The University of Sheffield
International Faculty

COMPUTER SCIENCE DEPARTMENT

Real-time Detection of Phishing Websites

This report is submitted in partial fulfillment for the degree of Master of Science in Advanced Software Engineering, by

Dimitar Goshevski

13.01.2018

Supervisor

Dr. George Eleftherakis

Real-time Detection of Phishing Websites

by Dimitar Goshevski

Supervisor: Dr. George Eleftherakis

Abstract

Phishing is a pervasive online security threat in which an attacker known as phisher tries to lure legitimate users to disclose confidential information, by mimicking electronic communication, from trustworthy organizations and companies. Fighting against phishing attacks is not a straightforward mission. Many different anti-phishing strategies have been proposed by the research community and the industry, but not all of them produced promising results. By following up-to-date anti-phishing literature and examining state of the art anti-phishing practice, this projects aims to tackle phishing scam using an alternative approach. This project implements an automated real-time anti-phishing platform, named PhishEduPro, which is capable to detect website based phishing attacks. The platform operates online and is able to extract all relevant information from suspected web pages, analyze it with state of the art techniques, generate relevant features from each technique, and store those features for each of the analyzed web pages. Each feature, used in the website evaluation process, is tagged and labeled with an appropriate color, based on the score obtained from a newly developed anti-phishing algorithm. Therefore**,** those features could be utilized to raise user awareness about phishing and could help in educating and training users to recognize and mitigate future phishing attacks. Finally, the performance of PhishEduPro platform has been evaluated in a real life scenario, by utilizing a dataset of phishing and legitimate web pages. The results obtained from the evaluation process are considered as very promising.

Declaration

All sentences or passages quoted in this dissertation from other

cross-referencing to author, work and page(s). I understand that failure to do these amounts to plagiarism and will be considered grounds for failure in this dissertation and the degree examination as a whole.

Name: DIMITAR GOSHEVSKI

Signed . . . . **Date**: 13.01.2018

ACKNOWLEDGMENT

I would first like to thank my supervisor Dr. George Eleftherakis for his unconditional support, valuable advice and encouragement during this challenging period. I have been extremely fortunate to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. The door to Prof. Eleftherakis office was always open, whenever I ran into a trouble spot or had a question about my research or writing. If I have been asked to repeat this journey once again, I would definitely choose him to guide me on the road towards success.

I would also like to thank the people from WhoisXmlApi.com, the best commercial WHOIS lookup service in the world. Without their goodwill to provide me with a free unlimited access to their WHOIS lookup service, this Master Thesis could not have been successfully conducted.

I must express my gratitude to Joana, for her endless support and encouragement. I was constantly amazed by her willingness to proof read countless pages of meaningless design patterns and anti-phishing algorithm workflows. She fully supported me all the way through the end. Finally, I would like to acknowledge the patience of my mother, father, brother and grandmother who experienced all of the ups and downs of my research. Thank you for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

Table of Contents

Chapter 1

1. Introduction

Human to human interactions are based on trust. Trust is a precondition when two or more individuals have to interact in their personal and professional life. The same rule applies in offline business to business interactions or in company to customer relationships. Customers must have trust in companies in order to buy and use their products. Accordingly, companies must build trust among their business partners in order to grow and be competitive on the free market. With the emergence of Internet, especially in the last 15 years, the concept of trust became a crucial element in online interactions. In year 2003, L.Corritore at al. presented a study where they defined online trust as a model that describes the interactions between end users and online mediums [1]. In their paper they identified that online decisions are usually trust sensitive because they are affected by one or more factors like: perception of credibility, ease of use of online services and the risk of personal or financial lost [1]. The biggest problem in online transactions among engaged entities is the problem of transparency and incomplete information sharing. For instance the receiver of an email knows much less about the true intention of the sender, the [2]. Similarly, social media users or web site visitors know little or nothing about how their personal data which is asked upon registration will be used by the owners of those websites.

Information security became a huge problem at online world. The number of online security breaches on companies and individuals is increasing exponentially day by day. Bruce Schneier in his blog post from 15th October 2000 identified and explained 3 waves of security attacks that internet community faces nowadays. The first wave of attacks is physical and it is targeted against computes, power supplies and electronics [3]. However, the online communities efficiently fought back against those attacks. Distributed protocols and system redundancy architectures were invented and single point of system failure became a past. Second wave of attacks is targeted against computer logic and networks and are known as syntactic attacks [3]. Vulnerabilities in software systems, cryptographic algorithms and protocols as well as denial of service vulnerabilities were the main targets of internet crackers. Such attacks are usually performed by installing malicious software (malwares) on corrupted machines so that the intruders could take control upon them. Those types of attacks are hard to detect on time, but a number of countermeasures exists to fight back against them. Real time network monitoring, digital signatures, antivirus programs and honeypots are part of the solution for this problem. Third and the last wave of attacks are known as semantic attacks and they are targeted against humans. People are the weakest link in cyber security chain, and semantic attacks are targeting the way how people assign meaning to content, interact with computers or interpret online messages [3]. Those attacks are much more devastating compared to physical and syntactic attacks because they could affect everybody: individuals, companies, big corporation and even governments.

Usually people who are victims of semantic attacks are not aware of the attack until they felt the consequences from it such as: personal information theft or financial loss.

Social engineering is a broad term that is used to describe a set of semantic attacks. Intruders often use their social engineering skills to break into computer systems and corporation networks, by psychologically manipulating and deceiving users to disclose to them some confidential information [4]. Common examples of social engineering attacks are: phishing emails, spoofing websites, obfuscated URLs, drive-by-download, ransomware attacks etc. Phishing is the most popular form of social engineering in which an attacker known as phisher tries to lure legitimate users to disclose confidential information, by mimicking electronic communication, usually emails, social network messages and websites from trustworthy organizations and companies [5]. Fighting against phishing is a very challenging process, because it affects the cognitive nature of human beings. In general there are two types of phishing detection approaches: user education and training and automatic phishing detection by using software classification [6]. User education and training is considered as a preventive approach where end users are educated and trained to distinguish potential phishing emails and fraudulent websites from legitimate one. This approach has some drawbacks because it is very expensive therefore it is not applicable to large user base i.e. PayPal and Amazon [6]. On the other hand, automated phishing detection software tries to eliminate the user error in phishing detection. However, warnings which are produced by anti-phishing browser plugins are often ignored by end users. Some information security researchers [2] strongly belief that both approaches are complementary one to another, but some of them like Stefan Gorling (a professor and researcher at KTH Royal Institute of Technology in Stockholm), do not agree with this statement [7]. Having all this in mind, security researchers come up with new and better anti-phishing strategies every day, but a silver built to fight against phishing scam is not yet discovered.

