



A WAY TO INVALIDATE ALTERATIONS IN A DATABASE STORED DATA FROM INSIDERS AND TO AVOID LOCATION-BASED DATA DECEIT.

The case of the ChinaBeeReady platform by Toorbee.

Georgios Chiotis, Aristotelis Skamagkis, Spyros Zavitsanos

Software Development Department,

Toorbee,

Vrilissia, Greece,

Abstract: ChinaBeeReady is a certification platform that, among other things, must ensure that any critical data cannot be altered manually outside of the platform user interface. Also, any data based on location has to be valid to avoid deceit. The main characteristics of this platform were the main focus of the company's research to determine if there was a way to implement such features for a software platform. All known security measures were being used during software development and release to secure the platform from external attacks. Because the data in this system had to be kept as unaltered as possible, it had to be protected from any insider attacks as well. The data, which is transferred from the mobile devices, also has to be trustworthy since the location of the device plays a crucial role in the certification process for specific roles. The research was conducted with these two targets in mind, and several options were examined. The produced results were promising for both of the researched topics and applied during the implementation of this software platform, in addition to all the existing practices for an overall security system.

Index Terms - Database fields encryption, location-based data, security, protection.

I. INTRODUCTION

Toorbee is a software house with a vertical market orientation in travel and tourism. From the time of any software inception, an integral part of the development process is eliminating any unauthorized access, avoiding a data breach, and quick recovery from a server infiltration incident, and there are some good reasons for this.

Although there is an important risk of sabotage from within, there are still some organizations that do not consider threats by insiders to be a severe risk that should be well controlled [1]. Insider security threats will always exist, and it is not a matter of trust [2]. Insider threats are a critical security problem. Insiders are trusted entities within an organization who have the authority to perform various operations, like employees, contractors, or business partners, and when they take advantage of their permissions and violate rules, it turns out to be a threat. In the context of a database, mechanisms that protect data from outside attacks cannot secure a database from authorized users.

Thus, the development of mechanisms that protect sensitive data from insiders has become a priority [3]. A Pokemon survey published in May 2016 reported that 55% of the 601 individual companies polled said their organization had a security incident or data breach from within [2]. The total average cost of insider-related incidents rose from 11.45 million dollars in 2019 to 15.38 million dollars in 2021, according to the 2020 and 2022 Cost of Insider Threats Global Reports published by the same organization [4]. In general, 34% of all breaches can be caused by insider threats [5].

It is a company practice to limit the authorized access to the production servers and data to only two people. However, since it is a certification system, it had to eliminate the possibility of even these trusted employees being able to alter any crucial data for any reason. In this case, having a user action log and various notifications reaching the system users during different points in the certification process allows for investigation of an incident after its occurrence, but this does not prevent any malicious data manipulation from happening in the first place. Usually, the main factors motivating insiders to perform data leakage activities are financial gain, lack of fairness and justice in the workplace, the psychology or characteristics of the insiders, new technologies, lack of education and awareness, and lack of management tools for understanding insider threats [1].

The ChinaBeeReady platform had to secure the data storage from alterations directly in the database by anyone that has access to it and verify the validity of the location-based data provided by the authorized users. Regarding the first axis of the research, it was crucial to find a way to disallow anyone with access to the database to be able to alter critical data like contracts, certifications, or auditing documents without invalidating them when updating them directly. The second axis of the research was also paramount for the company since it had to verify any information based on its location.

GPS signals are broadcast by Global Positioning System satellites to enable satellite navigation. Receivers on or near the Earth's surface can determine location, time, and velocity using this information [6]. Someone's geolocation can be determined both by the GPS receiver incorporated into smartphones and by cellular towers and Wi-Fi networks. All methods result in a location that consists of a latitude and longitude measurement. Depending on the method being used, locations can be measured accurately with a precision of about 5 m (GPS/Wi-Fi), although precision is much lower when cellular networks are used [7]. Location-based services need to determine where the user is to provide services and content relevant to their location. Therefore, positioning or location determination is a crucial technology for them. For indoor environments, other positioning methods and technologies are starting to appear, such as WiFi-based [8]. Location data is routinely used by websites and mobile apps to deliver better service to consumers. In cybersecurity, location data is used to help authenticate user identities. In addition, as with Uber, machine learning and other analytics can be used to detect suspicious user behavior [9].

An auditor should be present at the audited enterprise during the certification process. He should not be able to audit this enterprise from the comfort of his home. Even when he uses equipment provided by the company, there are numerous ways for someone to give a location different from the one he is currently at to the mobile application. However, the software should be able to countermeasure such actions and verify with certainty that the location-based data is valid. Is there any way to secure the data stored in a database from any modification from the inside of the company and any way to avoid deceit when location-based data is being used? The finding supports the statement, and the results may be used in the future to update the security of existing products regarding similar cases.

II. DEFINITION

Although a lot of effort is given to secure a company's infrastructure in the always-connected world, according to the 2022 Cost of Insider Threats Global Report from the Ponemon Institute, 67% of companies are experiencing between 21 and over 40 insider incidents per year, an increase of 60% from 2020. It takes an average of 85 days to contain one incident, costing over 480,000 dollars for each negligent incident and nearly 650,000 dollars for each malicious incident [10]. Most of the time, the weak point of an insider's attack is the company's trust in its employees, even when separation of access and well-defined roles exist. In a small-sized company, the most privileged user may be the owner, but the problem seems unsolvable when a company's size increases to even thousands of employees, with more than one having important roles like system administrators and auditors, managed security service providers [11].

Of all the potential attacks, the interested is in those having to do with the database. The traditional means of securing data are no longer enough [10]. Usually, negligent workers, departing employees, security evaders, malicious insiders, inside agents, and third-party partners are the origins of such an attack [12]. Such an attack can have many forms, like sabotage or even theft [13]. A recent event occurred in April 2022, when Block, a fintech company, confirmed that a former employee downloaded reports from the mobile payment app Cash App that contained information for 8.2 million users. The employee had been granted access to this data as a part of their job responsibilities, but the information was accessed without permission after their employment ended [10].

However, numerous other cases exist supporting the importance of being protected from the inside: In 2017, an employee stole more than 500,000 customer records and sold them on the dark web. An investigation revealed that the attacker had been accessing customer data since 2013. He saved three datasets, including credit card details, to his desktop and sent some information via email to his account. From 2008 to 2019, two employees of General Electric (GE) stole company trade secrets by downloading thousands of files from company servers. There was another insider attack at GE in 2017 when an engineer conspired with a Chinese businessman to steal trade secrets for their startup. In 2018, Cisco's cloud infrastructure was attacked, and the attacker deleted 456 virtual machines used for Cisco's WebEx Teams application. In 2018, a Tesla employee with the motive of revenge sabotaged the company and caused a data leak, which cost Tesla's share prices to fall by 5 percent, among others. In 2019, Microsoft employees made misconfiguration errors, and the database leaked on the Internet and remained publicly accessible for an entire month. Between early January and late February 2020, the attackers gained access to the personal data of 5.2 million Marriott guests. Tesla faced another insider threat in September 2020 when a foreign national attempted to "recruit" an employee to transmit malware onto Tesla's network to exfiltrate sensitive data [14].

