QR Code Security

Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, Edgar Weippl
SBA Research
Favoritenstrasse 16
AT-1040 Vienna, Austria
[1stletterfirstname][lastname]@sba-research.org

ABSTRACT

This paper examines QR Codes and how they can be used to attack both human interaction and automated systems. As the encoded information is intended to be machine readable only, a human cannot distinguish between a valid and a maliciously manipulated QR code. While humans might fall for phishing attacks, automated readers are most likely vulnerable to SQL injections and command injections. Our contribution consists of an analysis of the QR Code as an attack vector, showing different attack strategies from the attackers point of view and exploring their possible consequences.

1. INTRODUCTION

A QR ("quick response") code is a two dimensional barcode invented by the Japanese corporation Denso Wave. Information is encoded in both the vertical and horizontal direction, thus holding up to several hundred times more data than a traditional bar code (Figure 1). Data is accessed by capturing a photograph of the code using a camera (e.g. built into a smartphone) and processing the image with a QR reader.

Figure 1: Barcode

QR Codes (Figure 2) have rapidly gained international popularity and found widespread adoption, especially in Japan where its ability to encode Kanji symbols by default makes it especially suitable. Popular uses include storing URLs, addresses and various forms of data on posters, signs, business cards, public transport vehicles, etc. Indeed, this mechanism has a vast number of potential applications [4, 1, 2, 13, 9]. For instance, the sports brand Umbro have embedded QR codes into the collars of England football shirts, sending fans to a secret website where prizes can be won.

Figure 2: QR Code

In this paper, we explore the structure and creation process of QR codes as well as potential attacks against or utilizing QR codes. We give an overview of the error correction capabilities and possible ways to alter both error correction data and payload in order to either modify or inject information into existing codes. Furthermore, we explore numerous vectors that might enable an attacker to exploit either the user's trust in the content embedded in the code or automated processes handling such codes.

Our main contributions are:

- to outline possible modifications to different parts of QR Codes such as error correction codes or masking,
- to describe resulting attack vectors, both against humans (e.g. phishing attacks) and automated processes (e.g. SQL injections).

2. BACKGROUND

QR Codes [11] have already overtaken the classical barcode in popularity in some areas. This stems in many cases from the fact that a typical barcode can only hold a maximum of 20 digits, whereas as QR Code can hold up to 7,089 characters. Combined with the diversity and extendability offered, this makes the use of QR Codes much more appealing than that of barcodes. Statistically, QR Codes are capable of encoding the same amount of data in approximately one tenth the space of a traditional bar code. A great feature of QR Codes is that they do not need to be scanned from one particular angle, as QR Codes can be read regardless of their positioning. QR codes scanners are capable of determining the correct way to decode the image due to the

TwUC'10, 8-10 November, 2010, Paris, France.

three specific squares that are positioned in the corners of the symbol and the alignment blocks.

QR Codes were initially used by vehicle manufacturers for tracking parts. After a while, companies began to see the variety of different use cases for QR Codes. The most popular commercial use for QR Codes is in the telecommunications industry, where the increasing adoption of smartphones seems to be the biggest driver of their popularity [13, 5, 6]. With the technology of mobile phones constantly evolving, especially in the area of mobile internet access, QR Codes seem to be an adequate tool to quickly and efficiently communicate URLs to users. This also allows offline media such as magazines, newspapers, business cards, public transport vehicles, signs, t-shirts or any other medium that can accept the print of a QR Code to be used as carriers for advertisements for online products [7].

2.1 QR Codes

QR Codes consist of different areas that are reserved for specific purposes. In the following we refer to version 2 of QR Codes (Figure **??**), because version 1 does not contain all patterns.

