Mobile Security: Threats, Defences and Best Practices

By Yasin Zar-Khan

***Abstract*— In current times, mobiles have become so ubiquitous with work and personal lives that the value of data held on these devices can often surpass their monetary value. Because of this, these devices are increasingly being targeted by adversaries as a source of sensitive and confidential data. This paper will evaluate iOS and Android defences, current threats, and best practices and explore various attack vectors and malware that pose significant risk to personal users and businesses.**

Keywords—Device Security Posture, SEAndroid, Secure Boot Chain, Mobile Threat Landscape, Privilege Escalation, Mobile Device Management, App Permissions.

I. INTRODUCTION

Today, mobile devices are heavily relied upon and utilised for a wide manner of daily tasks. These devices commonly hold our personal and financial data and it's commonplace for businesses to issue smartphones to employees, which can contain confidential business data and sensitive documents. Projections show this reliance has no plans of slowing down, with previous trends forecasting that by 2027, there will be 7.7 billion smartphone users worldwide [1]. The prevalence and widespread use of these devices have certainly enticed threat actors to target these devices as a means of exploitation. Therefore, businesses and individuals need to practice proper security hygiene.

II. OPERATING SYSTEM SECURITY

A. Android Security

1) SEAndroid

Security Enhanced Android is a framework used to implement and enforce Mandatory Access Control (MAC) across all processes to reduce the potential attack surface for threat actors [3]. The framework operates with a deny-by-default principle, therefore, the model considers everything forbidden unless explicitly stated as allowed [4, p. 480]. This models enforcing mode is enabled, so any unauthorised actions are automatically blocked [2]. Implementing MAC at a kernel level prevents an adversary from attempting to gain initial access or perform privilege escalation on the device.

2) Filesystem

By default, Android will store all user and shared app data in '/storage/emulated/0'. This directory has read and write permissions but no execute access. Because of this, files cannot be assigned execute permissions, preventing users from running executables and scripts they may have downloaded from untrustworthy sources. In addition to this, Android Package Kit (APK) files used to install apps are stored in '/data/app' and will only have access to their directory unless explicitly allowed [4, p. 478]. This is a form of sandboxing and helps prevent defensive evasion techniques like process injection and masquerading.

3) Android Security Services

There are several security services Android have implemented to further improve their security posture.

- Google Play Protect is a solution on Android that scans and monitors apps for any malicious activity from apps downloaded from the Play Store as well as other sources.
- Play Integrity API replaces the previously now depreciated API SafetyNet and offers much of the same feature set, including device and app integrity checks [5].

4) Reflection on Android security

While these security controls can greatly increase the security posture of a device, they are also configured on a per-OEM basis leading to inconsistent policies that vary depending on the manufacturer. As researched by Reshetova, E. et al. [6] certain manufacturers including LG had overly permissive policies in place, which increases the landscape available to an attacker.

B. iOS security

1) Secure Boot Chain

The Secure Boot Chain is executed when the device initiates the boot process, starting with the lowest layer of code first, the Boot ROM. Due to this being read-only memory, it serves as the root of trust that performs integrity checks on the next stage in the chain. If these checks pass, the sequence progresses to the next layer, and checks are rerun. This process establishes a chain of trust that ensures only authorised and Apple-signed software is being executed upon startup. Once the OS kernel has been initialised, Kernel Integrity Protection (KIP) is enabled which prevents any write operations from being carried out on kernel and driver level code [4].

2) Encryption

During manufacturing, Apple embeds each iPhone with a Unique ID (UID) within the Secure Enclave coprocessor and a shared Group ID (GID) between primary processors of the same type. The Secure Enclave carries out cryptographic operations, such as creating unique encryption keys to encrypt sensitive data [4, p. 436]. Utilising a UID for encryption ensures that data can only be decrypted on the original device, preventing threat actors from decrypting it on other devices. On the other hand, the GID on the main processor allows for verifying the integrity of firmware and updates [7].

3) IOS Security Services

Apple have other features on their iOS devices that ensures data is protected.

- The iOS keychain, which holds app passwords and other user data is encrypted with secure AES-256-GCM encryption [4, 37].
- Apple have introduced a remote wipe feature for their devices that securely deletes encryption keys, rendering all data inaccessible. This ensures that no data can be exfiltrated from a stolen device [7].

