Optical Delusions: A Study of Malicious QR Codes in the Wild

Amin Kharraz*, Engin Kirda*, William Robertson*, Davide Balzarotti†, Aurélien Francillon†
*Northeastern University, Boston, USA
{mkharraz,ek,wkr}@ccs.neu.edu
† Institut Eurecom, Sophia Antipolis, France
{davide.balzarotti,aurelien.francillon}@eurocom.fr

***Abstract*—QR codes, a form of 2D barcode, allow easy interaction between mobile devices and websites or printed material by removing the burden of manually typing a URL or contact information. QR codes are increasingly popular and are likely to be adopted by malware authors and cyber-criminals as well. In fact, while a link can "look" suspicious, malicious and benign QR codes cannot be distinguished by simply looking at them. However, despite public discussions about increasing use of QR codes for malicious purposes, the prevalence of malicious QR codes and the kinds of threats they pose are still unclear.**

In this paper, we examine attacks on the Internet that rely on QR codes. Using a crawler, we performed a large-scale experiment by analyzing QR codes across 14 million unique web pages over a ten-month period. Our results show that QR code technology is already used by attackers, for example to distribute malware or to lead users to phishing sites. However, the relatively few malicious QR codes we found in our experiments suggest that, on a global scale, the frequency of these attacks is not alarmingly high and users are rarely exposed to the threats distributed via QR codes while surfing the web.

***Index Terms*—Mobile devices, malicious QR codes, malware, phishing**

I. Introduction

Smartphones have been recognized as one of the most quickly emerging technologies in the recent years. They are gaining significant attention as they become more capable in providing unique services due to their powerful computing resources and sensor capabilities. On the other hand, increasing demand, evolving usage patterns, and all-in-one capabilities of smartphones make them lucrative targets for malware writers. Spyware, worms, rootkits, bots, and trojans are all on the rise in mobile platforms and are becoming more sophisticated as professional criminals leverage new exploitation and propagation techniques. For example, in 2012, the Toll Fraud malware family caused significant financial damage to users by subscribing them to premium SMS services. Malware writers are constantly focusing on improving their attack strategies to expand the range of malicious actions they can deploy while evading emerging security defenses. These malicious actions constitute a spectrum that ranges from launching simple phishing sites to transforming compromised machines into bots.

Quick Response (QR) codes are an evolution of regular one-dimensional barcodes that are frequently used with mobile devices. While 1D barcodes encode data with bars of different width, 2D barcodes usually encode data as a matrix of dots. Several 2D barcode standards exist, but the QR Code ISO/IEC standard [1] is certainly the most commonly used on smartphones. A QR code include three square markers at the corners, called *finder patterns (FIP)*, designed to be easily identified by image recognition software. These patterns were designed to be easily scanned with a camera and decoded using simple image processing algorithms. The main advantage of QR codes when compared to regular barcodes is their increased information density (Figure 1).

In Q4 2011, 20.1 million users in the United States used their smartphones to scan QR codes [2]. This popularity has not only attracted business owners, but also users with malicious intentions that seek to gain revenues by abusing this new technology. Indeed, while QR codes are convenient to use, they are also opaque and can therefore be used to hide the final destination of a link.

In September 2011, malware that spread through QR codes on a Russian website was reported in the news [3]. The corresponding QR code directed victims to download a version of the Jimm-mobile ICQ client, infected with the `TrojanSMS.AndroidOS.Jfake.f` malware, that sends SMS messages to premium rate numbers [4]. The potential use of QR codes to perform malicious activities (e.g., directing to exploit sites or phishing sites, downloading malicious contents) has been discussed in a number of security blogs and forums [3]–[6]. Furthermore, several proof-of-concept QR code attacks have been posted on public security blogs and social media [7]–[9]. However, the prevalence of these attacks is still unclear.

In this paper, we perform a large scale study of threats posed by QR codes in the wild. We design a web crawler that retrieves images, detects QR codes, and checks their destination to identify malicious QR codes. We first assess the current prevalence of QR codes in the wild, and then perform an analysis on the extracted QR codes. This allows us to obtain more insights into the global usage of QR codes on the web. Malicious QR codes are identified by extracting URLs from the codes and comparing them to malicious domain blacklists. We conducted our crawling experiments over a ten month period starting in December 2012, seeding the crawl using Google searches as well as the 1,500 most popular websites published by Alexa [10].

(a) A link to the DSN 2014 CFP.

(b) Example of logo insertion, abusing error correction.

(c) Example with higher data density: DSN deadlines as free text.

Fig. 1: Examples of QR codes. If scanned, such QR codes could be used to deanonymize a reviewer that was not careful to check the extracted URL!

We performed an empirical analysis across 14 million web pages to discover the extent to which QR codes are leveraged by attackers in the wild. Our results show that QR codes are already being abused by attackers to distribute malware or direct to phishing sites on the public web. Based on our analysis, 0.16% of the QR codes we analyzed were designed to facilitate one or more types of malicious activities, such as directing users to phishing sites or distributing malware to vulnerable devices. Therefore, our findings confirm the folk wisdom that such attacks are real [11]. However, we also observed that, on a global scale, QR codes are not currently extensively used by attackers on the Internet. This contradicts recent discussions in security blogs about the increasing use of malicious QR codes [3]–[6]. Based on our analysis, we conclude that the probability of exposure to the threats introduced by QR codes is currently small.

In summary, this paper makes the following contributions.

- We design an automated system to detect malicious QR codes on the web and to determine the types of threats they might pose.
- We provide insight into the security and privacy issues of QR codes on the public web. Our research provides a large-scale measurement of QR code attacks in the wild. We have extracted and analyzed more than 94,000 QR codes to investigate the global statistics of QR codes in different parts of the web.
- We characterize QR code attack strategies and their current prevalence on the web. Our results reveal that using phishing sites and exploit sites in QR code attacks are more prevalent than other attack strategies in the wild. We also identify some malware families distributed via QR codes.

The rest of the paper is structured as follows. Section II introduces QR code attacks and possible attack scenarios. In Section III, we present our methodology, image crawler, and malicious QR code detector. In Section IV, we present the experiments we conducted and discuss our findings. We discuss possible defenses against QR code-based attacks in Section V. In Section VI, we present related work. Finally, we conclude the paper in Section VII.

II. QR Code Attacks

The increasing popularity of QR codes as a mobile media element can make QR codes attractive targets for malware authors. However, attacks relying on QR codes are relatively new [3], [12]. We define a QR code-based attack as an attack that attempts to lure victims into scanning a QR code that directs them to malicious websites. The key idea behind QR code attacks is that victims might trust the web page or the printed material on which the QR code is displayed, and assume that the associated code is harmless. Typically, a user scanning a malicious QR code is directed to an exploit or to a phishing site. In the rest of this section, we discuss in more detail a set of realistic attack scenarios.

A. QR Codes Leading to Phishing Sites

Phishing attacks have received a considerable amount of attention because they are relatively simple and efficient. A phishing attack relies on both technical deception and social engineering techniques. The attacker must persuade the user to perform a series of actions that provide access to confidential information. The attack relies on the fact that a large number of users judge a website's legitimacy by its look and feel, which can be easily copied by an attacker [13]. Therefore, a phishing attack often starts by impersonating a popular website to abuse user trust.