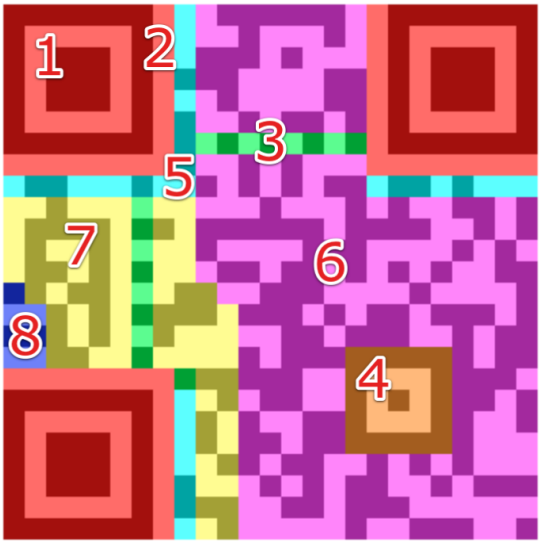

Figure 3: Structure of QR Code Version 2

- **Finder Pattern (1):** The finder pattern consists of three identical structures that are located in all corners of the QR Code except the bottom right one. Each pattern is based on a 3x3 matrix of black modules surrounded by white modules that are again surrounded by black modules. The Finder Patterns enable the decoder software to recognize the QR Code and determine the correct orientation.
- **Separators (2):** The white separators have a width of one pixel and improve the recognizability of the Finder Patters as they separate them from the actual data.
- **Timing Pattern (3):** Alternating black and white modules in the Timing Pattern enable the decoder software to determine the width of a single module.
- **Alignment Patterns (4):** Alignment Patterns support the decoder software in compensating for moderate image distortions. Version 1 QR Codes do not have Alignment Patterns. With growing size of the code, more Alignment Patterns are added.
- **Format Information (5):** The Formation Information section consists of 15 bits next to the separators and stores information about the error correction level of the QR Code and the chosen masking pattern.
- **Data (6):** Data is converted into a bit stream and then stored in 8 bit parts (called codewords) in the data section.
- **Error Correction (7):** Similar to the data section, error correction codes are stored in 8 bit long codewords in the error correction section.
- **Remainder Bits (8):** This section consists of empty bits if data and error correction bits can not be divided into 8 bit codewords without remainder.

The entire QR Code has to be surrounded by the so-called Quiet Zone, an area in the same color shade as white modules, to improve code recognition by the decoder software.

2.2 Capacity and Error correction code

The capacity of a QR Code depends on several factors. Besides the version of the code that defines its size (number of modules), the chosen error correction level and the type of encoded data influence capacity.

- **Version:** The 40 different versions of QR Codes mainly differ in the number of modules. Version 1 consists of 21x21 modules, up to 133 (lowest error correction level) of which can be used for storing encoded data. The largest QR Code (Version 40) has a size of 177x177 modules and can store up to 23,648 data modules.
- **Error Correction Level:** Error Correction in QR Codes is based on Reed-Solomon Codes [14], a specific form of BCH error correction codes [3, 8]. There are four levels (Table 1) of error correction that can be chosen by the user at creation time.

L	7%
M	15%
Q	25%
H	30%

Table 1: Error Correction Levels

Higher error correction levels increase the percentage of codewords used for error correction and therefore decrease the amount of data that can be stored inside the code.

- **Encoded Data:** QR Code can use different data encodings (see Section 3.2.2 for detailed information on character encoding modes). Their complexity influences the amount of actual characters that can be stored inside the code. For example, QR Code Version 2 with lowest error correction level can hold up to 77 numeric characters, but only 10 Kanji characters.

3. SECURITY OF QR CODES

3.1 Threat Model

One can distinguish two different threat models for manipulating QR Codes. First, an attacker may invert any module, changing it either from black to white or the other way round. Second, a more restricted attacker can only change white modules to black and not vice versa.

3.1.1 Both colors

The easiest approach for attacking an existing QR Code is by generating a sticker containing a QR Code with the manipulated QR Code in the same style as the original QR Code and position it over the code on the advertisement. Of course this would either require some preparation or a mobile printer and design applications for a mobile device. At least when attacking on a large scale against one chosen target, the time needed for preparation should not pose a serious limitation.

Since this attack is trivial, we have decided to exclude it from the scope of this paper. However, we believe that using this method in an attack against a real-world advertisement is a viable option for large-scale attacks.