4) Evaluation of iOS Security

Despite the inherent secure nature of using a boot chain that relies on a ROM as its root of trust, exploits found in the Boot ROM can provide permanent vulnerabilities for Apple devices. One such example covered by Wu, J. et al. [8] was the Checkm8 vulnerability which allowed read and write access to the ROM and created permanent exploits on all devices from the iPhone 4S to the iPhone X.

C. Evaluating Application Approval Process for iOS and Android

Process	iOS	Android
Submission	Undergo review process to ensure app meets guidelines.	Open distribution model allows submission without review.
Code Signing	Certificate issued valid by Apple.	Developers can use self-signed certificate.
Review duration	Manual review can take up to a week for approval.	Algorithm and manual check can take 1-7 hours.
Fee	Requires a $99/year subscription to create an Apple Developer account.	One-time fee of $25, which gives access to unlimited app submissions.
App Store Exclusivity	Apps distributed through iOS App Store only.	Supports sideloading which allows apps to be installed through third-party stores

Table 1: Comparing approval process for applications [9, p. 447], [10], [11]

The iOS approval process appears more thorough when compared to Android, which supports sideloading and only requires a one-time fee to be eligible for unlimited app submissions. This comparison does suggest that iOS offers better screening for potentially malicious applications, albeit with a higher bar of entry for developers. However, as we will further examine in Common Malware Threats, they are still vulnerable to malware and exploit attacks that circumvent security controls.

With regard to Android's lower entry fee and open distribution policies, their approval model benefits developers and users with accessibility and choice. However, it seems this inadvertently makes their OS more attractive to threat actors deploying malware.

III. Mobile Threat Landscape

A. Common Malware Threats

1) Spyware

Spyware is malicious software that can infiltrate and reside on mobile devices. It can collect user information such as location, photos, messages and access the victim's camera and microphone. An attacker can then use this data for various purposes like blackmail, extortion or committing identify theft. Spyware is one of the most common in the mobile threat landscape [12].

Exodus is an example of spyware that targeted both iOS and Android devices by masquerading as legitimate cellular support apps. Adversaries were able to cleverly bypass the App Store's security checks and distribute the app on iOS devices by exploiting Apple's Developer Enterprise Program, which allowed users to distribute apps internally for user testing. Once installed the spyware gathered sensitive user data and exfiltrated it to a command-and-control server [13, 38].

2) Ransomware

Ransomware is notorious for encrypting systems and files effectively preventing users or in many cases entire organisations from accessing their data. Systems are often locked behind a paywall that displays a message demanding a ransom in exchange for access back to encrypted data [14]. Even in situations where the victim does pay the ransom, there is no guarantee the threat actor will recover data or provide the decryption key [15].

Observed in 2014, Scarepackage was a ransomware targeting Android devices that masqueraded on the Play Store as free antivirus apps to coerce users into downloading the app. The victims would receive a message on their device informing them the FBI had locked them out of their phone and the only way to regain access was to pay them 700 dollars [16]. Users' devices remained locked even after restarting their phones and this effected 900,000 Android devices in the month of August alone [14].

3) Cryptojacking

Cryptojacking utilises the computational power of mobile devices to mine cryptocurrencies [4, p. 456]. This activity is usually unbeknownst to the user and is delivered as a trojan in simplistic applications downloaded from the Google Play Store. While cryptojacking on a mobile device may not be efficient, mining on a high number of devices can be very profitable for the attacker. For the user, the process can cause their device to become slow, drain the battery and overheat, which can lead to irreversible damage [17].

BadLepricon malware was also observed in 2014 as trojan wallpaper apps on app stores. Once downloaded, the malware periodically checked whether the device's display was off, connected to a network and above 50% battery. If these conditions were met the malware would begin mining bitcoin [18].

B. Exploitation of Device Features and Privileges

1) Rooting

Android users may decide to root their devices to elevate their privileges [4, p. 484]. Doing this gives users more granularity over the system and allows them to tweak hidden settings, remove bloatware and even overclock a device's CPU/GPU to increase performance. However, this comes at the cost of breaking the SEAndroid model; this means MAC will no longer operate on a deny-by-default principle and rooted apps can perform previously unauthorised actions. Additionally, installed apps will no longer be sandboxed and can access other app's data and sensitive files [9, 39].

Malware exists that can exploit vulnerabilities in an operating system to obtain and self-root the device itself. An example is the GODLESS malware which masqueraded as legitimate apps on third-party markets and the Play Store. Once installed, this malware would iterate through multiple known exploits until it found one the device was vulnerable against. Once a successful exploit had been performed, it would obtain root privileges and be capable of bypassing MAC and sandboxing [19].

