Development, Evaluation, and Implementation of SEQR – a Usable Secure QR Code Scanner

Mattia Mossano*
SECUSO
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
mattia.mossano@student.kit.edu

Maxime Fabian Veit*
SECUSO
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
maxime.veit@kit.edu

Tobias Länge
SECUSO
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
tobias.laenge@kit.edu

Benjamin Maximilian Berens
SECUSO
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
benjamin.berens@kit.edu

Filipo Sharevski
College of Computing and Digital Media
DePaul University
Chicago, Illinois, USA
fsharevs@depaul.edu

Melanie Volkamer
SECUSO
Karlsruhe Institute of Technology
Karlsruhe, Germany
melanie.volkamer@kit.edu

Abstract

QR codes are widely used, but can become the vector of phishing attacks (QRishing). To support users, we systematically developed a usable secure QR code scanner, SEQR (*Security Enhanced QR code scanner*). We based the SEQR's design on two systematic reviews: (i) of academic literature (2015–2025), identifying 96 papers on QRishing, and (ii) of the MITRE ATT&CK® Mobile repository, finding 36 QRishing techniques. From these two sources, we categorized 60 potential attacks, and divided them between those that SEQR addresses only at the technology level, and those where SEQR involves the users in the decision. We evaluated SEQR effectiveness in thwarting attacks in a between-subjects online study ($n = 556$), where SEQR achieved 93.35% correct answers, compared to 75.24% for the Apple iOS QR code scanner and 65.11% for the Samsung Android QR code scanner. We implemented SEQR as an open source Android application, available on GitHub.

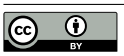
CCS Concepts

• **Security and privacy** → **Phishing**; **Usability in security and privacy**; **Mobile platform security**; **Human and societal aspects of security and privacy**; • **General and reference** → *Empirical studies*; • **Human-centered computing** → *User interface programming*.

Keywords

Security, Mobile devices: Phones/Tablets, Artifact or System, Usability Study

*Both authors contributed equally to this research.



CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3793213

ACM Reference Format:
Mattia Mossano, Maxime Fabian Veit, Tobias Länge, Benjamin Maximilian Berens, Filipo Sharevski, and Melanie Volkamer. 2026. Development, Evaluation, and Implementation of SEQR – a Usable Secure QR Code Scanner. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 33 pages. https://doi.org/10.1145/3772318.3793213

1 Introduction

Quick Response (QR) codes are two-dimensional barcodes invented in 1994 by Masahiko Hara (as per Denso Wave [26]). Aimed for the manufacturing industry, QR codes became user-approachable in 2017, when iOS 11 [8] and Android 8 [51] integrated a QR code scanner in their camera app, seeing further promulgation since the COVID-19 pandemic normalized their use, especially in the US (see Ricson [29]). While the approachability by users meant quick exchange of information such as URLs and phone numbers without the need of memorization, it also opened the possibility of misusing the QR codes for nefarious purposes.

For example, the Federal Bureau of Investigation (FBI) [30] reports that QR codes are increasingly used as phishing vectors. The goals of phishing through QR codes, or *QRishing* (so named by Vidas et al. [167]), are the same as other phishing vectors, e.g., emails (stated in Proofpoint [131]): account compromise, malware dissemination, direct financial appropriation, and IT systems disruption. Yet, distinctly from other forms of phishing, QRishing has some characteristics that set it apart, e.g., QR codes being only machine-readable (stated in, e.g., [9, 61, 172]).

Our *goal* in this paper is to address the growing QRishing threat by designing, developing and evaluating a usable secure QR code scanner, called *SEQR* (Security Enhanced QR code scanner). SEQR's threat model is fully elaborated in Section 9.8. To comprehensively inform SEQR's design, we surveyed known QRishing techniques from both industry and academia, ensuring protection covering the entire attack landscape. We conducted two separate systematic reviews, one of academic literature and one of the MITRE ATT&CK® Mobile techniques (both described in Section 4).

We then categorized the results in 10 categories, encompasing 60 attacks. The next step was to determine which techniques were addressable through a QR code scanner, be it through UI and user involvement, or at the technology level. We then developed SEQR's functionalities and UI to thwart the addressable techniques, basing our work on the results from related work, e.g., [16, 118, 168].

We evaluated SEQR in a between-subjects, online user study with $n = 556$ participants from the US, comparing our approach with the camera QR code scanners in Apple iOS and Samsung Android. Our results show that SEQR users have a significantly higher rate of correct answers (94.24%) than either Apple iOS users (68.94%) and Samsung Android ones (55.52%). Thus, SEQR is very promising towards reducing the QRishing risk for its users.

Based on participants' feedback, we added additional features to SEQR, published it as an open source Android app with the Privacy Friendly QR Code Scanner used as basis for its implementation (covered in Section 7), and made it available on GitHub[1] under license GPLv3. SEQR 2.0 was then evaluated in a between-subjects online user study with $n = 417$ participants from the US (covered in Section 8), comparing our approach with the Privacy Friendly QR Scanner. Our results show that SEQR 2.0 users have a significantly higher rate of correct answers (94.14%) than the Privacy Friendly QR Scanner users (76.26%). Thus, our results confirm that SEQR 2.0 is still a more promising solution than its baseline app towards reducing the QRishing risk for its users.

In short, our **contributions** are four-fold:

(1) We categorized QRishing attacks through two systematic reviews of QRishing techniques, one from the academic literature (2015-2025), and one of the mobile techniques in MITRE ATT&CK®;
(2) We developed SEQR, a usable secure QR code scanner, and evaluated it against the default scanner of both iOS and Samsung Android;
(3) With the corresponding user study we also confirmed the results from Berens et al. [16] that users are better supported through both awareness measures and support tools, rather than support tools alone;
(4) We improved SEQR based on the user study results and published the app as an open-source Android app on GitHub, and evaluated it against the base app used for its implementation.

2 Related Work

This section covers the related work highlighting the gaps in user protections against QRishing and the corresponding novelty contributions incorporated in SEQR. While users are increasingly attuned to the possibility of phishing through emails and SMS texts, doing so through QR codes is a threat yet to be incorporated in the users' mental models, indicated by the rate of QRishing attacks success (2.1). An illustrative case of the relevance of QRishing is its increased use to force users to switch from their corporate endpoint to a mobile device, bypassing multi-factor authentication and hijacking cloud identities [31]. As this and similar cases show a direct connection between QR code misuse and phishing susceptibility, we contrast the current phishing systematizations with the one we propose on QRishing to show that QR codes have been omitted from a comprehensive consideration, despite their relevance within the phishing landscape (2.2).

When users are in question relative to phishing, the focus shifts to anti-phishing measures to help them "spot" a phish (2.3). We review non-QRishing specific anti-phishing approaches for two reasons: (i) show that they are designed with email or SMS phishing in mind, building on the user experience with email clients and SMS applications that are neither present nor applicable in the interaction with QR codes (e.g., synchronous scan and tap instead of an asynchronous email or text message, absence of warnings, URL highlighting); and (ii) highlight concepts that resonate with the specific QR interaction but that few works have considered as potential solutions. We review those "few" (2.4) to show that no work so far: (i) has been built on a comprehensive taxonomy of QRishing attacks; (ii) considers users' familiarity with usable security warnings to enhance a QR code scanner; (iii) has been evaluated with actual users; (iv) has demonstrated the improvements of the proposed user support against the default smartphone OS QR scanners used for general purpose QR code scanning.

2.1 Users' Absence of QRishing Awareness

All the works in this section highlight the absence of QRishing awareness among the users, with Sharevski et al. [147], in particular, highlighting the perceived lack of user support from the existing QR code scanners. Our work builds on them by addressing the lack of awareness alongside providing user support in the form of a QR code scanner. The works are shown in ascending chronological order to underline the persistence of the QRishing awareness problem, even after more than ten years of research.

Vidas et al [167] conducted two studies, the first with fliers, and the other as a two weeks surveillance study, both on and around the Carnegie Mellon University campus. The first study revealed that 61% of fliers led to users visiting the encoded URL. The second study showed that, over the two weeks surveillance period, 85% that scanned the QR codes visited the encoded URL.

Krombholz et al. [63] ran an online user study over two months. The results showed that standalone QR codes attracted the highest number of scans and visits. Most participants scanned the QR codes out of curiosity or boredom. Regarding URL inspection, 50% of participants reported checking it before visitation.

Sharevski et al. [146] designed a field study incorporating malicious QR codes in legitimate-looking Centers of Disease Control posters. They found that 67% of their participants revealed their Google and Facebook credentials when asked during an optional (and malicious) sign-up step after scanning the QR codes.

Sharevski et al. [147] followed up on their work with a naturalistic study where they placed self-created posters with QR codes in public places around Chicago. The posters attracted 42 participants, all of whom scanned the QR codes, and were interviewed on why they chose to do so. The results showed that 67% of the participants visited the URLs encoded in the QR codes without inspecting them first, instead basing their choice on the posters layout. Further, 19% of the participants indicated limitations in the QR code scanner interface as the reason for not inspecting the URL.

Bekavac et al. [13] ran a user study aimed at evaluating if secure-by-design QR codes, e.g., with prominent graphical elements, are

[1] https://github.com/SecUSo/SEQR-CHI-2026 (also contains a video tour of SEQR 2.0)

sufficient to increase the rate of QRishing detection. They designed four posters, and placed them around their university and at a public event, recruiting 27 participants. Their results showed that standard QR codes raised no suspicion in the participants, with 13 out of 27 participants not checking the URL before visiting it.

Kowalewski et al. [61] explored the users' susceptibility to QR-code based scams through two user studies. The first study investigated the effectiveness of shopping related QRishing attacks delivered through emails, finding that the malicious QR code was visited by 53.2% of the participants. The second study was on online payments. An email for a donation requested payments from the users as either text, i.e., asking to enter the bank details manually, or having the bank details encoded in a QR code. While 54% of participants in the text condition proceeded with the payment, 87% of participants in the QR code condition did the same. Both studies also showed that only 33% of the participants heard of QRishing.

2.2 Systematic Reviews of Phishing Attacks

Several systematic literature reviews of phishing attacks exist.

Abdillah et al. [2] reviewed works published between 2010 and 2020. They created a categorization that divided the phishing techniques among "techniques," "datasets," "performance evaluation," and "phishing types." Arshad et al. [10] explored phishing techniques and their mitigation in a literature review of 20 previous works, identifying 12 phishing techniques and six mitigation. They also determined that the most crucial phishing techniques are email spoofing, spear phishing, and phone phishing.

Franz et al. [36] collected 2109 papers on user-oriented anti-phishing interventions. They identified 64 works as relevant, systematizing them in a taxonomy. Namely, they divided the interventions among "education," "training," "awareness-raising," and "design," concluding by proposing the use of digital nudges as interventions. Naqvi et al. [122] collected works published between 2018 and 2023. They categorized the phishing techniques based on the mitigation strategies used, e.g., machine learning, neural networks, human centric, etc. Veit et al. [165] systematized every known deception techniques in the email context. They evaluated eight email clients against the 23 deception techniques they found, reporting which ones are exposed to which deception techniques.

To the best of our knowledge, no similar recent systematic work on QRishing exists. We found a series of works that included QRishing as one of the discussed techniques, e.g. [3, 5, 6, 44, 143, 149, 153, 174, 183]. Yet, none of them went into details, e.g., regarding strategies or unique QRishing techniques. Only two works, i.e., Krombholz et al. [62] and Yong et al. [181], specifically focused on QRishing strategies and techniques. However, they are both relative dated (2014 and 2019, respectively), and published pre-COVID-19, meaning that they were not written in a setting where QR codes are an everyday encounter. Therefore, we determined that a systematic review of QRishing techniques was needed to proceed in our work. We detail such systematic review in Section 4.

2.3 Anti-Phishing Approaches

Several lessons learned within the general anti-phishing context are relevant for QRishing, e.g., that awareness measures (2.3.1) and link-centric interventions (2.3.2) work better alongside one-another than on their own (2.3.3).

2.3.1 Phishing Awareness. The most common forms of awareness measures are *facts-and-advice training*, e.g., [123, 133, 176, 184], and *phishing games*, e.g., [20, 39, 79, 125, 150, 177], usually tailored to get increasingly complex to encompass for any evolving phishing techniques, show in, e.g., Jampen et al. [53]. *Phishing videos*, e.g., [1, 16, 38, 171, 179], are similarly useful in that they could be very short while still covering a lot of phishing content, allowing the use of animation to visualize concepts, as mentioned in [49, 155]. In our case, we determined that a short facts-and-advice measure in the form of a tutorial, entirely focused on being of support in the use of SEQR, was the best way to proceed. Although a video allows a faster fruition, a text-based tutorial over multiple pages allows the users to retrieve information without having to watch the whole video again or guessing at which time the information is shown.

2.3.2 Anti-Phishing Support Tools. Another anti-phishing approach is *support tools*, interventions that support users at the moment of exposure to a harmful URL. The intervention itself is either: a) *just-in-time*, i.e., at the moment of decision, b) *just-in-place*, i.e., right next to the link being hovered, or 3) both just-in-time and just-in-place. Anti-phishing support tools come in three variants: 1) *browser warnings*, e.g., Safe Browsing [42] or SmartScreen [77], blocking the user before accessing a malicious webpage in their browser; 2) *banner warnings*, e.g., in email clients such as Gmail [41] and Outlook [76], showing a banner to highlight potential issues in an email; and 3) *link-centric warnings*, such as those in [16, 118, 129, 169, 170], displaying next to or instead of the link about to be clicked.

Petelka et al. [129] conducted a comparative evaluation of these variants' effectiveness, showing that the link-centric warnings are the most effective at preventing users from clicking on phishing links in emails. The benefit of link-centric warnings over browser warnings and banner ones is two-fold: 1) they appear next to the users' attention focus, i.e., next to the link itself, and 2) they do not require clicking the link and accessing it in the browser. Given the results from Petelka et al. [129], we determined to develop a just-in-time and just-in-place intervention in the form of an extended dialog appearing when a QR code is scanned. Namely, we decided to adapt to the QR codes context the email link-centric solution from [16, 169, 170], as it was shown to significantly improve the phishing detection of the users.

2.3.3 Combination of Awareness Measures and Tool Support. Berens et al. [16] showed that the best approach to prevent phishing is neither awareness measures, nor anti-phishing support tools, but rather a combination of both. They showed that including a *tutorial*, shown once at the installation of the anti-phishing support tool and then available on-demand, significantly increased the phishing detection of users compared to using the tool on its own. Furthermore, they showed that combining the intervention (with a tutorial) and an awareness measure, obtained the best results. Thus, we included a tutorial in SEQR that includes both the functionality and a awareness measure focused on QRishing.

2.4 Anti-QRishing Approaches

Based on the works in the section, SEQR checks the legitimacy of a URL against the PhishTank database. Yao and Shin [180] developed the SafeQR scanner based on Google Safe Browsing and PhishTank, focusing on malicious apps and credentials stealing websites. SafeQR had two types of warnings: 1) known malicious links (in red), and 2) known malicious apps (in yellow). They evaluated SafeQR with three (now defunct) other Android QR scanners, showing that it performed the best.

Dudheria [28] conducted a broad comparison of 14 QR code scanners available on Google Play in 2017, revealing a large variance in functionality. Most notably, only two scanners resolved URL redirects, and several failed to display the URL at all. Based on this, Dudheria recommended displaying resolved URLs, requiring user confirmation before opening, and providing transparency on potentially rogue URLs. Latif et al. [65] developed an Android QR code scanner that verified encoded URLs through PhishTank. The scanner showed three generic banner warnings (phishing, suspicious, or safe), and displayed the URL with no color or explanation of the risk involved. No user study evaluation was conducted to determine its effectiveness.

Pawar et al. [127] approached the problem through machine learning by training a Recurrent Neural Network to detect malicious URLs within QR codes. Using a dataset sourced from PhishTank, URLhaus, and Alexa, they achieved promising results with their best-performing model (Bi-LSTM, 83.79% accuracy). Sabri et al. [140] developed an Android QR code scanner that displayed a generic warning about a malicious website based on a backed query check on a GitHub repository of known malicious URLs. The warning includes the text "*malicious website detected*" and the GitHub repository was not officially maintained akin to Google Safe Browsing or PhishTank. Rafsanjani et al. [132] developed an Android QR code phishing scanner, dubbed QsecR, using heuristics (trained on URLhaus and PhishTank) to infer the legitimacy of a URL. Based on its inferences, the app showed a score and classification of a URL, but it neither included explained how to interpret this score nor it was subjected to user study evaluation.

3 Background

In this section, we introduce the general URL obfuscation techniques identified by Veit et al. [165] with a focus on the QRishing relevant attacks, i.e. those attacks SEQR aims to address (3.1). We also briefly describe the TORPEDO tool and its functionalities (3.2), as it is used as basis for SEQR's risk assessment and UI.

3.1 URL Obfuscation Techniques

URL obfuscation techniques are techniques that make it harder to correctly discern a URL destination by exploiting the users' lack of understanding of the structure of URLs (see Figure 1 for more information on the structure). Several studies (e.g., [4, 7, 27, 186]) show that users are unaware that (usually) the only part relevant to determine where a URL would take them is the *registrable domain*. Veit et al. [165] covers URL obfuscation techniques as part of a SoK on email phishing. They do not cover QRishing, but because URLs do not change depending on the context of use, the URL obfuscation techniques are still applicable. Hence, we briefly describe them in Table 1, and explain in Section 5 how SEQR addresses them.

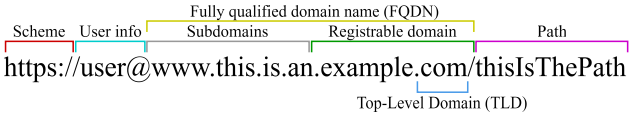


Figure 1: Structure of a URL.

3.2 TORPEDO

The TORPEDO tool, presented in [16, 169, 170], is an anti-phishing tool that provides link-centric warnings when a user hovers over a link in an email. TORPEDO assumes that known phishing emails

Table 1: URL obfuscation techniques from Veit et al. [165].

Attack	Short Description	Example(s)
IP based	IP address as registered domain	https://142.250.74.206
Obfuscate	Random characters as registered domain	https://aldslkskmskoewlc.at
Domain extension	Extend legitimate domain with related words	https://secure-google.com
Mangle	Very small deviations from legitimate domain obfuscation	https://mircosoft.com https://arnazon.com https://youutube.com
Mislead	Legitimate domain as subdomain or path	https://google.com.connection.io https://connection.io/google.com
Exceedingly long	Hide registered domain outside of rendering area	-
Hexadecimal encoding	Encode registered domain in hexadecimals	https://google.de%6D%61%6C%2E%69%73
Homographic spoofing	Abuse similar characters from different alphabets	https://paypal.com (Cyrillic y)
URL shortener	Abuse shortening service to hide URL	-
Redirect	Hide URL in redirection URL path	https://google.com/url?sa=t&source=web&rct=j&url=https://malicious-page.com
Different Top-Level-Domain	Entity name with a different top level domain	https://google.io

are blocked by automatic detection tools such as email filters, and focuses on supporting users in checking the URLs of those emails that reach their inbox. TORPEDO employs a series of tooltips that differ in coloration and content depending on the estimated risk-level of the link being hovered. In other words, TORPEDO's warnings are just-in-time and just-in-place. TORPEDO classifies each URL as one of the three distinct risk levels described in Table 2.

A visualization of the decision process is in Figure 2. TORPEDO can be set-up to automatically resolve short URLs and redirects[2]. When first installed, TORPEDO shows a tutorial explaining its functionality, as well as raising awareness for relevant phishing tricks, and the same tutorial is available on demand from the settings.

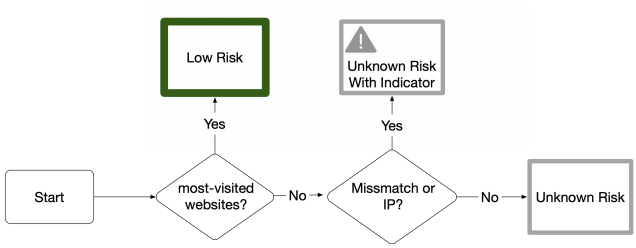


Figure 2: Decision process of TORPEDO, per Berens et al. [16].

4 QRishing Attacks Collection

As mentioned in Section 2.2, at present, to the best of our knowledge, there is no systematic categorization of QRishing techniques. We first cover the methodologies we followed to generate such categorization in a shortened format (4.1, 4.2); the details necessary for replication are available in Appendix A. We then describe the categorization in details (4.3) to provide the security community with a systematization that can be used as a basis for developing awareness material, tools, and usable security interventions. Lastly, we discuss those parts of the categorization that informed the SEQR's design and development (4.4).

4.1 Systematic Literature Review

We followed the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) standard described in Page et al. [126]. We identified repositories known for publishing works on QR codes (namely ACM Digital Library, IEEE Xplore, and Elsevier Science Direct) and constrained the search to the interval 2015-2025. We queried each database with the query "QR AND phishing," and identified a total of 478 entries. After several rounds of selection, we identified 72 relevant entries, expanded by a backward search with further 24 entries, leaving us with 96 entries in total. Further details are in Appendix A.

[2]Note, this is not mentioned in Figure 2 as it was not relevant for the research conducted by Berens et al. [16]

4.2 MITRE ATT&CK® Techniques Review

MITRE ATT&CK® (described in MITRE [92]) is a repository of adversarial behaviors divided into two parts: *Enterprise*, covering attacks against businesses, and *Mobile*, focused on mobile devices. We focused on the Mobile part, accessible at MITRE [116] and containing 121 entries at the time of our investigation (June 2025). We opened each entry one at a time and, after several rounds of screening, we identified 36 relevant entries (details in Appendix A).

4.3 QR Code Attack Categorization

In this section, we present and describe a categorization aimed at structuring the results of both systematic reviews. The categorization is shown in Figure 3.