Companies can suffer immediate losses as well as revenue losses, but the impacts of insider threat risks typically fall into one of the following categories: value, operations, or reputation [15]. Insider threat detection and prediction are important mitigation techniques. There are several proposed ways to solve this security issue in any company's size and organizational structure. One such way can be through online activities, which are the most widely used features in insider threat detection and prediction. Also, other ways are threat likelihood and graph algorithms [16]. Other practices, can be: a. Knowing and protecting your critical assets; b. Developing a formalized insider threat program; c. Deploying solutions for monitoring employees' actions and correlating information from multiple data sources; d. Documenting and consistently enforcing policies and controls; e. Incorporating malicious and unintentional insider threat awareness into all employees' periodic security training [17]. Generally, a combination of established methodologies in many IT disciplines can be combined to protect a database [18].

However, very little research has been performed on insider threats at a database level [19]. And this is where the research is targeted. One proposed way to detect such threats is through user profiling and machine-learning algorithms [20]. Most of the time, prevention is the key, and real-time user alerts can reduce the threat. A real-time alert can quickly remind users and keep them acting within corporate policy. If the user is misusing a database on purpose, it's still possible to prevent his actions using blocking [21].

Another proposed way to prevent such threats is to use a unified classification model to classify insider threat prevention approaches into two categories: a. Biometric and b. Asset-metric [22]. Even the evaluation of a user's personality trait of narcissism is used, which is deemed to be closely connected to the manifestation of malevolent insiders [23].

Such approaches to preventing an inside attack are validated because, by design, a database administrator can execute changes to the database with no limit to his/her privileges in terms of damage, including deleting rows, deleting tables, or revoking application privileges [24]. Another approach to detecting anomalous behavior of database users is using n-gram modeling to capture and analyze temporal patterns in sequences of SQL statements as proposed in [25]. Also, a company's employees are the human components for the detection and identification of an insider threat, and so by observing human behavior, their intentions can be disclosed even with non-verbal means [26].

Thus, it is crucial to have incorporated proper management of the user's privileges during his or her session, something that can reduce the impact of such a threat [27]. Some research places special attention on the definitions and taxonomies of the insider threat [28]. Another research presents an algorithm that demonstrates the authorized and unauthorized data items in which insiders can make changes, and to prevent such attacks, a modification graph is being used [29]. Nevertheless, all these methods of detection and prevention add complexity to a system, and it usually needs a lot of resources to maintain.

The main methods for securing a database from almost any attack are: a. cryptography, b. steganography, and c. access control. The cryptographic control secures the data by encrypting it [30]. Database encryption is the use of encryption techniques to convert a plain text database into an encrypted one, rendering its data unreadable to all but those who know the encryption key [31, 32]. This type of encryption provides data protection by only allowing authorized users to see the data [33]. Other research indicated that there is a substantial impact on the reading and writing performance of the database when it uses encryption. This impact is directly proportional to the level of security used [34]. However, encryption has a weak point and may be insufficiently effective against an insider threat. because the decryption key must also be stored somewhere in the system. Strict separation of duties is one of the most effective mechanisms for limiting the potential damage of such attacks [35]. Nevertheless, the advantage of encryption techniques over monitoring is clear since the latter is only as good as the audit trail, reporting, and vulnerability management systems being used [2]. Even in IaaS, where clients have the most access to the cloud infrastructure, cryptographic techniques can be used to safeguard the confidentiality and integrity of the database [35].

Regarding the location-based data, several things were considered to avoid deceit. The Global Positioning System (or GPS) offers information on position, velocity, and time. The Global Positioning System, on the other hand, is vulnerable to spoofing attacks [36]. GPS spoofing is an attack whose main goal is to override a GPS-enabled device's original location. There are a lot of ways for someone to do so, like the use of a radio transmitter that broadcasts fake GPS signals or for a smartphone owner to use third-party apps to fool other apps into thinking that the device isn't at its original location [37]. Such attacks are closely related to the research described. Thus, developing anti-spoofing algorithms requires a thorough understanding of GPS spoofing attack requirements, impacts, target type, and success rates. For various civilian and military applications, there are also several spoofing strategies [36].

Most anti-spoofing algorithms can detect the spoofing threat and maintain navigation capability and integrity [38]. The fake GPS or GPS spoofing apps are available on Android phones, and basically what they do is trick your phone into thinking you're in a different location [39]. Research showed that it is possible to take control of a Tesla's GPS with off-the-shelf tools in less than one minute [40]. Such attacks can be called "traffic poisoning" and create several dreadful scenarios [41]. All these make proving that a device is geographically present where it claims to be one of the most difficult problems. One geolocation technique aims to bind a client's identifier to a geographic location. However, the use of proxy servers, VPNs, anonymizers, or similar IP-hiding technologies makes this task difficult [42]. Geodetic location, as calculated from a location signature, adds a fourth and new dimension to user authentication and access control. It can be used to determine whether a person is attempting to log in from an approved location. Unlike most other types of authentication information, a user's location can serve as a common authenticator for all systems the user accesses [43].

To resolve such issues, the location of the user device can be analyzed against historical information. Also, location-aware fraud detection is used by the banks issuing credit cards where the location of a phone associated with the account is checked against the transaction's location [44]. The Global Positioning System is vulnerable to spoofing attacks, but there are some defenses against them as well [45]. The existing research goes both ways since there is one for the development of a GPS spoofing apparatus to attack a specific flying vehicle [46]. On the other hand, some algorithms exist that detect such attacks [47, 53]. Data-driven schemes can also become useful if enough training data is available and can operate on a per-satellite basis [48, 53]. Another approach can be to leverage time obtained over networks to which a mobile device can connect and detect discrepancies between the GNSS-provided time and the network time [49, 53].

Other research shows that the network-based exchange of GNSS data, such as the recently disclosed GNSS raw measurements in Android smart devices, may allow for the comparison or combination of such metrics to better identify spoofing and meaconing attacks [50, 53]. The European GNSS Agency (GSA) has funded the development of timing receivers for professional applications, intending to address vulnerabilities [51, 53]. Other solutions allow for the data to be digitally signed while the spreading code is protected by the insertion of cryptographically generated punctures [52, 53]. The measures against position tampering patterns may use MAC addresses in such a way that if at least one of the three MAC addresses used is far away, it is judged to be spoofed [54].