1.2 Motivation

Phishing scam is number one online security threat in the world. Since 2003 there is a dramatic increase in number of phishing attacks. According to the statistics from Anti-Phishing Working Group (APWG) in November 2003 only 21 unique phishing attacks were reported [8]. On the other hand the last report from APWG published on February 22nd 2017 show that in April 2016 a total of 158,988 unique phishing websites were reported [9]. From these statistics we could see that phishing is one of the hardest problems that cyber security is facing nowadays. Developing countermeasures to fight against phishing is a very challenging task because those attacks exploit weaknesses in human computer interactions. People are easily tricked by fraudulent websites and spoofed emails to disclose their personal and financial information because they look very legitimate.

There are different types of techniques which are developed by researchers to fight against phishing. Some of the techniques like embedded email training, game based learning and sending security notices via email are developed to increase user awareness about phishing by educating and training users to recognize and mitigate phishing attacks [2][10]. A study [11] showed that even when the users are trained with the best phishing awareness program they failed to identified 28% of the phishing attacks. On the other hand, automatic phishing detection is considered to be much more effective to fight against phishing compared to user training and education. Many automated anti-phishing solutions reviewed in [12] could easily achieve accuracy rate over 90% with minimal false positives. However, those approaches totally isolate users from having an insight in the decision making process. Automated phishing detection techniques such as: blacklists, heuristic rules, website visual similarity techniques and machine learning techniques show only if a specific website or email is phishing or legitimate without explaining to end users the decision steps and anti-phishing features which are used in the classification. This could have negative impact to end users and users might start to ignore the warnings produced by those tools because they will not completely understand the risk of phishing. Additionally, many of the proposed automated solution are only designed as proof of concept and are never implemented in practice as real-time anti-phishing tools. This is due to their limitations in performance, high implementational complexity and lack of resources, like fresh phishing datasets, especially for training and testing machine learning anti-phishing algorithms [13]. To our best knowledge, a study about a tool which will perform a real-time detection of phishing websites and will use calculated phishing statistics and features extracted from those websites to educate users why specific websites are classified as phishing or legitimate is not yet conducted.

1.3 Aim of the project

Build a real-time phishing detection tool able to extract, analyze and store relevant features from web pages, to effectively and efficiently detect, mitigate and report potential phishing attacks to end-users.

1.4 Objectives

1.4.1 Project Objectives

- **Perform a comprehensive research on phishing attacks and defense techniques** by reviewing the existing literature. The main research interest of this project is to evaluate tools and algorithms that are used to protect individuals and companies from phishing attacks.

- **Propose a design and system architecture for real time online phishing detection platform** , with main goal to analyze, detect and classify websites as phishing or legitimate.

- **Define any assumptions and limitation of the proposed platform.** Here should be identified all of the assumptions under which the proposed platform is supposed to work. Moreover, if there are any limitations regarding the platform they should be clearly stated.

- **Choose a software development process.** Since the aim of the project is to develop a working application, a suitable software development process must be followed. The platform development process should support all phases of software development lifecycle, but should not be limited to: planning, project management, risk management, analysis, design, implementation, testing and deployment. A detailed plan of the development process should be scheduled in respect to the final submission date of the project.

- **Evaluate project risks and propose a suitable risk prioritization and mitigation strategy**. A risk management matrix should be developed in the early stage of the project evolution to record risks associated with the project and to estimate their probability of occurrence.

- **Choose a development and deployment environment.** A web development programming language, external APIs services, third party libraries, suitable database management system, security model, version control system,

system,

techniques, generate relevant features from each technique, and store those features for each of the analyzed web pages.

- **Provide free RESTful API access to data and functions of** PhishEduPro platform -phishing algorithm to users, companies and organization. Thus, they could utilize it as an anti-phishing detection mechanism to improve security of their proprietary tools and services or it could be utilize as a complementary tool for enhanced anti-phishing education.

- **Improve the accuracy rate and minimize false positives.** To rank PhishEduPro platform among the best anti-phishing solutions available on the market, an accuracy rate over 95% should be achieved. In addition, misclassification of legitimate websites must be less than 3%. Therefore, PhishEduPro platform will employ a sophisticated algorithm that will aim to improve the effectiveness and at the same time the efficiency of identification of phishing websites.

- **Tag and label each anti-phishing feature of a suspected website** to help educating and training users to recognize and mitigate future phishing attacks. Generate and display detailed phishing statistics for each suspected website to raise user awareness about phishing. Each feature, used in the website classification process, should be tagged appropriately based on the score obtained from the anti-phishing algorithm. Computed features will be used to justify the final decision and potentially be used for continuous user education.

- **Enable easy tuning of anti-phishing rules.** PhishEduPro platform should be configurable enough to provide an easy to use interface to adjust the rules utilized by the anti-phishing algorithm. Authorized platform users should be able to change the threshold for each anti-phishing rule when new trend in phishing emerges to improve the accuracy rate of the algorithm.

- **Provide users with frequently updated dataset** of phishing and legitimate websites suitable to train machine learning algorithms at no cost**.**

1.5 Report Structure

Chapter 1 (Introduction) - The Introduction chapter talks about the concept of online trust and cyber security in general. Then it gradually starts to clarify to the reader the meaning of phishing and presents a brief history of phishing attacks and defense. The motivation behind this project is clearly presented in this chapter. The aim of the project and its detailed objectives are discussed as well.

Chapter 2 (Literature Review) - Literature Review chapter starts with sections talking about the concepts of Social Engineering and Phishing in details. Furthermore, the taxonomy of phishing attacks and defense is presented to the reader. A section explaining about anti-phishing evaluation metrics is also part of this chapter. Finally, a comparison table of similar anti-phishing tools and approaches followed by a summary section are presented to add value to the literature review.

Chapter 3 (System Requirements and Analysis) - The main focus of this chapter is to elaborate an idea about a real-time anti-phishing tool which will be capable to detect zero-day phishing attacks using automated routines. Furthermore, this chapter presents the main goal of the system, followed by functional requirements and use case diagram of PhishEduPro platform. The initial workflow of the PhishEduPro algorithm is illustrated using a flowchart diagram and non-functional requirements are also discussed.

