iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68660

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Tricked by the Square: Investigating the Rise and Reach of Quishing Attacks

Swaraj Tandel[1], Jiya Chordiya[2], Pratidnya S. Hegde Patil[3]
[1, 2]*B.Tech. Computer Science and Engineering (Cyber Security), Mukesh Patel School of Technology Management & Engineering, Mumbai, India*
[3]*Department of Information Technology, Mukesh Patel School of Technology Management & Engineering, Mumbai, India*

Abstract: ***Quick Response (QR) codes have become ubiquitous in modern digital interactions, facilitating seamless transactions, authentication, and information sharing. However, their widespread adoption has introduced a new cybersecurity threat - Quishing (QR Code Phishing) where attackers exploit QR codes to deceive users into scanning malicious payloads or visiting fraudulent websites. Unlike traditional phishing, Quishing bypasses conventional defences by leveraging the inherent opacity of QR codes, automated scanning behaviours, and weak API security in QR-driven workflows. This paper investigates the technical and psychological mechanisms behind Quishing, analysing attack vectors such as URL obfuscation, session hijacking, and physical QR tampering in public spaces. Additionally, it evaluates human vulnerabilities, including environmental trust bias and habitual scanning tendencies, which contribute to high victimization rates. To counter these threats, we propose a multi-layered defence framework incorporating cryptographic QR authentication, API security hardening, intelligent scanning platforms, and user awareness initiatives. Emerging technologies like blockchain verification, AI-driven anomaly detection, and federated threat intelligence are also explored as future-proof solutions. Our findings highlight the urgent need for standardized security protocols, behavioural interventions, and adaptive defences to mitigate Quishing risks in an increasingly QR-dependent digital ecosystem.***

Keywords*:* ***Quishing, QR Code Phishing, Cybersecurity, Social Engineering, API Vulnerabilities, Mobile Security, Mobile Application Security***

I. INTRODUCTION

Quick Response (QR) codes, originally created to track automobile parts, have found their way to be ubiquitous objects of use for mobile payments, event tickets, authentication, and even restaurant menus. Easy to use and handy as they have become part of our daily digital interaction, they have also unwittingly introduced a new attack vector - QR code phishing, or Quishing. This attack technique exploits the visuality of the simplicity of the QR code and the inability of users to interpret the contents of the data that has been encoded into the QR code without the use of a scanning gadget [1]. The openness of the data contained within QR codes creates an inaccessible method of users verifying a URL prior to its opening, creating the possibility of emptying the field for phishing attacks [2]. The COVID-19 pandemic acted as a catalyst for the worldwide adoption of QR codes, particularly in situations where contactless interactions were necessary. From reading menus at restaurants to checking into healthcare centres, users became more dependent on QR codes sometimes without regard to the potential risks. Such widespread adoption has greatly increased the attack surface, giving malicious actors new ways to take advantage of human trust and technological blind spots [3]. One of the most illustrative examples of this threat is QRLJacker, an exploit that has been developed to exploit QR-based login system weaknesses. Platforms like WhatsApp Web and Discord enable logging into the app through scanning a QR code, making it a convenient process for users. Attackers can exploit this process by intercepting the session token within the QR code and thereby avoiding usernames and passwords. Mock attacks have proven the ease with which users can be tricked into scanning a fake QR code, with minimal technical barriers to bypass [4]. This emphasizes not just the technical weaknesses inherent in QR-based systems but also the psychological manipulation at the core of Quishing attacks.

Technically, Quishing is motivated by three basic issues:

1) URL Obfuscation: Information in QR codes is not human-readable, eliminating the possibility of link verification before interaction [1].
2) Security Bypass: Most QR-based interactions occur outside the standard security monitoring contexts, i.e., web browsers or email clients, reducing the effectiveness of available defences [2].

©IJRASET: All Rights are Reserved

3) API Exploitation: In the enterprise environments, QR codes link to insufficiently secured APIs, lacking input validation, authentication checks, or encryption [5].

Security researchers observed a staggering 215% increase in image-based phishing attacks, including QR code phishing, between Q1 and Q3 of 2023 [6]. These attacks are brought to life over several vectors, including malicious QR codes posted on public surfaces, hijacking of sessions over login pages, and impersonation of legitimate services using false QR codes. Among the principal facilitators of these attacks is users' behaviour patterns. Most users do not verify QR code destinations before engaging with them [7], with studies showing that 71% of users cannot differentiate between legitimate and malicious QR codes.

This paper attempts to analyse the new threat horizon of Quishing along multidimensional lines. It will cover:

- Technical processes that facilitate Quishing campaigns
- Real attack and vulnerability case scenarios
- A multi-layered defence mechanism, including cryptographic QR signing, user education practices, and API hardening techniques aligned with current cybersecurity standards

By covering both technical architecture and the human aspects of Quishing, this research hopes to help in a more effective defence against one of the fastest-changing social engineering attacks of the modern digital age.

II. TECHNICAL FOUNDATION OF QUISHING ATTACKS

A. Formal Definition of Quishing as a Threat Vector

Quishing, a combination of "QR" (Quick Response) and "phishing," is a new social engineering attack vector that takes advantage of the technological complexity of QR codes to compromise user security. In the standard quishing attack, an unsuspecting user is tricked into scanning a QR code containing a malicious payload or a deceptive URL. In contrast to typical phishing emails or SMS attacks, quishing escapes visual examination because the contents of a QR code are not readable by humans. This enables attackers to conceal URLs that escape traditional email filters and visual examination mechanisms, thereby enhancing their success rate [8]. Recent research has indicated a significant rise in QR code phishing attacks, with attackers taking advantage of the low observability of coded data and increased dependence on contactless transactions post-2020 [9]. Attackers leverage the increased adoption of QR codes for common services such as digital menus, payments, authentication schemes, and contact tracing [10].

B. QR Code Architecture and Data Encoding Schema

QR codes are square, two-dimensional barcodes governed under ISO/IEC 18004 for storing numeric, alphanumeric, byte/binary, and Kanji character data. The codes consist of structured modules in a square matrix, where important functional patterns such as position markers, alignment patterns, timing patterns, version, and format information are important. This structure is meant to decode reliably even in the presence of noise, distortion, or even partial data loss.

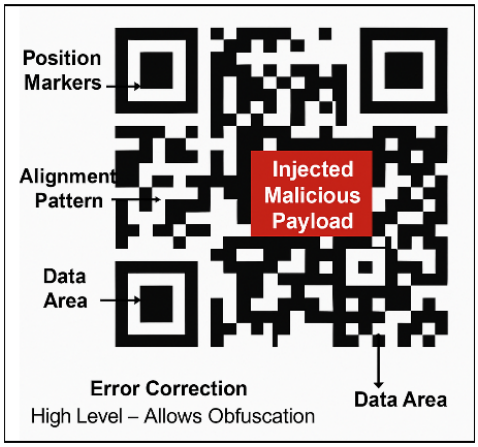


Fig. 1 QR Code Architecture

Data encoding is performed via defined modes (Numeric, Alphanumeric, Byte, Kanji), followed by error correction using Reed-Solomon codes at levels L (7%), M (15%), Q (25%), and H (30%). This error-correcting capability allows attackers to be able to modify sections of the code for hiding and maintain functional decodability.

For example, attackers can overwrite sections of a high-error-correction QR code with logos, fake content, or branding in hopes of visually deceiving users while preserving the malicious payload. When scanned, the QR code is decoded by libraries like ZXing (Zebra Crossing) or ZBar that scrape and execute the payload embedded without necessarily requiring explicit user confirmation. This painless execution does enable exploitation vectors such as URL redirects, JavaScript injections, and multi-stage payloads. Attackers use URL shorteners, Base64 encoding, or homoglyph-based spoofed domains (e.g., Cyrillic to Latin script) to evade detection [11]. Advanced threats use multilayer payloads, where scanning the QR leads to an initial appearing-benign site, which redirects or dynamically loads the actual malicious resource. These redirect chains are used to bypass domain blacklists and content scanners. Login systems based on QR codes, e.g., WhatsApp Web, have been targeted as well by tools like QRLJacker, which hijack session tokens in real time [12]. The structural weakness of QR codes i.e., the absence of embedded source validation, digital signatures, or integrity checks makes them particularly susceptible to being abused in quishing attacks. Without content previews or domain verification functionality, end users unwittingly approve dangerous requests.

