PROTECTCING USERS AGAINST PHISHING ATTACKS
WITH ANTIPHISH

Third Year Computer Engineering

By

Abhay Anavkar (02)

Pooja Balmiki (04)

Tanvi Mhatre (26)

Neha Yadav (67)

Under the supervision of

Prof. Dr. Monika Wagh

November 2020

SHREE L. R. TIWARI COLLEGE OF ENGINEERING, MIRA ROAD

(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)

DEPARTMENT OF COMPUTER ENGINEERING

2020-21

A Project Report

On

PROTECTCING USERS AGAINST PHISHING ATTACKS WITH ANTIPHISH

Submitted infulfillment of the requirements for the course in

Business Communication and Ethics

THIRD YEAR ENGINEERING

November 2020

SHREE L. R. TIWARI COLLEGE OF ENGINEERING, MIRA ROAD

(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)

DEPARTMENT OF COMPUTER ENGINEERING

2020-21

SHREE L. R. TIWARI COLLEGE OF ENGINEERING

(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)

CERTIFICATE

This is to certify that the project titled *‘Protecting users against phishing attacks with antiphish’* has been completed under our supervision and guidance by the following students:

Abhay Anavkar

Pooja Balmiki

Tanvi Mhatre

Neha Yadav

In fulfillment of the requirements for the course in ***Business Communication and Ethics,*** Third Year in Computer Engineering, as prescribed by the University of Mumbai during the Academic Year 2020-2021. The said work has been assessed and is found to be satisfactory.

Signature of HOD	**Signature of Project Supervisor**
Name:_________________	**Name: ___________________**
Date:_________	**Date:_________**

Dr. S. Ram Reddy

Principal

ACKNOWLEDGMENT

I would like to express my gratitude to all those people who helped me provided support, talked things over, read, wrote, offered comments, allowed me to quote their remarks and assisted in editing, proof reading and design.
Above all I would like to thank the following personnel in believing me and helped me in making this project:-

Our HOD, Prof. Neelam mam
Our Project Supervisor, Prof. Monica W.
Our Principal, Dr. Ram Reddy.

Abhay Anavkar

Pooja Balmiki

Tanvi Mhatre

Neha Yadav

ABSTRACT

Most of the anti-phishing solutions are having two major limitations; the first is the need of a fast access time for a real-time environment and the second is the need of high detection rate. Black-list-based solutions have the fast access time but they suffer from the low detection rate. In this project, we propose a novel approach to protect against phishing attacks using auto-updated white-list of legitimate sites accessed by the individual user. Our proposed approach has both fast access time and high detection rate. When users try to open a website which is not available in the white-list, the browser warns users not to disclose their sensitive information. Furthermore, our approach checks the legitimacy of a webpage using hyperlink features. Moreover, our proposed system is efficient to detect various other types of phishing attacks (i.e., Domain Name System (DNS) poisoning, embedded objects, zero-hour attack).

CONTENT

CHAPTER 1: INTRODUCTION

1.1 DEFINITION

Phishing is a form of online criminal trick of stealing victims' personal information by sending them spoofed emails urging them to visit a forged webpage that looks like a true one**.** Phishing is a cyber security threat which is performed with the help of social engineering techniques to trick Internet users into revealing personal and secret information , An attacker needs a way to inject malicious software, or malware, into the victim's computer or mobile device. One of the ways this can be achieved is by phishing. Phishing has become a substantial threat for internet users and a major cause of financial losses. In these attacks the cybercriminals carry out user credential information and users can fall victim. The current solution against phishing attacks are not sufficient to detect and work against novel phishes.

1.2 OVERVIEW OF ANTIPHISH

In phishing attack, the attacker makes a fake webpage by copying or making a little change in the legitimate page, so that an internet user will not able to differentiate between phishing and legitimate webpages. When a fraudster sends an email or text message to a user that appears to originate from trusted source, such as a bank, as in our original example. By clicking on a link or opening an attachment in the phishing message, the user can unwittingly load malware onto their device. The malware then installs itself on the browser without the user's knowledge.

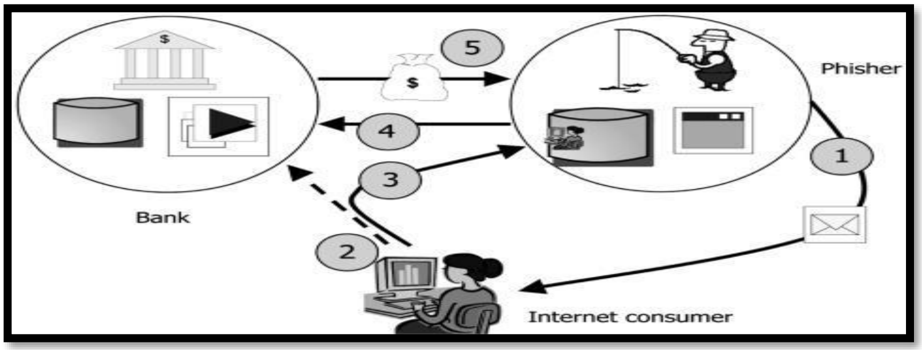


Fig:1 Illustration of Phishing Procedure

The malware records the data sent between the victim and specific targeted websites, such as financial institutions, and transmits it to the attacker.Phishing attacks fall into several categories:The earliest form of phishing attacks were e-mail based but the success rate from the point of view of the attackers is lower because many users have learned not to send sensitive information via e-mail. A possible reason is that many security-sensitive organizations such as banks do not provide interactive services based on e-mail where the user has to provide a password. Most organizations, obviously, use their web sites for providing interactive services because they can rely on encryption technologies such as SSL.

AntiPhish is based on the premise that for inexperienced, technically unsophisticated users, it is better for an application to attempt to check the trustworthiness of a web site on behalf of the user. Unlike a user, an application will not be fooled by obfuscation tricks such as a similar sounding domain name. AntiPhish is an application that is integrated into the web browser. It keeps track of a user's sensitive information (e.g., a password) and prevents this information from being passed to a web site that is not considered "trusted" (safe).

The development of AntiPhish was inspired by automated form-filler applications. Most browsers such as Mozilla or the Internet Explorer have integrated functionality that allows form contents to be stored and automatically inserted if the user desires.

