileges.

local.c. This exploitation could enable the attacker to elevate their privileges to a higher level without needing the user's interaction. The details of the vulnerability and its mitigation are generally provided to allow users to stay informed. This issue is disclosed if exploited by an individual with System execution privileges. It affects Android kernel versions associated with Android versions 4.4.2 through 5.0.2.

privacy risk to the device user.

An application could possibly read sensitive data from the kernel's memory space that was not intended to be accessible to user applications. Users should refer to the provided Android Security Bulletin link and the device manufacturer's website for more information.

Memory Allocation. If exploited, it could lead to local escalation of privilege without requiring any additional execution privileges. Mitigation should be prioritized accordingly.

from the user.

normally restricted from ordinary users or applications.

the \_for\_fw function to perform permissive memory allocation, allowing it to bypass existing mitigation strategies. Work is being done to address this issue.

o user interaction is necessary for exploitation.

ne within a secure and responsible context, such as for educational purposes or for developing countermeasures.

3 without needing user interaction. The exact impact depends on the data that is disclosed through this vulnerability.

s made public on December 16, 2022.

ed system.

lel.

re data, bypassing security restrictions, and potentially spreading to other connected systems.

refer to the Android security bulletin and updates from the Android Open Source Project (AOSP) for technical details.

re attacker could gain elevated privileges and perform unauthorized actions, such as extracting data, installing ad

re device could crash, resulting in a denial of service without requiring any additional privileges or user interaction.

scalation of privilege with the need for System execution privileges. No user interaction is required for the exploit

e critical memory structures and allow the attacker to gain escalated privileges on the system without any user in

ker could craft specific input to trigger the out of bounds write in the fillSetupDataCallInfo\_V1\_6 function of ril\_se

heck, which could lead to remote information disclosure without requiring any extra execution privileges. Essential patches should be addressed promptly by users and administrators.

manufacturer for detailed information about the affected versions and the availability of patches.

leverage the out of bounds read to gather sensitive information from the device's memory. The critical aspect of this vulnerability is that it can be exploited by the Android security team or their device manufacturers to mitigate this vulnerability. The details about the fix can be found at [this link](#). Users should check this link for details about the vulnerability, affected versions, and the security patch that addresses this issue without the user's knowledge. This could lead to a breach of privacy, unauthorized access to personal or corporate data, and a local escalation of privileges due to an erroneous bounds check, which could potentially allow for a local escalation of privileges.

control over the system.

locally.

The system could be compromised, leading to unauthorized access or the possibility to inflict further damage such as a denial of service.

CVSS: System execution privileges for exploitation. User interaction is not necessary to exploit this vulnerability.

Particular patch level or update addressing the vulnerability.

Public disclosure.

Vulnerability in DoSetCarrierConfig to read sensitive information from memory that it should not have access to, potentially leading to a local escalation of privilege with system execution rights. This vulnerability affects Android kernel versions and does not require user interaction. This vulnerability is considered critical as vulnerabilities given a higher rating.

Users should check this link for details about the vulnerability and the security updates or patches released to address them.

This vulnerability can be exploited by an uninformed attacker but rather by someone who has already obtained a significant level of access to the Android system. The vulnerability is triggered by a bounds write operation in the encode function of wlan\_data.cpp without any user interaction. This manipulation could lead to a local escalation of privileges. The bulletin should provide more insight into the vulnerability, including any available patches and recommended mitigations.

User interaction and requires system execution privileges for exploitation.

User knowledge. This could lead to unauthorized actions being performed on the device, such as accessing or modifying system files, which could lead to a stack clash leading to memory corruption which could allow an attacker to escalate privileges locally to the system.

Vulnerability to gain these privileges.

Potentially taking full control of the affected device.

Trigger a stack clash in the vulnerable component of the Android kernel, leading to memory corruption. This could allow an attacker to escalate privileges locally to the system.

i.cpp' without the user's knowledge. This could allow the attacker to execute arbitrary code with elevated system  
is or in updates from the Android Open Source Project repository and related patches.  
information disclosure, requiring System execution privileges to exploit. It affects Android kernel versions but does

a missing bounds check in the DoOemSetTcsFci function within miscservice.cpp, which would normally involve en  
authorized access to sensitive information stored in the device's memory. Since no user interaction is needed, the a

ibaAuth function. This could result in the application disclosing sensitive information stored in the memory segme  
ds read due to a missing bounds check and allow local information disclosure with the need for system execution

code (miscservice.cpp), but for actual examples, one would need to reference the source code repository at the p  
:VE description.  
vulnerability to access sensitive information that's ordinarily protected, adding to the malicious capabilities by pc  
missing bounds check, which could potentially lead to local information disclosure.

is could be done by triggering the faulty code path, which lacks necessary bounds checking, and reading data bey  
e to a missing bounds check which could potentially allow an attacker to perform a local escalation of privilege.

vulnerability was addressed.

ing to full system compromise.

ive nature of actively maintained vulnerabilities. The information provided does not include such examples, and t  
ilder::BuildSetSession. This could corrupt memory and potentially allow for execution of arbitrary code or cause i

As a result, a malicious actor could potentially exploit this vulnerability to disclose sensitive information from a sys

uthorized disclosure of information stored in memory. The attacker could potentially leverage this information to  
r to escalate their privileges to the System level without requiring user interaction.

rbmsdata.cpp code to perform an out of bounds write operation, potentially leading to memory corruption and a  
vided by the vendor, in this case, the Android security bulletin, to understand how to mitigate such vulnerabilities

/stem execution privileges needed. Furthermore, user interaction is not required for exploitation.

requestdata.cpp. Since no user interaction is required, the attacker could potentially achieve local escalation of p  
lead to local escalation of privilege with System execution privileges needed. User interaction is not required for e

write operation. Since no user interaction is required, this could be done silently and potentially lead to further ex



write due to a missing bounds check, which could potentially allow for local escalation of privilege with System execution control over the system. No user interaction is necessary to trigger the vulnerability.

Consider. As this method lacks proper bounds checking, it could overwrite crucial memory structures or inject malicious code. This could potentially allow a local attacker to escalate their privileges to the System level. The affected product is Android.

BuildSimUpdatePb3gEntry'. This manipulation could lead to an out-of-bounds write, ultimately allowing the attacker to bypass updates and patches to determine if they might be affected. No user interaction is necessary for the exploit to lead to local escalation of privilege requiring System execution privileges. No user interaction is necessary for

out of bounds write in the SimUpdatePbEntry::encode of simdata.cpp, indicating the issue arises in the code responsible for handling updates and patches to determine if they might be affected. Since user interaction is not required, the exploit could be triggered without user interaction, leading to unauthorized data access, or other privileged actions. Since user interaction is not required, the exploit could be triggered without user interaction, leading to unauthorized data access, or other privileged actions.

the integrity and confidentiality of its data. The vulnerability is patched.

The application wouldn't require any user interaction and could silently escalate its privileges to System level, allowing it to perform actions that require system-level permissions. This could lead to local escalation of privilege with system execution privileges.

version information and patches.

write. This could result in an escalation of privilege attack, where the attacker may gain unauthorized control or access to the system until the majority of affected systems are patched. For information on mitigation and patches, refer to the official Android security bulletin.

attacker to perform an escalation of privilege and execute with system-level permissions. It affects Android kernel versions 4.14.0-rc1 through 4.14.0-rc2.

should not have access to. This would enable the attacker to execute arbitrary code with system privileges without the need for user interaction. The exploit could be exploited to achieve local escalation of privilege with System execution privileges. No user interaction is necessary.

the application to execute arbitrary code with elevated privileges. Since user interaction is not required, the exploit could be used to achieve local escalation of privilege with System execution privileges, leading to undefined behavior or code execution.

the application to execute arbitrary code with system privileges without requiring user interaction for exploitation.

The application could potentially perform an out of bounds write which might allow them to execute arbitrary code with system privileges without requiring user interaction and does not require any user interaction for exploitation. The affected product is the Android kernel.

the application for version specifics.

the application on the affected device without the need for user interaction.

the application could read vulnerability, they could potentially execute arbitrary code on the device, gaining unauthorized access to

additional execution privileges. No user interaction is needed to exploit this vulnerability.

the application to exploit this without additional execution privileges and without requiring user interaction.

the application to exploit this without additional execution privileges and without requiring user interaction.

the application to exploit this without additional execution privileges and without requiring user interaction.

the application to exploit this without additional execution privileges and without requiring user interaction.

the application to exploit this without additional execution privileges and without requiring user interaction.

the application to exploit this without additional execution privileges and without requiring user interaction. Since the application could attempt to read sensitive information from the device, compromising the confidentiality of data. Since the application could attempt to read data outside the legitimate boundary of an array or buffer, potentially leading to undefined behavior or code execution.

LTE authentication needed. The attack does not require user interaction to be exploited.

ware, could trigger an out of bounds write due to the missing bounds check, leading to remote code execution on security vulnerabilities to prevent exploitation by malicious actors. The absence of a necessary bounds check. This vulnerability could lead to remote information disclosure with system execution privileges.

ded.

\_MiningCodecTableWithMsgIE. Since system execution privileges are needed for exploitation, the attacker would require root-level operations. These privileges are typically higher than those granted to standard users and would allow the attacker to bypass the missing bounds check, which could result in remote information disclosure without the need for additional execution privileges or root access.

it <https://source.android.com/security/bulletin/pixel/2022-12-01>. It's recommended to check this link for the latest information. This information could potentially include private data or system information that could be used for further exploitation.

:SAECommDbManagement.c file.

ForPlmn' function in the 'SAECommDbManagement.c' file.

memory. The attacker could possibly gather sensitive data from the affected device, which may be used for further exploitation.

unction. This could lead to remote code execution without user interaction, potentially granting the attacker system-level access.

vice. The attacker might exploit the vulnerability through a malicious app or a compromised system process to bypass security advisories, as well as the Android ID and details surrounding the nature of the vulnerability, are meant to help users understand the risk and ensure that their device's operating system is kept up to date with the latest security patches. It is also important

that the Android security bulletin would be necessary. Depending on the nature of the vulnerability, it might allow an attacker to remotely execute arbitrary code, or cause a denial-of-service condition, depending on the nature of the vulnerability in

the code involved, it's not possible to provide an accurate code example. Generally, any code that writes to an array, leading to an out-of-bounds write. This memory corruption could potentially allow for further escalation of privilege.

knowledge.

at <https://useboomerang.com/>.

that the parent intended to prevent.

t take advantage of the weakened state to uninstall the parental control app without being detected. This could o  
mory to a PC. This backup can expose the app's API token, which is essential for authenticating requests to the ap  
of the user's data within the app. This can be particularly worrying for a parental control application, as it may cor

:horized backup and access to the app's internal data.

the API token from it. With the exposed API token, the attacker would then be capable of making authenticated r  
bitrary API calls through the unsecured communication channel.

d.

data stream. This script could invoke any accessible API, allowing the attacker to perform unauthorized actions or  
ryption enforcement, such as HSTS (HTTP Strict Transport Security), can also be a good additional layer of securi  
iddle (MITM) attacks. As a result, an attacker could potentially inject scripts and invoke any API, compromising tl

i stream due to the application's acceptance of non-SSL/TLS communications. Such a script might be used to invok

in infected device.

ture validation where KernelSU assumes a V2 signature is present, but in reality, either a V1 or V3 signature migh

ernelSU.

may not be straightforward to exploit or may not impact all users.

could bypass the faulty signature verification logic and leverage the vulnerability to gain root privileges, potentia  
762932141cadd948c354f' includes the changes that fix the vulnerability.

understand the nature of the fix.

in privacy breaches. Since the exploitation requires user interaction, the attacker may need to deceive the user in  
l execution rights. Importantly, no user interaction is needed to exploit this vulnerability.

oid.com/security/bulletin/2023-10-01.

Application could perform the attack silently, potentially resulting in unauthorized actions or access to sensitive information. To incorporate the latest security enhancements and updates into their software distributions to protect devices against this weakness, users should ensure their devices are running the latest version of the application. This weakness allows an attacker, who controls a malicious MQTT broker on the same network as the device,

to manipulate the MQTT messages to execute arbitrary commands on the connected HMI device.

To deliver these specially crafted messages to the HMI device. Subsequently, the device will execute the commands contained in the manipulated messages. If an attacker sends manipulated messages to the HMI device, an attacker could manipulate the device's behavior in a way that serves the attacker's interests. As a result, the attacker could send deceptive messages to the HMI (Human Machine Interface) device, leading to

operational manipulations, or exposure of sensitive information.

Attackers to compromise the security of this communication channel by impersonating a legitimate broker. For example, if an attacker impersonates a legitimate MQTT broker, they could intercept and manipulate messages sent between the HMI devices and the broker, causing them to display incorrect information or execute unintended operations. For instance, in a vehicle context, an attacker could manipulate messages to cause a vehicle to perform unsafe maneuvers. To mitigate this risk, users should ensure their devices are running the latest version of the application and implement network security measures such as using VPNs, implementing robust DNS security, or network segmentation features.

Users should also ensure their devices are configured with appropriate security settings, such as disabling unnecessary services, modifying power management settings or enabling ADB debug mode, which could lead to further exploitation. Additionally, users should ensure proper physical security of devices, and configuring Kiosk mode settings to minimize the risk of unauthorized access. This weakness could be exploited in a malicious code-level attack. Attackers might use a sequence of gestures or inputs that the software has not adequately tested for, leading to unintended behavior.

The application uses the MQTT protocol, rather than HTTPS. As HTTP does not offer encryption, an attacker on the same subnet network could intercept and manipulate the communication.

5. With this information, the attacker could gain unauthorized access to the remote management system, modify network traffic for unusual patterns indicative of a man-in-the-middle attack. Users should refer to the vendor's documentation for more information on how to connect to a remote MQTT broker entity.

PS.

For more information, see the vendor's documentation. This weakness could be exploited in a malicious code-level attack, impersonate the server, disrupt the MQTT services, or even gain unauthorized control over the devices connected to the MQTT broker. To mitigate this risk, users should ensure their devices are running the latest version of the application and implement network security measures such as using VPNs, implementing robust DNS security, or network segmentation features.   
// Should use HTTPS // Code to send an HTTP request  
String url = "http://" + this.serverIp + "/getBrokerInfo";

are altered.

t provider. The app could change settings or data that the content provider controls, affecting the behavior of the ,  
to the utilization of a hardcoded RSA key pair, which enables the attacker to craft and send a valid but malicious b  
: traffic to a malicious server without the knowledge or consent of the user.

RSA key pair. When the Android Client application receives this manipulated intent, it may treat it as a legitimate  
e so: Intent maliciousIntent = new Intent(); maliciousIntent.setAction("com.example.androidclient.UPDATE\_SERVE

n-controlled phishing or malicious page. Since code specific to this vulnerability is sensitive information, a general

ility was patched in version 2023.9.2 of the app.

ers might also leverage the vulnerability to execute limited native code which could potentially compromise the u  
iation steps.

d leading to execution of unauthorized code. Developers would need to ensure that their WebView handling is se

d permissions, allowing them to access sensitive information or modify system settings without proper authorizat  
ny apps on an LG device.

ous app gaining access to internal functionalities of other apps that normally should not be accessible.  
urther exploits and access to sensitive operations.

f of the user without their knowledge, such as sending messages, transferring files, or modifying settings.

d even the possibility of code execution.

disruption, or unauthorized code execution.

that targets this component using the `android.intent.action.CALL` action along with a telephone URI to make a call.

`s://github.com/actuator/com.cutestudio.colordialer/blob/main/dialerPOC.apk`

nt with the `android.intent.action.CALL` action and a `tel:` URI containing a phone number, the malicious app could c

`blob/main/dial.gif`, and `https://github.com/actuator/com.full.dialer.top.secure.encrypted`.

ill being made, which could lead to unauthorized calls, potentially incurring charges or enabling eavesdropping.

