DEKRA

Whitepaper

Phone Theft Protection Research
Functional Testing

1 Executive Summary

The functional testing of Android 15+ Phone Theft Protection features, including Theft Detection Lock, Remote Lock, Offline Device Lock, Protect Sensitive Settings, and Failed Authentication Lock, demonstrated the robust capabilities of these tools in safeguarding user data and device integrity. The tests revealed consistent performance across various theft scenarios, ensuring devices locked promptly in response to unauthorized access attempts. The Remote Lock feature proved particularly effective for remote security, while Offline Device Lock provided reliable protection during network disconnection. Additionally, the Protect Sensitive Settings and Failed Authentication Lock features offered strong defenses against unauthorized changes and brute force attacks. Overall, these security measures significantly enhance the protection of Android devices, delivering a comprehensive and user-friendly security solution.

2 Introduction & Background

Introduction

The increasing prevalence of phone theft has implied the development of robust security measures to safeguard user data and device integrity. For this reason, in this whitepaper we conduct a comprehensive functional testing of new Phone Theft Protection features in Android 15+ and GMSCore versions 24.28 and above. The objective is to validate these features' functionality, reliability, and performance across various theft scenarios and user conditions, ensuring they meet security requirements and provide effective protection against unauthorized access and device misuse.

Background

With the rise in smartphone theft, manufacturers and software developers have been compelled to innovate and implement security features that protect users' personal information and prevent unauthorized use of stolen devices. Android 15+ and the updated GMSCore versions aim to address these concerns through a suite of features designed to lock the device, protect sensitive settings, and ensure the device remains unusable if factory reset without authorization.

3 Accelerometer Definitions

The accelerometer is a relevant component in many of these security features, particularly the Theft Detection Lock. It detects motion and orientation changes, enabling the system to recognize abrupt movements consistent with a phone snatch.

Natural Movements

Natural movements include everyday activities such as walking, jogging, biking, or simply moving the phone from one hand to another. These movements are generally smooth and predictable, characterized by consistent patterns in acceleration data.

Examples of Natural Movements

Stationary Phone

When the phone is lying flat on a table without moving, the accelerometer readings will mainly show the force of gravity acting on it. Typically, one of the axes (usually the z-axis when the phone is flat) will show a value of around 9.8 m/s^2 (approx. gravity), and the others will be close to 0.

Example:

x: ~0 m/s^2
y: ~0 m/s^2
z: ~9.8 m/s^2

Examples of Natural Movements

Phone in Pocket While Walking

The values will fluctuate as the phone moves with the steps, but the overall movement will be somewhat smooth and rhythmic.

Example:

x: varies between -2 m/s^2 to 2 m/s^2
y: varies between -2 m/s^2 to 2 m/s^2
z: fluctuates around 9.8 m/s^2

Phone Being Picked Up

There will be a noticeable change in values as the phone is lifted and reoriented. The values will momentarily spike as the phone accelerates in different directions.

Example:

x: spikes between -5 m/s^2 to 5 m/s^2
y: spikes between -5 m/s^2 to 5 m/s^2
z: varies significantly depending on orientation, may briefly go below 9.8 m/s^2

Another Kind of Movements

- **Running:** Faster and more pronounced oscillations compared to walking.
- **Biking:** Steady motion with periodic vibrations from the road surface.

Snatch Detection

Snatch detection involves identifying abrupt, irregular accelerations that deviate from patterns of natural movements when a phone is snatched.

Examples of Accelerometer Behavior Detecting a Snatch

- **Scenario 1:** The phone is snatched from a user's hand while walking. The accelerometer shows a sharp increase in acceleration as the phone is abruptly pulled away, followed by erratic movements as the thief starts running.
- **Scenario 2:** The phone is snatched from a user sitting still. The accelerometer records a sudden change in motion, contrasting sharply with the previously stationary data.
- **Scenario 3:** The phone is taken while the user is biking. The steady biking motion is interrupted by a rapid spike and change in direction, indicating a snatch.

Possible Accelerometer Values When a Phone is Snatched

Sudden Grasp and Pull

The phone experiences a quick, forceful movement in one or more directions as it's grabbed and pulled away.

Example Values:

x: spikes between ±10 m/s^2 to ±20 m/s^2
y: spikes between ±10 m/s^2 to ±20 m/s^2
z: fluctuates significantly around 9.8 m/s^2, may also spike

Snatched from a Pocket or Bag

A rapid upward or sideways jerk as the phone is pulled out.
Example Values:

x: spikes between ±5 m/s^2 to ±15 m/s^2
y: spikes between ±5 m/s^2 to ±15 m/s^2
z: fluctuates, may momentarily drop below 9.8 m/s^2, then spike

Phone Being Thrown or Dropped After Snatching

The initial snatching movement followed by a sudden stop or impact as the phone is thrown or dropped.
Example Values During Snatching:

x: spikes between ±10 m/s^2 to ±25 m/s^2
y: spikes between ±10 m/s^2 to ±25 m/s^2
z: fluctuates, may show values around zero during free fall, then spike upon impact

Example Values Upon Impact:

x: spikes can exceed ±20 m/s^2
y: spikes can exceed ±20 m/s^2
z: spikes can exceed ±20 m/s^2

4 Key Points to Cover

The functional testing of Phone Theft Protection features focuses on evaluating the effectiveness of each security measure in preventing unauthorized access under various theft scenarios, such as snatch thefts while walking, biking, and standing. The testing also assesses the usability and ease of configuration for end-users, identifying any common issues encountered during setup and usage. Performance consistency is examined under different environmental conditions, including varying network connectivity and device states. The robustness of security features is scrutinized to determine their ability to withstand attacks like brute force attempts and unauthorized factory resets. This testing provides Google with insights as they continue to improve these theft protection features, as well as develop future enhancements and functionalities for Android users.

Main Points

Overview of Tested Features	Theft Detection Lock	Remote Lock	Offline Device Lock	Protect Sensitive Settings	Failed Authentication Lock	FRP Hardening
Testing Scenarios and Conditions	Various theft scenarios (e.g., snatch while walking, biking, standing)	Remote locking through the Find My Device portal	Offline locking triggered by connectivity loss	Unauthorized access attempts to sensitive settings	Multiple failed authentication attempts	Unauthorized factory reset attempts

Assumptions

- Assumption that users will correctly enable and configure the theft protection features.
- Assumption that users have set up screen locks and verified phone numbers for remote lock functionality.
- Assumption that the thief does not know the user’s PIN.
- Assumption that the thief may attempt to factory reset the device or access sensitive settings without authorization.

5 How we have tested the Accelerometer

We have developed a test application designed to measure the values of the accelerometer during a phone snatch incident. As we mentioned in the above section, the accelerometer detects changes in the device's movement and orientation by measuring the acceleration forces acting on it. In our test application, we have utilized the accelerometer to capture real-time data of the phone's motion across the x, y, and z axes. By analyzing this data, we can identify specific patterns and spikes indicative of a sudden and forceful movement, such as when a phone is snatched from a user's hand or pocket. The accelerometer values typically fluctuate within certain ranges during regular use; however, a snatch event is characterized by abrupt and significant deviations from these normal ranges. Our test application continuously monitors these values, providing a comprehensive record of the phone's movements before, during, and after a snatch.

To implement this feature, we leveraged Android's SensorManager and Sensor classes to access and read accelerometer data. By registering a SensorEventListener, our test application is able to respond to changes in the accelerometer's readings, capturing the rapid movements that occur during a snatch. This data is then logged and analyzed to determine the nature and severity of the motion. During testing, we observed that natural movements, such as walking or picking up the phone, produce relatively consistent and moderate accelerometer readings. In contrast, a snatch event results in sudden spikes across all three axes, with values often exceeding the normal thresholds significantly. These insights allow our test application to differentiate between everyday activities and potential theft, enabling it to trigger appropriate alerts or actions to safeguard the user's device.