1.3 STATEMENT OF PROBLEM

The web is most important for corporate and product information and it is platform for business transaction but it is also vulnerable for attackers to attacks on the web servers over the internet ,when web server is attacked then the status can be damaged and money can be lost.

There are many web security threats like eavesdropping on the net ,modification of users data,memory ,message traffic and DOS attack but among all these phishing scams have been receiving extensive press coverage because such attacks have been escalating in number and sophistication.

Many online service providers believe that their reputation is at stake and fear that users will loose confidence in electronic commerce. According to a study by Gartner, 57 million US Internet users have identified the receipt of e-mail linked to phishing scams and about 2 million of them are estimated to have been tricked into giving away sensitive information.Many times untrained users are not aware of the security risks which exists and they do not have the tools or sufficient knowledge to take effective countermeasures Therefore to make internet users feel safe a Antiphish application have developed to protect users confidential information.

1.4 AIMS AND OBJECTIVE

AntiPhish, a browser extension that aims to protect inexperienced users against spoofed web site-based phishing attacks. AntiPhish tracks the sensitive information of a legimate user and generates warnings whenever the user attempts to give away this information to a web site that is considered untrusted.

The tool has been implemented as a Mozilla Firefox plug-in and is free for public use. our approach checks the legitimacy of a webpage using hyperlink features. For this, hyperlinks from the source code of a webpage are extracted and apply to the proposed phishing detection algorithm.. our proposed system is efficient to detect various other types of phishing attacks (i.e., Domain Name System (DNS) poisoning, embedded objects, zero-hour attack).

CHAPTER 2: REVIEW OF RELATED LITERATURE

2.1 FROM REFERENCE BOOK/ONLINE JOURNAL

Several different solution for phishing have been developed during the past few years.These solution include governmental policies against online frauds,creating awareness to users and technology countermeasurs.

Review of these research improved our basics understanding towards this problem and helped us to build a more comphrensive research model for the current study.

Researches on this area focus on developing user skills to avoid misclassification of phishing websites as legimate one manually.

❖ Hybrid Features for detecting phishing email

Ma et al conducted a comprehensive research on developing tools and techniques to detect phishing emails using hybrid features. Phishing has become so malicious, complex and sophisticated that it is able to avoid filters and anti-phishing systems. Email servers now can be installed with malicious detection devices since phishing emails have instigated a lot of researchers to work on creating these techniques. But these efforts have not proved to be worthy enough to stop phishing emails.

The approach adopted was primarily includes extracting feature vectors from the emails that well denote the instances. The four elements which have been used to create the phishing detection model includes: Feature Generator, Machine Learning, Method Selection, Inductor and Feature Evaluation. The greater the information gained (generated by induction), the more valuable a feature (possible instance) will be. These components have illustrated an effectiveness of the phishing email detection and provide evidence.

❖ Neural Network to detect phishing attack

Zhang and Yuan proposed yet another phishing detecting approach that makes use of the neural network a machine learning technique. A large number of phishing attacks go unrecognized by users as the emails seem to be coming from a legitimate source. The research uses a large number of emails to detect phishing attacks. The detection model incorporates multilayer Feedforward Neutral Networks (NNs) by selecting and defining a set of features relating to the email structure and external links.

This NN model uses one input layer, one hidden layer, and one output layer. The methodology first extracts features at the pre-processing stage then implements Neural Network to classify them. The result shows that Neural Network generates a 95% accuracy level with a little misclassification. We propose that Neu ural Network is brilliant at detecting phishing attacks.

❖ Data Mining Approach

Smadi et al. [23] proposed a phishing detection model based on data mining algorithms, and using features extracted from different parts of emails. The main goal in this research was to improve the overall metrics values of classifying emails. This was done by laying emphasis on the pre-processing phase of extracting features and finding out the best algorithm to be used. The designed model classified emails into two types: legitimate and phishing emails. This classification is created by the features pulled out from the header and content of the emails tested.

To generate this detection mechanism, data mining algorithms were used. The experiment model attained 98.87% precision for random forest algorithm, depicting the advantage of using the pre-processing phase to extract the set of features from emails. The authors have intelligently used the pre-processing phase and increased the overall metrics of the model.

Taking into account a large number of factors that represent almost all kinds of phishing attacks resulted in low false positive rates and high accuracy of the detection mechanism. Comparing the results of the model with the previous researches it can be fairly concluded that the model proves to be the best in terms of accuracy and false positive rate for approved dataset.

Citation : Shaikh, Anjum & Shabut, Antesar & Hossain, Alamgir. (2016). A literature review on phishing crime, prevention review and investigation of gaps. 9-15. 10.1109/SKIMA.2016.7916190

2.2FROM WEBSITES

❖ Search engine-based techniques

All SEB techniques extract and use webpage text, images, or URLs as a search string to determine the popularity of a website using search engines to detect phishing. The techniques are different, however, in terms of (i) type and number of features extracted (text, URL, or images) from a webpage; (ii) number of search engines used to determine webpage popularity, (iii) number of top results used for matching; (iv) the underlying decision making algorithm; and (v) additional use of logic from other anti-phishing schemes. A brief description of these schemes is as follows:

Varshney et al. focused on the need of lightweight phishing detection approach using search engines. Authors identified the lightest possible features (page title and domain name) that can be extracted from a webpage without a complete webpage loading. Based on this, authors developed an intelligent anti-phishing chrome extension named lightweight phish detector (LPD). LPD not only detects but also suggests the authentic webpage to the user when a user reaches a deceptive or phishing page on the browser.

Ramesh *et al*. proposed a technique that collects and matches a group of domains having direct and indirect association with the domain of the suspicious webpage to detect phishing. Jun *et al.* proposed a scheme wherein the URL of a website is searched using popular search engines such as Google, Bing, and Yahoo. The number of search results obtained and their rankings are then used for classification.

Hung *et al*. proposed an approach that captures a screenshot of the webpage and extracts the website logo, which is then searched using Google image search. The returned keywords are then fed to Google text search, and if current domain name does not match any of the top 30 domain names returned in the search results, then the website is identified as phishing.