2) Jailbreaking

While similar in concept to rooting, jailbreaking iOS devices does not permit the same amount of freedom and control. The main advantage of jailbreaking is that it allows a user to download applications and software outside of the app store, this is referred to as sideloading and is supported on stock Android operating systems [4, p. 484]. Much like rooting, jailbreaking will bypass built in protections like Secure Boot and sandboxing. Being able to sideload unverified apps and not having security controls enabled greatly increases the security risk and susceptibility to attacks [9].

KeyRaider is a malware family that targets jailbroken devices and is distributed through third-party stores like Cydia. The malware is capable of stealing Apple account data such as usernames, passwords and purchasing information [21]. It is reported to have been successfully deployed and stolen data from over 225,000 accounts [20].

C. Physical and Proximity-Based Attacks on Mobile Devices

1) Relay Attacks

In relay attacks, an adversary may exploit proximity-based features like NFC to communicate or relay messages/data from the victim's device to an attacker-controlled device. A practical example of this is conducting a relay attack on contactless payment wallets like Apple Pay or Google Pay, which use NFC during payments. Here, a victim approaches a payment terminal expecting to pay for certain goods and places their phone near the card-reader; at this point, a relay device placed close to the payment terminal will intercept the NFC signal and relay the payment token to an attacker-controlled device [22]. This token can then be used by an attacker to make a fraudulent purchase at another payment station [23].

2) Juice Jacking

An adversary may compromise a charging socket or cable that users may unknowingly connect their devices to in order to charge. Once connected, these compromised charging setups use the USB transfer feature on iOS and Android to gain access and steal data from a device. Additionally, this attack can be used to record user inputs, effectively capturing passwords and can deploy malicious payloads like spyware, which will remain on the device once disconnected [25]. The common attack vector targets charging stations and docks in airports, train stations and other public areas as users are more inclined to use these available services while travelling or in places where they lack personal charging equipment [24].

D. Mobile and Voice-Based Social Engineering Attacks

1) Smishing

In smishing, a threat actor sends a short message service (SMS) or text message to potential victims, encouraging them to visit a fake website, download an app or provide personal Identifiable Information (PII) and potentially Payment Card Information (PCI) [26]. Attackers will often spoof and impersonate a credible company such as PayPal, Amazon and Outlook to deceive victims into logging in to a credential harvester login page that collects and sends their details to an attacker for exploitation. A common factor of these attacks is to instil a sense of fear and urgency so the victim is more likely to comply, this is done often by claiming a victim's account is due to expire or their account has been locked and the user must verify their details to regain access [27, p. 129].

2) Vishing

Vishing is another form of phishing where the attack vector is a phone call that uses social engineering tactics to retrieve information or persuade a victim to carry out actions that compromise a device [28].

In 2017 DEF CON held a Social Engineering CTF that had contestants conduct vishing attacks on target companies. The winning call from this contest posed as an IT security firm calling a retail store requesting some 'quick help' from the victim regarding some IT issues the store may be experiencing [29].

Under the guise of troubleshooting potential issues, the attacker managed to get the target to provide the following:

- OS and browser versions, which indicate whether they are vulnerable to any exploits.
- The store's payroll schedule, which can be used to conduct targeted phishing attacks, known as spear-phishing.
- Was able to guide the target to a website of the attacker's choice which could have been exploited further to install malware.

IV. Defences and Best Practices

A. Enterprise Mobile Security

1) Mobile Device Management

An MDM solution can be incorporated into an infrastructure that issues work phones or supports Bring Your Own Devices (BYODs). This solution allows administrators to centrally manage all mobile devices and ensure they are secure and are in line with company policies [30].

Here are some ways MDM's can help mitigate or prevent some of the attack vectors and exploits we have covered:

- Disabling NFC and USB transfer mode on devices to prevent proximity-based attacks as well as juice jacking.
- Ensuring sideloading is disabled on Android devices prevents users from installing potentially malicious apps from third-party stores.
- Automatic remote rollout of updates and patches to all devices managed will guarantee no device is left vulnerable to potential critical exploits [31, 40].
- Whitelisting applications limits the landscape available to an adversary and enforces the principle of least privileges. This dictates that users should only be given permissions relevant to their role. For example, this prevents a member of HR installing network scanning utilities from an app store and performing reconnaissance [30].
- Should devices get stolen or compromised, an MDM can initiate a remote wipe of that device preventing any access to sensitive or confidential data stored [30].