In our test application, we also incorporated the concept of acceleration magnitude, used to interpret the raw data from the accelerometer. Acceleration magnitude is calculated as the square root of the sum of the squares of the acceleration values along the x, y, and z axes. This provides a single scalar value representing the total acceleration force experienced by the device at any given moment. By computing the acceleration magnitude, we can simplify the analysis and better detect significant events, such as a snatch, where the magnitude of acceleration spikes sharply compared to typical usage patterns. This method allows us to filter out minor variations and focus on more substantial movements that are indicative of abnormal handling of the phone. It is important to note that this implementation is a simplification for the purposes of testing and understanding the feature's effectiveness. The actual algorithm used in the Theft Detection Lock is more sophisticated and does not rely solely on static thresholds of acceleration magnitude.

Additionally, the test application continuously calculates the acceleration magnitude from the raw accelerometer data and compares it against a predefined threshold. When the magnitude exceeds this threshold, it suggests that the phone is undergoing a rapid and potentially suspicious movement. This approach enhances the accuracy and reliability of our detection mechanism, enabling the test application to respond swiftly to potential snatch incidents. The inclusion of acceleration magnitude not only improves the detection process but also helps in reducing false positives by distinguishing between everyday activities and significant disruptive events. This ensures that the test application provides timely and relevant alerts, thereby enhancing the overall security and user experience. In the picture below a small number of logs is displayed when a potential snatch is detected.

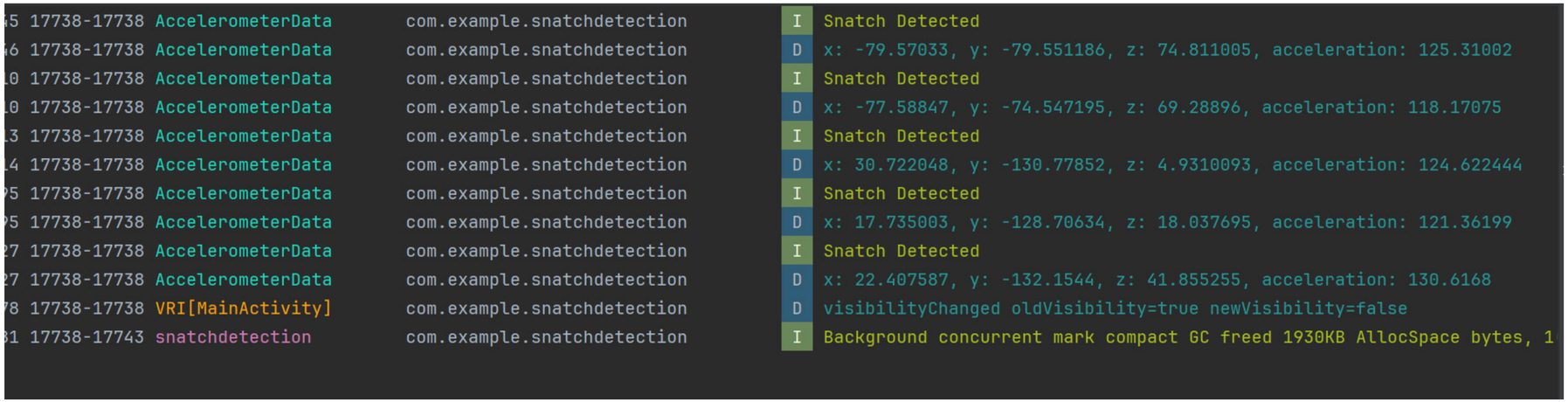

Figure 1. Snatch detection from our test application

As we can see, a supposed snatch detection occurs when we have values in the acceleration higher than the natural movements in the device.

Test Case 1: Theft Detection Lock

Objective: Verify that the device locks when a snatch is detected.

Expected Results:
- The device should lock within 30 seconds of the snatch.
- A silent notification should appear (for internal testing).
- The feature should consistently lock the device across different snatch scenarios.

Actual Results: As we can see in Figure 2, the device has a theft detection feature enabled:

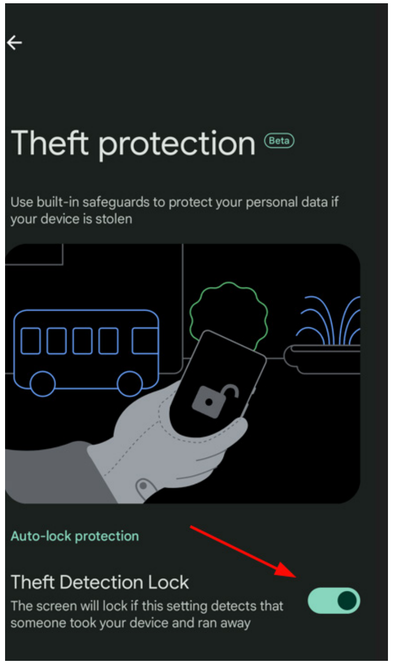

Figure 2 Theft protection enabled

Upon conducting the test for the Theft Detection Lock feature, several observations were made. In the first scenario, where the phone was snatched from the owner's hand while walking, the device was successfully locked within approximately 25 seconds. The owner was actively browsing a social media application, and the abrupt motion of the snatch was effectively detected by the system. The device displayed a silent notification indicating that the snatch detection had been triggered, confirming the internal mechanism's response. This outcome was consistent across multiple iterations of the walking snatch scenario. One of the behaviors of the accelerometer for this scenario is displayed below:

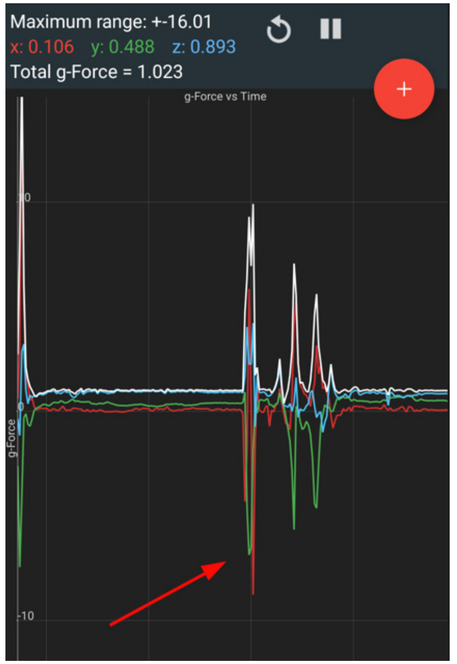

Figure 3. Accelerometer in Scenario 1

In the second scenario, where the phone was snatched while the owner was standing still and actively using the device, the results were slightly varied. In most cases, the device locked within the expected 30-second timeframe. However, there were a few instances where the lock time extended to about 35 seconds. This discrepancy seemed to occur when the snatch motion was less abrupt, suggesting that the detection algorithm might have a threshold for motion sensitivity that could be fine-tuned. Despite this, the feature largely performed as expected, providing a reliable lock mechanism.