`'/www.debian.org/security/2023/dsa-5499` and `https://lists.fedoraproject.org/archives/list/package-announce@`  
on the page, potentially luring the user into performing unintended actions, like entering sensitive information or  
atch notes, or security advisories for educational and patching purposes by developers and system administrator

`YAMPFV7IECQYGDEUIVVT/- https://www.debian.org/security/2023/dsa-5499- https://lists.fedoraproject.org/arcl`  
e. The information would be sent to the attacker instead of the intended legitimate service.

hing benign, while in reality, they might be giving away access to sensitive information or control over their device.  
.



71'.

Screen activity of wave.ai.browser, and since the URI and extra data are not properly checked, the JavaScript code as data theft, system control, or denial of service.

Y risk is augmented due to JavaScript execution being enabled in the WebView, which also loads web content directly.

This can lead to unauthorized actions such as accessing sensitive data, modifying system settings, or using the core 2470 These resources may include specific examples or POC applications that demonstrate how the vulnerability can be exploited.

This allows for remote code execution on the affected application.

An attacker fools the user into installing a malicious application that exploits this vulnerability, or it could exploit other vulnerabilities. For example, a simplified code snippet demonstrating this issue might look like this: `javaIntent intent`

to capture the intent and gain unauthorized access to specific files intended for the Camera app. This could lead to privacy concerns.

modify an Auto Hotspot setting without proper authorization.

Auto Hotspot setting without the user's awareness. For example, the malicious app could use something akin to `startAutoHotspot`

performing actions that they believe are in the legitimate app but are actually in the attacker's app. The user is actually interacting with the attacker's application, leading to potential unauthorized actions without the user's knowledge. The malicious app could also imitate the visual appearance of the WARP client to further deceive the user into giving sensitive information and a successfully installed malicious application on the user's device to be exploited.

A common way for attackers to deliver malicious software to devices is through social engineering.

The user's knowledge. When the user then interacts with what they believe to be the WARP client, they might, for instance, be tricked into installing a malicious application.

ers.cloudflare.com/warp-client/.

n likely targets individual users rather than mass infections.  
ating it, and responsibly disclosing information.

%84%E3%82%89%E3%83%BC%E3%81%8F%E3%82%A2%E3%83%97%E3%83%AA/id906930478), and the Japan V  
rg intents or URLs, an attacker might craft a malicious link that, when processed by the Skylark app, redirects the  
pear to be legitimate but would actually use the improper authorization flaw to bypass validations and redirect th

nter sensitive information which can then be stolen. Alternatively, the website could host malicious code intende

rmation with a trusted site while in reality, the data could be directed to an attacker-controlled site. This could res

. if the Autofill data contained sensitive information.

by Autofill. This could include personal data, login credentials, or payment details that the user had previously sav  
nts (<https://www.debian.org/security/2023/dsa-5479>), and Fedora package announcements (<https://lists.fedoraproject.org/>

lists.fedoraproject.org/message/OCFEK63FUHFZXH5MSG6TNQOXMQWM4M5S/- <https://lists.fedoraproject.org/>  
ving the user into believing they are on a legitimate site when they are not. This could lead to phishing attacks, st

heap corruption by using a crafted HTML page. This vulnerability has been given a high severity rating by Chromium

Android.

ment (<https://www.debian.org/security/2023/dsa-5479>), and announcements on the Fedora package announcement list (<https://fedoraproject.org/announce/2023/03/20230320-01.html>) and potentially allow the attacker to execute arbitrary code within the context of the browser or even lead to a browser crash.

system information. No code examples are provided as the issue pertains to permission settings in an application that is not part of the operating system.

security/advisories/GHSA-36f7-93f3-mcfj).

of its cache. The files could contain malicious content that compromises the integrity or confidentiality of the user data.

Unauthorized access to user data on different web domains, especially if combined with other browser-based attack vectors.

A maliciously crafted web page using special HTML, JavaScript, or other web technologies to bypass cross-origin policy.

authorized access to device functions.

Devices have up-to-date security software and follow best practices to guard against exploitation.

ORCPK4NNZNWJ/

data from other webpages viewed in the browser or enabling cross-site scripting (XSS) attacks.

and the flaw and ensure that security patches are effective.

exploitation such as phishing attacks or unintended disclosure of sensitive information.

on the package updates at <https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org>

option before the vulnerability was fixed.

credentials, session tokens, or other sensitive information.

, or identity theft.

It can access, install malicious apps, or read sensitive information from the TeleAdapt RoomCast TA-2400 device.

pull /data/data/com.example.app/` to exfiltrate sensitive data or `adb shell` to gain a command-line shell on the device.



23.02

ntially leading to unauthorized disclosure of memory contents or causing crashes and instabilities in the system.  
nt out-of-bounds operations. Examples of such code would include implementations of proper bounds checking a

oring mechanism that is supposed to filter out unacceptable content.

can limit the potential for exploitation by turning off the Bluetooth functionality when not in use or when in publ

asure of sensitive information that the user might otherwise notice and act upon.

interaction.

to read sensitive information from the system's memory, which can aid in further attacks or lead to privacy violat

essfully exploit this issue.

thereby gaining higher privileges than intended without additional execution privileges. Because user interaction is r

attacker to read sensitive information from the device's memory and potentially disclose it remotely.

calculations or failure to check the length of the input. Here's a hypothetical simplified C code snippet that might illu

incorrect calculation of an index or a failure to validate input data.

ounds read, leading to information disclosure, potentially leaking data such as user locations, credentials, or config  
ecessary for exploitation.

023-21222 vulnerability: missing bounds check } memcpy(buffer, data, data\_size); // potential out-of-bounds wri  
urther leveraged to move laterally within a network or to exfiltrate sensitive data. These scenarios highlight the ir

ary attacks based on the exposed data.

techniques. This could include personal data, login credentials, or other confidential information that the user ser

: the user's knowledge or interaction.  
use.

exploited. It does not require user interaction for exploitation.

: user's knowledge. Although no user interaction is required, it's important to remember that the attacker first ne  
raction is necessary for exploitation of this vulnerability.

ch mentioned in the bulletin.

educe its impact.

ight use this data for further attacks or to compromise user privacy. Since user interaction is not required, the att

require user interaction to be exploited.

a. The disclosed data could include personal user information, credentials, or other sensitive details stored in the c

ecessary for the vulnerability to be exploited.

the system. Since a code example for this specific vulnerability is not provided, general mitigation involves proper  
raction.



e access to, potentially leading to information disclosure or further exploitation to gain additional privileges on th

requiring any user interaction.

ation.

potentially gain enhanced system-level permissions without the user's knowledge, thereby compromising the inte

exploit this issue, an attacker would need System execution privileges, and no user interaction is required for expl

read memory out of bounds. This could potentially disclose sensitive information stored in memory, which the att  
at address the vulnerability. However, such information is generally intended for developers and maintainers to a

the system, thus leading to information disclosure.

s Android version 13.

data without any user interaction. This data could include information about the wireless networks the device has connected to.

There is no need for user interaction, possibly over a Bluetooth connection.

Local execution privileges. User interaction is not needed to exploit this vulnerability.

Out of bounds read. This can cause the Bluetooth service to crash and lead to a denial of service condition without the need for user interaction. The Bluetooth service fails to validate input properly, which might be exploited without any user interaction.

Notably, this exploitation does not require user interaction and does not need any additional privileges, making it easy to exploit. For more information, see <https://security.google.com/security/bulletin/pixel/2023-06-01> which should contain information on updates and mitigation steps.

This flaw can lead to unintended out of bounds memory access. Since the flaw can lead to information disclosure without requiring user interaction, there are no known workarounds.

without requiring any additional execution privileges or user interaction.

silently read sensitive information from memory and potentially disclose it to an adversary who controls the app.  
etermine if a patch has been released.  
s not require user interaction.

ince no user interaction is needed, devices are vulnerable unless the issue is patched.

evice's Bluetooth server without the need for user interaction.

aising the firmware with System execution privileges.

ie the incorrect bounds check in the vulnerable function of the Bluetooth stack to access data they should not be a

k. The specific exploitation mechanism is not detailed here, but typically, it would involve crafting specific Bluetooth version 13.

ttacker. The attack could be carried out remotely and would not require the user to take any specific action, incre

; device, leading to unauthorized access and manipulation of the system.

ger the vulnerability to escalate privileges and carry out actions on the device which would usually require user approval execution privileges.

ve information transmitted over that connection. This would likely require some level of user interaction, such as required for exploitation.

; a detailed attack scenario largely depends on the specifics of the logic error that aren't provided here.

itation of the affected system.

memory, which could be exploited similarly to how CVE-2023-21188 might be exploited: ````cppchar buffer[SIZE];in`

security bulletins and applying security patches promptly will help mitigate such risks.

are available as the specifics of the vulnerability have not been disclosed to prevent widespread exploitation.

otely, leading to a crash or impairment of the device's DNS resolution functionality.

led to exploit this vulnerability.

compromising the integrity of the device. As the vulnerability does not require user interaction to be exploited, a n

er to perform actions with elevated privileges on the device.

er.

NFC tags.

interaction, potentially accessing confidential data and leading to information disclosure.

without the user's knowledge. Since the exploitation does not require user interaction, the vulnerability could be exploited without user interaction, potentially leading to information disclosure or access to execution privileges.

is or out-of-bounds reads when writing code for parsing functions. This could lead to information disclosure or access to execution privileges, potentially through web services or email attachments that automatically process XML data.

execution privileges or user interaction. For example, they could tamper with system-level operations or access sensitive data.

uch an attack could expose sensitive data or credentials stored on the device.

e example. Typically, race conditions can be mitigated with proper synchronization mechanisms such as mutexes  
ninteraction for exploitation.

ormation if the app in use is meant to be private or revealing user habits.

leges.

or gaining further access to the system.

additional execution privileges. The affected product is the Android platform, particularly version Android-13.

egitimate users or be part of a broader attack to manipulate the device's communications or perform data exfiltration.

s would review the onCreate method logic to ensure proper permissions are enforced and guest users are blocked.

d access without user knowledge.

ier interaction is needed for exploitation.

ser interaction. This could lead to the exposure of sensitive user data or network usage patterns.

or without the knowledge of the device owner, leading to unauthorized control over call settings and potentially :

Remarkably, no user interaction is necessary for an attacker to exploit this flaw.

the user's knowledge, potentially leading to a local escalation of privilege.

rogram does not read outside of the intended memory area.

ired for the exploitation of this vulnerability.

uld read memory out-of-bounds. This could lead to information disclosure, such as sensitive user data or system i  
the issue to understand the precise nature of the bounds check that was missing. The code fix might involve addir



cution privileges or user interaction.

This disruption could persist, resulting in a local denial of service until the issue is rectified. This kind of attack aim

inds, potentially overwriting important system data or code. This could be used to gain higher privileges or execut

: within the context of the system, leading to a complete system compromise.  
to be exploited. It affects Android devices and does not require user interaction to be exploited.

ould lead to information disclosure, potentially exposing sensitive data from the device. As the exploitation does r

ce. This could compromise the integrity and security of the system without requiring any user interaction.  
set(buffer, 'A', 15); delete[] buffer;}```In this simplistic example, writing 15 'A' characters into a buffer that can only  
; to be exploited. This vulnerability affects the Android operating system.

nal execution privileges, and it does not need user interaction to be exploited.

ttacker without requiring additional privileges.

e intended boundary, possibly disclosing sensitive information that resides in the adjacent memory locations. For

ure without the user's knowledge.

leges required. The user interaction is not necessary to exploit this vulnerability.

ccess to user data or the ability to execute commands with elevated privileges.

nessed by the malicious app to read sensitive memory locations or manipulate the device's operation, ultimately  
to exploit. This affects various Android kernel versions.

ity in FaceStatsAnalyzer::InterpolateWeightList to read unauthorized memory areas, potentially leading to the disclosure of this flaw without any user interaction.

to memory and potentially allow the attacker to execute arbitrary code or induce a system crash, leading to a local denial of service.

to an attacker. The necessary System execution privileges means this attack could potentially compromise the integrity of the system. There is no need for user interaction to exploit this vulnerability.

services, potentially leading to information disclosure, service disruption, or other harmful consequences.

vided.

or their specific device and version to determine if they are affected.

to privilege escalation.  
Security updates relevant to the vulnerability.

by user interaction.

ion.

sed website, or through a network-based vector. Upon successful exploitation, the attacker would have the ability  
on. It affects Android version 13.

attacker's application. This kind of attack, known as tapjacking, can occur without the user's knowledge, as it ex-  
data, the disabling of security features, or the manipulation of system settings and applications.  
missions or performing sensitive actions without their knowledge. However, for security reasons and considering  
. This makes it possible for unintended content to be loaded in the context of the app, either through the fcrbs sch  
in.html).

up. The attack could be initiated by tricking a user into clicking a crafted link or through another app on the same c-  
s code. This intent is then started, resulting in the execution of the malicious code within the app's webview. As b-

sensitive information. In addition, the attacker could traverse the app's directory structure and write arbitrary files,

as or other security incidents.

s would generally be removed during the remediation process, and best practices such as using environment variables, access, manipulation of system configurations, or other malicious activities within the affected system that recc and Focus for Android versions prior to 112.

y exposed.

id uncover sensitive information such as URLs of privately visited websites, search queries, and any other input pr

r a third party. The issue needed to be resolved by the Firefox development team and addressed through a softwa pps to deceive users.

ieving they are still interacting with their browser or device UI when in fact they are interacting with the maliciou g unaffected by this vulnerability.

Android browser, leading to unauthorized disclosure of information, manipulation of data, or disruption of service. I refer to the official security advisory and Bugzilla report provided in the references. Documentation for how use- ly enable a bad actor to wipe personal data and applications, thereby disrupting the operation of the user's device

g in a loss of data and potentially leaving the device in an unusable state. user interaction.

helming the phone's resources, thereby making it unresponsive.

on the device without requiring additional execution privileges. No user interaction would be needed to exploit 1

responsive or unusable. This attack could be performed by an attacker with physical access to an unlocked device or a device that is in a state where the functions are not fully operational, and it might depend on the particular nature of the functions affected and the manner in which they are exposed.

An attacker may need to find a way to enable it or leverage another vulnerability to switch the device to developer mode.

Access to user data or system logs that could help in further attacks.  
The issue could be exploited without user interaction, granting additional execution privileges.

The issue could potentially lead to data theft, system manipulation, or the silent granting of access to sensitive functionalities.

Background activity launches with user execution privileges, meaning a malicious actor could gain elevated access to system resources.

The attacker could gain unauthorized access to system resources or sensitive information.  
The issue could be exploited without user interaction, allowing the attacker to execute operations in the background with elevated privileges without the user's consent or knowledge.  
This could be necessary for exploitation.

The issue could lead to a persistent denial of service state, where the affected component becomes unavailable or unresponsive.

to an app that uses JobStore functions, causing the device to crash during its next startup sequence. As this vulner

1 changes to notification settings and potentially gaining unauthorized access or permissions on the device.  
the ability to launch arbitrary activities in the Android settings without requiring additional privileges. The affecte

uire user interaction, the attack could occur without the knowledge of the user, leading to unauthorized actions a  
bility affects Android version 13.

bitrary code on the device, potentially giving the attacker the ability to control the device, access sensitive informa

allowed. This could lead to unauthorized access to certain privileged actions or information on the user's device

compromising the device integrity.

opening of a malicious attachment or downloading content from untrustworthy sources.

requiring additional privileges or user interaction, thereby gaining elevated permissions on the affected device.

result, a malicious actor could silently trigger an escalation of privilege, enabling them to execute privileged actions

and ensuring they come from trusted sources before they are processed, thus preventing the execution of arbitrary code. This vulnerability can be exploited without any user interaction.

a. As this does not require user interaction, the vulnerability can be exploited silently. Enabling an attacker to gain

to prevent misuse. However, developers can analyze the vulnerability further by checking the related Android code to ensure that a local attacker cannot gain higher privileges than intended by the system's security design.

needing additional execution privileges, and user interaction is not necessary for its exploitation. The affected versions

abuse this vulnerability to gain privileged access to system resources or sensitive information, potentially leading to



without the user's knowledge.