MDM's offer a comprehensive amount of control over an estate however implementing these solutions into small and medium enterprises (SMEs) could present a challenge as these solutions require trained IT staff capable of writing, monitoring and iterating on policies to ensure all infrastructure is securely covered. Additionally, as highlighted by Batool, H. and Masood, A. [31] SME's may not have the expenses available to implement these solutions or may utilise a MDM with limited functionality and minimal coverage of an infrastructure.

2) Security Awareness training

Organisations will often implement comprehensive training and assessments to educate and teach employees on the security threats they may face within the workplace. This has been proven to be effective in reducing the number of compromises an organisation has to face with 89% of companies surveyed in Fortinet's 2024 global research report claiming to have observed improvements to their security posture after implementing these programmes [32]. When delivered effectively, this can build awareness around specific threats and in turn minimise the reputational damage and financial losses associated with breaches [13, pp. 324-325].

For example, the use of phishing awareness training and fake phishing campaigns within organisations can train users to identify spoofed emails, securely handle calls to verify a caller's identity and avoid interacting with fraudulent texts from potential threat actors.

B. Strengthening Mobile Defences

Device owners outside of managed environments have the sole responsibility of ensuring a device remains secure and should follow certain guidelines and practices to strengthen their mobile security.

- It is important that users monitor and audit app permissions on their device. As previously mentioned, the principle of least privilege also directly applies to the applications installed on a device. Apps may request specific permissions during runtime and it is imperative that users understand the purpose of granting said privileges [4, 41]. For example, a SMS messenger requesting access to your stored contacts is considered justified. However, a basic utility app, such as a flashlight or calculator, requesting access to your location is not relevant to its core functionality and could be an indication of malicious activity.

- The use of passcodes to lock a device is a fundamental step in protecting these devices; however, according to research from cybersecurity provider Kaspersky, only 52 percent of device owners use a passcode and 22 percent have anti-theft solutions in place [33]. Users should always use strong passcodes and biometric authentication. For iOS specifically, users who have extremely sensitive and confidential information on their device may enable 'Erase Data' which formats a device after 10 failed passcode attempts [34]. This can prevent brute-force attacks against compromised or stolen devices.

- Enabling two-factor authentication is a crucial defensive feature that should be set up for all applications that support it as a security measure. It's an account's second line of defence should a user's credentials be compromised.

- Sideloading apps should be avoided as the associated risks are far too great. If it's an absolute necessity, users should verify the source of the download was from a reputable site/developer with credible reviews and review app permissions requested.

V. Reflection and Conclusion

In this paper, we researched the threats and common attack vectors targeting mobile devices and explored the security controls and best practices to mitigate these threats. Despite these guidelines, the security posture required to secure devices should constantly be iterated upon to rise to emerging threats and exploits. For example, as researched by Alahmed et al. [35] AI-assisted vishing was utilised on multiple occasions to impersonate C-level executives requesting a transfer of funds into an attacker's bank account. Successful attacks utilising a technology in its infancy illustrates how dangerous these tools could soon become.

After researching defences, it's clear these devices offer a plethora of features that positively and negatively affect the security posture of a device. In managed environments, restrictions and guidelines are enforced to ensure devices are secure; however, outside of these estates, users are not provided the same enforcement and as observed with

Kaspersky's research [33, 42], it's clear a significant percentage of the general public don't utilise security controls built into operating systems. It's here I would propose the suggestion to make passcodes mandatory and follow policy requirements as described in NCSC's most recent password guidelines [36]. Further to this, I would also suggest iOS and Android curate presets for security settings, that users must select from during the first time setup process. The addition of a 'secure mode' could enable settings such as, 'Always Ask to Join Networks', 'Erase Data after 10 failed passcode attempts' and introduce settings like 'Ask Before Opening' that could change system behaviour by explicitly asking for confirmation before opening links and downloading files.

REFERENCES

[1] Press, G. (2024) How many people own smartphones? (2024-2029), Whats the Big Data. Available at: https://whatsthebigdata.com/smartphone-stats/.

[2] Security-enhanced linux in Android : Android Open Source Project (2024) Android Open Source Project. Available at: https://source.android.com/docs/security/features/selinux.