Although several solutions have been proposed to address the threat of GPS spoofing, few have been adopted in practice. This is because existing techniques either require significant modifications to the current GPS devices or require specialized hardware or are not robust against sophisticated attackers. An example of this is a proposed software-based method to detect spoofing attacks that works for off-the-shelf GPS chipsets [55]. Another study offers an application of spatial processing methods for GPS spoofing detection and mitigation [56]. There are several ways to mitigate against an attack, but there is no solution that fits all [57]. The most important obstacle to developing protection measures is that the key equipment is in space and will not be replaced any time soon. Security measures so far have been more experimental in nature and not for large-scale applications. Makers of smartphone chips

may someday be able to embed something like a GPS firewall directly into devices' receivers, but it will be a few more years before it happens [58].

It's hard to put a solid number on the impact of GPS spoofing on the economy, but the reliance on location services is a part of the world's critical infrastructure. One can see that the estimated overall economic impact of GPS on nine critical US industries was 1.4 trillion dollars, and a similar study found that the economic contribution of GNSS to the UK economy was £6.7 billion every year [59]. It is of no surprise that the Department of Homeland Security recommends that companies hide GPS antennas from public view because GPS spoofing works well when an attacker can get close to an antenna [60]. Smartphones nowadays host a wide variety of sensors and is no longer limited to the telephony domain. According to a study, this set of sensors on the device can provide a solution for spoofing and jamming detection [61]. Many conventional network-based applications provide specific content and services to users according to their locations. The accuracy of the location provided by the device's GPS module is certainly important but most service providers are unable to effectively authenticate GPS values provided by their users. An example for this can be the popular Pokemon GO game, where fake GPS values have a negative impact on the stability of the system. This issue is the so-called "Fake GPS" problem [62].

In Android, there is only one way to spoof your device's GPS without rooting, and that's to use Android's built-in Mock Location API located in Developer Options. On Android 6.0 and above, you select the specific app you would like to use. In older versions, it's just a simple check box that enables Mock Location mode for any app on your device [63]. Changing the location on your iPhone or Android device involves tricking your phone into telling apps that you're located somewhere you're not. In most cases, when you spoof your GPS location, every location-based app on your phone will be fooled. There are several software makers have built desktop programs that make this easy [64]. Nevertheless, GPS spoofing is not the only way to deceive an application and the use of a VPN service would work better [65]. A lot of recent flagship phones have very strong GPS connectivity, which can result in what is known as rubberbanding. In simple terms, you don't want your phone to keep alternating between your actual location and the spoofed location [66].

Plenty of GPS spoofing apps that can be used to set up mock locations are available on Google Play but they might not work depending on the app that you're trying to trick. According to their feedback, there is no iOS GPS spoofing app on the Apple App Store work. Instead, you'll need to download a desktop app and then connect your iPhone to it via a USB cable. Alternatively, iOS users can jailbreak their phones and install the Protect My Privacy app from Cydia, a mobile package manager for jailbroken iPhones [67, 68, 69, 70].

III. METHODS

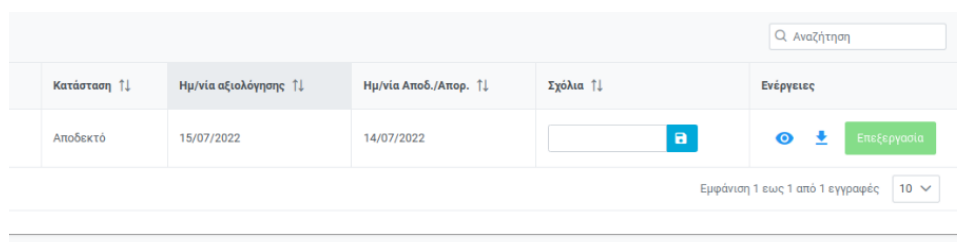
A. Database




According to [71] it was needed to use `pgp_sym_encrypt` for writing and `pgp_sym_decrypt` for reading. And this was what it has implemented according to Encryption For Specific Columns. The `pgcrypto` module allows certain fields to be stored encrypted. This was useful if only some of the data was sensitive. The client supplied the decryption key and the data was decrypted on the server and then sent to the client. The decrypted data and the decryption key were present on the server for a brief time while it was being decrypted and communicated between the client and server. This presented a brief moment where the data and keys can be intercepted by someone with complete access to the database server, such as the system administrator [72]. Several fields in the database table were encrypted based on [72] in addition to the other options that have been used. The first test took place in the database with the following steps:

1. The database's files

a) As a first step, it was verified that the files were valid using the platform's UI. The files could be viewed using a button in the user interface, as in Fig.1. When the user clicked on the eye-shaped button, he was able to view the file as in Fig.2. In the specific figure, the file shown is a signed application in PDF format

Figure 1. File row in the user interface.



Κατάσταση ↑↓	Ημ/νία αξιολόγησης ↑↓	Ημ/νία Αποδ./Απορ. ↑↓	Σχόλια ↑↓	Ενέργειες
Αποδεκτό	15/07/2022	14/07/2022	<input type="text"/>	  

Εμφάνιση 1 έως 1 από 1 εγγραφές 10 ▾

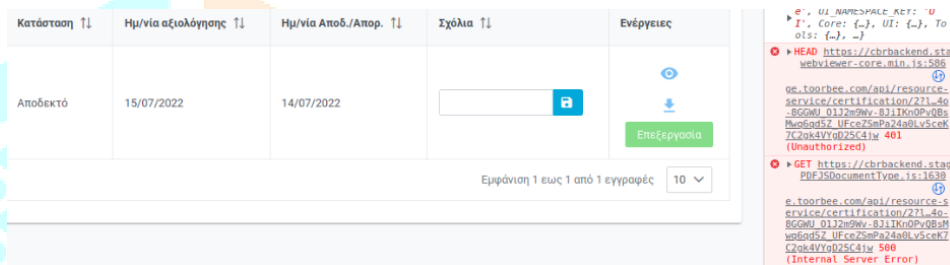
Figure 2. File preview in the user interface.



b) Next, a direct connection to the database was established and an update SQL command was executed. This was an actual action by someone willing to alter the data in a database by bypassing the user interface. The command was similar to this: UPDATE files SET data=<PDF_file_content> WHERE id=<file_id>. Executing a SELECT command, the data seemed valid, giving no indication that something was wrong with them. The command was similar to this: SELECT * FROM files WHERE id=<file_id>.

c) The files were invalidated after that update, though. Returning to the user interface, it was clear that the button was disabled and the file could not be viewed anymore. Also, the console log errors in the browser (Fig.3) confirmed that the file was irrecoverable and that the attack had just failed as a consequence of an update directly in the database.

Figure 3. File row in the user interface after the UPDATE without encryption.