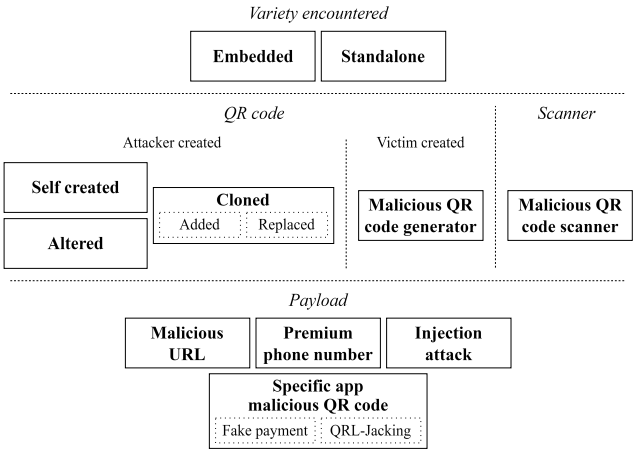


Figure 3: Visual representation of our results.

Two varieties of QR codes exist, be them legitimate or malicious:

- ***Embedded*** in a pretext [3, 13, 18, 33–35, 47, 48, 54, 62, 63, 72, 75, 127, 130, 142, 146, 152, 158, 162], e.g., a poster, a leaflet, an email, a social media post, etc.

Table 2: Description of TORPEDO risk levels and dialog windows.

Risk Level	Conditions	Description
Low Risk	URL among most-visited German websites list	Tooltip with green border
		No time delay before opening URL
Unknown Risk with Indicator	URL among most-visited German websites list	Tooltip with gray border
	Mismatch between URL and link-text / URL is IP address	Three seconds time delay before opening URL Triangle with exclamation mark symbol
Unknown Risk	URL not rated as either of the other two levels	Tooltip with gray border
		Three seconds time delay before opening URL

- ***Standalone***, i.e., without any pretext [11, 59, 75, 127, 130, 140, 167, 183], e.g., a sticker, on a sheet of paper, etc.

The difference between the varieties is the presence or absence of a pretext surrounding the QR code, which leverages two different human factors to convince a victim to scan the QR code. Sharevski et al. [147] shows that a pretext leverages one or more of the persuasion principles (detailed in Ferreira et al. [32]) to attract the attention of the victims and convince them to interact with the QR code. Standalone QR codes, instead, leverage the curiosity of the victims, as shown in, e.g., [63, 145]. Namely, what convinces the victims to interact with the QR code is the absence of a pretext, and the curiosity to discover what is behind.

Irrespective of the pretext, once an interaction happens, adversaries consider two aspects: 1) QR code, and 2) QR code scanner.

Regarding the QR code itself, there are two possible origins of a malicious QR code: *attacker created*, and *victim created*. In case of an attacker created QR code, this can happen in one of three ways:

- The attackers generate and disseminate a ***Self-created*** QR code (and pretext, if there is one) [11, 13, 59, 62, 121, 130, 152, 156]. The attackers generate a QR code (and a pretext) of their choosing, without relying on any existing legitimate version.
- The attackers generate and disseminate a ***Cloned*** pretext [5, 13, 18, 59] by modifying an existing, legitimate pretext to contain a malicious QR code. If the pretext already contained a QR code, this is ***Replaced*** with the attackers' created QR code [13, 46, 62, 130, 139, 143, 153, 160, 162, 174, 175, 181, 183]. Note, if there was no QR code, the attackers' QR code would be ***Added*** to the cloned pretext, although this was not mentioned in any of the papers we found.
- The attackers find a preexisting QR code to be ***Altered***. This can be done in three of ways. One way is modifying the modules of a printed QR code with a pen so that, when scanned, it would do something different than originally intended, e.g., directing the victims to a different URL [11, 48, 52, 58, 59, 62, 63, 127, 139, 148, 153, 164, 174, 175, 181]. Another way to alter a QR code is by adding a different, smaller QR code inside the original one (also known as *QR-in-QR)* [21, 25, 48, 63, 132, 153, 157, 164, 181], so that, when scanned, the priority would be given to the smaller, attacker created one. A third way is using more technically complex attacks involving light manipulation, e.g., by using infrared light to modify the modules of the original, legitimate QR code [187], as infrared light is not visible by humans, but a camera would react to them and scan the altered QR code.

We found only one case of victim-generated QR code, i.e., the victims generate and disseminate malicious QR codes through *Malicious QR code generators* [11, 28, 37, 127, 132, 140]. These generators are programmed to consistently substitute whatever the victims want to encode with content of the attackers' choosing. For example, a malicious generator might encode a malicious URL of the attackers choosing instead of the URL intended by the victims, who would then spread the malicious QR code for the attackers. A malicious QR code from a malicious generator leverages the victims' credibility to gain legitimacy, making it more believable.

Regarding the QR code scanner, there is only a single type of it, i.e., the attackers develop and disseminate a ***Malicious QR code scanner***. Such malicious QR code scanner can deliver three types of attack: 1) a trojan [44, 57, 83, 85–89, 93, 94, 97–99, 101, 103, 108, 117, 132, 148, 149], 2) a permission escalation attack [12, 33, 46, 48, 58, 59, 80, 81, 84, 100, 102, 105, 109–114, 132, 137, 151], and 3) redirect any scanned QR code containing a URL to a different URL of the attackers' choosing [33, 34, 37, 58, 132].

Each aspect, i.e., the QR code or the QR code scanner, can deliver one of four potential payloads, i.e., the part of an attack that execute a malicious action:

- A ***Malicious URL*** [3, 5, 13, 14, 17, 18, 28, 33–35, 47, 50, 57–59, 62, 64, 67, 69, 70, 82, 91, 106, 107, 121, 124, 130, 138–140, 156–158, 160, 164, 167, 178, 180], used to either lead: 1) to a webpage that harvests the credentials of the victims, or 2) to disseminate malware. While these papers consider various URL obfuscation techniques, we refer the reader to the more exhaustive list of obfuscation techniques from Veit et al. [165], described in Section 3.1.
- A ***Premium phone number*** [67, 84, 114, 180] charging victims when contacted through phone calls or SMS.
- An ***Injection attack*** [11, 12, 25, 33–35, 52, 55, 58, 59, 62, 63, 90, 95, 96, 104, 115, 124, 132, 135, 143, 148, 157, 160, 172, 175, 180] encoded in the QR code itself. Once scanned, the normal operation of the device used to scan the malicious QR code is disrupted, and the arbitrary code injected by the attacker is run. Such code allows the attackers to steal the victims' information, by commanding the infected device to share with the attackers, e.g., contact lists, images, documents, etc.
- A ***Specific app malicious QR code*** [5, 19, 35, 56, 63, 71, 73, 124, 146, 153, 157, 160, 161, 163, 173, 182, 185], i.e., a malicious QR code developed to interact with a specific app, or type of apps. In the literature, there are two types of such QR codes: 1) *Fake payment*, and 2) *QRL-Jacking*. The first comprise QR codes created for payment (e.g., PayPal), that, once scanned by the victim, send the amount transferred to the attackers account, who can then withdraw it. The second type targets those apps that use QR codes to authenticate the users, i.e., QR code login (e.g., WhatsApp Web). Once the victims scan the QR code, they transfer the necessary credentials to hijack the online session of the app to the attackers, who can then access it and/or take it over.

4.4 Implications for SEQR Development

Several techniques are irrelevant to the development of SEQR, namely: a QR code being embedded or standalone, whether a QR code is self-created, altered, or added to a cloned pretext, and whether a QR code was generated through a malicious generator. For a (honest) QR code scanner, it does not matter how the QR code got to the user, nor how it became malicious. Furthermore, for SEQR, the technique considering that a malicious QR code scanner is installed, is irrelevant. In addition, the payload "specific app malicious QR Code" is irrelevant too, as SEQR is not meant to replace the apps targeted by this group of attack techniques.

What is relevant for SEQR is to determine how to deal with the remaining three payloads. We will cover this in the next section.

5 SEQR 1.0 Description

This section describes how we designed SEQR 1.0 to address the three payloads identified in the previous section. Figure 4 shows an overview of these payloads and how they are addressed.

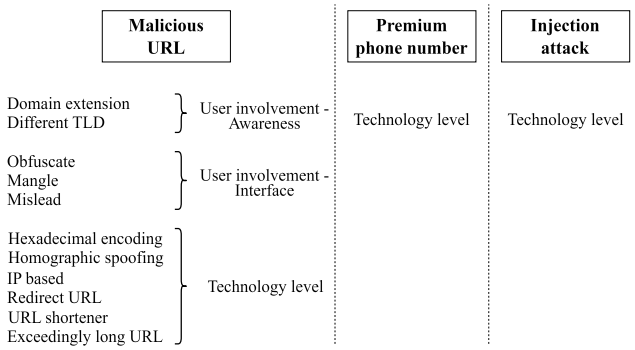


Figure 4: Overview of how SEQR address the relevant payloads described in Section 4.3.

Very briefly, SEQR 1.0 works as follows: once a QR code is scanned and analyzed, SEQR shows a dialog visualizing the URLs in a plain representation combining *domain-only* visualization and *kerning*-based text presentation. It distinguishes three *Risk levels* and has a *tutorial* to further support users in verifying a URL legitimacy. Depending on the risk level, *delay friction* is applied or not. The color of the dialog and the information provided in it also depends on the risk level (for some screenshots see Figure 6). The algorithm used by SEQR 1.0 is in Figure 5. In the following subsections, we describe the different features, how they address the payloads, and whether or not user involvement is required.

5.1 Risk Levels

SEQR 1.0 distinguished between three risk levels. Note, the general idea of risk levels is similar to the proposal from Berens et al. [16] while, due to the different context, we add one risk level (high risk) and remove one of theirs, as there was no need to keep it because not relevant in the QR code context (unknown risk with indicator[3]).

The different risk levels are determined by i) using the PhishTank database (see Cisco [22]), and ii) by querying the top 100 entries of the TRANCO list (described in Le Pochat [66]). Examples of the risk levels and the respective dialogs are in Figure 6.

High-risk. SEQR 1.0 queries the PhishTank database [22] for every URL scanned[4]. If the URL is labeled as "phishing," SEQR 1.0 blocked its opening, and showed a red-outlined warning noting that the URL was used in phishing attacks before (see Figure 6a).

Low-risk. If the URL is not in the PhishTank database and the registrable domain is part of the TRANCO top 100 entries, the URL is deemed Low-risk. The dialog has a green outline (see Figure 6b).

[3]SEQR 1.0 does not cover the TORPEDO's "Unknown-risk with Indicator," as there is no link mismatch in the QR code context and IP addresses are handled as text as we could not see any context in which a legitimate author would convert an IP address to a QR code. For more information on this risk level see Figure2 in the Background section

[4]We are aware that this comes with some privacy drawbacks. Therefore it is discussed in the limitation / future work section

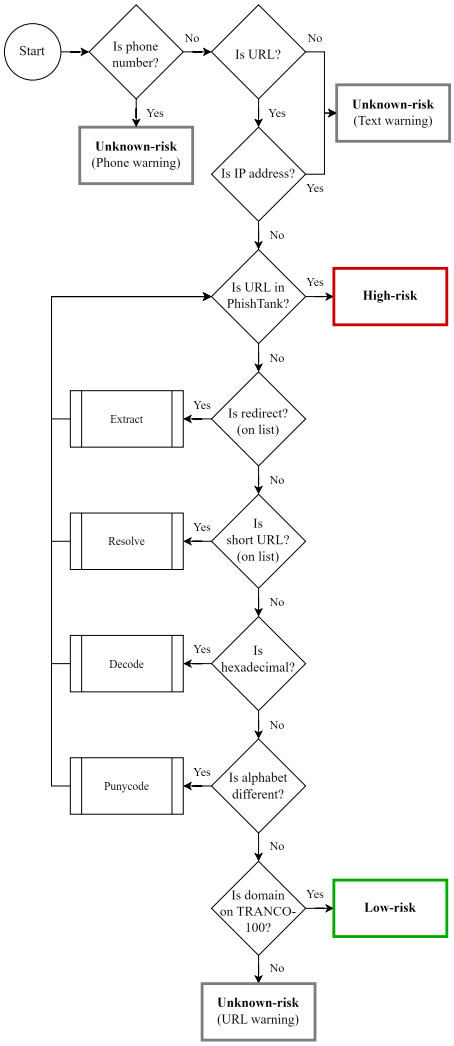


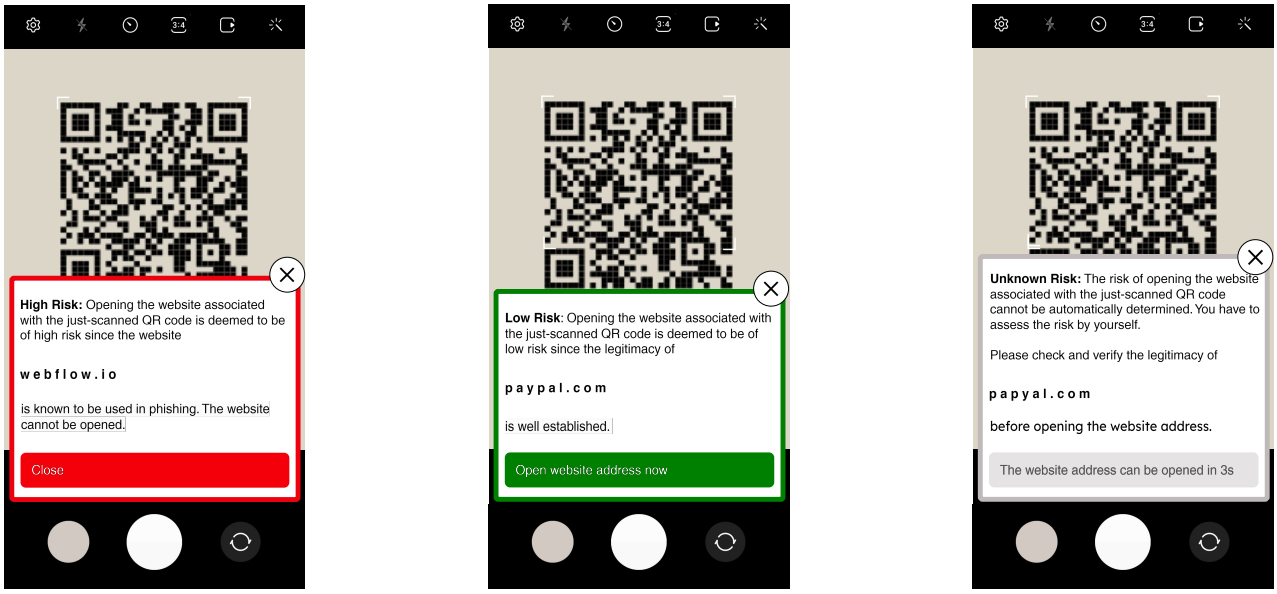
Figure 5: The decision algorithm of SEQR.

Unknown-risk. If the URL is not part of the PhishTank database, and the registrable domain is not on the TRANCO top 100 entries, the URL is deemed an "Unknown-risk," and the users are prompted to carefully inspect the URL themselves. The warning has a gray outline (see Figure 6c). To give users sufficient time to inspect the registrable domain, SEQR 1.0 introduces a 3-second *delay* when the "open website" button is disabled, as suggested in [16, 128].

5.2 URL Visualization

This subsection describes and justifies SEQR's URL visualization.

Domain-only. The most critical information for users to verify the legitimacy of an URL is the *registrable domain* (shown in Figure 1). While, for example, Apple iOS shows only the registrable domain but occasionally includes subdomains, SEQR 1.0 consistently displays only the registrable domain, similarly to other works (e.g., [16, 118, 119, 168]). SEQR 1.0's visualization prevents the distraction and the confusion caused by subdomains or misleading



(a) High-risk **(b) Low-risk** **(c) Unknown-risk**

Figure 6: Examples of SEQR 1.0's warnings for the three risk levels identified by the program as seen within the app interface.

prefixes/suffixes, such as those used in the *mislead* obfuscation techniques. Due to the limited display space on smartphones, restricting the URL visualization through the domain-only approach ensure that the registrable domain remains always visible. Thereby, the *exceedingly long URL* obfuscation techniques is automatically met. Furthermore, the URL visualization does not show HTTP/HTTPS. The choice is based on Kim et al. [60] showing that most phishing websites use HTTPS, and Mossano & Volkamer [120] showing that users commonly mistake HTTPS as indicating legitimacy.

Kerning. The *Mangle* technique exploits subtle character swaps or visually similar combinations. Berens et al. [16] show that increasing spacing between characters (i.e., kerning) significantly improves detection of mangled phishing URLs. Likewise, SEQR 1.0 displays registrable domains with increased letter spacing.

5.3 Tutorial

SEQR 1.0 includes a short, focused tutorial introducing SEQR 1.0 functionalities and risk levels (shown in Appendix B). Previous research has shown that both changing the UI itself, and including such a tutorial is beneficial to the users [16]. Furthermore, adding some general awareness information about the topic to the tutorial could increase the effectiveness of a tool support [16]. Thus, the tutorial was designed as a QRishing awareness measure through a short facts-and-advice walkthrough, covering the following URL obfuscation techniques: domain extension, different TLD, obfuscation, mangle, and mislead. The tutorial is shown the first time the users open the app, and it can be opened any time via the menu.

5.4 Attacks Addressed at the Technology Level

With the functionality described so far, users are supported in identifying QRishing using either domain extension, different TLD, obfuscation, mangle, or mislead.

SEQR 1.0 addressed the remaining attacks at the technology level (see Figure 4). This subsection describes how it does so.

5.4.1 Hexadecimal encoding and Homographic spoofing. These are decoded and presented to the user in the ASCII encoding. For homographic spoofing, SEQR 1.0 uses punycode, as commonly done in browsers, e.g., Google Chrome [40].

5.4.2 Redirect URL. SEQR 1.0 adopted a similar approach to [16, 118, 169]. Namely, SEQR 1.0 contained an expandable list of redirection services, describing the usual structure of the redirection URLs used by the services listed. SEQR 1.0 located the destination URL in the URL structure, and showed the resulting URL to the users instead of the original one.

5.4.3 URL shortener. SEQR 1.0 followed a solution introduced by [16, 118, 169], i.e., it resolved the short URL by loading the headers of the shortening service and following the HTTP 3xx server response. SEQR 1.0 then showed the destination URL to the users in one of its warnings, depending on the risk level assessed. This solution presents a privacy trade-off: the short URL would be considered as accessed, potentially leading to unwanted marketing profiling. As our focus was thwarting URL obfuscation techniques, we considered this an acceptable trade-off, but users can enable or disable it in the settings.

5.4.4 Premium phone number. SEQR 1.0 presented a warning with two main differences from the the default Unknown-risk warning: 1) the text mentioned that calling a phone number might cause

high phone bills, and 2) the button was changed to "Call the phone number." SEQR 1.0 did not open the dialer automatically, requiring the user to actively press the button to start a call. SEQR 1.0 did not check the users' contact list for the number because one injection attack steals the contact list of the user, as mentioned in Section 4.3.

5.4.5 IP based and Injection attack. In case the QR code was encoded with anything besides a URL or a phone number, e.g., an injection attack or regular text, SEQR 1.0 presented the content as plain text (shown in Figure 20c). This is also true for IP based URLs, which are treated as text. By presenting the content as plain text and offering to search online for it, we could thwart any unforeseen malicious content that might have come up in the future. Still, we clarified in the warning text that using an online search engine might potentially leak information through the online search, as the data was sent to the search provider. As some QR codes have huge data capacity (up to 7089 numerical characters) and may push the warning text outside of the rendering area of the screen, we limited the text to 100 characters (balancing security and usability in order not to dissuade users with long warnings).

5.4.6 Exceedingly long URL. Restricting the URL visualization by the domain-only approach ensures that the registrable domain remains always visible. Thereby, the exceedingly long URL obfuscation technique is automatically met.

6 Evaluation of SEQR 1.0

In this section, we describe the evaluation of SEQR 1.0. We start by stating our hypotheses and research question (6.1). We then describe our methodology (6.2), followed by the ethical considerations and the participants recruitment (6.3). The section closes with our results (6.4 and 6.5).

6.1 Research Questions and Hypotheses

We evaluated SEQR 1.0 by comparing it to the default QR code scanners of Apple iOS and Samsung Android OS respective to three research questions (RQ):

RQ$_1$. Are the users of SEQR 1.0 more effective than the users of the default Apple iOS QR code scanner in distinguishing phishing QR codes from legitimate ones?

RQ$_2$. Are the users of SEQR 1.0 more effective than the users of the default Samsung Android OS QR code scanner in distinguishing phishing QR codes from legitimate ones?

RQ$_3$. Are the users reading the tutorial more effective than the users not reading the tutorial in distinguishing phishing QR codes from legitimate ones?

Regarding RQ$_1$ and RQ$_2$, we based SEQR 1.0 on an approach proven effective in the email context, as shown in [16, 169]. Thus, we formulated two hypotheses:

H$_1$. SEQR 1.0 is more effective than the default Apple iOS QR code scanner with respect to users' ability to distinguish phishing QR codes from legitimate ones.

H$_2$. SEQR 1.0 is more effective than the default Samsung Android OS QR code scanner with respect to users' ability to distinguish phishing QR codes from legitimate ones.

Regarding RQ$_3$, Berens et al. [16] showed that combining a tutorial and an awareness measure significantly improved the effectiveness of an email link-centric approach. Yet, our awareness measure was integrated in the tutorial, not separated from it as in Berens et al. [16]. Thus, we decided not to formulate a hypothesis for RQ$_3$.

6.2 Methodology

In this section, we describe the methodology we followed to design and conduct our user study.

6.2.1 Study groups. We answer the RQs in Section 6.1 by dividing our participants into four groups.

iOS. This group saw a simulation of the QR code scanner interface used in iOS v. 17 on an iPhone SE.

Samsung. This group saw a simulation of the QR code scanner interface used in Android v. 13 on a Samsung 20 (One UI 5.1).