- o gain unauthorized access or control over the device's resources or data.

user interaction is necessary for exploitation.

- o additional execution privileges and user interaction is not necessary for its exploitation. As long as Bluetooth an

e sdp\_discovery.cc component, potentially leading to remote code execution. This could let them seize control of exploitation of this loophole.

- o breach of privacy. The attacker could exploit this vulnerability to gather personal photos, videos, or documents s

an access to sensitive data, bypass security restrictions, or cause denial-of-service conditions.

operation for exploitation.

Physical access to a vulnerable device might unlock certain functionalities or access sensitive information that should be protected. In some cases, updates are pushed without exposing potentially dangerous code to the public.

```
FileBackedOutputStream outStream = new FileBackedOutputStream(1024); // If sensitive data is written to the file, it could be accessed by an attacker, possibly leading to information disclosure, data tampering, or unauthorized actions if the files are used for critical operations. This is a the-middle attack and eavesdrop on encrypted communications.
```

```
state, the attacker could intercept and decrypt the data being transmitted, gaining access to confidential information. (me, IBinder service) { // Setting up the SSL context using an insecure TrustManager SSLContext ctx = SSLContext.getInstance("SSL");
```

Since code examples for this specific vulnerability and attack are not provided due to the nature of exploitation and the complexity of the attack, the following code examples are provided for educational purposes only.

se the Yandex Navigator app to malfunction, crash repeatedly, or become inoperable, leading to a denial of service.

ry data, leading to a persistent denial of service where the application cannot function properly or at all until the crash is fixed. If an attacker can trigger a crash before performing database file operations, an attacker could exploit this to overwrite or corrupt database files.

grant themselves higher access rights or execute actions that are normally restricted to certain user levels.

service for users trying to utilize the Sleep app.

```
);editor.putString("last_sleep_time", "00:00");editor.apply();``In this example, the app stores 'last_sleep_time' with the value '00:00'.
```

ing the app as intended.

Light Filter app. Typically, attackers would use Android's API to access and modify the preferences, likely requiring root access.

thereby escalating its privileges beyond what is normally allowed.

attack by modifying the stored preferences in a way that is not intended or expected by the Blue Light Filter app. To prevent similar vulnerabilities, developers should implement proper input validation and sanitization.

ed on security best practices to safeguard against similar escalation of privilege attacks.

istent crashes or become unable to function properly, which disrupts the user's experience.

, the malicious app could potentially alter those files to elevate its privileges or access sensitive information with  
`prefs = context.getSharedPreferences("VictimAppPrefs", Context.MODE_PRIVATE);`  
`SharedPreferences.Editor edito`

urther exploitation of the system.

be restricted, leading to a compromised user experience and various security risks.

Android devices are particularly at risk.

r dishonestly, potentially requiring root access or exploiting another vulnerability that grants file system access.

... some other code ... `free(provider);`};  
`int main() { char *provider = (char *)malloc(100); register_provider(provider);`  
execute arbitrary code. The specific attack vectors would depend on the broader context of how the vulnerable fun

L9486%2C1819492%2C1819957%2C1820514%2C1820776%2C1821838%2C1822175%2C1823547.

on the user's device, leading to unauthorized access, information theft, or further system compromise. Since accu  
offer technical details necessary for security professionals to understand the vulnerability and develop protection:

%2C1819493%2C1820389%2C1820602%2C1821448%2C1822413%2C1824828- <https://www.mozilla.org/security/adv>

rine within the context of the affected application. An attack could result in taking control of the affected system, ypass.

ially leading to leakage of data or functions into an unintended realm, which a malicious actor could exploit. on disclosure, privilege escalation, or other security compromises. This could be particularly dangerous if the san

ave unexpectedly, potentially leading to arbitrary code execution or information disclosure within the context of t

ity, or manipulate web sessions to gain unauthorized access to sensitive information. is and not code that can be written or executed by a user or developer.

re browser, results in the exploit. A successful exploit could lead to a crash and may allow the execution of arbitra the potential impact of the vulnerability, which could lead to arbitrary code execution on an affected system.

. Such an attack could lead to compromised end-user security, unauthorized access to sensitive data, or takeover

can make Firefox run a maliciously crafted .desktop file, it could result in running arbitrary commands.

cutes the .desktop file, it could run the attacker-specified commands, leading to unauthorized actions on the user's device without the required `allow-top-navigation-to-custom-protocols` attribute.

to initiate downloads of malicious software, or to perform other harmful activities.

exploitation.

v\_bug.cgi?id=1790542`.

interaction to exploit the vulnerability.

harmful behavior on the user's device. For example, the iframe could be redirected to a custom protocol that initiates

ies/mfsa2023-15/.

different extension or name than intended. The attacker could trick the user into believing they are downloading a harmless

file. The vulnerability might ignore everything after the NULL character (\x00) and only show 'safe\_document.pdf' as the

structure and potentially sensitive data on the user's device.

ion, leading to arbitrary code execution in the context of the application. As part of responsible disclosure, code execution

memory, leading to an assertion, memory corruption, or a potentially exploitable crash.

?id=1821959- <https://www.mozilla.org/security/advisories/mfsa2023-15/>

memory pointer. This could result in various outcomes such as executing arbitrary code, causing the application to crash, or redirecting the user to 102.10 or later.

ed to 102.10 or later.

arbitrary code on the user's machine or cause the browser to crash.

`window.name` assignments, and `setInterval` calls. This could lead to user confusion and potential spoofing attacks.

13/- [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1798219](https://bugzilla.mozilla.org/show_bug.cgi?id=1798219)- <https://www.mozilla.org/security/advisories/mfsa2023-01> screen mode, thus becoming susceptible to spoofing attacks. However, without the specific code that caused the issue, an attacker could impersonate the user's desktop or a trusted website, and the user might be tricked into entering sensitive information.

ceptive content, potentially tricking the user into providing sensitive information or downloading harmful content. Mitigating this issue requires an external application to mitigate this issue. This vulnerability affected Firefox versions prior to 111.

ore launching the application, the attacker could exploit the vulnerabilities within the unintendedly launched application.

of personal information.

w\_bug.cgi?id=1798798', giving more context to the issue.

When the user visits the attacker's page, the prompt would cover the notification bar, potentially making the user unaware of the risk.

reby increasing the risk of personal data exposure or account compromise.

[github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29746/CVE%20detail.md](https://github.com/LianKee/SO-CVEs/blob/main/CVEs/CVE-2023-29746/CVE%20detail.md)), and on the Google Play Store. This vulnerability could lead to unauthorized access to the user's data. For instance, an attacker could inject malicious code into SharedPreferences files that is not a valid Android API for storing key-value pairs, the attacker could change values to contain a payload that, when read, could cause the application to crash due to an Out of Memory (OOM) error, leading to a persistent denial of service.

It could be loaded into the application's UI. As the amount of injected data accumulates, it could overflow the available memory, causing the application to crash. The application is available on the Google Play Store (<https://play.google.com/store/apps/details?id=com.cuiet.blockCalls>), and an APK download link for the affected application is provided below.

privilege attack.

Access to.

It could be executed with higher privileges when the wallpaper app is opened.

Sources.

Code audits and security testing can help identify and fix potential security issues before the app is released.

By crafting the content of the overwritten files, the attacker could achieve arbitrary code execution, potentially leading to device compromise.

The application crash, resulting in a persistent denial of service attack.

For example, an adversary could use code similar to this: `Intent attackIntent = new Intent('com.glitter.unicornwallpaper.ACT`

It could be executed with higher privileges when the app is opened. An attacker could exploit this vulnerability to tamper with the data, potentially leading to device compromise.

By loading the tampered data into memory. By altering this data, the attacker could potentially execute code with higher privileges than they shouldn't have access to. An example could involve inadequate checks on 'ContentProvider' permissions.

The application's startup, can cause an Out of Memory (OOM) error and crash the application, leading to a persistent denial of service attack.

The application's startup, can cause a memory overflow, resulting in an OOM error that crashes the application. As a result, every attempt to start the application results in an OOM error upon startup.

For vulnerabilities. Additionally, developers can use ProGuard or similar tools to obfuscate the code to make it more difficult to analyze.



```
ces prefs = getSharedPreferences("MyPrefs", Context.MODE_PRIVATE); SharedPreferences.Editor editor = prefs.e
```

apps/details?id=com.TheThaiger.android - <https://play.google.com/store/apps/details?id=com.icoolme.android.w>  
sh or fail to function properly. Since the issue allows unauthorized apps to interfere with the database, it's likely t  
or other techniques to insert, delete, or modify the contents of the database. The attack could involve code that i

on the device.

ze input correctly to prevent such exploitation.

ld potentially exploit this vulnerability.

velopers for updates regarding this security issue.

on properly, leading to a persistent denial of service. For example, if the weather data is corrupted or made inaccur

ly leading to data leaks, unauthorized actions, or compromising the integrity and confidentiality of the data mana  
[Es/blob/main/CVEs/CVE-2023-29741/CVE%20detail.md](#).

the database, leading to a denial of service. This vulnerability takes advantage of the app's improper handling of

operation of other applications.

[wavekeyboard.com/](https://wavekeyboard.com/)

allow the attacker to gain unauthorized access to sensitive data or system resources and potentially compromise the

the app and the device.

regarding this issue.

. The official website for Call Blocker: <https://www.call-blocker.info/>

, or even disabling certain security measures. As a result, the attacker could gain unauthorized control over the device or even disable certain security measures. It's best to refer to the official sources or trusted security platforms for any code examples or analysis related to this issue.

'[CVE-2021-26858](#)detail.md), and the application's official website (<https://www.call-blocker.info/>).

exploit this vulnerability to perform an escalation of privilege attack, potentially gaining unauthorized access to the device's data.

designed to manipulate the app's database and delete or alter the data related to user privacy settings, which could result in a persistent denial of service.

When the application attempts to load this excessive data into memory, it would trigger an Out of Memory (OOM) error and crash the application.

t.blockCalls- The official Call Blocker website: <https://www.call-blocker.info/>

exploitation could result in the application failing to start or crashing during operation, thus denying service to legitimate users and applications.

protected and restricted from such applications.

and apply the principle of least privilege to apps installed on their devices to limit the potential impact of such vulnerabilities.

ulate the app's functionality, gain access to secured areas, or disable important security features, leading to an escape

files is loaded into memory. This might result in several types of attacks depending on how the tampered data is used

give users or facilitate further attacks, or even more severe consequences if SharedPreferences data is used for security. An OOM error and causing the application to crash at startup. This creates a persistent denial of service situation. An OOM error of attack results in a persistent denial of service because the app crashes every time it is launched. Denial of service on the app.

In the app, the application tries to load this tampered data into memory, resulting in an OOM error as the application cannot

of service. For instance, if the 'insert' function does not validate the size or content of the input before processing

either through physical access or through another app with permissions to access the app's database files. Proper permission and access controls are in place to mitigate such vulnerabilities. Without the parents' knowledge.

For example, Once in Safe Mode, the child navigates to the application settings and disables the 'Display over other apps' permission. Safe Mode on Android devices. External storage is not accessible to any component with permission to read the external storage.

the external storage, the files could be accessed by other applications or entities with permissions to read from the storage. Using the 'rand()' function. This fallback can be problematic because 'rand()' does not use a cryptographically secure

random number generation. For more information, see the Gentoo Security Guide at '<https://security.gentoo.org/glsa/202310-09>'. Since, if c-ares is used in a security context where randomness is essential, the predictable nature of 'rand()' could be a factor, such as DNS spoofing or session hijacking. Due to CVE-2023-31124, the following code snippet shows a vulnerability: 'rand()

could result in a privacy breach or further attacks if the exposed data includes credentials or personal details.

oper permissions.

VNVU97891206/

her iPrint&Scan app, potentially revealing sensitive information in a print preview format. It could be used for further misuse of the vulnerability.

Debian, Fedora, and Gentoo found at their respective security announcement pages. This vulnerability could be exploited by an attacker to compromise the security and stability of the affected system. The vulnerability is located in the c-ares library, potentially allowing them to execute arbitrary code on the victim's device. However, the details and steps to exploit this vulnerability are not provided.

If needed, the application could potentially access and disclose sensitive information from the device's memory without requiring additional execution privileges. The vulnerability affects Android version 13.

re. The exploit could run in the background, without the knowledge of the user, leading to unauthorized access to the permission checks within this method to ensure that only authorized callers can register a broadcast receiver and execution privileges required. The affected Android versions are Android 11, 12, 12L, and 13.

the user's knowledge. This could then be used as a foothold to carry out further attacks, such as gaining persistent access to a system app and execute a privilege escalation attack.

the Android device.

rvices.

necessary for exploitation.

ed by the Android security team within the proprietary PhoneAccountRegistrar.java file.  
versions 11, 12, 12L, and 13.

or a code example, one would need to review the Android Open Source Project (AOSP) repositories for the affected versions to identify the race exhaustion issue. This could allow the malicious app to gain unauthorized access to the notifications systems and potentially other sensitive data.

cts Android versions 11, 12, 12L, and 13.

consent or knowledge. This could lead to data theft, unauthorized actions, or further exploitation within the system.



bulletins could give insights into what was affected and how it was corrected.

ation of this vulnerability does not require user interaction.

ous app exhausting resources without any specific permissions or user interactions, taking advantage of this vulnerability to update their systems accordingly to protect against such vulnerabilities.

This vulnerability requires User execution privileges to be exploited, and user interaction is not needed for exploit

Unauthorized access to sensitive information such as OTPs, authentication messages, or private conversations, which

security measures, it could compromise the entire device.

ample might involve sending a specially crafted request or data packet that the application, due to inadequate input

d potentially causing a denial of service.

ed, leading to information disclosure. For example, an attacker might exploit the vulnerability to access confidential data. / Pseudocode for improper access checking in an Android application

```
public class SensitiveDataHandler {  
    private
```

application unresponsive or significantly slow down its performance, impacting the availability of the service for legitimate users. Exploitation of this vulnerability would be highly tailored to the internals of the Intel

processor. This could include accessing sensitive data, such as personal user information or retail insights, which should be protected.

The exploit could slow down the device or make it unresponsive, resulting in a denial of service condition. This vulnerability should be patched by Intel and its partners, and security professionals and software developers for the purpose of understanding and patching the vulnerability.

Actions such as accessing sensitive data, changing system settings, or installing unauthorized software.

Unauthorized content.

Exploitation of this vulnerability could lead to the theft of session tokens, credentials, or other sensitive information. It could also be used to perform actions such as displaying a JavaScript alert box. In an actual attack, the script would likely be crafted to perform more malicious actions, such as

Interfering with program events or program stages.



are not authorized to access, particularly if they use the Android Capture App, which displays the compromised data.

t. Samsung has acknowledged this issue and provided a fix in their SMR May-2023 Release 1.

leading to information disclosure and privacy breaches.

d incorrectly in the AndroidManifest.xml file, allowing other apps or attackers physical access to invoke it without  
uld have been protected in the application's sandbox environment.

e. The attacker might manipulate the Intents or similar mechanisms used by the application to perform this unauth

672.63.

rm actions without the user's knowledge or consent. Such actions could include tricking the user into disclosing se  
stable-channel-update-for-desktop.html), the Debian security advisory (<https://www.debian.org/security/2023/d>

ge.

: the victim to this page through phishing or other methods. Once the Omnibox is hidden, the attacker could then

erver by using a fraudulent certificate that the application fails to properly validate, which could deceive users into

ons leading to elevated privileges on the device, bypassing Android's security mechanisms, and gaining access to :  
uired to exploit it.

ith system-level processes, leading to a compromise of the affected device's security.

to exploit this vulnerability.

evice without the victim's knowledge, potentially gaining access to sensitive information or system settings.  
issue affects versions Android-11, Android-12, Android-12L, and Android-13.

iple is not available, as it typically requires an in-depth knowledge of the Android operating system internals and t  
information and patches can be found on the Android Security Bulletin.  
n privileges. This issue affects Android versions 11, 12, 12L, and 13.  
ising the device's integrity.

content with fraudulent information or phishing overlays to trick the user into entering sensitive information or could also be used to review installed applications and privileges and avoid installing apps from unknown or untrusted sources. Without additional execution privileges.