3.1.2 Single color

In this case we restrict ourselves to the modification of a single color only. The background for this restriction lies in the scenario of an attacker seeking to modify a single poster on the fly just by using a pen (thereby reducing the possible modifications to changing white modules to black). This restriction is the basis for the attacks further outlined throughout this paper.

3.2 Attacking different parts

Since QR Codes contain a lot of different information, including meta information on version, maskings and source encoding, several different regions exist that can be targeted either individually or in combination.

3.2.1 The masks

Masks are used to generate QR Codes with a good distribution of black and white modules (close to 50:50 and distributed well over the whole code). This increases the contrast of the picture and thus helps devices to decode it. According to the standard, when generating a QR Code, every mask of the 8 specified ones is applied and each result is rated. The mask that results in the best distribution according to the rating is chosen (the effect of using correct masks can be seen in Figure 4. The bar *before* shows the number of white and black elements before any masking was applied, *after* the numbers after the best masking was used). There is always only one mask in use in a given QR Code, it is encoded together with the version in a separate block of the code using strong BCH encoding.

In the conditions in Table 2, i refers to the row position of the module and j to its column position. The mask is black for every module the condition is valid for and white for the rest.

Targeting the mask can change quite a lot in the whole data and error correction part, still, this can be a useful basis for additionally applying other methods. A problem when changing the masking is that it is encoded separately, utilizing a strong error correction algorithm.

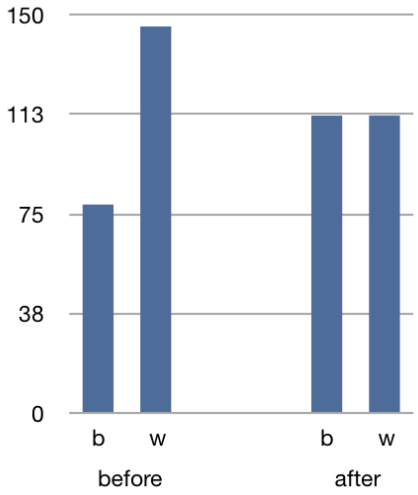

Figure 4: Effects of masking

Mask Pattern	Condition
000	$(i+j) \bmod 2 = 0$
001	$i \bmod 2 = 0$
010	$j \bmod 3 = 0$
011	$(i+j) \bmod 3 = 0$
100	$((i/2)+(j/3)) \bmod 2 = 0$
101	$(ij) \bmod 2 + (ij) \bmod 3 = 0$
110	$((ij) \bmod 2 + (ij) \bmod 3) \bmod 2 = 0$
111	$((ij) \bmod 3 + (ij) \bmod 2) \bmod 2 = 0$

Table 2: Mask Pattern References and conditions

3.2.2 The character encoding (mode)

There are several different source encodings (Table 3) specified for the information contained in the code, thereby maximizing the capacity in exchange for increased complexity:

- Numeric mode (just encoding digits, thus being able to pack a lot of data in one picture),
- Alphanumeric mode (a set of characters containing upper case letters and several additional characters like *$* or *whitespace*),
- 8-bit mode (able to encode the JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201) or
- Kanji characters (Shift JIS character set in accordance with JIS X 0208 Annex 1 Shift Coded Representation)

to name the most popular.

The character encoding itself is defined at the start of the data part by the leading 4 bits. The following table gives an overview on the possible values for the mode:

Changing the mode indicator gives the encoded data a whole new meaning. Especially when considering 8-bit Byte mode instead of other modes, or even alphanumeric mode instead of e.g. Numeric mode, launching code injections (e.g. SQL injections) could become feasible (8-bit mode even allows for even more complex tricks using control characters).

indicator	mode
0001	Numeric mode
0010	Alphanumeric mode
0100	8-bit Byte mode
1000	Kanji mode
0011	Structured append
0101	FNC1 (first position)
1001	FNC1 (second position)
0000	Terminator

Table 3: Mode indicators

An advanced attack against the mode can be mounted by mixing different modes in one QR Code (see the section below).

3.2.3 Character Count Indicator

Right after the mode indicator, the next bits indicate the character count of the data that follows. The actual size of the character count indicator largely depends on the mode in use and the version of the QR Code (higher versions contain more data, thus the character count indicator is longer). See the table below for lengths for the most popular modes (Table 4).

