Journal of Engineering and Applied Sciences 11 (3): 553-560, 2016
ISSN: 1816-949X

QRphish: An Automated QR Code Phishing Detection Approach

[1]Ahmad Y. Alnajjar, [1]Mohammed Anbar, [1]Selvakumar Manickam,
[2]Omar Elejla and [3]Homam El-Taj
[1]National Advanced IPv6 Centre of Excellence (Nav6),
[2]School of Computer Science, Universiti Sains Malaysia, Penang, Malaysia
[3]University of Tabuk, Tabuk, Saudi Arabia

Abstract: Due to their advent and widespread demand, the smart mobiles have increased the use of Quick Response (QR) code technology. However, some phishers have started using certain features of the QR code to spread phishing frauds through smartphones. The QR code is a matrix barcode that allows easy interaction between mobile devices and websites or printed material by removing the burden of manually typing a URL or contact information. Phishers have started using the QR code for the phishing attacks. In this study we have proposed a new approach called "QRphish" which detects URL phishing on QR code. It uses the QR code-specific features and the URL features to detect whether the QR code content has a URL phishing or not. The QR code specific features used in this research use the QR code content and its characteristics like length, type and level of error correction. This method uses the machine learning classification technique. The proposed approach is evaluated by using benchmark dataset; the result shows a high accuracy in term of detecting URL phishing.

Key words: QR code, smartphone, URL phishing, machin e-Learning classification, phishing detection

INTRODUCTION

The primary goal of security researchers is to defend the smart mobiles against attacks launched by the malicious users. There are a number of ways in which researchers and developers can work to protect the software that they write. Some are proactive, like code reviews and regression testing while others are reactive. One class of these tools is detecting phishing on the smart mobiles. Phishing is a social engineering crime generally defined as impersonating a third party to gain access to personal data (Whittaker *et al*., 2010). Businesses, around the world, lost a total of US$594 million due to phishing attacks during December 2014, according to a new report by RSA. Traditionally, there are three main phases to the phishing cycle in the QR code. First, the phisher creates a phishing website and then goes phishing by generating a QR code that contains the phishing URL for unsuspecting users. Because of the properties of QR codes (easy generation, distribution and opacity), their adoption can increase the user's susceptibility to phishing attacks. The phisher tries to convince the reader of the QR code to visit the link included in the QR code. With QR codes, URLs do not need to be manually entered anymore. When the user "bites" on the phish, the link in the QR code directs the user to the phishing site which appears legitimate and similar or identical to the legitimate target site. 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 who are redirected to a phishing website (Kharraz *et al*., 2014). The phish is successful when the user enters private information on the phishing page and it is dripped to the phisher. Then, the phisher tries to exploit the personal information by transferring money, opening bank accounts or making purchases using the obtained information or the phisher acts as a middleman and sells the information to other criminals. In this study, we have built the QRphish, an effective mechanism to detect URL phishing on QR code. The methodology exploits not only the basic phishing detection features that are dependent on the URLand the suspicious web page features but also the QR code specific and the host-based features. In this study, a combination of URL-based, host-based and QR code-based features have been used which help in the effective and real-time detection of phishing on the QR code. QRphish decides whether a URL is "phishing" or "legitimate" by employing a machine-learning technique

Corresponding Author: Ahmad Y. Alnajjar, National Advanced IPv6 Centre of Excellence (NAv6), Universiti Sains Malaysia, Penang, Malaysia

using the combination of the aforementioned features. In addition, we have built a QRphish API to provide real-time phishing detection to the smart mobile users. Furthermore, QRphish has a high accuracy of 93.34%.

The major contributions of this research are:

- The proposed mechanism proves to be more efficient than plain blacklisting method
- The mechanism is an automatic real-time phishing detection technique for QR code
- The study compares the proposed mechanism with other existing phishing detection mechanism
- The study applies the proposed mechanism in real-world

Literature review: Phishing is an attempt to acquire sensitive information and private identifications of the internet users such as credit card details and login information and is masked as anauthoritative entity in an e-Communication. This study gives a summary of studies that describe the phishing attacks and techniques used to detect phishing frauds. The anti-phishing techniques can be placed into three categories: email-based, URL-based detection and content-based methods.