SEQR. This group saw a simulation of the QR code scanner interface used in Android v. 13 on a Samsung 20 (One UI 5.1), but showing SEQR 1.0's warnings. This group went through the SEQR 1.0's tutorial before their main task.

SEQR$_{no\text{-}tutorial}$. This group saw a simulation of the QR code scanner interface used in Android v. 13 on a Samsung 20 (One UI 5.1), but showing SEQR 1.0's warnings. This group did *not* go through SEQR 1.0's tutorial before their main task.

6.2.2 Study Structure. We cover here the structure of our user study, going through each of its five steps. A visual representation of the study structure is in Figure 7.

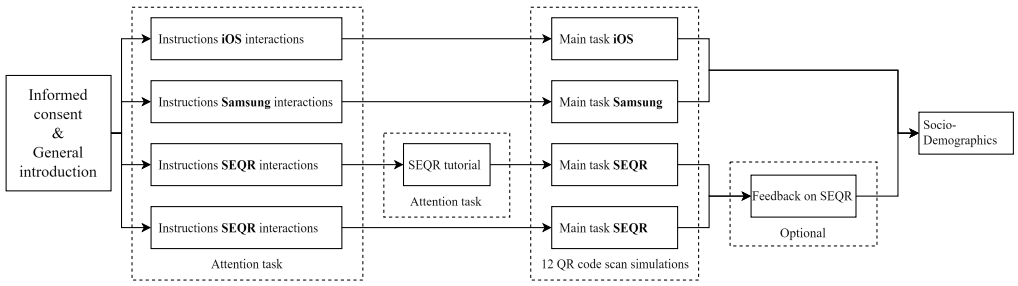


Figure 7: A left-to-right flow diagram of the study structure, showcasing each of the steps.

Informed consent & General introduction. At the beginning of the study, we informed the participants of the purpose of the study and the main task. We also informed the participants that all data would be collected anonymously, with no personally identifiable information recorded, and that they could withdraw from the study at any time, leading to the deletion of their data.

Group-Specific Instructions. Our participants then received a brief explanation on how to interact with the group-specific emulated QR code scanner, as shown in Figure 8. To confirm that the participants had properly read the instructions, we added two attention tasks, excluding those participants that failed them.

SEQR tutorial. At this step, those participants in the SEQR group saw the SEQR 1.0 tutorial and had to read it carefully. We then asked them three attention questions to verify that they engaged with the material, excluding those participants that failed them. The tutorial is shown in Appendix B.

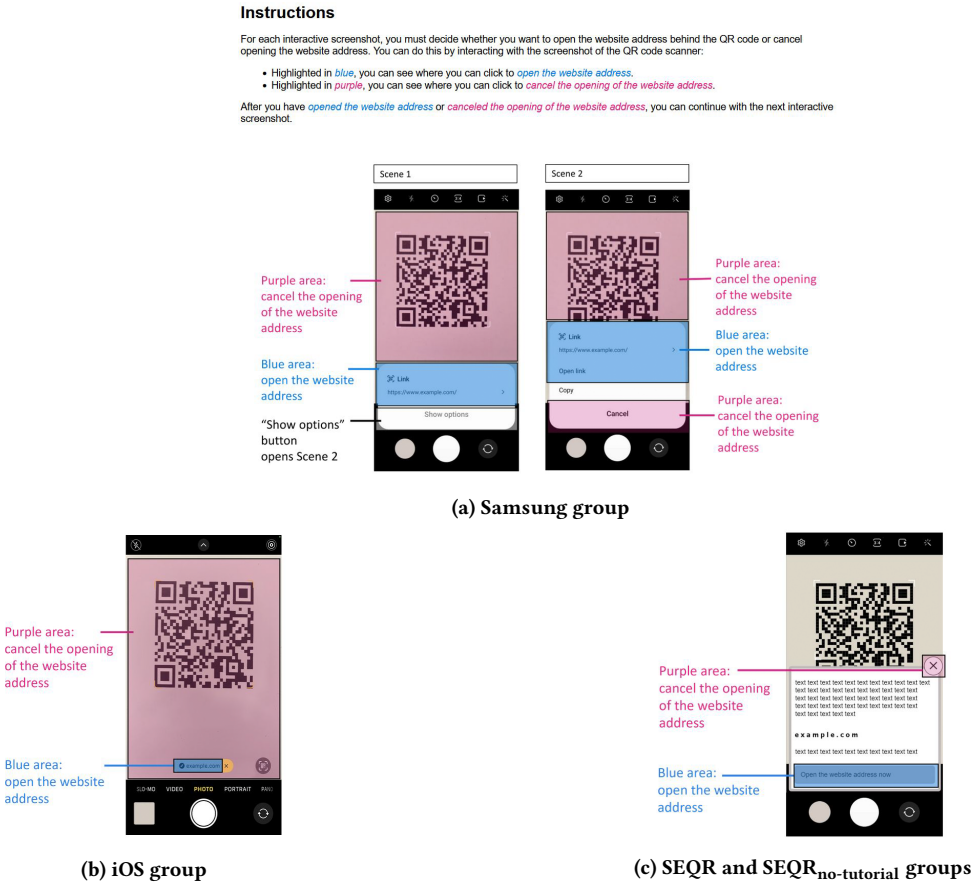
Group-specific Main task. As main task, we showed our participants 12 interactive screenshots of QR code scans. Each study group interacted with an emulated QR code scanner, fully integrated into the survey, and the screenshots were randomly shuffled for each participant. We instructed our participants to start the simulated scan by clicking on the "Scan QR code" button. They could then either *open the website*, classifying the QR code as legitimate (i.e., leading to a legitimate website), or *cancel the scan*, classifying the QR code as phishing (i.e., leading to a phishing website).

Feedback on SEQR. We asked the participants from both SEQR groups to provide feedback on the app usability and security in two open questions. Answering either (or both) questions was optional.

Socio-Demographics. We collected the participants' gender and age, which QR code scanner they use, the frequency of their QR code usage, and whether they were aware of QRishing.

6.2.3 QR Code Scenarios. As part of the main task, we defined scenarios depicting real-world phishing threats involving QR codes.

URL Obfuscation Techniques Used. We used those techniques where user involvement is needed, as identified in Section 5. Note, these are the same URL obfuscation techniques as Berens et al. [16] used in their study. We associated *two* URLs to each of the three URL obfuscation techniques (total of six phishing URLs and six legitimate URLs), to avoid influences from the specific URL chosen.



(a) Samsung group

(b) iOS group

(c) SEQR and SEQR$_{\text{no-tutorial}}$ groups

Figure 8: The instruction text shown to the participants. Figure a, b, c were mutually exclusive depending on the group.

Selection of URLs. We selected well-known organizations in the US (shown in Table 3) to avoid non-familiarity becoming a confounding factor. Berens et al. [16] conducted their study in Germany and used organizations widely recognized there, but both Albarki et al. [4] and Reynolds et al. [134] conducted similar research in the US. Thus, we selected five organizations common to both Albarki et al. and Reynolds et al.: Facebook, BBC, CNN, PayPal, and Google. We also included Bank of America, as banks are common targets for phishing attacks [166].

6.3 Ethics and Recruitment

This section covers the ethical considerations and the recruitment process of our study.

6.3.1 Ethics. The study was approved by our Institutional Review Board (IRB). We recruited participants at least 18 years old of age, from the US. We also anonymized the survey entries. The "phishing" URLs, at the time of the experiment, did not lead to active phishing websites, minimizing any potential risk to those participants who attempted to visit them.

6.3.2 Recruitment. We recruited participants using the panel service "Prolific," selecting as criteria to be 18 years or older, and speaking English. The recruitment message mentioned neither security nor phishing to lessen the self-selection bias. Both Volkamer et al. [169] and Berens et al. [16] found large effect sizes in their studies. Yet, they both addressed the email context, while we study the QR code context. Thus, following Cohen [23] for unknown contexts, we aimed for a medium effect size, and estimated our sample size accordingly.

Furthermore, we ran a small pilot study to make sure the survey was working as intended. The pilot did not highlight any issue with the survey, but we noticed that the data was violating the parametric assumptions. Hence, to be conservative, we assumed the data from the main study would also show the same violations.

Thus, we conducted a power analysis using G*Power, assuming a Wilcoxon non-parametric test with a power of 0.95 and an α-error of 0.05, estimating a sample size of 139 participants per group, or a total of 556 participants. The sample demographics are shown in Table 4. Even if we did not require familiarity with QR codes, 54.31% of our sample interacted with them at least once a week.

Table 4: Sample Demographics

Category	iOS	Samsung	SEQR	$\text{SEQR}_{\text{no-tutorial}}$
Gender				
Female	79	78	75	74
Male	60	59	63	65
Non-Cis	0	2	1	0
Age				
Mean	36.5	36.9	36.8	36.3
(σ)	14.8	13.8	13.5	13.7
Scanner Used				
Default iOS	69	70	77	57
Default Android	39	37	27	38
Other	31	32	35	44
Frequency				
Daily	6	7	4	3
Multiple Weekly	35	36	32	44
Once a Week	44	26	32	33
Once a Month	34	41	32	26
Rarely	18	28	34	30
Never	2	1	5	3
QR Phishing Aware				
Yes	53	56	56	49
No	86	83	83	90

6.4 Quantitative Results

We first tested the parametric assumptions of normality (Shapiro-Wilk, $W_{\text{iOS}} = .94$, $W_{\text{Samsung}} = .97$, $W_{\text{SEQR}} = .88$, $W_{\text{SEQR-no-tutorial}} = .61$, $p < .05$), and homogeneity of variances (Levene's, $F = 10.51$, $p < .05$). As both assumptions were violated, we proceeded with the non-parametric Wilcoxon rank-sum test.

In the remainder of this section, we report each testings separately, providing medians and interquartile ranges (Mdn, IQR) to

Table 3: URLs used in the study.

Organization	Type	Risk level	URL
Bank of America	Legitimate	Unknown-risk	https://bankofamerica.com/tx/lubbock/atm-lubbock-104491.html
BBC	Legitimate	Unknown-risk	https://shop.bbc.com/products/bluey-holiday-stocking-23715
CNN	Legitimate	Unknown-risk	https://info.cnn.com/politics/election-interest-rule/index.html
Facebook	Legitimate	Low-risk	https://facebook.com/profile.php?id=1000923902349
PayPal	Legitimate	Low-risk	https://paypal.com/qrcodes/managed/d341223t3-dzsd-42cq
Google	Legitimate	Low-risk	https://docs.google.com/document/d/4KWKwos30Odaew230eew/edit
Bank of America	Obfuscate	Unknown-risk	https://hostaddress58.com/tx/lubbock/atm-lubbock-104491.html
BBC	Obfuscate	Unknown-risk	https://shop.3nk.com/products/bluey-holiday-stocking-23715
CNN	Mislead	Unknown-risk	https://cnn.com.info/politics/election-interest-rule/index.html
Facebook	Mislead	Unknown-risk	https://linkedytz.com/facebook.com?id=1000923902349
PayPal	Mangle	Unknown-risk	https://papyal.com/qrcodes/managed/d341223t3-dzsd-42cq
Google	Mangle	Unknown-risk	https://docs.googie.com/document/d/4KWKwos30Odaew230eew/edit

contextualize the results (max score = 12 correct answers). Note that each test considers the overall results, i.e., both the phishing examples and the legitimate ones. An overview of the tests results is in Table 5.

Table 5: Wilcoxon Rank-Sum Test Results

Comparison	Mdn A	Mdn B	W	r
SEQR - iOS	100.00	75.00	16412.5	.63 (large)
SEQR - Samsung	100.00	66.67	17854.5	.75 (large)
SEQR - $\text{SEQR}_{\text{no-tutorial}}$	100.00	83.33	14007	.41 (medium)

6.4.1 H_1 and H_2. The SEQR group (Mdn = 100%, IQR = 8.33%) significantly outperformed both the iOS group (Mdn = 75%, IQR = 16.67%; W = 16412.5, $p < .001$, $r = .63$), and the Samsung group (Mdn = 66.67%, IQR = 16.67%; W = 17854.5, $p < .001$, $r = .75$). Thus, both **H_1 and H_2 are supported by our data**.

6.4.2 RQ_3. The SEQR group significantly outperformed the SEQR $_{\text{no-tutorial}}$ group (Mdn = 83.33%, IQR = 25.00%), W = 14007, $p < .001$, $r = .41$. As they only differed by having read the tutorial or not, **the answer to RQ_3** is that users distinguish more effectively between phishing QR codes and legitimate ones reading the tutorial than not reading it.

6.4.3 Further Findings. As further findings, we decided to also compare the $\text{SEQR}_{\text{no-tutorial}}$ group against both the iOS group and the Samsung one. The $\text{SEQR}_{\text{no-tutorial}}$ group significantly outperformed both of them (iOS: W = 12360, $p < .001$, $r = .25$; Samsung: W = 15234.5, $p < .001$, $r = .51$). This difference can be attributed to SEQR 1.0's interface features, which support phishing detection even without tutorial guidance.

6.4.4 Notable results for individual URLs. In this section, we present more granular results of our study. Table 6 shows a summary of the results per group, while Table 7 shows the false positive results and the false negative results for each relevant entry. Table 8 shows results by each of the phishing URL used.

Number of Correct Answers. Some groups exhibited low numbers of correct answers, as shown in Table 8. For example, the URL https://cnn.com.info/politics/election-interest-rule/index.html (mislead) on Samsung achieves only 19.42%. This outlier suggests potential issues with this specific interface that requires further investigation. Table 7 contains the number of false positive.

URL obfuscation techniques and examples. The mangle technique consistently demonstrates lower numbers of correct answers compared to obfuscate and mislead (see Table 8). For instance, the URL https://papyal.com/qrcodes/managed/d341223t3-dzsd-42cq (mangle) for iOS reaches 60.43%, and 41.01% for Samsung. This suggests that the mangle technique may be the most difficult obfuscation technique to be detected through the default QR scanning options.

Intervention Group Performance. Both intervention groups, SEQR and $\text{SEQR}_{\text{no-tutorial}}$, achieved scores above 75% for all tested URLs, regardless of the URL obfuscation technique employed. Notably, the

Table 6: Correct answers, separated by study group, risk assessment, and phishing technique.

		iOS		Samsung		SEQR		$\text{SEQR}_{\text{no-tutorial}}$	
	Max	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Overall	12	75.24%	15.73	65.11%	15.56	93.35%	12.21	82.73%	15.47
Legitimate	6	81.53%	20.43	74.70%	22.73	92.45%	15.31	81.53%	21.30
Low-risk	3	89.21%	21.32	81.53%	25.12	98.56%	8.86	97.12%	12.36
Unknown-risk	3	73.86%	27.15	67.87%	30.93	86.33%	28.04	65.95%	39.62
Phishing	6	68.94%	26.93	55.52%	24.36	94.24%	14.84	83.93%	27.17
Obfuscate	2	86.33%	31.16	81.30%	32.01	96.04%	14.82	85.61%	30.24
Mislead	2	63.31%	32.74	43.53%	31.18	94.96%	19.31	85.97%	30.71
Mangle	2	57.19%	41.07	41.73%	41.09	91.73%	22.20	80.22%	33.31

Table 7: False positive (FP) and false negative results (FN), per study group, risk assessment level, and phishing technique.

		iOS		Samsung		SEQR		$\text{SEQR}_{\text{no-tutorial}}$	
	Max	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Overall	12	24.76%	15.73	34.89%	15.56	6.65%	15.47	17.27%	12.21
Legitimate (FP)	6	18.47%	20.43	25.30%	22.73	7.55%	15.31	18.47%	21.30
Low-risk (FP)	3	10.79%	21.32	18.47%	25.12	1.44%	8.23	2.88%	9.16
Unknown-risk (FP)	3	26.14%	27.15	32.13%	30.93	13.67%	28.04	34.05%	39.62
Phishing (FN)	6	31.06%	26.93	44.48%	24.36	5.76%	14.84	16.07%	27.17
Obfuscate (FN)	2	13.67%	31.16	18.71%	32.01	3.96%	14.82	14.39%	30.24
Mislead (FN)	2	36.69%	32.73	56.47%	31.18	5.04%	19.31	14.03%	30.71
Mangle (FN)	2	42.81%	41.07	58.27%	41.09	8.27%	22.20	19.78%	33.31

Table 8: Correct answers by URL visualization, split by URL obfuscation technique and study group.

How the URL is Displayed at time of Scanning	Technique	Group	Correct in %
hostaddress58.com	Obfuscate	iOS	87.05%
https://hostaddress58.com/tx/lubbock/atm-lubbock-104491.html		Samsung	82.73%
hostaddress58.com		SEQR	97.12%
hostaddress58.com		$SEQR_{no\text{-}tutorial}$	82.73%
shop.3nk.com	Obfuscate	iOS	85.61%
https://shop.3nk.com/products/bluey-holiday-stocking-23715		Samsung	79.86%
3nk.com		SEQR	94.96%
3nk.com		$SEQR_{no\text{-}tutorial}$	88.49%
linkytz.com	Mislead	iOS	84.89%
https://linkedytz.com/facebook.com?id=1000923902349		Samsung	67.63%
linkytz.com		SEQR	95.68%
linkytz.com		$SEQR_{no\text{-}tutorial}$	85.61%
cnn.com.info	Mislead	iOS	41.73%
https://cnn.com.info/politics/election-interest-rule/index.html		Samsung	19.42%
com.info		SEQR	94.24%
com.info		$SEQR_{no\text{-}tutorial}$	86.33%
papyal.com	Mangle	iOS	60.43%
https://papyal.com/qrcodes/managed/d341223t3-dzsd-42cq		Samsung	41.01%
papyal.com		SEQR	88.49%
papyal.com		$SEQR_{no\text{-}tutorial}$	76.26%
docs.googie.com	Mangle	iOS	53.96%
https://docs.googie.com/document/d/4KWKwos30Odaew230eew/edit		Samsung	42.45%
googie.com		SEQR	94.96%
googie.com		$SEQR_{no\text{-}tutorial}$	84.17%

SEQR group outperforms the $SEQR_{no\text{-}tutorial}$ group for all phishing techniques and all tested URLs.

6.4.5 Quantitative Results Discussion. This section discusses the results of our quantitative analysis.

RQ_1 and RQ_2. Our results show that SEQR 1.0 is significantly better (93.35%) in supporting participants correctly distinguishing phishing QR codes from legitimate ones than the default alternatives from both Apple iOS (75.24%) and Samsung Android (65.11%).

This is especially noticeable from the phishing perspective (94.24 %, as shown in Table 6), which is barely better than chance for the Samsung group (55.55%). Still, the most striking indication of the URL visualization shortcomings is shown in the results from the CNN URL (mislead), where the Samsung Android URL visualization made it very hard for the participants to recognized it as phishing (19.42%) Given that the Samsung Android visualization shows the full URL, the results are in line with the results from Volkamer et al. [168]. Furthermore, they also confirm the results from several previous work (e.g., [4, 7, 27, 186]) that users tends to select as legitimate any URL showing the legitimate organization name within the URL structure, irrespective of where.

Nonetheless, also the iOS group has a worse phishing detection (68.94%) than the SEQR group. Given that the iOS URL visualization is close to SEQR 1.0 URL visualization (as shown in Table 8), it seems that the different risk levels are the reason for this difference, as shown by Berens et al. [15] within the email context.

RQ_3. On the influence of the tutorial, our quantitative results confirm the results of Berens et al. [16]. The $SEQR_{no\text{-}tutorial}$ group is consistently worse than the SEQR group (shown in Table 5).

Given that, differently from Berens et al., we did not use a separate awareness measure, but rather a short one built-in the tutorial itself, we hypothesize that associating SEQR with a separate awareness measure could likely lead to even better results. Still, this aspects needs to be investigated in a future study.

However, we also argue that, because the awareness measure is an integral part of the tutorial, it might works also as a refreshment measure to extend the retention period. As mentioned in several previous works (e.g., [15, 16, 20, 133, 171]), awareness naturally decreases over time, loosing its effectiveness after five to six months. Yet, if the users' awareness is refreshed every time they open the tutorial, this might lead to an extension of the retention period. Nonetheless, just like the influence of separate awareness measures, this hypothesis needs to be investigate further.

6.5 Qualitative Results

In this section, we outline the methodology used to analyze the optional feedback of the participants from the SEQR group and the $SEQR_{no\text{-}tutorial}$ group (see 6.2.2), and the corresponding results.

6.5.1 Analysis methodology. We analyzed the data with an inductive coding approach (as described in [141, 159]). One of the researchers open coded all the open answers, creating a first codebook. The codebook was then discussed with a second coder, who independently applied it. We determined the coders agreement by calculating the *Inter-Rater Reliability* (IRR) using Cohen's kappa [24], obtaining an IRR of $k = 0.52$, which we deemed insufficient.

After a new phase of discussion to solve the conflicts, we restructured the codebook to represent the new understanding reached, and both coders independently coded the data again. We calculated the IRR again, reaching $k = 0.87$, which we deemed acceptable.

6.5.2 Findings and Implications for SEQR 2.0 development. The full results are available in Appendix F, divided by group and by question. Note that, as both questions were optional, only a subset of participants responded, as shown in Table 9.

A frequent request was automated risk assessment (55 times, in total), particularly the inclusion of a "security indicators" and "high risk" blocking case to reduce the number of unknown-risk warnings. While we deliberately excluded such case in the study – since automatic blocking would have left nothing for the participants to evaluate – this feedback confirms the relevance of our decision to include a high-risk category. Relatedly, many participants expressed frustration with the number of unknown-risk warnings (28 times mentioned). This feedback highlighted the importance of this aspect, which we therefore addressed in SEQR 2.0 (see Section 7.1.2).

Participants also requested the display of full URLs, alongside the domain-only visualization, which we implemented in SEQR 2.0. In contrast, suggestions to use yellow or orange instead of gray for unknown-risk warnings seemed out of place. Such colors are typically interpreted as sign of danger, while the unknown-risk warning is meant to be neutral, hence why we opted to keep the same coloration in SEQR 2.0.