Because of the properties of QR codes (e.g., easy generation, distribution, and opacity), their adoption can increase the user's vulnerability to phishing attacks for three main reasons. First, since with QR codes URLs do not need to be manually entered anymore, users might not pay attention to the addresses they are directed to. As shown by Onarlioglu et al. [14], in a normal situation users might be able to distinguish a benign URL from its misspelled counterpart. Unfortunately, the scenario is totally different when the user is directed to a specific website via a QR code. To make things worse, mobile operating systems typically allow websites to hide their URL once the page is loaded. This is intended to improve usability on small screens, but this feature can also be used to deceive users redirected to a phishing website.

Second, because of limited screen size, mobile browsers cannot display very long URLs. Therefore, a phisher can construct a long URL that starts with a legitimate name as part of the URL, but actually points to a different domain. Here again the effectiveness of the attack is improved because, thanks to the QR code, the user might never see the complete URL.

Third, miscreants can use a combination of QR codes and URL shortening services to hide the malicious URL and avoid

suspicion. As a result, unsophisticated users may be tricked by attackers [15] to visit a web page even after observing the corresponding short URL.

B. Malicious Software Distribution

Attackers often use malicious websites to distribute malicious software and perform drive-by download attacks [16]. In a recent drive-by download attack aimed at Android smartphones [17], malicious links were posted on online social networks to redirect victims to a malicious page. This page was designed to infect Android smartphones with malware (a variant of Android OfFake). This malware then connected to a Command-and-Control (C&C) server to join a botnet, allowing the attacker to execute arbitrary commands and exfiltrate personal information from the phone.

Although there has been no report of QR codes used in drive-by download attacks in the wild, the adoption of QR codes together with drive-by download attacks is a growing concern [5]. For example, attackers might deceive victims into scanning a malicious QR code leading to a previously compromised website hosting an exploit kit.

Attackers use a variety of techniques in websites serving malicious code. Such code is often hosted on legitimate websites, in a hidden HTML `iframe`, or in obfuscated JavaScript code leading the victim to the server hosting the exploit kit. This redirection is often done in two steps: first, the exploit kit first fingerprints the victim's device and then, based on the information retrieved, serves a relevant exploit to the device.

Figure 2 presents a simple scenario of a QR code attack in which the victim is redirected to an exploit site after scanning the QR code and visiting the compromised website. The appropriate malicious code is automatically loaded by fingerprinting the OS version and identifying vulnerable applications on the victim's device. In order to launch successful attacks, attackers often target vulnerable client applications such as web browsers, Adobe Reader, and Adobe Flash, exploiting them using common memory exploitation techniques. For example, Flash is being actively exploited in the wild in earlier versions of the Android platform by attackers that embed malicious SWF objects in HTML code.

III. Methodology

Our aim is to analyze the prevalence of malicious QR codes on the web. For this purpose, we built a tool that: *a*) crawls the web and extracts image files; *b*) searches for QR codes in the extracted images and extracts from them any URLs discovered; and, *c*) identifies malicious URLs obtained from the QR codes.

A. System Architecture

In the following, we describe in more detail the architecture of our image crawler and the mechanism we used for detecting malicious QR codes.

1) Image Crawler Engine: The crawler engine relies on the scrapy framework [18] to extract URLs within each web page visited and append them to the crawler queue. Although we only crawl publicly viewable contents, our crawler processes `robot.txt` files and therefore complies with the Robots Exclusion protocol [19]. When a URL is disallowed, it is removed from the queue; otherwise, the URL is passed to the image crawler. The image crawler parses each page and extracts images using XPath (e.g., `//img/@src`). For each image we keep some metadata, including the download path, the primary referring URL, and the image checksum. The URL of a scraped image is used to schedule downloading of the image. Successfully extracted images are then stored in a MongoDB database along with their metadata.

2) QR Code Extractor: We use an open source QR code decoder [20], which provides a Python interface to a Java QR decoding library. While there are several QR code decoding libraries available, we chose it over other implementations because it was faster and allowed our system to scale to a larger number of images. The QR code extractor retrieves images from the database and attempts to find the three finder patterns in each image. If the special sequence of black and white pixels patterns are not found within an image, it presumes that the retrieved image is not a QR code and retrieves the next image to process from the MongoDB server. QR code metadata, including the URL of the website from which the QR code was extracted and any extracted target URL, is inserted into the image collection.

3) URL Matcher: URL matching is the final phase of detecting malicious QR codes. It compares the URLs extracted from the QR codes to a list of malicious websites created from a number of publicly available resources. To populate this list, we deploy a crawler to collect domain feeds provided by major URL blacklists (PhishTank [21], Malware Domains [22], the Malware Domain List [23], malc0de [24], Malware Block List [25], and vxvault.siri-urz [26]). Our system collects on average 1,600 entries per day from these sources. From these feeds, we generated a collection of approximately 640,000 unique malicious URLs to perform URL matching. The URLs obtained from QR codes are then compared with this list to identify malicious QR codes and the type of malicious actions for which they were intended.

4) Web Crawling: A suitable selection of seeds, or starting points for the web crawler, is an important precondition for an efficient and high-coverage crawler. We therefore selected a combination of sites from various sources in order to extract images from different categories of sites and increase the chances of finding possible attacks. In particular, we selected the initial crawling seeds from six different categories, including *free downloads*, *online games*, *adult*, *music*, *online news*, and *personal/business* websites. For each category, we extracted the results of English language Google searches. We also included the top 1,500 most popular websites as published by Alexa [10].

In addition, we added to the list some websites that were more likely to contain malicious QR codes. For this, we

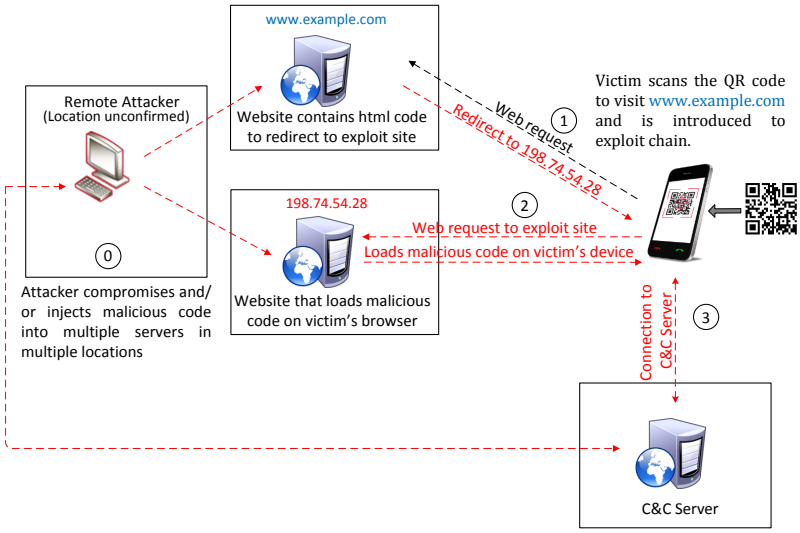


Fig. 2: A typical scenario of a QR code attack.

also included a set of randomly- selected malicious websites from the repository described in Section III-A3. Since we performed the experiments over three separate time intervals, we created independent initial seeds for the crawler in each experiment to cover different parts of the web. For the first two experimental runs, we chose a set of top-level pages as crawl seeds and restricted the crawler to a depth of three to balance the coverage of web pages across multiple domains. In the last experiment, we lifted the link depth restriction in our crawling policy to cover more web pages of the crawled domains.