Chapter 4 (Project Management) - Is dedicated to project management plan for the development of PhishEduPro platform. It starts with elaborating the Software Development Process and the Risk Management Plan used to guide the implementation of the final product. The project management tool and the development environment are also discussed and justifications for each decision step are presented to the reader.

Chapter 5 (System Architecture and Design) - The high-level system architecture followed by the detailed design for the RESTful API web services of PhishEduPro platform is presented to the reader. A detailed design of the anti-phishing algorithm and the design of each of the 22 anti-phishing features and rules are elaborated here. Various design diagrams showing different aspects of the system are illustrated and explained in details. Finally, the strategy for calculating of the final phishing score of any suspected web page is presented in this chapter.

Chapter 6 (Implementation, Testing and Deployment) – The main concepts discussed in this chapter are related to implementation details, testing techniques and deployment environment of PhishEduPro platform. The implementation of RESTful API endpoints is elaborated in details. External web services and third party libraries used to facilitate the implementation of PhishEduPro platform are presented to the reader. A review of the Graphical User Interface is illustrated and explained, while unit and integration tests are developed to verify and validate the design and implementation of PhishEduPro platform. Lastly, the decision choices regarding system deployment are properly justified.

Chapter 7 (Evaluation) - A real life experiment has been conducted to evaluate the performance of PhishEduPro anti-phishing algorithm. The evaluation methodology and strategy is presented in this chapter. Different evaluation metrics such as: accuracy rate, false positive rate, false negative rate etc. are calculated and those metrics are compared to evaluation metrics obtained from similar anti-phishing approaches and tools. Finally, the limitations of PhishEduPro anti-phishing approach are presented to the reader.

Chapter 8 (Conclusion) – This is the closing chapter of the Master Thesis project. It is focused on elaborating ideas regarding future work and presents a final conclusion on this topic.

Chapter 2

2. Literature Review

2.1 Social Engineering

Online community is spending a tremendous effort and resources to facilitate trustworthy online communications among participants. Many tools and technologies have been developed to secure those online interactions. The famous hacker Kevin Mitnick in his book ould spend a fortune on securing its valuable assets using technology, but their network could still remain vulnerable to simple social engineering attacks [14]. Social engineering is a technique used by attackers to deceive people that they are trustworthy interlocutors, by manipulating people s tendency to trust, in order to obtain valuable information about the subject of interest with or without the use of technology [11, 12]. To conduct a successful social engineering attack, social engineers use different approaches such as: phone calls, dumpster diving, phishing emails, spoofed websites, persuasions via impersonation and many other techniques [16]. Gaining unauthorized access to a system to obtain confidential information about the target by using fraud, network intrusion, identity theft, and/or industrial espionage is the main goal of social engineers.

Social engineering attacks are used to be very successful, because they affect the psychological aspects of human behavior. A simple yet powerful definition for social engineering is: The ability to hack into the human brain [17]. A technological expertise is not crucial for conducting a social engineering attack. Usually social engineers practice their communication skills and study about body language, voice control and corporate philosophy. They try to master their understating of other people feelings by observing body language, facial expressions and wording they use. feelings and behaviors is crucial for success of social engineering attack. Gaining trust and feelings by bluffing and deception encourages people to reveal confidential information about the target.

By studying the literature and the psychological aspects of social engineering four psychological attack vectors are identified: careless attack vector, comfort zone attack vector, helpful attack vector and fear attack vector [15]. The attacks belonging to careless attack vector escalate, because people are often indifferent and careless and do not obey to neither they enforce proper defensive countermeasures. que belonging to careless attack vector [15, 12]. Collecting trash from company dumpsters could reveal much valuable information to social engineers such as: old company letters, company bills, phone numbers, important notes and calendar of meetings. This

technique could even reveal a system application usernames and passwords written on sticky notes if they are not properly shredded. Having this information the attackers knowledge about the company expands so that they can launch a more complex attack in the future. Many people and companies suffer social engineering attacks because they feel very secure inside their comfort zone. A bank employee or company secretary could easily reveal his/her login details to a person wearing a company uniform that impersonates a guy from IT bank department, who is here to do a security update on her online payment account. Attacks belonging to helpful attack vecto
help to other people even if they do not know who are helping. FBI recently reported of a new trend of social engineering attacks known as romance scam [19]. Scammers use reconnaissance to obtain valuable personal information of the victim by browsing social networks. Later, they connect with the victims (usually divorced women around 50 years old) on social media to start a romantic relationship. They tend to build a trust relationship with the victims in a short time span and ask her to transfer them money because they are in very bad financial situation. Finally, when they sucked enough money from the victim they suddenly disappear. The last but not least important attack vector is the fear attack vector. Those attacks are considered to be the most psychologically aggressive for people, because they put people in unpleasant situation where they face high stress, fear and anxiety [15]. Social engineers usually use this technique to put company employees under pressure and that way to obtain confidential information about the company. By combining techniques from previous attack vectors, an attacker might try to impersonate a legitimate employee from and will ask from the secretary to provide him the password to access her company account, since it was suddenly hacked.

A diverse set of countermeasures to protect people from social engineering attacks have been proposed by researchers. A common strategy to employ is to strengthen the security policy of a company and increase the security of its physical assets [4, 12]. The policy should be written in a way that company management and employees will easily understand it, support it and apply it unconditionally. The security policy should be updated and revised on regular basis. Since people are the weakest link in security chain a proper education and user awareness program should be targeted towards them [2]. Users and employees should be educated how to recognize report and mitigate a social engineering attacks and should be regularly rewarded when they obey the security policy.

2.2 Phishing

an uses bait (usually malicious email or social network message) to catch the fish (collect confidential information from the victim) [5][20]. According to the literature the term was first mentioned by the alt.2600 hacker newsgroup in January 1996, to refer to the password stealing attack against America Online (AOL) subscribers [20]. The substitution due to term , which was referring targeted against the telephone networks [5].

