POSTER: Integrity and Origin Authentication of QR Codes

Ozge Lule
Dept. of Computer Eng.
METU, Ankara, Turkey
ozge.lule@metu.edu.tr

Esref Ozturk
Dept. of Computer Eng.
METU, Ankara, Turkey
ozturk.esref@metu.edu.tr

Ahmed Demirpolat
Dept. of Computer Eng.
METU, Ankara, Turkey
ahmed.demirpolat@metu.edu.tr

Ertan Onur *
Dept. of Computer Eng.
METU, Ankara, Turkey
eronur@metu.edu.tr

ABSTRACT

Quick Response (QR) codes are used in many applications such as business and personal cards and mobile payment systems. QR codes, as used today, introduce a very significant security vulnerability, nobody can be sure by whom the QR code is generated and whether or not the information provided in the code is modified. To discuss the impact of the lack of origin authentication, we have run an experiment in the department of computer engineering of our university. We prepared a poster containing a QR code to refer readers to a rogue web site. Around 6.5% of the students who scanned the code did not check the origin of the code and fell into the trap. Using rogue web links, sensitive information can be captured and disclosed that may have a huge impact on the victim. In this paper, we propose an origin authentication and integrity protection solution that is backward compatible and does not require any modification of the QR code standards.

Keywords

QR codes, data-origin authentication, QR code security

1. INTRODUCTION

QR codes were initially developed to track the car parts in automotive factories. However, their usage have been quickly spread out in other areas such as data sharing and transmitting. In today's world, URL redirection, payment information exchange, and electronic flight tickets are some popular examples of QR code usages [4].

Using QR codes have many advantages, albeit they may be harmful. In the scope of this research, we investigated one of the problems that originates from the design of QR Codes. QR codes do not contain any information about who generated it and how it is generated. The problem introduces many serious security vulnerabilities into the applications and systems that make use of QR codes. Some of the security risks originating from this problem are as follows:

Figure 1: Job announcement poster of the WINS Lab for QR code phishing.

- **Phishing**: Attackers may get information from a person by masquerading a trustworthy entity. For example, an attacker might provide the link to a rogue web login page using a QR Code as shown in Figure 1 that looks like the original page and acquire the password of the victim.
- **Linking to harmful websites**: Attackers may use QR codes for directing users to malicious websites to distribute malware. Malicious software distribution can be done by drive-by download attacks in which web browsers are subverted by malicious content delivered by web servers. Since QR codes are not readable by humans, users can easily be deceived. Linking to harmful websites may harm users through browser exploits such as enabling microphone/camera access, sending emails or assisting a botnet to carry out DDOS attack to another website. For example, a malicious QR code caused sensing short messages to premium mobile numbers costing $6 USD per SMS in Russia [1].

*Corresponding Author

ACM ISBN 123-4567-24-567/08/06.
DOI: 10.475/123_4

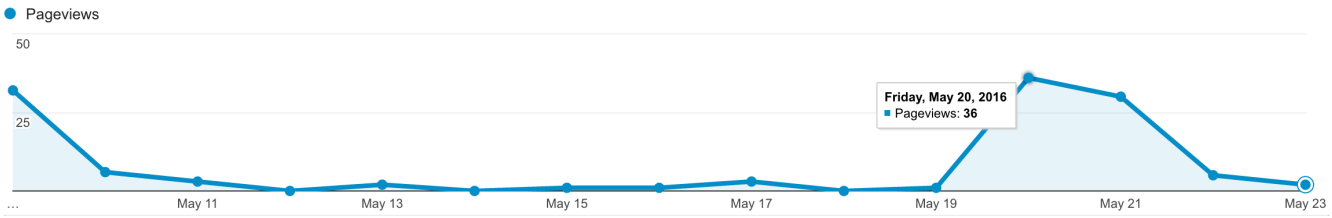

Figure 2: Google analytics results for QR code phishing.