Xiang *et al*. proposed a technique that uses “site: declared brand domain ‘page domain’ ” as a Google search engine query and checks whether the returned results indicate the same domain name or not. If the returned results do not indicate the same domain name, keywords from the webpage visited by the user are extracted and searched. If the domain name does not appear in the top N search results, the URL is declared as phishing. They also proposed that before using the Google search engine query, the URL should be searched on the whitelist and the page should be passed through a login form filter. If the URL is on the whitelist or if the page does not contain any login form, it is declared as normal, and further processing is not carried out.

Dunlop *et al*. proposed a technique where an IE toolbar takes a snapshot of the current page and the image contents, including logos. The image contents and logos are converted to text, which is searched using the Google text search. The top level and second-level domains are matched with the top four links obtained from the Google search to detect phishing. An analysis of these schemes in terms of novelty, dataset, accuracy, and drawbacks is given in Table

Web link: https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1674

CHAPTER 3 :SCOPE

- Most of the anti-phishing solutions are having two major limitations; the first is the need of a fast access time for a real-time environment and the second is the need of high detection rate. Black-list-based solutions have the fast access time but they suffer from the low detection rate while other solutions like visual similarity and machine learning suffer from the fast access time.
- Our Anti-Phishing Service model will enable the client to quickly identify new phishing sites and take down them in a short span of time. Its scope range from monitoring of domains /sub-domains for anti-phishing, phishing & pharming monitoring & detection, malware & Trojan monitoring and detection, incident response, phishing site take down service.
- Web server logs provide vast information about a phishing attack. We use web server logs at various stages of phishing analysis.This will help us detect any phishing attacks which are not detected in the earlier method.

CHAPTER 4 :DESIGN

4.1 TECHNOLOGIES USED

- ❖ FRONT-END
  - Html
  - Css
  - Materalize CSS
  - Bootstrap
  - Javascript
  - Jquery

- ❖ BACKEND
  - PHP
  - Ajax
  - Xml

- ❖ DATABASE
  - Mysql

- ❖ XAMP SERVER

- ❖ APPLICATION PROGRAMMING INTERFACE
  - IOS Rest API

4.2 SYSTEM ARCHITECTURE

The architecture of our proposed system is divided into two modules as shown in Fig. 3. The first module is the URL and DNS matching module which contains a white-list, which is used to increase the running time and decrease the false negative rate. Our white-list main- tains two parameters, domain name and corresponding IP address. Whenever a user accesses a website, then the system matches the domain name of the current website with white-list. If the domain of the current website is matched with the white-list, then the system matches the IP address to take the decision. When the user ac- cess a website which is already present in the white-list, then our system matches the IP address of the corre- sponding domain to check the DNS poisoning attack.

Our white-list starts with zero; it means that at the be- ginning, there is no domain in the list and the white-list starts increasing once a user accesses the new webpages. When a user accesses a website, then there are two pos- sibilities, either the user is accessing the website for the first time or it is already visited by the user. If the user is accessing the website for the first time, then the domain of the website will not be present in the white-list. In that case, our second module starts working.

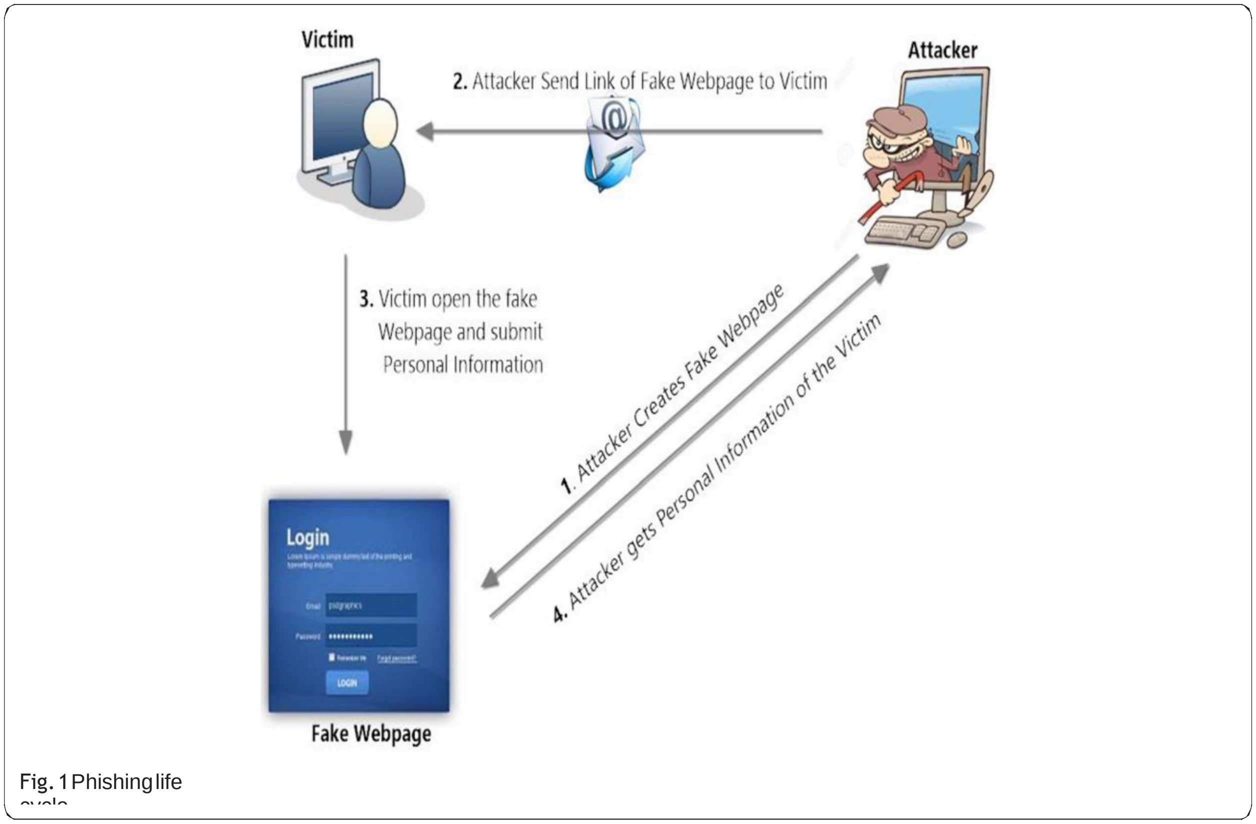

Victim
Attacker
2. Attacker Send Link of Fake Webpage to Victim
3. Victim open the fake Webpage and submit Personal Information
1. Attacker Creates Fake Webpage
4. Attacker gets Personal Information of the Victim
Login
LOGIN
Fake Webpage
Fig. 1 Phishing life cycle

Figure 2:Phishing attack

The second module is the phishing identification module, which checks whether a webpage is phishing. We extract the hyperlinks from the webpage and apply our phishing de- tection algorithm (the phishing detection algorithm is explained in Section 3.2). Our phishing detection algo- rithm examines the features from the hyperlinks to take the decision. After checking the legitimacy, if the website is phishing, then the system shows the warning to the user. Moreover, if the website is legitimate, then the system updates the domain in the white-list.