B. Limitations

Our study has a set of limitations due to our detection techniques, the nature of the web, and QR codes themselves. First, we cannot extrapolate how real users might react to these attacks or what types of scenarios are more dangerous from the user perspective, since this would require performing a behavioral analysis of the users when they are exposed to real attacks. Second, our results are based on a sampling of web pages from different categories which is not necessarily representative of the entire web. We tried to explore what we believe are interesting parts of the web from an attacker's perspective. However, we cannot prove that this is a sample of what most users encounter while browsing the web or those categories as a whole. Finally, since our current detection function cannot recognize QR codes that are a smaller portion of a larger image, some QR codes might fail to be detected by our current approach. Similarly, QR codes might be found less frequently in complex formats (e.g., videos, flash animations), which we do not currently analyze for performance reasons. Thus, there might be other scenarios or techniques that employ QR codes to attract unsophisticated users that we did not recognize, although this last limitation could be ameliorated by a more sophisticated detection function.

Despite these limitations, our study is a first step forward in analyzing malicious QR codes in the wild and the threats they pose.

C. Ethical Considerations

We only crawled publicly available content on the web for our research and analysis. Our experiments were designed not to attack the websites we contacted, to expose vulnerabilities, or to jeopardize the security and privacy of the website operators or users. Furthermore, our crawler was configured to comply with the Robot Exclusion protocol. Consequently, websites that disallowed crawling were automatically removed from the list of crawled URLs. Finally, in order to limit the load incurred by our crawler on the websites we contacted, we added an artificial delay between successive requests to the same server.

IV. Evaluation

Table I presents a summary of our crawling experiments since December 2012. In total, we crawled over 14.7 million URLs in three separate crawls over a ten-month period. Our crawler extracted 94,770 QR codes, approximately one for every 156 web pages that were visited. Many of those QR codes were extracted from websites that employed QR codes to promote products or services (e.g., electronics, special events) or that describe marketing strategies relying on QR codes.

Figure 3 shows how QR codes were collected over time. The sharp increase after month five is due to lifting the link depth restriction that was imposed up to that point in time. In total, we found 145 distinct malicious QR codes during the course of our measurements.

Figure 4 shows the distribution of how many links had to be followed to reach a QR code. Our observations suggest that in order to reach more than 95% of QR codes, a user needs

Results	Dec 2012 – Feb 2013	Mar 2013 – Apr 2013	May 2013 – Sep 2013	Total
URLs crawled	1,671,417 (11.34%)	2,269,309 (15.4%)	10,792,421 (73.2%)	14,733,147
Domains crawled	2,173 (12.7%)	2,847 (16.6%)	12,105 (70.6%)	17,125
Invalid pages	118,916 (17.1%)	142,721 (20.5%)	432,302 (62.2%)	693,939
Pages containing QR codes	9,413 (18.7%)	13,127 (27.5%)	47,622 (67.9%)	70,162
QR codes extracted	7,836 (8.27%)	9,571 (10.1%)	77,363 (81.6%)	94,770

TABLE I: Summary of crawling experiments performed from December 2012 to September 2013. 94,770 unique QR codes were identified across 14.7 million URLs.

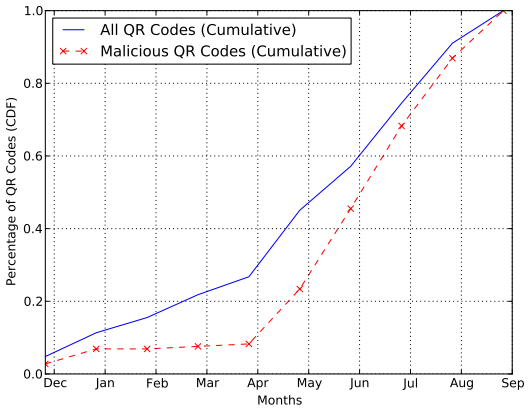


Fig. 3: CDF of benign and malicious QR codes found over time. The sharp increase in month five is due to lifting the depth restriction in our crawling policy.

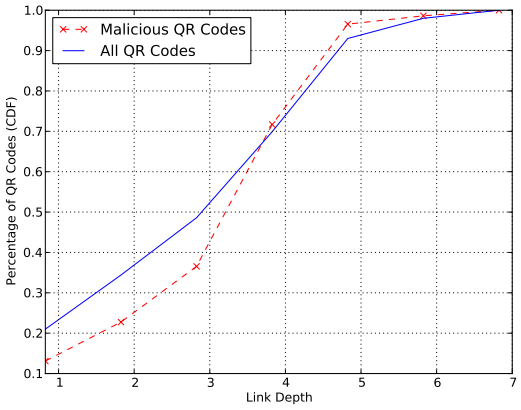


Fig. 4: CDF of QR code incidence versus link depth. The chance of reaching a QR code increases significantly if links of depth greater than three are followed.

to follow at least five links. By following links up to a depth of three, only 48% of all QR codes can be reached.

Figure 5 shows the types and proportions of malicious web objects found at each link depth. The results suggest that most of the malicious binaries, including APKs and EXEs, were injected in the first three depths. However, malicious web pages distributed by QR codes were mainly observed at depths greater than three.

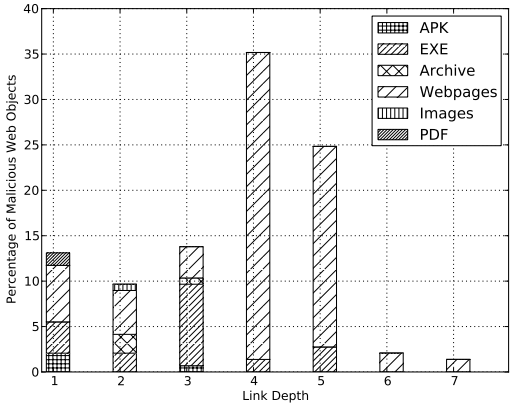


Fig. 5: Proportion of malicious web objects found in different link depths.

Web Categories	URLs	Domains	QR Codes
Free downloads	3,020,295 (20.5%)	3,014 (17.6%)	20,186 (21.3%)
Online games	2,106,840 (14.3%)	2,603 (15.2%)	15,258 (16.1%)
Adult	2,342,570 (15.9%)	2,877 (16.8%)	13,173 (13.9%)
Music	2,092,106 (14.2%)	2,722 (15.9%)	13,836 (14.6%)
Online news	2,224,705 (15.1%)	2,260 (13.2%)	13,267 (14.0%)
Personal/business	2,799,297 (19.0%)	3,647 (21.3%)	19,048 (20.1%)
Total	14,733,147	17,125	94,770

TABLE II: Distribution of identified QR codes among different categories.

