INFORMATION SECURITY CHALLENGE OF QR CODES

Nik Thompson
n.thompson@murdoch.edu
Tel: +61 893601285

Kevin Lee
kevin.lee@murdoch.edu.au
Tel: +61 893602059

School of Engineering and Information Technology
Murdoch University
Perth, WA 6150, Australia

ABSTRACT

The discipline of information security must adapt to new technologies and methods of interaction with those technologies. New technologies present both challenges and opportunities for the security professional, especially for areas such as digital forensics. Challenges can be in the form of new devices such as smartphones or new methods of sharing information, such as social networks. One such rapidly emerging interaction technology is the use of Quick Response (QR) codes. These offer a physical mechanism for quick access to Web sites for advertising and social interaction. This paper argues that the common implementation of QR codes potentially presents security issues that must be considered by professionals in the area. It analyzes potential privacy problems with QR codes and studies a range of devices as they may have implications for the processes and procedures used by Information Security professionals.

Keywords: QR codes, computer security, information security, digital forensics, quick response, smartphones.

1. INTRODUCTION

Information security is the domain concerned with protecting information systems from potential threats. Information security is commonly benchmarked in terms of the attributes of Confidentiality, Integrity and Availability (CIA). Information Security professionals are driven by ensuring that the information systems under their charge are protected in respect to these attributes. Practically, this means ensuring systems are trusted, privacy is maintained and information is always accessible.

To remain viable, the profession of information and computer security must keep abreast of changes in the increasingly interconnected digital world. In the domain of digital forensics, bodies such as Scientific Working Group on

Digital Evidence compile best practice documents to guide security professional (Scientific Working Group on Digital Evidence, 2013a). More recently, documentation of best practices has been extended to include devices such as mobile phones (Scientific Working Group on Digital Evidence, 2013b) and navigation systems as these widely used devices capture and store a large amount of personal and environmental information in their normal operation. New methods of interaction including personal social networks such as Facebook, photo and video sharing sites such as Flickr and YouTube are also increasingly capturing large amounts of information about users. The sheer scale, volume and pervasive nature of this data being accumulated impacts many information security domains including issues for digital evidence. Consequently new techniques have to be developed (e.g., Bell & Boddington, 2010; Piccinelli & Gubian, 2011) to extract, manage (Duranti and Endicott-Popovsky, 2010), and analyze this data.

A rapidly growing social interaction technology is the use of Quick Response (QR) codes, which are commonly used as physical shortcuts to Internet resources (see Figure 1). QR codes are matrix barcodes that were originally created in 1994 by Toyota subsidiary Denso-Wave to identify automotive components. The term QR code is a registered trademark of Denso-Wave Incorporated (Denso-Wave Incorporated, 2011); however the technology itself is open and free to use as it is published in ISO and JIS standards (International Organization for Standardization, 2006; Japanese Standards Association, 1999). QR codes are touted for their ease of use and convenience and are increasingly being used for marketing. This is commonly done by placing a QR code on an advertisement or poster, which when scanned with a mobile phone, directs the user to a Web site.

This paper highlights, clarifies and analyses the potential implications for information security of the use of QR codes. The remainder of the paper is structured as follows. Section 2 provides background on the technology and use of QR codes. Section 3 provides a discussion of related research. Section 4 highlights a series of research questions on information security issues with the use of QR codes. Section 5 presents a series of empirical investigations into a variety of issues related to these research questions. Section 6 discusses these research questions in detail in light of the empirical findings. Finally, Section 7 presents some conclusions.

2. QR CODES

QR codes are a rapidly growing technology for social interaction and advertising. The reason for this rapid uptake is the way in which they can provide a connection between the physical world and the digital world (e.g., Internet resources). In this role, they are increasingly being used in public spaces and on products to provide a bridge to Web sites.

In this method of usage, QR codes provide little more than a physical, machine recognizable representation of a hyperlink; appearing on business cards, posters, newspapers and even television advertisements. Typically, an individual uses their mobile phone camera to quickly capture the QR code which then directs them to a Web site. The user is presented with product information and is often asked for personal information. Marketers embrace QR codes as they allow them to target their advertising to particular groups of users and specific locations. Figure 1 illustrates a QR code for a simple information Web site (you may scan it with your smartphone QR reader). The fact that QR codes are machine-readable has the advantage of convenience (as little user involvement is required), however this brings with it many concerns for security, as the user is unable to ascertain the contents of the QR code prior to scanning.

Figure 1 An Example of a QR Code

The physical encoding of information in the QR code is covered by several standards, including JIS 0521 (Japanese Standards Association, 1999) and ISO/IEC 18004 (International Organization for Standardization, 2006). This is essential for the technology to be viable and interoperable. However at the application layer, no such standardization exists. Like many emerging technologies (especially Web based), the specifics of the implementation often differ greatly from platform to platform, and even vendor to vendor. A lack of standardization often has severe implications for the security of any device or platform. Users, vendors and security auditors alike must have confidence that their data and applications and privacy will be handled in a consistent, controlled and repeatable manner. The ad-hoc nature of QR processing applications does little to alleviate this concern.

Any individual or company can create QR codes by using simple Web-based generators that encode any text into its unique QR code representation. In fact, certain popular Web site redirection services now automatically generate a QR code for every Web site simply as a matter of course. QR codes typically hold around 50 characters, with newer more dense versions holding up to 1264 characters. This space is sufficient to allow the encoding of information such

as the QR code location (e.g., poster location); the use of URL shortening services makes it possible to encode longer URLs than would strictly be possible within a particular QR version.

As noted above, the non human-readable nature of the QR data has implications for the trust of a Web resource being accessed. Furthermore, the widespread use of URL shortening services also serves to further obscure the destination URL of a link. These issues undermine the inherent trust associated with when a user manually enters in the address of a site they wish to visit. This opens the door for malicious users to inadvertently divert traffic to their Web sites, giving the user little or no forewarning that this is occurring. A user interface redress attack is a common technique of tricking users into clicking something other than what they originally intended. This may cause the user to unwittingly reveal personal information, open security holes in their system or even unintentionally buy products online. For the user interface redress technique to work, the actual contents of a button or link have to be concealed somehow and complex scripting or the exploit of known interface vulnerabilities is used to this end. It can be seen that a non human-readable resource (such as a QR code), would potentially render the user highly susceptible to this kind of attack.

