An Anti-Web Phishing Application for Analyzing the Security of Websites

T. P. Fowdur and R. Abdool Khader

Abstract- **Nowadays, one of the major internet security problems being faced is 'Web Phishing', whereby attackers get hold of the personal and sensitive information of the internet users. Sometimes, attackers create fake web pages just to mislead users and give them wrong information. With the increase of more and more sophisticated attacks like Whale Phishing, Spear Phishing, and Ransomware among others, internet users easily fall in attackers' traps. Most web browsers are not able to counteract or block these attacks and hence internet users consider the spoofed webpages to be legitimate ones and end up giving their details like credit cards details, passwords and usernames among others. In this paper, an application has been developed in Java that performs several tests on a URL, on the different hyperlinks present on the web page and on the content of the web page and provides a security rating to the internet user. Together with the percentage security, the user is informed if the web page is safe, doubtful or unsafe. The security ratings of several website domains such as, .gov, .co, .edu, .info, .mu, .ac, .org, .net, .com were also analysed. Furthermore, tests using independent samples ANOVA and Tukey HSD were performed and they revealed that there was a significant difference between the security ratings of the websites.**

Index Terms— **Phishing, IP, URL, Web Security**

I. INTRODUCTION

WITH the increase in internet usage, there has been a significant increase of phishing attacks. Usually attackers impersonate as someone else to gather information or to provide the user with misleading information. Often, it might happen that the user receives an e-mail from a phisher where he is asked to upgrade his profile through an embedded hyperlink. However, when the user enters the hyperlink and enters his personal information, the attacker gets hold of this information and misuses them. Since these e-mails are malicious copies of legitimate ones, it is hard for the human eyes to distinguish between the legitimate and the malicious e-mail. According to [1], in 2015, at least 230,280 phishing attacks were recorded which increased to 255,065 in 2016 worldwide.

Several publications have proposed schemes to counteract web phishing. An overview is given next. In [2], to secure online transactions, an Anti-Phishing Prevention Technique [APPT] was proposed where a One-Time Password [OTP] will be generated and communicated to the user via an alternate e-mail or via SMS [3]. After that, a token containing the user's information will be created and stored in the user machine. The password and the token together will authenticate the user. When the user logs on a webpage, his personal data are checked and the token name is retrieved. In [4], a rating of webpages was provided based on users' experiences on a webpage. If a majority of people have rated a webpage positively, it will display 'This site is safe', if the majority have rated negatively, it will display 'This site is unsafe', else it will display 'Unknown site'. In [5], if alterations were detected on webpages, the webmasters are alerted and if a user browses a phishing webpage or is about to download a malicious file, a warning is displayed to him. Also, a transparency report is displayed to the user. Furthermore, several web extensions can be used to prevent phishing attacks. In [6], AntiPhish was used which is suitable for non-experienced web users as it keeps track of the user's personal information. It scans a webpage and if it finds it malicious, it prevents the submission of personal information to that webpage. However, AntiPhish is limited to webpages written in HTML. In [7], Link Guard Algorithm is proposed which can detect and stop 195 out of 203 attacks. This algorithm analyses the differences between the visual link and the actual link and calculates the similarity of the URI of the hyperlink with that of the legitimate website. If they are not the same, then it is considered to be an attack. In [8], ratings and reviews are collected from experienced users and the ratings of the webpages are displayed as traffic lights next to the search engine. In [9], GeoTrust developed TrustWatch where information is displayed to users so that the identity of websites for e-commerce services can be verified. TrustWatch can also block pop-up windows and report suspicious webpages. In [10], the web extension GoldPhish is proposed where the logo of suspicious webpages is extracted and are converted to text. The text is queried as a google search and the result obtained is compared with the suspicious webpage. If the results do not match, then it is considered as a possible attack. In [11], an approach based on K-Means and Naïve-Bayes was proposed to check the behaviour of browsed webpages. With this method, approximately 18,480 unique phishing webpages have been detected. Firstly, a K-Means Classifier is used and then a Naïve-Bayes Classifier is used and based on the results, the webpage is rated as phishing, non-phishing or suspicious. In [12], it has been proposed that through the source code, phishing webpages can be detected. For example, if the logo loads from an external

T. P. FOWDUR is with the Department of Electrical and Electronic Engineering, University of Mauritius (email: p.fowdur@uom.ac.mu)

R. ABDOOL KHADER is with the Department of Electrical and Electronic Engineering, University of Mauritius (email: bibi.abdool8@umail.uom.ac.mu)

Manuscript received April 10, 2018; accepted June 1, 2018.
DOI: 10.17694/bajece.435864

link, it is a phishing characteristic. Another characteristic would be if the URL contains characters such as '@' and '_'. Phishing webpages are normally short-lived and the content has language anomalies. The presence of pop-up windows asking to update and validate accounts usually means that the webpage has been compromised.