We also found 28 malicious proof-of-concept QR code attacks on security blogs. We did not consider these as real QR code attacks, and excluded them from our malicious data set.

Our results indicate that there is some interest and activity related to malicious QR codes. However, the relatively low rate of these attacks implies that users are rarely exposed to threats spread by QR codes.

A. Discussion

Table II presents the total number of QR codes and their distribution among different web categories. To better evaluate the use of QR codes on the web, we present a distribution of objects referenced by identified QR codes in different web categories in Table III. We list the most and least frequent categories where the web objects were reached via QR codes. In order to label resources with categories, each object was downloaded and associated with the proper type by using file

Web Object	Most Freq.	% of Web Cat.	Least Freq.	% of Web Cat.	% of Total QR Codes
APK	Free downloads	206 (1.02%)	Adult	0 (0.00%)	368 (0.39%)
	Online games	47 (0.31%)	Online news	67 (0.05%)	
EXE	Online games	3,570 (23.4%)	Online news	959 (7.13%)	15,681 (16.5%)
	Personal/business	4,171 (21.9%)	Music	1,250 (9.10%)	
Archive	Online games	2,486 (16.3%)	Online news	497 (3.70%)	10,119 (10.7%)
	Adult	1,647 (12.6%)	Music	989 (7.20%)	
Web Pages	Online news	7,226 (53.7%)	Music	3,998 (29.1%)	36,703 (38.7%)
	Adult	5,854 (44.7%)	Free downloads	6,560 (32.5%)	
Images	Music	2,487 (18.1%)	Personal/business	514 (2.70%)	6,824 (7.20%)
	Online news	1,489 (11.1%)	Online games	564 (3.70%)	
Videos	Personal/business	4,247 (22.3%)	Adult	1,163 (8.90%)	17,577 (18.5%)
	Online news	2,974 (22.3%)	Free downloads	2,673 (18.2%)	
PDF	Free downloads	2,280 (11.3%)	Personal/business	40 (0.21%)	3,545 (3.70%)
	Online games	640 (4.20%)	Adult	27 (0.21%)	

TABLE III: Distribution of web objects spread via QR codes among different categories. 38.7% of extracted QR codes were used to direct users to other web pages. Some of the categories are not provided in this table. So the percentage do not necessarily add up to 100%.

extensions and contents. Comparing the results of different web objects shows that 38% of the extracted QR codes were designed to direct users to other web pages. We also noticed some interest in using QR codes to directly download executable artifacts. For example, more than 16% of identified QR codes were used to directly download Windows EXE files. We also found 368 QR codes during the course of our measurements that were used to directly download Android APK files. However, since some of the extracted URLs were dead or broken, we could only collect 277 APK files from the extracted URLs. 186 (67.1%) of the downloaded APK files were free versions of paid Android apps. 48 (17.3%) of the downloaded APK files were applications for managing smartphones or playing multimedia. We also found 43 (15.5%) APK files promoted as QR code reader apps for Android smartphones.

To answer the question of whether there exists a relationship between the type of web objects distributed via QR codes and different neighborhoods of the web, we performed a chi-square test on the data from Table III using a significance level of $\alpha = 0.05$. Based on the number of web objects in each category, the chi-square statistic ($\chi^2 = 13930.71$, df $= 30$, p-value $< 2.2 \times 10^{-16}$ which is less than α) indicates that the relationship between the type of web objects and different web categories does in fact exist.

Since a large proportion of the identified QR codes pointed users to other web pages (38.7%), we analyzed these QR codes to identify the places where users are typically directed to. We manually analyzed those found in the free downloads category to gain a better understanding of scenarios that exist on the web. Because we did not observe substantial differences between this category and others, we have not provided a similar detailed analysis for other categories. Figure 6 provides an approximate view of final destinations of QR codes in the free downloads category. Based on our data, the most common use of QR codes was to demonstrate the use of QR codes in marketing campaigns and advertisements. We classified this

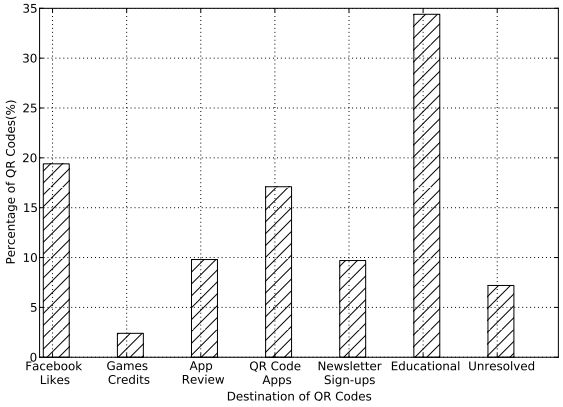


Fig. 6: Usage of QR codes in the free downloads category that direct users to other pages. QR codes were more frequently used for educational purposes.

type of QR code as *educational*, which comprised 34.3% of the identified QR codes. 19.4% of QR codes were designed to direct users to Facebook pages to "like" certain pages. We classified this type of QR code as *Facebook likes*. 9.8% of identified QR codes were used to direct users to online review websites to read other users' posts about an application before downloading it. We labeled this type of QR code *app reviews*. 17.1% of the identified QR codes were the output samples of particular QR code applications or links to download QR code reader applications. 9.7% of the extracted QR codes were used to encourage users to sign up for newsletters to receive news about discounted or free goods. 7.2% of the QR codes were pointing to domain names that were not possible to resolve. We also found a small portion of QR codes (2.4%) that were designed to direct users to buy game credits from legitimate websites.

During the course of the experiments, we also wanted to understand whether websites in certain web categories are used

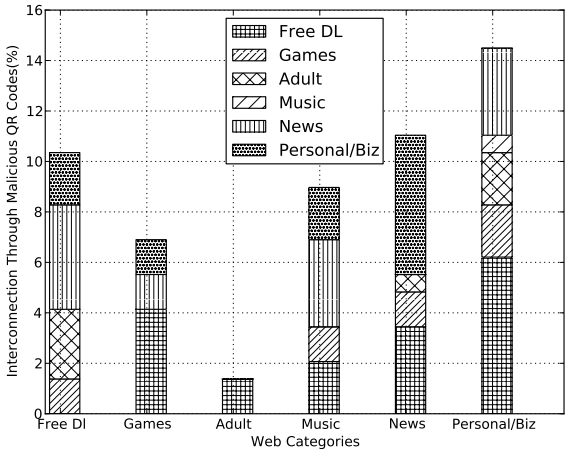


Fig. 7: Outgoing web category changes based on extracted malicious QR codes. For 53% of malicious QR codes, the category of the extracted URLs differed from the category where the QR codes were extracted. The largest fractions of category changes were observed between personal/business → free download, and news → personal/business.