Version	Numeric	Alphanumeric	8-bit	Kanji
1-9	10	9	8	8
10-26	12	11	16	10
27-40	14	13	16	12

Table 4: Length of character count indicator

Looking at this target we see two main approaches for an attack: Producing buffer overflows or buffer underflows.

- **Buffer underflow:** We change the character count indicator to resemble a lower number than in the original QR Code. Thus, a decoding device should only decode the first few letters as message, leaving out the rest. This is especially useful in case the original link that was encoded contained suffixes. Since the size of the data part is fixed, everything after the anticipated number of bytes is either seen as a new segment (see mixed modes), or (in case of a terminator mode indicator) as filler. Since this is only a minor change in the data part (only the length is changed), this should in turn result in only a minor change in the error correction values and might still be decodable. Special attention should also be paid to a possible combination of this attack with other targets in case mixed modes are used.

- **Buffer overflow:** We change the character counter indicator to a higher number, so the decoding device tries to decode parts of the filler as data (if we can even change the filler we gain valuable space for including additional data). Again, one of the drawbacks of this method lies in the error correcting (and especially error detecting) abilities of the Reed-Solomon-Code. It would be especially interesting to try to launch a code injection attack using this method.

3.2.4 Mixing modes

It is possible to use several different modes within one QR Code (this is especially useful to increase density when encoding different types of data). To achieve this, several data segments with their own mode indicators and character count indicators simply get concatenated (see figure 5). Again, we can identify four approaches of utilizing this feature of QR Codes for attacks:

- **Changing modes of segments:** This works like an attack on the mode of a QR Code that does not use mixed modes, but is reduced to one segment only, thus leaving other parts of the data untouched.

- **Inserting new segments:** Split an existing segment into two new segments, one with the old header (mode indicator and character count indicator) and one with a newly defined header.

- **Deleting existing segments:** Overwrite the space used by the mode indicator and the character count indicator of the segment with additional data, thus reducing the number of segments. Additionally the character count indicator of the segment before can be changed to allow the data of the deleted segment to be appended.

- **Structured Append:** The structured append mode allows for the concatenation of up to 16 QR Code symbols, so a lot of data can be stored in this sequence of QR Codes. It is designed in a way that decoding is independent from the order the symbols are read. The basic idea of exploiting this lies in adding such a segment that points to another QR Code that is stuck right beneath the original one. However, this attack requires stickers prepared in advance, making it less practical (it is far more work than just putting a prepared sticker on top of the original QR Code, especially since the structured append mode is quite complex).

The main problem with all attacks against the mode (especially insertion and deletion of segments) is that they usually have a very high impact on the error correction codewords. Probably this attack can be used in combination with an attack on the error correction part itself.

Changing modes of existing segments should be attempted to be combined with color changes in the data section (and of course the error correction part), since it could be a valuable aid in case there are not enough white modules left for changing (remember that, as a prerequisite, we only consider changing one color to the other without the possibility to do the reverse). Also changing modes like numeric and alphanumeric to 8-bit would suddenly allow for unprintable control characters like *delete* (Figure 5).

3.2.5 Data part and error correction

The largest part of a QR Code is made up of the parts containing data and error correction codewords. The data part itself can consist of segments using several different encodings, each with its own header specifying the mode in use and the length of the data following (see subsections on modes above). For a given version and error correction level, the part in the QR Code that represents the data codewords and the part representing the error correction codewords can be defined easily without decoding, since the length of the data

Segment 1			Segment 2				Segment n		
Mode Indicator 1	Character Count Indicator 1	Data	Mode Indicator 2	Character Count Indicator 2	Data	...	Mode Indicator n	Character Count Indicator n	Data

Figure 5: Message containing several modes

part is not depending on the actual length of the data (the data is filled up with padding patterns to the full length). For the position of the data part, please look at Figure 3 in Section 2. The exact length of the data part can be derived from the standard.