If additional execution privileges or user interaction are needed, the exploitation could happen transparently, resulting in the application accessing a file path outside of its designated directory, potentially accessing or modifying files that should be inaccessible. Exploitation of privilege without needing additional execution privileges. No user interaction is required to exploit this

If exploited, the app could receive broadcasts intended for system apps and potentially access or manipulate protected system files or perform privileged operations without the necessary user permissions.

Users should be advised to review installed applications and maintain good security practices to minimize the risk of installing potentially malicious apps that

can be exploited across user boundaries and does not require user interaction to be exploited.

This could interfere with other applications that rely on predefined locale settings, ultimately leading to a localized denial of service.

resulting in a persistent local denial of service.

This issue affects Android versions 11, 12, 12L, and 13, as identified by the Android version string. No additional execution privileges. This issue affects Android versions 11, 12, 12L, and 13, as identified by the Android version string.

This could result in the application performing privileged operations or access resources on the device for an extended period, potentially leading to the compromise of the device.

the restrictions intended to limit such actions when an app is not actively being used. This could give the attacker a patch, providing a concrete example is not possible. Once the patch is released, developers can review the chan

bled from a secondary account due to a permissions bypass. This flaw could lead to a local escalation of privilege f  
d use it to pair with nearby NFC-enabled devices or read NFC tags containing sensitive information or malicious lir

nds write vulnerability. This can potentially lead to arbitrary code execution on the device without any user intera  
its. No user interaction is required to exploit this flaw, and it affects Android version 13.

tially escalate privileges without the user's knowledge.

losure of sensitive information.  
the source files mentioned.

ly compromising the security of the device and the user's data.  
droid source code in PackageInstallerService.java and related files, where a logic error allows bypassing backgrou

cessitating additional execution privileges. It affects Android versions 11, 12, 12L, and 13.

elevation of privileges within the Android system. The specific technical details or a code example of the exploit f

: the user's knowledge, allowing for additional malicious activities such as data theft, system monitoring, or furthe

ng to a local escalation of privileges on the device.

ation privileges.

missions. The attacker could then potentially access sensitive information, manipulate system settings, or compromise system integrity, all of which are necessary for exploiting this vulnerability.

and disclosing it to an attacker, all without the user's knowledge or consent.

described in the referenced security bulletin.

out-of-bounds heap access, thereby gaining escalated privileges on the device without needing any extra execution permissions. No user interaction is necessary for an attacker to exploit this vulnerability.

This vulnerability could be abused to read sensitive information, manipulate system functionalities, or even install malware.

without requiring any additional execution privileges.

quired, the exploitation could happen transparently to the device owner.  
ntentially allowing the attacker to execute arbitrary code with elevated privileges.

.  
on of privilege without requiring additional execution privileges, and user interaction is not necessary for exploita  
ilege. The flaw affects Android devices running specific versions of the Android SoC.

ify memory in ways that could lead to escalating their privileges on the system. Since no additional execution priv  
em affects Android SoC versions and has been assigned a HIGH severity rating with a base score of 7.8.  
ution privileges or user interaction, meaning a malicious app could leverage this flaw silently to gain higher privile

action for exploitation.

cker higher system permissions without the need for additional execution privileges.  
ocal escalation of privilege without requiring any additional execution privileges and does not need user interacti

grity.

ce, installation of unauthorized software, or access to confidential information.  
privileges, and no user interaction is required for exploitation.

the heap. Exploiting this access, the attacker could modify critical data structures or execute arbitrary code with elevated privileges by an attacker to escalate privileges locally on a system without needing additional execution rights.

access to areas of memory that should be inaccessible, allowing them to execute arbitrary code with higher privileges.

Finally, leading to local privilege escalation without needing any user interaction or additional execution privileges. The input validation is improperly validated, leading to an overflow when large or specially crafted input is provided by an attacker, resulting in additional execution privileges.

software.

Finally, exploiting the flaw could allow the attacker's application to gain higher privileges on the device, potentially leading to a local escalation of privilege on an affected system without needing any additional execution privileges.

```
if (buffer != NULL) { // ... operations potentially leading to heap overflow ... } free(buffer);
```

In this example, since no user interaction is needed, the exploit could be performed silently without the user's knowledge.



igned a high severity score of 7.1.  
nponent of the DUALSPACE Lock Master application.

uptodown.com/android/download- The official website for DUALSPACE which offers the product: <http://www.dualspace.com>  
; of the exploit and potential proof-of-concept code.  
ior that crashes the app, leading to a denial of service. Alternatively, the attacker could extract sensitive information

89b1494s8RF8ksaOijKhKb-3B8gj3pLUmgn0dqg-jqs/edit  
d Android environment. This could potentially give the attacker access to functionalities that are normally restricted

oogle.com/store/apps/details?id=com.kyocera.kyoprinttautax- <https://play.google.com/store/apps/details?id=com.kyocera.kyoprinttautax>

mising the device's security. The attacker could leverage this to steal sensitive information, gain unauthorized access  
to perform unintended actions, such as downloading malicious files. It is vital for security reasons not to share exploit

703

Without access to the source code or specific debug output examples, it's not possible to provide exact code examples  
h as identity theft, selling the data, or launching further attacks based on the gathered information.

lls or the API provided by the Gralloc module, not through typical programming code.  
s obtained.  
al channels provided by Arm.

accessing private data, causing financial loss, or disrupting service operations.  
; included in the publicly available application package. Additionally, API keys should have scoped access with the

ictions, potentially leading to further exploitation such as phishing or redirecting to other malicious sites. Protect  
t, due to the navigation restriction bypass, it could redirect an unsuspecting user to a phishing site, effectively ste

4-8875-wxww-3rr8- <https://github.com/nextcloud/desktop/pull/5560>  
compromising the confidentiality and integrity of the stored data.  
ures or a flaw in the client's handling of encryption keys that a server can exploit.

d access file metadata like who shared the files, who the files were shared with, and the file activity, without the c

ing sensitive information.

ected product is the Android kernel.

allowing the attacker to take complete control over the affected device without the user's knowledge.  
s without the need for user interaction.

ileges on the device. As the vulnerability could be exploited without user interaction, it may be used as part of a  
xploitation.

nt to the buffer being overflowed, allowing the attacker to execute arbitrary code with system-level privileges. Sir

r the affected system or manipulate it without the user's knowledge.

eges required. This issue affects various versions of the Android kernel.

tem privileges, further escalating their control over the device without the user's knowledge. Since specific code c

ories.

ilable security patches from Android or their device manufacturer.

```
har buffer[BUFFER_SIZE];void update_buffer(char *input, size_t input_size) { if (input_size > BUFFER_SIZE) { // F
```

nselves to exploit the elevated privileges without needing any additional execution privileges.

orecise account of potential attack vectors.  
vulnerability.

ounds or alter memory unexpectedly, thereby leading to privilege escalation. The attacker could then use the esc

the app to carry out further malicious activities, such as intercepting sensitive information or taking control of the  
itory for commits that fix the issue, which may show what changes were made to mitigate the vulnerability.  
kernel, and user interaction is not required to exploit the flaw.

horized actions being performed without the user's awareness.  
itories, and databases should be conducted.

ly, ultimately compromising the integrity, availability, or confidentiality of the device.

iation from the victim's device memory without the user's knowledge or consent.  
or their device manufacturers for specific version information.

d then potentially access sensitive information from the device's memory without requiring any user interaction c  
es not read beyond the allocated memory. Regular code audits and employing memory-safe programming techni

rary code on the system.

ition on the device without requiring any user interaction.

out any user interactions. The attacker could then gain the capability to modify system components, access sensit  
ias been released to ensure responsible disclosure and prevent exploitation in the wild.  
necessary for exploitation.

s, further compromising the device's security.

g a system-level process or using another system-level application vulnerability to execute code remotely, thus ta  
nay occur. Typically, this might involve writing data beyond the allocated buffer sizes or not handling the data size

and disclosed, possibly revealing sensitive data from the device's memory to the attacker.

y.

erability to overwrite memory and potentially execute arbitrary code, leading to an escalation of privilege on the

the Android kernel source code.

g incorrect bounds check to overwrite kernel memory, potentially leading to arbitrary code execution within the k

d for exploitation.

/ allowing the attacker to execute arbitrary code with increased privileges on the device.

g System execution privileges. No user interaction is needed for the exploitation of this vulnerability.

sensitive details stored in the device's memory. The attacker could then potentially use this information for further execution privileges, and no user interaction is necessary for exploitation. It affects specific versions of the Android

disclosure of sensitive information or other impacts that are less than critical.

It updates provided by their device manufacturer to protect against known vulnerabilities.

It could lead to the disclosure of sensitive information stored in the device's memory, such as passwords, cryptographic

This app with the necessary privileges could automatically exploit this vulnerability without the user's knowledge.

A mitigation for this kind of issue could be as follows: 

```
void ConvertToHalMetadata(size_t bound) { if (index < bound) {
```

 The attacker would need System execution privileges and no user interaction is required.

such as user data, credentials, or system configurations. Since no user interaction is required, the attack could happen without local information disclosure without requiring any user interaction and necessitates System execution privileges.

on or further exploitation of the system.

It silently once the attacker has the required level of system access.



1.

carried out silently, potentially leading to the compromise of the entire system.

r' is now dangling // Simulated malicious use after free struct data \*new\_ptr = (struct data \*)malloc(sizeof(stru

hands on the device at the system level, leading to local escalation of privilege.

er to modify memory and gain higher privileges on the device.

ided information.

arbitrary code or alter memory they should not have access to, ultimately leading to a local escalation of privilege

or the exploitation of this vulnerability.

de with system privileges without any user interaction, leading to full device compromise.

from the intended security policies of the system. Since the vulnerability is considered medium severity, the impact

ssions as the legitimate app without the user's knowledge, leading to local escalation of privilege.

n 13.

the user's knowledge, leading to further malicious actions.

ons that normally require higher privileges. For instance, the app could log the device's movement patterns or ma

he device is restored, which would disrupt the user's ability to connect to any wireless networks.

updates periodically, and it would be advisable to check for and install these updates to protect against exploita

. Such an attack could leverage the vulnerability to read sensitive data from the heap that should not be accessible.

everage this vulnerability by crafting a malicious application or environment that initiates the race condition to gain access to sensitive data. n 13.

could exploit this flaw to perform actions with higher privileges than intended, potentially leading to unauthorized access to sensitive data. 3-21030 would be included in the security patch details provided by the Android Open Source Project (AOSP) or device manufacturer.

information about other app usages on the device without the user's knowledge or consent.

s behavior to receive sensitive data from the device's memory without any user interaction.

teractions.

n a denial of service where the user cannot access certain functions of their device.

and a malicious actor would need System execution privileges to exploit it. The affected product is Android, partici

y areas it should not have access to. This data could include keys, passwords, or other confidential data, which co

ilege. Since the vulnerability does not require user interaction, it could be exploited by malware already present

m resources or sensitive information, bypassing the security measures that normally isolate application data and  
l the allocated memory block, for instance:

for exploitation.

curity patch or mitigation strategy.

to network resources or the ability to manipulate network configurations to their advantage, leading to a local escalation of privilege with System execution privileges. No user interaction is required for the exploitation of this vulnerability.

by user intervention. This can potentially allow the attacker to gain further control over the device and escalate their privileges. Researchers are currently working on patching the vulnerability. No user interaction is necessary for the exploitation of this vulnerability.

It can also access passwords, personal data, or other confidential information stored in memory.

items.

It has been freed, leading to memory corruption and potentially allowing the attacker to escalate privileges without user interaction.

It can also silently, leading to elevated privileges for the malicious app without the user's knowledge.

Additional execution privileges.

It can also access system interfaces, leading to a local denial of service. Since the vulnerability can be exploited without user interaction, it is a high priority for researchers.

needing additional execution rights. Furthermore, user interaction is not necessary for an attacker to exploit this vulnerability.

If the attacker obtains the necessary privileges, the attacker could access or manipulate protected system functionalities or data, potentially leading to sensitive information disclosure.

The vulnerability could potentially be used to further compromise the affected system or reveal private user data.

Without spreading exploit techniques. For educational purposes on protection and understanding vulnerabilities, remote interaction is not required for exploitation.

Compromise the system or the user's data.

An attacker can remotely compromise a system service or exploit another vulnerability to achieve the necessary privileges, leading to a full system compromise.

al, the attacker could potentially access sensitive information from the system memory, leading to information disclosure or buffer length before attempting to read data from it. No execution privileges required. No user interaction is necessary for exploiting this vulnerability.

the background, making detection harder for the user.

access to, possibly disclosing information such as user data or system properties which could be used for further escalation of privileges, and no user interaction is required for the exploitation.

ected. Such information could include private user data, credentials, or system configurations, which could then be used for further escalation of privileges. This is a memory access vulnerability that exceeds its declared bounds, possibly due to missing or inadequate bounds checking in the code. An example of a memory access vulnerability, and no user interaction is necessary for exploitation.

to read memory it shouldn't have access to, potentially revealing sensitive information.

privileges.

the user's knowledge, potentially compromising the security of the device.  
es. No user interaction is required to exploit this vulnerability.

privileges. With the elevated privileges, the attacker could access or modify sensitive data, change system settings, associated with this vulnerability.

a malicious app running on the device could silently gain escalated privileges and carry out actions that would oth  
on privileges. The exploitation of this vulnerability does not require user interaction.

er exploitation of the device.



In the context of Android's MediaCodec.cpp, it would involve an incorrect sequence of locking and accessing free

the malicious app would need to be installed and executed by the user, after which it could exploit the vulnerability; vulnerability, a malicious actor would need User execution privileges. No user interaction is necessary for exploit

amples for vulnerabilities to prevent facilitating attacks.

ftware without knowing, which then exploits the vulnerability without the need for further user interaction. The p

t and prevent legitimate use of the device, thereby causing a denial of service. This kind of attack could be particularly execution privileges to trusted applications only and avoiding installing software from unknown sources can help i nerability.

of service by keeping the device in a constant state of rebooting, effectively preventing the user from using their additional execution privileges.

ges on the device.

with elevated privileges, without requiring any user interaction.

GH base score.

promised Android device.

uetooth server that requires system execution privileges for exploitation.

without needing user interaction, an attacker with system privileges could silently snoop on potentially sensitive

his vulnerability does not require user interaction and needs System execution privileges.

high level of access to the system, such as by compromising a system-level process or application. The information Android-13.

essing sensitive information or causing the system to behave unpredictably.  
1 steps.

ction is necessary for the exploitation.

e, potentially leading to further exploitation or data leakage.  
ature of the fix and how the bounds check was added to prevent such out-of-bounds reads in the future.

ss. The exploit could be performed without any direct interaction from the user.

hout the need for user interaction.

ice over Bluetooth, potentially leading to further exploitation or data compromise.  
security professionals to patch the affected systems and prevent exploitation.  
nerability, an attacker would need System execution privileges, and no user interaction is required.

bounds of allocated memory, leading to local information disclosure. Since user interaction isn't needed, the expl  
roperly checked when accessing array elements to prevent such an issue. For the actual code affected in Android  
es.

; this data could corrupt memory and allow an attacker to execute arbitrary code or cause a system crash, leading

compromise the security and privacy of the device user.

For successful exploitation, an attacker would need System execution privileges, and user interaction is not required for exploit

The amount of data an attacker could exploit this vulnerability may vary, but since no user interaction is needed, it could potentially be exploited

to exploit this vulnerability.

As a result, a malicious app or process running with system permissions could leverage this flaw without the user's knowledge.

Android includes a security patch for the issue.

System execution privileges. User interaction is not necessary for this exploit to occur. No user interaction is required at the time of exploitation.

The issue was identified by Android ID: A-256165737.