Figures 2 and 3 illustrate the series of events in a typical Web request both without and with the use of a QR code. These demonstrate the relationship between the user, phone QR code and server and highlight how the use of a technology such as QR codes introduces potentially unknown data that will be treated by the device in the same way as if it were manually entered by the user.
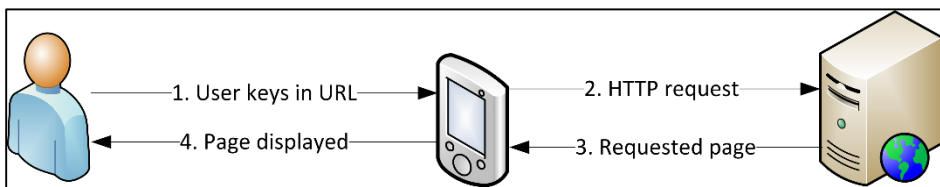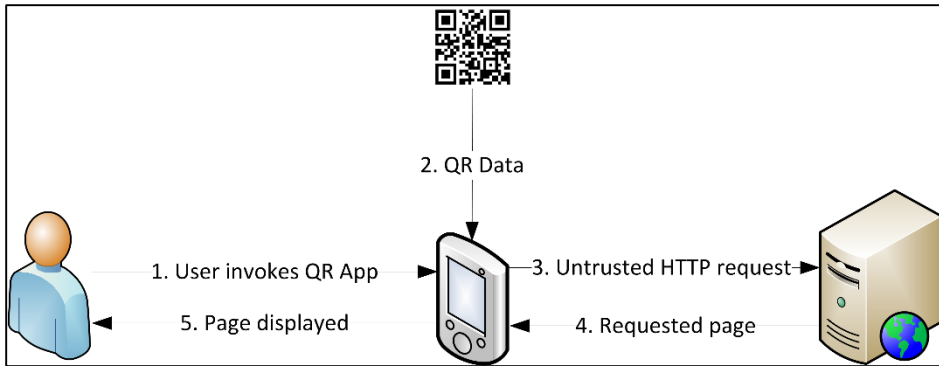

Figure 2 Typical Web Request

Figure 3 Web Request via QR

3. EXISTING RESEARCH

Publicly available information regarding whether QR codes have been involved in any recent security incidents is scarce, perhaps because there have been no high-profile incidents to date. However, the potential for this technology to be involved indirectly, or in future exploits is significant nonetheless. Around the 2007-2008 timeframe, the German hacker "FX" described a number of situations in which 1-d and 2-d barcodes may be exploited to achieve a variety of outcomes. Some of the attacks described methods to overcome ticketing checks such as airline boarding passes and baggage checks, as well as other exploits that may utilize cross site scripting vulnerabilities or buffer overflows by using a 2-d barcode to point to an untrusted resource (FX, 2007). In spite of the fact that the recent rise in smartphone ownership has made this attack vector applicable to a much wider target group, little has been done to address these concerns to date.

Kieseberg et al. (2010) also describe a substantial number of potential weaknesses in the implementation of QR code. These again hinge on the non-human readable nature of the code and how this results in it being often impossible to distinguish between a valid or manipulated code. These possible attacks include modifications to individual components of the code (such as the error correction or header information) as well as attacks based on entirely automated processes such as those used in logistics and assembly line.

Research is ongoing in a number of areas relating to QR codes, and this is especially valuable given the large measure of trust that is (often unwittingly) placed in the printed barcode. These codes are often used for many purposes other than the commonly seen advertising. For example, the West Midlands Police in the United Kingdom now employ the use of QR codes to provide public information in the fight against crime (West Midlands Police, 2012).

McAfee Labs has described an Android based malware that uses QR as its attack vector. Whilst the code and payload of the malware is very similar to other common examples, this variant differs in that it uses a simple QR code to spread. The code initiates a download of a trojanized application which, when installed, sends SMS messages to premium numbers that charge users large sums of money (Sabapathy, 2011).

Attackers have also attempted to embed QR codes into spam emails. Embedded links in spam email contain a shortcut to a legitimate QR code generation service. The bookmarked shortcut that is displayed is a QR code pointing to a site such as pharmaceutical spam. This may seem like an unusual way of attacking given that the email already contains embedded links. However, what it demonstrates is that this method of obscuring the destination URL has been identified as being a workable attack vector for the spammers to evade traditional malicious link detection routines (such as those commonly applied to incoming email) (Websense Security Labs, 2012).

As a demonstration of the level of trust that users place in the QR code, a poster was placed at a three day security conference, featuring the text "*Just scan to win an iPad*". Over the course of the three days, 455 unique users scanned the featured code and visited the associated Web page. Furthermore, the very presence of the poster was never called into question, in spite of it being unapproved. The fact that this potential attack was so successful even at a security conference highlights the risks that the general public may be exposed to (Maman, 2012).

4. RESEARCH QUESTIONS

To consider how QR codes impact the information security domain, there are three main areas to consider: the end user's interaction with QR codes, the technical implementation aspects of QR codes and how QR codes may influence the conclusions drawn from data in particular in areas such as forensic investigations. An essential part of any successful forensic investigation is the clear understanding of what data is being sought and what hypothesis is being tested (proven or disproven). This plays a pivotal role in the evidence recovery and examination (Noblett, Politt, and Presley, 2000), and in the development of the investigation and analysis methodology that will follow.

The following are a series of research questions that attempt to encapsulate this discussion.

1. Can a user be tricked into visiting an illegal/malicious resource via QR code?
2. Is it possible to track the users browsing history via a QR code?

3. Is it possible to determine if a user visited a resource via a QR code or via typing in the URL?
4. Is it possible to physically manipulate a QR code to alter its contents?
5. Can a QR code transaction lead to compromised personal data on a mobile device?
6. Are QR codes sufficient for establishing the location of a user?
7. Is it possible to uniquely identify a user who visits a resource via QR link?

An important principle when dealing with any evidence, either digital or otherwise is that the rules of evidence must be adhered to. This means that both inculpatory and exculpatory evidence must be submitted. Inculpatory evidence is evidence that supports a given theory (for example, did suspect A intentionally visit an illegal Web site). Exculpatory evidence, on the other hand is evidence that contracts a given theory. Irrespective of whether the evidence being collected appears to be inculpatory or exculpatory, it must be dealt with equally and consistently. This is firstly to ensure a correct and unbiased decision may be reached based on the evidence, but also to comply with legislation that covers rules of evidence should they be required to be used in legal proceedings at some future date. The analysis and discussion presented later in this paper does not attempt to prove or disprove any theory–but rather to convey all of the findings and present a discussion that will equip other security professionals with the insights to develop their own educated judgments about evidence specific to their particular cases or investigations.

5. EMPIRICAL INVESTIGATION

The study conducted involved a number of different tests which provided insights and empirical data related to the research questions that were posed above. The tests included data collection on both the smartphone itself, a forensic examination of the detailed server logs that hold the transaction information, and an analysis of the standards and implementation considerations of the technology in general. Smartphone analysis was conducted to study how the application handles the entire QR interaction from scanning to access of a Web resource. Next, a second analysis was conducted which involved access to a Web resource that attempted to access the contents of the smartphone sensors including location and position sensors. On the server side, detailed logs were kept during the entire study. The final step of the data collection involved subjecting these logs to a forensic examination to determine firstly if there are any inconsistencies between platforms, and secondly to establish what if any information is being communicated to the server without the users' knowledge.