Detection of phishing URLs: ALmomani *et al.* (2013) have defined phishing as "a kind of attack in which the criminals use spoofed emails and fake websites to trick financial organisations and customers. The criminals try to lure online users by convincing them to reveal the username, passwords, credit card number and updating account information or fill billing information" (Almomani *et al*., 2013). Thus, a phishing URL is a URL that leads the user to a phishing web page. However, there are strong email spam filters which successfully filter out the phishing emails and spam (Chandrasekaran *et al*., 2006; Ma *et al*., 2009). Fette *et al.* (2007) have used the machine-learning technique to classify an email as phishing by using the features like a number of dots in a URL, the age of URL and the HTML content of an email and have obtained a high accuracy of 99.5%. Many other techniques have also been widely used to detect the phishing websites Garera *et al.* (2007) have used logistic regression over the hand-selected features for classifying the phishing URLs. The features include the presence of red flag keywords in the URL, features based on the Google's web page quality guidelines and Google's page rank. Other studies considered classifiers using the URL features and search engine results as a mechanism for high detection rates while maintaining low false positive rates (Whittaker *et al*., 2010; Zhang *et al*., 2007; Garera *et al*., 2007; Miyamoto *et al*., 2009; Chandrasekaran *et al*., 2006) proposed a technique for identifying phishing emails by using the structural properties. Theanti-phishing techniques are not always applicable as the approaches require phishing emails. Justus used the host-based and lexical features of the URLs to detect malicious webpages. Nevertheless, since phishers keep changing their attacking approach, only using the URL features can be problematic for detection of malicious URLs (Ma *et al*., 2009; Zhang *et al*., 2007b) proposed Cantina (Carnegie Mellon Anti-phishing and Network Analysis Tool) wherein this approach detects phishing websites by checking the contents of the website. Cantina seeks to find out if the website was indexed by general search engines (e.g., Google) or not which is considered as a reference to determine a legitimate website (Zhang *et al*., 2007). Cantina used the contents of the website to analyse them, thereafter used the top five terms to determine if the website is phishing or legitimate by feeding them to a search engine (Zhang *et al*., 2007). Xiang *et al.* (2011) proposed CANTINA+ wherein, the approach uses a machine learning technique to classify the websites as phishing or legitimate by extracting the features of a website like webpage properties, URL properties and the uses. Blacklist is a basic access control mechanism used by the email filters and browsers to block users from the malicious content (e.g., websites and email messages). Some approaches like, Google Safebrowsing and APWG blacklist, feed the blacklists by detecting the phishing URLs. However, a major disadvantage of blacklists is their inability to determine the phishing URLs at the instant time of a phishing attack, due to their slow update process on. The mechanism proposed by us, i.e., QRphish uses the blacklists, the lexical and host-based features of URLs in making the classification decision.

Phishing and spamming on QR code: Scanning a QR code in the wild is not a legitimate practice because as mentioned previously it can generate attacks to the back-end system that serves the request or to the user's device. Currently, the only way to avoid phishing attacks based on QR codes, rely on the awareness and the ability of the user to identify malicious URLs or involve cues from the external tools such as the blacklisted domain services. These cuesare not always interpreted correctly by the users and are not very effective. QRishing is the term that defines the QR Code initiated phishing attacks

(Vidas *et al*., 2012). Vidas *et al*. (2012) deployed for their study, a QRishing experiment but they not only used flyers with QR codes but also used rip-off flyers which they posted in different places around the city of Pittsburgh. They used three different kinds of posters containing QR codes: plain QR code, QR code with commands that explained the process of scanning the QR code and QR code with information about their study. The locations selected for the study were mainly restaurants, bus stops and cafes. The people who would scan a QR code from a poster would be taken to a web page where they were asked to participate in an optional survey. Even though the participation was not enough to lead to better conclusions, the big majority (85%) of the people that scanned the QR code also visited the web page (Vidas *et al*., 2012). After reviewing the above techniques, it was clear that there was hardly any work that could detectthe phishing on QR code. To solve this problem, the authors in this paper have designed and developed the QRphish: it enhances the detection of phishing on the QR code defence blacklist, the lexical and the host-based features of URLs. One important contribution of this article is to demonstrate that the lexical and the host-based features of the URLs contain a wealth of information for detecting malicious URLs.

