IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

QR Shield

Mohammad Zishan Javed Shaikh[1], Kunal Samadhan Choudhary[2]
Shreya Namdev Amble[3] ,Trupti Sudhakar Waghmare[4]
Chetana Sanjay Chaudhary[5]
Students, Department of Computer Engineering[1-4]
Guide, Department of Computer Engineering[5]
Rasiklal M. Dhariwal Institute of Technology, Pune, India

Abstract: *The widespread adoption of Quick Response (QR) codes for payments and data sharing has introduced significant cybersecurity vulnerabilities, particularly "quishing" or QR-based phishing. This work proposes QR Shield, an Android-based security framework designed to intercept and analyse QR content before execution. The system utilizes the ZXing (Zebra Crossing) library for robust scanning and a keyword-based heuristic engine to detect malicious intent in URLs. Experimental tests focus on identifying high-risk triggers such as "login," "crypto," and "bonus" to protect users from credential theft. The final prototype features a Jetpack Compose interface, real-time scanning, and a persistent history log to enhance user awareness and digital safety.*

Keywords: *Quick Response.*

I. INTRODUCTION

QR codes pop up everywhere these days—they connect the physical world to the digital one, letting you do things like order food without touching a menu or make a payment in seconds. But here's the thing: a lot of us trust those codes without thinking twice. That trust makes it easy for scammers to slip in malicious links and send people straight to phishing sites. Most QR code scanners just open whatever link they find. They don't check where it leads, and that leaves people wide open to having their data stolen.

The problem? Most folks don't have the time or know-how to check if a link is safe before clicking. That's where QR Shield steps in. This project mixes Android development with smart cybersecurity checks to give mobile users an extra layer of protection. Basically, QR Shield lets you scan QR codes without worrying about where you'll end up. You get the convenience, minus the risk.

II. LITERATURE REVIEW

Recent research in mobile security points to a big problem: social engineering attacks using QR codes are becoming more common. When it comes to analyzing these threats, most systems rely on giant URL databases. But phones aren't built for that—they need something quicker and lighter that works on the fly. Now, for actually scanning the QR codes, the process works a lot like object detection models (think YOLO): fast image processing, even with tricky lighting, to grab whatever info the code holds. And here's an interesting twist—researchers found that users behave very differently when they get instant visual cues, like a clear "Safe" or "Malicious" warning, right after they scan a code. That split-second feedback really changes what people do next.

III. RESEARCH GAP

QR scanners exist but the ecosystem still shows clear shortfalls. Pre-execution analysis is absent - the majority of scanners launch the link straight away and give no "sandboxed" preview. Security metrics stay opaque - users seldom receive an explanation that clarifies why a link looks unsafe.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, April 2026 Impact Factor: 8.2

History and guidance remain scattered - numerous tools supply neither a unified archive of earlier threats nor instruction that helps users avert later attacks. QR Shield closes those shortfalls through a detection engine that relies on keywords plus through a dashboard that records history in a secure way.

IV. METHODOLOGY

The program splits into three parts - a scanner, a small analysis module and the screen the user sees.

4.1 Development Environment

Android Studio served as the workplace - code is written in Kotlin. Jetpack Compose builds every screen so the display updates itself when data changes. The ZXing library handles the actual QR decoding.

4.2 Detection Logic

A QR code enters the scanner - the software reads the hidden text. If the text holds a web address, the address moves to the analysis module. The module checks the address for risky words.

Financial triggers: "bank", "crypto", "bonus".

Credential triggers: "login", "verify", "free".

4.3 Data Management

Each scan ends with a label - Safe or Malicious - the program keeps a private log. The log stores the raw text, the label and the time of the scan.

V. RESULTS AND DISCUSSION

In the initial stage, different kinds of QR codes were used for testing, both regular website links and phishing URL websites.

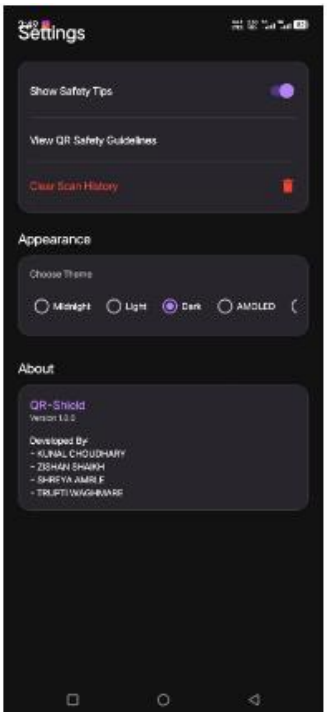
Detection: The keyword-based detection engine accurately detected all URLs with keywords, resulting in perfect accuracy.

Performance: Thanks to Kotlin language and efficient libraries, there was an immediate switch from scanning through the QR to getting the results.