United States to investigate the recent security breach in their presidential campaign), a spear-phishing campaign chairman John Podesta, on 19th March 2016, asking from him to change his Gmail account password immediately [26][27][28]. Attackers explained in the email that this is a step of precaution, because someone from Ukraine tried to access his Gmail account but it was stopped by Google. In the body of the ema was shown to him, but the actual link URL was not pointing to Gmail password change secure page. The link was shortened by bit.ly URL shorter service and it was pointing to a fraudulent website. A print screen of the spear-phishing email sent to John Podesta can be found in Appendix A. Later in the interview Phil Burdette explained that the link sent to Podesta was clicked twice. In addition, he also confirmed that 108 email addresses which belong to people involved in were targeted by the same phishing attack and that 20 of the sent links were clicked [26]. Although the consequences of this phishing campaign are yet to be confirmed, it could be concluded that a simple social engineering attack by and their poor decision making under stress might have a devastating impact to society.

2.2.2 The Cost of Phishing

The real cost of a phishing is very hard to be estimated. Phishing is a black market activity and hackers are not willing to advertise their success from phishing campaigns. On the other hand, companies, public organizations and individuals rarely report phishing attacks because they are afraid of additional financial losses, reputation damage and lawsuits against them. According to Jakobsson and Myers [5] there are 3 types of cost that should be consider: direct, indirect and opportunity cost of phishing. Direct cost is measured by the total value of money and goods which are stolen from the victim under attack. By looking into available security incident statistics from October 2013 to December 2016 reported by FBI, direct financial loss from phishing is estimated to be a few hundred million dollars per year only in US [29]. Indirect cost of phishing is represented by the cost for handling increased customer support by companies and organizations, their need for upgrading security infrastructure and policy which occurred after the security breach, as well as the cost of emotional distress of customers when they find out that their personal or financial data might have been corrupted. The cost of phishing suffered because individuals are afraid to engage in future online business transactions with financial services and online shops is considered as opportunity cost. Gartner Inc.
and advisory company conducted a survey involving 5000 US citizens in 2005 and they found out that nearly 30% of the participant in the survey changed their online banking habits because of the increased number of phishing attacks [30]. They reported that over ¾ of the people participating in the survey reduced their usage of online banking platforms and 14% of them stop using online banking services because they feel insecure to do real-time transactions [30].

2.2.3 Phishing Statistics

Number of phishing attacks is at constant rise since 2003rd. In their phishing threat report, published on February 22nd 2017, APWG reported that the total number of phishing attacks in 2016 was 1,220,523, which is a 65% increase compared to 2015 [9]. Additionally, they also reported that in 2016 they documented more phishing attacks compared to any previous year starting from 2004, when the first phishing threat report was published. In 4th quarter of 2004 on average 1609 phishing attacks per month were recorded by APWG. On average 92,564 phishing attacks per month were reported in the 4th quarter of 2016, which is a tremendous increase of 5753% over 12 years period [9]. APWG also tracks the number of unique phishing websites and the number of unique phishing email campaigns per month. The number of unique phishing websites is determined by the unique domain portion of the submitted URLs. In practice one phishing domain might host multiple phishing pages where the path portion of the URL is auto generated for each new visitor to the phishing domain. A monthly statistics of the number of unique phishing websites detected for 2015 and 2016 is shown in Figure 1.

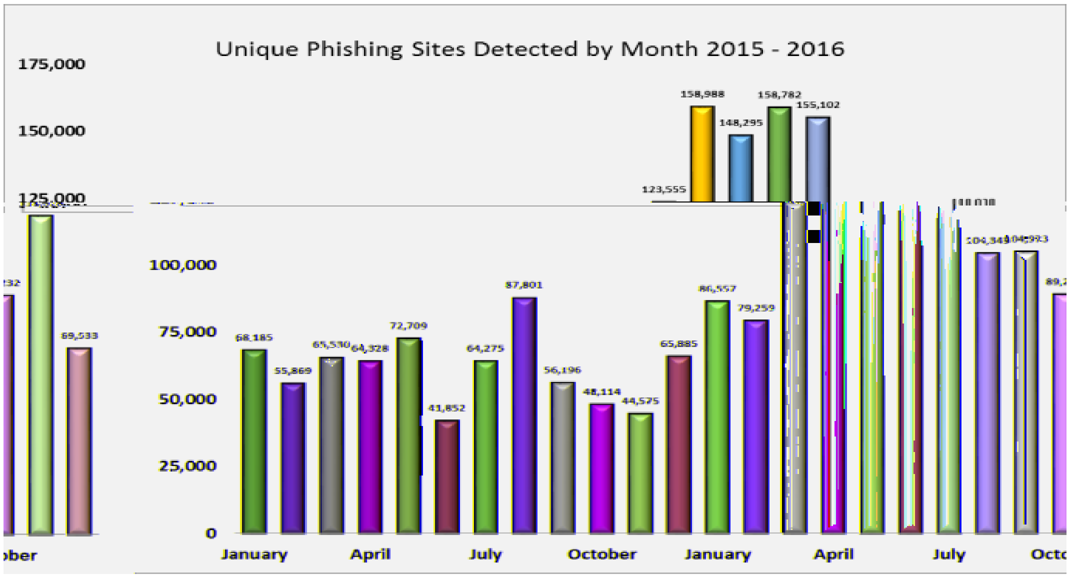


Figure 2.1: Unique phishing websites detected per month 2015-2016 [9]

The same analogy holds for calculating the number of unique phishing email campaigns hosted by attackers. A unique phishing email (an email that has the same subject line) sent to multiple users, directing them to a specific phishing website is counted as one unique phishing email campaign [9]. Multiple unique phishing email campaigns may point to the same phishing website. A total number of 211,032 unique phishing email campaigns were reported by APWG during the 4th quarter of 2016. The number of unique phishing email campaigns per month was constantly rising from October to December 2016 in the following manner: October 51,153, November 64,324 and 95,555 in December. The number of unique phishing email campaigns reported during the 4th quarter of 2015 was 380,280 [31]. To put this numbers in context it could be observed that there was a decrease in the number of unique phishing email campaigns hosted during the 4th quarter of 2016 by 55% compared to the 4th quarter of 2015. A reasonable answer to this phenomenon could be a possible change in strategy to exploit different attack channels such as social media networks, instant messaging services like Viber, WhatsApp and Slack to reach to their victims. Another interesting observation is given by Symantec in their Internet Security Threat report published in April 2017. They claim that the drop in email phishing rates is affected by the increased number of spear-phishing email campaigns where less number of users have been targeted [27]. However, spare-phishing email campaigns success rates are much higher compared to mass-mailing phishing campaigns where victims are not targeted upfront. According to Symantec statistics, phishing email rates drop from 1 in 1846 emails in 2015 to 1 in 2596 emails in 2016 [27].