In this paper, an application was developed in Java to give a percentage security rating to a webpage based on different features such as the Uniform Resource Locator (URL), the Hyper Text Markup Language (HTML) source code and the content of the webpage. However, compared to the previous proposed solutions, this application conducts its tests on nine sets of domains which are '.com', '.net', '.org', '.ac', '.gov', '.mu', '.info', '.edu', and '.co'. The percentage security obtained are recorded and an Analysis of Variance (ANOVA) test is performed on them with the HSD Tukey test to determine whether the difference between the means of the percentage security is significant or not.

The organisation of this paper is as follows. Section 2 gives an overview of the different Phishing attacks and some existing solutions. Section 3 describes the methodology employed to conduct the research. Section 4 describes the tests performed on the application and the results obtained and Section 5 concludes the paper.

II. BACKGROUND

This section starts with an overview of the different types of web-phishing attacks that have been recorded worldwide, followed by some existing solutions.

2.1 Overview of web-phishing attacks

Different types of web-based phishing attacks have been encountered till now, for instance:

a. Deceptive Phishing [13]: in this case, the user receives an e-mail with an embedded hyperlink and he will be asked to update his account or to warn him about system failures. When the user browses the provided hyperlink, he will be asked to enter his personal information and login credentials. However, the success of this type of attacks depend on how closely the phishing webpage is similar to the legitimate one.

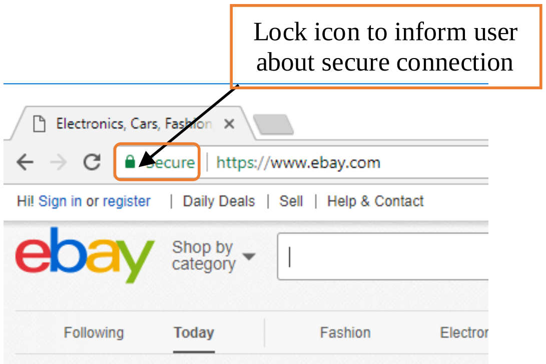
b. Spear Phishing [14]: this is the most common type of phishing where the goal is to lure users to click on malicious hyperlinks so that the attackers get hold of personal information.

c. Pharming [15]: in this case, the Domain Name System (DNS) server is the target and the Internet Protocol (IP) addresses are altered. When a user browses a webpage, he will be redirected to the webpage desired by the attacker.

d. Clone Phishing [16]: legitimate e-mails are cloned and the attachments and hyperlinks are altered to redirect the users to malicious webpages where their credentials are captured by the attacker.

2.2 Security features in Web Browsers

a. Connection Security: for some webpages, a lock icon is displayed in the location bar to inform the user that the connection to that particular webpage is safe. For example, this icon appears when a user browses to 'https://www.ebay.com'.

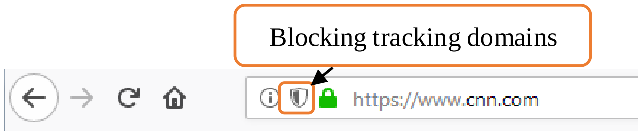


Lock icon to inform user about secure connection

Electronics, Cars, Fashion
Secure
https://www.ebay.com
Hi! Sign in or register | Daily Deals | Sell | Help & Contact
ebay
Shop by category
Following
Today
Fashion
Electror

Figure 1: Lock icon for secure connection

b. Protection against trackers [18]: information about user's browsed webpages can be collected by trackers. Furthermore, trackers also keep track of the device that the user is using to access the webpages. To seek protection from these trackers, Mozilla Firefox has developed a security feature to block them.