Usability-related comments included calls for a more modern UI and improved font readability. While the interface was constrained by Android's design requirements, we addressed the latter in SEQR 2.0. Reports of slow scanning were traced to loading delays in the study environment, as no actual scan was ever performed (as described in Section 6).

Finally, several participants – especially in the $\text{SEQR}_{\text{no-tutorial}}$ group – requested an introductory tutorial, aligning with our quantitative findings that tutorial support improves effectiveness.

6.6 Third Party QR Code Scanners Investigation

In this section we cover a short investigation into popular third party QR code scanners from both Google "Play Store[5]" and Apple "App Store[6]." The goal of the investigation was to check whether the five most popular QR code scanner apps on each store for the US region (i.e., the region of our participants): 1) used a URL visualization other than those covered in our evaluation, and 2) had any security features (e.g., automated risk analysis).

6.6.1 Methodology. We opened each app store in a Firefox private window, using a new profile without cookies. We did so to avoid suggestions based on previous activity on either app store. We then searched the "app" section of each store with the query "QR code scanner," and collected the first five results. If an advertisement promoting a specific app appeared, it was counted among the first five results. In case this result was repeated, then the sixth result would have been included, but this condition did not occur. The full list of apps is in Table 10. We then downloaded all the apps collected, installed them in either a Samsung 20 (OneUI 5.1) or an iPhone SE 2022 with iOS 17. We then scanned three QR codes encoding three different URLs. The URLs are shown in Table 11.

6.6.2 Results. Regarding the Google Play store, all the apps except one used the full URL visualization also seen by the Android group participants. The only scanner that differed was "Google Lens," which showed a URL visualization like that seen by the iOS group. None of the scanners had any further security feature, e.g., automated risk assessment. Regarding the App Store, all the apps

[5]Accessible at https://play.google.com/store/apps
[6]Accessible at https://www.apple.com/app-store/

Table 9: Number of participants that gave feedback. Answering was not mandatory, so the numbers are not equal.

Usability			Security			Overall
SEQR	$\text{SEQR}_{\text{no-tutorial}}$	Total	SEQR	$\text{SEQR}_{\text{no-tutorial}}$	Total	
max = 139	max = 139	max = 278	max = 139	max = 139	max = 278	max = 556
73	74	147	69	67	136	283

Table 10: List of third party QR code scanners in order of popularity.

Store	App Name	Developer	Version
Google Play	QR & Barcode Scanner	Gamma Play Limited	2.2.95
	QR Code Scanner - without ads	Triple Tap Limited	1.0.13
	QR & Barcode Reader	TeaCapps	3.3.4-L
	Google Lense	Google LLC	1.18.250731009
	QR Code Scanner, Barcode	Simple Design Ltd.	2.6.9
App Store	QR Reader for iPhone	TapMedia Ltd.	9.6
	QR Code Reader: Quick Scan	Komorebi Inc.	4.0.0
	QR Code & Barcode Scanner	TeaCapps	2.5.4
	QR Code Reader	TinyLab	2.4.28
	QR Code Reader - Barcode Scanner	VISARGERD, S.L.	4.4.0

Table 11: URLs used to test third party QR code scanners.

Type	URL
Low Risk	https://google.com/
Exceedingly long	https://example.com/very-long-path/very-long-path/very-long-path/very-long-path/very-long-path/very-long-path/very-long-path/very-long-path/very-long-path.html
Exceedingly long / Mislead	https://very-long-sub.very-long-sub.very-long-sub.very-long-sub.sub.example.com

used the full URL visualization, also seen by the Android group. None of the scanners had any further security features either.

7 SEQR 2.0 and Implementation

In this section, we cover how the study results in Section 6.5 informed our changes to SEQR, which led to the development of SEQR 2.0. We first describe what changed from SEQR 1.0 to SEQR 2.0 (7.1), and then provide some information regarding the evaluation as Android App (7.2). SEQR 2.0 is available open source (under license GPLv3) in a GitHub repository[7].

7.1 SEQR 2.0

Based on the feedback we received in our user study (see Section 6), we made several modifications to SEQR 1.0. In this section, we describe the changes both at the user involvement level (7.1.1), and at the technology level (7.1.2).

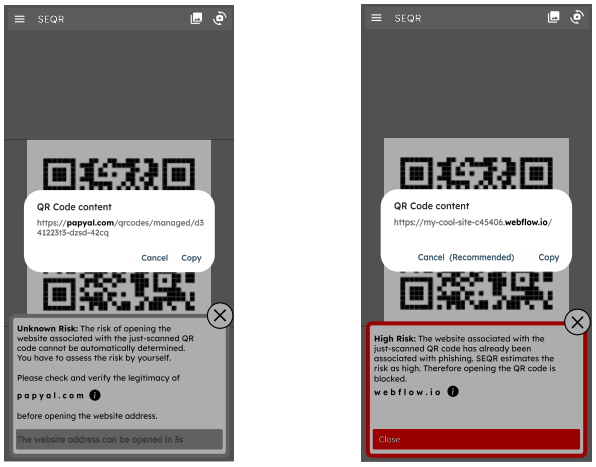
7.1.1 User Involvement Changes. At the UI level, we made three changes based on the feedback given by the users (see Section 6.5):

Expanded URL on request. Several participants complained that there is no option to show the full URL. For this reason, we expanded the SEQR warnings to contain an "i"-icon next to the URL visualization. Once the i-icon or registered domain are tapped, SEQR opens a dialog window with the full content of the QR code, as shown in Figure 9a. In the high-risk warnings, shown in Figure 9b, the dialog nudges the users towards closing the dialog without trying to copy the URL. This behavior is similar to that of browser warnings, e.g., Google Safe Browsing [43], and follows the User Interface Guidelines in Roessler & Saldhana [136] about providing a way for users to proceed anyway, while mentioning a recommended choice. Irrespective of the specific case, the extended URL visualization highlights the registered domain part to help users locating it.

Font readability. Many participants complained about the readability of the font used in SEQR 1.0. Thus, we implemented the Lexend font family [68] throughout SEQR 2.0. This font family is easily readable because each character is different from the others, increasing the readability of any text it is used in. A comparison of the font used in SEQR 1.0 and SEQR 2.0 is in Figure 10.

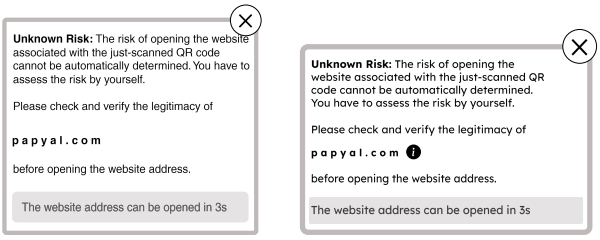
Feedback to users. SEQR 2.0 checks several online service, e.g., PhishTank or short URL providers. Depending on the network speed, and the complexity of the URL scanned (e.g., a short URL would need to go through the PhishTank step twice), this could take several seconds. To avoid the users thinking that nothing is happening, we implemented a visual feedback that informed them the scan was successful and the checks are being carried out (shown in Figure 11a). Furthermore, the base app we used as basis for SEQR

[7]https://github.com/SecUSo/SEQR-CHI-2026 (also contains a video tour of SEQR 2.0)



(a) Unknown-risk (extended) **(b) High-risk (extended)**

Figure 9: Examples of SEQR 2.0's warnings for the risk levels high-risk and unknown-risk within the app interface.

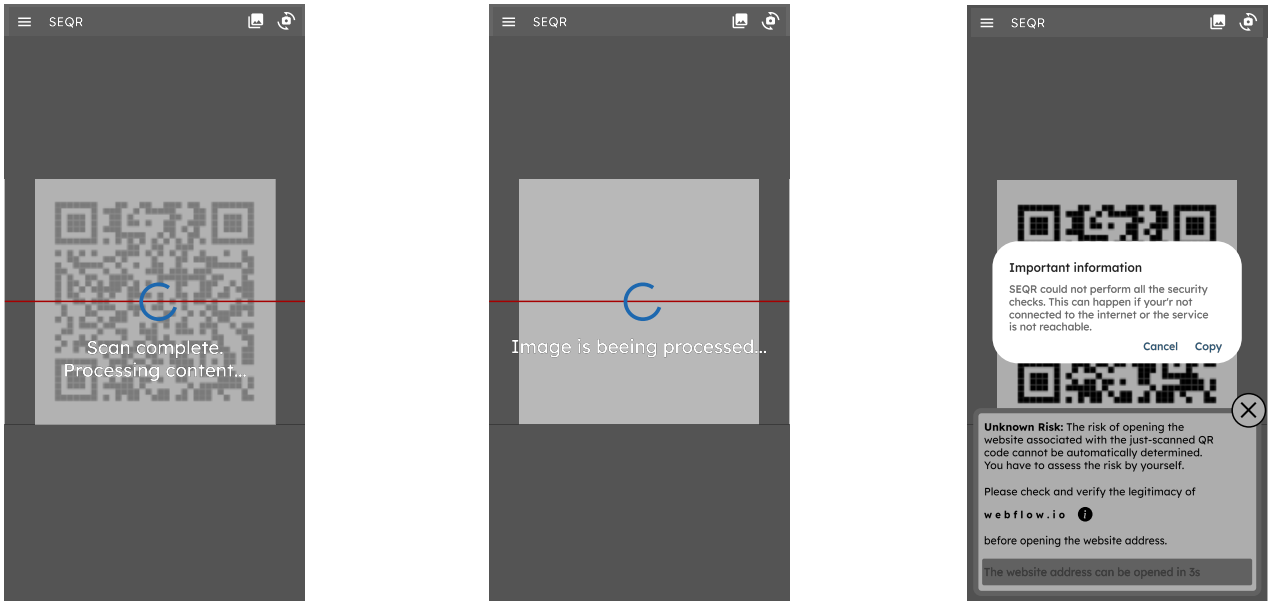


(a) SEQR 1.0 **(b) SEQR 2.0**

Figure 10: Comparison between (a) an unknown-risk warning in SEQR 1.0 and (b) the same warning in SEQR 2.0.

2.0 (described in Section 7.2.1) allowed the users to scan QR codes from their gallery. Thus, we added a separate dialog for this case, shown in Figure 11b. Lastly, if any of the online services used is unreachable, the risk assessment is not carried out. In this case, SEQR 2.0 informs the users of this failure, and show any URL as an unknown case, to reiterate that the program was not capable of checking automatically, and the users should do so themselves. This further dialog is shown in Figure 11c. A more detailed description of this challenge is in Section 7.2.3.

7.1.2 Technology Level Change. To address the feedback about having too many unknown-risk warnings, we implemented a further list in SEQR 2.0. This new list, called *Twice-visited* list, adapts over



(a) Feedback after camera scan. **(b) Feedback after gallery scan.** **(c) Warning when checks fail.**

Figure 11: User feedback dialogs in SEQR 2.0: (a) processing after scanning, (b) scanning from the gallery, and (c) unreachable services.

time to the user behavior. This feature is similar to an implementation from TORPEDO, where twice visited domains were added to the so called "history case"[16]. After the users visit the same registered domain twice, i.e., they judge two unknown-risk URLs with the same registered domain as legitimate, the registered domain is added to the twice-visited list. From this moment onward, SEQR 2.0 will consider that registered domain as low-risk, as if it was part of the TRANCO top 100 websites, and shows the corresponding low-risk warning. This way, over time, the number of unknown-risk warnings is reduced by excluding registered domains that a user frequently visits. Hence, the decision algorithm of SEQR 2.0 is expanded to consider one further decision node, as shown in Figure 12. Furthermore, we updated the tutorial accordingly.

7.2 Implementation

We first discuss how we identified an open source QR code scanner to build upon (7.2.1). Second, we describe how we implemented the various changes (7.2.2). Lastly, we explore the challenges we encountered during and how we overcame them (7.2.3).

7.2.1 Base app. Instead of developing SEQR from the ground up, we decided to build on previous results, i.e., to modify an existing app from the literature. In Section 2, we describe several related work that developed QR code scanners, i.e., [65, 127, 132, 140, 180], but none of them provided an open source version of their app, meaning that we could not build on their foundation. As we still wanted to avoid having to start from the ground up, we checked if any of the 14 QR code scanners in Dedheria [28] was available as an open source project. The paper is more than 7 years old, but our rationale was to only consider apps that are open source and that have been updated in the last year.

Only one of the apps fulfilled both these criteria: our Privacy Friendly QR Scanner [144] (*hereafter*, PFQRS), developed by SECUSO for Android and available on GitHub. The PFQRS has several more features than just scanning QR codes, e.g., it can also generate QR codes. Furthermore, it can read several other information contained in a QR code that SEQR does not cover at the moment, e.g., calendar invitations. Since the basic scanning functionality was already in place, we could focus solely on our enhancements.

7.2.2 Implementation of the Changes. For practical delimitation, the additional functionalities and formats considered by PFQRS were considered outside the scope of this paper. Thus, we disabled them, leaving only the QR code scanning feature. We also implemented other changes, namely: we adjusted the "About" section, we removed the Help section and the FAQ section, and we adapted and/or removed the icons of the disabled functions. As PFQRS already had a tutorial shown at first launch (albeit a very simple one, see Figure 13), we adapted it to show the SEQR 2.0 tutorial instead. Afterwards, we added various functionalities and changed the user interface scanning QR codes. In particular, we implemented the risk classification, the functionalities for checking PhishTank, validation of redirect URLs and resolution of redirects as helper classes. This allows easy maintenance and replacement of the specific implementation, if, e.g., PhishTank is to be replaced by another service. The changes in the user interface were implemented by adding a new overlay to the existing scanner view. This overlay is hidden by default and only shown once a QR code is scanned and the classification is complete. For the new functionalities we added new settings. Furthermore, we integrated the developed tutorial.

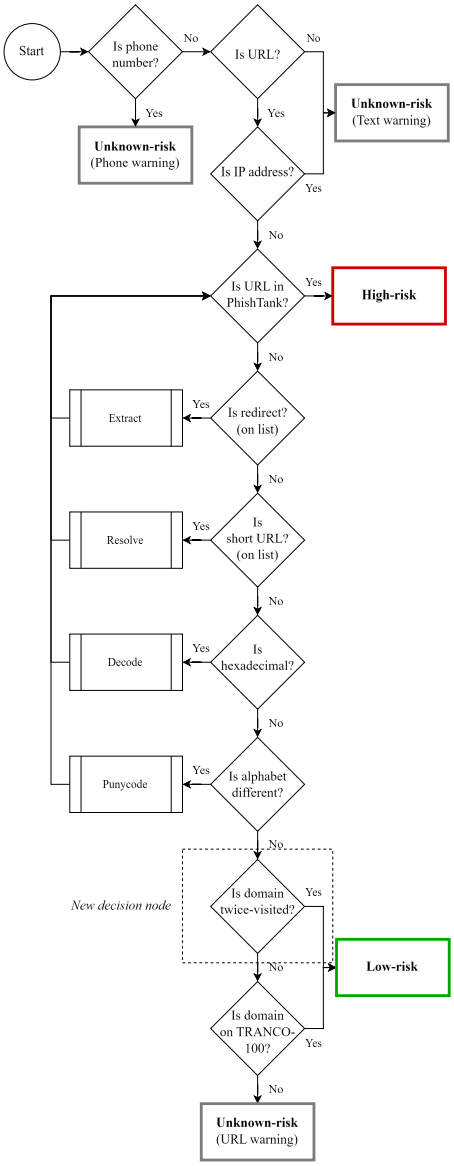


Figure 12: SEQR 2.0 decision algorithm, with the new decision node.

7.2.3 Challenges. During the implementation process, several challenges were encountered. The following section will address these challenges in detail. There are two possible methods for accessing the PhishTank database. The first method is to download the entire dataset. The second method is to use the PhishTank API. The primary concern with the first option is that it necessitates the download of the current dataset on an hourly basis to maintain accuracy, updating the list upon opening the application. Consequently, the user would be required to wait until the download and processing are complete. For this reason, we selected the option of using the API. For the TRANCO 100 list, it was determined that

(a) Step 1 **(b) Step 2** **(c) Step 3**

Figure 13: Individual Steps of the PFQRS Tutorial.

direct inclusion within the application would be the most effective approach, given the top 100 website are unlikely to frequently change. Updates to the aforementioned list can be made available to users in conjunction with the application.

Due to the time-consuming nature of the classification process, which involves querying PhishTank and resolving redirects and short URLs, a new Android ViewModel has been implemented to execute these tasks in the background. This approach ensures the continued responsiveness of the UI even in the event of a URL difficult to classify. A visual indication has been incorporated to inform the user of the ongoing process that includes a progress indicator and a text explanation. A further issue with the online checks is that they might not be reachable, which could result in a wrong classification, as is particularly relevant in the case of PhishTank. Given the inability to remediate the issue, the decision was made to inform the user in the event of such an occurrence and to assign the classification of unknown-risk.

8 SEQR 2.0 Evaluation

This section presents and discuss our comparison of SEQR 2.0 and Privacy Friendly QR Scanner (PFQRS). We first state our research questions (8.1). We then describe our methodology (8.2), the participant recruitment (8.3), and our results (8.4). Lastly, we discuss the implications of these results (8.5).

8.1 Research Questions

We evaluated SEQR 2.0 by comparing it to the PFQRS, as the latter was the baseline app modified for the SEQR 2.0 implementation. We investigated two research questions:

***RQ*$_4$.** Are the users of SEQR 2.0 more effective than the users of Privacy Friendly QR Code Scanner in distinguishing phishing QR codes from legitimate ones?

***RQ*$_5$.** Are the users reading SEQR 2.0 tutorial more effective than the users not reading SEQR 2.0 tutorial in distinguishing phishing QR codes from legitimate ones?

For each RQ we formulated one hypothesis:

H_4. SEQR 2.0 is more effective than the Privacy Friendly QR Code Scanner with respect to the users' ability to distinguish phishing QR codes from legitimate ones.

H_5. The users reading SEQR 2.0 tutorial are more effective than the users not reading SEQR 2.0 tutorial in distinguishing phishing QR codes from legitimate ones.

H_4 is based on SEQR 2.0 automated risk assessment and the use of a more focused URL visualization than PFQRS. H_5 is instead based on the results of the evaluation presented in Section 6.4.

8.2 Methodology

In this section, we describe the methodology we followed to design and conduct our user study. Note, the study structure is the same as the one used for SEQR 1.0 evaluation.

8.2.1 Study groups. We answer the RQs in Section 8.1 by dividing our participants in three groups:

PFQRS. Participants in this group saw a simulation of the PFQRS interface, as seen in a Samsung 20 (One UI 5.1). The participants went through the PFQRS tutorial before starting the task.

SEQR2. Participants in this group saw a simulation of the SEQR 2.0 interface, as seen in a Samsung 20 (One UI 5.1). The participants went through the SEQR2 tutorial before starting the task.

$SEQR2_{no\text{-}tutorial}$. Participants in this group saw a simulation of the SEQR 2.0 interface, as seen in a Samsung 20 (One UI 5.1). The participants did *not* go through the SEQR2 tutorial before the task.

8.2.2 Study structure. The structure of our user study followed that presented in Section 6.2.2, for comparability. We highlight here only the steps that differ, and refer the reader to the previous evaluation for the full description. The study structure is in Figure 14.

Differently from the time delay used by both SEQR 1.0 and SEQR 2.0 (respectively described in Section 5.1 and Section 7.1.1), PFQRS employs a delay based on checking a box to enable the URL visiting function. We implemented the PFQRS delay in the study to simulate the full interaction, and explained it to the participants in the instruction phase (see Figure 15b). The tutorial step of both PFQRS and SEQR2 followed the same flow as the SEQR1 tutorial step. We also did not collect user feedback as it was outside of our scope.

8.2.3 QR Code Scenarios. We used the same scenarios, i.e., URLs and obfuscation techniques, as in the SEQR 1.0 evaluation, for comparability. We refer the reader to Section 6.2.3 for their description.

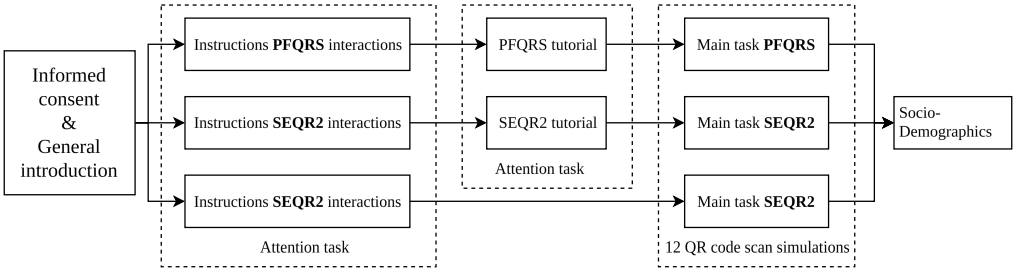
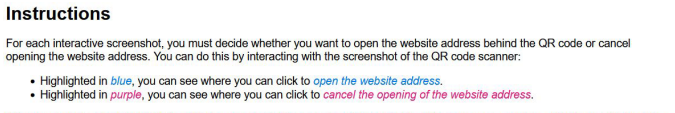
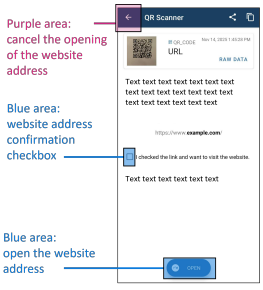


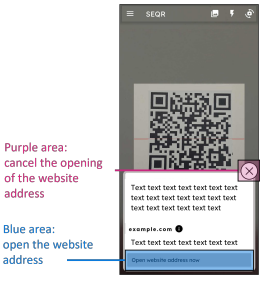
Figure 14: A left-to-right flow diagram of the second study structure, showcasing each of the steps.



(a) Instruction text



(b) PFA study group



(c) SEQR2 and $SEQR2_{no\text{-}tutorial}$ groups

Figure 15: Participation Instructions. Figure b, c were mutually exclusive depending on the group. Both showed a.