MATERIALS AND METHODS

Features selection for phishing detection: The previous studies demonstrate that the phishing URLs can be discovered using a structural analysis of the host-based and lexical features of URLs. However, the phishers always change these techniques for phishing, thus making detection more difficult. Hence in this study, we developed a study of Lexical and Host-based features to ensure a stronger and an efficient detection approach. This section describes some of the features that we identified for phishing detection on the QR code. The 20 discriminative features listed in Table 1 are used in QRphish. These features can be roughly classified into two groups: lexical features and host features.

URL analyser: Most studies on phishing URLs rely on the host-based and the lexical features of the URL and the structures as shown in Fig. 1. The format of the URL is described as:

A URL contains two parts-the hostname and the path. Example;Consider a URL: 'www.nav6.org/research_v2/front_page.php' where the hostname is 'www. nav6. org' and the path is 'research_v2/front_page.php'.

The proposed approach analyses the lexical-based feature such as the presence of suspicious characters

Table 1: Discriminative features used in the qrphish

Category	Feature	Type
Lexical	Length of URL	Integer
Lexical	Number of dots	Integer
Lexical	Number of sub-domains	Integer
Lexical	Number of hexadecimal characters	Integer
Lexical	Number of suspicious character	Integer
Lexical	Number of sensitive words in URL	Integer
Lexical	Average domain length	Real
Lexical	Average path length	Real
Lexical	Domain brand-name distance	Integer
Lexical	Path brand-name distance	Integer
Lexical	Presence of IP address	Integer
Lexical	Longest domain length	Integer
Lexical	Longest path length	Integer
Host	Domain page rank	Integer
Host	Registering domain name	Charset
Host	Age of domain	Integer
Host	Update date	Integer
Host	Country	Charset
Host	Within domain	Integer
Host	Domain confidence level	Real

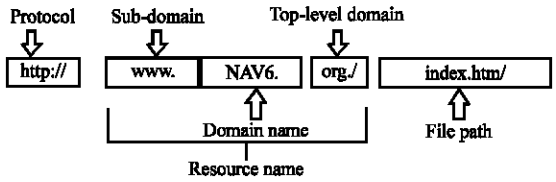

Fig. 1: Feature analyser

encoding the URL, hexadecimal character and malicious IP addresses to hide them. Various hosts-based features such as the age of the domain, page rank and domain confidence level are also analysed to avoid the end users from falling prey to the phishing attacks as shown in Fig. 2. This method is quite useful as the illegitimate users spoof their identities and it may pass the verification checks and content analysis and it may escape by avoiding the keywords of phishing. Some QR codes maycontain malicious links that urge the users to scan them and then it may lead them to fake websites.

Lexical features: Lexical features are the word-based properties of the URL itself and they have been widely used in the literature (Fette *et al*., 2007; Zhang *et al*., 2007a, b; Whittaker *et al*., 2010; Basnet *et al*., 2008) for detecting phishing attacks. These properties include the length of the entire URL, the length of the hostname as well as the number of dots in the URL, binary feature for each token in the hostname (delimited by '.') and in the path URL (strings delimited by '_', '?', '.', '-', ' = ' and '/'). We defined 13 lexical based features, five of which are elaborated as.