By design, any changes in the underlying data directly reflect on both the data and the error correction part. The Reed-Solomon encoding is able to detect several changes in either the data or the error correction part and provides decoding to the original message even after a moderate number of modifications/errors have been introduced, i.e. if Q_i denotes the 100% correct QR Code for message M_i, then a QR Codes Q'_i with only minor deviations to the original code Q_i can still be decoded back to M_i. This feature, while designed to protect the integrity of the code as much as possible, serves as an important prerequisite for our attack: We don't need to change the meaning of the original QR Code Q_i to exactly match the QR Code $Q_j, i \neq j$ containing our manipulated QR Code M_j, we just need to reach a code Q'_j that is decoded to the same message. In traditional computer security, this is analogous to a NOP sledge.

Figure 6 illustrates the outline of this attack. The green circles denote QR Codes representing exact messages (i.e. containing no errors), the blue circles the set of erroneous QR Codes that get decoded to the same message due to the error correcting features of the Reed-Solomon code. F denotes the set of QR Codes that get detected as incorrect by the encoding but can't be corrected.

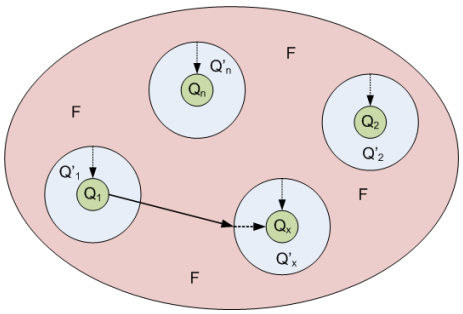

Figure 6: Attacking data codewords

4. QR CODES AS ATTACK VECTORS

We believe that manipulated QR Codes can be used for a plethora of attacks. Depending on whether the reader is a human or an automated program (e.g., in logistics), different scenarios are possible and outlined in this section.

4.1 Attacking Automated Processes

As QR Codes are a standardized way of encoding information we strongly believe that the majority of software developers do not treat the encoded information as possibly insecure input. As described in detail in the previous section, different parts of the QR Code could be manipulated in order to change the encoded information. Depending on the programs that process the encoded information, whether this would be in logistics, public transportation or in a fully automated assembly line, attacks on the reader software as well as the backend are theoretically possible. Without proper sanitation this could by used by an adversary for the following, non-exhaustive list of attacks. Similar attacks using RFID chips and SQL injections have been shown to be very effective [15], as input sanitation was not employed in these examples.

- **SQL injection**: We believe that many automated systems store and process the encoded information in a relational database. By appending a semicolon followed by a SQL query like `;drop table <tablename>` to the encoded information, manipulations to the backend database are possible (provided the DBMS allows for multiple queries in a single line). This would delete the table specified in the command, resulting in a denial-of-service attack. More specific attacks may include adding a user, executing system commands (e.g., by using the stored procedure `xp_cmdshell` on Microsoft SQL Server), or altering data such as prices or passwords within the database.
- **Command injection**: If the encoded information is used as a command line parameter without being sanitized, this could be easily exploited to run arbitrary commands on behalf of the attacker, which may have disastrous consequences for the security of the operating system e.g., installing rootkits, DoS, or connecting a shell to a remote computer under the control of the attacker.
- **Fraud**: Changes to the automated system can be used to commit fraud, by tricking the system e.g., into believing that it processes a cheap product A while processing the more expensive product B.

4.2 Attacking Human Interaction

Humans can not read the code without a reader software, the information stored within the code is completely obfuscated. But by reading the manipulated QR code, a vulnerability in the reader software or the browser might get triggered.

- **Phishing and Pharming**: If QR Codes are used for links in augmented reality scenarios, an attacker might

set up a fake website and redirect users by changing the QR Code. This is dangerous if some form of credentials are needed to access the website. The user has no possibility to verify that the link is not modified.

- **Fraud**: QR Codes are often used in advertisements to direct the target audience to special offers or additional information about specific products. If the QR Code can be manipulated to redirect the user to a cloned website, an adversary could sell the solicited product without ever fulfilling the contract. The victim implicitly trusts the advertising company by following the link.

- **Attacking reader software**: Different implementations of the reader software on computers or cell phones might be attackable via command injection or traditional buffer overflows if the encoded information is not sanitized. An attacker might gain control over the entire smartphone, including contact information or the victim's communication content like Email or SMS.