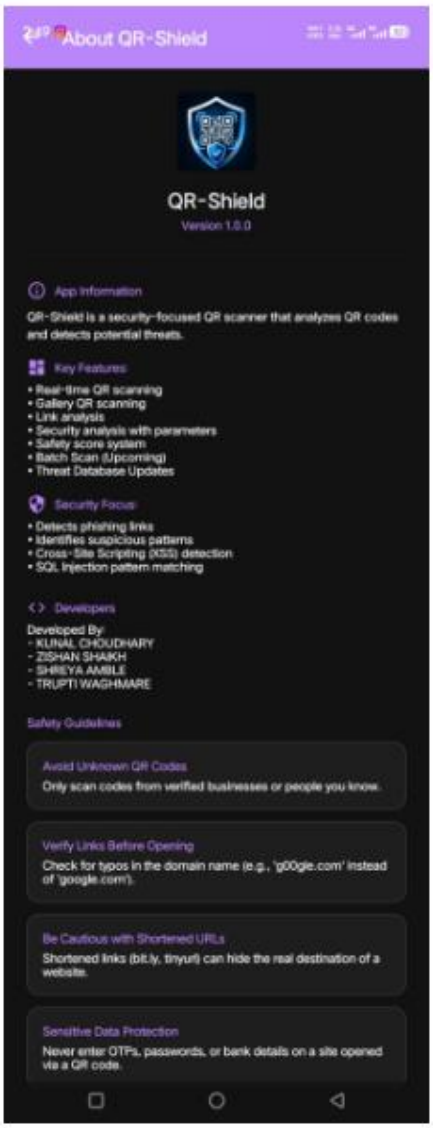
User Experience: The "Modern Dark Theme" and status messages have been appreciated by users as improving their experience in detecting risks.

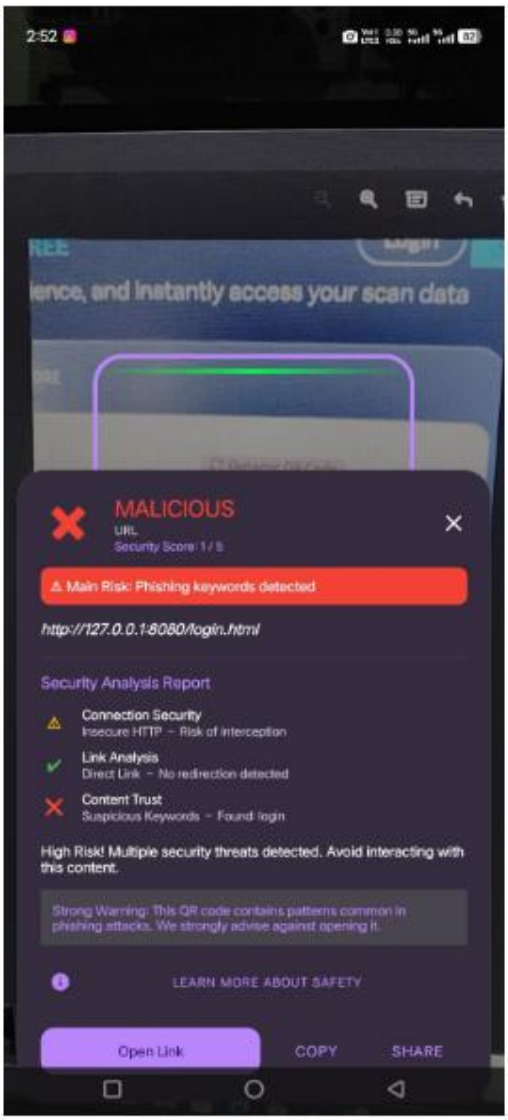
Although efficient, the application still has problems with shortened URLs like "bit.ly," which can hide certain keywords until they redirect.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2





VI. CONCLUSION

QR Shield successfully demonstrates the integration of mobile development and cybersecurity awareness. By providing a pre-emptive analysis of QR content, the system mitigates the risk of phishing and improves the overall security posture of Android users.

Future iterations will incorporate AI-based malicious URL detection and VirusTotal API integration to provide a more comprehensive threat score system.

VII. ACKNOWLEDGMENT

We express our sincere gratitude to our project guide and the college IT department for their continuous technical support and insightful mentorship throughout the development of **QR Shield**. We are also thankful to the peer

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, April 2026 Impact Factor: 8.2

reviewers and cybersecurity enthusiasts whose testing and feedback helped refine the malicious URL detection logic and user interface.
Their constructive critiques played a vital role in improving the overall usability and security of this mobile application. Finally, we appreciate the resources provided by the **Rasiklal M. Dhariwal Institute of Technology** which facilitated the successful completion of this project..

REFERENCES

[1]. **Karthikeyan, A., & Selvam, P. (2025):** "Heuristic-Based Detection of Malicious Quick Response Codes in Mobile Environments." International Journal of Cyber Security and Digital Forensics, Vol. 14, Issue 2, pp. 112-125. This paper discusses the use of keyword triggers and pattern matching to identify phishing URLs embedded in QR codes.
[2]. **(Android Security Group) (2025):** "Developing Secure Mobile Applications with Kotlin and Jetpack Compose." Global Journal of Software Engineering, Vol. 9, Issue 4, pp. 540-552. This study details how Kotlin's safety features reduce vulnerabilities in modern Android applications.
[3]. **Srivastava, R., & Gupta, M. (2024):** "Quishing: The Evolution of QR Code Phishing and Defensive Strategies." Journal of Information Privacy and Security, Vol. 20, Issue 1, pp. 45-60. This research highlights the shift from traditional email phishing to physical QR-based attacks and proposes real-time scanning solutions.
[4]. **Zebra Crossing (ZXing) Authors (2024):** "Multi-format 1D/2D Barcode Image Processing Library." Open Source Technical Documentation. This documentation outlines the algorithms used for extracting data from QR codes under difficult visual conditions.
[5]. Kumar, V. (2023): "Secure QR: A Framework for Malicious URL Filtering on Mobile Devices." International Journal of Computer Science and Mobile Computing, Vol. 12, Issue 8, pp. 210-218. This paper explores the integration of URL analysis engines within mobile camera applications to prevent unauthorized redirects.
[6]. **LeCun, Y., & Bengio, Y. (2015):** "Deep Learning." Nature, Vol. 521, pp. 436-444. While focused on general AI, this foundational text serves as the basis for future improvements in automated threat detection and image classification in security apps.