Taking into account the diverse nature of mobile devices and lack of standardization within QR code application software, the study was conducted

on a range of devices representative of the three major smartphone platforms currently on the market. These included an Apple device running iOS, a Samsung device running Android, and a Nokia device running Windows. All of the smartphones used the most current and patched versions of their respective operating systems at the time of conducting the study. As a number of third party applications are available from the respective application stores, the most highly ranked two applications for each platform were chosen for evaluation. The Nokia device natively handles QR codes with no additional software required so only one other third party QR reader used on this platform.

At the time of writing, the two most popular QR applications for the iPhone are *RedLaser v4.01* and *QRReader v3.0* (Apple Inc, 2012). For Android, the top two applications are *QR Droid v5.2.1* and *Barcode Scanner v4.3.1* (Android.com, 2012). For Nokia, *QR Code Reader v1.3.4462* was the highest rated application in the Windows application repository (Nokia Corporation, 2012). These are also the most current versions of the applications in the respective application repositories at the time of writing. For the purposes of the investigation, the default factory configuration of the devices operating system, browser and applications were used. The details of the platforms and QR reader applications used in this software are presented below in Table 1.

Table 1 Hardware and Application Platforms used in Study

Platform	Version	Operating System	OS Version	QR Reader
Apple	iPhone 4S	iOS	5.1.1	RedLaser v4.0.1
Apple	iPhone 4S	iOS	5.1.1	QRReader v3.0
Samsung	Galaxy S2	Android	2.3.3	QR Droid v5.2.1
Samsung	Galaxy S2	Android	2.3.3	Barcode Scanner v4.3.1
Nokia	Lumia 800	Windows	7.5	Native Support
Nokia	Lumia 800	Windows	7.5	QR Code Reader v1.3.4462.27495

To ensure that the test conditions and environment did not confound any findings, the devices were rebooted prior to each test and any memory resident applications were terminated where applicable. Network functionality was

provided by 802.11g Wi-Fi connectivity, with other forms of data communication (i.e., GPRS) turned off. All devices connected to the same access point with IP addresses allocated by DHCP. Each device was tested separately, and no other devices were allowed to connect to the access point during the testing.

5.1 Test 1 Client side analysis of QR application software handling of QR transaction from initiation, scanning through to access of the encoded web resource

This test studied the different handling of otherwise innocuous Web links encoded as QR codes. A Web link to a blank Web page was encoded into a basic QR image compatible with all the QR application software in use. The behavior of the device during this access was recorded regarding the extent and type of feedback provided to the user and whether any security controls were in place that required the user's acknowledgement before proceeding.

Data was recorded regarding the following aspects of the QR transaction:

1. Is the URL displayed to the user? If applicable, how much of the field is shown?
2. Is a history of previous QR codes stored?
3. Is user interaction required to confirm the transaction (i.e., to visit the Web page once the QR code is scanned)?
4. Is any warning given when Web site URL is obscured or redirected?
5. Is the real un-obscured URL displayed if a redirection has taken place?

The results for each of these five questions are presented in tabular form. Each row contains data about a particular device/reader combination and the numbered columns correspond to the above questions. Discussion of these findings is included in Section 6 of this paper. The raw findings are presented below in Table 2.

Table 2 Client Side Analysis of Standard QR Transaction

Platform	1. URL Display	2. QR History Stored	3. User Confirmation	4. Redirection Warning	5. Redirection URL Display
Apple iPhone *(RedLaser)*	Yes	Yes	Yes	No	Doesn't show real URL
Apple iPhone *(QRReader)*	No	Yes	No (default)	No (default)	Doesn't show any URL
Samsung Galaxy *(QR Droid)*	Yes	Yes	Yes	No	Shows real URL
Samsung Galaxy *(Barcode Scanner)*	Yes	No	Yes	Yes	Shows both
Nokia Lumia *(Native)*	17 characters	Yes	Yes	No	Shows real URL
Nokia Lumia *(QR Code Reader)*	Yes	Yes	Yes	No	Shows real URL

5.2 Test 2 Client side analysis of access to smartphone sensor
5.3 data via QR link

This test studied the extent to which data from smartphone sensors could be obtained via a Web resource accessed via QR. As smartphone operating systems often expose sensor data to application layer processes such as the Web browser, it may be possible to read this information through a Web page linked via QR.

The means by which sensor data is obtained is often platform specific and the three platforms surveyed do include different system level APIs to deal with the specific type and configuration of sensors installed on a given platform. However, the Standards for Web Applications on Mobile (W3 Consortium, 2012) include several APIs to facilitate this interface between the sensor data on a mobile device and Web applications. The Geolocation API provides an interface for locating the device (independent of the underlying technology);

this is considered "widely deployed" and the functionality is implemented on most current platforms. Other APIs are in development to provide support for motion and proximity sensors, although these are still in development and not as widely deployed.

There are a number of ways data can be collected from the user's device and simply sent back to the server; this is described as follows. Some of these are collected at the server side, some need to be collected in the client and sent back to the server: all this takes place when the Web page is opened. Data may include general device information, location (GPS) of the device and the physical orientation of the device.

To ascertain the extent to which sensor data is revealed to a potentially untrusted Web site, a Web page was created which attempts to poll each of the above mentioned APIs to display current sensor data. The address of this Web page was encoded in QR format and this was used to initiate the Web transaction. As with the previous test, the steps were repeated for each combination of device and platform and the results are detailed in Table 3. In the table, the columns refer to the following items:

1. Device Information: refers to if the QR application has access to device information, e.g., make and model of the phone.
2. Geolocation W3C API: refers to if the QR application has access to the W3C Geolocation API. Prompted means that the user was prompted to allow this.
3. Device Orientation: refers to the QR applications access to either basic HTML device orientation or detailed W3C device orientation including tilt. This is especially important as access to tilt sensors (or accelerometers) may reveal users on-screen keyboard patterns, including passwords (Aviv, Sapp, Blaze, & Smith, 2012).
4. Motion Sensors: refers to if the application has access to the W3C standard calls for motion sensors.

Table 3 Client Side Analysis of Access to Smartphone Sensor Data via QR Link

Platform	1. Device Information	2. Geolocation API	3. Device Orientation	4. Motion Sensors
Apple iPhone *(RedLaser)*	Yes	Prompted	Detailed	Success
Apple iPhone *(QRReader)*	Yes	Prompted	Detailed	Success
Samsung Galaxy *(QR Droid)*	Yes	Prompted	Basic	Device Not Supported
Samsung Galaxy *(Barcode Scanner)*	Yes	Prompted	Basic	Device Not Supported
Nokia Lumia *(Native)*	Yes	Prompted	Basic	Device Not Supported
Nokia Lumia *(QR Code Reader)*	Yes	Prompted	Basic	Device Not Supported

For this test, the W3C APIs were used as a common denominator to evaluate the ways in which different platforms handle the same test. It should be also noted that individual platforms also have their own proprietary APIs which may potentially expose the information in different ways. It is a trivial task for a Web site to automatically generate the content based on the platform being used to access the resource, therefore should a vulnerability or exploit become known for a specific platform, it is possible for a potential attacker to target only specific devices.