8.3 Ethics and Recruitment

We used the same study structure, recruitment, and reimbursement as in the SEQR 1.0 evaluation, so the ethical approval we obtained for the SEQR 1.0 evaluation covered this one as well. Regarding participants, we recruited them through the same process and applying the same constraints as those described in Section 6.3. We also recruited the same number of participants per group (139) as in the SEQR 1.0 evaluation for comparability, excluding those participants that took part in the first study. The sample demographics are shown in Table 12. Note, 45.56% of our participants declared to interact with QR codes at least once a week.

Table 12: Sample demographics of SEQR 2.0 evaluation.

Category	PFQRS	SEQR2	$\text{SEQR2}_{\text{no-tutorial}}$
Gender			
Female	64	68	72
Male	72	70	62
Non-Cis	3	1	5
Age			
Mean	45.7	44.5	45.3
(σ)	13.5	12.8	12.9
Scanner Used			
Default iOS	66	64	81
Default Android	47	49	41
Other	26	26	17
Frequency			
Daily	1	4	4
Multiple Weekly	16	25	21
Once a Week	43	36	40
Once a Month	39	33	34
Rarely	35	37	37
Never	5	4	3
QR Phishing Aware			
Yes	52	52	57
No	87	87	82

8.4 Results

We first tested the parametric assumptions of normality (Shapiro-Wilk, $W_{\text{PFQRS}} = .95$, $W_{\text{SEQR2}} = .59$, $W_{\text{SEQR-no-tutorial}} = .83$, $p < .001$), and homogeneity of variances (Levene's, $F = 7.1$, $p < .001$). As both assumptions were violated, we proceeded with the non-parametric Wilcoxon rank-sum test. As we did in Section 6.4, we provide medians and interquartile ranges (Mdn, IQR) for context, and report the overall results only, i.e., both the phishing examples and the legitimate ones. Table 13 shows a summary of the results per group, while Table 14 shows the false positive and the false negative results. Table 15 contains the overview of the results.

8.4.1 H_4. The SEQR2 group ($Mdn = 100\%$, $IQR = 8.33\%$) significantly outperformed the PFQRS group ($Mdn = 75\%$, $IQR = 20.83\%$; $W = 16267.5$, $p < .001$, $r = .61$). Thus, **$\mathbf{H_4}$ is supported by our data**.

8.4.2 H_5. The SEQR2 group significantly outperformed the $\text{SEQR2}_{\text{no-tutorial}}$ group ($Mdn = 91.67\%$, $IQR = 16.67\%$; $W = 12976$, $p < .001$, $r = .32$). Thus, **$\mathbf{H_5}$ is supported by our data**.

8.4.3 Further Findings. As further findings, we decided to also compare the $\text{SEQR2}_{\text{no-tutorial}}$ group against the PFQRS one. The $\text{SEQR2}_{\text{no-tutorial}}$ group significantly outperformed the PFQRS groups ($W = 13852.5$, $p < .001$, $r = .38$). This difference can be attributed to the changes to PFQRS introduced through SEQR 2.0, which support phishing detection even without tutorial guidance.

8.5 Implications

SEQR 2.0 (with and without tutorial) supports users significantly better than the app used as its basis, i.e., PFQRS. Meaning that our modifications considerably increased the effectiveness of the QR code scanner past the basic app. Albeit PFQRS provides several functionalities more than SEQR 2.0, from the point of view of secure scanning our results clearly show that SEQR 2.0 is more effective than PFQRS. Still, as explained in Section 9.7, the SEQR 2.0's scan needs expansion before it is released. The further findings confirm once more the results from Berens et al. [16] that using a tutorial with a short awareness measure significantly increases phishing detection compared to a tool on its own.

9 Discussion

This section discusses our results, covers the limitations of our work, and SEQR's threat model.

9.1 QR-Code Scanners Status Quo

Our results show that users are currently left without enough support to protect themselves against the QRishing threat. As shown in Table 6, in line with the results from recent related work (e.g. [61, 147]), the participants that used the default QR code scanners from both Apple iOS and Samsung Android were unable to detect QRishing attacks. Furthermore, Table 17 shows that the most requested function from a QR code scanner is support in determining which QR codes are phishing ones and which are not. This indicates that, when asked, users are aware of their unpreparedness and need for support, with answers such as "*Risk assessment was all I needed. I do not think much more could be added.*" This is also in line with related works (e.g., [146, 147]), where participants often pointed out the lack of help from their QR code scanner.

It is therefore surprising that a relatively small number of proposals have been advanced with the users at their center (see Section 2.4). Most of the proposals advance approaches lacking user involvement or lacking user studies to confirm their usability, e.g., [127, 132, 140]. Such approaches are a good first step, but usable secure applications such as SEQR are more likely to be adopted by the users, mostly because they have to compete with the simplicity of opening the smartphone's camera. Of course, the ideal solution would be for the iOS and Android developers to integrate SEQR-like functionalities into the default camera of their OS. Until such point, however, the arguably best trade-off is to create applications that are both effective and easy to use. The latter point is demonstrate by the results of the $\text{SEQR}_{\text{no-tutorial}}$ group, whose participants were still able to perform better than the iOS group and the Samsung one even without the guidance of the tutorial.

Table 13: Correct answers, separated by study group, risk assessment, and phishing technique.

		PFQRS		SEQR2		$SEQR2_{no\text{-}tutorial}$	
	Max	Mean	SD	Mean	SD	Mean	SD
Overall	12	76.26%	15.24	94.18%	11.25	87.47%	13.51
Legitimate	6	78.06%	25.30	93.29%	15.24	84.77%	22.11
Low-risk	3	82.25%	27.31	97.84%	12.82	95.68%	16.46
Unknown-risk	3	73.86%	31.54	88.73%	25.55	73.86%	36.06
Phishing	6	74.46%	20.49	95.08%	14.53	90.17%	17.12
Obfuscate	2	91.37%	21.64	96.04%	17.09	94.96%	16.26
Mislead	2	71.22%	28.85	94.24%	19.11	94.24%	19.11
Mangle	2	60.79%	37.96	94.96%	19.31	81.29%	32.01

Table 14: False positive (FP) and false negative results (FN), per by study group, risk assessment level, and phishing technique.

		PFQRS		SEQR2		$SEQR2_{no\text{-}tutorial}$	
	Max	Mean	SD	Mean	SD	Mean	SD
Overall	12	23.74%	15.24	5.82%	11.25	12.53%	13.51
Legitimate (FP)	6	21.94%	25.30	6.71%	15.24	15.23%	22.11
Low-risk (FP)	3	17.75%	27.31	2.16%	12.82	4.32%	16.46
Unknown-risk (FP)	3	26.14%	31.54	11.27%	25.55	26.14%	36.06
Phishing (FN)	6	25.54%	20.49	4.92%	14.53	9.83%	17.12
Obfuscate (FN)	2	8.63%	21.64	3.96%	17.09	5.04%	16.26
Mislead (FN)	2	28.78%	28.85	5.76%	19.11	5.76%	19.11
Mangle (FN)	2	39.21%	37.96	5.04%	19.31	18.71%	32.01

Table 15: Wilcoxon Rank-Sum Test Results

Comparison	Mdn A	Mdn B	W	r
SEQR2 - PFQRS	100%	75%	16267	.61 (large)
SEQR2 - $SEQR2_{no\text{-}tutorial}$	100%	91.67%	12976	.32 (medium)

9.2 Ecosystem Adoption Implications

Our study demonstrates that secure QR-code scanning can be introduced into the mobile ecosystem incrementally, without an immediate overhaul of the default scanners in iOS and Android. First, the adoption of a new security methodology is a time-consuming process; a third-party app that has already proven usability, privacy protection and security through a systematic design and a rigorous field study could serve as a bridge until operating-system vendors are convinced of the need for change. Second, mobile developers tend to wait for empirical evidence and peer-reviewed results before modifying their SDKs or APIs; the systematic analysis, controlled evaluation and real-world deployment we present provide exactly that scientific validation. Third, a growing segment of users explicitly distrusts large tech firms and is already switching to independent scanners, indicating that there is a user demand for third-party solutions. By publishing the design principles underlying SEQR, we empower iOS and Android developers to adopt and adapt these principles in their own products. Finally, while a comprehensive global market analysis would be valuable, incorporating such a study would dilute the focus of this paper; the systematic, empirically grounded contributions we provide are already actionable for developers and could catalyze incremental improvements in QRishing security across the ecosystem.

9.3 Phishing Indicator Needed

Another aspect that the security community should consider is the lack of understanding of what constitutes a phishing indicator, with several participants asking for a preview of the landing website to check its appearance, e.g., "*Would it be possible to provide a web preview of the link? That way, one can see if the preview is similar to the actual page even before opening the link.*" This shows some users are still unaware that the appearance of a website can be easily cloned by an attacker, again confirming that little has changed since the results from previous work, e.g., Sharevski et al. [147] or Greene et al. [45]. Thus, the security community should take this as a sign that the efforts to increase awareness among the population are still not enough, and that misconceptions are still present. Our results should also convince the security interventions developers that merging awareness and interventions is the right way to go, as shown before by Berens et al. [16] within the email context and

now by our results in the mobile context. Even short awareness measures like ours appear to make a significant difference.

9.4 Long-Term Implications

We have already incorporated the findings from our literature review and evaluation into an existing QR code scanner. With this implementation come two main benefits: 1) Users can now test and utilize the accumulated knowledge to protect themselves in their daily lives, and 2) Developers and researchers could learn from our findings to develop secure and effective tools in the future. Currently, no other scientifically tested QR code scanner is available, making our solution the only one that has been proven to work better than the status quo for iOS and Samsung.

Furthermore, the challenges we highlighted regarding the implementation of SEQR, especially the reliance on online services that might become unreachable, should shed light on the need for alternative ways to deliver automated risk assessments. At first glance, it may seem tempting to simply integrate a list via API, continuously accessing and performing checks on it. However, in actual implementation, there are significant obstacles, such as finding a solution for the unavailability of such services. Therefore, future research should focus on exploring ways to perform automated risk management like SEQR does, especially with regard to the balance between self-development and the use of third-party providers.

In addition, the changes implemented based on the findings from our evaluation and implementation should be re-evaluated in a study. Features that have an impact on effective security, such as the ability to fully display the URL, should be re-checked. Above all, all features and their implementation should be evaluated within the framework of a long-term study. Such long-term studies over several months are already common in the field of security awareness. For example, different forms of phishing education were tested for their effectiveness over several months [133]. These long-term studies can provide new insights and potentially lead to further design changes, especially as some aspects, such as the frequency of different risk levels, only become apparent in everyday use.

9.5 Systematic Reviews Limitations

We point to two limitations in our systematic literature review: 1) the choice of query, and 2) the choice of databases. Regarding 1), i.e., "QR AND phishing", we purposely kept the keywords as abstract as possible. Using more specific queries, e.g., "QR" AND "attacks," we might have missed relevant works that were not openly speaking of attacks. Thus, we determined that finding any work containing both the "QR" and the "phishing" keywords would have sufficed, as any works talking of both phishing and QR codes was likely to cite some form of attack. Hence, we do not believe our chosen query influenced our ability of finding relevant papers. Regarding the choice of databases, as mentioned in Section 4.1, ACM, IEEE and, Elsevier are known for publishing works on QR codes.

Still, we might have missed some attacks because not published in those databases. Yet, based on 1) the number of citations, and 2) finding little known attacks (i.e., attacks with fewer citations), we believe that we collected and categorized, if not all, the majority of QRishing techniques. Lastly, only using the MITRE ATT&CK ® might seems a limitation. However, given that ATT&CK ® is a recognized standard across industry actors (endorsed by, e.g., Microsoft [78]), we believe our choice of using it did not detract from the contribution of our systematic review.

9.6 User Studies Limitations

Five limitations, pertain to both our user studies. Participants might have been primed by the scanning flow (open or cancel a QR code embeded URL), showing how to perform these actions within the scanner interface. This was a necessary trade-off to avoid using participants' personal devices and to maintain a controlled environment. As all groups underwent the same procedure up to the main task, internal validity remains intact. External validity is relatively limited. Participants likely examined URLs more carefully than in everyday situations, setting an upper bound on real-world attentiveness. Moreover, phishing detection was their primary task, whereas in practice attention would be divided. Still, our goal was a controlled comparison across groups, which required a clearly defined task with minimal distractions. This type of focused studies was used before in the phishing context in similar situations, e.g., see [16, 74, 119, 129, 134].

54.31% of the first study participants, and 45.56% of the second study participants declared to interact with QR codes at least once a week (reported in Section 6.3 and 8.3). Participants less familiar with QR codes might lead to different results. Still, as we did not screen for this trait specifically, familiarity with QR codes might be a result of their ubiquity in everyday interactions in a US context.

We employed emulated QR code scanners, which cannot fully replicate real-world devices behavior. For the iOS group, the button to show the full URL was not clickable; our logs show no attempts to use it. Our studies focused on the default scanners of Apple and Samsung devices – two brands covering over 80% of the US market share [154]. While we cannot generalize to all devices, this selection offers a strong basis for comparison.

There was a difference in the studies length between groups due to the tutorial (SEQR, SEQR2, and PFQRS groups) and the feedback step. Regarding the first, our results show that this did not made the performance worse, rather the opposite. The feedback step was after the main task, not influencing the participants' performance.

Note also that the font used in the screenshots of the first study is Helvetica irrespective of the actual font that would be used in reality (SF Pro for iOS and Roboto for Android).

Finally, OS updates may alter the default QR code scanners behavior. Although our results are valid at the time of evaluation, any future update introducing differences might require confirmation.

9.7 Implementation Limitations

SEQR 2.0 currently only works with URLs (i.e., starting with HTTP/HTTPS), and telephone numbers. For everything else (e.g., vCard, calendar events, etc.) SEQR 2.0 falls back to a text representation. This means that URIs with different protocols, e.g., `otpauth`, are treated as text and cannot be directly opened with a supported app. Other login flows that use URLs with the HTTP/HTTPS scheme, e.g., WhatsApp Web, work fine, i.e., the corresponding app is opened automatically once the user chooses to open the URL. This is because such URLs are handled directly by the operating system. Before releasing SEQR 2.0, the most common text-based formats should be

implemented to assist the user in dealing with them, only falling back to the text representation if the QR code content is unknown. We also believe that such URIs should be implemented on a per-protocol basis to be able to decide how it should be presented and if it might be a threat to the user, like we did with the phone URI.

How SEQR 2.0 behaves when a risk assessment cannot be completed is different than how a browser would behave (as the browser maintains an off-line list of dangerous websites). We explain in Section 7.2.3 why we chose to use the PhishTank API instead of downloading the database, but we still want to implement a local database in a future version as back-up.

Lastly, the list of URL shortening services and redirection services that we use to resolve such cases can be incomplete and not cover small such services. We plan to add a function in a future version of SEQR that allows users to add their own URL shortening services and redirection services.

A broader, longitudinal focus on third party and in-app scanners adoption in several regions, e.g., China, would offer valuable context for the generalizability of our results. While such an investigation aligns with our long-term research agenda, it would necessitate a dedicated, large-scale data collection effort that falls outside the scope of this paper. Accordingly, we leave this comprehensive market analysis to a subsequent publication, where we can devote the necessary methodological depth and resources to it.

9.8 Security Threat Model

We present here the threat model behind SEQR 2.0 to give a full overview of the security assumptions we make.

Our first assumption is that the smartphone operating system, SEQR 2.0 itself, and any app opened through a QR code (e.g., WhatsApp, browser, dialer) are neither compromised nor vulnerable.

As user decision support is only provided for HTTP/HTTPS URLs and phone numbers, while all other URIs are presented as text, we assume these are not malicious or, if they are, users are not manually executing the text provided by SEQR 2.0.

Our other assumptions relate to the automated risk assessment.

SEQR 2.0 determines low-risk through: 1) the first 100 entries of the TRANCO list, and 2) a user generated list of visited websites. We make three assumptions related to these lists. First, we assume that the first 100 entries of the TRANCO list are not poisoned with phishing entries, and that the listed websites are not compromised after being added. Second, we assume that the user generated list is not populated by wrongly assessed websites, and that the listed websites are not compromised after being added. Third, we assume that website operators monitor the safety and legitimacy of user-generated content on their domain (e.g., on `docs.google.com`). Thus, domains allowing user-generated content that are part of the low-risk lists are still considered low-risk.

Regarding high-risk websites, SEQR 2.0 flags them through PhishTank. We assume that an attacker cannot flag legitimate websites as phishing, blocking users from accessing them.

Finally, we assume that the attackers cannot generate phishing URLs undetectable by users within the time delay of the Unknown-risk case. This assumption derives from SEQR 2.0 not reaching 100% phishing detection in the user study (albeit still being better than the existing support).

10 Conclusion

In conclusion, our research demonstrates the effectiveness of SEQR, a usable secure QR code scanner in thwarting phishing attacks (QRishing). Through a systematic review of academic literature and the MITRE ATT&CK® Mobile repository, we identified 60 potential attacks and developed SEQR to address them at both the technology and user involvement levels. Our evaluation of SEQR in a between-subjects online study with 556 participants showed a significant improvement in correct answers compared to existing QR code scanners (Samsung and iOS). Furthermore, SEQR also showed in a between-subjects online study with 417 participants a significant improvement in correct answers compared to the Privacy Friendly QR Scanner used as basis for its implementation.

The open-source Android implementation available on GitHub provides a foundation for enhancing QR code security for everyone and improving the usable security protections against QRishing. To further improve QR code security, future research should investigate solutions to mitigate risk assessment issues, expand SEQR's functionalities to other data encoded in QR codes (e.g., vCards), and evaluate its effectiveness through real-world devices and/or diary studies. Additionally, evaluating SEQR's accessibility and implementing it on iOS will be essential steps towards widespread adoption. By building upon our work, we can create a more secure QR code ecosystem that protects users from phishing attacks.

Acknowledgments

This work was supported by funding from the project "Engineering Secure Systems" of the Helmholtz Association (HGF) [topic 46.23.01 Methods for Engineering Secure Systems] and by KASTEL Security Research Lab.

References

[1] Jemal Abawajy. 2014. User preference of cyber security awareness delivery methods. *Behaviour and Information Technology* 33, 3 (2014), 237–248. doi:10.1080/0144929x.2012.708787

[2] Rahmad Abdillah, Zarina Shukur, Masnizah Mohd, and Mohd Zamri Murah. 2022. Phishing classification techniques: A systematic literature review. *IEEE Access* 10, 1 (2022), 41574–41591. doi:10.1109/ACCESS.2022.3166474

[3] Rana Alabdan. 2020. Phishing Attacks Survey: Types, Vectors, and Technical Approaches. *Future Internet* 12, 10 (2020), 168. doi:10.3390/fi12100168

[4] Sara Albakry, Kami Vaniea, and Maria K. Wolters. 2020. What is this URL's destination? Empirical evaluation of users' URL reading. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3313831.3376168

[5] Ahmed Aleroud and Lina Zhou. 2017. Phishing environments, techniques, and countermeasures: A survey. *Computers and Security* 68, 1 (2017), 160–196. doi:10.1016/j.cose.2017.04.006

[6] Mojtaba Alizadeh, Saeid Abolfazli, Mazdak Zamani, Sabariah Baharun, and Kouichi Sakurai. 2016. Authentication in mobile cloud computing: A survey. *Journal of Network and Computer Applications* 61, 1 (2016), 59–80. doi:10.1016/j.jnca.2015.10.005

[7] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. 2015. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies* 82, 1 (2015), 69–82. doi:10.1016/j.ijhcs.2015.05.005

[8] Apple. 2025. *QR code recognition on iOS 11 - Tech Talks - Videos.* Apple Inc. Retrieved 2025-08-23 from https://developer.apple.com/videos/play/tech-talks/206/

[9] K. S. Arikumar, A. Deepak Kumar, Sahaya Beni Prathiba, K. Tamilarasi, Rajalakshmi Shenbaga Moorthy, and M. Mohamed Iqbal. 2022. Enhancing the security of WPA2/PSK authentication protocol in Wi-Fi networks. *Procedia Computer Science* 215, 1 (2022), 413–421. doi:10.1016/j.procs.2022.12.043

[10] Ayesha Arshad, Attique Ur Rehman, Sabeen Javaid, Tahir Muhammad Ali, Javed Anjum Sheikh, and Muhammad Azeem. 2021. A Systematic Literature

Review on Phishing and Anti-Phishing Techniques. arXiv:2104.01255 [cs] doi:10.48550/arXiv.2104.01255

[11] Andrey Averin and Natalya Zyulyarkina. 2020. Malicious QR-code threats and vulnerability of blockchain. In *Proceedings of the 2020 Global Smart Industry Conference* (Chelyabinsk, RU) *(GloSIC '20)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 82–86. doi:10.1109/GloSIC50886.2020.9267840

[12] Wenying Bao, Wenbin Yao, Ming Zong, and Dongbin Wang. 2017. Cross-site Scripting Attacks on Android Hybrid Applications. In *Proceedings of the 2017 Conference on Cryptography, Security and Privacy* (Wuhan, CN) *(CSP '17)*. Association for Computing Machinery, New York, NY, USA, 56–61. doi:10.1145/3058060.3058076

[13] Luka Jure Lars Bekavac, Simon Mayer, and Jannis Strecker. 2024. QR-code integrity by design. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, 1–9. doi:10.1145/3613905.3651006

[14] Debalina Bera, Obi Ogbanufe, and Dan J. Kim. 2023. Towards a thematic dimensional framework of online fraud: An exploration of fraudulent email attack tactics and intentions. *Decision Support Systems* 171, 1 (2023), 113977. doi:10.1016/j.dss.2023.113977

[15] Benjamin Berens, Mattia Mossano, and Melanie Volkamer. 2022. Phishing awareness and education – When to best remind?. In *Proceedings of the 2022 Workshop on Usable Security and Privacy* (San Diego, CA, USA) *(USEC '22)*. Internet Society, San Diego, CA, USA, 1–15. doi:10.14722/usec.2022.23075

[16] Benjamin Maximilian Berens, Florian Schaub, Mattia Mossano, and Melanie Volkamer. 2024. Better together: The interplay between a phishing awareness video and a link-centric phishing support tool. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3544548.3581460

[17] Akashdeep Bhardwaj, Fadi Al-Turjman, Varun Sapra, Manoj Kumar, and Thompson Stephan. 2021. Privacy-aware detection framework to mitigate new-age phishing attacks. *Computers & Electrical Engineering* 96, 1 (2021), 107546. doi:10.1016/j.compeleceng.2021.107546

[18] Akashdeep Bhardwaj, Varun Sapra, Aman Kumar, Naman Kumar, and S Arthi. 2020. Why is phishing still successful? *Computer Fraud & Security* 2020, 9 (2020), 15–19. doi:10.1016/s1361-3723(20)30098-1

[19] Eric Blancaflor, Mark Brendon Calida, Matthew Chan, Nicole La Ysico, and Goten Sup An Keith. 2024. Impact Analysis and Attack Simulation on Quishing (a QC Code Phishing) using QRLJacker. In *Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies* (Sydney, AU) *(ICECET '24)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–5. doi:10.1109/icecet61485.2024.10698628

[20] Gamze Canova, Melanie Volkamer, Clemens Bergmann, and Roland Borza. 2014. Nophish: An anti-phishing education app. In *Proceedings of the 2012 Workshop on Security and Trust Management* (Wroclaw, PL) *(STM '12)*. Springer, Cham, 188–192. doi:10.1007/978-3-319-11851-2_14

[21] Kuo-Chien Chou and Ran-Zan Wang. 2020. The Nested QR Code. *IEEE Signal Processing Letters* 27, 1 (2020), 1230–1234. doi:10.1109/lsp.2020.3006375

[22] Cisco Talos Intelligence Group. 2025. PhishTank | Join the fight against phishing. Retrieved 2025-08-27 from https://phishtank.org/

[23] Barry H. Cohen. 2008. *Explaining psychological statistics*. John Wiley & Sons, Hoboken, NJ, USA.