IP address: Phishing URLs often contain IP addresses tohide the actual URL of the website. URL detection

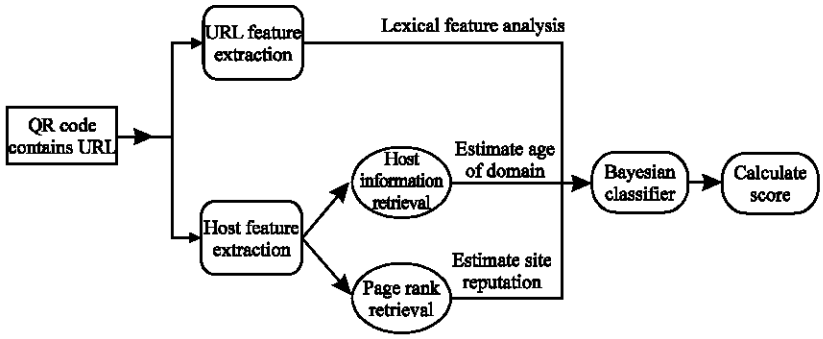

Fig. 2: URL features extraction

methods look for an IP address in the URL and add to the phishing score if one is found. However, the legitimate websites also sometimes use IP addresses; especially for internal private devices that are not accessible to the public. Network devices such as routers, servers and networked printers are often accessed using an IP address.

Hexadecimal characters: The web browsers know the hexadecimal values and they can be used in URLs by preceding the hexadecimal value with a '%' symbol. For instance, the value 20% is the hexadecimal equivalent of the space character on the keyboard. Usually, the phishers use hexadecimal values to cover the actual letters and numbers in the URL.

Suspicious character: Spoofguard (Chou *et al*., 2004) identified two characters common in phishing URLs, the '@' and the '-' character. The username precedes the '@' symbol and the target URL follows the '@' symbol. A @ symbol in a URL causes the string to the left to be disregarded with the string on the right treated as the actual URL for retrieving the page which is a phishing site. For example: the URL "http://www.bankofamerica.com@phishingsite.com" will navigate to the destination URL which is "phishingsite.com" and will attempt to login using "www.bankofamerica.com" as the username. Hence, the actual URL of the website is disguisedand when combined with an IP address it can really hide the phishing site while the URL appears to be legitimate. The second suspicious character is a dash '-'. However, it was determined through experimentation that the dash is not a good indicator of a phishing site. Many legitimate URLs use a dash whereas few legitimate sites use a '@' symbol.

Number of dots in URL: This feature counts the number of dots in the URL. Phishing URL tends to use more dots in their URLs than the legitimate sites (Xiang *et al*., 2011).

Number of sensitive words in URL: Garera *et al*. (2007), the researchers summarised a set of eight sensitive words that frequently appear in phishing URLs and we have created this feature by counting the number of eight sensitive words that are found in a page URL (ALmomani *et al*., 2013). The sensitive words include "secure", "account", "webscr", "login", "ebayisapi", "signin", "banking" and "confirm".

Host-based features: Host-based features can describe "where" the website is hosted, "who" owns them and "how" they are managed. The properties of the hosts that are identified by the hostname as part of the URL are described. We have defined seven host-based features, two of which are elaborated as.

WHOIS features: WHOIS is a query and response protocol that provides information about a domain name or IP address. This feature checks the age of the webpage domain name. Many phishing sites are hosted on recently registered domains and thus have a relatively young age. In order to exploit that property, this feature measures the number of months since the domain name is first registered. One can use the WHOIS server to search for information such as dates of domain creation of the queried URL, ownership details and if the domain registration entry is not found on the WHOIS server then this feature will return-1, considering it suspicious.

Domain page rank: This feature represents the relative importance of a page (the page rank) within a set of web pages. Either the Phishing web pages are short lived and have a very low page rank or they do not exist. Page Rank is an algorithm used by the Google Search to rank websites in their search engine results (Simon, 2005). If the page rank value for a target webpage is not available, it gives a score value of "-1".