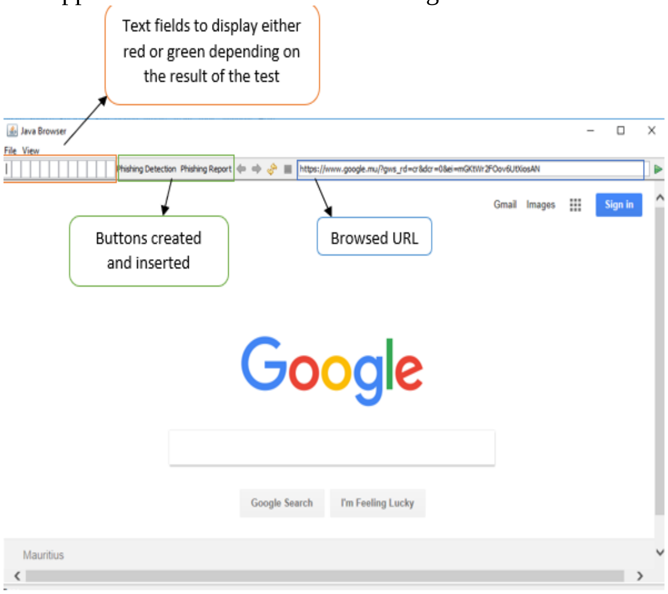
Blocking tracking domains

Figure 2: Protecting Mozilla Firefox from trackers

III. PROPOSED ANTI-PHISHING APPLICATION

3.1 Software Architecture

The application interface is shown in Figure 3.



Text fields to display either red or green depending on the result of the test

Buttons created and inserted

Browsed URL

Figure 3: Application Interface

Twelve text fields are created to be set to either red or green depending on the result of the different tests conducted. Furthermore, two buttons are created namely ‘Phishing Detection’ and ‘Phishing Report’. All these are implemented in the Java Web Browser. When ‘Phishing Detection’ button is clicked, tests will be carried out on the URL, HTML and content of the webpage and the text fields will be set accordingly. And when ‘Phishing Report’ button is clicked, a report is provided to the user based on the results of the different tests.

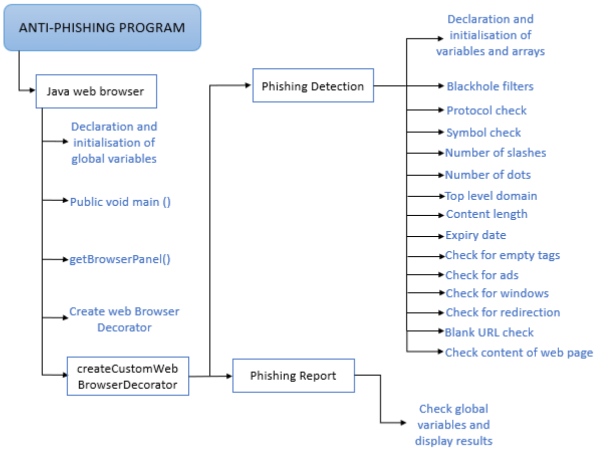


Figure 4: Software Architecture

The architecture of the anti-phishing software is shown in Figure 4. The Java web Browser declares and initialises the global variables and contains two buttons: ‘Phishing Detection’ and ‘Phishing Report’. The pseudocode for ‘Phishing Detection’ is given in Section 3.2 and that of ‘Phishing Report’ in Section 3.3.

3.2 Pseudocode for ‘Phishing Detection’

```
Declare and initialise global variables
txt_protocol, txt_symbol, txt_slashes, txt_dots, txt_tld, txt_content, txt_length, txt_date, txt_tag, txt_ads, txt_window, txt_redirect, txt_blankurl, txt_content: set to either red or green depending on the result of the tests
Create Java web browser
Create text fields degree1 to degree12 null, of length ‘1’
degree1 – degree12: Background set to either red or green depending on the result of the tests.
Declare and initialise scale to ‘0’: scale counts the number of anomalies detected.
Declare and initialise arrays for malicious words and percentage spam
Extract browsed URL
Pass URL in blackhole filter and check if present
If yes,
Set text fields degree1 to degree12 to red and display ‘Phishing Webpage’
Else,
Extract protocol
If protocol = https,
Set degree1 to green and set txt_protocol to green
Else,
Set degree1 to red and add 1.0 to scale
Check for ‘@’ and ‘_’
If present,
Set degree2 to red and add 1.0 to scale
Else,
Set degree2 to green and set txt_symbol to green
If no. of slashes > 5
Set degree3 to red and add 1.0 to scale
Else,
Set degree3 to green and set txt_slashes to green
If no. of dots > 5
Set degree4 to red and add 1.0 to scale
Else,
Set degree4 to green and set txt_dots to green
Extract Top-Level Domain
If (tld > 4) && (tld < 2),
Set degree5 to red and add 1.0 to scale
Else,
Set degree5 to green and set txt_tld to green
If content length of browsed URL = -1
Set degree6 to red and add 1.0 to scale
Else,
Set degree6 to green and set txt_content to green
If (expiry date < last modified date),
Set degree7 to red and add 1.0 to scale
Else,
Set degree7 to green and set txt_date to green
If empty tags present in source code
Set degree8 to red and add 1.0 to scale
Else,
Set degree8 to green and set txt_tag to green
If ads present in source code
Set degree9 to red and add 1.0 to scale
Else,
Set degree9 to green and set txt_ads to green
If pop-up present in source code
Set degree10 to red and add 1.0 to scale
Else,
Set degree10 to green and set txt_popup to green
If URL is redirected,
Set degree11 to red and add 1.0 to scale
Else,
Set degree11 to green and set txt_redirect to green
If webpage contains blank URL,
Set degree12 to red and add 1.0 to scale
Else,
Set degree12 to green and set txt_blankurl to green
Read content of webpage
Compare each word of webpage against elements in array of malicious words
Declare percent of type double and initialise to ‘0.0’: percent counts the percentage of all malicious words[19] that have been detected.
If present,
Add respective percentage spam to percent
If (percent >50),
Add 3.0 to scale
Result = 100 – ((scale/15)*100)
```

```
Display result
If (result > 70),
Display ‘High Safety Level’ and set txt_content to green
If (result <70) && (result > 50),
Display ‘Average Safety Level’ and set txt_content to red
Else,
Display ‘Not Safe’
```