[24] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46. doi:10.1177/001316446002000104

[25] Adrian Dabrowski, Katharina Krombholz, Johanna Ullrich, and Edgar R. Weippl. 2014. QR Inception. In *Proceedings of the 2014 Workshop on Security and Privacy in Smartphones & Mobile Devices* (Oxford, UK) *(SPSM '14)*. Association for Computing Machinery, New York, NY, USA, 3–10. doi:10.1145/2666620.2666624

[26] Denso Wave Incorporated. 2025. QR Code development history. Retrieved 2025-08-27 from https://www.denso-wave.com/en/technology/vol1.html

[27] Rachna Dhamija, J. Doug Tygar, and Marti Hearst. 2006. Why phishing works. In *Proceedings of the 2006 CHI Conference on Human Factors in Computing Systems* (Montreal, CA) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 581–590. doi:10.1145/1124772.1124861

[28] Rishabh Dudheria. 2017. Evaluating Features and Effectiveness of Secure QR Code Scanners. In *Proceedings of the 2017 Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (San Jose, CA, USA) *(CyberC '17)*. Association for Computing Machinery, New York, NY, USA, 40–49. doi:10.1109/cyberc.2017.23

[29] Ricson E. 2025. 61+ QR Code Statistics & Trends 2025 Full Report. Retrieved 2025-08-27 from https://www.qrcode-tiger.com/qr-code-statistics-2022-q1

[30] Federal Bureau of Investigation. 2022. Cybercriminals tampering with QR codes to steal victim funds. Retrieved 2025-08-27 from https://www.ic3.gov/PSA/2022/PSA220118

[31] Federal Bureau of Investigation. 2026. North Korean Kimsuky Actors Leverage Malicious QR Codes in Spearphishing Campaigns Targeting U.S. Entities. Retrieved 2026-01-22 from https://www.ic3.gov/CSA/2026/260108.pdf

[32] Ana Ferreira, Lynne Coventry, and Gabriele Lenzini. 2015. Principles of persuasion in social engineering and their use in phishing. In *Proceedings of the 2015 Conference on Human Aspects of Information Security, Privacy, and Trust* (Coventry, UK) *(HAS '15)*. Springer, Cham, CH, 36–47. doi:10.1007/978-3-319-20376-8_4

[33] Riccardo Focardi, Flaminia L. Luccio, and Heider A. M. Wahsheh. 2018. Usable cryptographic QR codes. In *Proceedings of the 2018 International Conference on Industrial Technology* (Melbourne, AU) *(ICIT '18)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1664–1669. doi:10.1109/icit.2018.8352431

[34] Riccardo Focardi, Flaminia L. Luccio, and Heider A. M. Wahsheh. 2019. Usable security for QR code. *Journal of Information Security and Applications* 48, 1 (2019), 102369. doi:10.1016/j.jisa.2019.102369

[35] Jason Ford and Hala Strohmier Berry. 2024. Feasibility of Machine Learning-Enhanced Detection for QR Code Images in Email-based Threats. In *Proceedings of the 2024 Cyber Awareness and Research Symposium* (Boston, MA, USA) *(CARS '24)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–9. doi:10.1109/cars61786.2024.10778732

[36] Anjuli Franz, Verena Zimmermann, Gregor Albrecht, and Katrin Heatwik. 2021. SoK: Still plenty of phish in the sea – a taxonomy. In *Proceedings of the 2021 Symposium on Usable Privacy and Security* (Virtual Conference) *(SOUPS '21)*. USENIX Association, Berkeley, CA, USA, 21334. doi:10.5555/3563572.3563590

[37] Michael Froehlich, Philipp Hulm, and Florian Alt. 2021. Under Pressure. A User-Centered Threat Model for Cryptocurrency Owners. In *Proceedings of the 2021 Conference on Blockchain Technology and Applications* (Virtual Conference) *(BTA '21)*. Association for Computing Machinery, New York, NY, USA, 39–50. doi:10.1145/3510487.3510494

[38] Vaibhav Garg, Lesa Huber, L. Camp, and K. Connelly. 2012. Designing risk communication for older adults. In *Proceedings of the 2011 Symposium on Usable Privacy and Security* (Eindhoven, NL) *(SOUPS '11)*. USENIX Association, Berkeley, CA, USA, 1–10. doi:10.1016/j.ssci.2012.05.002

[39] C. J. Gokul, Sankalp Pandit, Sukanya Vaddepalli, Harshal Tupsamudre, Vijayanand Banahatti, and Sachin Lodha. 2018. Phishy – a serious game to train enterprise users on phishing awareness. In *Extended Abstracts of the 2018 Symposium on Computer-Human Interaction in Play* (Melbourne, AU) *(CHI PLAY '18 Extended Abstracts)*. Association for Computing Machinery, New York, NY, USA, 169–181. doi:10.1145/3270316.3273042

[40] Google. 2020. Chromium Docs - Internationalized Domain Names (IDN) in Google Chrome. Retrieved 2025-08-25 from https://chromium.googlesource.com/chromium/src/+/main/docs/idn.md

[41] Google. 2023. Advanced phishing and malware protection. Retrieved 2025-08-27 from https://support.google.com/a/answer/9157861?hl=en

[42] Google. 2025. Google safe browsing. Retrieved 2025-08-27 from https://safebrowsing.google.com

[43] Google. 2025. Manage warnings about unsafe sites. https://support.google.com/chrome/answer/99020?sjid=17508935508680702909-EU

[44] Mahesh Gopinath and Sibi Chakkaravarthy Sethuraman. 2023. A comprehensive survey on deep learning based malware detection techniques. *Computer Science Review* 47, 1 (2023), 100529. doi:10.1016/j.cosrev.2022.100529

[45] Kristen K Greene, Michelle P Steves, Mary F, and Jennifer. 2018. User Context: An Explanatory Variable in Phishing Susceptibility . In *Proceedings of the 2018 Wokshop on Usable Security (USEC '18)*. Internet Society, Reston, VA, USA. doi:10.14722/usec.2018.23016

[46] Yuqing Guan and Andrea Tick. 2024. Literature Review on Security of Personal Information in Electronic Payments. In *Proceedings of the 2024 Symposium on Applied Computational Intelligence and Informatics* (Siofok, HU) *(SACI '24)*. WIT Press, Southampton, UK, 533–540. doi:10.1109/saci60582.2024.10619864

[47] Dong Guo, Jian Cao, Xiaoqi Wang, Qiang Fu, and Qiang Li. 2016. Combating QR-Code-Based Compromised Accounts in Mobile Social Networks. *Sensors* 16, 9 (2016), 1522. doi:10.3390/s16091522

[48] Frank Offei Gyimah, Ernest Ofori-Mensah, Henrietta Boowuo, and Sakhi Aggrawal. 2024. Empowering Shared Mobility Vehicle Riders, Stopping Scams: A Cyber Kill Chain and Awareness Approach to QRishing on College Campuses. In *Proceedings of the 2024 Cyber Awareness and Research Symposium* (Grand Forks, ND, USA) *(CARS '24)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–7. doi:10.1109/cars61786.2024.10778789

[49] Kamran J. Hamdani and Muhammad I. E. Mustafa. 2021. *Effectiveness of Online Anti-Phishing Delivery methods in raising Awareness among Internet Users*. Master's thesis. Luleå University of Technology, Department of Computer Science, Electrical and Space Engineering.

[50] Ryan Heartfield and George Loukas. 2018. Detecting semantic social engineering attacks with the weakest link: Implementation and empirical evaluation of a human-as-a-security-sensor framework. *Computers & Security* 76 (2018), 101–127. doi:10.1016/j.cose.2018.02.020

[51] Simon Hill. 2021. How to scan a QR code on your smartphone. Retrieved 2025-08-27 from https://www.wired.com/story/how-to-scan-a-qr-code/

[52] Safwati Ismail, Mohammed Hazim Alkawaz, and Alvin Ebenazer Kumar. 2021. Quick Response Code Validation and Phishing Detection Tool. In *Proceedings of*

the 2021 International Symposium on Computer Applications and Industrial Electronics (Kuala Lumpur, MY) *(ISCAIE '21)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 261–266. doi:10.1109/iscaie51753.2021.9431807

[53] Daniel Jampen, Gürkan Gür, Thomas Sutter, and Bernhard Tellenbach. 2020. Don't click: Towards an effective anti-phishing training. A comparative literature review. *Human-centric Computing and Information Sciences* 10, 1 (2020), 33. doi:10.1186/s13673-020-00237-7

[54] Jurjen Jansen and Rutger Leukfeldt. 2015. How people help fraudsters steal their money: an analysis of 600 online banking fraud cases. In *Proceedings of the 2015 Workshop on Socio-Technical Aspects in Security and Trust* (Verona, IT) *(STAST '15)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 24–31. doi:10.1109/stast.2015.12

[55] Xing Jin, Xunchao Hu, Kailiang Ying, Wenliang Du, Heng Yin, and Gautam Nagesh Peri. 2014. Code Injection Attacks on HTML5-based Mobile Apps. In *Proceedings of the 2014 Conference on Computer and Communications Security* (Scottsdale, AZ, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 66–77. doi:10.1145/2660267.2660275

[56] Mohammed Alhassan Jubur. 2024. Comparative Technical Analysis of QR Code and NFC in Contactless Payments. In *Proceedings of the 2024 International Conference on Computer Technology Applications* (Vienna, AT) *(ICCTA '24)*. Association for Computing Machinery, New York, NY, USA, 242–246. doi:10.1145/3674558.3674593

[57] Amin Kharraz, Engin Kirda, William Robertson, Davide Balzarotti, and Aurélien Francillon. 2014. Optical Delusions: A Study of Malicious QR Codes in the Wild. In *Proceedings of the 2014 Conference on Dependable Systems and Networks* (Atlanta, GA, USA) *(DSN '14)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 192–203. doi:10.1109/dsn.2014.103

[58] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, and Edgar Weippl. 2010. QR Code Security. In *Proceedings of the 2010 Conference on Advances in Mobile Computing and Multimedia* (Paris, FR) *(MoMM '10)*. Association for Computing Machinery, New York, NY, USA, 430–435. doi:10.1145/1971519.1971593

[59] Peter Kieseberg, Sebastian Schrittwieser, Manuel Leithner, Martin Mulazzani, Edgar Weippl, Lindsay Munroe, and Mayank Sinha. 2012. Malicious Pixels – Using QR Codes as Attack Vector. In *Trustworthy Ubiquitous Computing*. Atlantis Press, Paris, FR, 21–38. doi:10.2991/978-94-91216-71-8_2

[60] Doowon Kim, Haehyun Cho, Yonghwi Kwon, Adam Doupé, Sooel Son, Gail-Joon Ahn, and Tudor Dumitras. 2021. Security Analysis on Practices of Certificate Authorities in the HTTPS Phishing Ecosystem. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21)*. Association for Computing Machinery, New York, NY, USA, 407–420. doi:10.1145/3433210.3453100

[61] Marvin Kowalewski, Leona Lassak, Markus Dürmuth, and Theodor Schnitzler. 2025. Scanned and Scammed: Insecurity by obsQRity? Measuring User Susceptibility and Awareness of QR Code-based Attacks. In *Proceedings of the 2025 USENIX Security Symposium* (Vancouver, CA) *(USENIX Security '25)*. USENIX Association, Berkeley, CA, USA, 1415–1434. https://www.usenix.org/conference/usenixsecurity25/presentation/kowalewski

[62] Katharina Krombholz, Peter Frühwirt, Peter Kieseberg, Ioannis Kapsalis, Markus Huber, and Edgar Weippl. 2014. QR Code Security: A Survey of Attacks and Challenges for Usable Security. In *Human Aspects of Information Security, Privacy, and Trust*. Springer, Berlin, DE, 79–90. doi:10.1007/978-3-319-07620-1_8

[63] Katharina Krombholz, Peter Frühwirt, Thomas Rieder, Ioannis Kapsalis, Johanna Ullrich, and Edgar Weippl. 2015. QR Code Security: How Secure and Usable Apps Can Protect Users Against Malicious QR Codes. In *Proceedings of the 2015 Conference on Availability, Reliability and Security* (Antwerp, BE) *(ARES '15)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 230–237. doi:10.1109/ares.2015.84

[64] K.T.A.U Lakmal, L.M.C Perera, S.P.K Padmika, S.P.A De Silva, Dinithi Pandithage, and Deemantha Siriwardana. 2024. Email Armour: A Multi-Layered Email Defense Solution. In *Proceedings of the 2024 International Conference on Information Technology Research* (Colombo, LK) *(ICITR '24)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–6. doi:10.1109/icitr64794.2024.10857760

[65] Kurniadin Abd Latif, Bambang Sugiantoro, and Yudi Prayudi. 2019. Anti-QRishing Real-Time Technique on the QR Code Using the Address Bar-based and Domain-based Approach on Smartphone. *International Journal of Cyber-Security and Digital Forensics* 8, 2 (2019), 134–144. doi:10.17781/P002571

[66] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS '19)*. Internet Society, Reston, VA, USA, 5 pages. doi:10.14722/ndss.2019.23386

[67] Adam Lerner, Alisha Saxena, Kirk Ouimet, Ben Turley, Anthony Vance, Tadayoshi Kohno, and Franziska Roesner. 2015. Analyzing the Use of Quick Response Codes in the Wild. In *Proceedings of the 2015 Conference on Mobile Systems, Applications, and Services* (Florence, IT) *(MobiSys '15)*. Association for Computing Machinery, New York, NY, USA, 359–374. doi:10.1145/2742647.2742650

[68] Lexend. 2025. Lexend — Change the way the world reads. https://www.lexend.com/

[69] Tie Li, Gang Kou, and Yi Peng. 2020. Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems* 91, 1 (2020), 101494. doi:10.1016/j.is.2020.101494

[70] Tian Lin, Daniel E. Capecci, Donovan M. Ellis, Harold A. Rocha, Sandeep Dommaraju, Daniela S. Oliveira, and Natalie C. Ebner. 2019. Susceptibility to Spear-Phishing Emails. *ACM Transactions on Computer-Human Interaction* 26, 5 (2019), 1–28. doi:10.1145/3336141

[71] Wei-Han Lin, Guan-Yan Yang, and Kuo-Hui Yeh. 2022. Integrating FIDO Authentication with New Digital Identity in Taiwan. In *Proceedings of the 2022 Global Conference on Consumer Electronics* (Las Vegas, NV, USA) *(GCCE '22)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 311–312. doi:10.1109/gcce56475.2022.10014031

[72] Enze Liu, George Kappos, Eric Mugnier, Luca Invernizzi, Stefan Savage, David Tao, Kurt Thomas, Geoffrey M. Voelker, and Sarah Meiklejohn. 2024. Give and Take: An End-To-End Investigation of Giveaway Scam Conversion Rates. In *Proceedings of the 2024 Internet Measurement Conference* (Madrid, ES) *(IMC '24)*. Association for Computing Machinery, New York, NY, USA, 704–712. doi:10.1145/3646547.3689005

[73] Neelanjan Manna. 2022. Encrypted Message Traversal using QR codes. *International Journal of Research Publication and Reviews* 3, 10 (2022), 1073–1075. https://ijrpr.com/uploads/V3ISSUE10/IJRPR7472.pdf

[74] Ioana Andreea Marin, Pavlo Burda, Nicola Zannone, and Luca Allodi. 2023. The Influence of Human Factors on the Intention to Report Phishing Emails. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–18. doi:10.1145/3544548.3580985

[75] Richard McPherson, Suman Jana, and Vitaly Shmatikov. 2015. No Escape From Reality: Security and Privacy of Augmented Reality Browsers. In *Proceedings of the 2015 Conference on World Wide Web* (Florence, IT) *(WWW '15)*. International World Wide Web Conferences Steering Committee, Geneva, CH, 743–753. doi:10.1145/2736277.2741657

[76] Microsoft. 2023. Overview of the junk email filter. Retrieved 2025-08-27 from https://support.microsoft.com/en-us/office/overview-of-the-junk-email-filter-5ae3ea8e-cf41-4fa0-b02a-3b96e21de089

[77] Microsoft. 2024. Microsoft Edge support for Microsoft Defender SmartScreen. Retrieved 2025-08-27 from https://learn.microsoft.com/en-us/deployedge/microsoft-edge-security-smartscreen

[78] Microsoft. 2025. What is the MITRE ATT&CK framework? | Microsoft Security. Retrieved 2025-09-04 from https://www.microsoft.com/en-us/security/business/security-101/what-is-mitre-attack-framework

[79] Gaurav Misra, Nalin Asanka Gamagedara Arachchilage, and Shlomo Berkovsky. 2017. Phish Phinder: A Game Design Approach to Enhance User Confidence in Mitigating Phishing Attacks. In *Proceedings of the 2017 Symposium on Human Aspects of Information Security & Assurance* (Adelaide, AU) *(HAISA '17)*. Springer, Adelaide, AU, 41–51. doi:10.48550/arXiv.1710.06064

[80] MITRE. 2025. Abuse Elevation Control Mechanism, Technique T1626. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1626/

[81] MITRE. 2025. Access Notifications, Technique T1517. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1517/

[82] MITRE. 2025. Account Access Removal, Technique T1640. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1640/

[83] MITRE. 2025. Boot or Logon Initialization Scripts, Technique T1398. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1398/

[84] MITRE. 2025. Call Control, Technique T1616. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1616/

[85] MITRE. 2025. Command and Scripting Interpreter, Technique T1623. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1623/

[86] MITRE. 2025. Command and Scripting Interpreter: Unix Shell, Sub-technique T1623.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1623/001/

[87] MITRE. 2025. Compromise Client Software Binary, Technique T1645. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1645/

[88] MITRE. 2025. Credentials from Password Store: Keychain, Sub-technique T1634.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1634/001/

[89] MITRE. 2025. Credentials from Password Store, Technique T1634. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1634/

[90] MITRE. 2025. Data Destruction, Technique T1662. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1662/

[91] MITRE. 2025. Drive-by Compromise, Technique T1456. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1456/

[92] MITRE. 2025. FAQ. Retrieved 2025-08-27 from https://attack.mitre.org/resources/faq/

[93] MITRE. 2025. Foreground Persistence, Technique T1541. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1541/

[94] MITRE. 2025. Generate Traffic from Victim, Technique T1643. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1643/

[95] MITRE. 2025. Hijack Execution Flow: System Runtime API Hijacking, Sub-technique T1625.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1625/001/

[96] MITRE. 2025. Hijack Execution Flow, Technique T1625. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1625/

[97] MITRE. 2025. Hooking, Technique T1617. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1617/

[98] MITRE. 2025. Impair Defenses: Disable or Modify Tools, Sub-technique T1629.003. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1629/003/

[99] MITRE. 2025. Impair Defenses, Technique T1629. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1629/

[100] MITRE. 2025. Indicator Removal on Host: File Deletion, Sub-technique T1630.002. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1630/002/

[101] MITRE. 2025. Indicator Removal on Host, Technique T1630. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1630/

[102] MITRE. 2025. Indicator Removal on Host: Uninstall Malicious Application, Sub-technique T1630.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1630/001/

[103] MITRE. 2025. Input Capture: Keylogging, Sub-technique T1417.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1417/001/

[104] MITRE. 2025. Input Injection, Technique T1516. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1516/

[105] MITRE. 2025. Location tracking: Impersonate SS7 nodes, sub-technique T1430.002. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1430/002/

[106] MITRE. 2025. Masquerading: Match legitimate name or location, sub-technique T1655.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1655/001/

[107] MITRE. 2025. Masquerading, technique T1655. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1655/

[108] MITRE. 2025. Out of band data, technique T1644. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1644/

[109] MITRE. 2025. Protected user data: Calendar entries, sub-technique T1636.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1636/001/

[110] MITRE. 2025. Protected user data: Call log, sub-technique T1636.002. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1636/002/

[111] MITRE. 2025. Protected user data: Contact list, sub-technique T1636.003. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1636/003/

[112] MITRE. 2025. Protected user data: SMS messages, sub-technique T1636.004. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1636/004/