[3] Yu, D. et al. (2021) SEPAL: Towards a Large-scale Analysis of SEAndroid Policy Customization. Available at: https://dl.acm.org/doi/pdf/10.1145/3442381.3450007.

[4] Kleymenov, A. and Thabet, A. (2022) Mastering malware analysis: A malware analyst's practical guide to combating malicious software, Apt, cybercrime, and IOT attacks. Birmingham, UK: Packt Publishing Ltd.

[5] M. Lefoane, I. Ghafir, S. Kabir, I. -U. Awan, K. El Hindi and A. Mahendran, "Latent Semantic Analysis and Graph Theory for Alert Correlation: A Proposed Approach for IoT Botnet Detection," in IEEE Open Journal of the Communications Society, vol. 5, pp. 3904-3919, 2024, doi: 10.1109/OJCOMS.2024.3419570.

[6] About the SafetyNET Attestation API deprecation : security : android developers (2024) Android Developers. Available at: https://developer.android.com/privacy-and-security/safetynet/deprecation-timeline.

[7] Reshetova, E. et al. (2015) Characterizing SEAndroid Policies in the Wild, arXiv. Available at: https://arxiv.org/abs/1510.05497.

[8] Mohamed, I. and Patel, D. (2015) Android vs iOS Security: A Comparative Study, IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/7113562/authors#authors.

[9] Wu, J. et al. (2022) A research of digital forensic method based on the Checkm8 heap vulnerability, IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/9688162.

[10] Abdulhamid, S. Kabir, I. Ghafir and C. Lei, "Dependability of the Internet of Things: Current Status and Challenges," 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives, Maldives, 2022, pp. 1-6, doi: 10.1109/ICECCME55909.2022.9987845.

[11] Meredith, D. (2022) Certified Ethical Hacker (CEH) v12 312-50 Exam Guide. Birmingham: Packt Publishing, Limited.

[12] Nicart, E.B. (2013) Security Analysis of iOS and Android as Basis for Mobile Operating System Security Enhancement, Academia. Available at: https://www.academia.edu/7835051/Security_Analysis_of_iOS_and_Android_as_Basis_for_Mobile_Operating_System_Security_Enhancement.

[13] Stoyko, T. (2022) How to obtain app approval on App Store and google play?, Incora. Available at: https://incora.software/insights/app-approval-on-app-store-and-google-play.

[14] Naser, M., Al Bazar, H. and Abdel-Jaber, H. (2023) Mobile Spyware Identification and Categorization: A Systematic Review, Informatica. Available at: https://www.informatica.si/index.php/informatica/article/view/4881/2472.

[15] Diogenes, Y. and Ozkaya, E. (2022) Cybersecurity – Attack and Defense Strategies. 3rd edn. Birmingham: Packt Publishing, Limited.

[16] Chen, J. et al. (2017) Uncovering the Face of Android Ransomware: Characterization and Real-Time Detection, IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/8241433.

[17] Alsoghyer, S. (2019) Ransomware detection system for android applications, MDPI. Available at: https://www.mdpi.com/2079-9292/8/8/868.

[18] S. Eltanani and I. Ghafir, "Coverage Optimisation for Aerial Wireless Networks," 2020 14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 2020, pp. 233-238, doi: 10.1109/IIT50501.2020.9299076.

[19] Cunningham, C. (2020) 'Ransomware goes mobile', in Cyber Warfare – Truth, Tactics, and Strategies. Birmingham: Packt Publishing, Limited, p. 77.

[20] Dashevskyi, S. et al. (2019) Dissecting Android Cryptocurrency Miners, arXiv. Available at: https://arxiv.org/abs/1905.02602.

[21] Dredge, S. (2014) BadLepricon malware caught stealth-mining bitcoin in Android apps, The Guardian. Available at: https://www.theguardian.com/technology/2014/apr/25/badlepricon-malware-bitcoin-mining-android-apps.

[22] Gasparis, I. et al. (2017) Detecting Android Root Exploits by Learning from Root Providers, Usenix. Available at: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-gasparis.pdf.

[23] Xiao, C. (2015) KeyRaider: IOS malware steals over 225,000 Apple accounts to create free app utopia, Unit 42. Available at: https://unit42.paloaltonetworks.com/keyraider-ios-malware-steals-over-225000-apple-accounts-to-create-free-app-utopia/.