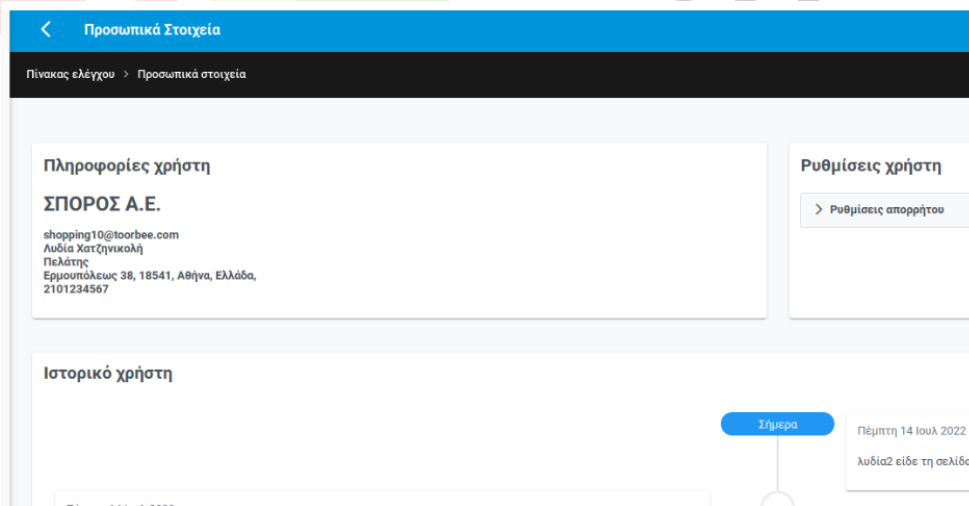


Using the above steps a to c, it was confirmed that any other files like the contracts, certifications, and other data were also invalidated after a manual update.

2. User's personal details in the database

a) Again, as a first step, it was verified that the user details were valid using the Platform UI. Before the test, the client information was visible in the related platform view in Fig.4.

Figure 4. User details in the user interface.



b) An update SQL was executed directly after connecting to the database. This update command had the purpose of trying to alter the client's information and was similar to: UPDATE users SET first_name=<Firstname>, last_name=<Lastname> WHERE users_id=<user_id>. Verifying the user data with a select query like this: SELECT * FROM users WHERE users_id=<user_id>, there was no indication something altered in the process.

c) Once more, it was verified that the user details were invalidated after the database update. In this case, the data was unusable and an error appeared in the web console, rendering the page empty as in Fig.5.

Figure 5. User details in the user interface after the UPDATE without encryption.

```

Show CORS errors in console
run main.js:136
true main.js:139
Failed to load resource: the server responded with a status of 500 (Internal Server Error)
Uncaught (in promise) Error: Request failed with status code 500
    at e.exports (createError.js:16:15)
    at e.exports (settle.js:17:12)
    at XMLHttpRequest.k (xhr.js:66:7)

```

B. Location-based data

Apps should only ask for the type of location permission that's critical to the user-facing feature and properly disclose this to users. The majority of use cases only require location when the user is engaging with the app. If your app requires background location, such as when implementing geofencing, make sure that it's critical to the core functionality of the app, offers clear benefits to the user, and is done in a way that's obvious to them. If background location access is essential for your app, keep in mind that Android preserves device battery life by setting background location limits on devices that run Android 8.0 (API level 26) and higher. On these versions of Android, if your app is running in the background, it can receive location updates only a few times each hour. Learn more about background location limits [73].

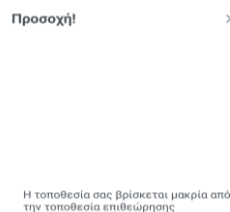
Since the app was developed in React, there is the react-native-mock-location-detector package in which [74] it is written: "If you are building a location-based app in RN, you have to validate if the user is using location spoofing apps or not." This library doesn't let the user use the app if any mock location apps are active on the device. However, since the app was developed with the Expo, the expo-location package was used, which allows reading geolocation information from the device. Your app can poll for the current location or subscribe to location update events [75]. But, even if the device isn't jailbroken, it is still possible to fake the GPS location with a nearby GPS transmitter. So let's assume (although it is highly unlikely) that your user has this. Technically, it is possible to detect if the GPS location is fake or not. However, your ability to do so is dependent on whether the person faked other information as well, such as nearby Wifi networks or their external IP (such as using a VPN) [76].

The second part of the tests used two mobile devices, one for the Android and one for the iOS operating system. The software on both the devices had many checks enabled for the available internet connectivity and location services.

1. a smartphone running Android 10 (QKQ1.19)

The normal setup was comprised of the data and location services being turned on. In this case, the first scenario runs with an auditor located at a specific position, e.g. 39.87391899335835, 25.064528320760136 auditing a company that requested a certification from being at the same position. The application worked as expected. Similarly, the second scenario was run with an auditor at another position from that of the company, e.g., at 35.335044557048995, 25.14158880387833. Since the setup on the device was expected, i.e. with both the data and location services turned on, the device reported no problem with that. However, the auditor being outside the audited company range was caught by the server, resulting in a relative message in the application like in Fig.6.

Figure 6. Android error message when the location is not matched.



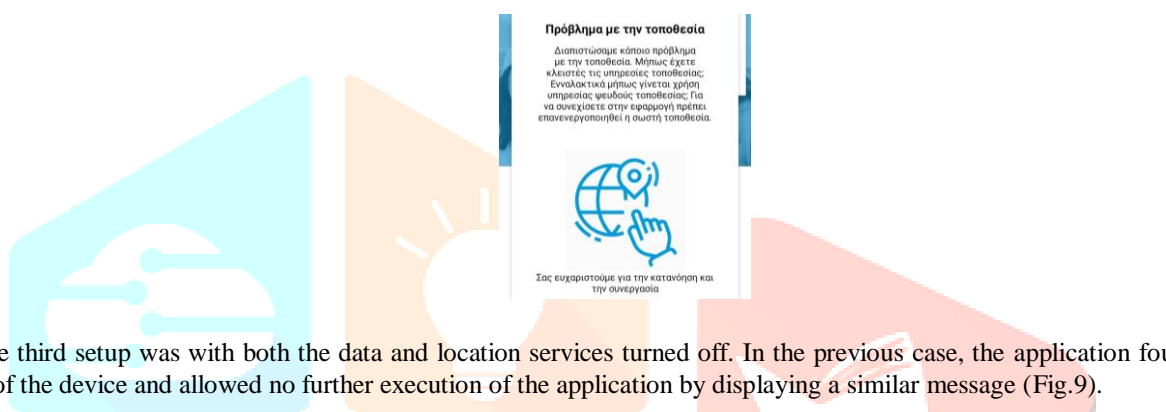
The second setup was for a device with data and location services turned on but with a Fake GPS application [77] running at the same time on the device. A sample screen of the app execution can be seen in Fig.7.