The vulnerability allows an attacker to perform memory out-of-bounds disassembly, potentially accessing sensitive information that should otherwise be inaccessible. System execution privileges are required to exploit this vulnerability, and no user interaction is necessary for exploitation.

CVSS: 7.5. To address this vulnerability, users should update their devices to the latest version of the operating system.

isue of sensitive memory information that should not be accessible, affecting the confidentiality of the data on the device. This could lead to local information disclosure under certain conditions where the attacker has compromised the system.

memory due to the flaw in the `btm_ble_read_remote_features_complete` function, potentially accessing sensitive information in a timely manner.

on the device. Since this vulnerability is related to user interaction, it could be presented as a seemingly benign issue.

potentially leading to the leaking of sensitive information.

mit this vulnerability.

r user interaction. This could undermine the confidentiality of data stored on the device. It's not explicitly stated, I exploitation.

eges. Once exploited, the vulnerability could allow the attacker to gain access to sensitive information, possibly le  
d. Additionally, user interaction is not required to exploit this vulnerability.

ity of the device, leading to a local elevation of privileges.

and out of bounds read.  
ccess to, therefore disclosing sensitive information or potentially gaining further privileges.

if a need for user interaction means the vulnerability could be exploited without the user's knowledge.

user interaction, it could be triggered by malware or a malicious app that has already obtained System execution permissions. This vulnerability could be exploited by a user interaction to be exploited.

Additional malware, or taking control of the device. This vulnerability could be exploited by a user interaction to be exploited. An example scenario could be a malicious app i

permissions or potentially escalating privileges by executing commands the app itself does not have the permission to

information, alter system settings, or perform actions that are usually restricted to system or admin-level privileges.

This is a method to discretely gather information or trick the user into divulging more sensitive information by impersonating a system service. This vulnerability affects Android versions Android-12L and Android-13.

ion, the attacker could achieve a local escalation of privileges on the device.

lly restricted to users with higher privileges. The specific impact could vary depending on the privileges gained and the threat.

action or additional execution privileges.

'eputy' may unknowingly help bypass Factory Reset Protections, leading to privilege escalation.  
e operating system up to date.

ex, int value) { if (index is not properly checked) { array[index] = value; }}

In the context of CVE-2023-20956

l escalation of privilege without the need for additional execution privileges. Moreover, user interaction is not required.

resulting in unauthorized removal of apps and potential undermining of device security and user data.  
al privileges. It has been marked as 'CRITICAL' with a base score of 9.8.



device without the user's knowledge, leading to a full compromise of the system. Attackers could use this to gain unauthorized access to sensitive data, causing substantial damages for users and organizations affected by the vulnerability. The attack does not require the need for additional execution privileges. It affects Android version 13.

The attack bypasses protection by factory reset protection. The attack would likely require physical access to the device or some form of social engineering. No user interaction is not required for the exploitation of this flaw.

The attack could lead to sensitive information disclosure. The attack could be performed locally without any user interaction, possibly compromising sensitive data.

The attack requires additional execution privileges. It affects Android versions 11, 12, 12L, and 13.

The attack could lead to unauthorized access to sensitive information without user knowledge or consent. This could lead to unauthorized access to sensitive data.

The attack requires additional execution privileges.

) repository for the patches to see how the vulnerability was addressed.

and the user's data, such as accessing sensitive information, modifying system settings, or installing additional mali

ipromised device.

uthorized distribution of sensitive information and a subsequent local escalation of privilege.

2L, and 13.

urity patches to mitigate the vulnerability.

ta or interfering with system operations.

ecution privileges or user interaction.

escalation of privileges.

oitation.

entially read memory it's not supposed to, thus leading to a potential information disclosure of sensitive user or s  
tation of this vulnerability does not require user interaction.

nature of the vulnerability, it could be exploited by malware already present on the device, or through chaining w

an be exploited without user interaction and does not require additional privileges, an attacker could gain unauth

vulnerability to execute arbitrary code on the device with the same permissions as the cellular firmware, potentia  
are code to identify the exact location of the vulnerability and apply the necessary patches or updates provided b  
rability affects Android version 13.

on or running on the device, potentially gaining elevated privileges without the user's knowledge. This could then

in is necessary for exploitation.

d privileges. Since no user interaction is needed after the file is opened, this makes the vulnerability particularly c  
d 13.

er's knowledge, and could persist across device reboots, affecting the usability of the device.  
way that triggers an error not caught by the system.  
ry additional execution privileges. However, user interaction is required to exploit this flaw. The affected Android

y grants the malicious application access to private information, resulting in a local information disclosure.  
ccess is tricked into performing unauthorized actions on behalf of a malicious entity. While a concrete code exampl

malicious sites. The impact and attack scenarios depend on the level of access the attacker has and the context in  
y leading to actions such as session hijacking if the script accesses and steals cookies or other authentication infor

;-palm/tree/main/CVE-2022-45634'.  
n that are insecure, ultimately leading to the disclosure of information that should be restricted.  
equest parameters to retrieve information that an attacker shouldn't have access to. The exact details would depe

nstance, if the API endpoint `/unlock\_model` was improperly secured, an attacker could send a POST request to th

without authorization and gain unauthorized access to the account.  
These usages could also reduce the risk of exploitation.

information. For instance, an attacker could use common password dictionaries or perform automated password guessing attempts, account lockouts after multiple failed attempts, and encouraging the use of multi-factor authentication (MFA).

Access to the secret mode in the Samsung Internet application under certain conditions.

Ability to bypass the security checks that prevent unauthorized access to this information under certain conditions, potentially such a vulnerability.  
Access permissions.

Access-based access controls on networks.

Exfiltrated data on the device, or through other methods that leverage the disclosed vulnerability.

Access through a malicious application that the user has installed, which then exploits the vulnerability to leak or exfiltrate data. Additionally, following best security practices and regularly monitoring for updates provided by Microsoft can help reduce the risk of exploitation.

lead the user into taking an action they believe is genuine, such as entering personal information, that are then sent as alphanumeric tokens during token setup.  
is. Though the exploitation only results in reading sensitive information, it does not allow for the modification or c

one-time password or is in the process of setting up a new token, the malicious app could capture the OTP or the secrets to the users' accounts protected by the SAP Authenticator.

that could corrupt memory.

ing of the system. For example, if there's a function that expects an array index and an attacker finds a way to pa

browser. This could deceive the victim into believing they are visiting a legitimate website when they are, in fact, i

of social engineering.

TPS) and the real web address of the page you're visiting.

ibox. The attacker could then trick the user into disclosing sensitive information under the belief they are interacting with a legitimate service.  
ge.

ate WebApp, the attacker's crafted page spoofs the installer, potentially leading to misinformation or further exploitation.

ure of sensitive information without the user's knowledge. For instance, it could result in phishing attempts, stealing credentials, or other malicious actions.

would otherwise be restricted by the same-origin policy. This could result in the unauthorized disclosure of personal information if the autofill feature is used in a way that lets the attacker capture the autofilled information.

This could lead to the exposure or corruption of sensitive data if successfully exploited.

Personal data, thereby disclosing sensitive information to the attacker. The exploit could be delivered via a website, email, or other means. Additionally, developers maintaining affected systems should review the related security advisories and apply necessary patches to prevent execution privileges.

on mobile devices, or potentially access sensitive data conveyed over Bluetooth connections. Given that no user interaction

is required, this could compromise the device's security and the user's data.

ation issue to escalate privileges and perform unauthorized actions, such as accessing protected information or al  
to the permissions requested by apps to reduce the risk of inadvertently installing a malicious application that co

1. The vulnerability requires User execution privileges to be exploited.

te or alter system files. This could allow the attacker to disable software, breach data integrity, or escalate privile  
ist such exploits.

'.

privileges within the system, which may then be used to take control over the device.  
trusted sources and keep their devices secure with a strong lock screen to prevent physical attacks.  
ditional execution privileges. It affects Android versions Android-12, Android-12L, and Android-13.

ithout authorization.

ensitive data, bypass security measures, or carry out further attacks on the device.

now freed./\* ... \*/\*(int\*)buffer = 42; // Use after free: undefined behavior, might corrupt memory or lead to code



tails of the vulnerable functions within the kernel.

hout the owner's knowledge. The exact method of exploitation would depend on the context and the way the kei

rther exploit this elevated access to tamper with other system settings or exfiltrate sensitive data.

such a vulnerability could have been exploited.

require additional execution privileges, a low-privileged user or process could potentially gain higher privileges or

ing to unauthorized information disclosure.

tion is not necessary.

nt. This recorded audio could then be transmitted to an attacker, leading to a privacy breach.

ails that the user believed were removed from the device.

without the user's knowledge, leading to degradation of device performance or making the device unresponsive or with specially crafted parameters to cause resource exhaustion. For security reasons, and to prevent misuse,

thout their consent.

acks or privacy violations.

r to track the user's activity within the app, potentially gathering sensitive information about the user's music pre

3.

over, data theft, and unauthorized actions using the compromised accounts.

Iterable code pattern  
`SQLiteDatabase db = this.getWritableDatabase();`  
`ContentValues values = new ContentValues();`  
or integrating Android's Keystore system for encryption are both effective mitigation strategies.

ta, or perform actions with the privileges of the application. Such SQL Injection vulnerabilities typically arise due to password as ' OR '1'='1'. If the application directly concatenates these input fields into an SQL query without proper endpoint and gain unauthorized access to retrieve application data.

ies on the lack of robust authentication checks on certain server-side features.

ersonal information, application settings, or other sensitive data managed by the app.  
ed access to their data.

visiting the malicious page or embed the exploit in advertisements served on legitimate websites.

swords or authentication tokens, which might be stored or handled insecurely within the application's context.

s required, the attacker could discreetly run the exploit to read sensitive data out of bounds, potentially gaining u

-level permissions that enable the attacker to execute actions or access data that should be restricted, leading to i  
p.

information disclosure. Moreover, they might write arbitrary data to files within the app's storage directory, but th  
001/', and the detailed HackerOne report at '<https://hackerone.com/reports/377107>'.

l.

which the app might accept as valid. The attacker could then intercept and decrypt communications transmitted l

o user accounts or perform actions on behalf of the user without their knowledge.

G: Hypothetical vulnerable code snippetString username = 'user1';String password = 'pass1234';// Logging creden  
e to ensure their data is secure.

I information. Furthermore, adhering to a secure software development lifecycle that includes thorough testing a

disrupt the device's operations, leading to further exploitation or causing the device to crash.

n that is usually restricted. The attack requires physical access to the device or execution of a malicious application es by properly validating and sanitizing all inputs in their applications to ensure that untrusted input does not cor

l privileges. This could allow the attacker to perform unauthorized actions or access sensitive information that is n

he device, potentially leading to information disclosure or control of the device.

al access to the vulnerable device to exploit this vulnerability.

unused code paths.

sical access to the device or to deceive the user into running a malicious application that exploits the vulnerability l be specific to Routine's internal workings and not easily representable without proprietary source code access or

osing sensitive information, as they might believe the deceptive UI to be a legitimate part of the website or Chron

the iOS or Android client or 30 more messages can be written into the affected conversation.

that history not being displayed. Although other conversations would not be affected, this could disrupt communic

rol over the binder communication channel and influencing kernel memory management to escalate their privileg  
d to exploit this flaw.

elevate the attacker's privileges on the device to gain unauthorized access to system resources or sensitive inform  
g further execution privileges. It requires no user interaction to be exploited.

eding user interaction or additional execution privileges.  
l for this CVE at <https://source.android.com/security/bulletin/pixel/2023-01-01>.

l privileges. The attacker could potentially access sensitive information, install malicious applications, or make cha

ereby leading to local information disclosure.

ve and resulting in a local denial of service. Since no user interaction or additional privileges are needed, this attac

calation of privilege.

es on the affected device.

· could thus maintain a persistent presence on the device and possibly gain elevated privileges without alerting th

opsService.java. This could enable the attacker to perform actions on the device that would normally require high  
n privileges.

possibly gaining unauthorized access to sensitive information or performing actions on behalf of the user without

use of the device's telephony capabilities such as making calls, sending SMS messages, or accessing sensitive information. Additional execution privileges.

without the owner's knowledge. This could happen without the user's direct interaction, increasing the ease with which an attacker could exploit such vulnerabilities. Security updates to protect against such vulnerabilities.

crashes, rendering the device unusable and achieving a denial of service until the issue is resolved.

loss of data within protected memory areas. This might result in gaining unauthorized privileges or access to sensitive information. Mitigations and maintain a minimum set of privileges for each app to limit the potential impact of such vulnerabilities.



privileges.

attacker could craft a malicious application or use a pre-existing app to launch another activity that shouldn't normally require elevated privileges. Developers should ensure that all intents are properly validated and that activities are only accessible to authorized users.

e. If no user interaction is required, an attacker could take advantage of this without alerting the user.

exploit the vulnerability.

to leaks or further exploitation. Since the attack relies on social engineering, it may come in the form of a seemingly legitimate request.

If no user interaction is required, this attack could be executed silently and potentially go unnoticed by the user.

no user interaction required for exploitation.

should normally require elevated privileges, all without the user's knowledge or interaction. To prevent exploitation.

allowing an unauthorized user to gain elevated access.

the device without the user's knowledge.

lead to local escalation of privilege in regards to Bluetooth Low Energy (BLE) operations. This vulnerability does not

to Bluetooth functionality, intercept or tamper with Bluetooth communications, or potentially gain further access or user intervention.

return value of this method, potentially exposing user's account names and other sensitive details.

by theft or phishing attempts.

stener", "Notification posted: " + sbn.getKey());}""In this hypothetical code, sbn.getKey() should not be logged since

or gain control over certain device functions without any required user interaction.

es, escalation of privileges if the corrupted data can influence behavior of the system in a beneficial way for the a

t requiring additional execution privileges.

ds to the MasterClearConfirmFragment's critical functionality.

sions 10, 11, and 12.

the settings switch, unknowingly providing the malicious app with permission to change critical system settings. The user's knowledge, thereby gaining permission to alter system settings for malicious purposes. User interaction is necessary for the exploitation to occur. The affected products include Android versions 10, 11, and 12.

effectively causing a denial of service. Since the attack requires user interaction, it might involve convincing the user to interact with the same device.

VARP client. During this hijacking, the attacker could then capture sensitive user information, such as login credentials. Activities have the correct task affinity and task re-parenting settings, and confirm that exported components are in the foreground. The user is prompted to perform actions, thinking they are securely interacting with the website.

<https://security.gentoo.org/glsa/202305-10>- <https://security.gentoo.org/glsa/202311-11> for more information on phishing attacks, or tricking the user into granting permissions or revealing sensitive information.

<https://security.gentoo.org/glsa/202305-10> and <https://security.gentoo.org/glsa/202311-11> for more information on the main origin. This could allow the attacker to gain unauthorized access to certain device functions or data that is sensitive. For more information and patches provided by Google and the references linked in the CVE report to understand the nature of the attack, see the following links:

with what they perceive to be a familiar website.

authorization.

ive files and also write access, potentially leading to data corruption or the injection of malicious code. Sharing of such data involve reporting to the concerned parties and not sharing the exploit code in public.

[advisories/security/advisories/GHSA-wvr4-gc4c-6vmx](#).

er's confidential Nextcloud files and private conversations without authorization.

l-of-service condition, depending on the context of the affected system within the vehicle's infrastructure. e.g., overly long strings, unexpected data types), an attacker could overwrite critical memory structures and control flow. Validation improvements should be undertaken to enhance the overall security posture and reduce the likelihood of

tion of memory, allowing the attacker to execute arbitrary code or cause the system to crash, leading to a denial-

tially redirect the user to a nefarious site or directly initiate the takeover of the user's RYDE account. authority. Regular security audits and following good security practices in handling external inputs and deep links

rowser, when in fact they are giving it to the attacker's website. higher scores, such as 'HIGH' or 'CRITICAL'.

to Opera Mini, they might unintentionally grant the malicious website access to their location data. The attacker could perform other malicious activities.

ate website, when in reality they are directed to a malicious site controlled by the attacker.