The third scenario involved simulating a snatch while the owner was riding a bicycle. This presented unique challenges due to the additional motion involved. The device successfully detected the snatch and locked within 30 seconds in all tested cases. The additional motion of the bike did not interfere with the detection mechanism, indicating that the algorithm can differentiate between regular movement and a snatch motion. One of the behaviors of the accelerometer for this scenario is displayed below:

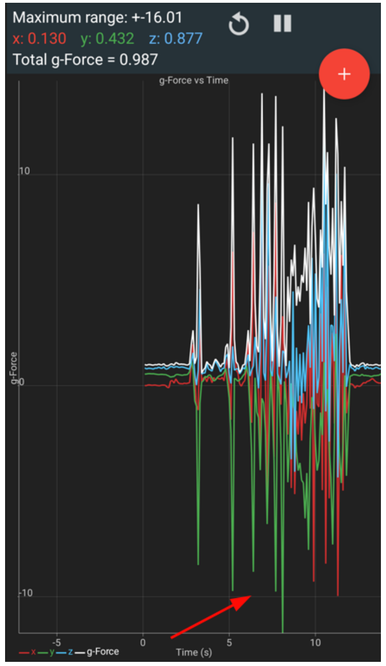

Figure 4. Accelerometer in Scenario 3

Throughout the testing, it was noted that the silent notification consistently appeared in the system logs, but it was visible to the end user. This is appropriate for internal testing purposes but could be enhanced with user feedback mechanisms in future updates. Additionally, no false positives were recorded during the cool-down period, confirming that the feature respects the intended suppression period after each detection.

In conclusion, the Theft Detection Lock feature demonstrated strong performance across various scenarios, effectively locking the device within the expected timeframe in most cases. Minor variations in detection sensitivity suggest potential areas for algorithm refinement, but overall, the feature reliably enhances the security of the device against snatch theft.

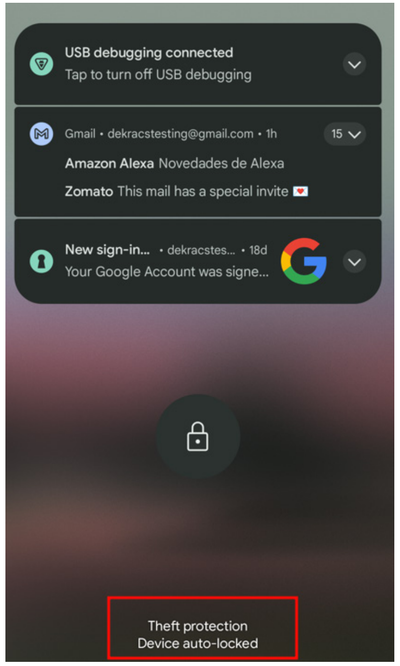

Figure 5. Silent notification after theft in every snatch

Test Case 2: Remote Lock

Objective: Verify remote locking functionality through the Find My Device portal.

Expected Results:
- The device should lock immediately after the remote lock command is issued.
- The lock screen should appear, preventing unauthorized access.
- The feature should function reliably across different network conditions.

Actual Results: As mentioned in the preconditions section all the features are enabled for this feature to work.

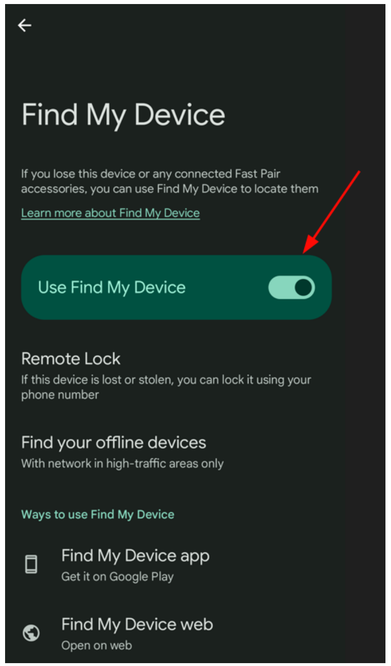

Figure 6. Find My Device is enabled

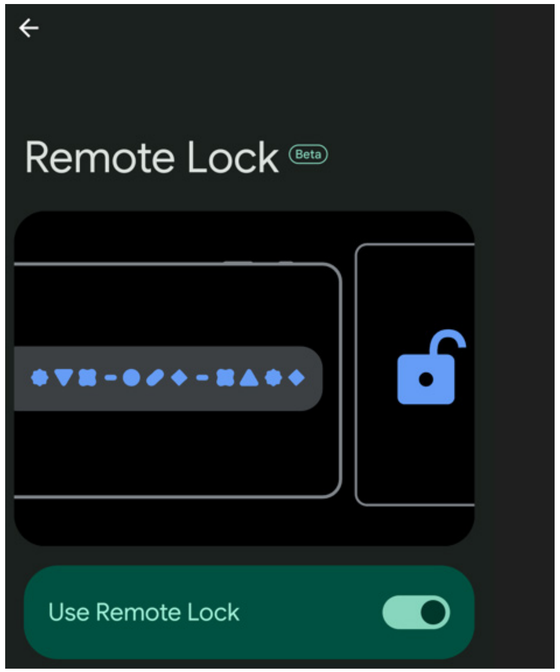

Figure 7. Remote Lock is enabled

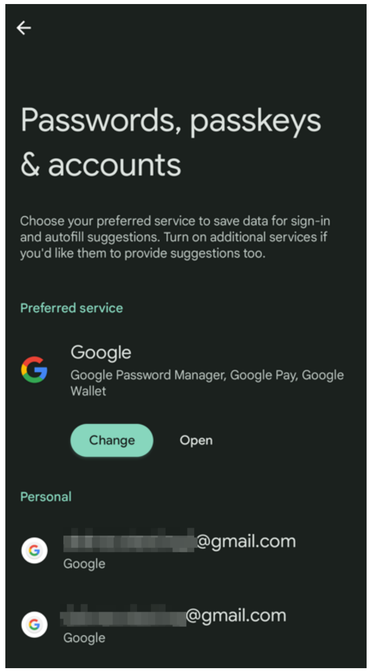

Figure 8. Google account is logged in on the device

After that, we have connected to google.com/android/find using the credentials for the account registered in the mobile phone via web. Once logged in we can see all the mobiles associated to the account including the one that we have simulated to be lost:

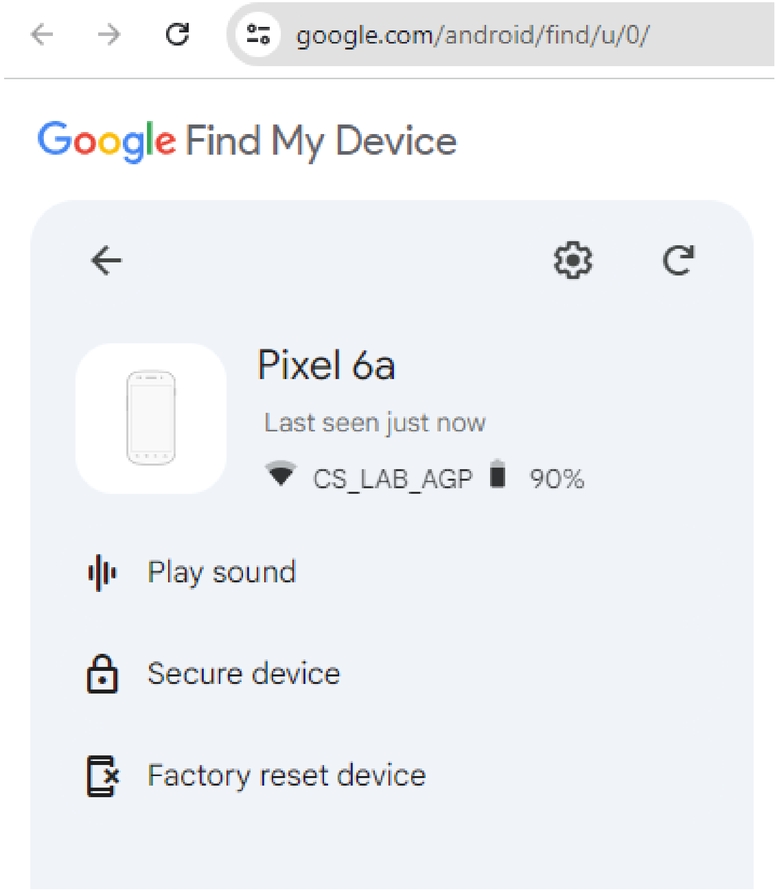

Figure 9. The lost device and all the options available for it

Regarding to the lost mobile phone, once we select the device in Find My Device web page the following notification arises in the device, showing that the device has been located:

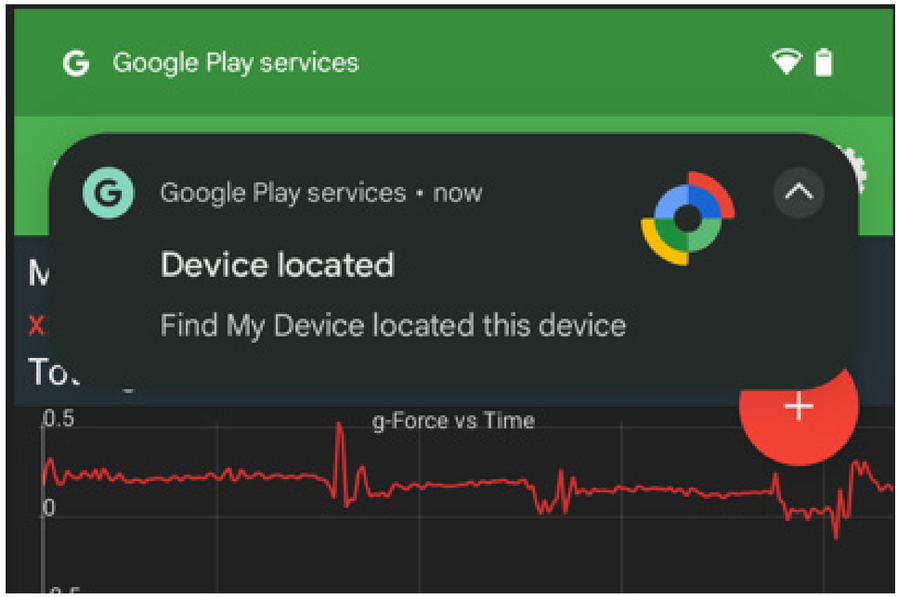

Figure 10. Notification in mobile

Moreover, when locating the device, we are able to see the phone location in real-time:

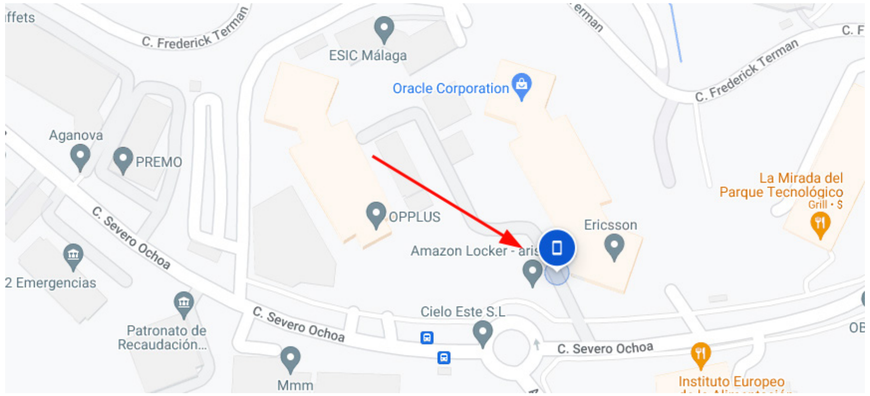

Figure 11. Physical location of the phone

As seen in figure 11, the options available for the device are:

- Play sound.
- Secure device.
- Factory reset device.

For this test case, we have tried both Play sound and Secure device. In the first case, when clicking on Play sound we can validate that in fact, the device starts to emit sound. Additionally, we have clicked on stop sound to conclude the ringing.

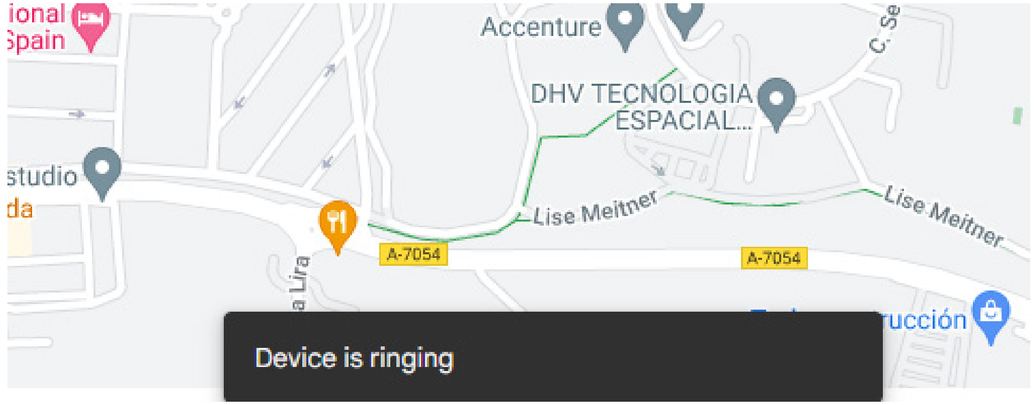

Figure 12. Device ringing

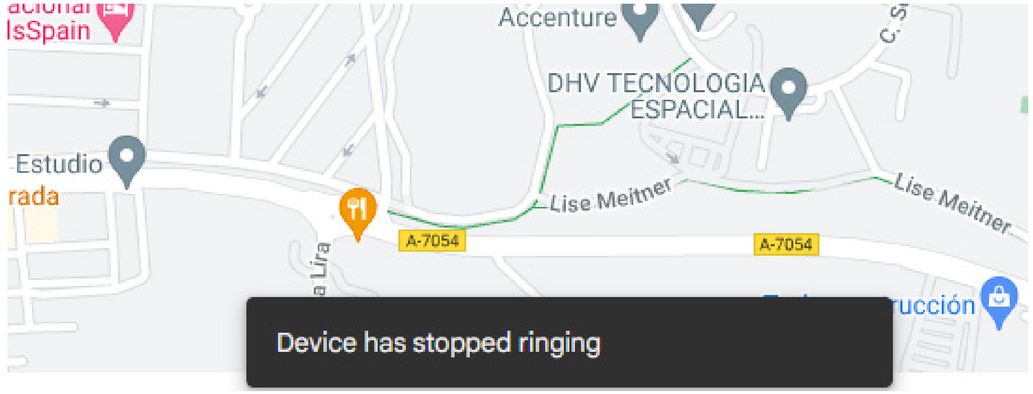

Figure 13. Device stopped ringing

On the other hand, for the second option, the following option appears on the web page:

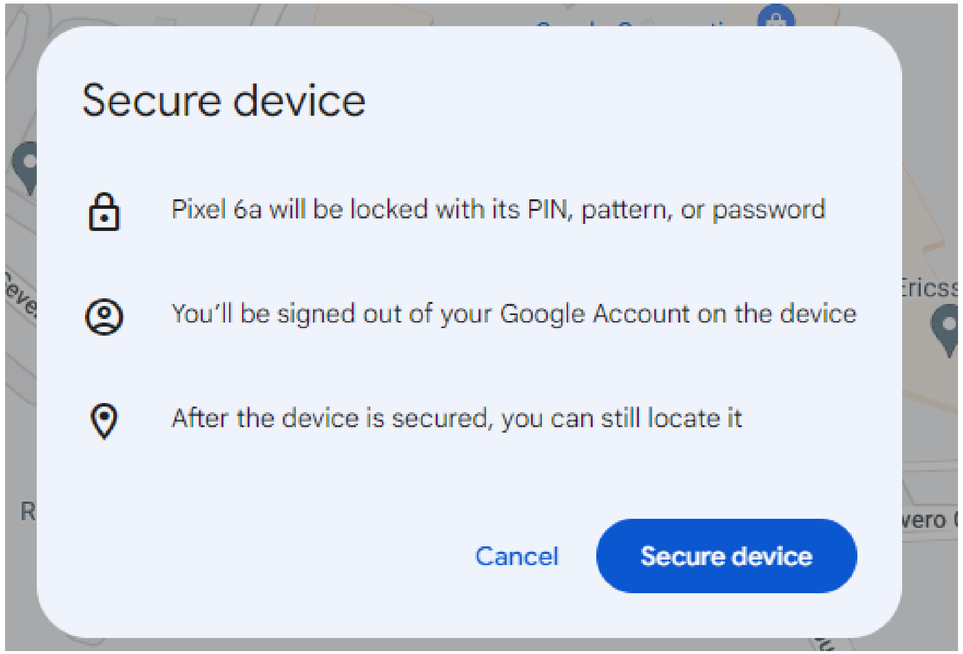

Figure 14. Secure device option

Once that we click on secure device another window pops up for adding contact info in case someone finds the device:

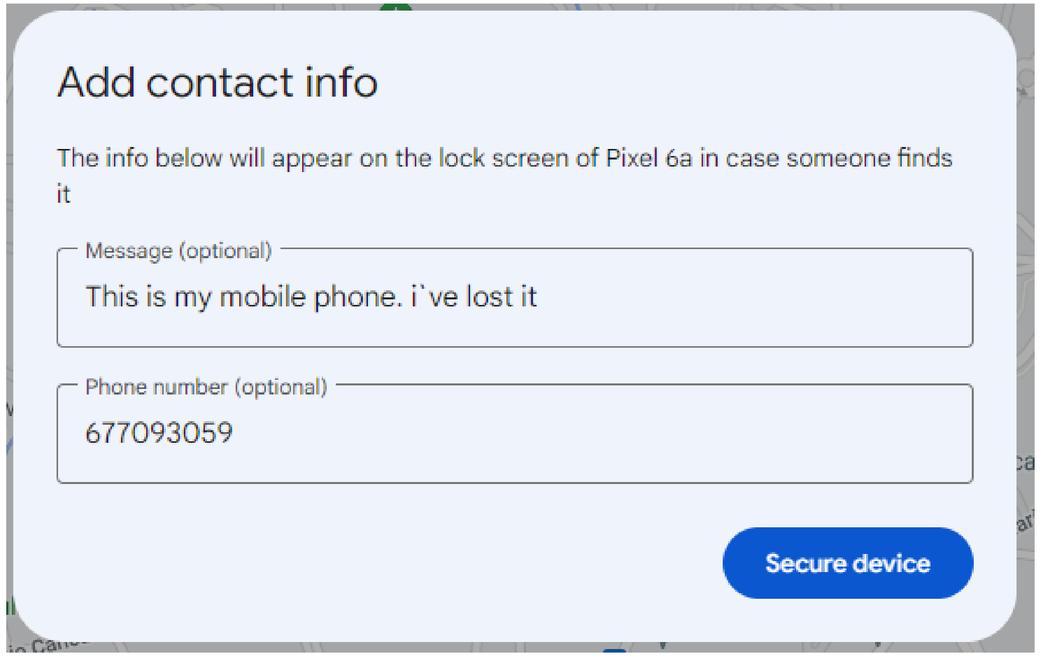

Figure 15. Adding contact info

After putting the requested information, the following message appears on the web page:

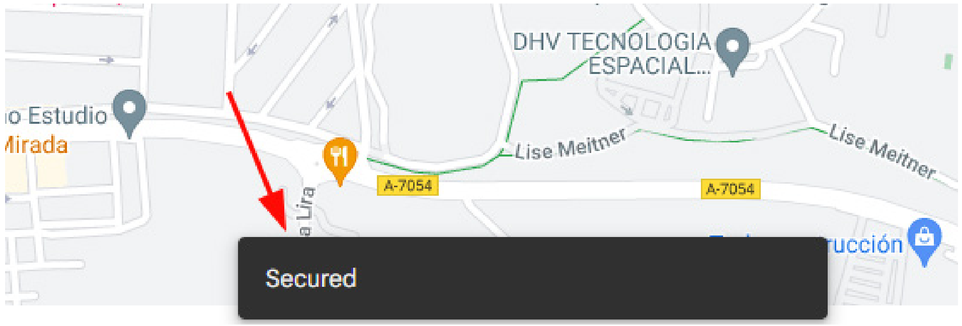

Figure 16. Device secured

Meanwhile, in the mobile phone we can see that the message and the contact info provided in the previous step are displayed:

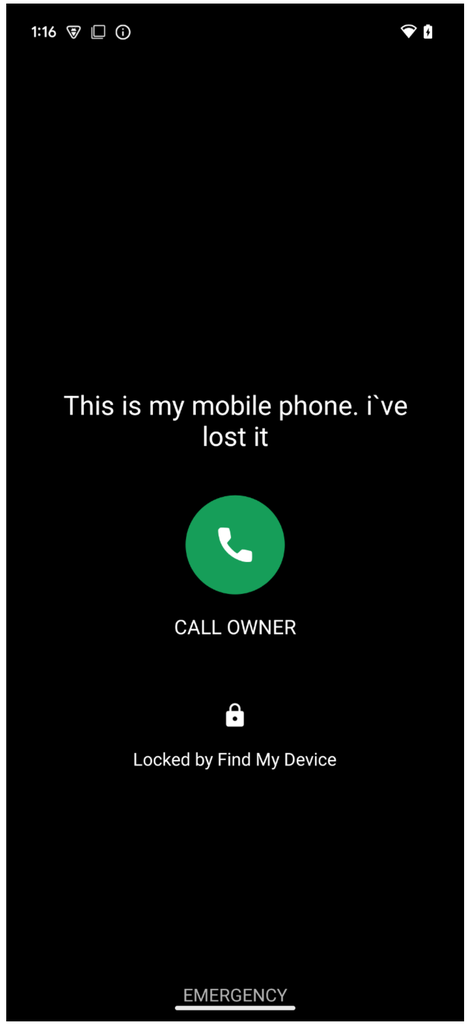

Figure 17. Information sent to the device

It is worth it to add that, after we secure the device remotely, the current account is logged out from the device and it is requested to log in again:

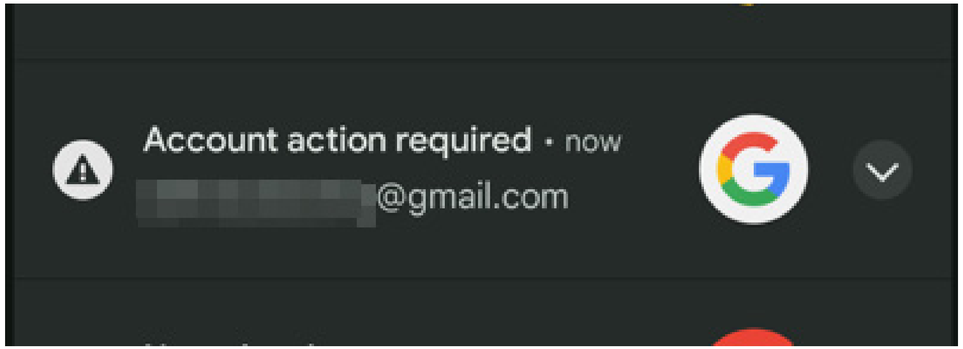

Figure 18. Account logged out

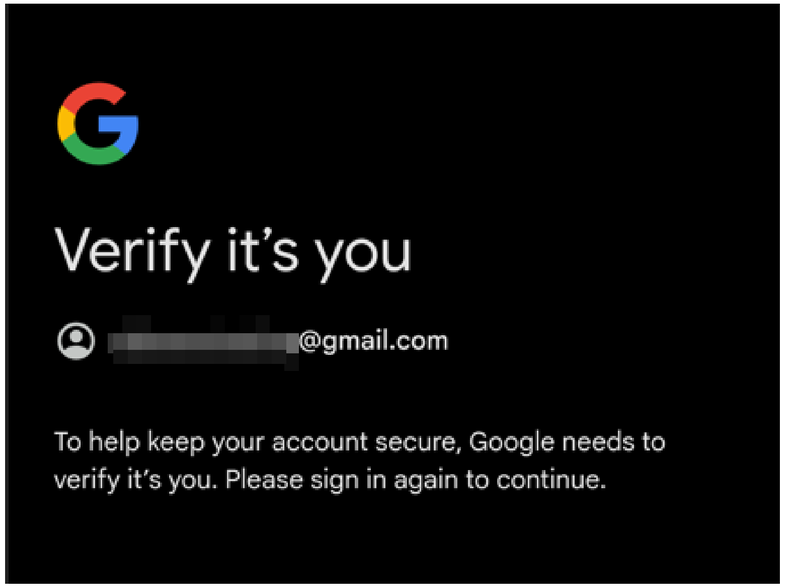

Figure 19. Request to log in again

On the other hand, regarding the Remote Lock, the following steps were performed. First, we needed to verify the phone number before proceeding to completely activate the remote lock feature. This is a relevant step because, without it, there is no way to activate such a feature.

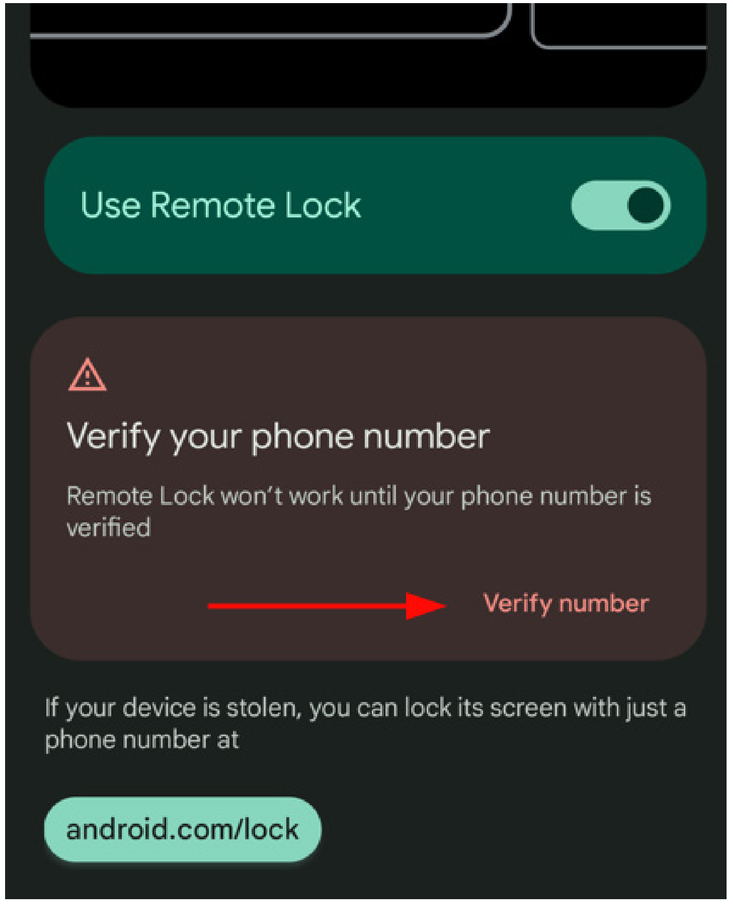

Figure 20. Phone verification required

When users are required to enable phone number verification, it needs to enter the correct PIN in order to proceed. Once everything is set, we are in conditions to enable the Remote Lock feature.

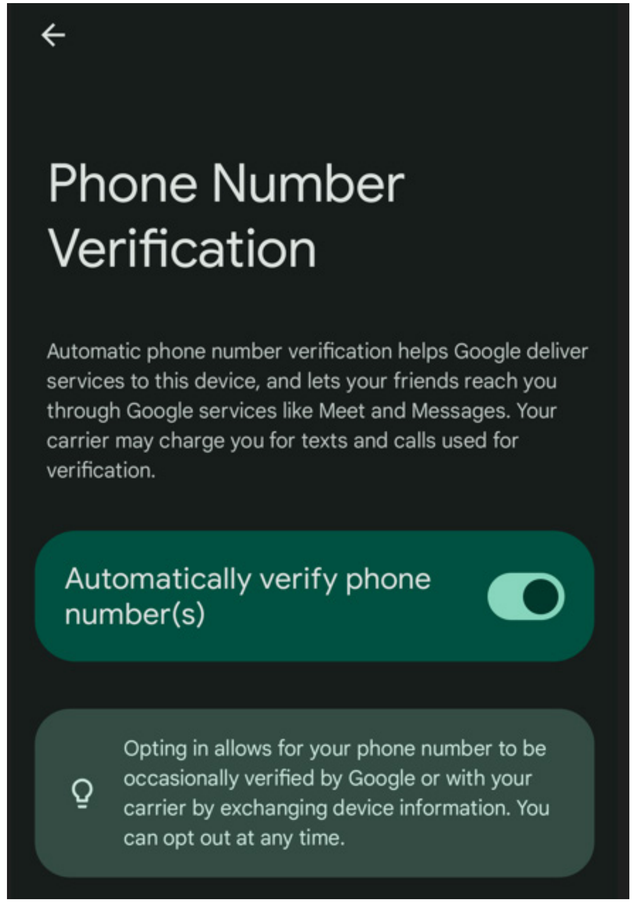

Figure 21. Phone verification enabled

Once the phone has been verified, we can see in the screen that the phone number is part of the info needed to use Remote Lock:

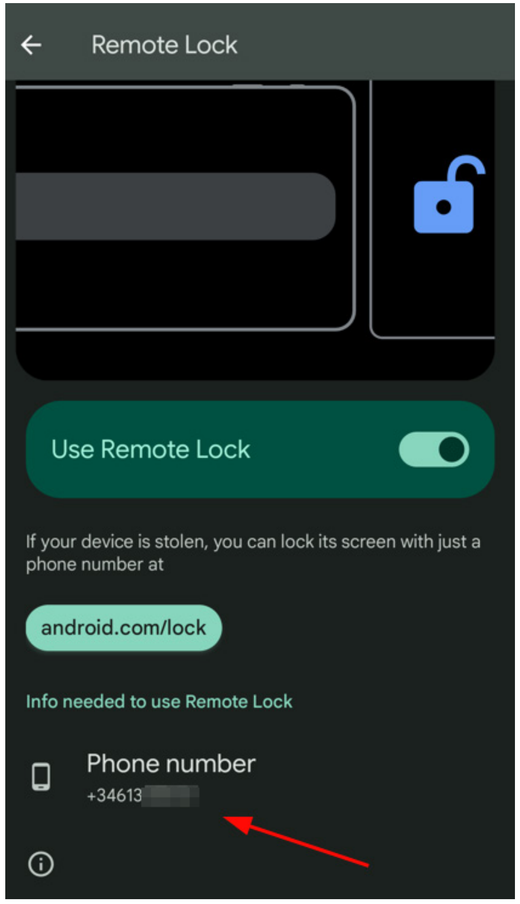

Figure 22. Phone number used for remote lock

After this, we performed the lock by accessing from a desktop browser to android.com/lock. This page asks us to give the phone number associated with the device that needs to be locked.