Figure 7. User interface of the Fake GPS app [77]



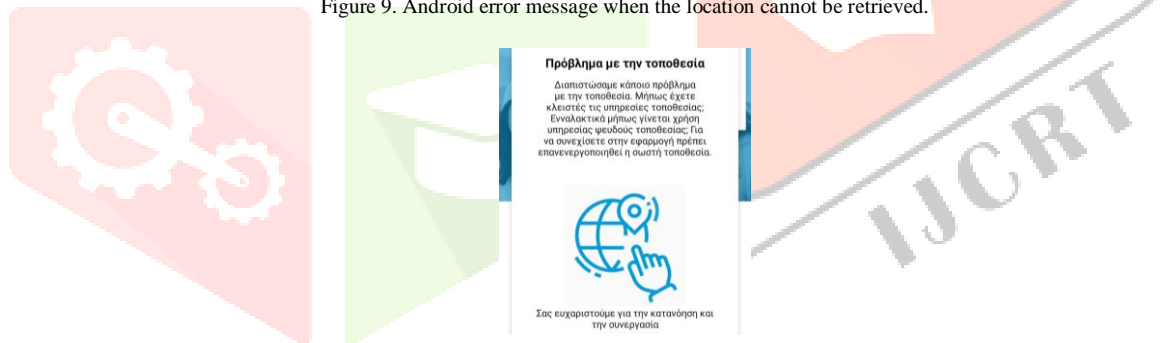
In this case, the application was not allowed to continue, and a popup message appeared, blocking the normal process for the auditor in Fig.8.

Figure 8. Android error message when the location cannot be retrieved.



The third setup was with both the data and location services turned off. In the previous case, the application found the invalid setup of the device and allowed no further execution of the application by displaying a similar message (Fig.9).

Figure 9. Android error message when the location cannot be retrieved.



The fourth setup was with both the data and location services turned off, but this time the WiFi was turned on. Again, the application was not allowed to continue with a popup message blocking the normal process of the auditor, as shown in Fig.10.

Figure 10. Android error message when the location cannot be retrieved.



The fifth setup was with no internet connection at all and with the location services turned off. As expected in this case also, the application was not allowed to continue with a popup blocking the normal process of the auditor displaying a message like in Fig.11.

Figure 11. Android error message when there is no connection.



The sixth setup was with no internet connection at all but with the location services turned on this time. As expected in this case also, the application was not allowed to continue with a popup blocking the normal process of the auditor, indicating a problem with the settings as in Fig.12.

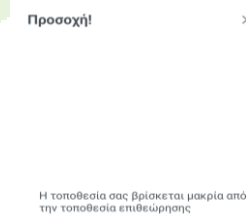
Figure 12. Android error message when there is no connection.



2. an iPhone or iPad running iOS 15.5-19f77

The normal setup was comprised of data and location services being turned on. In this case, the first scenario was run with an auditor at the same position as the company being audited, e.g. at a position with latitude and longitude of 39.87391899335835, 25.064528320760136. The application worked as expected, allowing the audit to be conducted by the auditor. The second scenario was run with an auditor at a different position from the company's, e.g. at 35.335044557048995, 25.14158880387833. Since the setup on the device was the expected one, the location difference was outside the allowed range, caught by the server by forcing a message to appear and blocking the process. See Fig.13.

Figure 13. iOS error message when the location is not matched.



The second setup was with both the data and location services turned off. This time the application was blocked with a popup and allowed no further execution of the application. The message is shown in Fig.14.

Figure 14. iOS error message when the location cannot be retrieved.



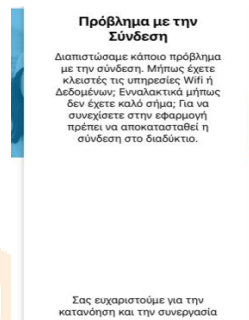
The third setup was with both the data and location services turned off but with the WiFi turned on. Again, the application was not allowed to continue with a popup blocking the normal process for the auditor (Fig.15).

Figure 15. Android error message when the location cannot be retrieved.



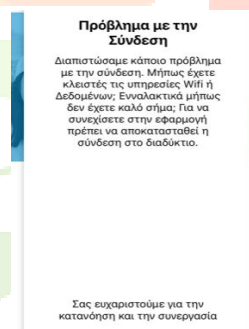
The fourth setup was with no internet connection at all and with the location services turned off. As expected in this case also, the application was not allowed to continue with a popup blocking the normal process for the auditor, showing the message in Fig.16.

Figure 16. iOS error message when there is no connection.



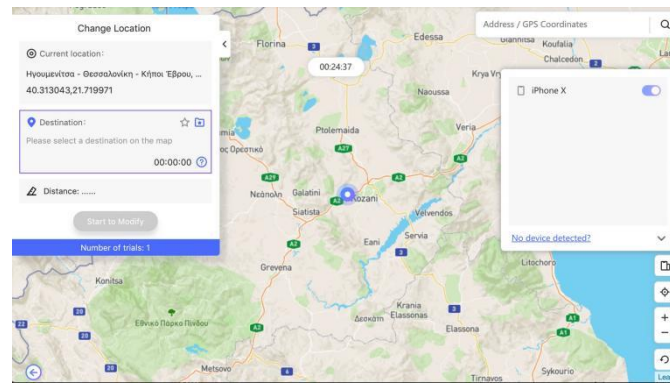
The fifth setup was with no internet connection at all but with the location services turned on. As expected in this case also, the application was not allowed to continue with a popup blocking the normal process of the auditor (Fig.17).

Figure 17. iOS error message when there is no connection.



Regarding the usage of an application that works like a Fake GPS application [77] in Android, this is not possible in iOS with the out-of-the-shelf setup by default. However, on a device that is not jailbroken, there is another way to try to overcome these imposed by the implementation restrictions, but it is something that the average user is not aware of. This test used the iAnygo application [78]. The user interface of this application can be seen in Fig. 18, and the existing internet and location services checks of the software on the device did not work. The application did not recognize this device setup and failed to block the auditor's process. That did not allow the server side to recognize the deception either.

Figure 18. User interface of the iAnyGo app [78]



IV. RESULTS

After implementing the features based on the contacted research, the tests that followed showed these results: Regarding the data in the database, the addition of the encryption in specific fields, i.e., fields that contain the information they had to be secured from any alteration by someone with access to the database, helped to secure them when a modification attempt was made by someone unaware of the encryption key. Technically, the attempt to modify the data in the database in such a manner invalidated the encrypted fields, and a restored backup was the only option to make them valid again. Thus, the attack failed, and the appropriate measures were taken to restore the validity of the data. This was true for all the tested data, like database entries, files stored in the database, and such.

Regarding the location-based data sent from a mobile device, either an Android or an iOS one, the security features implemented in both the front-end (client side) and the back-end (server side) were enough to prevent someone from deceiving the system in most cases. The check for turned-on data and location services, which provide and send the device's GPS tracked position, caught all but one case in Android and all but one case in iOS. The case that the test failed required specific paid software and a laptop connected to the device, though. In the back-end, when a location received that it was outside the expected location range, it was able to identify it successfully. However, it was not possible to know if it was a real or a fake location, assuming the device had sent verified position coordinates.