Secure SAFE Browser 19.1 enters the intended web address, the browser might not differentiate the deceptive domain actions.

cookies could be sent to this malicious site, possibly resulting in session hijacking or other security breaches.

execute arbitrary code on the affected device. This vulnerability relies on the attacker having the ability to manipulate improper handling of unexpected driver names leading to a buffer overflow during the startup sequence of Firefox status indicator that informs users about active audio recording.

initial eavesdropping and privacy violations.

automatic tab upon restart, leading to a persistent denial of service condition.

measure of the HSTS policy.

This could lead to the exposure of sensitive information due to a man-in-the-middle (MitM) attack, phishing, or other attempts to intercept or tamper with their secure communications. It's important to note that this issue only affects

elopment team in the browser's codebase itself.

rowser does not properly save this information, the user could close and reopen the browser and subsequently a

lso be done by looking at the changes made in the patches to get an idea of the nature of the fix.

s of the user or cause a denial-of-service condition through application crash. It's also been reported that this vuln

ve information or credentials. For example, an attacker could create a phishing page that shows a legitimate bank

the user's account, similar to how a cross-site request forgery attack operates.

ccount.

the user's account settings or trigger other actions without the user's consent, depending on the specific phone o

that the user did not intend or may lead to phishing, malware distribution, or exploitation of further vulnerabilities  
ker-controlled JavaScript code in a privileged context.

a user with a vulnerable version of Firefox, Firefox for Android, Firefox ESR, or Thunderbird visits the crafted page, Firefox for Android version 91.9.1 or later.

illeges of the parent process.

ique to create a phishing site that looks like a known bank's website, tricking users into entering sensitive information. If a victim sharing the same source IP and could allow the attacker to view session states and disconnect VPN sessions.

9bd2c15cdfd39ac173665fad3f2598b54 - Windows: <https://github.com/mozilla-services/guardian-vpn-windows/> victim's session states and disconnecting their VPN sessions.  
iks, potentially enabling phishing attacks.  
legitimate site when in fact it could redirect a user to a malicious website for phishing.

ior to version 3.9.4.

vw.bank.com', but because of the Unicode manipulation, the actual link could be 'https://malicious-site.com'. An  
mation/, [https://www.docomo.ne.jp/service/plus\\_message/](https://www.docomo.ne.jp/service/plus_message/), and [https://www.softbank.jp/mobile/service/plus-](https://www.softbank.jp/mobile/service/plus-message/)  
ecution privileges. Interestingly, this vulnerability can be exploited without any user interaction.

escalate its privileges on the device without alerting the user, potentially gaining access to restricted data or perform

itions or a compromised system process could attempt to exploit this vulnerability silently. r with system execution privileges to compromise the device.

Since user interaction is not required, the application could use this weakness to perform unauthorized actions o for exploitation. The affected product is Android, specifically version Android-13.

tricted tables and subsequent information disclosure. The attacker would first need to gain User execution privile,

user's interaction or any additional execution privileges. For instance, a malicious application could use this flaw t ecurate their systems without exposing the actual exploit method. ociated with the Android operating system, requiring no user interaction for exploitation.

ble, leading to information disclosure. The specific code that triggers the out of bounds read is not provided in the follow instructions from the device manufacturer to ensure their system is updated with all available security pat i privileges. Furthermore, this vulnerability could be exploited without any user interaction.

With the mitigation bypassed, the malicious application could elevate its privileges and gain access to system-level



lity.

ails and patches.  
ditional malware, or compromising the integrity of the system.

n.

ation of this vulnerability.

teraction. Such an application would need to already have System execution privileges to exploit this flaw effectively.

ervice\_1\_6.cpp, potentially overwriting memory and executing arbitrary code on the device without any user interaction. If successful, it's a security flaw that can be exploited by an attacker to gain access to potentially sensitive information from the device.

This vulnerability is that it does not require user interaction to be exploited, which makes it more dangerous as users are often unaware of the issue. This could lead to data theft or manipulation, or it could serve as a stepping stone for further attacks. The exact details are not clear.

is data theft or manipulation.

Initially disclosing user data or system information without the user's knowledge. This could lead to user interaction for exploitation.

could lead to a local privilege escalation, allowing the attacker to execute code with elevated privileges. The exact details are not clear.

ing sensitive data, or gaining control over the device's systems. This could lead to a high level of damage.

allow the attacker to overwrite crucial data structures or execute arbitrary code with elevated System privileges, which could lead to a high level of damage.

privileges, leading to full control of the affected device.

is not require user interaction for exploitation.

ensuring that an array index or pointer dereference is within the expected range before access.

attack could be carried out without the user's knowledge, potentially giving the attacker access to confidential data

elements that are accessed, compromising the privacy and security of the user's data.

privileges. No user interaction is necessary for the exploitation of this vulnerability.

to report when the vulnerability was present, or rely on Android's official security bulletins or patch notes for more details

potentially gathering data for further attacks or compromising user privacy.

and the intended limits. Such an attack could lead to information disclosure without the need for user interaction

Typically, responsible disclosure practices would prevent the sharing of exploit code until most affected systems are patched to prevent a system crash. Exploitation could be performed by malware or a malicious app that would execute without the user's knowledge on the system where they have execution privileges.

Attackers could further compromise the system or elevate their privileges without the need for user interaction, though the specific details of the exploit are not disclosed.

Allowing the attacker to execute arbitrary code with elevated privileges. Since no user interaction is required, the exploit could be used to compromise the system.

Attackers could gain privilege without the device owner's knowledge, potentially gaining unauthorized access to sensitive areas of the system through this exploitation.

Other potential exploits, such as gaining persistent access, bypassing security mechanisms, or compromising other processes on the system, are also possible.

execution privileges needed. This issue affects Android kernel versions but does not require user interaction to be exploited.

malicious code. Since no user interaction is required, if a user installs a compromised application with the necessary privileges, specifically the Android kernel.

attacker to execute code with escalated privileges. For instance, the attacker could inject malicious code or overwrite sensitive data. The vulnerability to be exploited.

responsible for encoding protocol buffer entries within SIM data management functions.

exploited by the malicious application without the user's awareness.

to compromise the device's security.  
needed.

access over the system, modify system data, or disrupt system services without any user interaction.  
Additional Android Security Bulletins or vendor advisories.

exploited and does not require user interaction for exploitation.

it any user interaction, leading to a local escalation of privilege. An attack scenario could involve a malicious app c  
sary for exploitation.

happen without the user's knowledge, potentially leading to unauthorized access and control over the device.

ivileges, potentially leading to a local escalation of privilege on the device. Since user interaction is not required to

sensitive information or control over the device.

uccessful exploitation would not require additional execution privileges or user interaction.  
accessing sensitive information such as passwords, encryption keys, or personal user data. Since this type of explc

the device without the user's knowledge. The attacker could then perform malicious activities such as data theft, execution privileges required to exploit it. No user interaction is needed for the exploitation of this vulnerability.

need to have a certain level of access to the system or may have to combine this exploit with another vulnerability per access to the inner workings of the operating system. privileges. No user interaction is required for exploitation.

st security updates related to this CVE.  
further attacks or exploitation.

er attacks or exploitation.

em-level access to the device.

pass security restrictions and compromise the integrity, confidentiality, or availability of the system.  
developers and system administrators understand the risk and apply necessary patches or mitigations.

ant to only install applications from trusted sources, as these measures can help mitigate the risk of exploitation.

1 the Android kernel.

ay or buffer without proper bounds checking could be susceptible to this kind of vulnerability.  
ivileges or the execution of unauthorized actions without the user's knowledge.



occur without any complex technical skills or tools, simply by exploiting the functionality of Safe Mode on Android application's API.  
tain sensitive information regarding the controlled devices and their activities.

requests to the Boomerang app's API, potentially altering parental controls, accessing sensitive data, or otherwise

access sensitive information that should typically be protected by secure communication channels.  
ty to ensure connections default to secure protocols.  
he security of communications within the application.

se LINE APIs maliciously, possibly stealing sensitive information or performing unauthorized actions on behalf of t

t be used, leading to a potential downgrade or upgrade in the actual signature used.

lly allowing the attacker to take full control of the device.

ito performing certain actions to facilitate the attack.

data on the device.  
against such vulnerabilities.  
e device, to craft and send arbitrary commands to the HMI device.

contained in these messages, which could range from altering system configurations to installing malware or exfiltrating data for other goals.  
device.

In an industrial setting, this might result in the malfunctioning of machinery or halting production lines, with potential safety risks. Proper security measures can also help protect against such attacks.

of the device, data theft, or unauthorized control over the device.  
possibilities of unauthorized actions.  
tightly restricted to trigger the unintended behavior.

potentially intercept the username and password needed to authenticate with the MQTT server.

control operations, or even compromise the integrity and availability of the system managed by the affected MQTT broker. Users should refer to the vendor's security advisory or contact the vendor for specific patching and mitigation guidance.

connected to the MQTT broker.  
request and process the server's response // ...Credentials and server IP are retrieved here using HTTP... } // Other

Android Client application linked to the agent. An example code that might cause the vulnerability could be a content broadcast intent encrypted with that key, leading to unauthorized alterations in the server connection configuration.

configuration update and change its server settings to point to an attacker-controlled server. Subsequently, the application sends the following code example:

```
maliciousIntent.putExtra("R_SETTINGS");maliciousIntent.putExtra("ENCRYPTED_DATA", rsaEncryptedServerConfigData);sendBroadcast(maliciousIntent);
```

The code example is not provided here to prevent misuse.

The interpinning system of the affected device.

To secure, following recommended guidelines from Android on managing WebView safely.

ion.

ause the phone to dial the specified number without the user's knowledge or consent. Here's a hypothetical code

lists.fedoraproject.org/message/6T655QF7CQ3DYAMPFV7IECQYGDEUIVVT/ respectively.  
granting permissions without realizing the security implications.  
's.

hives/list/package-announce@lists.fedoraproject.org/message/KUQ7CTX3W372X3UY56VVNAHCH6H2F4X3/- http

:e.

e could be executed within the context of the WebView, leading to various types of attacks such as data exfiltration directly.

compromised device as a foothold for further attacks within a network. It can be exploited.

Vulnerabilities in communication or system software to trigger the camera feed. However, the practical application is as follows: `Intent intent = new Intent(context, CameraActivity.class); startActivity(intent);` // This would start the camera without checking

for security breaches and unauthorized data access.

`startActivity(new Intent("com.example.WifiApAutoHotspotEnablingActivity"))` to change settings without proper u

user's consent. This can be used to gain unauthorized information or permissions to the attacker's app.

Users should be providing sensitive information or agreeing to certain permissions that are actually being collected or granted.

vulnerability Notes (<https://jvn.jp/en/jp/JVN03447226/>).

user to an attacker-controlled website. This could potentially expose the user to phishing attacks, malware download, or redirect the user to a rogue website. The attack could be carried out via email, text message, or another app that invokes the user's device.

could be used to exploit additional vulnerabilities in the user's device.

could result in phishing attacks or the unintended disclosure of sensitive information.

saved in their browser for convenience.

[fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/OCFEK63FUHFXZH5MSG6TNQOXN](https://fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/OCFEK63FUHFXZH5MSG6TNQOXN)

[archives/list/package-announce@lists.fedoraproject.org/message/2DMXHPRUGBUDNHZCZCIVMWAUIEXEGMGT/](https://fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/2DMXHPRUGBUDNHZCZCIVMWAUIEXEGMGT/)  
stealing of credentials, or other malicious activities.

n's security team.

railing list (<https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/OCF>  
rowser crash. Successful exploitation of this vulnerability could result in unauthorized actions being performed on

it are not distributable via code.

's data. For example, an attacker could potentially overwrite system files or place executable files in areas where

tors.

olicies.

[g/message/PQKT7EGDD2P3L7S3NXEDDRCPK4NNZNWJ/](https://lists.fedoraproject.org/archives/list/package-g/message/PQKT7EGDD2P3L7S3NXEDDRCPK4NNZNWJ/) and <https://lists.fedoraproject.org/archives/list/package>

device.



/ the attacker.

e lock.

often include detailed explanations of the vulnerability, how it was discovered, and potential mitigation strategies.

that the logs on their devices that contain BLE messages are secured and cleared regularly to reduce the chance of exploitation. Mobile Security Suite. Successful exploitation could lead to unauthorized access to critical data or complete access to the device.

s and patches.

articulate or compromise any user accounts.

leading to a compromise of system integrity.

with the site.

%BA%E3%83%94%E3%83%83%E3%82%AF%E3%82%B9-%E3%83%93%E3%82%B8%E3%83%8D%E3%82%B9%E3%  
ccess to private user data or service operations which normally require authentication.

ind memory allocation routines.

ic areas where attackers might be present.

ions.

required, the attacker may trick the user into performing certain actions to facilitate the bypass.

ustrate such an error: ``c#include <stdio.h>#include <string.h>void vulnerable\_function(char \*input) { char buffer

guration details of the vulnerable device.

te}''In such a case, passing a `data\_size` larger than 128 to `load\_dt\_data` function could lead to writing beyond 1  
importance of securing privileged processes and ensuring that they do not contain such vulnerabilities.

nds or receives over the cellular network.

eds to obtain System execution privileges.

ack could go unnoticed by the device's user.

device's memory.

· input validation and bounds checking in the implementation of `initiateHs20IconQueryInternal` function in `sta_ifa`

e device.

grity of the device.

oitation. It affects Android version 13.

acker could use for further malicious activities such as gaining additional privileges or accessing private data.  
pply patches and not for the general public or potential attackers.

nnected to, passwords stored for these networks, or other data processed by the wifi server.

ed for any user interaction on the victim's device.

asier to exploit if the attacker finds a way to reach the vulnerable code path.

user interaction, it's feasible that sensitive data stored in adjacent memory could be exposed to the attacker.

The specific code execution path or data accessed would depend on the memory layout and the nature of the ma

able to.

both messages or requests to trigger the out of bounds read.

Increasing the likelihood of successful exploitation.

Approval or higher-level privileges.

establishing a Bluetooth pairing or connection.



```
it data_length = receive_data(buffer, sizeof(buffer)); if (data_length > SIZE) { // This condition might prevent a heap
```

malicious application could attempt to trigger the vulnerability without the device owner's knowledge.

exploited silently, leading to information disclosure or a stepping stone for further attacks against the system.

ensitive user data.

or by designing code to prevent concurrent access to shared resources.

ation over the mobile network.

d from accessing sensitive functionalities.

sensitive information being compromised.

nformation, which could be further used to escalate privileges or perform additional attacks.  
ing proper validation to ensure that array or buffer accesses are within the intended boundaries.

s to disrupt the normal operation of the device, potentially hindering the user's ability to interact with system fea

ie unauthorized code, giving the attacker increased control over the device. Such an attack could be performed wi

not require user interaction, the user might remain unaware of the attack if it occurs. Regularly updating the Andr

hold 10 would cause an overflow, potentially leading to undefined behavior and system vulnerabilities. The actual

instance, an attacker could execute code that interacts with the affected function and captures information like u

leading to an escalation of privileges without the user's knowledge.

closure of sensitive information stored on the device.

escalation of privilege.

integrity of the affected system.