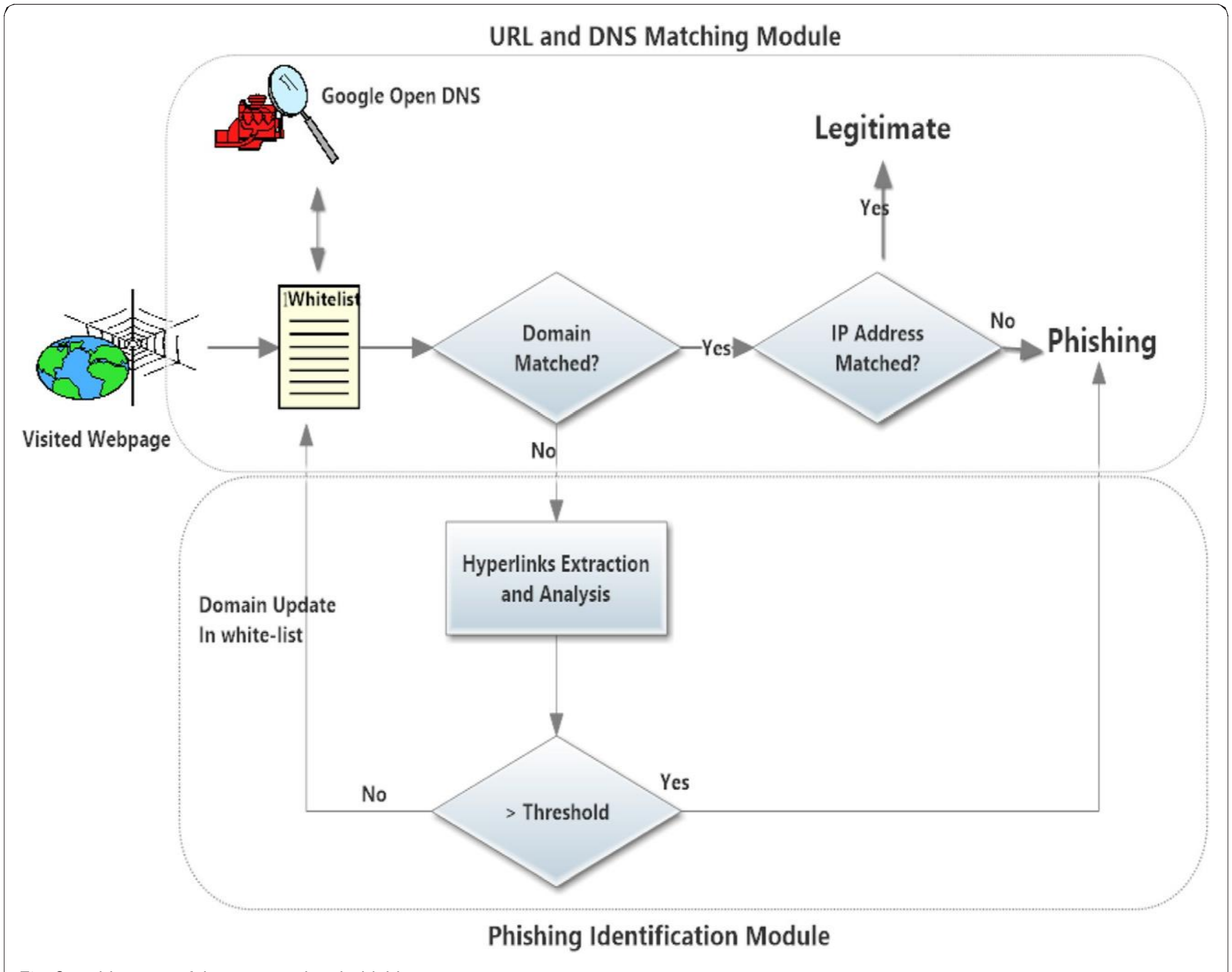

URL and DNS Matching Module
Google Open DNS
Legitimate
Yes
Whitelist
Domain Matched?
Yes
IP Address Matched?
No
Phishing
Visited Webpage
No
Hyperlinks Extraction and Analysis
Domain Update In white-list
No
> Threshold
Yes
Phishing Identification Module