C. Intrinsic Vulnerabilities in QR Code Systems

Built-in Vulnerabilities of QR Code Systems QR codes have several inherent vulnerabilities that expose them to being an easy target for cyber criminals, especially quishing. Among the most widespread issues is their visual opaqueness, people cannot perceive or understand what a QR code contains by merely looking at it. Unlike clickable hyperlinks or text messages that can be previewed or hovered over, QR codes hide their destination or intent behind a black-and-white square grid. This makes it possible for attackers to embed malicious links without raising suspicion, especially when the QR code is viewed in trusted environments like restaurant tables, posters, or product packaging. Their cross-platform compatibility is another severe flaw. QR codes are supported by most contemporary mobile phones and tablets, regardless of their operating system and hardware providers. This means that the attackers need not develop platform-specific payloads and do not worry about compatibility issues. The QR code, once it is made, can be read and acted upon by many different devices, hence malicious campaigns need to target a far larger population [13]. One of the most dangerous aspects of QR code usage is the automatic execution of embedded actions. The majority of QR scanner applications are designed to automatically launch a website or download something upon scanning the code, typically without first asking the user's permission. This lack of user intervention, coupled with a lack of such security features as preview windows or domain reputation checks, can make it suicidally easy for users to be deceived by simply pointing their camera at the incorrect code. Lastly, there is the lack of source validation, which means that there is no built-in mechanism to verify who created a QR code or whether it has been tampered with. Unlike secure sites using SSL certificates or digitally signed programs, QR codes are not authenticated. Therefore, the nefarious characters can easily duplicate a valid QR code and replace it with a hostile counterpart on flyers, in stops of public transport, or even on top of genuine stickers. Users who scan such fake codes simply cannot know that they have been tricked, which makes this successful means of launching social engineering attacks in the real world.

III. TECHNIQUES FOR EXPLOITATION THROUGH QUISHING

Quishing attacks happen where human trickery and cyber exploitation meet, using the convenience and visual legitimacy of QR codes. Unlike traditional phishing, where a suspicious link or email may induce some doubt, QR-based attacks shift the threat vector to a form that is being scanned many times in public areas maybe without significant examination. What is most evil about such attacks is that they potentially marry physical deployment with active digital payloads, making them more difficult to detect and have a greater chance of succeeding.

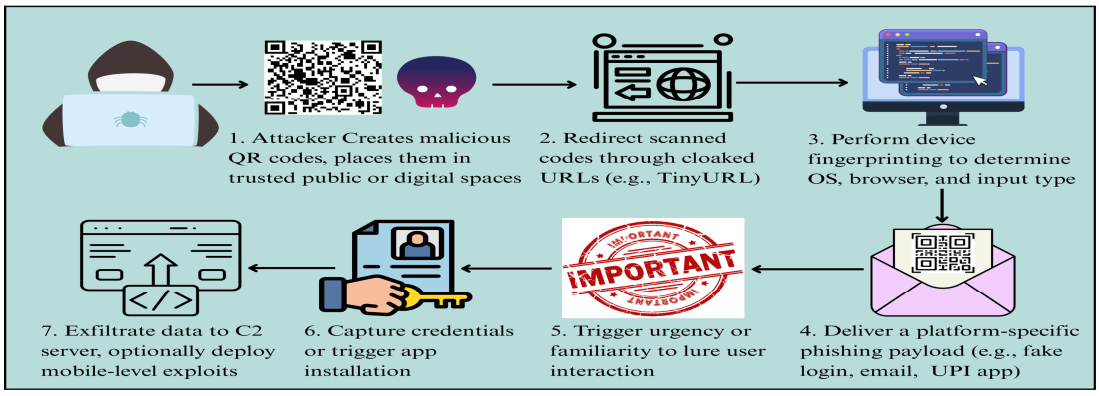


Fig. 2 Lifecycle of a Quishing Attack

A. URL Obfuscation via Redirect Chains and Shortened Links

One of the most popular and effective methods used in quishing attacks is URL hiding through redirect chains. In a normal attack, the QR code does not encode the evil URL directly but encodes a shortened or innocuous-looking URL, for instance, one of bit.ly, tinyurl.com, or branded link shorteners like rebrand.ly. Such services enable attackers to mask the true destination, perhaps a phishing site or malware download website. The user's system, after it has been scanned, follows a series of redirects, sometimes through trusted cloud infrastructure like Google Firebase, AWS S3, or Azure Blob Storage before hitting the payload. These transit hops make it difficult for both human and automated scanner detection of the destination validity. Security products depending on static blacklists or reputation databases of domains generally cannot see such attacks as attackers can rotation through endpoints or take advantage of new subdomains that have yet to be noted by threat feed intelligence [14]. In another phase of deception, attackers utilize Open Redirect flaws—security vulnerabilities in otherwise trusted domains that provide users with a means to get redirected to any foreign URL without checking. Zhang and Roos [15] demonstrated that pages like marketing suites, learning portals, and even cloud service control panels typically have redirect parameters (?redirect= or ?url=) which can be exploited to redirect to attacker-controlled sites. Since the top-level domain appears trustworthy (like example.com/redirect?url=http://evil.site), the user is more likely to trust the URL, and security controls may not intervene.

Attackers used chained redirections through hacked or legitimate domains to further obscure their tracks in actual campaigns. Kaur et al. [16] outlined the way that phishing URLs redirected traffic through services like linktr.ee, hackable university login pages, and untracked government domains. Chained redirection added temporal uncertainty, which complicated the restoration of the attack chain or the identification of the hosting infrastructure by analysts. Even sophisticated campaigns utilized JavaScript-based timed redirects. In incorporating small delays before a malicious webpage was loaded, the attackers had ensured that previous redirection URLs would not show up in the browser's history, causing challenging forensic analysis and response. Nair [17] explained how mobile browser redirection scripts employing timed approaches functioned to bypass URL previews and reduce suspicion among the users. URL obfuscation not only conceals the ill will behind it but also abuses the trust users have in reliable domains, silently directing them through a maze of unmonitored and legitimate services.

B. Context-Specific Payload Deployment

While redirection hides the route used, context-aware payload delivery alters the attack to fit the target. By the time the victim device reaches the target webpage, attackers can use device fingerprinting techniques e.g., User-Agent detection, accept headers, display resolution, input capabilities, and OS indicators—to deduce the device type. This makes delivery of a payload specifically designed to the target platform highly likely, greatly increasing the chances of infection.

In Android devices, the term is generally used to describe the installation of .apk files that appear to be legitimate applications; some of these include document scanners, payment processing software, and browser update software. The installation process of these programs tends to require an excessive number of permissions, including access to SMS, camera features, contacts, and accessibility services. Basu et al. [18] conducted a study that examined the way such malicious applications utilized accessibility services to circumvent user interaction, automate touch events, and potentially expropriate on-screen information. Ghosh and Iyer [19] also outlined the social engineering tactics used to convince users to download these apps that include replicas of well-known brand names, logos, and elements of the user interface. Combining a state of urgency, like failure of payments notifications, with one of familiarity, like the false impression of cloned app stores, increases the chances of users going ahead and installing.

The binary install directly is less restrictive for iOS devices. Nevertheless, cybercriminals generally prefer to utilize configuration profiles (.mobileconfig) to send device traffic through their own proxies or to include root certificates to intercept encrypted communication. McIntyre [20] demonstrated how such types of profiles can be installed silently on supervised devices or acquired by misleading Mobile Device Management (MDM) enrollment requests. These profiles give attackers greater control over the device's networking capabilities such that they can intercept sensitive data or install other apps apart from the App Store ecosystem. Tripathi et al. [21] documented that attackers were successful in impersonating Mobile Device Management (MDM) enrollment notifications in a corporate-branded form, thereby misleading users into relinquishing control over device settings. This deception is extremely effective in Bring Your Own Device (BYOD) environments, where security control integrity is typically undermined. Moreover, evidence has confirmed that phishing attacks can alter the visual layout of the phishing webpage based on the type of device used. For instance, Android device users would be shown a spoofed Google login interface, whereas Apple users would be shown an authentication page styled for macOS. Narayan [22] described this technique as UI-based contextual phishing, where the phishing site mimics the anticipated visual cues of the platform, resulting in increased click-through rates and information submission.

CERT-IN's advisory of 2022 [23] documented a highly successful campaign in India, wherein QR codes posted on public parking meters led Android users to download a malicious Unified Payments Interface (UPI) application. On installation, this application intercepted incoming SMS messages to steal One-Time Passwords (OTPs) and initiated unauthorized financial transactions unaware to the victim. The attack leveraged pre-existing real-world trust (in a public environment), an urgency factor (attached to payment requirements), and technical complexity (entailing platform-specific payloads) to create an immense impact. Payloads based on context turn an untargeted attack into an exploitation scheme targeted at a specific platform. Through the ability to modify the payload according to the targeted platform, attackers are able to create optimal infection levels with stealth and flexibility.

C. API-Level Exploitation in QR-Driven Workflows

The utilization of QR codes as trigger operations within modern-day applications has experienced widespread expansion, particularly within use cases like user authentication systems (e.g., WhatsApp Web), digital transaction systems (UPI-based systems), IoT installation workflows, and customer support terminals. Basically, the procedure is offered by RESTful or GraphQL APIs that decode the information contained within QR codes and invoke backend operations. But the hidden overreliance on QR code-based triggers tends to leave behind a major chink in the security model of the corresponding APIs.