5.3 Test 3 Analysis of HTTP Header Information

This test studied the data that is encoded in the HTTP headers sent by the smartphone when a Web resource is accessed via a QR code. As there are many optional headers in addition to those required by the HTTP standards, it is possible that different combinations of QR reader/platform may encode different information in these headers, potentially exposing personal information to the Web server.

In a typical Web transaction, the browser requests a specific resource from the server. Along with this request, the HTTP standard (The Internet Society, 1999) includes several lines of header information. These provide the server

with context for the Web request and details on the kind of data that the browser can handle, what type of browser is being used and so forth.

A Web server based tool was used which prints out the full HTTP headers of any given Web requests. This tool was used to collect the header information from the five platform/reader combinations being used. The raw header data was then captured and is presented below.

iPhone 4

```
HTTP_USER_AGENT          Mozilla/5.0 (iPhone; CPU iPhone OS
                         5_1_1 like Mac OS X)
                         AppleWebKit/534.46 (KHTML, like
                         Gecko) Version/5.1 Mobile/9B206
                         Safari/7534.48.3
HTTP_CONNECTION          keep-alive
REMOTE_ADDR              Confirmed IP Address
HTTP_HOST                testurl.org
REQUEST_URI              /pc.cgi
HTTP_ACCEPT
      text/html,application/xhtml+xml,application/xml;
                         q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE     en-us
HTTP_ACCEPT_ENCODING     gzip, deflate
HTTP_X_WAP_PROFILE
       http://wap.samsungmobile.com/uaprof/GT-I9000.xml
HTTP_ACCEPT_CHARSET      utf-8, iso-8859-1, utf-16, *;q=0.7
```

Android

```
HTTP_USER_AGENT          Mozilla/5.0 (Linux; U; Android
2.3.3; en-au; GT-I9000
                         Build/GINGERBREAD)
                         AppleWebKit/533.1 (KHTML, like
                         Gecko) Version/4.0 Mobile
                         Safari/533.1
REMOTE_ADDR              Confirmed IP Address
HTTP_HOST                testurl.org
REQUEST_URI              /pc.cgi
HTTP_ACCEPT
      application/xml,application/xhtml+xml,text/html;

      q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
HTTP_ACCEPT_LANGUAGE     en-AU, en-US
HTTP_ACCEPT_ENCODING     gzip
```

Windows Phone

```
HTTP_USER_AGENT         Mozilla/5.0 (compatible; MSIE 9.0;
Windows Phone OS 7.5;
                        Trident/5.0; IEMobile/9.0; NOKIA;
Lumia 800)
HTTP_CONNECTION         Keep-Alive
REMOTE_ADDR             Confirmed IP Address
HTTP_HOST               testurl.org
HTTP_UA_CPU             ARM
REQUEST_URI             /pc.cgi
HTTP_ACCEPT             text/html, application/xhtml+xml,
*/*
HTTP_ACCEPT_LANGUAGE    en-US
HTTP_ACCEPT_ENCODING    gzip, deflate
```

5.4 Test 4 Analysis and Modification of QR Code Data

This test examined the structure of a QR Code to evaluate whether it is possible to modify parts of a code to alter its contents imperceptibly. As the codes are not human-readable, there is potential for changes to a code to go unnoticed to the casual observer. One concern is that a malicious user may modify a part of a QR code to point to a slightly different resource. For example, a link to "www.murdoch.edu.au" may be subjected to a single character change to point to "www.murdoch.edu.ai".

To investigate the possibility for such an attack, the QR standards were examined. The ISO 18004:2006 standard describes the layout and organization of a QR code. In addition to the easily recognizable matrix of black/white pixels (known as the data area), there are several other fixed characteristics that are common to all QR codes. These include a finder pattern, a set of 3 blocks which are located in three of the corners of the code. These enable the scanning device to determine the size, orientation and angle of the symbol–without the finder pattern it is not possible for the scanner to recognize that a QR code has been presented.

The timing pattern provides a reference for the cell pitch–this is to describe how wide in pixels the rows and columns are to be expected in the code. Finally, the margin around the data area is known as the quiet zone–this simply facilitates the task (for the CCD) of discerning the code from the surrounding image in the field of view. Figure 4 illustrates the standard organization of a QR code as described in ISO 18004 (International Organization for Standardization, 2006).
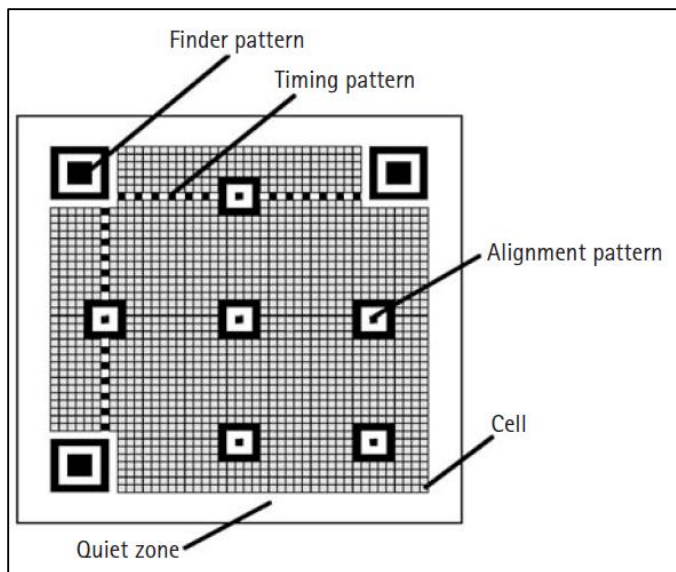
Figure 4 Structure of a QR Code

Also included in the ISO standard are details of the error correcting codes that are to be implemented in all QR implementations. QR codes are encoded using Reed-Solomon error-correcting codes (Reed & Solomon, 1960). This allows the content to be decoded even if a certain amount of degradation of the data area has occurred. There are several levels of error correction available at creation time, and depending on the final intended use of the code, different requirements for error correction will be appropriate. At the highest level of error correction the algorithm is capable of withstanding loss or corruption of up to 30% of the data area and still operating correctly.

