EvilQR – when QR code goes bad
A security assessment of mobile QR readers

Chilik Tamir (www.appsec-labs.com)

Abstract:
Quick Response code, also known as QRCode has been around for several years, but in the last months there has been an incline in adoption of QRcodes as a marketing channel. A QRcode can encode a variety of information into a 2-dimentional barcode that is presented to the costumer. Customers are often referred by vendors into scanning QRCodes in order to receive coupons, discounts or other marketing media such as website, flash movie etc. The QRCode is parsed by QR-reader software on a mobile phone equipped with a camera. The true nature of QRcode content is an enigma until it is scanned; there is no possibility for the customer to authenticate the content of a QRcode without scanning it first. Because of the latter fact, an attacker with evil intent could craft a malicious QRCode (or **EvilQR**) and lure an innocent customer to scan it. Once scanned, the evilQR would be parsed by the customer mobile phone software and would initiate its' attack. Attack vectors could vary from browser-based such as Cross-Site-Scripting (XSS) to specific buffer-overflow and command injection. The key for a successful attack lays in the default behavior of the mobile QRCode reader software. If as an example, a QRCode reader parses a link from an evilQR and preforms a URL redirection without proper confirmation of the customer - the attack would succeed. In this assessment we have compared the default behavior of several QR-readers for and noted their behavior upon the parsing of two evilQRs. Best practices for mobile users are also discussed.

The problem:
An innocent customer can be easily tricked into scanning a malicious-crafted QRCode (**evilQR**) by an attacker, upon scanning the customer mobile would be attacked by the encoded payload.

Motivation:
The motive for executing such attack is very clear - the mobile phone is a gold mine for an attacker, because today's phone contains very sensitive information such that can be abused by an attacker in several ways:

- Personal information compromised by an attacker could lead to impersonation, fishing and identity theft
- Calendar and meetings compromised by an attacker could lead to business or other information disclosures.
- Address book compromised by an attacker could lead to impersonation, fishing and identity theft
- Private and Cooperative email access compromised by an attacker could threaten internal business IT infrastructure.
- Geo-location compromised by an attacker could lead to harassment, surveillance and privacy loss

- Connectivity – (3G, GPRS, Wi-Fi, Blue-Tooth, etc.) could enable the attacker to remote control his attack
- Credit card information compromised by an attacker
- Social networking accounts (Twitter, Facebook, Path, LinkedIn, etc.) compromised by an attacker could lead to defacement, impersonation fishing and identity theft

Assessment:
Our assessment goal was to verify that QRCode reader software will not process an evilQR payload without proper confirmation from customer. In order to perform the test two test cases were created:

a. JavaScript QRCode:
In the first test case we have encoded a simple java-script code into an evilQR. The java-script that was used was very simple – an alert message that is shown upon parsing. This test demonstrates a simple case of a Cross-Site-Scripting web attack (XSS). In this kind of an attack the customer web-browser is lured into executing malicious code on behalf of the customer current context and permissions. The object of this test case was to test the autonomous parsing capabilities of the QRCode reader software. If the QRCode reader software executes the java-script code without proper confirmation of the customer – the test is regarded as failed, whereas if the QRCode reader software executes the java-script code only after customer notification – the test is regarded as success.

evilQR 1: QRCode with payload: hello<script>alert('XSS')</script>hello

b. Web link to a malicious site:
In the second test case we have encoded a simple web link into a QRCode. The web link refers to http://www.appsec-labs.com as an example for an evil website. This test demonstrates a simple case of a fishing web attack. In this kind of an attack the customer web-browser is lured into visiting a malicious website that will attack the customer. The object of this test case was to test the autonomous website redirection capabilities of the QRCode reader software. If the QRCode reader software performs redirection to the encoded website URL without proper confirmation of the customer – the test is regarded as failed, whereas if the QRCode reader software executes the website redirection only after customer notification – the test is regarded as success.

evilQR 2: QRCode with payload: http://www.appsec-labs.com

In hope to shed light on the likelihood of this attack, we have chosen fourteen different QRCode reader applications, and kept their setting to the default. For each application we performed two scanning cycles. The first was aimed to test the autonomous java-script parsing of the QRCode reader application using the first test case. The second was aimed to test the autonomous parsing of website URLs by the application.

Results:

The QRCode reader assessment comparison chart is shown below (Table 1). We can learn that from the selected applications only one was found vulnerable to java-script evilQR (QuickMark). Furthermore, we can deduce that about 35% of the applications that were used were found vulnerable to direct website redirection. These results confirm our prior assumption that QRCode reader application may be used to introduce a malicious evilQR and to inflict an attack on an unaware customer. What more can be learned from the table below is the fact that the current QRCode reader applications parsing of java-script is not yet fully supported by the majority but could be but could be in the near future.

Application	Test a: java-script parsing	Test b: website redirection
TapReader (TapBase LLC)	No Parsing	User confirmation
QR+ (Alexandr Balyberdin)	No Parsing	User confirmation
QRReader (Tap Media Ltd)	No Parsing	Automatic Redirection
Scan (QR Code City, LLC)	No Parsing	Automatic Redirection
RedLaser (Occipital, LLC)	No Parsing	User confirmation
i-nigma (3GVision Ltd)	No Parsing	Automatic Redirection
BeeTagg (connvision AG)	No Parsing	User confirmation
QR Code Reader (ShopSavvy, Inc.)	No Parsing	Automatic Redirection
QuickMark (SimpleAct Inc.)	JavaScript Execution	Automatic Redirection
QR+Emoji (Ching-Lan Huang)	No Parsing	User confirmation
Bakodo (Dedoware Inc.)	No Parsing	User confirmation
Optiscan (Airsource Ltd.)	No Parsing	User confirmation
QR-Scanner (Grip'd LLC)	No Parsing	User confirmation
quiQR (Mark Tholking)	No Parsing	User confirmation

Table 1: Comparison table of application performance in two tests

From these results we can confirm that the evilQR attack vector is indeed a widespread phenomenon, and it should be taken into consideration by customer and application vendors.

Recommendations:

Many QR-reader software are delivered with default setting of the QR reader to interact with URI links automatically. This behavior may expose the mobile user into scanning an evilQR which will be parsed and trusted by the user's QR-reader software. As a general security recommendation to our customers follow these thumb rules:

a. You should choose a configurable QR-reader software that enables you to confirm QR-code output prior to its' acceptance.

b. Never scan a QR-code that has an unknown origin

c. You can check if your mobile QR reader is vulnerable by scanning the two evilQR above