In the following bullet list an interesting statistics regarding trends in phishing attacks for 2016 and 2017 are presented:

- Number of unique brands targeted by phishers in April, May and June 2017 is 460, 457 and 452 respectively [32]. Compared to the same months from 2016 on average 9% more unique brands were targeted in 2017.

- The most targeted industry sector by phishing in 3nd quarter of 2017 is finance sector with 47.54%. More specifically, Banks with 24.1%, Payment Systems with 13.94% and Online Stores with 9.49% [33]. More info about this statistic category is shown in Figure 2.

- APWG reported that in the first half of 2017, 7990 phishing incidents were reported in the world [32]. Attackers primarily used Facebook as a channel to spread the phishing scam to their victims. Half of the phishing incidents or more precisely 4026 happened in USA. Second on the list is Brazil where 1499 incidents have been reported.

- During the 3rd quarter of 2017 Kaspersky antivirus prevented 59,569,508 attempts from users to open phishing web pages [33]. Furthermore, 9.49% of the total unique Kaspersky users were under phishing attack.

- Kaspersky Lab reported that their heuristic anti-phishing tool recorded the top 3 most targeted brands by phishing in the 3rd quarter of 2017: Facebook with 7.96%, Microsoft with 7.79% and Yahoo with 4.79% of the total phishing links [33].

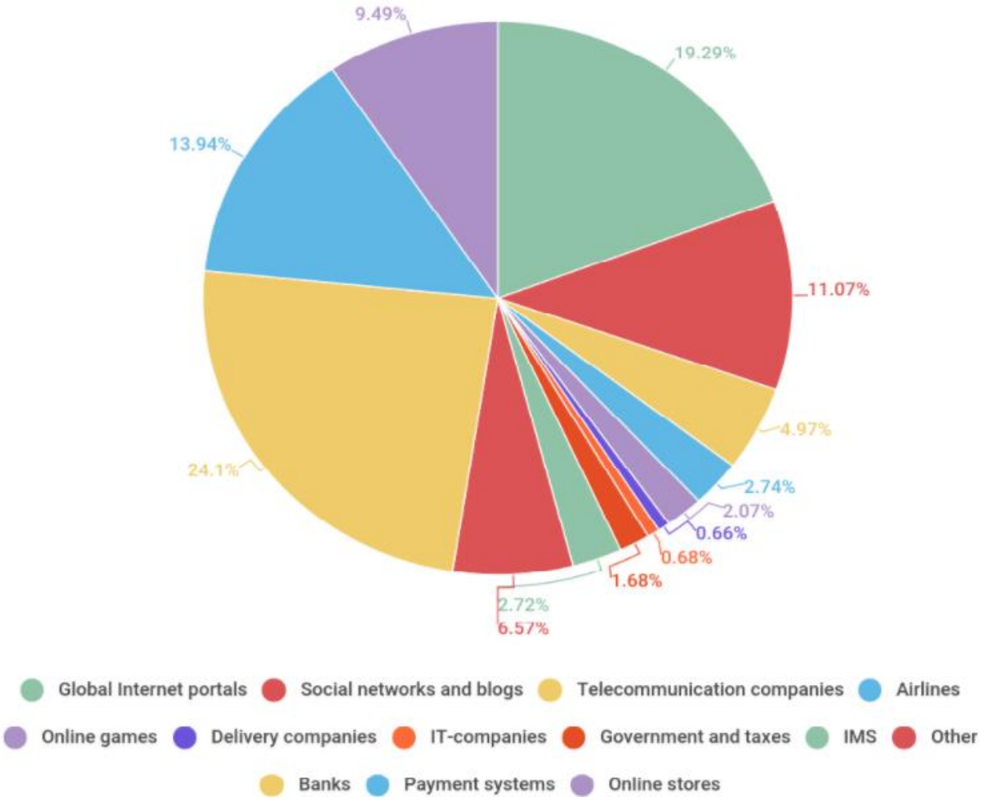


Figure 2.2: Most Targeted Industry Sectors 3nd Quarter 2017[33]

2.3 Taxonomy of Phishing Attacks

,
phishing attacks can be classified in 2 distinct categories, Deceptive Phishing and Technical Subterfuge. Deceptive Phishing is a category of phishing attacks where social engineering is playing a key role into deceiving victims, to disclose their confidential data [5]. Common attack channels used by phishers to run this type of phishing attacks are: phishing emails, social network messages, instant messages, SMS, IoT devices and spoofed websites. On the other hand, is a category of phishing attacks where technical proficiency of the phisher is crucial. Therefore,
financial information phisher uses techniques like: malicious code infiltration, user screen captures, keypad key loggers, session hijacking and DNS poisoning [18]. A basic taxonomy of phishing attacks is shown in Figure 3.

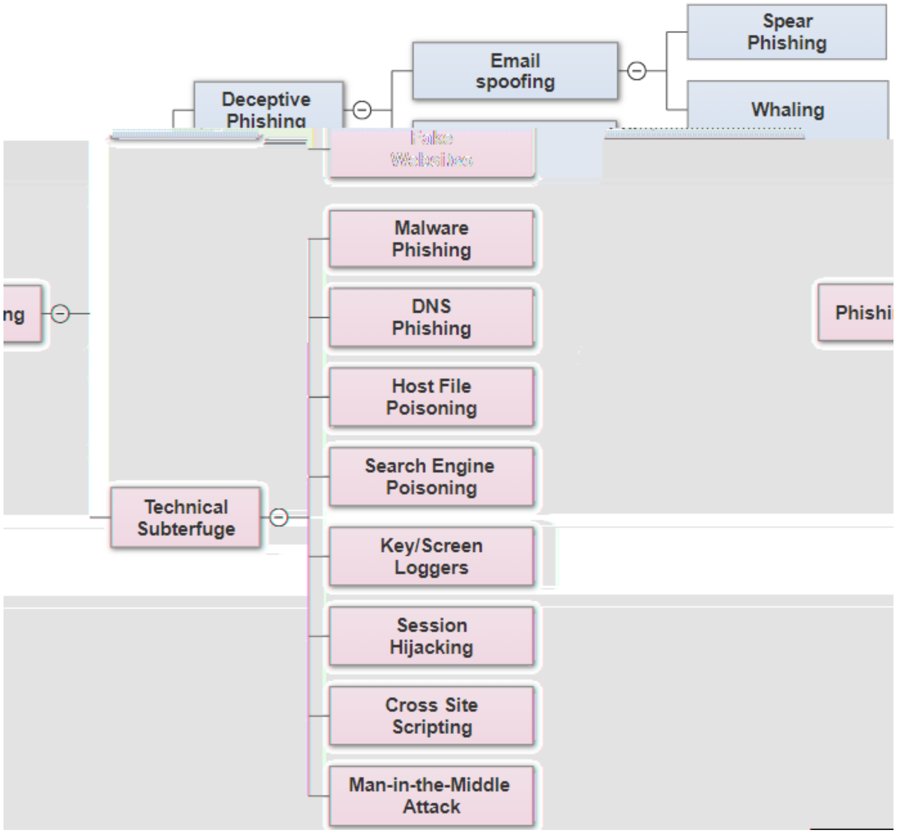


Figure 2.3:

information.