V. DISCUSSION

The research conducted had to answer the question of the existence of a secure software platform application with two main objectives: a. No one with access to the database should be able to manually change any important content; b. No one providing location-based data to the system should be able to deceive the system by providing a different than the device's current location. Based on the test results of the chosen implementation, taking into account the research, there is indeed a way to achieve both. Having encrypted fields in the database allows protection against attempting to alter data if the encryption key is unknown or cannot somehow be found in the same place as the database. This also protects against the case of an external attacker gaining access to the system from an internal attacker, e.g., an employee of the company. All this is in addition to all other measures that can be taken during the development of the platform software that secures the system at all levels, from the user interface to the database, at a low level as well as a high level.

Perhaps a further investigation is needed into whether solutions could be implemented that secure the system, and in some other cases, however, apart from the technical solution but more of the human factor. Also, mobile application development libraries allow control of specific features or functions that can be used to cheat the system. It was generally impossible to cheat the system using only the available means. But using tools or external devices, this became possible. But it is something that is not expected from someone who does not know or an employee of the company who is asked to perform a specific task. Surely, with newer research and additional testing, this can be eliminated and the system made completely safe. In a normal execution of the process by a user, it can be assumed that the checks have shown that the system cannot be cheated.

VI. CONCLUSION

Usually, during the development and release of software like the aforementioned platform, there is an effort to secure a system like the certification platform that has been implemented. The technical team of software developers, database administrators, and system administrators focuses on external threats and tries to secure the system by every possible means available. They sometimes overlook insider attacks, either when an intrusion occurs or by individuals within an organization. Certainly, the encryption of the fields in fields concerning important data works as a deterrent. But someone with access to both the encryption key and the database itself—for example, someone high up on the software development team who knows the code but can also connect to the database—could bypass this security.

But as long as the roles remain distinct and the access is limited and isolated, as the research showed, if the software developer does not have access to the database or the database administrator access to the code, any attempt to change the data causes its destruction, and a backup needs to be restored to reverse the destruction of this one. On the other hand, the result of this data security

function can also be a malicious action. Someone may not want to gain a benefit by changing a piece of information but try to prevent it from functioning or causing damage. In this case, this security feature could cause massive data loss, which could be passed on to previous backups, causing incalculable losses or business delays. Perhaps future research into this feature or a better internal organization would find a solution to this new problem.

Regarding the attempt by someone to deceive the system into stating that he is in a position other than the one he should be to perform a task, such as by an auditor who must be physically present at the company he audits, surely research has shown that it can be limited to a considerable extent. Device usage without conversions is prevented using a combination of controls at either the device level (front-end) or the application level (back-end). Surely a more knowledgeable user might be aware of the existence of apps that could fool the authentication app. These applications are useful for a software developer who develops applications using location, but they can also be used in a case of deception like the one examined. Especially on iOS, this requires specialized software and an external device such as a laptop, but it is not impossible as it emerged from the research.

Certainly, in the future, cases outside of the present research, such as the existence of rooted or jailbroken devices and their behavior in terms of location-based data security, should be investigated and tested. For a business that can provide its staff with devices to perform specific tasks, such as the case investigated, the existing implementation based on this investigation can be considered to fully ensure the correctness of location-based data. If employees' devices will be used, a newer search is required.

VII. ACKNOWLEDGMENT

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code:T2EDK-02489).

REFERENCES

- [1] Al-Harrasi, A., Shaikh, A. K., & Al-Badi, A. (Accepted/In press). Towards protecting organisations' data by preventing data theft by malicious insiders. *International Journal of Organizational Analysis*. <https://doi.org/10.1108/IJOA-01-2021-2598>.
- [2] Mike O. Villegas, K3DES LLC, E-Handbook: Crafting an insider threat program: Why and how, <https://www.techtarget.com/searchsecurity/tip/Insider-security-threats-What-CISOs-can-do-to-mitigate-them>.
- [3] Ragavan, H. (2012). Insider Threat Mitigation Models Based on Thresholds and Dependencies. Graduate Theses and Dissertations Retrieved from <https://scholarworks.uark.edu/etd/313>.
- [4] EKCRAN, 5 Real-Life Data Breaches Caused by Insider Threats, April 13, 2022, 5 Real-Life Data Breaches Caused by Insider Threats | Ekcran System.
- [5] Chris Smith, How can you prevent insider threats when none of your insiders are actually "inside"? <https://delinea.com/blog/insider-threats-in-cyber-security>.
- [6] GPS signals, Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/GPS_signals.
- [7] Elevelt, A. ; Bernasco, W. ; Lugtig, P. ; Ruiters, S. ; Toepoel, V. / Where You at? Using GPS Locations in an Electronic Time Use Diary Study to Derive Functional Locations. In: *Social Science Computer Review*. 2019 ; Vol. 39, No. 4. pp. 509-526.
- [8] Haosheng Huang, Georg Gartner, Jukka M. Krisp, Martin Raubal & Nico Van de Weghe (2018) Location based services: ongoing evolution and research agenda, *Journal of Location Based Services*, 12:2, 63-93, DOI: 10.1080/17489725.2018.1508763.
- [9] Maria Korolov, What is GPS spoofing? And how you can defend against it, Contributing writer, CSO, MAY 7, 2019, <https://www.csoonline.com/article/3393462/what-is-gps-spoofing-and-how-you-can-defend-against-it.html>.
- [10] Eldad Chai, Modern Data Security: Protecting Your Sensitive Data from Insider Threats, June 3, 2022, <https://www.dataversity.net/modern-data-security-protecting-your-sensitive-data-from-insider-threats/>.
- [11] N. Pitropakis, Detecting malicious insider threat in cloud computing environments, 2015.
- [12] Insider Threat: Types, Examples, Detection, and Prevention, Mike Puterbaugh, <https://pathlock.com/insider-threat-types-examples-detection-and-prevention/>.
- [13] [SPIRION, Understanding malicious insider threat examples to avoid an insider attack, November 10, 2021, Understanding malicious insider threat examples to avoid an insider attack - Spirion.
- [14] Reciprocity, June 23, 2022, Insider Threat Examples: 7 Real-Life Cases to Guide Your Cybersecurity Program.
- [15] Nate Olson-Daniel, Vice President and Technical Fellow, Why Insider Threats Are the Biggest Danger to Your Data, November 14, 2019.
- [16] Gheyas, I.A., Abdallah, A.E. Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. *Big Data Anal* 1, 6 (2016). <https://doi.org/10.1186/s41044-016-0006-0>.
- [17] Randy Trzeciak , 5 Best Practices to Prevent Insider Threat , November 6, 2017, <https://insights.sei.cmu.edu/blog/5-best-practices-to-prevent-insider-threat/>.
- [18] R. M. Barrios, "A Multi-Leveled Approach to Intrusion Detection and the Insider Threat," *Journal of Information Security*, Vol. 4 No. 1, 2013, pp. 54-65. doi: 10.4236/jis.2013.41007.
- [19] Q. Yaseen, Y. Jararweh, B. Panda and Q. Althebyan. An Insider Threat Aware Access Control for Cloud Relational Databases. *Journal of Cluster Computing*, Springer, 2017.
- [20] Mathew, S., Petropoulos, M., Ngo, H.Q., Upadhyaya, S. (2010). A Data-Centric Approach to Insider Attack Detection in Database Systems. In: Jha, S., Sommer, R., Kreibich, C. (eds) *Recent Advances in Intrusion Detection*. RAID 2010. Lecture Notes in Computer Science, vol 6307. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15512-3_20.
- [21] Proofpoint Staff, March 28, 2019 , How to Protect Databases from Insider Threats, <https://www.proofpoint.com/us/blog/insider-threat-management/how-protect-databases-insider-threats>.