As it can be observed in the above pseudocode, different checks are performed and based on these results, the text fields are set to either red or green and a value of 1.0 or 0.0 is added to scale. Then the percentage security is calculated and if it is greater than 70, the webpage is considered as having a high safety level. If the percentage safety is between 50 and 70, then it is considered to be having an average safety level, else it is considered as being not safe.

3.3 Pseudocode for ‘Phishing Report’

```
Create button ‘Phishing Report’
Declare and initialise local variables
Pro: secure or not depending on content of txt_protocol
Sym: contains symbols or not depending on the content of txt_symbol
Slash: more than 5 slashes or not depending on the content of txt_slash
Dot: more than 5 dots or not depending on the content of txt_dot
Tld: valid or not depending on the content of txt_tld
Len: -1 or not depending on the content of txt_length
Date:  last modified date greater than expiry date or not depending on the content of ‘txt_date’
Tag: contains tag or not depending on the content of ‘txt_tag’
Ads: contains ads or not depending on the content of txt_ads
Win: opens another window or not depending on the content of txt_window
Redirect: redirects webpage or not depending on the content of txt_redirect
Burl: contains blank url or not depending on the content of txt_blankurl
Con: set to ‘safe’, ‘doubtful’ or ‘unsafe’ depending on the content of txt_content
If (txt_protocol = green)
Set pro to ‘secure’
Else,
Set pro to ‘not secure’
If (txt_symbol = green)
Set sym to ‘not contain ‘@’ and ‘_’’
Else,
Set sym to ‘contain ‘@’ and ‘_’’
If (txt_slashes = green)
Set slash to ‘not contain more than 5 slashes’
Else,
Set slash to ‘contain more than 5 slashes’
If (txt_dots = green)
Set dot to ‘not contain more than 5 dots’
Else,
Set dot to ‘contain more than 5 dots’
If (txt_tld = green)
Set tld to ‘valid tld
Else,
Set tld to ‘invalid tld’
If (txt_length = green)
Set len to ‘not -1’
Else,
Set len to ‘-1’
If (txt_date = green)
Set date to ‘expiry date not less than last modified date’
Else,
Set date to expiry date less than last modified date
If (txt_tag = green)
Set tag to ‘not contain empty tags’
Else,
Set tag to ‘contain empty tags’
If (txt_ads = green)
Set ads to ‘not contain ads’
Else,
Set ads to ‘contain ads’
If (txt_window = green)
Set win to ‘not open another window’
Else,
Set win to ‘open another window
If (txt_symbol = green)
Set sym to ‘not contain ‘@’ and ‘_’’
Else,
Set sym to ‘contain ‘@’ and ‘_’
If (txt_blankurl = green)
Set burl to ‘not contain blank URLs’
Else,
Set burl to ‘contain blank url’
If (txt_content = green)
Set con to ‘safe content’
Else If (txt_content = green),
Set con to ‘doubtful content’
Else,
Set con to ‘content not safe’
Concatenate all the strings
Display result
```

It can be observed from this pseudocode that the global variables are set to green or red. Depending on these global variable, the local variables are set. At last, all the local variables are concatenated and displayed to the user.