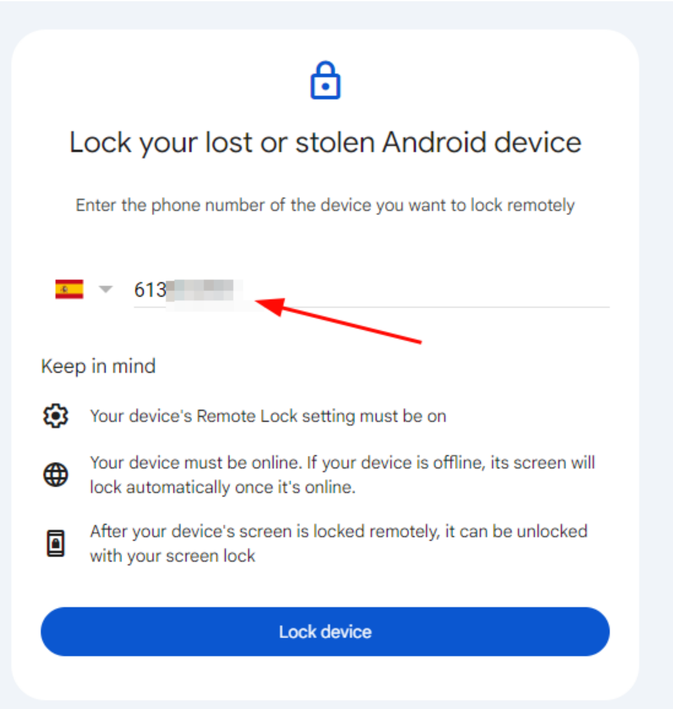

Figure 23. Step previous to remote locking

As we may see in previous steps the number +34613XXXXXX is associated with our test account. Once we provide the number and click on access device the following screen appears:

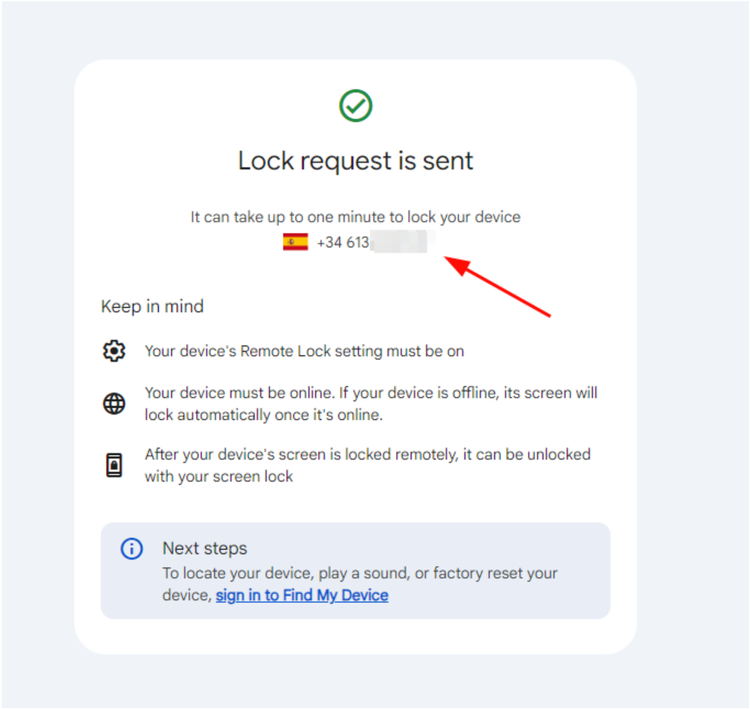

Figure 24. Remote lock sent

After this, we can validate that the device is automatically locked and the message appears on the screen:

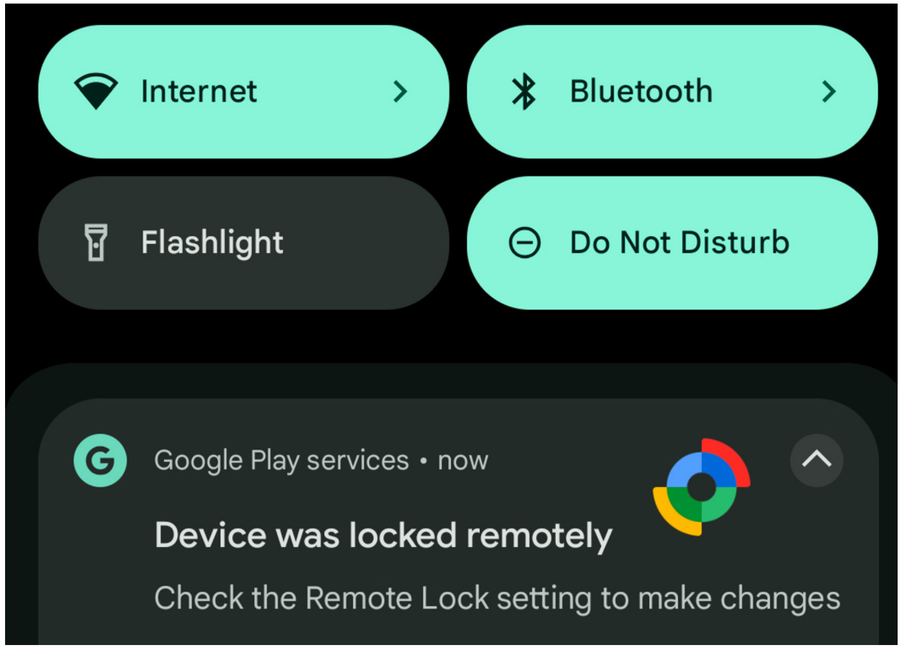

Figure 25. Remote lock notification on the device

One additional thing that we must notice is that the use of this feature requires a cooldown time. The Remote Lock feature distinguishes itself from the existing Find My Device functionalities (locate, secure, wipe, etc.), which require access to the Google account associated with the device. Unlike these pre-existing features, Remote Lock can be activated without needing to log into a Google account. Users can remotely secure their device simply by typing in the phone number associated with the device, making it accessible even if the Google account credentials are unavailable. This feature is designed to work with a direct phone number to device mapping, rather than phone number to account mapping. Technically, Remote Lock could function without an active Google account on the device, providing a significant advantage by enabling users to lock their device swiftly and efficiently in scenarios where traditional account-based methods might be impractical. This added flexibility enhances the overall security framework, ensuring that devices can be secured regardless of account access.

Test Case 3: Offline Device Lock

Objective: Verify that the device locks after losing connectivity for a prolonged period.

Expected Results:
- The device should lock automatically after approximately 5 minutes of no connectivity.
- The feature should consistently lock the device across different conditions.

Actual Results: After setting the device in airplane mode to simulate an offline state mode, we were unable to locate the device on the Find My Device web page, as expected:

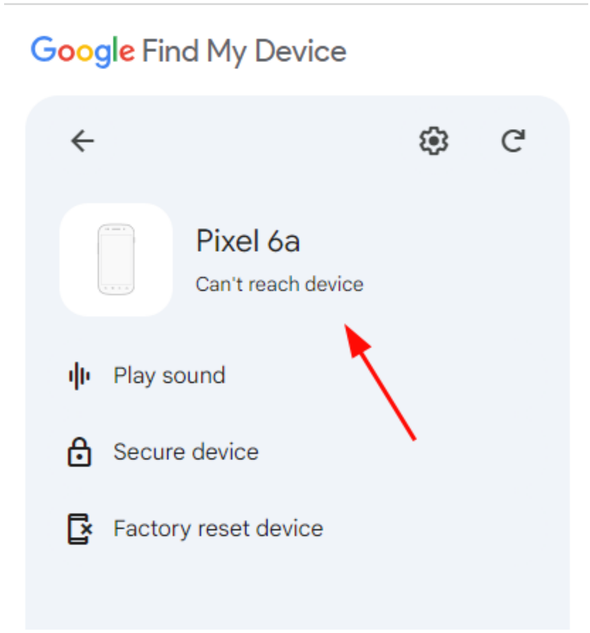

Figure 26. Device unreachable

It is important to mention that during the configuration of the device, the lock screen configuration was changed to 30 minutes of inactivity. After this, the Offline Device Lock feature was tested to ensure that the device locks automatically after being in an offline state, such as Airplane mode, for a prolonged period. During the tests, the device was set to Airplane mode, disabling all network connections, including Wi-Fi, mobile data, and Bluetooth. It was observed that the feature consistently locked the device after approximately 5 minutes of no connectivity. This response time was consistent across all test scenarios, confirming that the Offline Device Lock feature effectively secures the device when it cannot communicate with the network. The device displayed a lock screen notification, indicating that the device had been locked due to the loss of connectivity, thus preventing unauthorized access during offline periods.