- [22] Alsowail RA, Al-Shehari T. Techniques and countermeasures for preventing insider threats. *PeerJ Comput Sci.* 2022 Apr 1;8:e938. doi: 10.7717/peerj-cs.938. PMID: 35494800; PMCID: PMC9044369.
- [23] Kandias, M. (2017). Insider threat prediction: Psychosocial characteristics extraction and security data science techniques on OSN OSINT. Department of Informatics Athens University of Economics & Business Athens, Greece.
- [24] Walid Rjaibi, Safeguarding Databases Against Insider Threats, February 24, 2016, <https://securityintelligence.com/safeguarding-databases-against-insider-threats/>.
- [25] Makovoz, David & Tavernier, Jean. (2009). Database Insider Threat Detection.
- [26] Sneha Vilas Kotawadekar, Database Security and Insider Threats, 2022 IJCRT | Volume 10, Issue 5 May 2022.
- [27] B. Mahesh Babu and Mary Saira Bhanu, Prevention of Insider Attacks by Integrating Behavior Analysis with Risk based Access Control Model to Protect Cloud, *Procedia Computer Science* 54 (2015) 157 – 166.
- [28] Ivan Homoliak, Flavio Toffalini, Juan Guarnizo, Yuval Elovici, and Martín Ochoa. 2019. Insight Into Insiders and IT: A Survey of Insider Threat Taxonomies, Analysis, Modeling, and Countermeasures. *ACM Comput. Surv.* 52, 2, Article 30 (March 2020), 40 pages. <https://doi.org/10.1145/3303771>.
- [29] Yaseen and B. Panda, "Malicious Modification Attacks by Insiders in Relational Databases: Prediction and Prevention," 2010 IEEE Second International Conference on Social Computing, 2010, pp. 849-856, doi: 10.1109/SocialCom.2010.128.
- [30] Anil Dixit, Dr. Suchithra R, "Advanced Database Security and Encryption", *ICRET – 2016 (Volume 4 – Issue 21)*, *IJERT*, 24-04-2018 <https://www.ijert.org/advanced-database-security-and-encryption>.
- [31] Luc Bouganim, Yanli Guo. Database encryption. S. Jajodia and H. van Tilborg. *Encyclopedia of Cryptography and Security*, Springer, pp.1-9, 2009, 978-1-4419-5905-8. [ff10.1007/978-1-4419-5906-5_677ff](https://doi.org/10.1007/978-1-4419-5906-5_677ff). [ffhal-00623915f](https://doi.org/10.1007/978-1-4419-5906-5_677).
- [32] Bouganim, Luc & Guo, Yanli. (2010). Database Encryption. [10.1007/978-1-4419-5906-5_677](https://doi.org/10.1007/978-1-4419-5906-5_677).
- [33] Erez Shmueli, Ronen Vaisenberg, Yuval Elovici, Beer Sheva, Database Encryption – An Overview of Contemporary Challenges and Design Considerations, *SIGMOD Record*, September 2009 (Vol. 38, No. 3).
- [34] André Gomes, Carla Santos, Cristina Wanzeller and Pedro Martins (2021), "Database Encryption for Balance Between Performance and Security", *Journal of Information Assurance & Cyber security*, Vol. 2021 (2021), Article ID 614511, DOI: 10.5171/2021.614511.
- [35] Kandias, Miltiadis, Nikos Virvilis, and Dimitris Gritzalis. "The insider threat in cloud computing." In *International Workshop on Critical Information Infrastructures Security*, pp. 93-103. Springer Berlin Heidelberg, 2011.
- [36] Bethi, Pardhasaradhi & Pathipati, Srihari & Pulikala, Aparna. (2020). Stealthy GPS Spoofing: Spoofer Systems, Spoofing Techniques and Strategies. 1-7. [10.1109/INDICON49873.2020.9342317](https://doi.org/10.1109/INDICON49873.2020.9342317).
- [37] Jomilë Nakutavičiūtė, Everything you need to know about GPS spoofing, NordVPN, Oct 24, 2021.
- [38] Ali Jafarnia-Jahromi, Ali Broumandan, John Nielsen, and Gérard Lachapelle, Position Location and Navigation (PLAN) Group, Schulich School of Engineering, University of Calgary, 2500 University Drive, NW, Calgary, AB, Canada T2N 1N4, 29 May 2012.
- [39] Harry Campbell, How Drivers Can Get Deactivated For Using 'Fake GPS' Apps, February 14, 2017, <https://maximumridesharingprofits.com/drivers-can-get-deactivated-using-fake-gps-apps/>.
- [40] Regulus Cyber, Tesla Model S and Model 3 Prove Vulnerable to GPS Spoofing Attacks as Autopilot Navigation Steers Car off Road, Research from Regulus Cyber Shows, 19 Jun, 2019, <https://www.prnewswire.com/il/news-releases/tesla-model-s-and-model-3-prove-vulnerable-to-gps-spoofing-attacks-as-autopilot-navigation-steers-car-off-road-research-from-regulus-cyber-shows-300871146.html>.
- [41] Raponi, Simone ; Sciancalepore, Savio ; Oligeri, Gabriele et al. / Road Traffic Poisoning of Navigation Apps : Threats and Countermeasures. In: *IEEE Security and Privacy*. 2022 ; Vol. 20, No. 3. pp. 71-79.
- [42] AbdelRahman Mohamed Abdou, Internet Location Verification: Challenges and Solutions, Carleton University, Ottawa, Ontario, Canada, 2015, arXiv:1802.05169v1 [cs.CR] 14 Feb 2018.
- [43] Denning, D.E., & MacDoran, P.F. (1996). Location-based authentication: Grounding cyberspace for better security. *Computer Fraud & Security*, 1996, 12-16.
- [44] Sheldon Meredith, Mark Austin, John Pastore, Fraud analysis for a location aware transaction, patent US9053513B2, 2010-02-25.
- [45] Mark L. Psiaki/Todd E. Humphreys, Protecting GPS From Spoofers Is Critical to the Future of Navigation , 29 Jul 2016, <https://spectrum.ieee.org/gps-spoofing>.
- [46] Horton, E., Ranganathan, P. Development of a GPS spoofing apparatus to attack a DJI Matrice 100 Quadcopter. *J. Glob. Position. Syst.* 16, 9 (2018). <https://doi.org/10.1186/s41445-018-0018-3>.
- [47] Rothmaier, Fabian; "Optimal Sequential Spoof Detection Based on Direction of Arrival Measurements." <https://doi.org/10.33012/2020.17538>.
- [48] Borhani-Darian, Parisa; Li, Haoqing; Wu, Peng; Closas, Pau; "Deep Neural Network Approach to Detect GNSS Spoofing Attacks." <https://doi.org/10.33012/2020.17537>.
- [49] Spanghero, Marco; Zhang, Kewei; Papadimitratos, Panagiotis; "Authenticated Time for Detecting GNSS Attacks."
- [50] Rustamov, Akmal; Gogoi, Neil; Minetto, Alex; Dovis, Fabio; "GNSS Anti-Spoofing Defense Based on Cooperative Positioning."
- [51] Catalano, Valeria; Prata, Ricardo; Carvalho, Filipe; Nunes, Rui; Marradi, Livio; Franzoni, Gianluca; Puccitelli, Marco; Campana, Roberto; Gioia, Ciro; "Galileo OSNMA Preliminary Implementation in the GIANO GNSS Receiver." <https://doi.org/10.33012/2020.17714>.