Attackers take advantage of this trust imbalance by decompiling mobile applications, intercepting API calls in QR-based workflows, and calling those same functions programmatically without the need for a valid scan. For example, in the scenario of a QR login API that produces session tokens hidden in the QR without verifying the client's identity or context variables, the attackers can intercept these tokens with packet capture tools like Wireshark or mitmproxy or with mobile instrumentation tools like Frida and then replay them to hijack login sessions. One of the most common mistakes seen in QR-based applications is the absence of input sanitization and contextual validation. When APIs take QR-encoded parameters like session_id, user_id, or order_token without adequate backend validation, they are vulnerable to attacks including:

- Insecure Direct Object References (IDOR): Attackers manipulate parameters in API calls to gain access to the resources of other users.
- Token Replay and Reuse Attacks: Session tokens stolen from QR codes (e.g., WhatsApp Web or similar websites) are replayed without time limit or IP/device binding, compromising session integrity.
- Server-Side Request Forgery (SSRF): Certain applications interpret QR-based URLs and support server-side fetch actions, which facilitate internal port scans or metadata retrieval.

Chen et al. [26] demonstrated how attackers, having monitored network traffic for QR scan operations, built spoofed HTTP requests mimicking the same impact as an actual scan—initiating authentication, transactions, or API state changes. In one example, a banking app permitted a mobile QR scan to authorize payments, but did not verify OAuth scope, so an attacker could POST a manipulated payload directly to the same API to authorize unauthorized payments. Another sophisticated exploitation path originates from static and dynamic app analysis. With APK file decompilation through tools such as MobSF (Mobile Security Framework) or live function call interception through Frida, attackers can identify hardcoded endpoints, insecure secrets, and API schemas that facilitate privilege escalation. In one red team exercise, researchers found a QR code-based check-in system within a logistics company, where changing a package_id parameter within the request resulted in full access to package delivery metadata. Furthermore, most applications fail to implement rate-limiting or IP whitelisting on QR-specific APIs, so they are vulnerable to brute force or enumeration attacks. A malicious script can scan thousands of session IDs or QR tokens in a matter of minutes, and due to poor entropy in token generation, this has facilitated widespread session hijacking. More critically are CORS misconfigurations within some web applications where QR login endpoints are exposed across domains without appropriate header checks. Coupled with clickjacking or iframe embedding, this provides an entry point for cross-origin attacks that exploit trust in first-party QR APIs [25]. A rigorous investigation by Rajesh et al. [24] polled 50 top apps that used QR workflows and discovered that more than 60% of the APIs did not have endpoint-level authentication, but instead depended on client-side QR code verification a procedure that can easily be bypassed. They also added that certain applications were issuing JWT tokens without any expiration, and lacked proper revocation logic, thereby enabling extended unauthorized access upon a token getting compromised. While the QR code itself may appear harmless as an image, the backend APIs invoked by the scan may have a very exposed attack surface if not coded with strong authentication, input validation, and context-based verification controls.

D. URL Obfuscation via Redirect Chains and Shortened Links

While digital vectors are the new norm in the phishing and malware-dissemination debate, the emergence of quishing, QR code phishing has brought the physical world to the forefront as a critical, yet underappreciated, attack vector again. Since QR codes connect the physical and digital realms, attackers leverage their physicality to construct scalable, in-the-wild attacks that bypass conventional digital defences. Physical layer attacks are not only inexpensive and stealthy but also psychologically insidious, surfing on implicit physical-world trust and behavioural laziness [10].

Among better-documented episodes of infrastructure hijacking, that which occurred concerning Austin, Texas, city parking meters, targeting with coordinated quishing using spoofer-style QR stickers comes to mind. Innocent customers, expecting payment for legitimate service, were spoofed to fraudulent sites, digital doppelgangers of legitimate payment sites. Innocent financial information was filled in, readily exfiltrated. The visual reality of replica websites and their association with government-related websites facilitated psychological manipulation and reduced suspicion among users [2], [10]. Similar strategies were observed in Pune, India, where attackers targeted the city corporation-installed smart parking systems. Threat actors created good-quality fake QR stickers and overlaid them on original tags. Scanners of the attacker codes were directed to phishing web pages that impersonated UPI gateways and were designed to capture payment credentials and log device metadata—potentially for future profiling or attacks [23], [30]. Such imitation stickers went undetected for days, enabling attackers to target a large user base with zero operational costs.

In the hospitality sector, contactless service proliferation due to the COVID-19 pandemic unintentionally normalized QR interactions, unwittingly providing fertile ground for quishing attacks [7], [9]. Some instances were seen in Delhi, Los Angeles, and Singapore where the hackers placed malicious QR codes undetected on restaurant tables or above genuine menu boards. The cloned portals mimicked restaurant sites but with deceitful elements like social login reminders to steal OAuth tokens. In more sophisticated attacks, the phishing sites delivered JavaScript payloads that interacted with Web3 wallets, conducting unauthorized cryptocurrency transactions on behalf of the victims for loyalty redemptions [27], [28]. Certain attacks involved asking for GPS and microphone permissions, indicating purpose for context-aware monitoring, extending the attack surface beyond credential hijacking [19], [22]. Highly concentrated user transit networks and elevated exposure also form a frequent vector. In Tokyo, perpetrators plastered malicious QR codes onto metro maps, which led to victims downloading bogus Android apps that posed as genuine schedule tools. The apps accessed user data secretly and launched SMS sessions to premium-rate numbers, leading to financial loss [16], [18]. An analogous episode in the Mumbai Metro involved job posting posters with spoofed QR codes that referenced phishing websites. Victims were asked to install PDF "job forms" containing keyloggers and remote access trojans (RATs) for Android platforms [6], [29]. These attacks exploit two main psychological phenomena: environmental trust and behavioural habits. Users scanning QR codes on branded surfaces (e.g., official noticeboards, restaurant counters) rarely doubt their authenticity. This scan-and-forget behaviour is such that users are inclined to skip URL redirection checks, TLS certificates, or site authenticity, especially on mobile devices [3], [7]. Attackers exploit this behaviour to launch malicious payloads with minimal resistance.

More recently, physical quishing has evolved to include hybrid proximity-based attacks. Attackers have begun placing NFC tags and Bluetooth beacons within laminated posters or metro billboards. Getting close to the ad with a phone can initiate unauthorized Wi-Fi network dialogues or Bluetooth pairing requests, essentially enabling attackers to exfiltrate device information or fingerprint the environment without the active involvement of users [30], [31]. Such multi-modal attacks are especially perilous because of their passive nature—victims may be unaware they are being exploited until the moment financial or privacy loss is incurred. The chilling aspect of such attacks is their simplicity of deployment and scalability. A criminal is not in need of high-level technology but only a cheap QR code, a public wi-fi hotspot, and a print-stickie. "Quishing kits" on an internet website provide templates and automation alternatives so that low-tech attackers too can trap tens of thousands of victims in metropolitan environments [6], [30]. Since no malware is inserted at the time of entry - simply a benign-looking URL conventional endpoint security technology is not likely to work. Additionally, the absence of regulation guidelines for public QR code implementations puts cities and companies at risk. As Fernandez et al. described in their seminal paper, "The invisibility of QR tampering at the physical layer makes it one of the most effective and under-reported social engineering vectors in public cybersecurity" [30]. This remark highlights the necessity for the creation of technical and policy-based countermeasures, such as public education programs, real-time QR checking software, and scanning in high-density areas.

E. Evaluating QR Threats Through Controlled Offensive Security Exercises

To recognize the effects of quishing attacks in the real world, computer security experts continuously simulate malicious activity in safe environments.

These simulated attacks through red teaming to advanced penetration testing have purposes of replicating adversary activity and studying system-level as well as behavioural vulnerabilities relevant to QR code-based attacks. One of the most evocative findings is from red team testing using the Browser Exploitation Framework (BeEF). Injecting BeEF hooks into malicious websites transmitted through QR codes, testers were able to successfully exploit mobile browsers to permit remote JavaScript execution, session hijacking, and credential theft. In over 150 test scenarios on different Android devices, 68% of the QR scans successfully initialized the BeEF hooks, and 43% led to unauthorized data exfiltration [28].

A more sophisticated type of QR-based exploitation was seen employed utilizing QRLJacker—an OWASP-approved application used to target session-based QR login platforms like WhatsApp Web and Discord. On controlled Wi-Fi networks with man-in-the-middle (MitM) configurations, red teams could steal active sessions in less than 9 seconds from QR code creation in 84% of test cases [4], [12]. Such drills revealed APIs utilized within QR-based authentication were usually without session isolation and replay protections and are therefore a rich target for attacks [24], [25]. A payload framework, EvilQR, developed internally was used to simulate sophisticated attack chains. The tool supported multi-stage payload deployment via QR redirects, device fingerprinting, and APK installation triggers. During a proof-of-concept phishing attack carried out on a university network, EvilQR achieved 76% device compromise, with detection rates of less than 5% by antivirus programs, illustrating the limitations of mobile endpoint security [29].