ANOVA [20] is a statistical method that is used to test the differences between the means of two or more groups of data. This test is done on a general basis among the means. One-way ANOVA [21] is when only one qualitative variable is taken into consideration. Each set must contain same number of elements and must be normally distributed with the same variance.

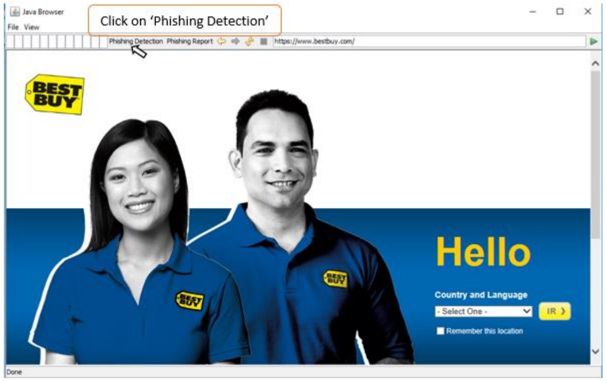
In this work, for nine domains, the security percentage of 50 websites have been recorded. The mean security level of each of these domains was then computed and an independent samples one-way ANOVA was used to determine if the difference between the means was significant.

IV. RESULTS

As mentioned earlier, nine different domains of websites were tested, each containing fifty (50) links.

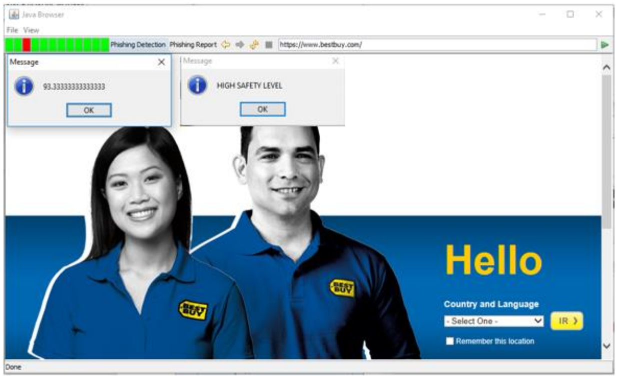
4.1 Testing with '.com'

The tested URL is 'https://www.bestbuy.com'



Click on 'Phishing Detection'

Figure 5: Navigating to the desired webpage



Message

93.3333333333333

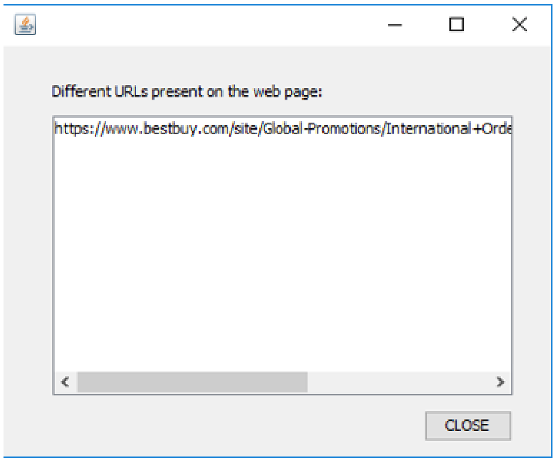
OK

Message

HIGH SAFETY LEVEL

OK

Figure 6: Result of test

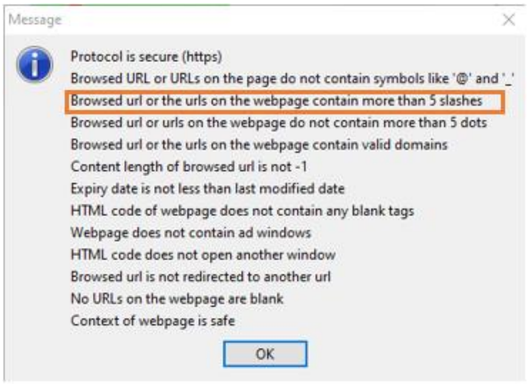


Different URLs present on the web page:

https://www.bestbuy.com/site/Global-Promotions/International+Orde

CLOSE

Figure 7: Hyperlinks present on webpage



Message

Protocol is secure (https)
Browsed URL or URLs on the page do not contain symbols like '@' and '_'
Browsed url or the urls on the webpage contain more than 5 slashes
Browsed url or urls on the webpage do not contain more than 5 dots
Browsed url or the urls on the webpage contain valid domains
Content length of browsed url is not -1
Expiry date is not less than last modified date
HTML code of webpage does not contain any blank tags
Webpage does not contain ad windows
HTML code does not open another window
Browsed url is not redirected to another url
No URLs on the webpage are blank
Context of webpage is safe