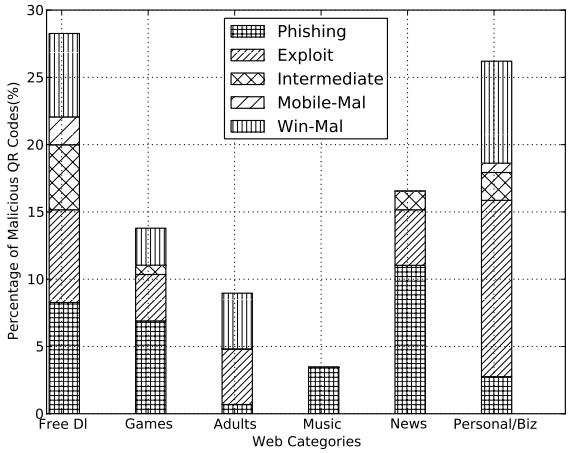


Fig. 8: Distribution of malicious QR codes across web categories. The free download and personal/business categories hosted the highest number of malicious QR codes. Phishing and exploits were the most prevalent attack types overall.

more often as landing pages for malicious QR codes. We were particularly interested in finding QR attacks in which a web category change could occur. We calculated this by analyzing how many malicious QR codes are used to interconnect web categories. We examined whether scanning a malicious QR code extracted from a certain web category brings the victim to a landing page in another web category resulting in an outgoing category change during the course of an attack.

Figure 7 represents the outgoing web category changes based on extracted malicious QR codes. Our analysis shows that for 53% of malicious QR codes, the category of the extracted URL differed from the category where the QR codes were originally extracted. For each category, we calculated the fraction of malicious QR codes that resulted in a category change. The personal/business category, with approximately 14% category changes, interacted with all other categories. We found some malicious QR codes posted by attackers in websites where people typically share posts or links that attempted to lure the readers visit download pages for certain web resources hosted in the free download category.

Our analysis also shows that miscreants use QR codes in distinct attack scenarios. In particular, we identified five main attack strategies used by attackers. Malicious QR codes can be designed to deliver malware, both for Windows and for mobile platforms, via direct download links, or to direct victims to phishing sites, intermediate sites, or exploit sites. We define intermediate sites as those with known vulnerabilities that are exploited by attackers to host malicious dynamic scripts. These sites are typically used to redirect users to phishing or exploit sites. Exploit sites are those that host exploit kits that are designed to automatically detect and compromise vulnerable applications on victim devices.

In all scenarios that we encountered, attackers employed QR codes to facilitate interactions with end users and abuse trust in known brands or websites. In phishing scenarios, the attackers leveraged QR codes to direct victims to fake versions of popular websites requiring the input of authentication credentials or payment information.

In the following, we provide more details on the types of malicious QR codes that were designed to launch phishing attacks or distribute malware.

B. Distributing Malware

1) Directing Users to Exploit or Intermediate Sites: Miscreants can deliver malware by directing users to exploit or intermediate sites. Attackers can inject malicious codes into intermediate websites to redirect the victim's browser to landing pages designed to initiate drive-by download attacks. We classified malicious QR codes by identifying the types of actions they were intended to perform after extracting their corresponding URLs. In scenarios where extracted URLs ended in `htm`, `html`, or `php`, we compared the URLs with our archive of malicious URLs (Section III-A3). We employed the original classification of the Malware Domain List which determined the type of malicious activity that the URL was designed to perform. We also performed manual inspection of the identified websites to determine how they redirect requests to exploit pages.

Figure 8 summarizes the attack scenarios that we observed in the wild. Directing users to intermediate sites accounts for 8.8% of all the attacks. These websites were mainly observed in the free download category. These websites provide a rich source of opportunity for attackers since they can be compromised to redirect traffic to exploit or phishing websites. The extracted URLs were typically websites running outdated versions of WordPress or plugins with known vulnerabilities. The

attackers compromised these websites and typically injected hidden `iframes` with obfuscated JavaScript that redirected the user to an exploit site (Figure 9). When a user visits such a web page, the malicious code included in the page silently redirects the browser to the exploit site. The malicious code is silently loaded into the victim's device, giving the attacker a foothold on the victim's system. We also identified three malicious QR codes that used intermediate websites to redirect victims to phishing sites.

Our results show that some attack strategies are more prevalent than others, in particular that miscreants mainly redirect users to *phishing sites* and *exploit sites* in QR code attacks. As depicted in Figure 8, while these two attack strategies are common in all categories, they mainly occur within the free download and personal/business categories rather than being uniformly distributed amongst all categories. We found that 32% of total QR codes directed users to exploit websites. We observed that the lifetime of this class of malicious QR codes was very short, and the corresponding URLs failed to be resolved after a short period of time. Only five of the extracted URLs were successfully resolved at the time of writing. Although the identified QR codes in this category were unique and contained different domain names, they all resolved to two IP addresses. This dynamic address generation characteristic can be considered as a strategy to prevent automatic systems from downloading exploits, and we hypothesize that they are part of the same malware campaign.

In a typical scenario, the web request to the site is directed to `attack.php` that serves up the payload containing the exploit kits. The page is valid only once and refuses incoming connections when a unique user sends multiple requests to visit the site in order to avoid multiple downloads of the malware being served. This makes downloading the malware for analysis more difficult as the malicious code cannot be obtained from a known malicious URL or one that has been visited twice. Figure 8 also provides an approximate view of the types of QR codes attacks that are more likely to occur in each web category.

We also identified some interest in distributing Windows malware. Our analysis shows that about 66% of Windows malware spread via QR codes was found again in the free download and personal/business categories.

We investigated whether the relationship between the attack strategies and web categories is statistically significant by again performing a chi-square test with $\alpha = 0.05$. The result of the test confirmed that the relationship between attack strategies and web categories holds ($\chi^2 = 52.6$, df $= 20$, p-value $< 9.38 \times 10^{-5}$ which is less than α).

2) Direct Malware Download: Figure 10 shows the distribution of the main types of malicious QR codes among the six web categories we defined in our experiments. In order to classify whether a downloaded program was malicious, we checked the MD5 file hashes using a set of AV products. This allowed us not only to identify individual samples, but also the malware families involved in QR code attacks. We identified 38 unique malware samples in five distinct malware families

```
if (document.getElementsByTagName('body')[0]) {
    iframer();
} else {
    document.write("<iframe src='http://exploit.com/
        links/column.php'
        width='10' height='10' style='visibility:hidden;
        position:absolute;
        left:0;top:0;'></iframe>");
}

function iframer() {
    var f = document.createElement('iframe');
    f.setAttribute('src', 'http://exploit.com/links/column.php');
    f.style.visibility = 'hidden';
    f.style.position = 'absolute';
    f.style.left = '0';
    f.style.top = '0';
    f.setAttribute('width', '10');
    f.setAttribute('height', '10');
    document.getElementsByTagName('body')[0].appendChild(f);
}
```

Fig. 9: A sample of malicious JavaScript code in an intermediate website that includes a hidden `iframe`.