/ to execute arbitrary code on the victim's device remotely, potentially gaining access to sensitive information or f

loits the transparency or other visual tricks to deceive the user into interacting with the overlaid interface.

g responsible disclosure, it is not appropriate to share potentially malicious code examples. Users are advised to u  
rema or by invoking an explicit intent.

device that sends an explicit intent to the vulnerable activity.

est practice, no actual exploit code should be written or shared for ethical and security reasons.

which could lead to further exploitation or negative impacts on the app's functionality.



ables or secure vaults implemented to handle sensitive information securely.  
ignizes these embedded credentials.

ovided through the keyboard.

are update.

s site. This could lead to the user disclosing sensitive information or executing unintended actions on behalf of the

e on the affected device. It's important for users to update their browser to mitigate this threat.  
-after-free vulnerabilities arise generally involves scenarios where memory is accessed after it has been freed, po  
s.

this flaw.

or possibly through another vector if the vulnerability can be triggered remotely.  
used to potential malicious input.

node.

on, or further compromise of the device's security.

and perform unauthorized actions without user interaction.

e, potentially leading to data theft, unauthorized access, or other malicious activities.

e.

rability doesn't require user interaction, it could be exploited silently without the user's knowledge.

ected Android versions are Android 11, Android 12, Android 12L, and Android 13.

and access to sensitive information.

tion, or disrupt its normal operation.

that are normally protected, effectively escalating the privileges of the malicious app.

ns without the user's consent or knowledge.

trary activities. Users and developers should refer to the Android Security Bulletin for detailed patches and code c

1 elevated privileges could allow for further exploitation, such as accessing sensitive information, bypassing securi

imits or changes that address the CVE.

ions include Android 11, Android 12, Android 12L, and Android 13.

to a local escalation of privilege.

d Hands-Free Profile (HFP) support are enabled, an attacker could exploit this to take control of the affected system

the device without the owner's knowledge. Given the complexity of Bluetooth exploits, code examples are typically

stored on the device.

uld otherwise be secured behind the lockscreen.

en to this output stream, it can be // accessed by other users/apps on the same machine. outputStream.write('sen  
l processes.

on.

tt.getInstance('TLS'); TrustManager[] trustAllCerts = new TrustManager[] {new X509TrustManager() { public X

id ethical considerations, the attack would likely involve unauthorized changes to configuration values stored with

e for the user.

damage is remedied.

s leading to denial of service. Here's a hypothetical insecure pseudo-code that might lead to such a vulnerability:

thout proper validation or protection. An unauthorized app might access these preferences and edit them, potent

g them to break Android's security model or exploit another vulnerability to gain unauthorized access.

.

out the user's consent.

```
ir = prefs.edit();// Malicious change performededitor.putString("user_privilege", "admin");editor.commit();In this l
```

```
der); unregister_provider(provider); // this may lead to double free if provider is already freed within register_pro  
ction is used within the Android system, the permissions required to reach the vulnerability, and other security m
```

rate exploitation requires significant effort and knowledge of the particular memory vulnerabilities, this type of a  
s or patches without empowering attackers.



data theft, or the introduction of additional malicious software.

andbox is designed to run untrusted code securely.

the application.

ary code or an attacker gaining the same privileges as the user running the affected application.

of the affected system. Users should update their browsers as soon as possible to reduce the risk of exploitation.

's computer. The possible impact ranges from data theft to system compromise.

ites a software download or opens another application without the user's explicit permission.

rmless file while actually downloading and executing malicious software.  
ie file name, while the actual file could be an executable (EXE) containing malware. The user might thus be tricked

xamples are typically not published for such vulnerabilities to prevent misuse.

crash, or other forms of memory corruption which can then be leveraged to compromise the user's system. Attac

cks.

023-15/

ssue, this is a hypothetical example. The actual exploitative code would be dependent on the website and how it i  
ormation like passwords or credit card details, thinking they are operating in a legitimate context.

t.

lication to perform various malicious actions, such as stealing sensitive data or compromising the device's security

believe they are not in fullscreen mode. This could be used in phishing attacks where the attacker might spoof a t

e page for The Thaiger app (<https://play.google.com/store/apps/details?id=com.TheThaiger.android>).

hen executed by the app, compromising the user's device and data.

and executed by The Thaiger app, leads to code execution. An example (hypothetical code snippet) is shown belc  
of service attack.

ilable memory, causing an OOM error and making the app crash repeatedly, effectively rendering it unusable and  
op version on APKPure (<https://apkpure.com/cn/bt21-x-bts-wallpaper-hd-4k/com.bungaakp007.bt21wallpaperofi>

iding to further exploitation such as data theft, additional malware installation, or taking complete control of the i

```
ION_INSERT_DATA');attackIntent.putExtra('data', largeMaliciousData);sendBroadcast(attackIntent);This code wou
```

ng to an escalation of privilege attack.

- privileges or manipulate the app's behavior to compromise the device or user's data.
- improper use of 'SQLiteDatabase' without appropriate access controls.

il of service.

t to open the application fails, causing a persistent denial of service until the corrupted data is removed or the app

· challenging for an attacker to identify and exploit such vulnerabilities.

`dit(); editor.putString(key, value); editor.commit();}`""An attacker could potentially use this method from outside the

weather

that the BestWeather app is missing proper security checks or encryption for its database operations.

executes SQL queries or employs any other method to tamper with the database, such as using the Android Content

possible, users may be unable to retrieve current weather conditions using the app.

ignored by the application.

external input, effectively rendering the Alarm Clock app unusable or causing it to behave unpredictably, which could

device's security.

vice, access sensitive information, or manipulate the behavior of the device.  
ted to this CVE.

protected features or data on the device.

l result in privilege escalation or other unintended behavior. For example: ``Intent maliciousIntent = new Intent(),

d crash, resulting in a denial of service.

ite users.

vulnerabilities.

calation of privilege on the device.

tilized, including but not limited to ad display issues.

curity-sensitive functionalities. Moreover, affected data could compromise user settings, manipulate application b

annot handle the large amount of data. This causes the app to crash immediately and consistently, denying the u

, an attacker could overload the function with an excessively large payload or malformed data, leading to applica

permission for Kids Place. This effectively removes all parental restrictions until the device is rebooted or the permi

at storage, potentially leading to unauthorized disclosure of sensitive information contained in the notes.

re pseudorandom number generator (CSPRNG), which could allow an attacker to exploit the lack of entropy and r

allow an attacker to predict DNS query IDs or other values that are meant to be random, leading to DNS spoofing

d()' is used instead of CSPRNG}```In this hypothetical snippet, 'rand()' generates a DNS query ID, which should be u

ther attacks such as gathering data for social engineering or preparing for more targeted exploits.

pecifics of this particular exploit are not provided to prevent misuse.

without the user's knowledge.



system resources or private information.  
er with special features.

access, stealing sensitive information, or spreading to other devices on the network.

ed versions to identify specific improvements or fixes related to this CVE.  
or other privileged system functionalities without any user interaction, leading to a local escalation of privilege.

re device owner.

ces, potentially leading to further compromise of device security.

egitimate user from registering a phone account successfully.

erability to disrupt the normal operation of the device.

ation.

ich could be leveraged for further attacks or identity theft.

ut validation, attempts to process incorrectly, causing a crash or resource exhaustion.

ial meeting details or other sensitive organizational data accessible via the Unite app.

```
void displaySensitiveInformation() { // Method that displays sensitive information } public void handleReq
```

itimate users. As it requires local access, the attacker would need to have physical access to the device or have a Smart Campus Android application and would require insider knowledge about resource-consuming functionality.

It should be confined to users with the appropriate permissions.

It should not perform actions on behalf of users or alter content displayed to them within the application. It should not steal cookies or session tokens.

ata despite sharing settings.

proper restrictions.

authorized operation.

ensitive information, modifying user settings, or redirecting the user to malicious sites.

sa-5398), the Fedora Project package announcements, and the Gentoo Linux Security Advisory page (<https://secu>

attempt to deceive the victim into thinking they are visiting a legitimate site, possibly for phishing purposes, by d

sharing confidential information.

sensitive information or system resources.

the context of the vulnerability.

ould display deceptive messages or control system feedback to manipulate the user's actions.  
rces.

n unauthorized access to sensitive information or even altering the behavior of other apps for gains such as privilege  
ccessible. Developers need to carefully check and sanitize file path inputs to prevent such vulnerabilities.  
; vulnerability.

rstem functionalities without the user's knowledge, leading to an escalation of privileges. This could happen silent  
hat could exploit vulnerabilities.

of service.

droid ID: A-237766679.

romise of the device's security.

or a pathway for further exploitation or unauthorized data access without needing any additional permissions. Check the Android Open Source Project (AOSP) repository to understand the specific nature of the fix.

from the Guest account without requiring any additional execution privileges. Exploiting this vulnerability does not require root access, thus potentially leading to further attacks or data compromise.

action or privileges. The attacker's code could access sensitive data, alter functionality, or even take over the device.



and activity restrictions.

for CVE-2023-20967 are not provided here to avoid facilitating misuse of the information.

or exploitation of other system vulnerabilities.

romise the integrity of the device's operating system.

privileges.

ition.

ileges or user interaction are required, the exploit could be more easily executed compared to vulnerabilities that require user interaction on the system.

on.

rated privileges, eventually compromising the system's security.

ages than initially granted to the application, ultimately leading to local escalation of privilege on the device.

ading to additional exploitation or compromise of the system. This can be done without the user's knowledge, as i

ple, there's no check for 'size \* sizeof(int)' overflowing, which when passed a large size value, could lead to a smal

alspace.com/pc/en/products.html

ion due to improper access controls or handling of data within the insert method. It's important to note that for th

ed by the payment application's security measures.

m.kyocera.kyoprint

cess, or cause other malicious activities without notifying the user.  
licit exploit code for vulnerabilities like CVE-2023-25954.

mples, but the scenario would involve interaction with the application's debug interface to extract information th

least privilege necessary and be rotated regularly to minimize the impact of potential exposure. It is also advisabl

ion against this vulnerability involves updating Google Chrome to the latest version.  
aling sensitive information like login credentials.

device owner's consent.

larger chain of exploits to take control over a system remotely if combined with other vulnerabilities.

nce the vulnerability does not require user interaction, the malicious activity could occur without the user's knowl

examples are not typically provided for vulnerabilities to prevent malicious use, a theoretical code example isn't a

```
'proper bounds checking should be here } memcpy(buffer, input, input_size); // Potential out of bounds write if in|
```



alated privileges to interfere with system operations, extract sensitive data, install further malicious software, or 1

ie device.

or additional permissions.

ques can help prevent such vulnerabilities.

ive data, or take control of the affected device.

king over the affected device.  
as correctly.

device.

ernel.

ier attacks or to compromise the user's data privacy.  
id kernel.

raphic keys, or personal information, without the device owner's knowledge or consent.

```
nd) {    // Processing code here  } else {    // Handle out of bounds condition or throw an error  }}``Developer
```

open without the user being aware, and the information obtained could be used for additional attacks or exploitation. Additional information is available on the Android Security Bulletin website.

```
ict data)); new_ptr->sensitive = 1337; // 'ptr' might be reused here, causing corruption return 0;}```Keep in min
```

on the device.

ct might be limited based on the privileges required for exploitation and other mitigating factors.

manipulate sensor-dependent features for a malicious objective.

tion of this vulnerability.

e, hence compromising the confidentiality of the system.

in access to sensitive information without the user's knowledge.

d access or control over the device.

evice vendors, and it is generally recommended to apply these official patches rather than manually altering syste

ularly version Android-13.

uld then be exfiltrated or used to perform further attacks on the system.

on the device or via a background process without the user being aware.

code execution.



escalation of privilege.  
his vulnerability.

eir privileges to fully compromise the system.

any user interaction.

increases the risk of an attacker silently causing disruption on an affected device.

Inerability.

sitive information disclosure, system disruption, or further security compromises.

viewing the security patch code or advisory from the manufacturer would be helpful.

they might be able to silently gather information or compromise the integrity of the system.

closure.

attacks. Since user interaction is not required, the vulnerability could be exploited silently without the user's knowledge.

exfiltrated to an attacker-controlled server.

Example of potentially vulnerable pseudo-code might look something like this: `char buffer[10]; int index = getUserInput();`

, or even take over the device's functionality, all without the user's awareness.

erwise be prohibited.

ed memory, which could be exploited by malicious code to manipulate the program's execution flow and elevate p

ity silently in the background.  
tation, making it possible for the vulnerability to be triggered programmatically.

ersistent nature of the reboot loop may require technical interventions to break the cycle and restore normal dev

ilarly harmful if the affected device is critical for business operations or used in emergency situations.  
reduce the risk of exploitation.

device normally.

information.

1 disclosure could then be used to further compromise the system or as a step in a larger attack chain.

oitation could be silent and difficult for the user to notice.

, one would need to refer to the Android Open Source Project (AOSP) repositories and reviews associated with the

; to a privilege escalation on the Android device.

oitation.

loited through a background process or a malicious app with elevated privileges.

dge.

This information could then be used for further attacks or data leakage.  
loitation.



re device.

n firmware and possesses System execution privileges. No user interaction is necessary for this vulnerability to be

e information stored in adjacent memory locations and disclosing it to the attacker.

choice or configuration option, capitalizing on social engineering tactics.

but such vulnerabilities might, under certain conditions, also be leveraged to execute arbitrary code or escalate pr

ading to further system compromise. Since out-of-bounds reads could lead to undefined behavior, the exact data

privileges on the device.

installed on the device that exploits the vulnerability in the background.

directly execute.

iating a legitimate foreground service.

it.FLAG\_UPDATE\_CURRENT);To mitigate this issue, it is recommended to secure the PendingIntent by flagging it a

d what malicious activities the attacker intends to carry out, such as data theft, deployment of malware, etc.

i, a similar lack of bounds checking could result in writing data outside the intended buffer, leading to information required for exploitation.

Unauthorized access to sensitive data, install additional malware, or perform other malicious activities.

Engineering to coerce the user into performing the necessary actions.

itive information without the need for elevated privileges.

tive data or system modification.

icious software.

ystem data.

ith other vulnerabilities that allow initial access at the system privilege level.

orized control over the affected device and perform malicious activities.

ally gaining control over the device.  
y the Android security bulletins.

allow the attacker to access sensitive information, manipulate system data or functions, or create a stepping stor

dangerous.

versions include Android 11, Android 12, Android 12L, and Android 13.

le isn't available, developers should review and follow best practices for input validation and access controls to p

which the vulnerability is exploited.  
rmation.

and on the application's codebase and are not provided publicly to avoid promoting misuse.

his endpoint with arbitrary data to unlock a model.



using attacks against known user accounts.

FA) wherever possible. Users should also be educated on the importance of using strong, unique passwords for tl

entially leading to information disclosure.

trate data without the user's knowledge.

p in protecting against potential exploitation.

to the attacker. Since specific code examples or attack techniques for this vulnerability might contain sensitive information, the disclosure of such details could lead to the deletion of data.

secret setup token, allowing the attacker to bypass two-factor authentication and potentially gain unauthorized access.

Accessing an index outside the valid range, this could trigger the vulnerability and its consequences.

Accessing a resource on a malicious site, potentially leading to phishing attacks or the disclosure of sensitive information.

Interacting with a trusted site.

Exploitation such as phishing attacks.

Stealing of session cookies, or redirecting the user to malicious sites.

onal data, such as addresses, phone numbers, or even saved passwords.

il attachment, or other means that persuade the victim's device to process the malicious video content, all without  
ssary code changes or workarounds recommended by Android's security team.

is required, the attack could happen surreptitiously, potentially leading to data leakage or unauthorized access to

tering system settings.

ould exploit this vulnerability.

ges further, gaining deeper control over the device.

execution. ""In the context of the binder kernel driver, this would involve accessing the binder buffer after it has l

ernel functions are implemented and used by the Android system.

n the system, allowing access to sensitive areas or perform unauthorized actions, effectively compromising the int

actual exploit code for vulnerabilities is generally not shared publicly.

ferences, search history, or any other data transmitted in clear text.

```
s());values.put("username", username);values.put("password", password); // Storing password in cleartextlong res
```

o insufficient input validation mechanisms within the application's code.

er validation or sanitization, the resulting query might look like: SELECT \* FROM users WHERE username='admin'

nauthorized access to private information stored in the device's memory.

a local escalation of privilege.

is write capability is limited to files with a .txt extension which somewhat limits the potential impact.

between the app and the server, gaining access to potentially sensitive information.

tials (unsafe practice)  
`Log.d('SUSHIRO', 'User login info: ' + username + ' / ' + password);` This code would be vulnerable  
and security reviews can help identify and fix potential vulnerabilities before release.



on the device to exploit the vulnerability.  
romise the app's security.

ormally protected. Alternatively, a malicious app installed on the device could exploit the vulnerability without n

/.  
r additional technical details.

ne's interface.

ation within that specific chat history and potentially be used as a denial-of-service attack.

ges.

nation.

anges to system settings without the owner's knowledge or consent.

ck could be executed without the user's knowledge. However, it would likely require physical access to the device

e user.

er privileges, thus achieving a local escalation of privilege. Due to the nature of the vulnerability, it is not feasible

: their consent.

rmation.

i which the attack can be carried out.

ive information.

nally be accessible, potentially leading to unauthorized access or control over certain functionalities of the device  
ers.

gly benign app requiring permissions that exploit the vulnerability.

t require additional execution privileges or user interaction to be exploited.

to the system.

ce it could contain sensitive information, and it is being called in a build type that is not userdebug or eng.

ttacker.

er to download and install a malicious application or to visit a website that can exploit this vulnerability.

entials or session tokens, which could be used for further exploitation or unauthorized access.  
intended to be exposed to other apps or have the proper security controls in place.

t require permission, thus compromising the user's privacy and security.

e of the fix and how to secure systems against such vulnerabilities.

rol the flow of execution, leading to arbitrary code execution or causing the system to crash.  
of exploitation.