The next step of the analysis was to evaluate the differences in QR code representation of two similar text strings. To this end, the strings "ABCDEF" and "ABCDEG" were encoded in QR form. As the data content is very small, and the difference between the two strings is limited to a single character, it was anticipated that the QR representation would likely be quite similar. Figure 5 shows the QR representations of the strings ABCDEF and ABCDEG respectively.
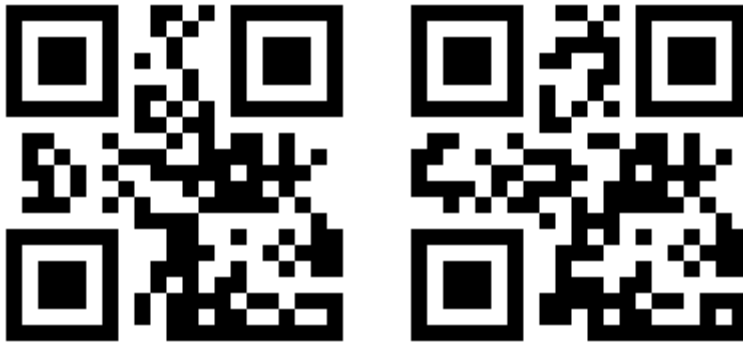
Figure 5 QR Representation of "ABCDEF" and "ABCDEG" Strings

This type of comparison is complicated by the fact that the QR standard dictates that the data area is XOR'ed with an obfuscating mask during encoding of final output. The mask simply changes which bits are on and which bits are off according to a rule. There are eight obfuscating mask patterns defined in the QR standard. At creation time, the algorithm will automatically select the most appropriate mask to generate a code that will be the easiest for the scanner to read–this is not an option that is selectable by the user at run-time. This means that there are eight possible representations of the same data string. For this test, the QR code generator was forced to utilize the same mask when creating the above two codes to allow for direct comparison of their contents.

These codes shown above in Figures 5 were masked and overlaid to visually demonstrate the extent of change caused to the code when a single character modification is made to the data area. The resulting difference map is presented below in Figure 6.

Figure 6 Difference Map

6. DISCUSSION

The reliance on computer records as evidence carries additional risks as their admissibility may be challenged as hearsay (United States Department of Justice, 2009). This challenge comes from the fact that digital evidence may somewhat fit the definition of being a statement made by one other than the declarant as evidence (Federal Rules of Evidence, 2011a). Therefore, computer-generated records which fit into this definition may thus be challenged under common law. Fortunately, this is an area which has received significant attention and statutes such as the Federal Rules of Evidence now make exemptions to the hearsay rule for these computer generated business records, provided the supporting conditions are met including, amongst others, reliability and relevance.

In the United States, the Federal Rules of Evidence Rule 801(6) states that business records are not hearsay:

> *(6) Records of a Regularly Conducted Activity. A record of an act, event, condition, opinion, or diagnosis if:*
> > *(a) the record was made at or near the time by—or from information transmitted by—someone with knowledge;*
> > *(b) the record was kept in the course of a regularly conducted activity of a business, organization, occupation, or calling, whether or not for profit;*
> > *(c) making the record was a regular practice of that activity;*
> > (Federal Rules of Evidence, 2011b).

Similarly, in the United Kingdom, the 114-136 of Part II Criminal Justice Act 2003, also clarifies that business records *"created or received by a person in the course of a trade, business, profession or other occupation"* are exempt from the hearsay rule and are initially admissible (Criminal Justice Act, 2003).

Although the wording varies, the basic effect of these rules has been to relax the common law requirement that the person who recorded the information be present to testify if available. This has been quite successful in clarifying the position of computer-generated records. Even before the computer age, in the case of *Transport Indemnity Company vs. Seib. 178 Neb. 253* (1965), the Supreme Court of Nebraska permitted *systematically entered records without the necessity of identifying, locating and producing as witnesses the individuals who made entries in the records in the regular course of business*. More recently, many courts have clearly established that computer records are

admissible under Federal Rules of Evidence without first asking if the records are hearsay[1].

The regular use of a network enabled computer (such as a smartphone) creates a wealth of data including computer stored information on the device such as history files or caches, to computer generated usage logs or server access logs if Web transfers took place. Many of these logs are system created and may also possess audit trails which can be used to support their authenticity. Therefore the analysis of these sources of information is potentially very valuable as they may directly provide a timeline of a user's activities.

Each research question from Section 3 will now be discussed based on the data collected in the empirical investigation in Section 4.

6.1 RQ 1: Can a user be tricked into visiting an illegal/malicious resource via QR code?

In many cases it is possible for the user to be directed to an untrusted resource with no prior notification or warning. As the results from Test 1 indicated, the lack of application level standardization is evident and the different combinations of platform and reader handled the scanning and access of a QR code URL in markedly different ways.

The most concerning implementation was the *iPhone/QRReader* combination. In its default configuration, the application did not display the contents of the QR code to the user, and also failed to prompt the user for confirmation before connecting to the specified resource. This means that the user could be tricked into visiting any kind of resource, simply by encoding its URL into a QR code. The *Lumia/Native* combination was also cause for concern, as the displayed URL was truncated to a maximum field size of 17 characters. Therefore, the user would not be able to view any field larger than that limit. This makes it relatively trivialfor an attacker to hide any suspicious elements of the URL outside that range. For example, the URL www.safecomputer.hackersdomain.com would be displayed as www.safecomputer on this device.

Other combinations of platform and reader properly displayed the URL and also prompted the user with a confirmation button before accessing the resource.

Test 1 also evaluated how URL redirection was handled by the various QR readers. Web site redirection is a common technique used by attackers as a means of hiding the true URL from view from the user. URL redirection is

---

[1] For further examples of cases in which computer records have been exempted from hearsay rules, please see *Haag v. United States,* 485 F.3d 1, 3 (1st Cir. 2007); *United States v. Fujii*, 301 F.3d 535, 539 (7th Cir. 2002); and *United States v. Briscoe*, 896 F.2d 1476, 1494 (7th Cir. 1990).

becoming more common due to services such as bit.ly and tinyurl.com providing free URL shortening services with no subscription or signup requirements.

There was a large amount of variance in the way that the different platforms handled this test. The *Nokia/Native* and *Nokia/QR Code Reader* combinations both displayed the real URL of the resource being visited. This is the most desirable behaviour as the user is presented with all of the facts and may not be misled into a visiting a malicious resource. The *Samsung/QR Droid* and *Samsung/Barcode Scanner* also both displayed the real URL of the Web site– the *Samsung/Barcode Scanner* also had the advantage that the real and redirect URLs were both presented on screen.

The *Apple/RedLaser* and *Apple/QRReader* combinations were the biggest cause for concern, as neither of these combinations displayed the real URL of the resource being visited. The *Apple/QRReader* combination did not even display the redirect URL thus giving the user zero feedback as to what resource they were accessing. The net result of this is that users using either of these combinations may easily be tricked into visiting a malicious resource.

For the forensics investigator it may be difficult to establish intent when considering the users Web access history. The user may claim ignorance, and state that they were not aware of what they were accessing at the time. The lack of feedback from the QR application, combined with the lack of prompting in certain cases means that it is certainly a possibility that a user may scan an untrusted QR link and be automatically taken to a malicious or illegal Web site without their consent.