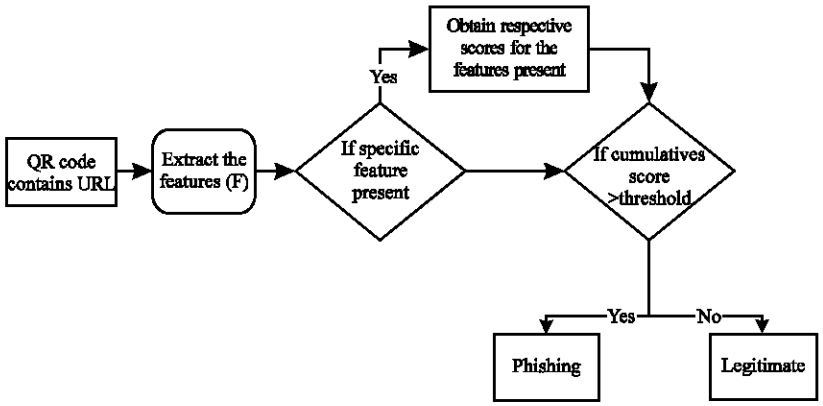

Fig. 3: Phishing URL classification

Approach: In this study, we have described the mechanism used for classifying the phishing QR code that contains a URL. To understand the proposed mechanism, it is necessary to identify the most efficient and correct classification methodology for setup experience. Furthermore, we explain the experimental setup for this study. Severa lmachine-learning techniques were used for the classification of phishing URLs. Machine learning techniques include classification of anexisting dataset using a classification model built on a pre-labelled dataset. Therefore, the experiment contains three phases. In the first phase, to build a labelled dataset, we collect QR codes containing URL and label these URLs as 'legitimate' or 'phishing'. In the second phase, a classifier model is trained by a classification algorithm. In the third phase, a QR code that contains a URL is collected and the trained model is used to classify this new URL. In this study, we describe the machinelearning algorithms usedin this study and the evaluation metrics which determine the accuracy and indicate the quality of the classification task.

Training set (bayes classifier): Usually used in spam filters, the Bayes model adopts it for a given label; the separate features of the URLs are distributed independent of the values of other features. Bayes theorem provides a way to calculate the probability of a hypothesis, for the event B, given the observed training data, represented as A:

$$P(B|A = \frac{P(A|B)P(B)}{P(A)} \quad (1)$$

This simple formula has enormous practical importance and is used for many applications. It is often easier to calculate the probabilities, P (A|B), P(A) and P(B), for the probability P(B|A) that is required. Extrapolating the Bayes rule and assuming that the malicious and legitimate web sites occur with equal probability, the posterior probability that the feature, vector x, belongs to a malicious URL is computed as:

$$P(B=1|A) = \frac{P(A|B=1}{P(B|A=1)+P(B|A=0)} \quad (2)$$

$$P(B|A) = \frac{P(A|B)}{P(A|B)+P(A|B')} \quad (3)$$

$$P(B|A) = \frac{P(A|B)P(B)}{P(A|B)P(B)+P(A|B')P(B')} \quad (4)$$

Where:

P(A) = Probability of feature F in phishing and legitimate dataset
P(B') = Legitimate dataset
P(B) = Phishing dataset
P(B(Phishing) = P(B'(Legitimate) = 0.5

The classifier has a training dataset of malicious phishing URLs and legitimate URLs (Friedman *et al*., 1997). The possibility of occurrence of each feature in the dataset is calculated and their respective scores are obtained (i.e.,) the occurrence of features in the dataset are counted up and the cumulative score is calculated. If the Cumulative score >Threshold, it is considered as phishing URL; else as legitimate URL as illustrated in Fig. 3:

- How many times does feature F(F1,F2) appear in the phishing dataset?
- How many times does feature F(F1,F2) appear in the legitimate dataset?

Let:

- F1 = Lexical features
- F2 = Host features

Calculating probability: In order to calculate the probability of a specific feature in the phishing dataset, we considered 1000 URLs, 500 phishing URLs and 500 legitimate URLs.