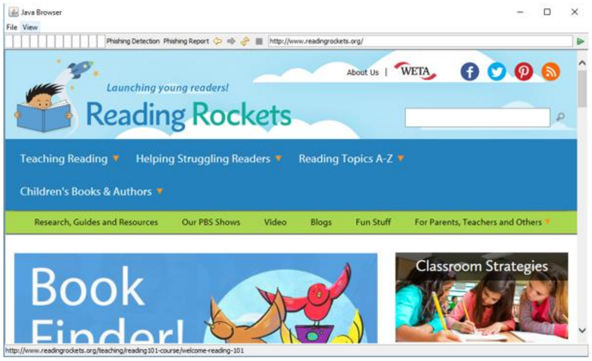
OK

Figure 8: Phishing Report

Figure 5 shows the navigated webpage. When 'Phishing Detection' is clicked, a percentage security of 93.33 is shown along with 'High Safety Level' as shown in Figure 6. Figure 7 shows the different hyperlinks present on the webpage and Figure 8 is displayed when the user clicks on 'Phishing Report'. The selected result in Figure 8 shows the anomaly that has been detected by the application.

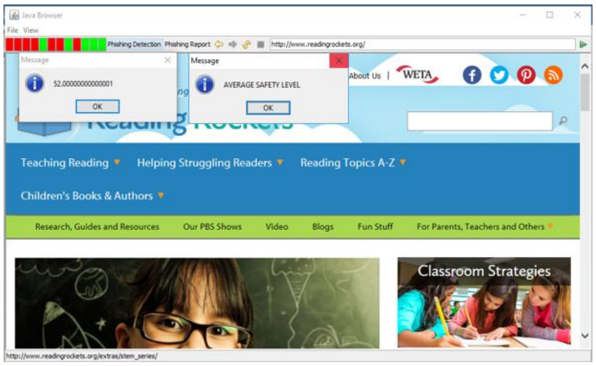
4.2 Testing with '.org'

The tested URL is 'www.readingrockets.org'



Java Browser
File View
Phishing Detection
Phishing Report
http://www.readingrockets.org/
About Us
WETA
Launching young readers!
Reading Rockets
Teaching Reading
Helping Struggling Readers
Reading Topics A-Z
Children's Books & Authors
Research, Guides and Resources
Our PBS Shows
Video
Blogs
Fun Stuff
For Parents, Teachers and Others
Book Finder!
Classroom Strategies
http://www.readingrockets.org/teaching/reading101-course/welcome-reading-101

Figure 9: Navigating to the desired webpage



Message

52.0000000000001

OK

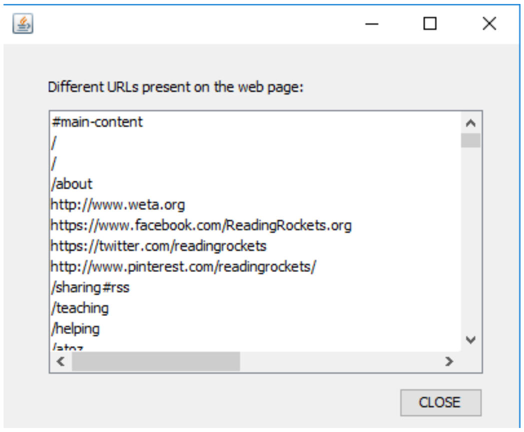
Message

AVERAGE SAFETY LEVEL

OK

Java Browser
File View
Phishing Detection
Phishing Report
http://www.readingrockets.org/
About Us
WETA
Teaching Reading
Helping Struggling Readers
Reading Topics A-Z
Children's Books & Authors
Research, Guides and Resources
Our PBS Shows
Video
Blogs
Fun Stuff
For Parents, Teachers and Others
Classroom Strategies
http://www.readingrockets.org/extras/stem_series/

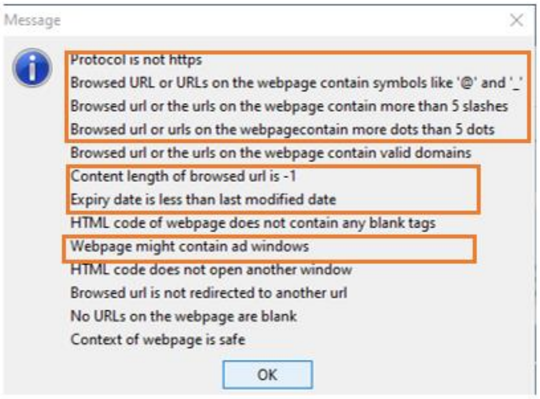
Figure 10: Result of test



Different URLs present on the web page:

```
#main-content
/
/
/about
http://www.weta.org
https://www.facebook.com/ReadingRockets.org
https://twitter.com/readingrockets
http://www.pinterest.com/readingrockets/
/sharing#rss
/teaching
/helping
/atoz
```