There are many other security vulnerabilities such as information/code injection, causing unauthorized actions, or information leaking. All these vulnerability indicate the significance of origin authentication of QR codes. In literature, there are mainly two kind of security solutions for QR codes. The first is a technique that allows the creator to encrypt the QR code for a specific reader. In this scope, Google developed a technique for secure login with QR code in 2011 [3]. The idea is very straightforward and can be summarized as follows: a user is logging in the Google's services with his or her own mobile device by using some secret key such as password. In this case, user tries to login with a new device which he/she has not logged in with it before. The Symmetric Encrypted QR code (SEQR) that can only be scanned with the secret key is given to the user. The SEQR that contains the URL can then be decrypted by the verified mobile device. Finally, the user may quickly log in with the new mobile device by using the URL. Otherwise, the URL expires very quickly since it is generated uniquely for this mobile device for a short duration.

The second solution allows the reader to verify the origin of the QR code. Namely, Signed QR codes (SQR). The main idea of this technique is to prevent the reader to do any action without verifying the source of the SQR [5]. The user can progress to open the URL or fulfill any other action that is started by QR code if the verified source is trusted. The SQR method needs more modification than the other encryption techniques. Because, the message, the signature, and unique identifier to request public key from the signer must be included in the code. This approach requires the modification of the original QR code encoding and decoding process and is not backward compatible.

In the next section, we present the experiment we have carried out in the Department of Computer Engineering of Middle East Technical University. Then, we focus on the solution of the problem in Section 3.

2. QR CODE PHISHING EXPERIMENT

We conducted an experiment to examine a case where using QR codes may be harmful. A job announcement poster for our research lab as illustrated in Figure 1 was designed. There was a QR code on the poster that refers readers to a rogue web site instead of the job description. Posters were announced on the boards of the Department of Computer Engineering of Middle East Technical University (METU). Most of the students are connected to the WLAN service provided in the department that is secured through a captive portal that authenticates users before they acquire access to the Internet. When QR Code was scanned using a smart phone, it displays a rogue web page that exactly looks like the original WLAN login page of the captive portal. It cannot be distinguished from the original page unless the URL is checked. In this experiment, we intend to observe how many people will think they are redirected to the original login page.

None of the credentials are stored or viewed in this experiment; only the statistics are collected for two weeks. Google Analytics is used to observe how many times the link has been accessed (Figure 2). The login page was viewed 122 times in total in two weeks where 8 people entered their true credentials. It had the most page views on May 20th, 2016 with 36 page views.

3. ORIGIN AUTHENTICATION

To solve the defined problem, we claim that some origin authentication and integrity protection mechanism must be added to QR Codes. In this work, we aim at backward compatibility and try not to modify the QR code standard that are widely accepted and employed. If QR Codes include any means for origin authentication, users may determine whether or not the code is generated by a legitimate entity. Integrity protection part also convinces users that the information contained the QR code is not modified.

3.1 Operations

We employ public-key infrastructure [2] where the principals are publicly and uniquely identifiable. The proposed encoding process uses the traditional QR Code encoding as shown in Figure 3. The generator of the QR code will be referred to as Alice. We assume Alice wants to encode a message on a QR Code; it may be be a web site URL or something else. In order to protect integrity and provide freshness, Alice creates a string vector in which the message is hashed and it is concatenated with a timestamp and the expiration date is included. Alice signs the vector using its private key and generates a new message by concatenating the original message, the signature and her unique identifier. Then, Alice encodes the new message using the traditional QR Code encoding. The authentication vector can be extended with a sequence number if Alice has to create separate QR codes simultaneously.

When a person (will be referred to as Bob) scans the QR code with its smartphone, the origin of the QR core (Alice) is checked and authenticated as follows. The decoding process also uses the traditional QR Code decoding as presented in Figure 4. After decoding the QR code, Bob splits the message into its parts: the original message, the signature and the unique identifier. Bob hashes the original message. He acquires Alice's public key using the public key infrastructure. Bob verifies the signature using Alice's public key. He checks the expiration date for freshness and discards the expired QR codes. The hash also provides integrity protec-

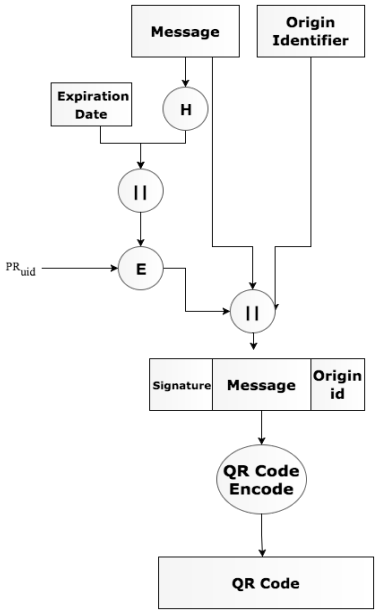

Figure 3: The encoding process for origin authentication and integrity protection.