6.2 RQ 2: Is it possible to track the users browsing history via a QR code?

Provided the mobile phone manufacturer and the browser developers adhere to the W3C standards, this is not possible. Test 2 enumerated the "History" object while accessing the sensor data. Were this exposed, it would simply allow the user to determine how many items are currently in the client history (not what they contain). In almost all cases, the QR application initiated a new browser session with each scan thus resetting the History contents value to zero. In certain circumstances the device can be forced to use the same session, but this task simply increments the integer value of the History object size and did not yield any useful data. If a history list is present on the device, it is technically possible for a malicious page to force the browser to go to a previous page, but this is unlikely to cause any security problems as the redirection is limited to resources which have already been previously visited.

Test 4 examined the HTTP_REFERER header. This optional header contains the URL of the resource from which the request was initiated. This allows the new Web page to determine where the user is visiting from. This was of

interest, as in some cases this may potentially give the new Web site valuable information about the user. As certain Web sites may encode personal information as part of the query string (e.g., http://mysite.com?login=username&password=hello), this was considered to be a potential attack vector. However, this header was not sent by any of the device combinations evaluated. Furthermore, the fact that the QR readers appear to initiate a new session with each scan, this means that there is no actual referring Web site that may be documented in this header.

These findings have quite different implications for different stakeholders. From the end users point of view, this reflects a positive outcome that this aspect of their personal data is not directly visible to an outside party. This is, of course, a desirable situation–and no doubt a product of careful design on the part of the software and operating system developers. From the point of view of the forensics investigator, who may be called upon to develop a profile or pattern of usage for a particular user or device, this means that this particular mechanism may not be of use to them in this instance. However, there are many other existing sources of information by which an investigator may pull together patterns of usage.

6.3 RQ 3: Is it possible to determine if a user visited a resource via QR or via typing in the URL?

On the server side all of the requests appear identical, therefore it is not possible to determine if the user clicked on a QR link or typed in a URL manually. On the client device itself, there are traces of the transaction left behind that may be analyzed to ascertain the origin of the request. The internal browser on the device may store a history of all Web transactions. This would confirm that the Web site has been visited but once again does not show where the request originated from.

The QR reader applications also store a history, and in some cases this includes meta-data regarding when the link was scanned and accessed. This is the only information that can be used to link a Web access to a QR code, and given that all of the applications handle this task differently, it is not unreasonable to expect that the integrity of this data may be questioned.

As with the case of Research Question 1 described above, in the case of access to illegal or prohibited resources, the process of establishing intent may be confounded by this blurring of QR vs. manually visited Web resources. In the course of an investigation there is potential for a user to simply claim ignorance and state that they clicked on a QR link and that took them to the illegal resource. In many jurisdictions it is necessary to demonstrate that the accused committed a deliberate act (i.e., prove intent) or that they did indeed have knowledge and awareness of the outcome of their actions. The act in itself does not necessarily make a person guilty if these elements are not present.

If a single access is being considered, and no other record of activity or access is present, then it may be impossible to prove or disprove this assertion. During a forensic investigation it may be necessary to demonstrate a certain pattern of access on the part of the user. In the absence of a QR code history, then this may be a complicated task and evidence of a single illegal transaction, may prove little at face value.

6.4 RQ 4: Is it possible to physically manipulate a QR code to alter its contents?

As QR codes are increasingly being used and trusted by the public, the task of ensuring that QR codes are legitimate becomes more important. QR codes are often presented alongside easily recognizable and protected brand material which people implicitly trust. However, the QR codes themselves are visually unidentifiable from one other. There is therefore a concern that the contents of the QR codes could be modified or that QR codes could be simply replaced by covering the QR code with another.

The second concern is that QR codes could be slightly manipulated to change the URL being represented by them. Due to the fact that, like barcodes, different QR codes are visually very similar, there is a concern that legitimate QR codes may be slightly modified to direct users unwittingly to an untrusted resource.

Test 4 performed an analysis on QR codes to see the output of QR generators with slightly different text and investigated if small changes could be made to QR codes to change the encoded URL address. This analysis revealed that it is not feasible for the contents of a legitimate QR code to be modified or altered as the modifications needed to the QR code would be substantial. Changes of between 7 and 30% of the pixels (depending on ECC in use) of an existing QR code may still result in no net change to the QR contents. Furthermore, any inadvertent changes to the finder or timing patterns would render the code unusable.

Two QR codes containing almost identical character strings were encoded and compared. Due to the low data density, the codes utilized in the test were V1 codes providing a 21x21 matrix totaling 441 blocks. As can be seen from the difference map in Figure 5, the QR images are significantly different. Pixel by pixel analysis indicated that the single character change in the encoded message resulted in a reorganization of 10.2 % of the total pixels in the code. The V1 standard states that of the total 441 addressable blocks, only 208 of these blocks are actually data blocks. The rest are used for timing, reference and positioning information as mentioned above. Taking this into consideration, the analysis was repeated and revealed that this single character

change in the encoded message actually resulted in a change to 21.6% of the data region.

Thus, this test has demonstrated that whilst it is technically possible to alter the contents of a QR code, it is by no means a trivial undertaking. Simple filling in of white spaces with dark space would not be enough as the replacement of large parts of the QR code is necessary. However as it is very difficult to visually discern between any two QR codes, there remains the potential for an attacker to entirely replace a QR code with another. There is also the possibility that the attack may concentrate not on the data part of the QR code, but on the header information. This could theoretically change the character encoding or character count fields and cause a buffer underflow or overflow. This potential attack has been previously identified in literature however no practical evaluation was conducted at that time (Kieseberg, et al., 2010).

These tests are more directly related to the potential computer security vulnerabilities than a forensics investigation process. However, there are foreseeable situations in which these findings have a bearing on an investigation. Forensics investigation routinely involves either the attribution of a document or record to its source or authentication of the document authentication. As the QR code is a physical and not an electronic record, the mechanisms by which this record may be validated and assessed are limited to more traditional means, outside the domain of digital forensics. However, as Test 4 has demonstrated very small changes in the data content of a QR code result in a large and easily detectable change to the final QR output. Therefore the process of document authentication is greatly simplified provided the investigator is aware of the original and intended contents of the QR code.

6.5 RQ 5: Can a QR code transaction lead to compromised personal data on a mobile device?

There is nothing inherent in the nature of the QR code transaction that would result in the vulnerability of personal data. However, as discussed above in Research Question 1, the QR code is a viable attack vector by which malicious users may direct traffic to their own Web site. To this end, the dangers to the user are the same as those associated with visiting any untrusted Web site.

Vulnerabilities in computer systems are regularly discovered and exploited by attackers to acquire personal data. Smartphones are not immune to this form of attack, and should be treated in the same way as a home or office computer, and protected adequately. The recently announced Android malware genome project (Zhou & Jiang, 2012) has already catalogued over 1200 examples of malware on this one platform alone. Many of these samples use Web technologies to replicate and spread. This highlights the extent and rapid growth of malware in this arena. It is conceivable that attackers may employ QR based "clickjacking" techniques to direct users to spread their malware.