[113] MITRE. 2025. Screen capture, technique T1513. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1513/

[114] MITRE. 2025. SMS control, technique T1582. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1582/

[115] MITRE. 2025. System network configuration discovery: Internet connection discovery, sub-technique T1422.001. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1422/001/

[116] MITRE. 2025. Techniques - Mobile. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/mobile/

[117] MITRE. 2025. Virtualization solution, technique T1670. Retrieved 2025-08-27 from https://attack.mitre.org/techniques/T1670/

[118] Mattia Mossano, Benjamin Berens, Philip Heller, Christopher Beckmann, Lukas Aldag, Peter Mayer, and Melanie Volkamer. 2022. Smile - Smart Email Link Domain Extractor. In *Proceedings of the 2021 Wokshop on Security, Privacy, Organizations, and Systems Engineering* (Copenhagen, DK) *(SPOSE 2021)*. Springer, Cham, CH, 403–412. doi:10.1007/978-3-030-95484-0_23

[119] Mattia Mossano, Oksana Kulyk, Benjamin Maximillian Berens, Elena Marie Häußler, and Melanie Volkamer. 2023. Influence of URL Formatting on Users' Phishing URL Detection. In *Proceedings of the 2023 European Symposium on Usable Security* (Copenhagen, DK) *(EuroUSEC '23)*. Association for Computing Machinery, New York, NY, USA, 318–333. doi:10.1145/3617072.3617111

[120] Mattia Mossano and Melanie Volkamer. 2025. Literature Review: Misconceptions About Phishing. In *Proceedings of the Symposium on Human Aspects of Information Security and Assurance* (Skövde, SE) *(HAISA '24)*. Springer, Cham, 215–228. doi:10.1007/978-3-031-72559-3_15

[121] Francois Mouton, Louise Leenen, and H. S. Venter. 2016. Social engineering attack examples, templates and scenarios. *Computers & Security* 59, 1 (2016), 186–209. doi:10.1016/j.cose.2016.03.004

[122] Bilal Naqvi, Kseniia Perova, Ali Farooq, Imran Makhdoom, Shola Oyedeji, and Jari Porras. 2023. Mitigation strategies against the phishing attacks: A systematic literature review. *Computers & Security* 132, 1 (2023), 103387. doi:10.1016/j.cose.2023.103387

[123] Stephan Neumann, Benjamin Reinheimer, and Melanie Volkamer. 2017. Don't be deceived: The message might be fake. In *Proceedings of the 2017 Conference on Trust, Privacy and Security in Digital Business* (Lyon, FR) *(TrustBus '17)*. Springer, Cham, 199–214. doi:10.1007/978-3-319-64483-7_13

[124] Xinyu Niu, Jiandao Zhao, and Bo Tian. 2024. The Security Threat and Precautionary Measures of QR Code of Internet of Things Technology. *Advances in Engineering Technology Research* 11, 1 (2024), 775. doi:10.56028/aetr.11.1.775.2024

[125] Adebukola S. Onashoga, Oluwafolake E. Ojo, and Oluwadamilola O. Soyombo. 2019. Securix: a 3d game-based learning approach for phishing attack awareness. *Journal of Cyber Security Technology* 3, 2 (2019), 108–124. doi:10.1080/23742917.2019.1624011

[126] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher. 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* 372, 1 (2021), n71. doi:10.1136/bmj.n71

[127] Atharva Pawar, Chirag Fatnani, Rajani Sonavane, Riya Waghmare, and Sarang Saoji. 2022. Secure QR Code Scanner to Detect Malicious URL Using Machine Learning. In *Proceedings of the 2022 Asian Conference on Innovation in Technology* (Ravet, IN) *(ASIANCON '22)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–8. doi:10.1109/asiancon55314.2022.9908759

[128] Justin Petelka, Benjamin Berens, Carlo Sugatan, Melanie Volkamer, and Florian Schaub. 2025. Restricting the Link: Effects of Focused Attention and Time Delay on Phishing Warning Effectiveness. In *Proceedings of the 2025 IEEE Symposium on Security and Privacy* (San Francisco, CA, USA) *(SP 2025)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–19. doi:10.1109/SP61157.2025.00007

[129] Justin Petelka, Yixin Zou, and Florian Schaub. 2019. Put your warning where your link is: Improving and evaluating email phishing warnings. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, UK) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3290605.3300748

[130] Krassie Petrova, Adriana Romaniello, B. Dawn Medlin, and Sandra A. Vannoy. 2016. QR Codes Advantages and Dangers. In *Proceedings of the 2016 International Conference on e-Business and Telecommunications* (Qingdao, Shandong, CN) *(ICE-B '16)*. SciTePress, Setúbal, PT, 112–115. doi:10.5220/0005993101120115

[131] Proofpoint. 2021. What is phishing? - Meaning, attack types & more. Retrieved 2025-08-27 from https://www.proofpoint.com/us/threat-reference/phishing

[132] Ahmad Sahban Rafsanjani, Norshaliza Binti Kamaruddin, Hazlifah Mohd Rusli, and Mohammad Dabbagh. 2023. QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework. *IEEE Access* 11, 1 (2023), 92523–92539. doi:10.1109/access.2023.3291811

[133] Benjamin Reinheimer, Lukas Aldag, Peter Mayer, Mattia Mossano, Reyhan Duezguen, Bettina Lofthouse, Tatiana von Landesberger, and Melanie Volkamer. 2020. An investigation of phishing awareness and education over time: When and how to best remind users. In *Proceedings of the 2020 Symposium on Usable Privacy and Security* (Virtual Conference) *(SOUPS '20)*. USENIX Association, Berkeley, CA, USA, 259–284. https://www.usenix.org/conference/soups2020/presentation/reinheimer

[134] Joshua Reynolds, Deepak Kumar, Zane Ma, Rohan Subramanian, Meishan Wu, Martin Shelton, Joshua Mason, Emily Stark, and Michael Bailey. 2020. Measuring Identity Confusion with Uniform Resource Locators. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3313831.3376298

[135] German Rodriguez, Jenny Torres, Pamela Flores, Eduardo Benavides, and Daniel Nuéz Agurto. 2019. XSStudent: Proposal to Avoid Cross-Site Scripting (XSS) Attacks in Universities. In *Proceedings of the 2019 Cyber Security in Networking Conference* (Quito, EC) *(CSNet '19)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 142–149. doi:10.1109/csnet47905.2019.9108965

[136] Roessler, Thomas and Saldhana, Anil. 2010. Web Security Context: User Interface Guidelines. https://www.w3.org/TR/wsc-ui/

[137] Christoph Rottermanner, Peter Kieseberg, Markus Huber, Martin Schmiedecker, and Sebastian Schrittwieser. 2015. Privacy and data protection in smartphone messengers. In *Proceedings of the 2015 Conference on Information Integration and Web-Based Applications Services* (Seattle, WA, USA) *(iiWAS '15)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/2837185.2837202

[138] Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, and Shirin Nilizadeh. 2024. From Chatbots to Phishbots?: Phishing Scam Generation in Commercial Large Language Models. In *Proceedings of the 2024 IEEE Symposium on Security and Privacy* (San Francisco, CA, USA) *(SP '24)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 36–54. doi:10.1109/sp54263.2024.00182

[139] Dale Rutherford and Ningning Wu. 2023. Cybersecurity Risks in the Deployment and Use of Digital Business Cards: Implications for Organizations and End-Users. In *Proceedings of the 2023 Conference on Computational Science and Computational Intelligence* (Las Vegas, NV, USA) *(CSCI '23)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 765–770.

doi:10.1109/csci62032.2023.00130
[140] Nur Sahira Aziyan Mohd Sabri, Noorhayati Mohamed Noor, and Zolidah Kasiran. 2023. Secured QR Scanner (SQR) based on Query Method. In *Proceedings of the 2023 International Conference on Recent Advances and Innovations in Engineering* (Kuala Lumpur, MA) *(ICRAIE '23)*. Springer, Singapore, SG, 1–6. doi:10.1109/ICRAIE57573.2023.10012345
[141] Johnny Saldaña. 2013. *The coding manual for qualitative researchers*. SAGE Publications, Los Angeles, CA, USA.
[142] Gargi Sarkar and Sandeep K. Shukla. 2023. Behavioral analysis of cybercrime: Paving the way for effective policing strategies. *Journal of Economic Criminology* 2, 1 (2023), 100034. doi:10.1016/j.jeconc.2023.100034
[143] Yerkezhan Sartayeva and Henry C. B. Chan. 2023. A survey on indoor positioning security and privacy. *Computers & Security* 131, 1 (2023), 103293. doi:10.1016/j.cose.2023.103293
[144] SECUSO. 2025. SecUSo/privacy-friendly-qr-scanner. GitHub. Retrieved 2025-09-03 from https://github.com/SecUSo/privacy-friendly-qr-scanner
[145] Jan Seeburger. 2012. No cure for curiosity. In *Proceedings of the 2012 Nordic Conference on Human-Computer Interaction: Making Sense Through Design* (Copenhagen, DK) *(NordiCHI '12)*. Association for Computing Machinery, New York, NY, USA, 247–256. doi:10.1145/2399016.2399054
[146] Filipo Sharevski, Amy Devine, Emma Pieroni, and Peter Jachim. 2022. Phishing with Malicious QR Codes. In *Proceedings of the 2022 European Symposium on Usable Security* (Karlsruhe, DE) *(EuroUSEC '22)*. Association for Computing Machinery, New York, NY, USA, 160–171. doi:10.1145/3549015.3554172
[147] Filipo Sharevski, Mattia Mossano, Maxime Veit, Gunther Schiefer, and Melanie Volkamer. 2024. Exploring Phishing Threats through QR Codes in Naturalistic Settings. In *Proceedings of the 2024 Symposium on Usable Security* (San Diego, CA, USA) *(USEC '24)*. Internet Society, Reston, VA, USA. doi:10.14722/usec.2024.23050
[148] Vishrut Sharma. 2012. A Study of Malicious QR Codes. *International Journal of Computational Intelligence and Information Security* 3, 5 (2012), 1–6. https://www.academia.edu/1864785/A_Study_of_Malicious_QR_Codes
[149] Shu Shen, Zhao-Qing Wei, Li-Juan Sun, Yang-Qing Su, Ru-Chuan Wang, and Han-Ming Jiang. 2018. The shared bicycle and its network—internet of shared bicycle (iosb): A review and survey. *Sensors* 18, 8 (2018), 2581. doi:10.3390/s18082581
[150] Steve Sheng, Bryant Magnien, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. 2007. Anti-phishing phil: the design and evaluation of a game that teaches people not to fall for phish. In *Proceedings of the 2007 Symposium on Usable Privacy and Security* (Pittsburgh, PA, USA) *(SOUPS '07)*. Association for Computing Machinery, New York, NY, USA, 88–99. doi:10.1145/1280680.1280692
[151] Nadiia Shulzhenko and Snizhana Romashkin. 2020. Internet fraud and transnational organized crime. *Juridical Tribune* 10, 1 (2020), 162–172. https://www.tribunajuridica.eu/arhiva/An10v1/11.%20Romashkin_Shulzhenko.pdf
[152] Jun Song, Kun Gao, Xinyang Shen, Xiaotian Qi, Rui Liu, and Kim-Kwang Raymond Choo. 2018. QRFence: A flexible and scalable QR link security detection framework for Android devices. *Future Generation Computer Systems* 88, 1 (2018), 663–674. doi:10.1016/j.future.2018.05.082
[153] S S Sravan, Susmita Mandal, P J A Alphonse, and P L Ramesh. 2024. A partial offline payment system for connecting the unconnected using internet of things: A survey. *Comput. Surveys* 57, 2 (2024), 1–35. doi:10.1145/3687132
[154] StatCounter. 2024. Mobile vendor market share in the United States. Retrieved 2025-08-27 from https://gs.statcounter.com/vendor-market-share/mobile/united-states-ofamerica/
[155] Simon Stockhardt, Benjamin Reinheimer, Melanie Volkamer, Peter Mayer, Alexandra Kunz, Philip Rack, and Daniel Lehmann. 2016. Teaching phishing-security: Which way is best?. In *Proceedings of the 2016 ICT Systems Security and Privacy Protection* (Ghent, BE) *(SEC '16)*. Springer, Cham, 135–149. doi:10.1007/978-3-319-33630-5
[156] Sikiru Subairu, John Alhassan, Shafii Abdulhamid, and Joseph Ojeniyi. 2020. A Review of Detection Methodologies for Quick Response code Phishing Attacks. In *Proceedings of the 2020 International Conference on Computer and Information Sciences* (Sakaka, SA) *(ICCIS '20)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–5. doi:10.1109/iccis49240.2020.9257687
[157] Naeem Firdous Syed, Syed W Shah, Rolando Trujillo-Rasua, and Robin Doss. 2022. Traceability in supply chains: A Cyber security analysis. *Computers & Security* 112, 1 (2022), 102536. doi:10.1016/j.cose.2021.102536
[158] Divyanshu Thakur, Srijan Sah, Priyansh Ailsinghani, and Virender Ranga. 2024. A Comparative Study of Machine Learning Models for Network Traffic Classification in the IoT Landscape. In *Proceedings of the 2024 International Conference on Applied Artificial Intelligence and Computing* (Salem, IN) *(ICAAIC '24)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 670–676. doi:10.1109/icaaic60222.2024.10575268
[159] David R. Thomas. 2006. A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation* 27, 2 (2006), 237–246. doi:10.1177/1098214005283748
[160] Yu-Ju Tu, Wei Zhou, and Selwyn Piramuthu. 2021. Critical risk considerations in auto-ID security: Barcode vs. RFID. *Decision Support Systems* 142, 1 (2021), 113471. doi:10.1016/j.dss.2020.113471
[161] Gaurav Varshney, Padmavathi Iyer, Swati Goel, and Tarun Kumar Singh. 2024. UPIρ: An Envisioned Policy-Based UPI Architecture for Secure Transactions. In *Proceedings of the 2024 Conference on Mobile Ad-Hoc and Smart Systems* (Seoul, KR) *(MASS '24)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 658–663. doi:10.1109/mass62177.2024.00105
[162] Gaurav Varshney, Rahul Kumawat, Vijay Varadharajan, Uday Tupakula, and Chandranshu Gupta. 2024. Anti-phishing: A comprehensive perspective. *Expert Systems with Applications* 238, 1 (2024), 122199. doi:10.1016/j.eswa.2023.122199
[163] Gaurav Varshney, Manoj Misra, and Pradeep Atrey. 2018. Secure authentication scheme to thwart RT MITM, CR MITM and malicious browser extension based phishing attacks. *Journal of Information Security and Applications* 42, 1 (2018), 1–17. doi:10.1016/j.jisa.2018.07.001
[164] Reddy B Varun and S Rashmi. 2023. Prevalent Cyber Attacks and Defense. In *Proceedings of the 2023 International Conference on Integrated Circuits and Communication Systems* (Raichur, IN) *(ICICACS '23)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–6. doi:10.1109/icicacs57338.2023.10099516
[165] Maxime Veit, Oliver Wiese, Fabian Lucas Ballreich, Melanie Volkamer, Douglas Engels, and Peter Mayer. 2024. SoK: The past decade of user deception in emails and today's email clients' susceptibility to phishing techniques. *Computers & Security* 150, 1 (2024), 104197. doi:10.1016/j.cose.2024.104197
[166] Verizon. 2024. *Data Breach Investigations Report*. Technical Report. Verizon, New York, NY, USA. https://www.verizon.com/business/resources/reports/2024-databreach-investigations-report-dbir.pdf
[167] Timothy Vidas, Emmanuel Owusu, Shuai Wang, Cheng Zeng, Lorrie Faith Cranor, and Nicolas Christin. 2013. QRishing: The Susceptibility of SmartphoneUsers to QR Code Phishing Attacks. In *Proceedings of the 2013 Confernece on Financial Cryptography and Data Security* (Okinawa, JP) *(FC '13, Vol. 7862)*. Springer, Cham, CH, 52–69. doi:10.1007/978-3-642-41320-9_4
[168] Melanie Volkamer, Karen Renaud, and Paul Gerber. 2016. Spot the phish by checking the pruned URL. *Information and Computer Security* 24, 4 (2016), 372–385. doi:10.1108/ICS-07-2015-0032
[169] Melanie Volkamer, Karen Renaud, and Benjamin Reinheimer. 2016. TORPEDO: TOoltip-poweRed Phishing Email DetectiOn. In *Proceedings of the 2016 ICT Systems Security and Privacy Protection (SEC '16)*. Springer, Cham, 161–175. doi:10.1007/978-3-319-33630-5_12
[170] Melanie Volkamer, Karen Renaud, Benjamin Reinheimer, and Alexandra Kunz. 2017. User experiences of torpedo: Tooltip-powered phishing email detection. *Computers & Security* 71, 1 (2017), 100–113. doi:10.1016/j.cose.2017.02.004
[171] Melanie Volkamer, Karen Renaud, Benjamin Reinheimer, Philipp Rack, Marco Ghiglieri, Peter Mayer, Alexandra Kunz, and Nina Gerber. 2018. Developing and evaluating a five minute phishing awareness video. In *Proceedings of the 2018 Conference on Trust, Privacy and Security in Digital Business* (Regensburg, DE) *(TrustBus '18)*. Springer, Cham, 119–134. doi:10.1007/978-3-319-98385-1_9
[172] Heider AM Wahsheh and Flaminia L Luccio. 2020. Security and privacy of QR code applications: A comprehensive study, general guidelines and solutions. *Information* 11, 4 (2020), 217. doi:10.3390/info11040217
[173] Paul Keng Fai Wan and Shanshan Jiang. 2025. Enabling a dynamic information flow in digital product passports during product use phase: A literature review and proposed framework. *Sustainable Production and Consumption* 54, 1 (2025), 362–374. doi:10.1016/j.spc.2025.01.014
[174] Yifei Wang. 2022. A survey of phishing detection: from an intelligent countermeasures view. In *Proceedings of the 2022 Conference on Telecommunications, Optics and Computer Science* (Dalian, RC) *(TOCS '22)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 761–769. doi:10.1109/tocs56154.2022.10016193
[175] Yunjia Wang and Ishbel Duncan. 2019. A novel method to prevent phishing by using OCR technology. In *Proceedings of the 2019 Conference on Cyber Security and Protection of Digital Services* (Oxford, UK) *(Cyber Security '19)*. IEEE Computer Society, New York, NY, USA, 1–5. doi:10.1109/cybersecpods.2019.8885101
[176] Rick Wash and Molly M. Cooper. 2018. Who provides phishing training? facts, stories, and people like me. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal, CA) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3174066
[177] Zikai Alex Wen, Yiming Li, Reid Wade, Jeffrey Huang, and Amy Wang. 2017. What.hack: Learn phishing email defence the fun way. In *Extended Abstracts of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, CO, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 234–237. doi:10.1145/3027063.3048412
[178] Quan Xiao. 2021. Understanding the asymmetric perceptions of smartphone security from security feature perspective: A comparative study. *Telematics and Informatics* 58, 1 (2021), 101535. doi:10.1016/j.tele.2020.101535
[179] Takashi Yamanoue, Michio Nakanishi, Atsushi Nakamura, Izumi Fuse, Ikuya Murata, Shozo Fukada, Takahiro Tagawa, Tatsumi Takeo, Shigeto Okabe, and Tsuneo Yamada. 2005. Digital video clips covering computer ethics in higher

education. In *Proceedings of the 2005 SIGUCCS Conference on User services* (Monterey, CA, USA) *(SIGUCCS '05)*. Association for Computing Machinery, New York, NY, USA, 456–461. doi:10.1145/1099435.1099536

[180] Huiping Yao and Dongwan Shin. 2013. Towards preventing QR code based attacks on android phone using security warnings. In *Proceedings of the 2013 Asian Symposium on Information, Computer and Communications Security* (Hangzhou, RC) *(ASIACCS '13)*. Association for Computing Machinery, New York, NY, USA, 341–346. doi:10.1145/2484313.2484357

[181] Kelvin S. C. Yong, Kang Leng Chiew, and Choon Lin Tan. 2019. A survey of the QR code phishing: the current attacks and countermeasures. In *Proceedings of the 2019 International Conference on Smart Computing & Communications* (Sarawak, MY) *(ICSCC '19)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1–5. doi:10.1109/icscc.2019.8843688

[182] Mohd Imran Md Yusop, Nazhatul Hafizah Kamarudin, Nur Hanis Sabrina Suhaimi, and Mohammad Kamrul Hasan. 2025. Advancing Passwordless Authentication: A Systematic Review of Methods, Challenges, and Future Directions for Secure User Identity. *IEEE Access* 13, 1 (2025), 13919–13943. doi:10.1109/access.2025.3528960

[183] Xiao Zhang, Griffin Klevering, Xinyu Lei, Yiwen Hu, Li Xiao, and Guan-Hua Tu. 2023. The security in optical wireless communication: A survey. *Comput. Surveys* 55, 14s (2023), 1–36. doi:10.1145/3594718

[184] Yue Zhang, Jason I. Hong, and Lorrie F. Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 2007 Conference on World Wide Web* (Banff, CA) *(WWW '07)*. Association for Computing Machinery, New York, NY, USA, 639–648. doi:10.1145/1242572.1242659

[185] Rui Zhao. 2023. CamPass: a Secure Camera-based Password Manager for Kiosk Browsing. In *Proceedings of the 2023 Conference on Trust, Security and Privacy in Computing and Communications* (Chengdu, CN) *(TrustCom '23)*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1580–1585. doi:10.1109/trustcom60117.2023.00215

[186] Sarah Zheng and Ingolf Becker. 2022. Presenting suspicious details in user-facing e-mail headers does not improve phishing detection. In *Proceedings of the 2022 Symposium on Usable Privacy and Security* (Boston, MA, USA) *(SOUPS '22)*. USENIX Association, Berkeley, CA, USA, 1–12. https://www.usenix.org/conference/soups2022/presentation/zheng

[187] Anfu Zhou, Guangyuan Su, Shilin Zhu, and HuaDong Ma. 2019. Invisible QR Code Hijacking Using Smart LED. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–23. doi:10.1145/3351284

A Complete QR Code Attacks Methodology

A.1 Systematic Literature Review

Our systematic literature review followed the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) standard (shown in Fig 16 and described in Page et al. [126]).