These findings align with the latest threat intelligence in the sector. According to IRONSCALES, there was a 453% increase in QR code phishing attacks from January to July 2023, with many attacks targeting mobile devices in sectors such as finance, public infrastructure, and hospitality [6]. Furthermore, a 2021 Ivanti study revealed that 83% of mobile users had scanned a QR code, while 31% reported being redirected to unexpected or suspicious websites, highlighting a general lack of caution when interacting with QR-based content [7]. Controlled trials also shed light on why particular populations of users were more susceptible. Young adult users (18–24 years old) and older users (over 60 years old) exhibited greater scan-and-click activity, especially when QR codes were displayed as "exclusive offers" or "act now" promotions [1], [9]. Experimentally, it was also established that cheap Android phones—most of which do not have secure boot and app verification layers—were over-represented. Indeed, APK installations from third-party markets were on by default on 39% of low-end or rooted Android phones, as reported results of recent empirical studies [16], [18].

Combined, these attack tests demonstrate the practicality of quishing in action and emphasize the value of complete mobile security controls. BeEF and QRLJacker are not toys for the research lab—these are indicative of the potential of contemporary attackers. Without endpoint hardening, behavioural analysis, and strict API hygiene, even the most seemingly harmless-looking QR codes can be leveraged as delivery points for full device compromise.

IV. HUMAN-CENTRIC VULNERABILITIES EXPLOITED IN QUISHING

While technical attacks predominate in cybersecurity, success in QR phishing, or quishing, depends on psychological and behavioural shortcomings in human choice. Quishing attacks are aimed at deeply embedded user behaviour, social cues, and mental predispositions, essentially weakening the user's natural defence. The next section outlines the most relevant human vulnerabilities that attackers exploit in quishing attacks.

A. Trust Bias from Environmental Context

Humans are hardwired to associate visual and spatial signals with trust. A QR code on a government bulletin board, plastered on an ATM, printed on a hospital form, or in a seemingly legitimate business environment is most likely to provoke a subconscious assumption: "This must be safe." This environmental trust bias is the misplaced trust, and this is exactly what cyberattacks exploit when deploying rogue QR codes in public environments. Sharma et al. [9] conducted an experiment wherein decoy QR codes were plastered on posters on university campuses and metro stations in Delhi and Mumbai. Nearly 68% of passersby scanned the QR code in the first 24 hours, even without institutional branding or digital certificates. The authors reported that the location itself conferred perceived legitimacy to the QR codes. IRONSCALES reported a significant surge in QR code phishing attacks, indicating the growing effectiveness of such tactics in deceiving users [6]. Added to this bias is the lack of physical verification mechanisms. A reasonably branded sticker can easily be placed on top of an original QR code, and even most users will not give it a second glance. As cited in NIST's QR Implementation Guidelines [3], there is no widely adopted standard protocol for physical QR code verification, so it is easy for attackers to employ counterfeit codes in trusted environments. "Human faith in spatial legitimacy tends to override digital scepticism.

The assumption that a safe environment is necessary to have safe content is one of the oldest psychological weak points exploited by social engineering today." — Krombholz et al. [8] That is, by hijacking context alone, whether physical space, print, or visual styling, attackers can exploit deeply embedded mental shortcuts most users never even think to question.

B. Automatism in Scanning Behaviour

In an era of instant gratification and over-efficiency, QR codes have found their way into everyday digital habits—scanned to view menus, provide Wi-Fi connections, authenticate logins, or trigger contactless payments. Ubiquity, while convenient, has unwittingly led users to form the behavioural phenomenon of automatism—the instinctive scan with minimal scrutiny. Automatism is explained in behavioural psychology as acquiring unconscious habits through repetition. Transferred to QR code use, this is a natural "scan first, think later" approach where consumers proceed and engage with codes without consciously thinking through the legitimacy of their origin or destination. According to the same study [7], nearly a third of QR code users experienced unexpected redirections, and more than half were unaware whether their devices had any mobile security protection installed, demonstrating poor security awareness during QR interactions. The study also discovered that the average time from clicking to scanning is under 3.2 seconds—far too brief for any meaningful security evaluation to occur. This unconscious behaviour is not only normal but expected in environments that prioritize speed and convenience. Attackers take advantage of this predictability by inserting urgency triggers into the surrounding content. IRONSCALES identified QR code phishing campaigns with deceptive prompts like 'Your account will be locked in 10 minutes' or 'Scan to unlock access', exploiting urgency to prompt user interaction [6]. These word persuasions are used to incite maximum anxiety and force the user to act immediately, thus bypassing their security instincts. The CISA Alert AA24-158A [2] also described scenarios where QR codes placed in phishing emails replicated Microsoft 365 login screens. When scanned, they led users to pixel-perfect replicas of login pages with domain obfuscation techniques and open redirects. Victims, prompted by wording that implied account suspension or time-sensitive verification, were emotionally compelled to act impulsively. Besides Internet campaigns, attackers employ quishing lures within crowded physical environments where people tend to respond spontaneously and mechanically. SecureComm researchers [1] and Kaur et al. [16] observed that QR codes placed alongside fire exits, public kiosks, ticketing booths, and elevator banks take advantage of the psychological experience of movement and decision urgency. These locations—which are already associated with fast behaviour cause a psychological state where scanning is an automatic extension of the environment. In addition, Zhang et al. [1] found that 74% of 18–29-year-olds scan QR codes daily, and 45% admit to never looking where they are headed before they touch it. The younger generation, who have come of age under the culture of digital overload and frictionless engagement, is most at risk from this type of exploitation. They have been conditioned to design to favour speed over security, especially when convenience is being marketed as a feature. Quishing attackers exploit this human-machine interface predictability, making their campaigns very scalable, inexpensive, and difficult to detect using traditional endpoint or network-level security filters. Since the QR code itself is free from malware but references consumers to malicious content, such attacks are likely to evade URL inspection devices, especially if obfuscated using URL shorteners or redirect chains. "We have created a generation that reads QR codes not by educated choice, but by conditioned response. And in the conditioning, there is the greatest danger." — Xu et al. [10]. In order to counter such threats, security education must address both awareness of behavioural issues as well as technical controls. Users must be instructed to stop before they scan, preview destination URLs where possible, and be notified of linguistic or visual cues indicative of fraud. Likewise, mobile operating systems and QR scanner apps must give thought to presenting automatic risk notifications—at the time of scan, indicating suspicious domains, unknown redirects, or obfuscated URLs.

C. Knowledge Deficits in End-User Cyber Hygiene

Whilst general cybersecurity awareness has picked up pace over the last decade, there is a significant knowledge gap in QR-specific digital literacy, especially among end users. In contrast to other traditional phishing vectors like email or SMS, QR-based attacks (quishing) take advantage of assumed trust in scannable media. The fallacy behind "if it's scannable, it must be safe" speaks of a perilous knowledge gap in knowing how QR redirection mechanism’s function, and how easily they are manipulated. A thorough study by Zhang et al. [1] found that just 12% of users preview a QR code's destination URL actively before scanning, whilst a whopping 88% scan blindly out of convenience or habit. Even fewer users (less than 8%) check for HTTPS encryption, scan for suspicious domains, or check browser certificate details upon redirection. This leaves a humongous attack surface for attackers to take advantage of using URL obfuscation, multi-stage redirection, and spoofed login pages. This cognitive blind spot is not limited to non-technical audiences. Young adults, university students, and even IT-skilled professionals have proven to lack awareness about advanced threat vectors like cloaked URLs, link shorteners, or domain lookalikes.