- **Social engineering attacks**: Building on these attacks, more specific attacks like spear phishing or other variants of social engineering are enabled, depending on the goal of the attacker. Leaving a poster of a QR Code on the parking lot of a company (instead of the traditional attack with an USB drive) offering discount in a nearby restaurant is a new attack vector which is likely to be successful.

4.3 Further Research

The possibilities for attacks proposed in this paper open up a quite large field for further research. The main target lies in the accurate analysis and practical application of one or more of the outlined attacks on a given target. Furthermore, it should be investigated which parts of a QR Code are the easiest to attack, and what countermeasures can be taken to thwart attacks like the ones proposed in this paper. In even more general terms, it would be very interesting to find metrics that can be used to measure the vulnerability of QR Codes depending on a given type of attack outline and with respect to characteristics like black/white-distribution, version, masking, etc.

Last but not least, other 2D-Codes such as Aztec [12] or DataMatrix [10] need to be analyzed in the same way to identify possible attack vectors and find suitable countermeasures.

5. CONCLUSION

In this paper we outlined the dangers of possible attacks utilizing manipulated QR Codes. Since QR Codes gain increasing popularity through their use for marketing purposes, we expect that this kind of attack will receive more and more attention by the hacking community in the future. Furthermore, many mobile devices (e.g., smartphones) at present are able to decode QR Codes and access the URLs contained in them. This adds a new dimension to the topic of trust, especially since most users are not security-conscious enough when using their mobile phones (which also enables the use of novel phishing techniques). In addition to phishing, a multitude of other attack methods, both against humans and automated systems, might be performed using QR codes. This especially holds true if proper input sanitization is not performed prior to processing the contained data.

6. REFERENCES

[1] H. S. Al-Khalifa. Utilizing qr code and mobile phones for blinds and visually impaired people. In *ICCHP*, pages 1065–1069, 2008.

[2] A. Alapetite. Dynamic 2d-barcodes for multi-device web session migration including mobile phones. *Personal and Ubiquitous Computing*, 14(1):45–52, 2010.

[3] R. Bose and D. Ray-Chaudhuri. On a class of error correcting binary group codes*. *Information and control*, 3(1):68–79, 1960.

[4] M. Canadi, W. Höpken, and M. Fuchs. Application of qr codes in online travel distribution. In *ENTER*, pages 137–148, 2010.

[5] J. Gao, V. Kulkarni, H. Ranavat, L. Chang, and H. Mei. A 2d barcode-based mobile payment system. In *MUE*, pages 320–329, 2009.

[6] J. Z. Gao, L. Prakash, and R. Jagatesan. Understanding 2d-barcode technology and applications in m-commerce - design and implementation of a 2d barcode processing solution. In *COMPSAC (2)*, pages 49–56, 2007.

[7] J. Z. Gao, H. Veeraragavathatham, S. Savanur, and J. Xia. A 2d-barcode based mobile advertising solution. In *SEKE*, pages 466–472, 2009.

[8] A. Hocquenghem. Codes correcteurs dâĂŹerreurs. *Chiffres*, 2(147-156):4, 1959.

[9] Y.-P. Huang, Y.-T. Chang, and F. E. Sandnes. Ubiquitous information transfer across different platforms by qr codes. *J. Mobile Multimedia*, 6(1):3–14, 2010.

[10] ISO 16022:2006. *Data Matrix bar code symbology specification*. ISO, Geneva, Switzerland.

[11] ISO 18004:2006. *QR Code bar code symbology specification*. ISO, Geneva, Switzerland.

[12] ISO 24778:2008. *Aztec Code bar code symbology specification*. ISO, Geneva, Switzerland.

[13] S. Lisa and G. Piersantelli. Use of 2d barcode to access multimedia content and the web from a mobile handset. In *GLOBECOM*, pages 5594–5596, 2008.

[14] I. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[15] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. Is your cat infected with a computer virus? In *PERCOM '06: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 169–179, Washington, DC, USA, 2006. IEEE Computer Society.