[24] Ciaramella, G. et al. (2022) A Model Checking-based Approach to Malicious Family Detection in iOS Environment, ScienceDirect. Available at: https://www.sciencedirect.com/science/article/pii/S1877050922011437.

[25] Avoine, G. et al. (2021) From Relay Attacks to Distance-Bounding Protocols, Springer Nature. Available at: https://link.springer.com/chapter/10.1007/978-3-030-10591-4_7#Sec1.

[26] Thorpe, C., Tobin, J. and Murphy, L. (2020) An ISO/IEC 7816-4 Application Layer Approach to Mitigate Relay Attacks on Near Field Communication, IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/9229124.

[27] R. Omowaiye, I. Ghafir, M. Lefoane, S. Kabir, A. Qureshi and M. R. Daham, "Artificial Intelligence and Big Data Analytics for the Detection of Fake News on Social Media," 2024 4th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Male, Maldives, 2024, pp. 1-6, doi: 10.1109/ICECCME62383.2024.10796409.

[28] Veerasamy, N. (2021) 'The Threat of Juice Jacking', in ECCWS 2021 20th European Conference on Cyber Warfare and Security. Academic Conferences and Publishing Limited, pp. 449–453. Available at: https://www.google.co.uk/books/edition/ECCWS_2021_20th_European_Conference_on_C/wCo4EAAAQBAJ?hl=en&gbpv=0.

[29] Singh, Debabrata et al. (2022) Juice Jacking: Security Issues and Improvements in USB Technology, MDPI. Available at: https://www.mdpi.com/2071-1050/14/2/939.

[30] Akande, O.N. et al. (2022) SMSPROTECT: An automatic smishing detection mobile application, ScienceDirect. Available at: https://www.sciencedirect.com/science/article/pii/S2405959522000868.

[31] Bravo, C. (2021) Mastering Defensive Security. Birmingham: Packt Publishing, Limited.

[32] Ashfaq, S. et al. (2024) Defending Against Vishing Attacks: A Comprehensive Review for Prevention and Mitigation Techniques, Springer Nature. Available at: https://link.springer.com/chapter/10.1007/978-981-99-9811-1_33.

[33] I. Ghafir, and V. Prenosil. "DNS query failure and algorithmically generated domain-flux detection." In International Conference on Frontiers of Communications, Networks and Applications, Malaysia, pp. 103-107, IET, 2014.

[34] Social Engineering - Winning SECTF call at DEF CON 25 (2018). Christian Kirsch. 5 January. Available at: https://www.youtube.com/watch?v=yhE372sqURU.

[35] Rocchi, W. (2022) 'Managing mobile devices', in Cybersecurity and Privacy Law Handbook. Birmingham: Packt Publishing, Limited, pp. 181–182.

[36] Batool, H. and Masood, A. (2020) Enterprise Mobile Device Management Requirements and Features, IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/9162763.

[37] Fortinet (2024) 2024 security awareness and training, Fortinet. Available at: https://www.fortinet.com/content/dam/fortinet/assets/reports/report-2024-security-awareness-and-training.pdf

[38] Rimol, M. (2018) Kaspersky Lab finds over half of consumers don't password-protect their mobile devices, Kaspersky Lab Finds Over Half of Consumers Don't Password-Protect their Mobile Devices. Available at: https://usa.kaspersky.com/about/press-releases/kaspersky-lab-finds-over-half-of-consumers-don-t-password-protect-their-mobile-devices?srsltid=AfmBOor9TD-AmIKOErTdOnHIbsnkIRH8hYZAQsxa9_rNttlxvcvOuthK.

[39] Apple (no date) Set a passcode on iPhone, Apple Support. Available at: https://support.apple.com/en-au/guide/iphone/iph14a867ae/ios.

[40] M. Lefoane, I. Ghafir, S. Kabir and I. -U. Awan, "Latent Semantic Analysis for Feature Selection: A Proposed Approach for Anomaly Detection in Network Traffic," 2024 14th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 2024, pp. 517-522, doi: 10.1109/ACIT62333.2024.10712556.

[41] Alahmed, Y., Abadla, R. and Al Ansari, M.J. (2024) Exploring the Potential Implications of AI-generated Content in Social Engineering Attacks, IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/10703950.

[42] NCSC (2018) Password policy: Updating your approach, NCSC. Available at: https://www.ncsc.gov.uk/collection/passwords/updating-your-approach#tip5-password-collection.