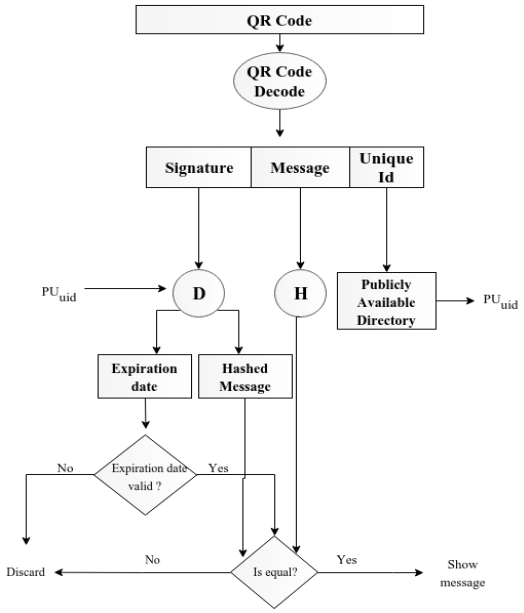

Figure 4: The origin authentication and integrity check process.

tion for not only the message but also in general for all the values in the authentication vector. If Alice's signature cannot be verified by Bob, he discards the message immediately without taking the action provided in the QR code.

3.2 Message Structure

In order to add origin authentication and integrity protection to QR codes, the encoded information (or message) has to be restructured. There are two possible structures, namely with or without a header.

If we introduce a header definition, it has to include pieces of information about the size and offset of other parts. Structuring the message with a header has two significant advantages. No extra space may be used for padding the content and the overall QR code message size may be smaller. As the disadvantage, the header will occupy some extra space. Depending on the QR code standard used, the size of the authentication vector and header may be an influential design concern. For instance, micro QR codes may contain only 35 numerals. However, the QR codes typically contain around 7089 numerals.

If a header is not employed, all parts in the message should have a fixed size. All spaces in all parts will be padded with a special character that will be forbidden in the message. This approach may lead to additional security vulnerabilities due to padding. In this approach, the space consumed for the header is not wasted but the QR code size becomes fixed.

4. CONCLUSION

QR codes are equipped with various advantages for data sharing operations. They allow distribution of structured data in different forms easily. There are many commercial applications that use QR codes. However, QR codes cannot be used securely since it has vulnerabilities inherent in its standard form. We carried out an experiment to prove these vulnerabilities and students (victims) were redirected to a malicious web site by a publicly accessible QR code. Out of 122 students, eight submitted their true credentials without checking the originality of the code. In this paper, we propose a security mechanism that perform origin authentication to counteract the mentioned vulnerabilities. As a future work, we plan to improve the integrity protection feature by introducing sequence numbers and some additional information. We have implemented the proposed solution in this paper on Android operating system and have shown the applicability in practice.

Acknowledgments

This work is partially supported by the Wireless Systems, Networks and Cybersecurity Laboratory of the Department of Computer Engineering at Middle East Technical University.

5. REFERENCES

[1] Jargon Watch. *Wired*, 20(1), January 2012.

[2] C. Adams and S. Lloyd. *Understanding public-key infrastructure: concepts, standards, and deployment considerations*. Sams Publishing, 1999.

[3] Dante D'Orazlo. Google experiments with new QR-based secure login. http://www.theverge.com/2012/1/17/2714263/google-experiment-qr-code-secure-login-sesame. Accessed: 31 Jul 2016.

[4] DENSO WAVE Inc. History of QR code. http://www.qrcode.com/en/history/. Accessed: 31 Jul 2016.

[5] K. Peng, H. Sanabria, D. Wu, and C. Zhu. Security overview of QR codes. *MIT [Online]. Available: https://courses.csail.mit.edu/6.857/2014/files/12-pengsanabria-wu-zhu-qr-codes.pdf*, 12, 2016.