2.3.1 Deceptive Phishing

There are multiple techniques cla
further elaborated is details in the following section.

Email spoofing and **Fake Websites** frauds are strongly related. The first step in launching a deceptive phishing attack is building a fake phishing website. This fake website will try to mimic a legitimate website of a famous company or organization, and it is usually hosted on a phishing server, on a corrupted legitimate server or on a free shared hosting server. The next step of the attack is crafting a deceptive email which will be used to trick users to click on a malicious link embedded in it. This embedded link is usually obfuscated by placing misleading anchor text to lure victims to click on it, while the actual link URL is pointing to the phishing website. The text of the phishing email is crafted in a way to convince the users that the email is very important and it is coming from legitimate source. Furthermore, the phishing email will ask from victims to provide their personal or financial information to the phishing website.

In the next step of the attack the phisher launches a new phishing campaign where a large volume of phishing emails are delivered to end users. Some phishing emails will pass by the embedded phishing filters in email clients es. A large number of users will assume that this is a legitimate email coming from legitimate sender and will not take precaution steps such as: looking into email address to verify the sender. Some of these users will probably click on the malicious link embedded in the email therefore they will be redirected to the phishing website. There, they will be deceived to enter their confidential information. That information will be used by the phisher to gain an illegal financial benefit or will use them to launch a more sophisticated spear-phishing attack.

Spear phishing is a type of phishing attack which is not random and it is highly targeted toward specific employee or group of employees within an organization [34]. The aim of the phisher, launching a spear phishing attack is to acquire sensitive corporate secrets, innovation blueprints, intellectual properties or even military secrets. Spear-phishing emails are very personalized and are crafted in a way to address victims by their first and last name, rank or position. Additionally, they contain information that are very own to users such as: their hobbies and shopping habits, project that the user currently works on, information about his/her coworkers and many other personal and professional information. By conducting user reconnaissance using social media networks like Facebook, LinkedIn, Twitter and Instagram phishers could easily collect enough data about specific targets. Furthermore, the phisher camouflages the email to look like it is coming from a familiar source, usually a colleague within the same organization, which dramatically increases the chances that the email receiver will be lured to open it [12]. Spear-phishing has become very popular nowadays. According to Jagatic at al., spear-phishing success rate is 4.5 times higher than the success rate of the ordinary phishing attacks where victims are randomly selected [35]. RSA, a cybersecurity company suffered a security breach in 2011. Their forensic analysis revealed that the security incident started by opening of a spear-phishing email by one of their employees [34].

Whaling also known as CEO fraud, is nothing more than a spare-phishing attack targeted towards high-rank employees in big corporations, government or military institutions and politicians [23]. One of the latest whaling attacks happened in March 2016, when Russian hackers send a phishing email presidential election campaign chairman John Podesta asking from him to change his Gmail account password immediately [28]. More information about this attack can be found in section 2.2.1 from the literature review.

2.3.2 Technical Subterfuge

Phishing in general does not rely solely on social engineering. Phishers use other techniques like embedding malicious code in websites that could listen for input from visitors and steal their login credentials, or attach self-executable file to phishing emails that might corrupt computer, ask for ransom or steal some data from it. A diverse set of phishing methods that use technical subterfuge to obtain user confidential information in an illegitimate way are discussed in details in the sequel.

Malware Phishing
beyond his/her knowledge, and it is capable confidential information like login credentials and financial data to the phisher [36]. Malware software is usually distributes as self-executable email attachment where the phisher employ social engineering to lure victims to open the attachment on his/her machine.

DNS Phishing – In DNS Phishing attacks the phisher runs a fake DNS server and uses some technical tricks to lure the client to connect to it. At the moment when the client request a legitimate domain name it is redirected to a malicious website, because the fake DNS server will resolve the targeted domain to malicious IP address [37].

Host File Poisoning – This type of attack happen when the host file of client computer is poisoned by injecting new pairs of bogus IP addresses and legitimate domain names [38]. When the client requests a URL of a legitimate domain, before it is send over the Internet it will be first resolved to an IP address. Following the bogus IP address from the poisoned host file the client will be redirected to a phishing website and will be lured to provide his/her personal and financial information. This type of attack is also known as **Pharming**.

Search Engine Poisoning – Attackers could increase the rank of phishing websites in search engines and position them at the top of the search results by artificially building up their search index with technique known as spam indexing [12]. This could be done by inserting some keywords into the webpage that will make it to look relevant to search engines. Usually, these fraudulent websites offer a very high discount to expensive products and services, therefore end users are tempted to shop there by using their credit cards [39].

Key/Screen Loggers - There are different types of hardware and software key/screen loggers which represent a high security threat to privacy of the people. They are used by phishers to capture keyboard keystrokes, mouse movements and computer screens, in order to exfiltrate their sensitive information [40].

Session Hijacking might occur either at application level (when the HTTP session is interfered by the attacker) or at network level (when UDP or TCP session is hijacked by the attacker) [41]. HTTP session hijacking occurs when the session ID, a unique identifier of a client HTTP session, is exfiltrated from the URL that the browser receives on a HTTP client GET request. Additionally, session ID can be extracted from cookies that are stored on client machine or within the form fields of a webpage. Typical way to obtain session ID is to sniff for unencrypted packages used in the client server communication and to look for user login information [41]. This method is the same as **men in the middle attack**. Nowadays, a WLAN session hijacking is a popular way to obtain confidential information. By performing a denial of service attack the attacker can manage to disconnect a specific machine from the access point (router). Than -the-attacker can copy the MAC address of the legitimate machine and that way connect smoothly to the existing access point [42].