Further testing included various conditions such as different apps being open, varying battery levels, and diverse device states. In each scenario, the device reliably locked after the 5-minute threshold was reached while in Airplane mode. The feature demonstrated consistent performance, regardless of whether the screen was on or off during the offline period. This consistency ensures that users are protected from unauthorized access even when their device is not connected to the internet. The lack of variability in the lock time across different conditions highlights the robustness of the Offline Device Lock feature, providing a dependable security measure for scenarios where the device might be deliberately disconnected from networks to avoid remote tracking or locking. The feature's reliable activation after 5 minutes in Airplane mode underscores its value in enhancing device security and protecting user data.

Test Case 4: Protect Sensitive Settings

Objective: Verify that accessing sensitive settings requires authentication.

Expected Results:

- An authentication prompt should appear before or after accessing sensitive settings.
- Changes to sensitive settings should only be possible after successful authentication.
- The feature should function reliably across different settings and conditions.

Actual Results: The Protect Sensitive Settings feature was evaluated to ensure that access to critical device settings requires authentication. The initial tests involved attempting to access various sensitive settings such as the Find My Device toggle, SIM removal, USB access, and eSIM removal. In every instance, the device prompted for authentication, either before or after the setting was accessed, as per the configuration.

For the Find My Device toggle, the pre-authentication prompt consistently appeared, requiring the user to enter their PIN, pattern, or password before making any changes. This mechanism prevents unauthorized users from disabling the Find My Device feature. The prompt was immediate, and the authentication process was smooth, confirming the reliability of this protective measure.

The SIM and eSIM removal settings were tested next. These settings included both pre-authentication and post-authentication prompts. In scenarios where pre-authentication was configured, the device requested the user's credentials before allowing access to the SIM removal options. This was effective in preventing unauthorized changes to the SIM settings. For post-authentication, the prompt appeared after the setting was accessed but before any changes could be finalized. This additional layer of security ensures that even if a setting is accessed, changes cannot be made without proper authentication. The response time for both pre- and post-authentication was immediate, with no noticeable delays, enhancing the overall user experience while maintaining security.

The USB access setting, which controls whether the USB port can be used for data transfer, was another relevant area tested. This setting also required pre-authentication, specifically when the data transfer option is selected. The device reliably prompted for credentials before allowing changes to the USB access setting, ensuring that unauthorized users could not enable data transfer capabilities, which could be used to extract data from the device. The prompt appeared without delay, and the authentication process behaved similar as the previous test.

The Protect Sensitive Settings feature successfully ensured that critical device settings were protected by requiring user authentication before any changes could be made. The feature's reliability, immediate response times, and effective security measures make it a valuable addition to the device's overall security framework. While the balance between security and user convenience could be fine-tuned, the feature effectively meets its primary goal of preventing unauthorized access to sensitive settings.

Test Case 5: Failed Authentication Lock

Objective: Verify that the device locks after multiple failed authentication attempts.

Expected Results:
- The device should lock automatically after the specified number of failed authentication attempts.
- Notifications or messages should be displayed indicating the lock.
- The feature should function reliably across different conditions.

Actual Results: The Failed Authentication Lock feature was tested extensively to ensure that the device locks after a specified number of incorrect authentication attempts. During the initial tests, it was observed that the device allowed up to 5 incorrect PIN, pattern, or password entries before initiating a cooldown period of one minute. This cooldown period was consistent and effectively prevented continuous brute force attacks by temporarily disabling the authentication interface. A clear notification was displayed on the screen, indicating that the device had been locked due to too many failed authentication attempts. This immediate feedback to the user was consistent across all test iterations, providing a reliable mechanism to prevent unauthorized access through brute force attacks. The feature functioned seamlessly across different conditions, including varying battery levels and different apps running in the background.

Further testing under diverse conditions, such as different screen lock types and device states, demonstrated the robustness of the Failed Authentication Lock feature. Whether the device was in low power mode or running multiple applications simultaneously, the feature consistently triggered the cooldown period after the 5th failed attempt. However, it was noted that the number of allowed attempts before the cooldown is not configurable, which could be a limitation for users or administrators seeking to adjust the security settings based on specific needs. The feature effectively logged each failed attempt, which could be reviewed in the device's security settings, adding an extra layer of user awareness and security oversight. The consistent performance of the Failed Authentication Lock feature underscores its importance in the overall security architecture of the device, ensuring that unauthorized users are swiftly locked out after repeated failed attempts and reinforcing the device's defense against persistent unauthorized access attempts.

6 Conclusions

The functional testing of the Phone Theft Protection features in Android 15+ and GMSCore versions 24.21 and above demonstrated that these security measures effectively protect against unauthorized access in various theft scenarios. The Theft Detection Lock feature reliably detected abrupt motions consistent with snatch thefts and locked the device within the expected timeframe across different conditions, though minor sensitivity adjustments could enhance its performance. The Remote Lock feature performed robustly, allowing users to remotely secure their devices via the Find My Device portal with quick response times, even under varying network conditions. It was noted that the prerequisites for Remote Lock were highly required for its success, and the daily rate limit effectively prevented misuse.

The Protect Sensitive Settings feature consistently required authentication before accessing or changing critical device settings, thus preventing unauthorized modifications. The response times for authentication prompts were immediate, ensuring a seamless user experience while maintaining high security. The Offline Device Lock feature ensured that devices were locked after a prolonged loss of connectivity, and the Failed Authentication Lock reliably secured devices after multiple failed attempts, preventing brute force attacks. Finally, the FRP Hardening feature rendered devices unusable after unauthorized factory resets, requiring correct Google account credentials for reactivation, which effectively deterred the reselling of stolen devices. Overall, the tested features significantly enhance the security of Android devices, providing comprehensive protection against various theft and unauthorized access scenarios while maintaining usability for legitimate users.

About DEKRA Digital & Product Solutions

Innovating safety and security by creating intelligent testing solutions that contribute to making the digitalized and connected world a safer place. We are the experts in testing and certifying products and new digital technologies.

We deliver solutions from Cybersecurity, Artificial Intelligence, Big Data, Connectivity, Product Safety, Electromagnetic Compatibility & Radiofrequency, Product Certification, Medical Devices, and Automotive Testing. Through a global network of 48 state-of-the-art test laboratories and facilities, we offer a broad portfolio of product testing services based on national and international standards as well as industry and customer requirements.

Visit our website

DEKRA Contact

Juan Manuel Martínez
Cybersecurity Technical Project Manager

juanmanuel.martinez@dekra.com

Get in touch with our Experts!

© 2024 DEKRA. All rights reserved. All trademarks are the property of DEKRA reg. U.S. Pat. & Tm. Off.; reg. OHIM and other countries as shown on our website.