Feature F1 (Lexical features): The feature, F1, involves the occurrence of the lexical features that appeared in 283 phishing URLs and 19 legitimate URLs. Hence, its probability is calculated as follows:

$$P(B|A) = \frac{P(A|B)}{P(A|B)+P(A|B')}$$

$$P(B|A) = \frac{P(283|500)}{P(283|500)+P(19|500)} = 0.937$$

Since, P(B(Phishing)) = P(B‘(Legitimate)) = 0.5

Feature F2 (age of domain): The feature, F2 (host features), appeared in 312 phishing URLs and 68 legitimate URLs. Hence, its probability is calculated as follows:

$$P(B|A) = \frac{P(A|B)}{P(A|B)+P(A|B')}$$

$$P(B|A) = \frac{P(312|500)}{P(312|500)+P(68|500)} = 0.821$$

RESULTS AND DISCUSSION

This study consists of two stages. The first stage involves the development of a classification model (URL based, Host based and QR code feature) and classification of the QR code (phishing or legitimate). Thisstage forms the QRphish API which uses a trained model and classifies the QR code containing URL, based on the described features. The next stage creates an end-user solution by developing an android application, named QRphish which makes a call to the QRphish API and the popular blacklists and then labels each QR code containing URL aslegitimate or phishing. In this section, we describe the results and observations based on the classification mechanism.

Data collection a dataset: In this study, we describe how we collected the dataset for examination and how we built a true positive dataset of QR code containing phishing URL for this study. Data collection includes two phases as shown in Fig. 4, the first phase is collecting data from QR codes and the second phase is labelling the URL as legitimate or phishing.

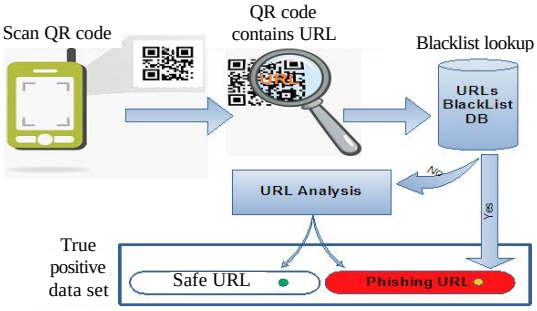

Fig. 4: Architecture for data collection

For this study, we required only QR code containing URLs. We used the QRphish application to collect such URLs. To initially label the URLs as phishing or legitimate for creating an annotated dataset, we used the PhishTank blacklist for URLs in every QR code and hence queried the dataset. PhishTank provides an open API and dataset for developers and researchers to integrate anti-phishing data into their applications at no charge. However, we observed that the blacklists did not update on the same day. Therefore, we used QRphish to label the URL as legitimate or phishing by studying the lexical and host-based features of URLs. When we used this method, we were able to phish the most recent malicious URLs which were not listed in the blacklist, thus improving the performance of phishing URLs. We collected over 22,750 malicious URLs from 1 Dec 2014-15 April 2015.

QRphish approach: QRphish is an android application written in Java using the android studio. The android studio enables the apache server to host a dataset. The Qrphish is published on the Google store. The dataset of the blacklist is hosted in a private server. The QRphish API provides a URL analysis. Once a QR code is scanned from a mobile, it classifies the URL as legitimate or phishing by using an existing trained classifiermodel on the private server. The main goal of this approach is to provide a real-time suggestion to users as we need the timefor features extraction and classification to be minimal. Therefore, the QRphish has multiprocessing modules thatextract features concurrently to saving time while processing. Once the classification iscompleted, the result is an output in the form of a Java Script Object Notation (JSON) string. Figure 5 shows the integration of the QRphish framework.

Table 2: Confusion matrix for classification

	Actual	
Expected	Phishing	Legitimate
Phishing	True positive	False positive
Legitimate	False negative	True negative

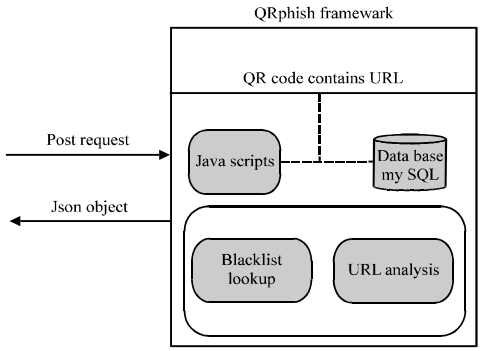

Fig. 5: Integration of the QRphish framework

Evaluation: In order to evaluate the efficiency of the classification method, a standard information retrieval metrics includes: Accuracy (ACC), Precision or Positive Predictive Value (PPV) and Recall, Sensitivity or True Positive Rate (TPR). Precision (also called as positive predictive value) is the fraction of the retrieved instances that are relevant. Sensitivity (also called as the true positive rate or the recall rate) measures the proportion of actual positives that are correctly identified and is complementary to the false negative rate. To explain the predicted positive, a 'confusion matrix' is used which is described in Table 2.