- [52] Gamba, Micaela Troglia; Nicola, Mario; Motella, Beatrice; "GPS Chimera: A Software Profiling Analysis." <https://doi.org/10.33012/2020.17717>.
- [53] Tracy Cozzens, Research Roundup: Combatting jamming and spoofing, September 23, 2021, <https://www.gpsworld.com/research-roundup-combatting-jamming-and-spoofing/>.
- [54] Nobuo Suzuki, Taiga Harada, and Takuya Fujihata. 2021. Realization and countermeasures for current location spoofing attacks. *Procedia Comput. Sci.* 192, C (2021), 2115–2121. <https://doi.org/10.1016/j.procs.2021.08.219>.
- [55] Shinan Liu and Xiang Cheng and Hanchao Yang and Yuanchao Shu and Xiaoran Weng and Ping Guo and Kexiong (Curtis) Zeng and Gang Wang and Yaling Yang}, Stars Can Tell: A Robust Method to Defend against {GPS} Spoofing Attacks using Off-the-shelf Chipset}, 30th USENIX Security Symposium (USENIX Security 21), 2021, 978-1-939133-24-3 p. 3935--3952.
- [56] Magiera, Jaroslaw & Katulski, R.. (2015). Detection and Mitigation of GPS Spoofing Based on Antenna Array Processing. *Journal of Applied Research and Technology.* 11. 45-57. 10.1016/S1665-6423(15)30004-3.
- [57] Guy Buesnel, DEFCON25: GPS time spoofing now "simple party trick" - researcher, September 14, 2017, <https://www.spirent.com/blogs/defcon-25>.
- [58] Serge Malenkovich, Is it possible to guard against GPS attacks?, May 3, 2019, <https://www.kaspersky.com/blog/gps-spoofing-protection/26837/>.
- [59] Ben Ball, Head of Marketing, Why GPS spoofing is a problem (and what to do about it), Nov 12, 2020, <https://nextnav.com/gps-spoofing/>.
- [60] McAfee, What is GPS spoofing?, AUG 25, 2020, <https://www.mcafee.com/blogs/internet-security/what-is-gps-spoofing/>.
- [61] Miralles, Damian & Levigne, Nathan & Akos, Dennis & Blanch, Juan & Lo, Sherman. (2018). Android Raw GNSS Measurements as the New Anti-Spoofing and Anti-Jamming Solution. 334-344. 10.33012/2018.15883.
- [62] Yu-Heng Chang, Chih-Lin Hu, Yan-Ling Hwang, Chih-Wen Ou, Fu-Hau Hsu, Fake GPS Defender: A Server-side Solution to Detect Fake GPS, July 22, IARIA, 2018, ACCSE 2018, The Third International Conference on Advances in Computation, Communications and Services, 2018, 978-1-61208-658-3, p.36 to 41.
- [63] App Ninjas, What Are GPS Spoofing Apps Actually Doing?, Jan 20, 2017.
- [64] Tim Fisher, How to Fake a GPS Location on Your Phone, June 23, 2022.
- [65] Robert Hayes, How To Spoof your GPS Location on an Android, March 15, 2021, <https://www.alphr.com/fake-spoof-gps-location-android/>.
- [66] Sumukh Rao, How to Fake your Location on Android using GPS Spoofing, August 4, 2021, <https://www.xda-developers.com/how-to-fake-location-android-gps-spoofing/>.
- [67] Paul Bischoff, How to use a VPN to fake your GPS location, January 28, 2022, <https://www.comparitech.com/blog/vpn-privacy/vpn-fake-gps-location/>.
- [68] Bipin Dhungana, Best Fake GPS Location Apps for Android, August 27, 2021, <https://mobilespy.io/blogs/best-fake-gps-location-apps-for-android/>.
- [69] Ste Knight, The 7 Best Free Android Apps to Fake Your GPS Location, Jan 24, 2022, <https://www.makeuseof.com/best-android-location-spoofing-apps/>.
- [70] My GPS Tools, Top-5 best fake GPS apps for Android phone, March 5, 2022, <https://mygpstools.com/fake-gps-location-apps>.
- [71] [FIXED] How to encrypt a column in Postgres using Hibernate @ColumnTransformer, October 06, 2021, <https://www.javafixing.com/2021/10/fix-how-to-encrypt-column-in-postgres.html>.
- [72] PostgreSQL 14, Documentation, Chapter 19. Server Setup and Operation, 19.8. Encryption Options
- [73] Access location in the background, Android Developers, Docs, Guides, <https://developer.android.com/training/location/background>.
- [74] React Native Mock Location Detector / Location Spoof Apps Detector, <https://www.npmjs.com/package/react-native-mock-location-detector>.
- [75] Expo, Location, <https://docs.expo.dev/versions/latest/sdk/location/>.
- [76] Detect/Stop Spoofing of Location possible in iOS sdk?, <https://stackoverflow.com/questions/20966733/detect-stop-spoofing-of-location-possible-in-ios-sdk>.
- [77] Fake GPS location, Google Play, <https://play.google.com/store/apps/details?id=com.lexa.fakegps&hl=en&gl=US>.
- [78] Tenorshare iAnyGo, Freely Change GPS Location without Jailbreak!, <https://www.tenorshare.net/products/ianygo-change-gps-location-iphone.html>.