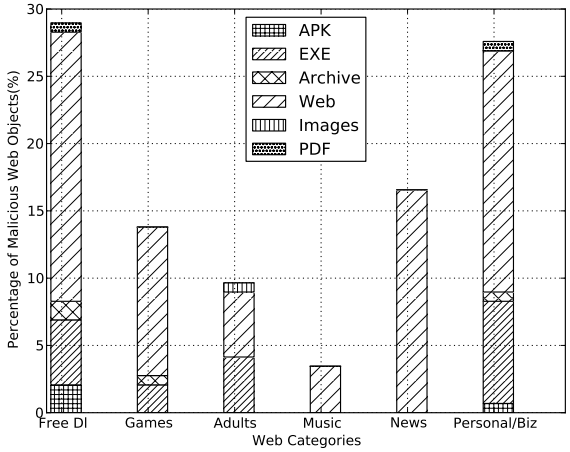


Fig. 10: Distribution of malicious web objects spread via QR codes across different categories. Distributing malicious websites via QR codes was observed as the top malicious activity across all categories.

(based on Kaspersky's malware labels).

Figure 11 lists the malware families distributed via QR codes. For each family, the number of samples and types of web resources that delivered the samples are provided. We did not observe a high diversity of malware families compared to the number of files we acquired during our experiments. However, it shows that QR codes are a possible attack vector that is used by some malware owners. We observed that distributing malicious EXE files via QR codes turned out to be more frequent than other malware types. As expected, scanning a QR code found on a website that can potentially redirect users to download a malicious EXE file requires special social engineering techniques. One scenario that we observed during our analysis was that the web page which hosted the malicious QR code was used to download copyrighted materials. The website attempted to trick unsophisticated users by first requiring them to prove they were not automated

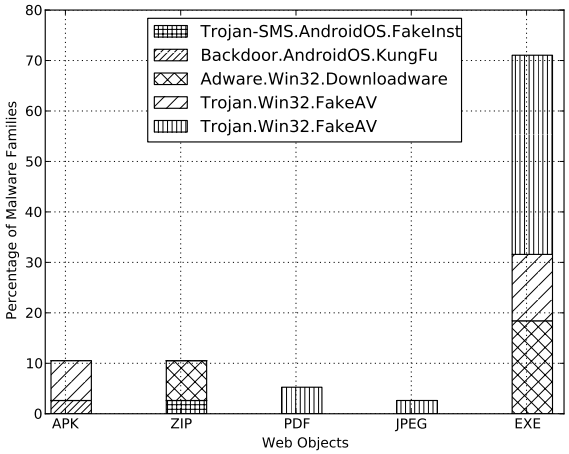


Fig. 11: Malware families distributed via QR codes across different web objects. Distributing malicious EXE files via QR codes turned out to be significantly higher than other malware types. To download copyrighted materials, victims had to scan malicious QR codes in order to prove they were not automated machines.

machines by scanning the QR code to be able to download the copyrighted materials. The victims had to resolve the QR code to reach the download link. We also observed similar scenarios where attackers tried to deliver malicious APK files to victims.

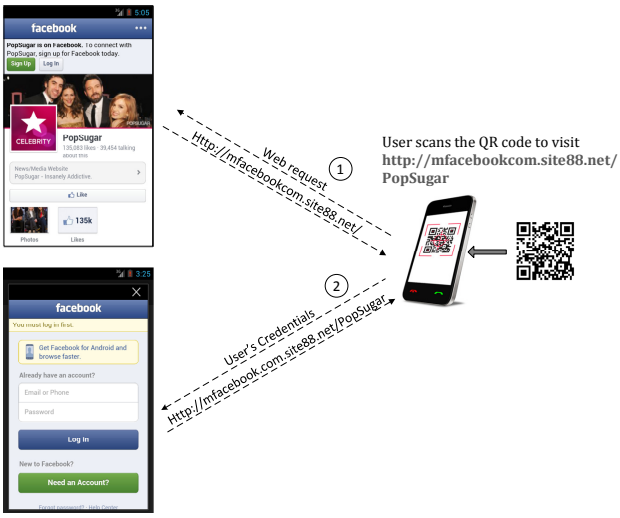
As depicted in Figure 11, 50% of the identified malware samples distributed via QR codes belong to the `Trojan.Win32.Generic` family that is primarily known for injecting malware into clean processes (e.g., Explorer.exe). QR codes were also used by the `Trojan.Win32.FakeAV` family to distribute fake antivirus programs. Once installed, the trojan prompts the victim to make a purchase from sites under the control of the attacker. We also found a few malicious QR codes that delivered the `Trojan-SMS.AndroidOS.FakeInst` family that is mainly used to send SMSs to premium rate numbers without the owner's knowledge or consent.

C. Phishing

1) Fake Business Websites: Fake business websites are categorized as a special type of phishing attack that can also be used to distribute malware. We identified 11 malicious QR codes that led users to fake versions of Google Play that offered malicious Android apps. In four cases, the corresponding URLs were designed to impersonate Google Play by prefixing the legitimate name at the beginning of the URLs, and by being long enough to prevent displaying the full URL on smartphone browsers.

Figure 12a presents a website from the free downloads category that we observed in our experiments. The website hosted a malicious QR code that contained a spoofed Google Play URL. This type of attack can be considered as a strategy to increase the chances of distributing malware or trojanised versions of popular apps by directly targeting unsophisticated users rather than compromising vulnerable websites.

(a) A real-world malicious QR code that encourages users to download an app from a fake Google Play site.



(b) A typical scenario abusing "liking" a page. A user first scans the QR code which redirects his browser to a fake Facebook page. If the user likes it, the browser is redirected to a fake login page to enter his credentials.

Fig. 12: Examples of QR code-based attack scenarios.

2) Fake Password-protected or Payment-related Websites: 37 QR codes redirected users to fake password-protected or payment-related websites aimed at obtaining user credentials or payment information. In these scenarios, attackers launch attacks by mimicking the appearance and behavior of legitimate sites. Consequently, inattentive users will not be suspicious when they are requested to enter their password. Facebook was identified as one of the most abused brands for phishing attacks among the attacks we observed. In particular, linking users to fake versions of Facebook accounts for 48% of total phishing attacks.

In order to launch a phishing attack, attackers typically encourage victims to like a page on Facebook via a QR code and invoke a fake Facebook web page that requests

TLD	Incidence	Percentage
com	61	42.1
net	17	11.7
org	11	7.60
biz	9	6.21
ru	19	13.1
br	13	8.96
info	2	1.38
Others	13	8.97

Country Code	Incidence	Percentage
US	49	33.8
Russia	21	14.9
Netherlands	19	13.1
China	16	11.0
Brazil	14	9.66
Germany	12	8.28
France	10	6.90
Vietnam	4	2.76

TABLE VI: Top seven TLDs and countries originating QR code attacks. US websites hosted 33% of malicious QR codes in our experiments. In line with other malware measurement studies, we speculate that the disproportionately large US presence is due to the attacker's reliance on its relatively robust Internet infrastructure.

users to enter their password. Figure 12b shows a typical scenario where users are encouraged to "like" a page on a fake Facebook page reached from a QR code and, in a second step, are required to enter their credentials.