Cross Site Scripting (XSS) is a vulnerability that could be found on dynamically generated web pages when user input is not properly validated on insert [43]. This security hole can be exposed by the phisher to inject a malicious JavaScript code via any form input field. Hence, the attacker could listen for user input or silently redirect them to a fraudulent website.

Men-in-the-middle attack is very hard to be detected, because the phisher is positioning himself between the client and the legitimate website to eavesdrop upon their communication [44]. When the client sends information to the server, the phisher intercepts the request, copy the transmitted data and forward the request to the server. The same thing occurs when the server transmit some data to the client. Later the phisher might use this data to obtain a personal financial gain; or to use it to launch a spare-phishing attack against different target. Because the communication between the client and the server is flawless, this type of attack is very hard to be detected.

2.4 Taxonomy of Phishing Defense

Developing countermeasures to fight against phishing is not a straightforward process. Successful phishing attack rates are very high in practice, because phishers usually exploit weaknesses in human computer interactions which are hard to be eradicated. Many information security researchers proposed different types of anti-phishing techniques to detect and mitigate phishing scam. In general, anti-phishing detection could be classified in 2 distinct categories: user education and training, and automatic phishing detection by using software classification [12]. User education and training approaches enhance the ability of end users to recognize and mitigate phishing attacks and they could be further divided into: game based training, embedded email training and training by sending email security notices to end users. On the other hand, automatic phishing detection help in classifying phishing emails and websites on us eliminate user error in phishing detection. Automatic phishing detection approaches are further classified as: phishing detection by blacklists, phishing detection by heuristics (generating and applying phishing detection rules), phishing detection by visual similarities and phishing detection by machine learning [6]. In this section of the report an overview and a comparison of existing anti-phishing approaches and tools is presented and various phishing evaluation metrics are explained in details. A basic taxonomy of phishing defense mechanisms is shown in Figure 4.

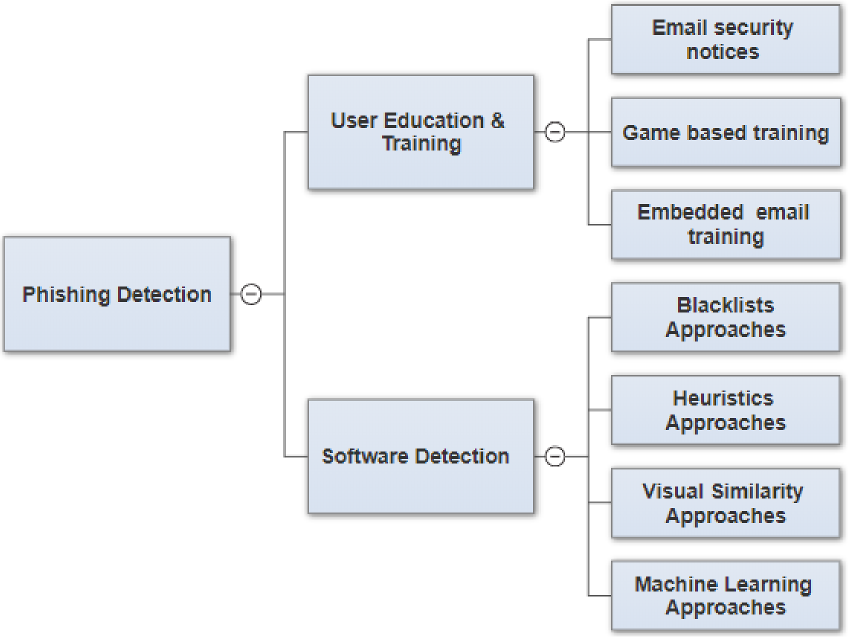


Figure 2.4: Taxonomy of phishing defense approaches

2.4.1 Phishing Performance Evaluation Metrics

Phishing detection is a binary classification problem. A website or email can be classified either as phishing (phish) or legitimate (ham). In a mixed dataset of ham and phishing websites only 4 classification categories exist: **N P - P** (denotes the number of phishing instances correctly classified as phishing), **N L - P** (denotes the number of legitimate instances that are incorrectly classified as phishing), **N P - L** (denotes the number of phishing instances that are incorrectly classified as legitimate) and finally **N L - L** (denotes the number of legitimate instances that are correctly classified as legitimate). By looking at the available literature the following metrics are used to evaluate the performance of phishing detection tools and algorithms [8][43][46]:

- True Positive (**TP**) represents the rate of correctly classified phishing instances in a data set compared to all existing phishing instances.

$$TP = \frac{NP-P}{NP-P+NP-L} \quad (1)$$

- False Positive (**FP**) represent the rate of incorrectly classified legitimate instances in relation to all existing legitimate instances.

$$FP = \frac{NL-P}{NL-P+NL-L} \quad (2)$$

- True Negative (**TN**) - represent the rate of correctly classified legitimate instances in relation to all existing legitimate instances.

$$TN = \frac{NL-L}{NL-P+NL-L} \text{ (3)}$$

- False Negative (**FN**) - represent the rate of incorrectly classified phishing instances in relation to all existing phishing instances.

$$FN = \frac{NP-L}{NP-P+NP-L} \text{ (4)}$$

- Precision (**P**) - measures the rate of correctly classified phishing instances in relation to all instances that are classified as phishing.

$$P = \frac{NP-P}{NL-P+NP-P} \text{ (5)}$$

- Recall (**R**) equivalent to **TP.**

$$R = TP \text{ (6)}$$

- $\boldsymbol{f}$**1** score represents the harmonic mean between Precision (**P**) and Recall (**R**).

$$f\mathbf{1} = \frac{2PR}{P+R} \text{ (7)}$$

- Accuracy (**ACC**) - measures the overall rate of correctly classified instances (phishing and legitimate) in relation to all instances in a data set.

$$ACC = \frac{NL-L+NP-P}{NL-P+NL-L+NP-P+NP-L} \text{ (8)}$$

- Weighted Error (**WErr**) - measures the overall rate of incorrectly classified instances (phishing and legitimate) in relation to all instances in a data set. In the following equation, **λ** is used to penalize the misclassification of legitimate misclassification of legitimate instances is penalized 3 times more than the misclassification of phishing instances.

$$WErr = 1 - \frac{\lambda * NL-L+NP-P}{\lambda * NL-P+\lambda * NL-L+NP-P+NP-L} \text{ (9)}$$

2.4.2 User Education & Training

and inexperience to recognize if a specific email or website is a phish or not. To fight back against phishing, one possible solution is to educate and train users to recognize and mitigate phishing attacks. Downs at al. conducted a survey on 232 internet users and they concluded that educating and training them to recognize the signs of phishing attacks is much more effective than warning them about the risk and consequences of those attacks [47]. There are many different education and training approaches which are used to spread awareness among online users and educate them about phishing in general. Three of the most popular are: training by sending email security notices to end users, game based training, embedded email training and.