Fig. 3 Architecture of the proposed anti-phishing system

Figure 3: Architecture of Antiphish application

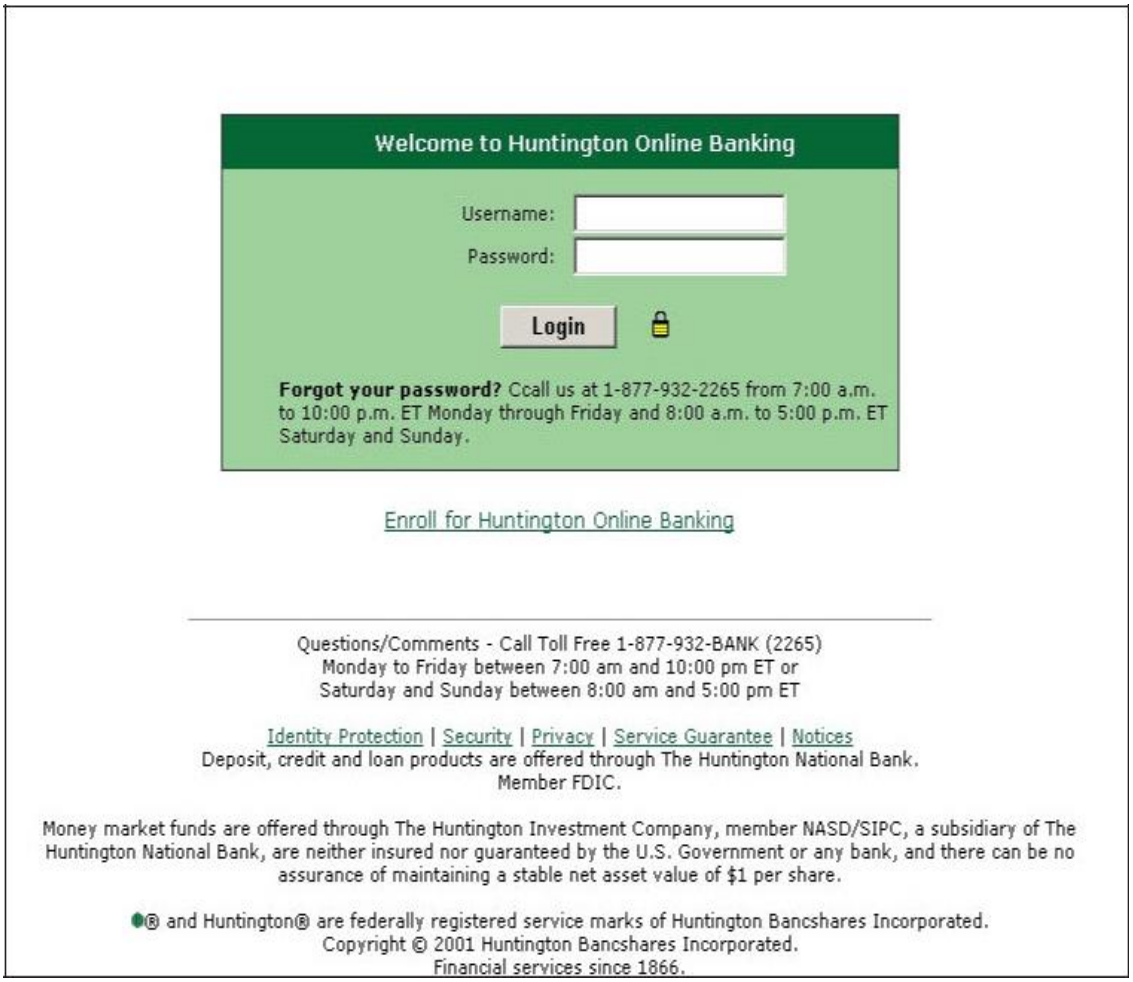
CHAPTER 5: IMPLEMENTATION

5.1 MAIN FUNCTIONALITY

AntiPhish is an application that is integrated into the web browser. It keeps track of a user's sensitive information (e.g., a password) and prevents this information from being passed to a web site that is not considered "trusted". The development of AntiPhish was inspired by automated form-filler applications. Most browsers such as Mozilla or the Internet Explorer have integrated functionality that allows form contents to be stored and automatically inserted if the user desires. This content is protected by a master password.We implemented the prototype of AntiPhish as a Mozilla browser extension (i.e., plug-in). Mozilla browser extensions are written using the Mozilla XML User-Interface language (XUL) and Javascript.

The Mozilla implementation of AntiPhish has a small footprint and consists of about 900 lines of Javascript code and 200 lines of XUL user interface code. We used Paul Tero's Javascript DES implementation for safely storing the sensitive information.Once this password is entered by the user, a login form that has previously been saved, for example, will automatically be filled by the browser whenever it is accessed. Antiphish takes this common functionality one step further and tracks where this information is sent. Figure 3 shows the right-click pop-up menu in the browser with the integrated AntiPhish menu items.

After AntiPhish is installed, the browser prompts a request for a new master password when the user enters input into a form for the first time. After this password is entered, the AntiPhish menu can be used to capture and store sensitive information. The master password is used to encrypt the sensitive information before it is stored. The symmetric DES algorithm is used for the encryption and decryption.



Welcome to Huntington Online Banking

Username:

Password:

Login

Forgot your password? Ccall us at 1-877-932-2265 from 7:00 a.m. to 10:00 p.m. ET Monday through Friday and 8:00 a.m. to 5:00 p.m. ET Saturday and Sunday.

Enroll for Huntington Online Banking

Questions/Comments - Call Toll Free 1-877-932-BANK (2265)
Monday to Friday between 7:00 am and 10:00 pm ET or
Saturday and Sunday between 8:00 am and 5:00 pm ET

Identity Protection | Security | Privacy | Service Guarantee | Notices
Deposit, credit and loan products are offered through The Huntington National Bank.
Member FDIC.

Money market funds are offered through The Huntington Investment Company, member NASD/SIPC, a subsidiary of The Huntington National Bank, are neither insured nor guaranteed by the U.S. Government or any bank, and there can be no assurance of maintaining a stable net asset value of $1 per share.

●® and Huntington® are federally registered service marks of Huntington Bancshares Incorporated.
Copyright © 2001 Huntington Bancshares Incorporated.
Financial services since 1866.

Figure 4. Screenshot of the spoofed Hunting online banking page. The login screen closely resembles the legitimate login page [9].

In our current implementation, user interaction is needed to tell AntiPhish that a piece of information on a page is important and that it should be protected against phishing attempts. After the user enters sensitive information such as a password, the AntiPhish menu is used to scan the page and to capture and store this information. Currently, the contents of all HTML text field elements of type password are captured and cached.