By this confusion matrix, we can calculate the precision Eq. 5 and the recall Eq. 6 for both the 'legitimate' and the 'phishing' classes. Furthermore, we can calculate the accuracy Eq. 7 of the classifier. It is the ratio of the correctly classified elements of either class to the total number of elements:

$$\text{Precision phishing} = \frac{TP}{(TP+FP)} \quad (5)$$

$$\text{Recall phishing} = \frac{TP}{(TP+FN)} \quad (6)$$

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (7)$$

Compare QRphish with existing systems: We use the Naive Bayes classification method. We have shown the

Table 3: Performance analysis with the existing systems

	Percentage				
Variables	Accuracy	Precision phishing	Precision legitimate	Recall pishing	Recall legitimate
Qrphish	93.34	94.78	96.54	93.14	96.11
Cantina	90.24	92.13	93.19	90.08	93.12
Cantina+	92.86	94.31	95.71	92.71	95.23

results of the classification using the Naive Bayes methods. From the 1500 phishing URLs, we detected 1401 URLs. Therefore, in our true positive dataset, we consider these 1401 phishing URLs and 500 legitimate URLs selected randomly from the QR code marked as 'legitimate' throughoutthe data collection process. This dataset was used for the rest of the experiments of our classification method. Weconcluded that the Naive Bayes classifier works very well for the phishing URL detection on our dataset and obtains a high accuracy of 93.34%, a recall of 93.14% for phishing URLs and 96.11% for legitimate cases. We compared our proposed method with other effective systems like Cantina and Cantina+The results are described in Table 3. Moreover,we obtained a high recall and precision for both the 'legitimate' and the 'phishing' classes. In our study, it is important to obtain a perfect precision of both 'legitimate' and 'phishing' classes to reduce the number of false positives and false negatives.

Table 3 shows that out of 1500 Phishing QR code with malicious URLs, the above results were obtained for identifying various lexical and host-based features.

Comparison of QRphish with the blacklists: The blacklists have a major limitation as they cause a delay while detecting the phishing URLs. It is important that the QR codes are detected as phishing as soon as possible to alert the users. In such situations, blacklists are obviously ineffective. We have compared the performance of PhishTank blacklist with QRphish. During the data collection stage, we collected URLs from scanning QR codes and immediately searched the URLs present in the QR codes in this blacklist. Since the blacklist takes some time to add newly constructed phishing URLs, we delayed for several days and thenrechecked the URLs that were collected several days ago in PhishTank blacklist. We have also used the QRphish to classify each of these QR codes as phishing or legitimate. We found that the QRphish detected 82.9% of the QR codes as phishing on 0 day. However, the strategies of the phishers always change hence, the detection mechanisms by blacklists are often unsuccessful. We combined our proposed features along with other features for a better phishing detection system for efficient real-time detection. This indicates that the QRphish can complement the blacklisting mechanism for QR code for detection of more phishing URLs in real-time.

CONCLUSION

In this study, we built the QRphish which is an effective mechanism to detect phishing on QR code. This approach uses not only the normal phishing detection features but also employs the QR code features, host-based features and URLs-based features. We have used a set of host-based features and URLs-based features which allowfor an effective and real-time detection of phishing on QR code. We also develop a Qrphish API that can be retrieved using a JSON Method. Moreover, it has been proved that our methodology works faster than popular blacklisting technique. Our detection mechanism has an ability to detect 82.9% more URLs than public blacklists such as PhishTank on 0 day with an accuracy of 93.34%. Finally, our approach is also a new and innovative technique on 0 day as we do not succeed with 100% accuracy hence, there could be a possibility of false negatives. However, our methodology can be coupled with the blacklisting mechanism and the URLs analyser mechanism for a better, more accurate real-time detection of phishing on QR code. In future works we add new rules to enhance the detection ofadvanceURL phishing and modify the QR code structurebyembedded a cryptograph key into QR code (Trust QR cod).