Figure 11: Hyperlinks present on webpage



```
Protocol is not https
Browsed URL or URLs on the webpage contain symbols like '@' and '_'
Browsed url or the urls on the webpage contain more than 5 slashes
Browsed url or urls on the webpagecontain more dots than 5 dots
Browsed url or the urls on the webpage contain valid domains
Content length of browsed url is -1
Expiry date is less than last modified date
HTML code of webpage does not contain any blank tags
Webpage might contain ad windows
HTML code does not open another window
Browsed url is not redirected to another url
No URLs on the webpage are blank
Context of webpage is safe
```

Figure 12: Phishing Report

The navigated webpage is shown in Figure 9 and after clicking on 'Phishing Detection', the results are displayed as shown in Figure 10. Figure 11 shows the different hyperlinks present on the webpage and Figure 12 shows the different anomalies that have been detected by the application.

The mean security percentages for each domain is shown in Figure 13.

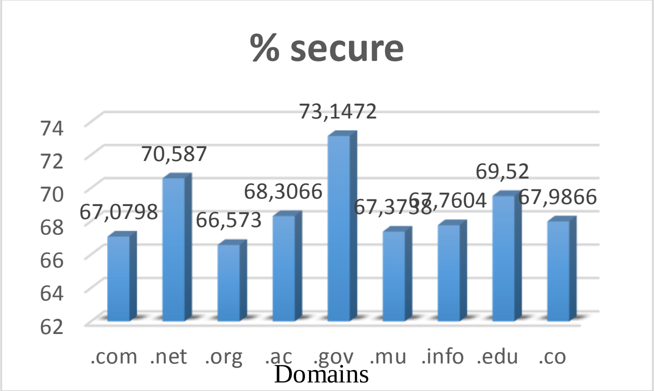


Figure 13: Bar Chart for average percentage security

An ANOVA Independent test and an HSD Tukey test were performed on the nine sets of results that were obtained. The value obtained for F_{obs} and F_{table} were 2.7618 and 1.9594 respectively. Hence, it can be concluded that the difference between the means is significant. Furthermore, the HSD Tukey test revealed that the pairwise differences between the means of '.com and .gov' and '.gov and .mu' are significant.

V. CONCLUSIONS AND FUTURE WORK

Web phishing is a rising security risk which target many people and provide them with misleading information. In this paper, an application has been designed which gives a rating when the button 'Phishing Detection' is clicked. The rating is based on tests on different characteristics of the URL, the HTML code and the content of the webpage. Compared to [12], the application can check for anomalies in the content of the webpages and based on the results, a rating is displayed to the user.

The program also rates a webpage based on the different tests carried out. In contrast with [4] where the rating is based on the users' experiences. Moreover, the ANOVA test performed confirmed that the difference between the security of different domains is significant. Finally, it will be interesting to investigate the possibility of integrating the schemes developed in [22] and [23] into the proposed application.

ACKNOWLEDGMENT

My gratitude to all those who have helped and supported me throughout this research work.

REFERENCES

[1] Aaron G. and Rasmussen R., Global Phishing Survey: Trends and Domain Name Use in 2016, pp. 5, 2017.

[2] Chelliah G. A and Aruna S., Preventing Phishing Attacks Using Anti-Phishng Prevention Technique, International Journal of Engineering Development and Research, pp. 60- 63, 2014.

[3] Khan A. A., Preventing Phishing Attacks using One Time Password and User Machine Identification, International Journal of Computer Application, vol. 68, no. 3, pp. 7-11, 2013.

[4] Avast Support. 2018. Avast Online Security browser extension – Getting Started | Official Avast Support. [Online] Available at: https://support.avast.com/en-au/article/18/ [Accessed 23 November 2017]

[5] Safe Browsing – Google Safe Browsing. 2018. [Online] Available at: https://safebrowsing.google.com/ [Accessed 9 January 2018]

[6] Kirda E. and Krugel C., Protecting Users Against Phishing Attacks, The Computer Journal, vol. 00, no. 0, pp. 1-8, 2005.