Besides storing the sensitive information, An-tiPhish also stores a mapping of where this in-formation “belongs” to. That is, the domain of the web site where this information was orignally entered is also stored. We use domains in-stead of web site addresses because some web sites are hosted on multiple servers with different ad-dresses (e.g., the main web site might have the ad-dress *www.ba-ca.com* and based on load, the online banking service might be hosted on *online1.ba-ca.com* and *online2.ba-ca.com*). Hence, if web server addresses or URLs are used instead of domains, false phishing alarms could be generated. In our prototype, we provide simple dialogs for the management of stored sensitive information. The user can see a list of web site domains from which sensitive information has been captured and has the possibility of clearing this cached informa-tion

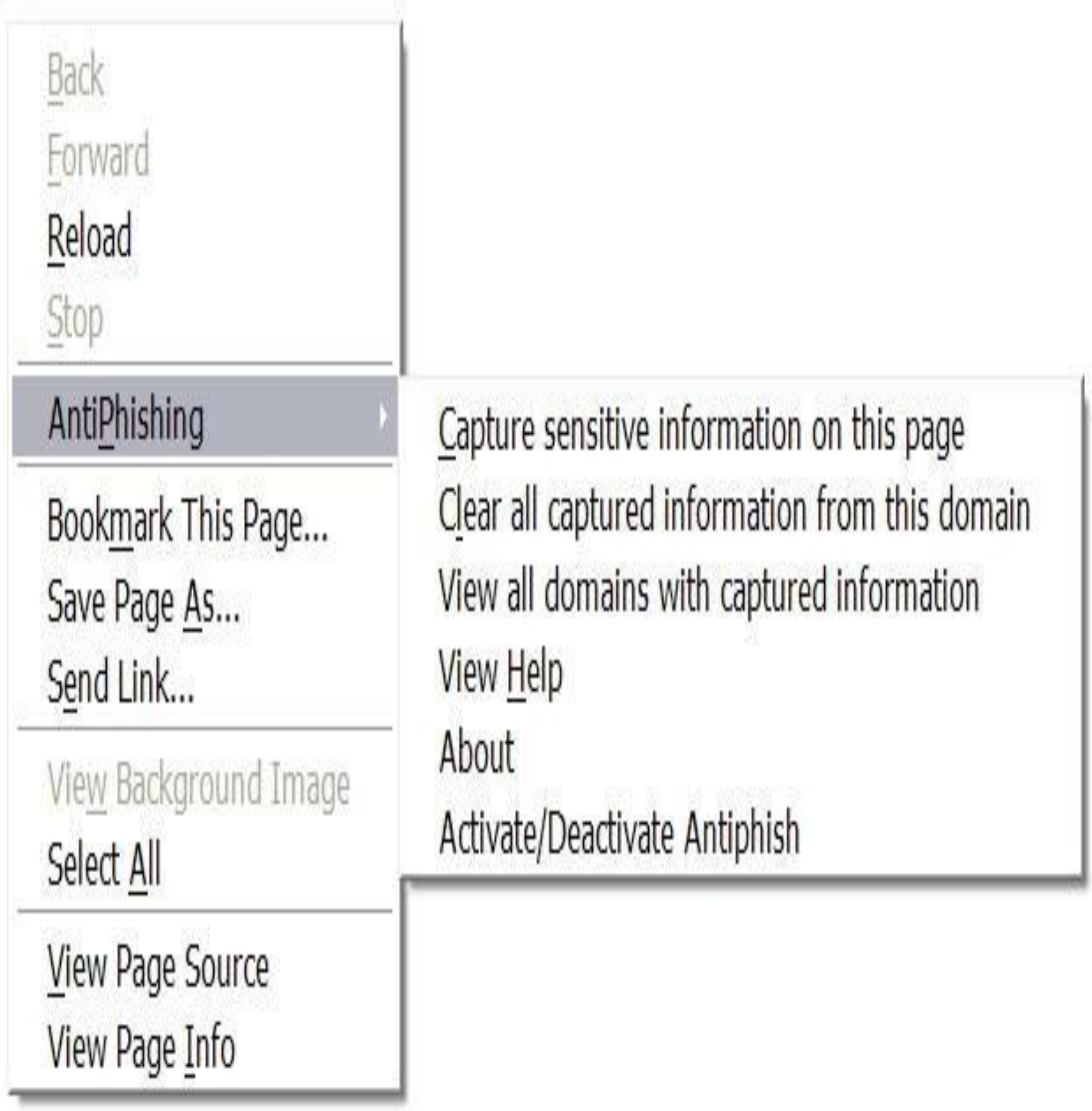
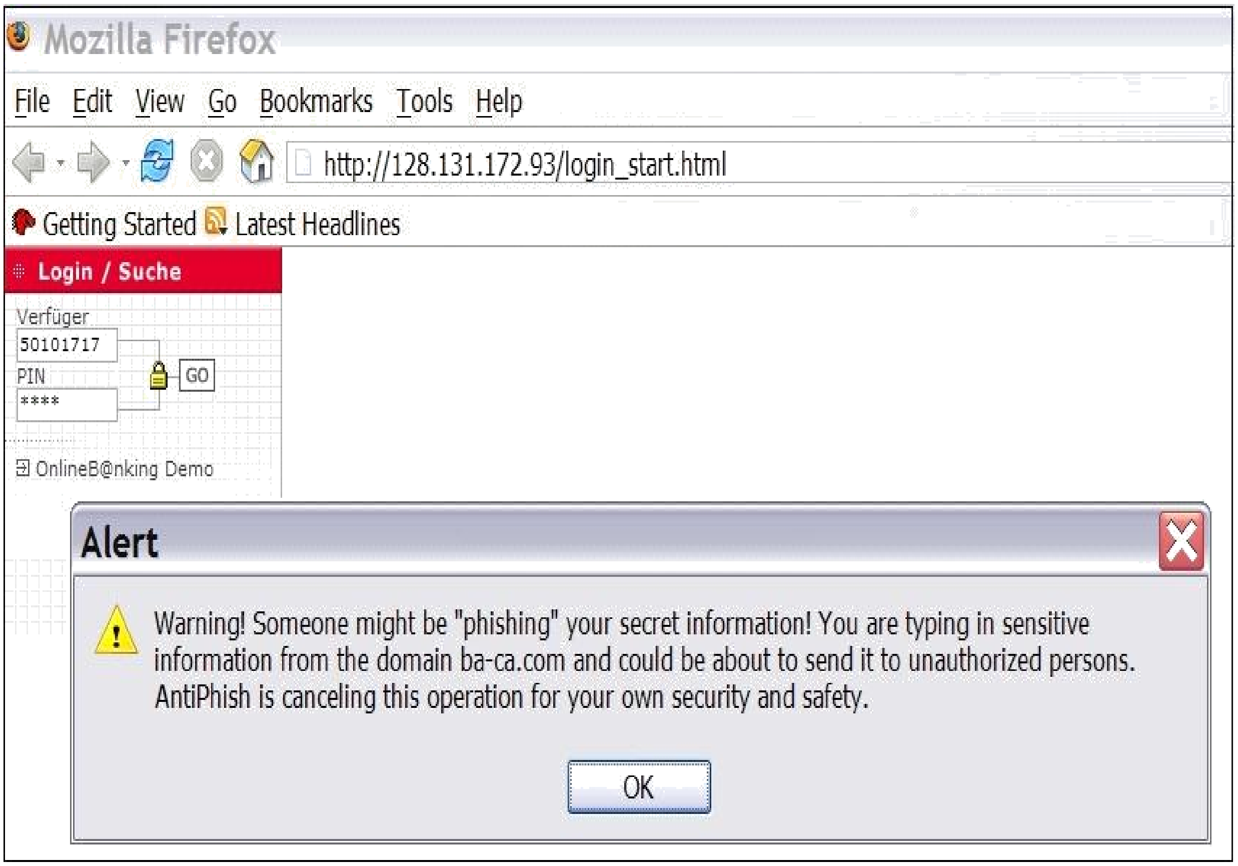