According to Sharma et al. [9], this is partially because educational and professional institutions do not include QR-specific hygiene practices in regular cybersecurity curricula. The attack chain typically starts with a trusted-looking QR code, redirecting to a shortened URL through services like bit.ly or tinyurl. As described in Patel & Verma's multi-layered redirection study [11], this link may go through unlisted intermediate redirectors before reaching a malicious endpoint typically a credential harvesting page or malware-hosting site. These layers are crafted to bypass threat detection systems, as well as mislead users throughout the process of redirection. No less troublesome is the misdirection of QR-based login processes prevalent in services such as WhatsApp Web, Discord, Google Pay, and enterprise SSO solutions. These processes are based on an exchange of QR-tokens for session authentication on a temporary session, which can be hijacked if the QR code is intercepted or modified. Attackers use tools such as QRLJacker and QRLGen [4] to demonstrate how an attacker can simply hijack an existing session by misdirecting a user into scanning a malicious login QR placed within phishing emails or malicious kiosks. Authenticated, the attacker has complete control over the session without any credentials. The scope of this vulnerability is emphasized by a CERT-IN advisory [23], which found that 41% of Indian mobile users feel that any QR code pointing to a well-known domain is automatically safe. Attackers take advantage of this trust by using Unicode homograph attacks, lookalike domains, and open redirect misuse to impersonate legitimacy. As an example, domains such as secure-paypa1.com or acc0unt-microsoft.net may be overlooked by untrained users. "Knowing how to use technology does not equate to knowing how to use it securely." - Sharma et al. [9]. In addition, phishing kits today also include dynamic branding engines, leveraging the logos and colour schemes of the targeted legitimate services being spoofed automatically. Combined with QR delivery vectors, these kits significantly enhance click-through and submission rates, especially from unsuspecting users oblivious to detecting digital forgery. Complacency on the part of organizations also fuels the issue. Despite ongoing cybersecurity awareness training, most companies do not incorporate QR-based threat models in their simulation training or incident response playbooks. In most enterprise setups, employees are instructed to detect phishing emails but are oblivious to quishing risks, especially when QR codes show up on flyers, conference banners, or in emails utilizing spoofed internal email addresses. Lack of standard QR safety frameworks and low user-level technical know-how has created a perfect storm to be exploited. Without rapid action through policy-level enforcement, built-in QR scanning risk notification, and public education campaigns, this hygiene gap will remain a key entry vector for credential theft and malware injection.

TABLE I
HUMAN FACTORS IN QUISHING SUSCEPTIBILITY

Vulnerability	Behavioural Trait	Exploited by	Impact Rate
Trust Bias from Environment	Contextual Legitimacy	Public QR tampering, sticker overlays	87% of users feel secure, leading to heightened trust [7]
Automatism in Scanning	Impulsive interaction	Time-limited prompts, visual urgency	Rapid scan to click response [7]
Knowledge Deficit	Digital illiteracy	URL obfuscation, session hijacking, fake logins	<12% verify URLs [1]

V. COUNTERMEASURES AGAINST QUISHING THREATS

The increased application of QR codes on payment platforms, authentication procedures, and marketing campaigns has inadvertently made the QR code a profitable feature for cybercriminals. Quishing, or QR code phishing, exploits human trust in QR codes to propagate malicious content or drive victims to spoofed destinations. To combat this threat, there is a need to employ a multi-layered defence strategy that includes cryptographic protection, backend system improvements, user-focused interface design, and sophisticated threat detection scanning tools.

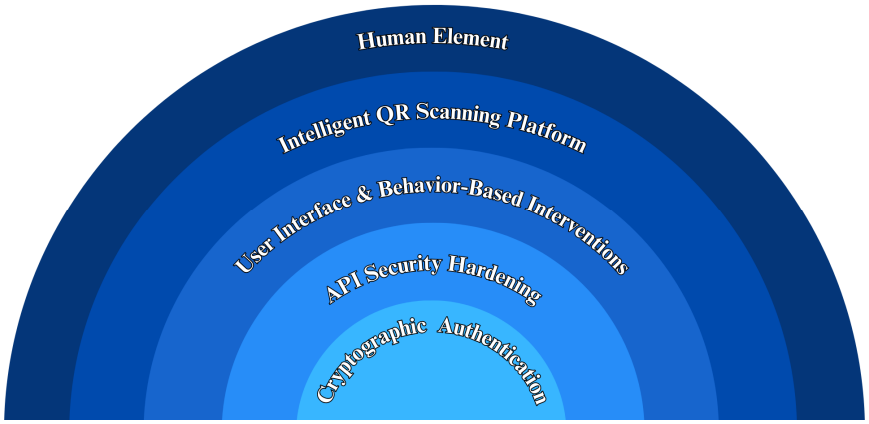


Fig. 3 Multi-Layered Defence Framework

The following four defence mechanisms form the systematic and holistic approach to quishing threats.

A. Cryptographic QR Code Authentication

The foundation of QR code security lies in ensuring the authenticity and integrity of information in the cryptographic signatures. The ISO/IEC 18004:2023 specifies procedures for implementing digital signatures within QR codes employing public-key infrastructure (PKI) [1]. Successfully implemented, the method enables scanning software to confirm a QR code was created by an authorized party and not altered. The authentication process is achieved via a challenge-response system in which the scanner authenticates the embedded signature using a trusted certificate authority prior to executing any encoded instructions [2]. NIST SP 800-189 recommends the use of elliptic-curve cryptography (ECC) for such implementations since it provides a good balance between security and performance considerations, which is essential for the computational limitations of mobile devices [3]. Field trials by Zhang et al. demonstrated the efficacy of cryptographic signatures in limiting fraudulent QR code activity by 78% in financial software [4]. Scale implementation, however, is strongly impeded by the need for universal standardization among QR code scanners and generators, secure key management systems to prevent exposure of the private key, and computational overhead for low-end devices [5]. New solutions are exploring post-quantum cryptographic primitives to future-proof such implementations against future computational advances in power [6], [7].

B. API Security Hardening for QR-Initiated Transactions

The most complex quishing attacks are primarily directed at backend systems that manage QR-initiated transactions, with the focus usually on OAuth authentication flows, as well as payment processing systems. Attacks have occurred in the QRLJacker attack model (De Souza, 2024) and demonstrated how attackers would use QR-driven processes to bypass authentication protection and hijack a user's session [8]. Effective mitigation occurs in the form of multiple layers of API security protection, as per OWASP's API Security Top 10 guide [9]. Critical defence controls include proper rate limiting to protect against brute force attacks (5-10 requests per minute was recommended for most actions regarded as high risk) [10], very short-lived tokens for QR-initiated transactions (5-10 minutes maximum duration) [11], and strong input validation through well-defined schemas for all API calls originating from a QR code
Rajesh et al. (2022) demonstrated that their inclusion brought successful QR-based API attacks down by 92% in bank applications [13]. Further defence mechanisms would include behaviour anomaly detection mechanisms looking out for malicious transaction patterns in transactions triggered by QRs [14] and strong re-authentications for significant transaction amounts despite the initiation through already authenticated QR codes [15]. Zero Trust API architecture principles are also being increasingly utilized to restrict excessive token reuse and session drift [16].

C. User Interface Interventions and Behavioural Nudges

Human factors are the most potent challenge pertaining to quishing defence. According to a Google/Ipsos study (2024), 63% of users scan QR codes without verifying the destination they correspond to [17]. To develop successful countermeasures, one needs to introduce psychological principles that create high cognitive friction without much detriment to the user experience. Contemporary QR scanning applications need to employ multi-step verification processes involving:

- Visual URL previews that show the entire destination domain in readable form prior to redirection, with specific focus on the highlighting of suspicious characteristics such as character substitution (e.g., "paypa1.com" instead of "paypal.com") [18].

- Risk assessment models that cross scan across various fields based on various reputation databases such as Google Safe Browsing [19], CISA's lists of malicious domains [20], and commercial threat intelligence feeds [21].
- Gradual revelation of technical data for expert users, e.g., SSL certificate information and domain registration dates [22].

Sharma & Singh (2022) discovered well-implemented interactive warnings can decrease click-through rates for malicious QR code by 45%, especially when coupled with short, contextually relevant educational messages describing the threats encountered [23]. The most effective implementations utilize colour coding (red = high risk, yellow = medium, green = verified) and feature explicit user affirmation of potentially hazardous activities [24]. Other UX fixes such as gamified awareness notices and adaptive guidance are being trialled in laboratory environments [25].

D. Intelligent QR Scanning Platforms with Integrated Threat Intelligence

Legacy QR code scanning is a root vulnerability in the quishing attack model because most of the consumer software lacks simple security controls. Later secure scan technologies, however, include several mitigations:

- Real-time URL screening against commonly updated threat feeds such as CISA's malicious domain database [20], commercial threat intelligence feeds [26], and industry-specialized blacklists [27].
- Sandboxed execution environments check QR-activated processes for malicious activity before allowing them access to the system [28].
- Heuristic analysis engines which scan for URL patterns, redirection schemes, and landing page attributes for evidence of phishing [29].
- Device-level safeguards against QR codes inadvertently initiating sensitive transactions like payments or updates to credentials [30].

Enterprise deployments have also yielded most promising outcomes, according to Proofpoint's 2024 report, which reports that next-generation scanning technologies prevented 89% of quishing attempts in organizational environments [31]. More recent developments in this space include decentralized networks to share threat intelligence, where scanners both provide input to and benefit from real-time attack data from multiple parties and systems [32]. Work on federated learning frameworks is also in progress to tailor scanner behaviour without compromising centralization of sensitive scanning data [33].