-of-service condition on the affected device. Typically, such attacks require the attacker to have some level of acc

s can also prevent similar issues.

could then use this data for various nefarious purposes, such as tracking the user's movements or targeting them



ain from the legitimate one, leading the user to the phishing site where they could inadvertently divulge sensitive

ulate the graphics driver's name or inserting a malformed driver in the startup sequence.  
fox for Android.

other attack vectors aimed at compromising the integrity and confidentiality of the user's interaction with the web  
ected Firefox for Android, and other operating systems were not impacted.

ccess the site over an unencrypted HTTP connection. This could give an attacker the opportunity to perform a ma

erability has been exploited in the wild, suggesting that attacks might involve methods like spear-phishing or driv

ing site in the background while an alert prompts the user to re-enter their login details.

r carrier.

as if the URL triggers unexpected behavior in the browser.

or executes the script, the attacker's code could run with elevated privileges, potentially allowing unauthorized

ation.

ns.

commit/ac6f562973a83f6758cd7ab7aa313e863047d41b - Android: <https://github.com/mozilla-mobile/guardian->

unsuspecting user might tap on the displayed link thinking it's legitimate and be taken to a fake website where se  
message/.

ming unauthorized actions.

r gain access to sensitive data without the user's knowledge, thereby escalating its privileges on the device.

ges to run their malicious application on the device.

to gain system-level access or to bypass security mechanisms that would normally be enforced at its current privil

≡ CVE description, so a detailed code example cannot be given. The fix for the vulnerability would typically involve  
:ches.

resources, potentially leading to the compromise of the entire device. There is no specific code example available

rely.

raction.

n a victim's device without the need for user interaction.

ers might be unaware that their device has been compromised.

details of the exploit would depend on the device's configuration and the nature of the improper input validation

potentially gaining full control over the device without the user's knowledge.

ia or sensitive system parameters.

chnical details.

.

re patched.  
eed for user interaction.

ific impact depends on the nature of the data accessed and the system's configuration.

exploitation could occur without the device owner's knowledge.

Android system.

: system.



exploited.

leges, an attacker could leverage this vulnerability to achieve local escalation of privilege on the device without th

ensitive parts of memory, thereby gaining unauthorized access or control over system processes or data.

or process that exploits this vulnerability without the user's knowledge.

o exploit this vulnerability, any app or process running with system privileges could be used to launch the attack w

o it does not require elevated permissions, it could be executed by any app, raising concerns over the potential for

interception, or device compromises.

ity that provides system access. Once exploited, the attacker could potentially gain access to sensitive information



devices.

: manipulating the app's behavior.

he user. Since no direct code example is provided for the vulnerability, the attack relies on exploiting the lack of e

trating sensitive data.

al safety and financial repercussions.

TT broker.

er methods}""In this example, the code would be vulnerable because it is using the HTTP protocol when construct

tent provider declaration without proper permission checks, accessible by any third-party app: ContentValues val  
on.

ttacker could intercept, manipulate, or redirect the client's network traffic, potentially gaining access to sensitive  
iciousIntent);In this example, 'rsaEncryptedServerConfigData' represents the configuration data encrypted with th

· snippet showcasing the vulnerability: ``javaIntent intent = new Intent(android.intent.action.CALL); intent.setCon

os://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/WTRUIS3564P7ZLN



on or interface manipulation.

of this attack is limited to a small subset of devices running an outdated version of Android.  
for permission```In modern Android versions, you would handle camera permissions with a runtime permission re

ser interaction.

ed to the malicious application.

oads, or other malicious activities.  
ie custom URL scheme without proper checks.

1QWM4M5S/ and <https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/1QWM4M5S/>

·EK63FUHFXZH5MSG6TNQOXMQWM4M5S/, <https://lists.fedoraproject.org/archives/list/package-announce@list>  
the user's device, disclosure of sensitive information, or other security-related impacts.

they might be automatically executed.

:-announce@lists.fedoraproject.org/message/YKLJ3B3D5BCVWE3QNP4N7HHF26OHD567/.

es.

f these being used in an attack.

ss to all data available to the Oracle Mobile Security Suite.

81%AB%E5%BD%B9%E7%AB%8B%E3%81%A4%E7%B5%8C%E6%B8%88%E3%83%8B%E3%83%A5%E3%83%BC%f



```
[100]; // Incorrect buffer size calculation or lack of input length check can cause overflow strcpy(buffer, input);}ir
```

the buffer's memory space, causing memory corruption or other unexpected behavior.

ice.cpp.







licious app's design.



p buffer overflow by ensuring that // no more data than the buffer can hold is written to it. handle\_error();}""The









tures or requiring a device restart to restore functionality.

ithout any direct action from the device user.

oid operating system and ensuring apps are obtained from trusted sources can mitigate such risks.

al vulnerability in CVE-2023-21157 would be more complex and involve specific conditions within the encode func

user credentials or other system data.



further compromising the device's security.

update their systems and apply security patches to protect against such vulnerabilities.

e attacker.

tentially leading to undefined behavior or execution of harmful code.





changes.

ity measures, or installing additional malicious software.



m.

ally very specific to the exploit and often combine numerous low-level details that are beyond the scope of generat

```
isitive data'.getBytes()); }}
```

```
X509Certificate[] getAcceptedIssuers() { return null; }    public void checkClientTrusted(X509Certificate[] certs, Str
```

in the app's SharedPreferences, which can affect app behavior or state.

```
function handleDatabaseOperation(userInput) { dbFile = openDatabaseFile(userInput.fileName); if (userInput.ac
```

ially leading to denial of service if the affected app does not handle such changes correctly.

hypothetical scenario, the malicious app would access the SharedPreferences of the target app (Twilight v.13.3) ar

provider return 0;}```In the context of the vulnerability, calling `unregister\_provider` after `register\_provider` might l  
mechanisms in place that may mitigate exploitation.

ttack is more likely to be conducted by sophisticated threat actors.



l into thinking they are opening a PDF when in fact they are executing an EXE file.

:kers might use methods like phishing, compromised websites, or man-in-the-middle attacks to exploit this vulner

interacts with the browser's APIs.

y.

trusted entity to manipulate the user into providing sensitive information or executing unwanted actions.

```
import javaSharedPreferences sharedPrefs = context.getSharedPreferences(\
```

achieving a denial of service condition.  
fline130920/download/12-APK).

affected device.

Id attempt to insert a large amount of malicious data into the app's database that manages user preferences, which

plication is reinstalled.



the application to modify preferences, leading to unintended consequences.

Content Provider if the database was accessible to other apps. After database manipulation, the BestWeather app would

could hinder alarm functionality and impact users who rely on the app to wake up.

```
;maliciousIntent.setComponent(new ComponentName('com.cuiet.blockCalls', 'com.cuiet.blockCalls.SomeVulnera
```

behavior, or potentially enable other application-based attacks if the compromised data impacts critical functional

ser access to the app's functionality.

tion failure.

ssion is re-enabled, allowing the child full access to the device without the parent's knowledge. Since no coding is

potentially predict values generated by 'rand()'.

attacks or injecting malicious responses.

npredictable. Due to the known weaknesses of 'rand()', an attacker might be able to predict this value, which coul





```
uest(User requester) {    if (checkIfPrivilegedUser(requester)) {    displaySensitiveInformation();    } else {
```

foothold in the local network to carry out the attack.  
es.



irity.gentoo.org/glsa/202309-17).

isplaying a fake URL or webpage content, misleading the user into providing sensitive information.



age escalation.

tly in the background without any interaction from the user.

ot require user interaction.

ce, all without the victim being aware of the compromise.





t require such conditions.

it doesn't require any user interaction.

ller than expected allocation and subsequent heap-based buffer overflow when writing to 'buffer'.



For this exploit to happen, the attacker needs to have a presence on the local device, as remote exploitation is not indi

that should otherwise be restricted.

le to use encryption to protect API keys and other sensitive data if it must be stored on the client side.



edge, leading to a privilege escalation that could compromise the integrity of the system.

available, but developers would look at the `dhd_prot_ioctcmplt_process` function in the `dhd_msgbuf.c` file for imp

`put_size > BUFFER_SIZE}`""In this example, if the 'input\_size' exceeds the 'BUFFER\_SIZE', it would result in writing

take control of the device entirely. Due to the lack of specific code details, exact exploitation methods are not pro





rs should review the bounds of arrays or data structures whenever accessing them to prevent out-of-bounds reads

tion.



d that the above code is just a general illustration. The actual exploit for CVE-2023-21043 could be much more cor



em-level code.







wledge.

```
input()); // This input could be manipulated by an attackerif (index < sizeof(buffer)) { // Improper or missing bounds
```





privileges.

vice functionality.



e fix for this CVE.



è exploited.

privileges by chaining with other exploits.

that might be disclosed cannot be predicted without further technical analysis of the specific application and sys

s immutable, adding proper intent authorization checks, or using the latest API practices for handling PendingInte

disclosure or other unintended behaviors.







re for further attacks on the device.

revent similar vulnerabilities.

neir accounts.

formation, I cannot provide a detailed code example.

ess to the user's accounts. The vulnerability does not allow for the malicious app to modify or delete data, which li

at user interaction.

o device capabilities.

been released.



egrity of the device.



`ult = db.insert("users_table", null, values);`The correct approach would involve encrypting the password before stor

`AND password=" OR '1'='1';` This would always evaluate as true, potentially granting the attacker access regardless

able because it logs sensitive information such as usernames and passwords, which could then be read by anyone

eeding physical access once the app is running on the device.



or the user to install a malicious application unknowingly.

to provide actual code examples without risking the propagation of an exploit.











ess or interaction with the targeted system.

with location-specific phishing attacks.

information or download malware.

site.

n-in-the-middle attack and intercept or alter communications between the user and the site.

/e-by downloads to coerce victims into accessing the malicious content.

actions on the user's system or compromising the user's data.

·vpn-android/commit/981c840276ef3aee98cf5d42993d484ee99b28d9

sensitive information could be stolen.

lege level.

» adding the appropriate bounds checks in the relevant part of the firmware's code to prevent illegal memory access

» as the technical details of the vulnerability's exploit are not disclosed. The mitigation involves updating the Android





mentioned in the vulnerability's description.





re user's knowledge.

without the user's knowledge.

information disclosure.

stored in the device's memory.





ncryption rather than a specific script.

ting the URL for retrieving the MQTT broker information, which should instead use HTTPS for secure communicati

```
ies = new ContentValues();values.put("setting_name", "new_value");getContentResolver().update("content://cor
```

information or performing other malicious activities without the user's knowledge.

e hardcoded RSA public key. A vulnerable application would extract this data, decrypt it using the corresponding p

```
nponent(new ComponentName("com.cutestudio.colordialer", "com.cutestudio.dialer.activities.DialerActivity")); i
```

```
/12S2IH4Y4KZ327LI4I/
```

request like so: ``javaif (ContextCompat.checkSelfPermission(context, Manifest.permission.CAMERA) == PackageM

ge/2DMXHPRUGBUDNHZCZCIVMWAUIEXEGMGT/).

:s.fedoraproject.org/message/2DMXHPRUGBUDNHZCZCIVMWAUIEXEGMGT/).





3%82%B9%E3%82%A2%E3%83%97%E3%83%AA/id640956497).



```
it main() { char *malicious_input = "...long input that exceeds buffer size..."; vulnerable_function(malicious_inpu
```













⌘ code above is only a conceptual representation. In the case of CVE-2023-21188, the issue would be within Andre









tion of wldata.cpp.















al information; therefore, specific code examples are not provided here.

```
ring authType) {}    public void checkServerTrusted(X509Certificate[] certs, String authType) { }    });    ctx.init(null
```

```
tion == 'corrupt') { corruptFile(dbFile); }}
```

In this pseudo-code, the fileName is taken from user input without valid

and modify a setting, such as user privilege levels, without proper authorization.

lead to double free if the memory was already freed during the registration process, leading to memory corruption





ability.



ch eventually leads to a crash.

uld likely crash upon accessing the altered or corrupted data, leading to a persistent DoS situation.

```
ableComponent')));maliciousIntent.putExtra('extra_data_key', 'data_to_delete');sendBroadcast(maliciousIntent);`
```

lity.

; involved from the attacker's side, a code example isn't applicable in this context.

ld lead to security breaches like cache poisoning or request interception.







// Improper access control issue if non-privileged users are not adequately blocked      displaySensitiveInfo





















cated in the CVE description.



proper input validation handling, which is causing the vulnerability.

past the end of the buffer, potentially overwriting adjacent memory and leading to unexpected behavior or system



vided here.





; or writes.

complex and depend on specific kernel structures and timing to manipulate memory after it has been freed, potentially













check    printf('%c', buffer[index]); // Potential out-of-bounds read}```In this oversimplified example, if `getUserInp















tem environment.

nts.















imits the potential impact but still represents a significant security risk.









bring it, using a secure hashing algorithm or encryption method.

is of the legitimate user's password.

: with access to the log files.



























ess.

oid kernel to a version where the vulnerability has been patched.





















on.

n.example.agent.provider/settings", values, null, null);The above code demonstrates a third-party app attempting

private key, and apply the server settings, ultimately compromising the application. To mitigate this, developers sh

```
intent.setData(Uri.parse("tel:+123456789")); startActivity(intent);``In this scenario, the phone call would be initial
```



```
anager.PERMISSION_GRANTED) { // Permission is granted, start camera startActivity(intent);} else { // Reque
```











t); return 0;}```In a real-world scenario, the overflow could lead to out of bounds memory access and potential da













oid's Bluetooth stack implementation and could involve more complex data structures and operations.





























```
, trustAllCerts, new SecureRandom()); // Setting the SSL context to the service ((MyService.LocalBinder) service
```

lation, which could allow an attacker to provide a path or actions that cause the database to malfunction.

























```
ormation();    } } // Dummy placeholder for actual privilege checking mechanism    private boolean checkIfPri
```





























n compromise.











ally overwriting critical data or function pointers to gain escalation of privilege.











`out()` can be influenced by an attacker to provide an index beyond the bounds of `buffer`, it could lead to an out-of-



































































































; to update settings through the exposed content provider without any user authorization or privilege checks.

ould use secure techniques for validating the integrity and source of received intents, avoid hardcoding keys, and

ted by exploiting the improperly exported DialerActivity component.

st permission from the user    `ActivityCompat.requestPermissions(activity, new String[] {Manifest.permission.CAN`













ta exposure as seen with CVE-2023-21224.









































```
e).setSSLContext(ctx); } ...};``This example demonstrates the improper trust of al
```





























```
vilegedUser(User user) {    // Incorrect implementation might
```





















































f-bounds read. Mitigation would involve properly validating ``index`` before using it to access elements of ``buf``

































































































implement robust e





1ERA}, CAMERA\_RE









































































































