Our system recognized seven malicious QR codes in this category that were used as Facebook profile pictures. The attackers posted status updates on websites encouraging users to scan the QR code and like a Facebook page to receive notifications on special savings offers. This provides a potentially large victim population for the attackers since they can directly interact with victims by easily distributing their update posts among popular discussion groups.

Based on our experiments, Paypal was identified as the most phished brand in payment-related phishing attacks. We found 11 QR codes in our experiments that referred users to websites for online purchases – e.g., a fake Paypal website. Once a victim scans the malicious QR code, she is directed to a fake checkout page that requires the user's credentials to proceed. We also found three QR codes that directed users to fake versions of other financial services such as Cielo, the largest Brazilian credit and debit card operator. Though some posts in popular discussion groups indicate that the security community has expressed concerns about QR code-based payment systems [5], we did not find any reports showing that this type of malicious QR code has actually been identified in the wild before.

Table VI represents an overview of the most frequent domains used in QR code attacks as well as the countries of the associated IP addresses. Although the countries are widely spread, the number of malicious QR codes associated with US IP addresses is at the top of the list by a wide margin. Our analysis also shows that generic top-level domains (e.g., .com, .net, .org) are the most abused TLDs and account for 61% of URLs extracted from malicious QR codes. Furthermore, country name top-level domains such as .ru and .br are significantly used compared to other country domain spaces. We did not observe a significant use of URL shortening services to launch QR code attacks, finding only five malicious QR codes that abused tinyurl.com to hide the malicious links.

V. Proposed Defenses

The great advantage for an attacker in using QR codes to distribute malicious URLs lies in the inability of users to distinguish benign URLs from malicious URLs via visual inspection of a QR code. Therefore, a first step towards preventing QR code-based attacks is to ensure that users are at least given the chance to inspect a decoded URL prior to directing a browser to the URL.

However, many popular QR code-capable mobile applications already follow this practice. For instance, at the time of writing, after scanning a QR code, the ZXing Barcode Scanner [27] will present a visual overlay depicting the particular barcode scanned, the data (including any URL) extracted, as well as other metadata such as the code type and timestamp. For the careful and attentive user, this information is likely sufficient to help determine whether the data contained therein presents a threat.

However, as research and experience has demonstrated, users are not necessarily always able to distinguish between legitimate and malicious URLs [13]. Attackers have developed an impressive array of techniques to disguise malicious URLs using techniques such as domain prefixing, where attackers take advantage of space-limited browser URL bars by prefixing attacker-controlled domains with subdomains mimicking legitimate domains (e.g., www.paypal.com.attacker.com), or IDN homograph attacks [28] where Unicode characters are used to generate domain names that appear visually similar to legitimate domains (e.g., paypal.com, where a Cyrillic "a" has been used instead of an ASCII "a" in this document).

In response, it has become common to embed an automated check for malicious URLs directly into browsers – e.g., Google Safe Browsing. In the case of malicious QR codes, we suspect that a similar defensive tactic would also prove useful. In particular, we advocate for the integration of domain or URL blacklists such as those we have used for our experiments to detect malicious QR codes on the public web into mobile applications that decode QR codes. Integration of blacklist checks into these mobile applications would allow for immediate feedback to the user as to whether a decoded URL might be malicious, and would allow the user the opportunity to avoid browsing to a potentially unsafe website.

Another possible solution might be to depend upon similar blacklist checks to be integrated into mobile browsers or other applications that load content from potentially untrusted URLs decoded by QR code decoders. We note, however, that as of the time of writing, many mobile browsers, including Google Chrome for Android, do not incorporate malicious domain

blacklist checks.

VI. Related Work

Attacks against mobile devices have been a subject of intense scrutiny in recent years. Several studies have been been performed to measure and characterize the nature of the threat against mobile devices in terms of mobile-specific malware [29], [30]. Researchers have also investigated a number of approaches for detecting or preventing security and privacy violations against mobile devices and the sensitive user data stored on them [31]–[33]. One effort has studied user susceptibility to malicious QR codes physically posted in public areas [34]. Furthermore, current status of existing QR code scanners in terms of their detection of malicious URLs have also been investigated [35]. However, to our knowledge, prior work has heretofore not measured the extent of the threat posed by malicious QR codes on the public web as a vector for attacks against mobile device users.

Malicious QR codes share some characteristics of other web security issues, where attackers attempt to disguise malicious links as benign links in order to trick unsuspecting users into exposing themselves to attacks. The class of attack that is closest in spirit to malicious QR codes is the distribution of malicious links using URL shortening services, which has been studied in several efforts [15], [36]. However, QR codes pose an intrinsically different problem, in that while URL shorteners necessarily require some interaction with a remote service that has the opportunity to screen URLs for malicious behavior, QR codes are decoded on the mobile device itself. However, as we mention in Section V, integrating existing URL blacklisting services into QR code applications might help ameliorate the threat posed by malicious QR codes.

In general, mobile devices have lagged behind traditional desktop UIs with respect to helping users avoid attacks. For example, Dhamija et al. investigated the causes for users succumbing to phishing attacks by studying the user interface elements of web pages and browsers in order to highlight errors that users might make when identifying malicious websites [13]. Also, Niu et al. identified several vulnerabilities in the iOS browser interface that increases the chances of successful attacks by performing user studies demonstrating the users failed to recognize spoofed browser UI elements [37]. In the same vein, Rydstedt et al. presented framing attacks on websites displayed in mobile phones [38]. By spoofing the iOS browser address bar, they were able to perform "tapjacking" attacks, thereby motivating the necessity of using frame busting in mobile websites.

A number of studies have demonstrated that aside from defenses against low-level attacks, robust UI mechanisms for informing users of potential security and privacy violations are lacking on traditional platforms [13] as well as mobile devices [39]. Malicious QR codes can be viewed as yet another facet of the problem of providing accurate and understandable security indicators to users.

VII. Conclusion

In this paper, we carried out a large-scale experiment to estimate to what extent QR codes are currently used for malicious purposes. We performed our analysis on over 14.7 million unique web pages. Our analysis shows that QR codes are already being used by cyber-criminals in the real world. However, our analysis shows that only 145 of the 94,770 QR codes we extracted exhibit evidence of malicious intent. The low observed base rate of malicious QR codes suggests that users are rarely exposed to QR code-based attacks, contradicting recent claims in security blogs surrounding the increasing use of malicious QR codes. Additionally, we did not find any significant change in the frequency of launching QR code attacks in our experimental runs to draw a concrete conclusion about the future use of QR codes for malicious purposes.

We found 48 malicious QR codes in our data set that were designed to direct victims to phishing sites. Spoofing password-protected websites for social sharing (Facebook) was observed as the most prevalent form of QR code-based phishing attack. We also found 11 malicious QR codes that directed victims to a fake version of the Google Play app market. Consequently, users who typically scan QR codes to download an app or link to online social networks are at higher risk of exposure to a QR code phishing attack.