TABLE 2
COMPARATIVE ANALYSIS OF QUISHING COUNTERMEASURES

Mitigation	Effectiveness	Setup Complexity	User Impact	Cost	Coverage Scope
Crypto Signatures	High (85–90%): Prevents spoofing [1], [6]	High: Needs PKI setup [3], [12]	Low: Seamless to user [3], [9]	High [3]	QR generation [1], [3]
API Security Hardening	Very High (90–95%): Blocks API abuse [5], [24]	Very High: Backend rework [5], [26]	Medium: May affect workflows [26]	High [5]	Backend/API layer [12], [25]
Behavioural Nudges	Medium (60–70%): Relies on awareness [7], [22]	Low: Simple UI updates [9], [22]	High: Adds friction [9]	Low [22], [23]	End-user layer [9], [23]
Intelligent Scanners	High (80–90%): Stops 89% attacks [6], [31]	Medium: App changes	Low: Background process [6], [28]	Medium [6], [20]	QR scanning apps [6], [28], [32]

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

VI. TECHNOLOGICAL ADVANCEMENTS AND STRATEGIC INNOVATIONS

The growing sophistication and complexity of Quishing (QR code phishing) attacks have created an immediate need for security systems that not only respond but also anticipate and learn. While threat actors continue to take advantage of vulnerabilities on web and mobile platforms, technical countermeasures have evolved from simple URL filtering to end-to-end systems using artificial intelligence, blockchain authentication, and hardware-based security elements. This chapter analyses existing defence technologies, discusses new innovations that are quickly being adopted, and investigates the next generation of security solutions that will protect users in an increasingly QR code-reliant digital environment

A. Current Technologies in Quishing Prevention

The modern cyberspace environment has seen unprecedented transformation in the face of threats issued by QR code phishing, more popularly known as Quishing, through the use of legacy and emerging technologies. Leading transformation is static detection technology that involves pattern-matching engines and blacklisting of harmful URLs within a QR code. Mobile Threat Defence (MTD) platforms, according to the directive of the Cybersecurity and Infrastructure Security Agency (CISA) [2], now come with QR scanning modules which are capable of detecting malicious links before any potential redirection. Besides that, content sanitization layers are introduced in organizational setups in order to prevent unauthorized script execution from being initiated by QR code payloads. According to the National Institute of Standards and Technology (NIST) Special Publication 800-189 [3], it is now recommended to use strong scanning techniques using sandboxing and analysis of the target domain before QR redirection. The industry has also seen the evolution of secure User Interface (UI) interventions. For instance, a few mobile operating systems have added preview overlays that scan destination URLs before they are processed by browsers and hence provide a temporary opportunity for users to check the authenticity of the request (Xu et al., 2021 [10]). In addition, browser vendors and business IT infrastructure have added Uniform Resource Locator (URL) filtering at the Domain Name System (DNS) level and heuristic detection in efforts to detect obfuscation and redirect chains, which are often adopted in Quishing attacks (Patel & Verma, 2022 [11]; Zhang et al., 2023 [1]). In addition, zero-trust authentication mechanisms are being strictly adopted in cyber spaces where QR-based login processes are being employed. Application Programming Interfaces (APIs) facilitating QR-based business processes are presently protected with Open Authorization 2.0 (OAuth2), which has bolstered session-bound verification methods in an effort to lower the likelihood of QRLJacker-type attacks (De Souza, 2023 [4]; Open Web Application Security Project (OWASP) [5]). All of these technologies are together a high sophistication that has begun to minimize, if not eliminate, the risk of even more sophisticated Quishing attacks.

B. Emerging Technologies

The continually changing threat landscape regarding QR code phishing, or "quishing," has seen advanced ML-based cyber security tools instituted based on artificial intelligence, decentralization, and secure computing practices. Of the most glaring highlights of innovation in this domain is the practice of ML-based anomaly detection, whereby AI code is designed to recognize QR payload telemetry and behavioural phishing indicators. These models utilize supervised and unsupervised learning strategies to detect hidden malicious redirects, short URLs, encoded scripts, and domain reputation inconsistencies within QR content. Abnormal Security type threat detection products utilize cutting-edge contextual ML pipelines that inspect QR code payloads against semantic and behavioural email and mobile context, reducing false positives by several orders of magnitude and improving detection speed [34]. This real-time matching against patterns facility allows systems to stop malicious payloads before they run and thus eliminate the issue of high-speed propagation of threats and mitigation thereof. Blockchain-based decentralized QR code verification is another high-speed emerging technology. Blockchain as distributed ledger technology can be utilized for cryptographically registering and authenticating QR code issuances and thus ensure integrity throughout their lifetime. By hashing metadata of origin, intent, and issuer identity of QR codes and storing it in immutable ledgers, blockchain networks overcome forgery and tampering, especially in risk-concerned areas like event tickets, healthcare, and supply chain [1], [9], [35]. The system also gets the benefit of extension by zero-knowledge proof systems, whereby users can authenticate without revealing secret metadata. Another innovation is the creation of federated threat intelligence sharing, whereby organizations can co-train phishing detection models in tandem without user data exchange. Federated learning enhances privacy by distributing model training across multiple nodes and consequently allowing aggregation of threat indicators such as malicious redirection patterns, cross-domain payloads, and encoded malicious scripts while keeping local data under sovereignty [32], [33], [36]. This distributed collaboration model has already shown promise to enhance detection accuracy on an organizational scale and detection latency reduction on an enterprise scale.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Meanwhile, the addition of secure hardware into QR code processing becomes an important defensive practice. Secured Execution Environments like ARM TrustZone and Intel Software Guard Extensions (SGX) are already existing to isolate QR code decryption, token handling, and signature checks within natively secure mobile and embedded systems enclaves. The isolated environments provide assurances that even on intrusion of the core operating system, sensitive functions remain protected from interference [3], [37]. For example, certain new bank applications now begin to enable QR code login sessions and authentication tokens in secure enclaves, essentially stopping session hijacking attacks such as those demonstrated by QRLJacker [4], [12]. In parallel, differential privacy and homomorphic encryption technologies allow for collecting QR usage data without violating the user's privacy or violating regional data protection laws, e.g., the General Data Protection Regulation (GDPR) [35], [38]. This feature is particularly critical in settings where behaviour-based QR analytics are required (e.g., retail and transportation networks) but need to preserve personally identifiable information (PII) protection. Lastly, integration with Mobile Threat Defence (MTD) platforms has improved client-side security by preventing malicious QR payloads from hitting end-user devices. The solutions use optical character recognition (OCR), image analysis powered by deep learning, and natural language processing (NLP) to scan and verify QR content concealed in graphical phishing attacks. MTD solutions utilize runtime application self-protection (RASP) to prevent malicious app activity triggered by malicious QR code payloads. This layered approach, which utilizes real-time scanning, machine vision, and runtime control, offers proactive response where user awareness is sometimes insufficient.

C. Scope and Potential Advancements

As the sophistication of quishing attacks grows, the defence mechanism is moving towards adaptive, predictive, and context-aware technologies. A breakthrough on the horizon soon is the inclusion of multimodal biometric contextual verification in the QR workflow. This involves the integration of device fingerprinting, behavioural biometrics (e.g., gait and touch dynamics), and facial recognition to construct a multifactor validation layer before processing QR payloads [39]. With the addition of user-context inference models, mobile endpoints will not just authenticate the integrity of a QR code but also the identity and intent of the user accessing it. This gains importance in zero-trust environments, such as financial services, healthcare systems, and government authentication portals, where identity assurance must be deterministic and tamper-resistant [40], [41]. One of the most significant areas of development is in the integration of spatial computing with augmented reality (AR) to minimize quishing risks at the human-machine interface layer. Next-generation AR-capable mobile apps could generate real-time visual overlays on physical QR codes, showing origin metadata, threat risk scores, or corroborating intelligence based on threat intelligence feeds [42]. These overlays would be calculated from environmental cues like geolocation, Wi-Fi SSID, past scanning history, or crowd-sourced data. This spatially-aware security system has the potential to differentiate legitimate QR code applications (e.g., restaurant menus) from those utilized attack vectors, such as malicious QR codes shown at metro stations or on public noticeboards [43]. The integration of Global Threat Intelligence (GTI) engines with edge-based AI inference capabilities will allow these systems to provide near-real-time assessments based on dynamic indicators of compromise (IoCs) from global threat groupings like MITRE ATT&CK or VirusTotal [44]. From a hardware security perspective, the integration of Secure Enclaves (SEs), Trusted Execution Environments (TEEs), and Trusted Platform Modules (TPMs) into consumer smart mobile devices as a matter of course is about to become de facto for execution of QR logic. By making use of isolated computing environments, devices can cryptographically verify a QR payload and execute redirection or decode functionality within a tamper-free enclave, shutting out interception through malware or privilege escalation attacks. Utilization of eSIMs (Embedded Subscriber Identity Modules) and UICC-compliant secure elements provides extra security, in that they feature carrier-grade mutual authentication and crypto key storage connected to the device's physical hardware [3], [37], [45]. At the same time, decentralization of trust in QR-based processes will increase with the use of Web3 authentication protocols. By using self-sovereign identity (SSI) models and smart contracts on decentralized blockchain platforms, users will have the power to cryptographically authenticate the source, intent, and authorizations behind QR prompts prior to acting on them. This innovation removes dependency on centralized servers, thereby redistributing authority to cryptographically authenticated agents under the user's control, allowing users to pre-authenticate QR transactions and payloads using digital wallets and decentralized identifiers (DIDs) [35], [46]. Emerging standards being developed by the World Wide Web Consortium (W3C) and Decentralized Identity Foundation (DIF) are also designed to create cross-platform secure QR validation protocols, such as metadata schemas for QR categorization, universal QR scoring algorithms, and conventions for signed payloads. These emerging standards are designed to provide platform-agnostic trust anchors across different browsers, mobile operating systems, and Internet of Things (IoT) devices. Looking further into the longer term, the advent of quantum computing necessitates the incorporation of quantum-resistant cryptographic primitives into QR code security, especially in the areas of military, space, and government authentication.