REFERENCES

Almomani, A., B. Gupta, T.C. Wan, A. Altaher and S. Manickam, 2013. Phishing dynamic evolving neural fuzzy framework for online detection zeroday phishing email phishing email. Ind. J. Sci. Technol., 6: 3960-3964.

Basnet, R.B., S. Mukkamala and A.H. Sung, 2008. Detection of phishing attacks: A machine learning approach. In: Studies in Fuzziness and Soft Computing, Prasad, B. (Ed.), Springer Berlin Heidelberg, New York, Pp: 373-383.

Chandrasekaran, M., K. Narayanan and S. Upadhyaya, 2006. Phishing email detection based on structural properties. Proceedings of the 9th Annual NYS Cyber Security Conference: Symposium on Information Assurance, June 14-15, 2006, Albany, New York, Pp: 1-7.

Chou, N., R. Ledesma, Y. Teraguchi, D. Boneh and J. Mitchell, 2004. Client-side defense against web-based identify theft. Proceedings of the 11th Annual Network and Distributed System Security Symposium, February 5-6, 2004, San Diego, CA., USA., pp: 1-16.

Fette, I., N. Sadeh and A. Tomasic, 2007. Learning to detect phishing emails. Proceedings of the 16th International World Wide Web Conference, May 8-12, ACM Press, Banff, Alberta, Canada, pp: 649-656.

Friedman, N., D. Geiger and M. Goldszmidt, 1997. Bayesian network classifiers. Mach. Learn., 29: 131-163.

Garera, S., N. Provos, M. Chew and A.D. Rubin, 2007. A framework for detection and measurement of phishing attacks. Proceedings of the 5th ACM Workshop on Recurring Malcode, October 29-November 2, 2007, Alexandria, VA., USA., pp: 1-8.

Kharraz, A., E. Kirda, W. Robertson, D. Balzarotti and A. Francillon, 2014. Optical delusions: A study of malicious QR codes in the wild. Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, June 23-26, 2014, Atlanta, GA., USA., pp: 192-203.

Ma, J., L. Saul, S. Savage and G. Voelker, 2009. Identifying suspicious URLs: An application of large-scale online learning. Proceedings of the 26th Annual International Conference on Machine Learning, June 14-18, 2009, Montreal, Quebec, Canada, Pp: 681-688.

Miyamoto, D., H. Hazeyama and Y. Kadobayashi, 2009. An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites. In: Advances in Neuro-Information Processing, Koppen, M., N. Kasabov and G. Coghill (Eds.). Springer, Berlin, Germany, ISBN: 978-3-642-02489-4, pp: 539-546.

Vidas, T., E. Owusu, S. Wang, C. Zeng and L. Cranor, 2012. QRishing: The susceptibility of smartphone users to QR code phishing attacks. Proceedings of the 19th ACM Conference on Computer and Communications Security, October 16-18, 2012, Raleigh, NC, USA., Pp: 1-12.

Whittaker, C., B. Ryner and M. Nazif, 2010. Large-scale automatic classification of phishing. Proceedings of the Network and Distributed System Security Symposium, February 28-March 3, 2010, San Diego, California, USA -.

Xiang, G., J.I. Hong, C.P. Rose and L.F. Cranor, 2011. CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inform. Syst. Secur., 14: 21-48.

Zhang, Y., J. Hong and L. Cranor, 2007a. CANTINA: A content-based approach to detecting phishing web sites. Proceedings of the 16th International Conference on World Wide Web, May 8-12, 2007, Banff, Alberta, Canada, pp: 639-648.

Zhang, Y., S. Egelman, L. Cranor and J. Hong, 2007b. Phinding Phish: Evaluating anti-phishing tools. Proceedings of the 14th Annual Network and Distributed System Security Symposium, February 28-March 2, 2007, Catamaran Resort Hotel, San Diego, CA., USA., pp: 1-16.