Back
Forward
Reload
Stop
AntiPhishing
Bookmark This Page...
Save Page As...
Send Link...
View Background Image
Select All
View Page Source
View Page Info
Capture sensitive information on this page
Clear all captured information from this domain
View all domains with captured information
View Help
About
Activate/Deactivate Antiphish

Figure 5. The AntiPhish application menu in-tegrated into the browser.

5.2 CONTROLLING THE SENSITIVE INFORMATION FLOW

As far as AntiPhish is concerned, every page that contains a form is a potential phishing page. HTML form elements that can be used by the at-tacker to phish information from the user are text field elements of type *text* and *password* and the HTML text area element. Hence, whenever the user enters information into any of these form elements (e.g., the user presses a key or pastes text), AntiPhish checks the list of previously captured values. For each value in this list that is *identical* to the one just entered by the user, the corre-sponding domain is determined.



http://128.131.172.93/login_start.html

Verfüger: 50101717

PIN: ****

Warning! Someone might be "phishing" your secret information! You are typing in sensitive information from the domain ba-ca.com and could be about to send it to unauthorized persons. AntiPhish is canceling this operation for your own security and safety.

Figure 6. The phishing alert message box

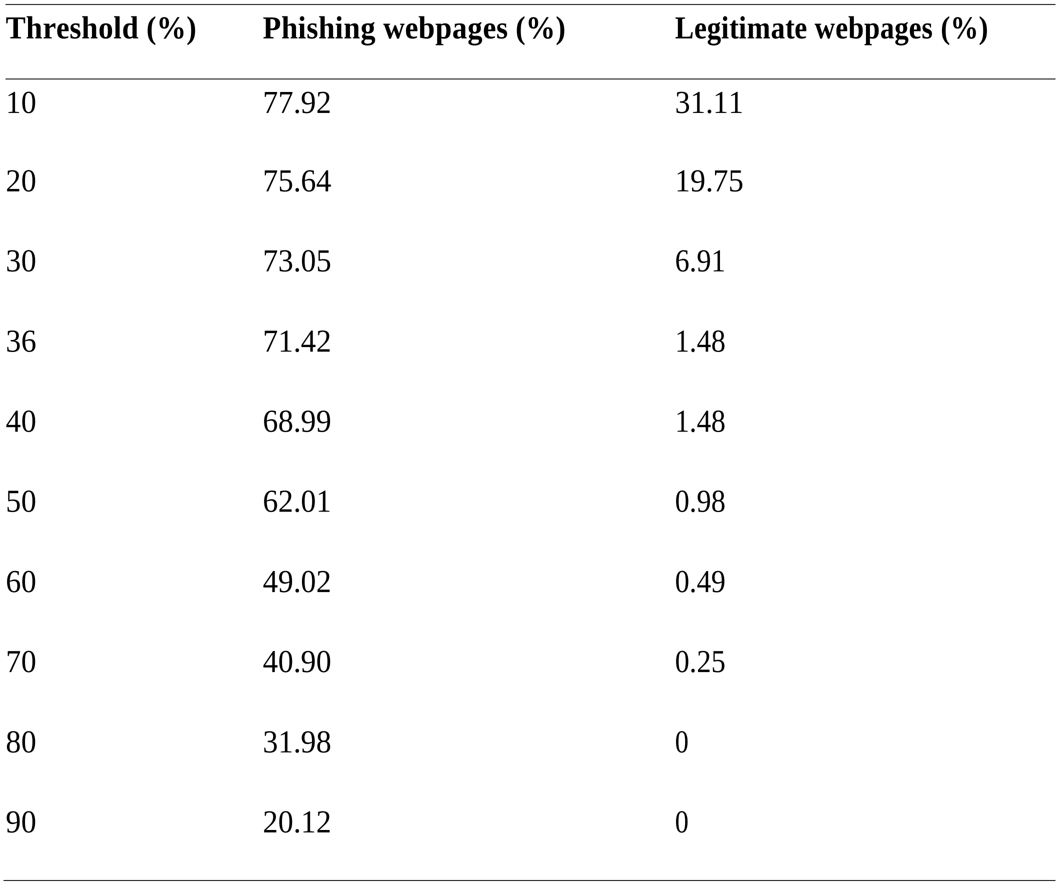
If the current site is not among these domains, a phishing attempt is assumed. The reason is that sensitive information is about to be transmitted to a site that is not explicitly listed as trusted. If AntiPhish detects, for example, that the user has typed his online banking password into a text field on a web site that is not in the online banking web site domain (i.e., an “untrusted” web site), then it generates an alert and redirects to an information page about phishing attacks. Our white-list maintains two parameters, domain name and corresponding IP address. Whenever a user accesses a website, then the system matches the domain name of the current website with white-list. If the domain of the current website is matched with the white-list, then the system matches the IP address to take the decision. When the user access a website which is already present in the white-list, then our system matches the IP address of the corresponding domain to check the DNS poisoning attack.