[7] Naresh U., Sagar U. V. and Reddy C. V. M, Intelligent Phishing Website Detection and Prevention System by Using Link Guard Algorithm, IOSR Journal of Computer Engineering (IOSR-JCE), vol. 14, no. 3, pp. 28-36, 2013.

[8] WOT Services Ltd.,2018. Web of Trust (WOT) – Crowdsourced web safety | WOT (Web of Trust). [Online] Available at: https://www.mywot.com/en/aboutus [Accessed 10 September 2017]

[9] TrustWatch – WEB SITE VERIFICATION SERVICE. [Online] Available at: https://www.trustico.co.in/material/DS_TrustWatch.pdf [Accessed 29 October 2017]

[10] Jain A. K. and Gupta B. B., Phishing Detection: Analysis of Visual Similarity Based Approaches, Security and Communication Networks, vol. 2017, pp. 1-20, 2017

[11] Wanawe K., Awasare S. and Puri N. V., An Efficient Approach to Detecting Phising A Web Using K-Means and Naïve- Bayes Algorithms, International Journal of Research in Advent Technology, vol. 2, no. 3, pp. 106-111, 2014

[12] Alkhozae M. G. and Batarfi O. A., Phishing Websites Detection based on Phishing Characteristics in the Webpage Source Code, International Journal of Information and Communication Technology Research, vol. 1, no. 6, pp. 283-291, 2011

[13] The State of Security. 6 Common Phishing Attacks and How to Protect Against Them, 2018. [Online] Available at: https://www.tripwire.com/state-of-security/security-awareness/6-common-phishing-attacks-and-how-to-protect-against-them/ [Accessed 8 April 2018]
[14] Ho G., Sharma A., Javed M., Paxson V. and Wagner D., Detectimg Credential Spearphishing Attacks in Enterprise Settings, usenix, pp. 469-484, 2017.
[15] The State of Security. 6 Common Phishing Attacks and How to Protect Against Them, 2018. [Online] Available at: https://www.tripwire.com/state-of-security/security-awareness/6-common-phishing-attacks-and-how-to-protect-against-them/ [Accessed 8 April 2018]
[16] Chaudhry J. A, Chaudry S. A. and Rittenhouse R. G., Phishing Attacks and Defences. International Journal of Security and its Application, vol. 10, no. 1, pp. 247-256, 2016.
[17] Tracking Protection | Firefox Help. 2018. Tracking Protection | Firefox Help. [Online] Available at: https://support.mozilla.org/en-US/kb/tracking-protection [Accessed 3 January 2018]
[18] Lane D. M., Analysis of Variance. Online Statistics Education B. pp. 517-598
[19] Sun T, Spam Filtering based on Naïve Bayes Classification, pp. 1-42, 2009
[20] Heron E., Analysis of Variance – ANOVA, 2009. [Online] Available at: https://www.tcd.ie/medicine/neuropsychiatric-genetics/assets/pdf/2009_3_ANOVA.pdf [Accessed 9 December 2017]
[21] Analysis of Variance (ANOVA) [Online] Available at: https://www.calvin.edu/~scofield/courses/m143/materials/handouts/anova1And2.pdf [Accessed 15 December 2017]
[22] Rami M. Mohammad, Fadi Thabtah and Lee McCluskey, "Intelligent rule-based phishing websites classification", IET Information Security Volume: 8, Issue: 3, pp. 153 – 160, May 2014, DOI: 10.1049/iet-ifs.2013.0202.
[23] Yasin Sönmez, Türker Tuncer and Hüseyin Gökal, "Phishing web sites features classification based on extreme learning machine", 6th IEEE International Symposium on Digital Forensic and Security (ISDFS), 22-25 March 2018, Antalya, Turkey, DOI: 10.1109/ISDFS.2018.8355342.

BIOGRAPHIES

Dr. T.P. Fowdur received his BEng (Hons) degree in Electronic and Communication Engineering with first class honours from the University of Mauritius in 2004. He was also the recipient of a Gold medal for having produced the best degree project at the Faculty of Engineering in 2004. In 2005 he obtained a full-time PhD scholarship from the Tertiary Education Commission of Mauritius and was awarded his PhD degree in Electrical and Electronic Engineering in 2010 by the University of Mauritius. He is presently an Associate Professor at the Department of Electrical and Electronic Engineering at the University of Mauritius. His research interests include Communications Theory, Multimedia Communications, Mobile and Wireless Communications, Networking, Security and Internet of Things.

Miss R. Abdool Khader is currently a final year student for BSc (Hons) Information and Communication Technologies at the University of Mauritius. Her research interest include Networking, Security and system development.