Furthermore, 94 malicious QR codes were used to direct users to either exploit or intermediate sites, or to distribute malware via direct download links. We observed that the lifetime of this class of malicious QR code was very short, and the corresponding URLs failed to be resolved after a short period of time. Only five QR codes were successfully resolved from this class. Our results also indicate that QR code attacks are most often found on free download and personal/business websites.

Although we identified a relatively small number of malicious QR codes in the wild, we believe that defenses against such codes should be clearly defined. For example, enabling browsers to automatically resolve QR codes and perform URL reputation checks before visiting the URLs could help to prevent QR code-based attacks on mobile devices.

References

[1] International Organization for Standardization, "ISO/IEC 18004:2006 Information technology – Automatic identification and data capture techniques – QR Code 2005 bar code symbology specification," 2006.

[2] I. P. Release, "Smartphone market hits all-time quarterly high due to seasonal strength and wider variety of offerings, according to idc," 2012. http://www.idc.com/getdoc.jsp?containerId=prUS23299912.

[3] M. DeCarlo, "AVG: QR Code-based Malware Attacks to Rise in 2012." Techspot News. http://www.techspot.com/news/47189-avg-qr-code-based-malware-attacks-to-rise-in-2012.html.

[4] D. Maslennikov, "Malicious QR codes pushing android malware." Kaspersky Securelist Blog, 2011. http://www.securelist.com/en/blog?print_mode=1&weblogid=208193145.

[5] E. Chickowski, "QR Code Malware Picks Up Steam." Dark Reading Blog. http://www.darkreading.com/mobile-security/167901113/security/news/232301147/qr-code-malware-picks-up-steam.html.

[6] T. Wasserman, "New Security Threat: Infected QR Codes." http://mashable.com/2011/10/20/qr-code-security-threat/.

[7] The Jester, "Curiosity Pwned the Cat." Blog Post, 2012. http://jesterscourt.cu.cc/2012/03/09/curiosity-pwned-the-cat/.

[8] P. Roberts, "QR Tags Can Hide Malicious Links, Experts Warn." Kaspersky's ThreatPost Blog, 2011. http://threatpost.com/en_us/blogs/qr-tags-can-hide-malicious-links-experts-warn-091211.

[9] MasterButcher68, "Backtrack 5 R3-QRcode Attack with SET." http://www.youtube.com/watch?v=GeE5-6kTO_U.

[10] Alexa Internet, Inc., "Alexa." http://alexa.com/.

[11] D. Storm, "Hack or hoax? Jester brags of QR code smartphone attack against Anonymous," 2012. http://blogs.computerworld.com/19875/hack_or_hoax_jester_brags_of_qr_code_smartphone_attack_against_anonymous.

[12] G. Funaro, "Malicious QR Codes: Attack Methods and Techniques Infograph," 2011. http://usa.kaspersky.com/about-us/press-center/press-blog/malicious-qr-codes-attack-methods-techniques-infographic.

[13] R. Dhamija, J. D. Tygar, and M. Hearst, "Why Phishing Works," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (New York, NY, USA), pp. 581–590, ACM, 2006.

[14] K. Onarlioglu, U. O. Yilmaz, E. Kirda, and D. Balzarotti, "Insights into user behavior in dealing with internet attacks," in *19th Annual Network and Distributed System Security Symposium (NDSS 2012)*, 2 2012.

[15] F. Maggi, A. Frossi, S. Zanero, G. Stringhini, B. Stone-Gross, C. Kruegel, and G. Vigna, "Two Years of Short URLs Internet Measurement: Security Threats and Countermeasures," in *Proceedings of the International World Wide Web Conference*, ACM, 2013.

[16] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The Ghost in the Browser Analysis of Web-based Malware," in *Proceedings of the Workshop on Hot Topics in Understanding Botnets*, (Berkeley, CA, USA), USENIX Association, 2007.

[17] E. Protalinski, "A First: Hacked sites with Android Drive-by Download Malware." ZDNet Zero Day Blog, 2012. http://www.zdnet.com/blog/security/a-first-hacked-sites-with-android-drive-by-download-malware/11810.

[18] The Scrapy Project, "An Open-Source Web Scraping Framework for Python." http://scrapy.org/.

[19] robotstxt.org, "The Web Robots Pages." http://www.robotstxt.org/.

[20] Y. Yanbe, "Open Source QR Code Library." http://qrcode.sourceforge.jp/.

[21] OpenDNS, Inc., "The Phishtank Project." http://phishtank.com/.

[22] DNS-BH Operators, "DNS-BH – Malware Domain Blocklist." http://malwaredomains.com/.

[23] Malwre Domain List Operators, "The Malware Domain List Project." http://malwaredomainlist.com/.

[24] Malc0de Operators, "Malc0de." http://www.malc0de.com/.

[25] ParetoLogic, Inc., "Malware Black List." http://www.malwareblacklist.com/.

[26] VX Vault Operators, "VX Vault." http://vxvault.siri-urz.net/.

[27] ZXing Team, "Barcode Scanner." https://play.google.com/store/apps/details?id=com.google.zxing.client.android&hl=en.

[28] E. Gabrilovich and A. Gontmakher, "The Homograph Attack," *Communications of the ACM*, vol. 45, February 2002.

[29] Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," in *Proceedings of the IEEE Symposium on Security and Privacy*, IEEE Computer Society, May 2012.

[30] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A Survey of Mobile Malware in the Wild," in *Proceedings of the ACM workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 3–14, ACM, 2011.

[31] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "PiOS: Detecting Privacy Leaks in iOS Applications," in *Proceedings of the Network and Distributed System Security Symposium*, Internet Society, February 2009.

[32] W. Enck, P. Gilbert, B. gon Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, USENIX Association, October 2010.

[33] R. Xu and R. Anderson, "Aurasium: Practical Policy Enforcement for Android Applications," in *Proceedings of the USENIX Security Symposium*, USENIX Association, August 2012.

[34] P. Kieseberg, M. Leithner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, and E. Weippl, "QR Code Security," in *Proceedings of the International Workshop on Trustworthy Ubiquitous Computing*, January 2010.

[35] H. Yao and D. Shin, "Towards preventing qr code based attacks on android phone using security warnings," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, (New York, NY, USA), pp. 341–346, ACM, 2013.

[36] J. B. A. Neumann and U. Meyer, "Security and Privacy Implications of URL Shortening Services," in *Proceedings of the Workshop on Web 2.0 Security and Privacy*, 2010.

[37] Y. Niu, F. Hsu, and H. Chen, "iPhish: Phishing Vulnerabilities on Consumer Electronics," in *Proceedings of the Conference on Usability, Psychology, and Security*, (Berkeley, CA, USA), pp. 1–8, USENIX Association, 2008.

[38] G. Rydstedt, B. Gourdin, E. Bursztein, and D. Boneh, "Framing Attacks on Smart Phones and Dumb Routers: Tap-jacking and Geo-localization Attacks," in *Proceedings of the USENIX Workshop on Offensive Technology*, USENIX Association, 2010.

[39] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android Permissions: User Attention, Comprehension, and Behavior," in *Proceedings of the Symposium on Usable Privacy and Security*, 2012.