CHAPTER 6 :ANALYSIS

- Our experimental results show that the proposed approach is very effective for protecting against phishing attacks as it has 86.02 % true positive rate while less than 1.48 % false negative rate.
- Moreover, our proposed system is efficient to detect various other types of phishing attacks (i.e., Domain Name System (DNS) poisoning, embedded objects, zero-hour attack). Our proposed approach has both fast access time and high detection rate.
- When users try to open a website which is not available in the white-list, the browser warns users not to disclose their sensitive information. Furthermore, our approach checks the legitimacy of a webpage using hyperlink features.
- For this, hyperlinks from the source code of a webpage are extracted and apply to the proposed phishing detection algorithm.

S. number	Database	Number of URLs	Phishing/legitimate	URL of dataset
1	PhishTank [27]	1120	Phishing	https://www.phishtank.com
2	Alexa [31]	200	Legitimate	http://www.alexa.com/topsites
3	Stuffgate [32]	150	Legitimate	http://stuffgate.com/stuff/website/top-sites
5	Online payment service provider	55	Legitimate	http://en.wikipedia.org/wiki/List_of_online_ payment_service_providers

Figure 7: Table 1 Database uses to test system



Threshold (%)	Phishing webpages (%)	Legitimate webpages (%)
10	77.92	31.11
20	75.64	19.75
30	73.05	6.91
36	71.42	1.48
40	68.99	1.48
50	62.01	0.98
60	49.02	0.49
70	40.90	0.25
80	31.98	0
90	20.12	0

Figure 8: Ratio of hyperlinks pointing to a foreign domain versus total hyperlinks

CHAPTER 7 :SUGGESTION FOR FURTHER STUDY

- We are planning to implement a version of An-tiPhish for the Internet Explorer (IE) browser. Supporting IE is important because a large ma-jority of Internet users are using this browser.Antifish is free for public use .We are also planning to officially register the project with the Mozilla extensions web site .
- As we know, AntiPhish currently needs user support to capture and store sensitive information. For some users, it might be better to provide a mode where sensitive information is automatically captured and stored.
- This could be done by capturing and caching the information every time information is entered and submitted to a web site. In order to implement this functionality, submission events also need to be intercepted.
- Another issue is that the data set used by some typical anti-phishing solutions and limited for specific languages like English. So that phishers can defeat some antiphishing solutions by phish websites hosted in some other languages like Arabic and Chinese.
- The security and prevent from phishes attacks are necessary in development and growth in different fields such as financial banks, industries, transportation, and new technologies, etc. As a result the main challenge of new researchers is to conduct investigations toward finding an optimum anti-phishing solution in terms of detection capability against novel phishes along with efficacy factors for wider-scale detection of existing ant-phishing solutions. For future our suggestion is that an optimum anti-phishing solution, which is based on a combination of antiphishing approaches require.

CHAPTER 8:CONCLUSION

- To our knowledge, only two academic phishing solutions have been presented to date and both solutions have limitations. Several companies such as AOL have announced plans to provide some phishing support with their browsers.
- Most proposed phishing solutions are based on the crawling of web sites to identify "clones" and the maintenance of black lists of phishing web sites. Such solutions, however, require the antiphishing organizations to be much faster than the attackers.AntiPhish that aims to protect users against spoofed web site-based phishing attacks.
- AntiPhish tracks the sensitive information of a user and generates warnings whenever the user attempts to transmit this information to a web site that is considered untrusted. As the number of phishing scams continues to grow and the costs of the resulting damages increases, we believe that AntiPhish is a step in the right direction and a useful contribution for protecting users against spoofed web site-based phishing attacks.

CHAPTER 9 :REFERENCES

9.1 BIBLIOGRAPHY

- A Almomani, BB Gupta, S Atawneh, A Meulenberg, E ALmomani, A survey of phishing email filtering techniques. IEEE Commun. Surv. Tutorials 15(4), 2070–2090 (2013).
- A Mishra, BB Gupta, Hybrid solution to detect and filter zero-day phishing attacks, in *proceeding of Emerging Research in Computing, Information, Communication and Applications (ERCICA-14)*, Bangalore, India, August 2014
- A Tewari, AK Jain, and BB Gupta, Recent survey of various defense mechanisms against phishing attacks. J. Inf. Privacy Sec. 1-11. 12(1), 3–13 (2016)
- S Sheng, B Magnien, P Kumaraguru, A Acquisti, LF Cranor, J Hong, and E Nunge, Anti-Phishing Phil: the design and evaluation of a game that teaches people not to fall for phish, in *Proceedings of the 3rd symposium on Usable privacy and security*, July 18-20, Pittsburgh, Pennsylvania, 2007 pp. 88-99
- A Almomani, BB Gupta, T Wan, A Altaher, Phishing Dynamic Evolving Neural Fuzzy Framework for Online Detection Zero-Day Phishing Email. Indian J. Sci. Technol. 6, no. 1, 3960–3964 (2013)
- G Xiang, J Hong, C Rose, L Cranor, Cantina+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans Inf Syst Secur (TISSEC) 14(2), Article no. 21 (2011)
- S Sheng, B Wardman, G Warner, L Cranor, J Hong, and C Zhang, An empirical analysis of phishing black-lists, in Proceeding of the Sixth Conference on Email and Anti-Spam, CEAS, 2009.
- A Almomani, BB Gupta, T Wan, A Altaher, Phishing Dynamic Evolving Neural Fuzzy Framework for Online Detection Zero-Day Phishing Email. Indian J. Sci. Technol. 6, no. 1, 3960–3964 (2013).
- . M Moghimi, AY Varjani, New rule-based phishing detection method. Expert Syst. Appl. 53, 231–242 (2016)
- R Gowtham, I Krishnamurthi, A comprehensive and efficacious architecture for detecting phishing webpages. Comput. Secur. 40, 23–37 (2014).
- . GA Montazer, S Yarmohammadi, Detection of phishing attacks in Iranian e-banking using a fuzzy–rough hybrid system. Appl. Soft Comput. 35, 482–492 (2015)

9.2 WEBLIOGRAPGY

- http://www.fbi.gov/pressrel/pressrel03/ spoofing072103.htm.
- http://www.heise. de/newsticker/meldung/52935.
- http://stuffgate.com/stuff/website/ top-1000-sites.
- https://developers.google.com/speed/ public-dns/.
- http://jsoup.org/apidocs/org/jsoup/parser/ Parser.html.
- https://www.phishtank.com.
- :http://jsoup.org/apidocs/org/jsoup/parser/