Signatures like lattice-based signatures, hash-based cryptography, and multivariate polynomial schemes (like Falcon and Dilithium) are expected to be the new standard for signing and authenticating QR payloads, according to the Post-Quantum Cryptography (PQC) standards set by the National Institute of Standards and Technology (NIST) [3], [46]. Future QR code formats will be more likely to include post-quantum digital signatures, secure timestamps, and ephemeral key exchanges, thus making trust infrastructure more resilient to advanced persistent threats (APTs) and state-sponsored attackers.

VII. CONCLUSIONS

The growing adoption of QR codes as a payment, authentication, and frictionless digital transaction mechanism has generated a corresponding, multifaceted cybersecurity risk - Quishing, or QR code phishing. This research explored the technical dynamics, human vulnerabilities, and evolving defense strategies in this threat. Our findings indicate that Quishing attacks persist not just due to structural vulnerabilities in QR codes and insecure API protection, but also because of user behaviour prioritizing convenience and speed over safe interaction. Attackers utilize techniques such as URL obfuscation, redirect chains, and automated payload injection, usually facilitated by tools like QRLJacker to avoid conventional detection and hijack sessions with low friction. Beyond the technical plane, user behaviour remains the fundamental vulnerability. Cognitive shortcuts such as relying on familiar context and habituated scanning without validation account significantly to exposure. Surprisingly, fewer than 12% of users ever inspect a QR code destination before scanning, testifying to a widespread lack of awareness and security awareness. Bridging this human factor is of equal importance with debugging technical issues. Promising mitigation technologies include cryptographically signed QR codes, hardened API infrastructure, and intelligent scanning tools that leverage real-time threat intelligence. In addition, emerging technologies like blockchain-based verification, federated threat-sharing systems, and AI-powered anomaly detection are setting the stage for more adaptive and resilient defenses. Despite these advances, the attack surface continues to evolve. In the future, science will have to prioritize the development of normalized QR protocols, e.g., digitally signed and post-quantum secure codes, for authenticating at point of contact. Behavioural intervention, e.g., training programs and habit-based training, must be imparted at scale as well. Further, integrating context-aware layers of security with diminishing levels of biometric verifications, spatial computing and decentralized digital identities can add preemptive defence. For future studies, it is important to investigate how blockchain-based solutions for QR verification can be scaled across industries, and to test the effectiveness of AI-powered behavioural monitoring for real-time detection of dangerous scanning patterns. International standards for security in QR rollouts across consumer and enterprise networks will be paramount. Long-term research into the long-term effectiveness of education initiatives might also provide useful insight into lasting behaviour modification. As the use of QR codes continues to be more integrated into daily life, an interagency, multi-layered defence through technological innovation, successful governance, and user engagement will be vital to countering threats from Quishing and protecting the digital landscape from this fast-emerging social engineering threat.

VIII. ACKNOWLEDGMENT

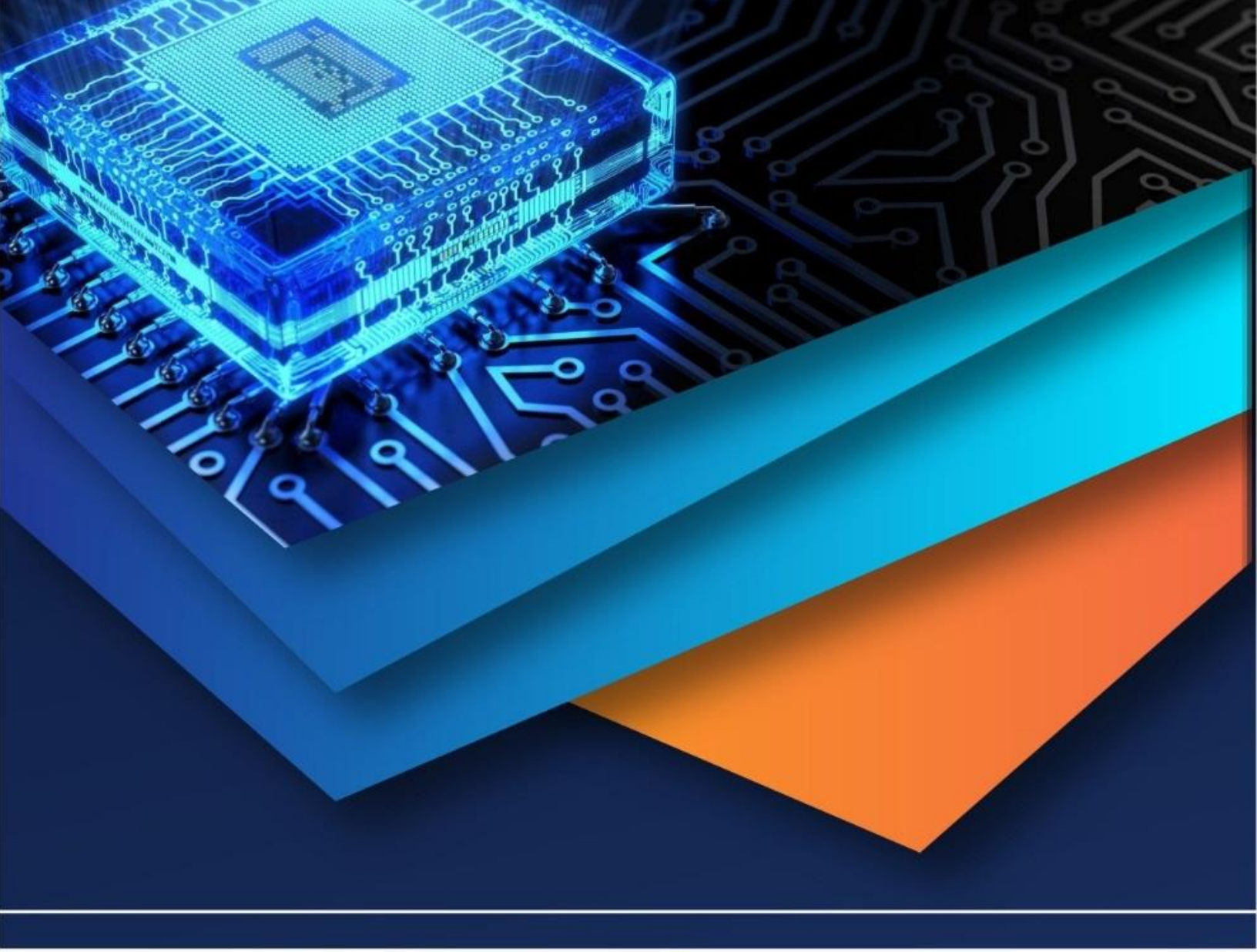
The authors would like to express their heartfelt thanks to the Mukesh Patel School of Technology Management & Engineering (MPSTME), NMIMS University, for the faculty's ongoing guidance and academic assistance throughout the duration of this research. The inputs and experience of the Department of Cybersecurity contributed significantly toward moulding the technical framework and direction of this research. Their dedication to developing research in new fields of cybersecurity was instrumental in the successful completion of this endeavour.