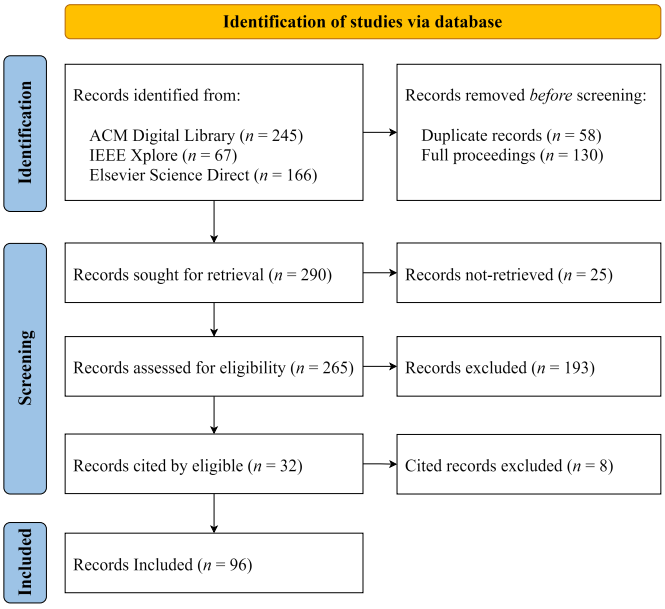


Figure 16: The process followed to identify the studies included. Modified from Page et al. [126].

First, we identified repositories known for publishing works on QR codes: ACM Digital Library, IEEE Xplore, and Elsevier Science Direct. We constrained our search to works published in the last ten years at the time of the search, i.e., 2015-2025 [8].

We queried each database with the query "QR AND phishing," and identified a total of 478 entries (ACM – 245, IEEE – 67, Elsevier – 166). After merging repeated results (58 entries), we excluded full proceedings (130 entries). Before removal, we opened each proceedings' website, expanded every track, and used Microsoft Edge's search function with our query. No further paper was found this way. We then excluded papers behind paywalls beyond our institutional access, as we could not assess them (25 entries). This left us with 265 entries.

The second phase of a literature review is usually where the exclusion of entries through their title and abstract takes place, per PRISMA. However, it might be that the QRishing techniques were only a secondary product of the research presented in a paper, e.g., mentioned as example of what an authentication scheme defends from, but not relevant to answer the research questions. Hence, the QRishing techniques might not appear in either title or abstract, but still be present in the text. Thus, we went over each of the 265 entries and searched their text with our keywords, ('QR" and "phishing") one at a time. When a passage seemed to describe a QRishing technique, we investigated further, e.g., by back-tracking to the beginning of the section. In so doing, we identified 72 relevant entries and 193 not relevant ones, i.e., not describing any QRishing technique.

We then collected the references used in the relevant sections of the 72 relevant entries (if any), and added those papers to our pool (32 further entries). We applied the same methodology as described above, and found further 24 relevant entries (excluding eight entries as non-relevant). This left us with 96 entries in total.

A.2 Systematic MITRE ATT&CK® Techniques Review

MITRE ATT&CK® (described in MITRE [92]) is a repository of adversarial behaviors divided into two parts: *Enterprise*, covering attacks against businesses, and *Mobile*, focused on mobile devices. We focused on the Mobile part, accessible at MITRE [116] and containing 121 entries at the time of our investigation.

We accessed ATT&CK® through Microsoft Edge, opened each one of the 121 entries (database version available in June 2025), and stored them locally as interactive snapshots in Zotero. The process followed is shown in Figure 17.

We then divided the entries in two groups: 1) *relevant*, i.e., entries related to QR codes or QR code technologies, 2) *not relevant*, i.e., entries unrelated to either QR codes or QR code technologies, and 3) *patched by an OS update*, i.e., an operating system update already addressed these entries. Regarding 2, ATT&CK® contains entries pertaining mobile devices as a whole, i.e., some techniques, e.g., T1451 "Sim Card

[8] May 2025; QR codes first full working ISO standard specification was released in 2015: https://www.iso.org/standard/62021.html

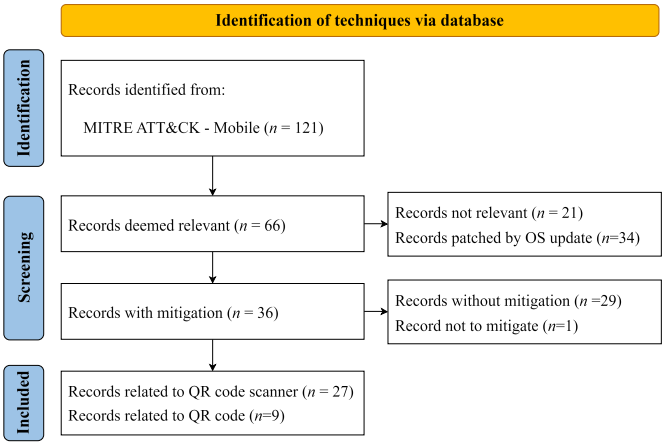


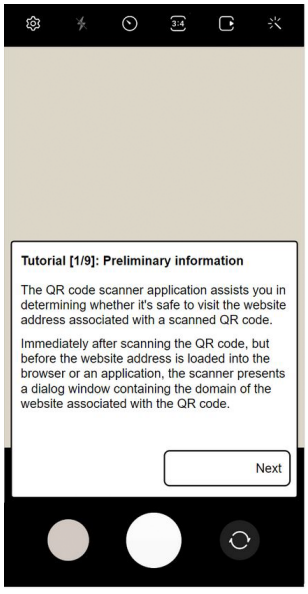
Figure 17: The process followed to identify the MITRE techniques included. Modified from Page et al. [126].

Swap," are not relevant for the QRishing context. After the first round of screening, we labeled 66 entries as "relevant," 21 entries as "not relevant," and 34 entries as "patched by an OS update."

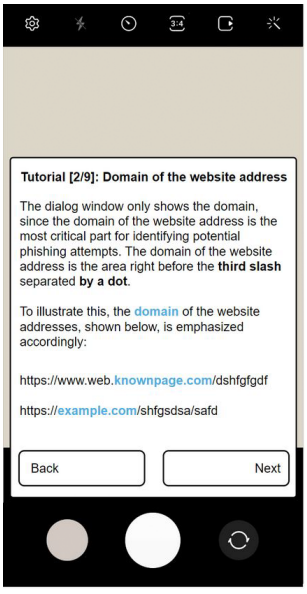
We then focused on the 66 entries labeled "relevant," and further screened them as: i) *with mitigation*, i.e., entries for which a mitigation is present in the MITRE database, ii) *without mitigation*, i.e., entries labeled as having no mitigation, and iii) *do not mitigate*, i.e., MITRE recommends not to mitigate these entries. We labeled 36 entries as "with mitigation," 29 entries as "without mitigation," and 1 entry as "do not mitigate." Regarding iii, technique T1628.003 "Hide Artifacts: Conceal Multimedia Files," is flagged by MITRE as not requiring mitigation, because the `.nomedia` files used by the attackers can also have legitimate uses. Hence, blocking them all would likely break the legitimate uses too.

We then focused on the 36 entries labeled "with mitigation," further screening them between: A) *QR code scanner related*, i.e., techniques that would require the installation of a malicious QR code scanner, and B) *QR code related*, i.e., techniques that could be encoded in a QR code. Note that B also contains techniques that are carried out on a malicious webpage, as the URL of the webpage can be encoded in the QR code. After the third screening, we labeled 27 entries as "QR code scanner related," and nine entries as "QR code related."

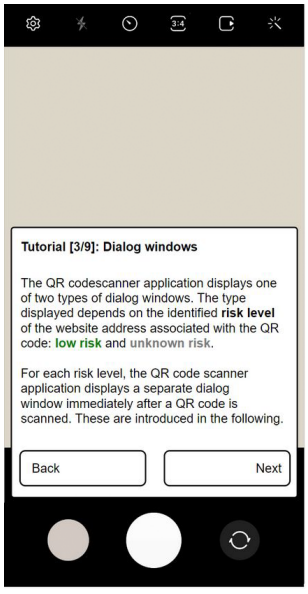
B SEQR 1.0 Evaluation Tutorial



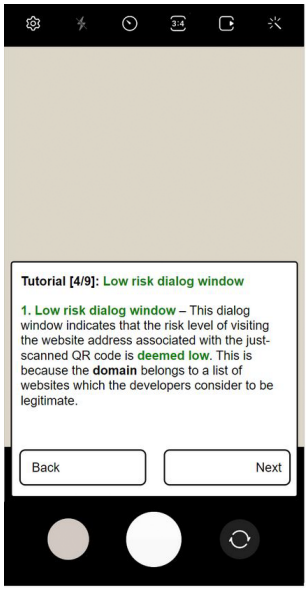
(a) Step 1



(b) Step 2

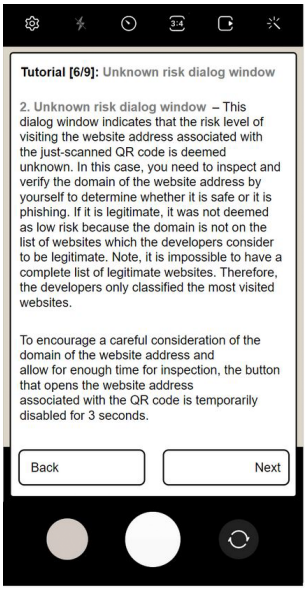


(c) Step 3

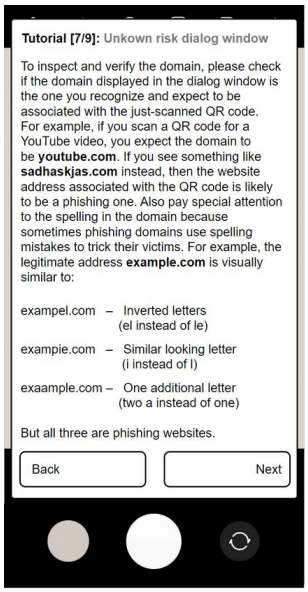


(d) Step 4

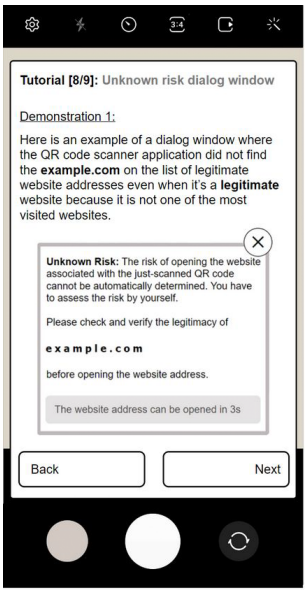
(e) Step 5



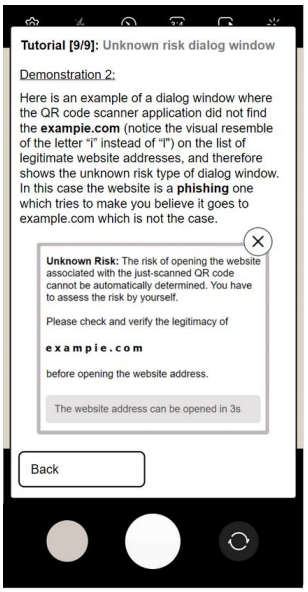
(f) Step 6



(g) Step 7



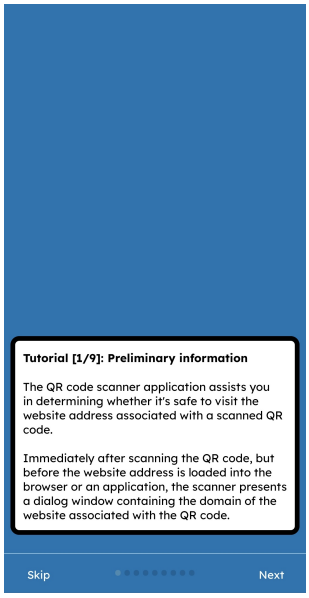
(h) Step 8



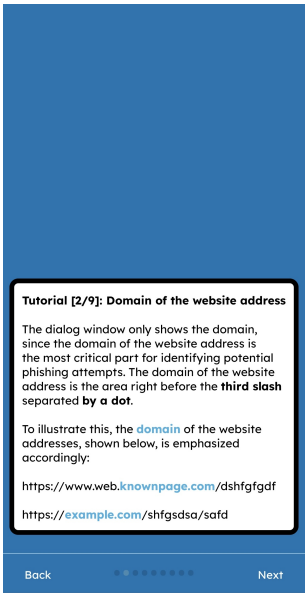
(i) Step 9

Figure 18: Individual Steps of the SEQR Tutorial

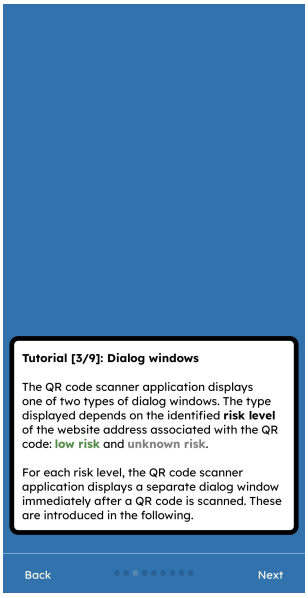
C SEQR 2.0 Evaluation Tutorial



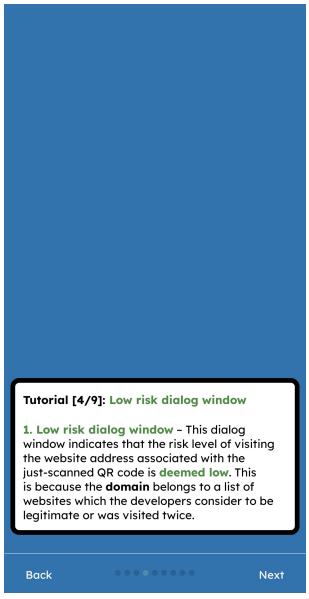
(a) Step 1



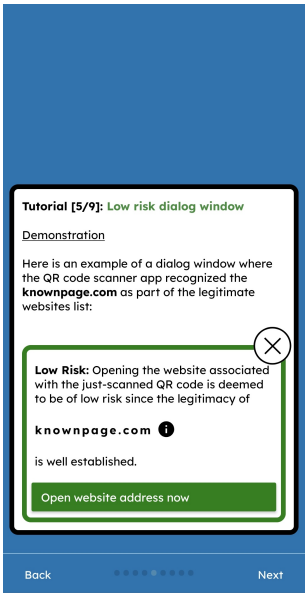
(b) Step 2



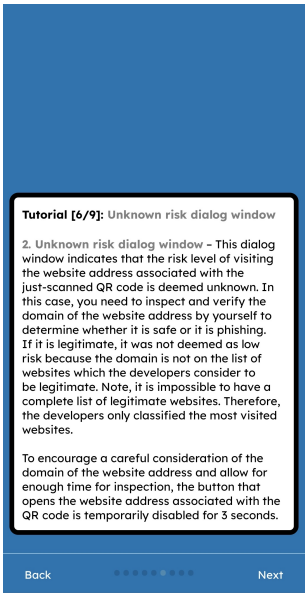
(c) Step 3



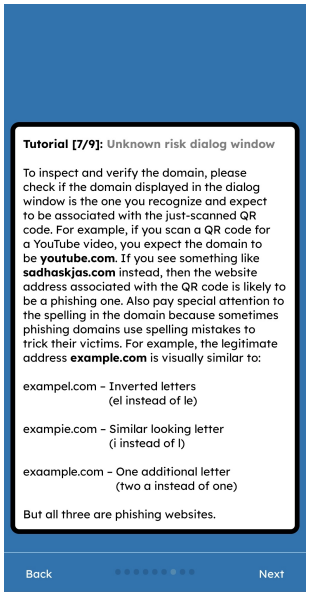
(d) Step 4



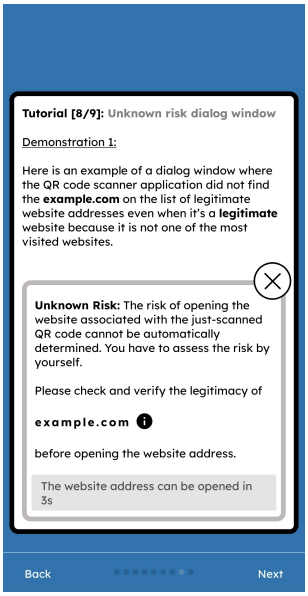
(e) Step 5



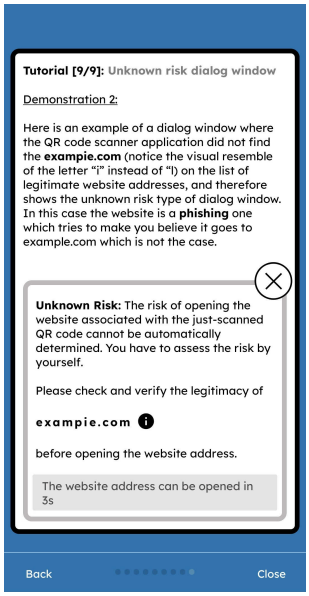
(f) Step 6



(g) Step 7



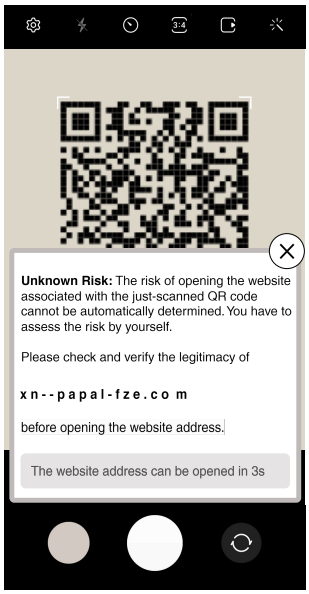
(h) Step 8



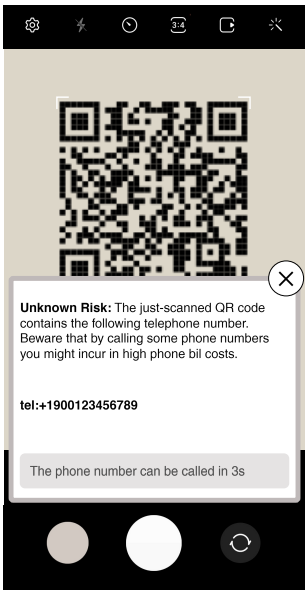
(i) Step 9

Figure 19: Screenshot of each step of tutorial as seen by the participants in the SEQR2 study group.

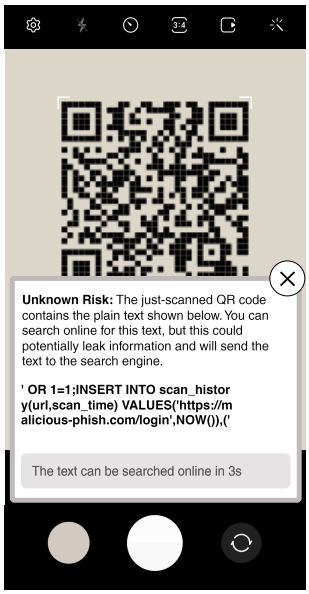
D SEQR 1.0 technical warnings



(a) Homograph spoofing



(b) Premium phone number



(c) Other content – SQL injection

Figure 20: Examples of SEQR 1.0's warnings for: homograph spoofing (a), premium phone number (b), and an SQL injection (c).

E Additional Analysis

Table 16: Post Hoc Matching Comparison

Study Group x Scanner	Correct (Matching)	Correct (Non-Matching)
iOS x iPhone	73.62869	75.23981
Samsung x Samsung	65.87302	65.10791
iOS x Samsung	75.54348	75.23981
Samsung x i0S	64.09465	65.10791
SEQR-T x iPhone	78.86473	82.73381
SEQR x iPhone	93.25397	93.34532
SEQR-T x Samsung	85.16667	82.73381
SEQR x Samsung	93.25397	93.34532

F Qualitative Analysis Full Results

Table 17: Results of the qualitative analysis, divided by question and groups. The order is by the overall values.

Code	Usability			Security			Overall
	SEQR $n = 73$	$\text{SEQR}_{\text{no-tutorial}}$ $n = 74$	Total $n = 147$	SEQR $n = 69$	$\text{SEQR}_{\text{no-tutorial}}$ $n = 67$	Total $n = 136$	
Autom. risk assessment	9	7	16	19	20	39	55
Fine as is	16	12	28	10	7	17	45
Assessment explanation	7	3	10	15	19	34	44
High risk	1	6	7	13	15	28	35
Too many unknown	5	3	8	9	11	20	28
Useless feedback	2	10	12	5	10	15	27
Color used	7	8	15	5	6	11	26
Add visited list	6	1	7	6	7	13	20
Modernize UI	7	9	16	2	1	3	19
Font readability	4	6	10	0	3	3	13
Slow scanning	6	5	11	0	1	1	12
Tutorial needed	0	6	6	2	2	4	10
Website ranking	3	0	3	5	0	5	8
"More info" button	1	0	1	6	1	7	8
Add preview	0	1	1	4	2	6	7
Buttons dimension	0	5	5	2	0	2	7
Feedback to users	1	5	6	0	0	0	6
X not usable	2	1	3	1	0	1	4
Open low-risk	2	0	2	1	0	1	3
No kerning	0	2	2	0	1	1	3
No delay friction	1	1	2	0	0	0	2
Tutorial too complex	2	0	2	0	0	0	2
Better target area	1	1	2	0	0	0	2
Integrate in camera	1	0	1	1	0	1	2
Punycode	0	0	0	0	1	1	1
Report function	1	0	1	0	0	0	1
Too basic	0	1	1	0	0	0	1
QR code generator	0	1	1	0	0	0	1