Another potential way in which personal data may be compromised comes via a more indirect route. Test 2 investigated which, if any of the smartphone sensors' values may be exposed to an attacker via a QR code. Amongst the sensors evaluated, was the on board accelerometer. It has previously been demonstrated that the onboard accelerometer can be used to infer the keystrokes that the user is entering on a touchscreen (Cai, 2012). This is done by mathematically modeling the relation between onscreen tap events (i.e., touching an on screen keyboard) and the motion of the phone. This proposed mechanism has been successfully implemented, and several key loggers have been demonstrated which use only the accelerometer of the device is used as an input. The results from Test 2 indicated that the iOS based applications did provide detailed motion and tilt information to the calling application (Web site), thus it is conceivable that this could be another potential area to exploit. It is also likely that investigation of the device specific APIs may provide further scope for smartphone sensor access, which may reveal similar vulnerabilities in the Windows and Android platforms.

The investigation process, either traditional or digital, is ultimately a fact-finding exercise. Thus data obtained from smartphone onboard sensors is a potentially valuable and rich source of information about both the event that took place, and the context such as environmental and situational characteristics that surrounded that event. These, often very diverse sources of information may appear peripheral when considered in isolation, but when combined, these may form an indispensable information source to the investigator. As such, the analysis and understanding of specific device sensors and the range of APIs in use is a crucial area of digital forensics.

6.6 RQ 6: Are QR codes suitable for establishing the location of a user?

In some circumstances, a QR code scan may result in the location of the user at the time being divulged. This can happen through several means. Firstly, the QR code itself may be unique to a particular location. As the codes are not human-readable, there is no way of determining if the QR code is unique to the location and it is thus possible that different variants of QR codes may be situated in different places, thus making it possible to determine the physical location of the client at the time of scanning. However, this task is confounded by the history functionality provided by the majority of readers. Test 1 showed that 5 out of 6 of the most popular readers store a copy of the QR codes. These may later be scanned and revisited at leisure. Therefore it is entirely possible that the user may appear to "scan" the code when they are actually in a different location altogether. From the point of view of the forensic investigator, this information alone may not be sufficient to establish the location of the user and it must be used in conjunction with other data such as the originating IP address of the access.

Test 2 attempted to access the sensor API on the smartphone itself. All of the smartphones utilized in the study contained on-board GPS chips, so it was logical to attempt to access this. As seen in Table 2, it was possible to access the GeoLocation API on all devices; this gives the current latitude and longitude of the smartphone within 5-6 meter accuracy. However, all QR reader attempts to access this API were preceded by a prompt. Whilst prompting the user for permission is indeed a necessary element of secure browsing, the de-sensitization of users to these prompts may render them to be little more than an inconvenience that the user will pay little attention to before clicking.

Establishment and verification of alibis is a routine part of an investigation. Thus the forensics investigator may often be called upon to provide insight into this area. As discussed above, the fact alone that a QR code has been used is not necessarily sufficient to establish that the user was in a particular physical location at the time of access. Other sources of information must be used to complement this date in order to make any concrete assertions.

6.7 RQ 7: Is it possible to uniquely identify a user who visits a resource via QR link?

As well as possibly revealing a user's physical location (as discussed in question 6), QR codes offer the possibility of identifying a user's Internet location and device details. When a user uses a QR code to visit a Web site, various details of the user and device are revealed. Test 3 investigated this by looking at the HTTP header information sent with the QR application request.

The HTTP_USER_AGENT header identifies the hardware device, operating system browser of any HTTP request. This is present in order to assist the HTTP server in targeting the correct content for the device, e.g., providing a mobile device optimized version of Web page rather than a full screen desktop version. The HTTP_USER_AGENT reveals rich information about a mobile phone; including the make and model that can then be used to find out further information from other sources such as the manufacturer.

As well as hardware information, the HTTP headers include the Internet Protocol (IP) address of the device, which is unique on the Internet. Although services such as network address translation (NAT) may allow multiple devices to share addresses, the address still is useful as it permits identification of the locality and Internet provider, information which may later be used to uniquely identify a device and user. Aside from this data, there were no other non-standard headers sent by any of the devices.

To clarify, this test was not to ascertain if a particular known user had accessed a QR link from their mobile device. If this were the aim, then more straightforward mobile forensics techniques may be a more suitable first port of

call. This test was rather to ascertain if, from the server side alone, it is possible to know which mobile user is scanning (and thus accessing) the QR link. This information would potentially be a security risk as it would provide malicious user knowledge of unique user patterns and physical locations at various points in time. However, the results of the server log analysis have indicated that on the devices tested, there is no uniquely identifying information sent during the QR link access.

7. CONCLUSIONS

The dramatic rise in the uptake of QR codes is evident in market research statistics. The MultiChannel Merchant annually surveys approximately 1000 respondents primarily based in the USA. In 2011, the results indicated that only 8% of retailers were using QR technology. This is a significant amount, but certainly not the landslide that had been predicted. This year however, the same survey indicates that 47% of the respondents said they are using QR codes, with an additional 15% of respondents planning to implement this or similar technologies in the near future (MultiChannel Merchant, 2012).

As with any technology that experiences such rapid growth and uptake, there is a clear risk that there may not be commensurate developments in the area of security. Technical security exploits and weaknesses are often only discovered after they are exploited by malware, in many cases this may have already spread and caused widespread damage. In this particular situation the matter is confounded by the combination of both technical vulnerability and human factors involved in this interaction. The likelihood of security breaches, potentially without the user's knowledge has profound implications for the digital forensics or security investigator as they may be required to investigate an incident about which the alleged perpetrator has no actual knowledge. In such a situation, it will be necessary to have a clear understanding of the technology and the risks it poses in order to distil the facts from the large amounts of (potentially conflicting) evidence that may be presented.

The lack of application layer standardization in the manner in which this technology is handled is cause for concern. The empirical tests discussed in this paper have demonstrated the diversity of implementations, and the ad-hoc nature in which the data is processed–in many cases these go against well-established practices for secure interface design. The tests revealed that there are platforms that do not prompt the user before visiting an untrusted resource, those that do not display the actual URL of the resource to the user (even if they attempt to locate it), and those which reveal the contents of the smartphone onboard sensors (such as GPS and positioning) to an untrusted Internet host. The fact that such diverse results were found even with a relatively small variety of QR application software and hardware also has implications for the forensic investigation process. In a domain where accuracy

of facts and consistency of procedures is essential, this means that access logs and auditing information from a mobile device may not be sufficient in isolation. Instead, these must be supplemented with additional investigation regarding the specific details of the software and hardware involved so as to place any findings within the context of the expected behavior of the platform. A "one size fits all" approach is not currently possible, although the development of a standardized set of procedures is a valuable direction for future research in this area.