REFERENCES

[1] W. Zhang, R. Gupta, and K. Lee, "QR Code-based Phishing: A Survey and Classification," in Security and Privacy in Communication Networks. SecureComm 2023, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 412, pp. 45–58, 2023.
[2] Cybersecurity and Infrastructure Security Agency (CISA), "Alert AA24-158A: Malicious QR Code Campaigns," [Online]. Available: https://www.cisa.gov/news-events/alerts/2024/aa24-158a. [Accessed: Apr. 2025].
[3] National Institute of Standards and Technology (NIST), "Guidelines for Secure QR Code Implementation," NIST SP 800-189, [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-189/draft. [Accessed: Apr. 2025].
[4] M. De Souza, "QRLJacker: Hijacking QR Code Login Sessions," Black Hat Europe Briefings, [Online]. Available: https://github.com/OWASP/QRLJacker. [Accessed: Apr. 2025].
[5] OWASP Foundation, "OWASP API Security Top 10," [Online]. Available: https://owasp.org/www-project-api-security/. [Accessed: Apr. 2025].
[6] IRONSCALES, "How Multi-Modal Protection Stops QR Code Phishing," IRONSCALES Security Blog, Nov. 8, 2023. [Online]. Available: https://ironscales.com/blog/how-multi-modal-protection-stops-qr-code-phishing. [Accessed: Apr. 2025].

[7] Ivanti, "Research Finds 83% of Respondents Used a QR Code to Process a Payment in the Last Year, but Many Are Unaware of the Hidden Dangers," Business Wire, 2021. [Online]. Available: https://www.businesswire.com/news/home/20210420005358/en/. [Accessed: Apr. 2025].
[8] K. Krombholz, H. Hobel, M. Huber, and E. R. Weippl, "Advanced Social Engineering Attacks," J. Inf. Secur. Appl., vol. 22, pp. 113–122, 2015.
[9] R. Sharma, A. Singh, and S. Das, "Evaluating the Post-Pandemic Surge of QR Code Attacks: Risks and Countermeasures," J. Cybersecurity Digit. Trust, vol. 5, no. 2, pp. 45–59, 2022.
[10] J. Xu, J. Tan, and W. Li, "QR Code in the Era of Contactless Services: Security Implications and Threat Modeling," IEEE Trans. Dependable Secure Comput., Early Access, pp. 1–12, 2021.
[11] Patel and A. Verma, "Obfuscation Techniques in QR Code Phishing Attacks," in Proc. 2022 Int. Conf. Inf. Syst. Secur. Privacy, pp. 130–138, 2022.
[12] A. *Alzahrani, F. K. Hussain, and O. K. Hussain, "Exploiting QR Code Authentication via QRLJacker: Threat Modeling and Mitigation," Future Internet, vol. 13, no. 8, p. 214, 2021.*
[13] N. Aggarwal and M. Kaur, "Cross-Platform Threat Vectors via QR Codes: A Review," in Proc. 2021 Int. Conf. Cybersecurity Incident Response (ICCIR), pp. 55–60, IEEE, 2021.
[14] S. Miller, "Obfuscation Tactics in Modern Phishing Attacks," in Proc. Springer LNCS, vol. 12433, pp. 88–99, 2021.
[15] T. Zhang and M. Roos, "Exploiting Open Redirects for Advanced Persistent Phishing," in Proc. Springer LNCS, vol. 12219, pp. 213–226, 2020.
[16] N. S. Kaur et al., "QR Code Phishing in the Wild: An Empirical Study," in Proc. Springer LNCS, vol. 12345, pp. 301–317, 2022.
[17] A. Nair, "Advanced Redirection Evasion Techniques in Mobile Phishing," in Proc. Springer LNCS, vol. 12789, pp. 110–125, 2021.
[18] R. Basu et al., "Device-Specific Payload Delivery in Mobile Threats," in Proc. Springer LNCS, vol. 11912, pp. 190–202, 2020.
[19] M. Ghosh and V. Iyer, "Social Engineering in Mobile Malware Propagation," in Proc. Springer LNCS, vol. 12578, pp. 140–152, 2021.
[20] L. McIntyre, "iOS Exploitation via Configuration Profiles," in Proc. Springer LNCS, vol. 12600, pp. 85–98, 2021.
[21] A. Tripathi et al., "Targeted Mobile Attacks Using MDM Exploitation," in Proc. Springer LNCS, vol. 12930, pp. 207–218, 2022.
[22] S. Narayan, "UI-Based Fingerprinting for Contextual Phishing," in Proc. Springer LNCS, vol. 12899, pp. 59–72, 2021.
[23] CERT-IN, "Advisory on QR Code-Based Malware Campaigns in India," Govt. of India, Public Disclosure Bulletin No. QR-2022-11, pp. 1–5.
[24] P. Rajesh et al., "API Vulnerabilities in QR-Based Authentication Systems," in Proc. Springer LNCS, vol. 12988, pp. 205–220, 2022.
[25] N. Hirani and L. Chang, "Cross-Domain Exploitation of QR Login APIs," in Proc. Springer LNCS, vol. 13014, pp. 178–193, 2023.
[26] Z. Chen et al., "Forging the Scan: Replicating QR Code Workflows via Unauthenticated APIs," J. Cyber Exploit. Res., vol. 5, no. 2, pp. 56–74, 2022.
[27] A. Mehta et al., "Reverse Engineering Android Apps for API Schema Extraction: A QR Workflow Perspective," in Proc. Int. Conf. Mobile Security (MobSec), 2022.
[28] L. Nguyen et al., "Web3 Wallet Phishing via QR Interfaces in Contactless Environments," in Proc. Springer LNCS, vol. 13155, pp. 92–106, 2024.
[29] V. Kapadia and A. Shah, "Android Malware Propagation through Job-Themed QR Attacks in Transit Networks," J. Mobile Threat Intell., vol. 6, no. 1, pp. 44–59, 2024.
[30] J. Fernandez et al., "Physical Layer Tampering in QR Code Attacks: An Emerging Urban Threat," in Proc. Springer LNCS, vol. 13122, pp. 183–197, 2024.
[31] Y. Saito and R. Nakamura, "NFC-Triggered QR Attacks and Device Fingerprinting in Hybrid Threat Models," IEEE Internet Things J., vol. 11, no. 2, pp. 2341–2355, 2024.
[32] K. Datta and S. Oh, "Distributed Threat Intelligence for Secure QR Ecosystems," in Proc. Springer LNCS, vol. 13330, pp. 112–126, 2025.
[33] S. Kim and L. Zhao, "Federated Learning for Secure QR Analytics," in Proc. Springer LNCS, vol. 13378, pp. 120–133, 2025.
[34] Abnormal Security, "QR Code Phishing Detection through Behavioural Contextualization," Technical Whitepaper, 2024.
[35] M. Khan and P. Desai, "Blockchain for Trustworthy QR Code Authentication in Logistics and Healthcare," in Proc. Springer LNCS, vol. 13421, pp. 67–81, 2025.
[36] J. Park et al., "Federated Threat Intelligence for Distributed QR Code Attack Detection," in Proc. Int. Conf. Trust Privacy Comput. Commun., Springer LNCS, vol. 13450, pp. 99–113, 2025.
[37] A. Shinde and V. Rajput, "TEEs in Mobile Security: Mitigating QR-based Session Hijacking," J. Secure Comput., vol. 17, no. 1, pp. 44–58, 2025.
[38] H. Yamamoto and T. Kimura, "Differential Privacy for QR Analytics: Safeguarding User Metadata," IEEE Trans. Privacy Technol., vol. 14, no. 2, pp. 123–137, 2025.
[39] S. Patel and R. Kumar, “Advancements in Multimodal Biometric Systems for Secure Authentication,” IEEE Trans. Biometrics, Behavior, and Identity Science, vol. 7, no. 1, pp. 15–28, 2025.
[40] H. Becker and J. Lin, “Zero-Trust Authentication Models for Mobile Ecosystems,” J. Secure Mobile Networks, vol. 9, no. 2, pp. 98–113, 2025.
[41] A. Vasquez, Y. Lee, and P. Narang, “Context-Aware QR Code Security using Behavioral Risk Analytics,” Proc. Int. Conf. Mobile Cyber Intelligence, Springer LNCS, vol. 13467, pp. 112–124, 2025.
[42] L. Chen and M. Nguyen, “Augmented Reality Overlays for QR Code Security Enhancement,” J. Cybersecurity Technology, vol. 12, no. 3, pp. 199–214, 2025.
[43] T. Delgado et al., “Differentiating Real vs Malicious QR Codes in Public Spaces Using Geospatial Trust Anchors,” Proc. Springer LNCS, vol. 13489, pp. 133–145, 2025.
[44] T. Anderson et al., “Integrating Global Threat Intelligence with Edge AI for Real-Time QR Code Threat Detection,” IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3550–3562, 2025.
[45] K. Zhao and H. Tanaka, “Post-Quantum Cryptography for QR Code Security: Implementing Lattice-Based Signatures,” Quantum Computing and Cryptography, vol. 5, no. 1, pp. 88–102, 2025.
[46] Brooks and N. Fedorov, “Decentralized Identity for Web3 QR Protocols: A Post-Quantum Future,” J. Web Authentication & Blockchain Systems, vol. 3, no. 2, pp. 56–71, 2025.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)