Security vendors are beginning to take note of these problems, and whilst some of them mention the risk on their Web sites or technical reports they offer little in the way of solutions. Symantec software has recently released QR code scanning software called Norton Snap (Symantec Software, 2012). When smartphone users scan a QR code with this application the data is relayed to Norton's threat database, which then returns a threat rating for the resource. Based on this information the user may then opt to visit or not visit the Web site in question. Tools like this are a valuable step in the right direction. However, as the user's security behaviour is the root cause of the vulnerability such applications will be unlikely to entirely solve the problem.

As long as the common misconception persists that smartphones are any different or more secure than a regular PC, such attacks will always exist. Widespread awareness and understanding of these issues amongst security professionals and end users alike is the front line of defense against the vulnerabilities associated with new and emerging technologies. It is hoped that the research based insights and discussion presented in this paper will contribute to this goal, and to a more secure mobile communications environment.

REFERENCES

Android.com. (2012). Android app collections. Retrieved from http://www.android.com/apps/ on February 13, 2012.

Apple Inc. (2012). App store. Retrieved from https://itunes.apple.com/au/genre/ios/id36?mt=8 on February 13, 2012.

Aviv, A. J., Sapp, B., Blaze, M., & Smith, J. M. (2012). Practicality of accelerometer side channels on smartphones. Paper presented at the *28th Annual Computer Security Applications Conference*, Orlando, Florida.

Bell, G. B., & Boddington, R. (2010). Solid state Ddrives: The beginning of the end for current practice in digital forensic recovery? *Journal of Digital Forensics, Security and Law, 5*(3), 1-20.

Cai, L. (2012). Trust and trustworthy computing on the practicality of motion based keystroke inference attack. *Lecture notes in computer science, 7344*, 273-290. doi: 10.1007/978-3-642-30921-2_16

Criminal Justice Act. (2003). § 117 *c.44*. United Kingdom.

Denso-Wave Incorporated. (2011). Patents pertaining to the QR Code. Retrieved from http://www.qrcode.com/en/patent.html on June 7, 2013.

Duranti, L., & Endicott-Popovsky, B. (2010). Digital Records forensics: A new science and academic program for forensic readiness. *Journal of Digital Forensics, Security and Law, 5*(2), 1-12.

Federal Rules of Evidence. (2011). § 6. United States.

Federal Rules of Evidence. (2011a). United States.

Federal Rules of Evidence. (2011b). § 6. United States.

FX. (2007). *Toying with barcodes.* Paper presented at the 24th Chaos Communication Congress, Berlin, Germany.

International Organization for standardization. (2006). ISO/IEC 18004. Retrieved from http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=43655 on June 2, 2012.

Japanese Standards Association. (1999). JIS X 0510. Retrieved from http://www.webstore.jsa.or.jp/webstore/Com/FlowControl.jsp?lang=jp&bunsyoId=JIS%20X%200510%3A2004&dantaiCd=JIS&status=1&pageNo=0 on June 2, 2012.

Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Schrittwieser, S., Sinha, M., et al. (2010). QR code security. *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, 430-435. Paris, France.

Maman, D. (2012). The QR code: A new frontier in mobile attackability. Retrieved from http://www.net-security.org/article.php?id=1766 on December 20, 2012.

MultiChannel Merchant. (2012). Ecommerce outlook report. Retrieved from http://multichannelmerchant.com/research/2012/ecommerce/ on August 15, 2012.

Noblett, M., Politt, M., & Presley, L. (2000). Recovering and examining computer forensic evidence. *Forensic Science Communications, 2*(4).

Nokia Corporation. (2012). Nokia windows apps. Retrieved from http://www.nokia.com/au-en/apps/ on February 13, 2012.

Piccinelli, M., & Gubian, P. (2011). Exploring the iPhone backup made by iTunes. *Journal of Digital Forensics, Security and Law, 6*(3).

Reed, I. S., & Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics, 8*(2), 300-304.

Sabapathy, A. (2011). Android malware spreads through QR code. Retrieved from http://blogs.mcafee.com/mcafee-labs/android-malware-spreads-through-qr-code on December 21, 2012.

Scientific Working Group on Digital Evidence. (2013a). SWGDE document repository. Retrieved from https://www.swgde.org/documents/Current%20Documents on June 3, 2013.

Scientific Working Group on Digital Evidence. (2013b). Best practices for mobile phone examinations. Retrieved from https://www.swgde.org/documents/Current%20Documents/2013-02-11%20SWGDE%20Best%20Practices%20for%20Mobile%20Phone%20Examinations%20V2-0 on June 3, 2013.

Symantec Software. (2012). Features of Norton Snap 1.0. Retrieved from https://www-secure.symantec.com/norton-support/jsp/help-solutions.jsp?docid=v64690996_EndUserProfile_en_us&product=home&pvid=f-home&version=1&lg=english&ct=us on August 15, 2012.

The Internet Society. (1999). RFC 2616–Hypertext Transfer Protocol HTTP/1.1. http://www.w3.org/Protocols/rfc2616/rfc2616.txt.

Transport Indemnity Company vs Seib, 178 Neb. 253 (1965).

United States Department of Justice. (2009). Searching and seizing computers and obtaining electronic evidence in criminal investigations. Retrieved from http://www.justice.gov/criminal/cybercrime/docs/ssmanual2009.pdf on June 3, 2013.

W3 Consortium. (2012). Standards for Web applications on mobile, 6th ed. Retrieved from http://www.w3.org/2012/05/mobile-web-app-state/ on July 3, 2012.

Websense Security Labs. (2012). Spam emails link to QR codes. Retrieved from http://community.websense.com/blogs/securitylabs/archive/2012/01/09/spam-emails-link-to-qr-codes.aspx on December 20, 2012.

West Midlands Police. (2012). Rogues Gallery of wanted people. Retrieved from http://www.west-midlands.police.uk/np/coventry/news/newsitem.asp?id=8143 on 21 December, 2012.

Zhou, Y., & Jiang, X. (2012). Dissecting Android Malware: Characterization and Evolution. *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, San Francisco, CA.

ABOUT THE AUTHORS

Dr. Nik Thompson is a Lecturer in Information Technology at Murdoch University in Western Australia. He holds MSc and PhD degrees from Murdoch University and teaches in the area of Systems Analysis. His research interests include affective computing, human-computer interaction and information security.

Dr. Kevin Lee is a Lecturer at Murdoch University in Perth, Australia. He received his PhD from Lancaster University; was a Research Associate at the University of Manchester and a postdoctoral fellow at the University of Mannheim. He has published more than 50 refereed academic papers in international conferences and journals. His research interests focus on adaptive and autonomic systems in the areas of High-speed Networking, Sensor Networks, Scientific Workflow Processing, Physiological Computing and Peer-to-Peer networks.