2.4.2.1 Email Security Notices

Ideally, an email is the right way to inform and educate users about new trends in security, potential phishing treats and other related news. However, this does not hold in practice because of many reasons such as:

- The content of the email might not be interpreted as it was expected by the sender.
- Users may never read your security notice because they are not interested to read long text about how they could protect themselves from phishing attacks.
- They might think that this email is not relevant to them, because they already feel very secure, which by the way in not true.

Big corporations like Microsoft, PayPal and EBay are constantly revising and improving their strategy on how they could write better email security notices that will be engaging for end users. Those types of security email notices are regularly sent to users to raise their awareness and to educate them on latest security threats like phishing. By conducting a survey to a total number of 1001 participants, it was concluded that sending security notices to educate users, increase user awareness about phishing. This is represented by increasing the TP and decreasing the FN rate when this type of training material is used [48]. On the other hand this study also revealed that users awareness education and training programs in general are likely to decrease the TN rate which is a bed indicator [48]. According to the study, popular training programs like sending security email notices to end users decrease TN rate from 67% to 61%. This probably occurred because some of the participants misclassified some of the legitimate emails since they become oversensitive to the training materials. In one of his studies Kumarguru at al. concluded that sending periodically security notices is not very effective. Despite the fact that they improve awareness about phishing, security notices failed to change user behavior [49].

2.4.2.2 Game Based Training

Game based training and education of the users to raise awareness about phishing is inspired by computer games. The philosophy behind this approach is that by playing computer games intrinsic motivation is increased to perform some actions and to learn new things. By practicing game based training approach long term knowledge about phishing could be acquired more easily.

Anti-Phishing Phil, is a game based approach to teach people how to deal with phishing attacks [10]. The main character in the game is a baby fish named Phil who needs to eat worms (potential phishing attacks) in order to grow up. Each worm is associated with website URL and Phil needs to eat only the good ones, hence rejects bad ones for which he gives some tips to Phil how to detect bad worms (phishing URLs). The primary objective of this game is to teach users how to [10]:

- Effectively identify phishing URLs and successfully distinguish them from legitimate URLs
- Look for clues in the web browser in order to confirm if URL is from legitimate site or not
- Utilize a search engine to check the validity of a suspicious URL

Furthermore, "**Anti-Phishing Phil”** teaches users how to identify 3 types of URL related phishing attacks [10]:

- IP address URLs (URLs having numbers instead of a domain name)
- Subdomain URLs (URLs which have a brand name as part of their subdomains)
- Deceptive URLs (URLs that look like original domain URLs, i.e. brand name followed by a hyphen followed by additional word, security related keywords in the domain name, typo in the brand name etc.)

In a study conducted to evaluate the usefulness of the game it was concluded that the users who played the game were more accurate in detecting phishing and legitimate URLs compared to the users who were trained with existing online training materials offered by PayPal, eBay and Microsoft. With empirical evaluation of the "**Anti-Phishing Phil”** it was reported that the false negative rate drop from 31% to 17% after training [10]. It was also concluded that the demographic variables such as: gender, age, sex and education were not significantly correlated to participan [49] [50] [51]. The results from the study suggested that game based education on phishing attacks mitigation is very promising and effective if it is used as a complementary approach to automatic threats detection techniques [10] [6]. For instance, browser toolbars which implement rule based heuristics or machine learning algorithms could be combined with game based user education to achieve higher rates on phishing attacks detection.

A drawback of game based approach is that some users even after their training could still fail to identify a phish, in situations when malicious URL look very similar to legitimate one. This is because users are mainly trained to examine only the URL and not to look for other clues, like the data that is requested by the phishing website. With evaluation of this game it was reported that the false positive rate increased from 37% to 48% which is a very high number compared to false positive rate generated by software detection approaches [10]. Furthermore, some users suggested that using a fish as a main character of the game is appropriate for children, but not for adults. Anti-phishing Phil was first implemented by Carnegie Mellon University, but it was later commercialized by Wombat Security [52].

2.4.2.3 Embedded Email Training

An embedded email training system aims to educate people how to protect themselves from phishing attacks, while they are doing their regular email activities [49]. This approach consists of periodically sending fake phishing emails to end users, usually from system admin or training company. If a person receives an email and click on a link contained in it, he/she is instantly redirected to an educational web page where an intervention message is shown to him. This intervention message provides immediate feedback to the users about what happened and suggests them further steps how to mitigate similar phishing threats in the future. Kumaraguru at al. [49] proposed two similar design prototypes on how those interventions should be crafted. The first design prototype shows a screenshot of the user browser, in state identical to the browser state when the user opened their email and clicked on the phishing link. Additionally, the intervention contained clues why this is a phishing emails and some tips on how the user could protect themselves from this kind of treats in the future. The second prototype presented the same logical information, but in a comic strip format with less text and more images. A lab experiment was conducted to evaluate those intervention designs with real users. The survey results indicated that embedded training is helpful to teach people to mitigate phishing attacks. The comic strip embedded training showed better results compared to browser screenshot intervention design. Only 30% of the users, from embedded training comic group, failed to detect the last phishing email, after 2 intervention messages were shown to them. 70 % of the users from the first intervention group (browser screenshot design) failed to detect the last phishing email [49].

PhishGuru is an implementation of embedded email training system that sends fake emails

[2]. This moment occurs when the user actually click on the malicious link located inside the phishing email [2]. Kumaraguru the creator of **PhishGuru,** conducted a study which showed that users who attended 3 training session with PhishGuru retain their knowledge about phishing, even 28 days later [53]. Additionally, the study concluded willingness to click on links shown in legitimate emails was not decreased after showing them the embedded training intervention messages. PhishGuru training interventions are also adopted in a real-world project developed by APWG [54] whenever some user visits a phishing website which is already taken down by the authorities, he/she will be redirected to APWG landing page. Additionally, APWG asks ISP providers and website registrants to participate in their initiative and to implement the redirect for each phishing site that it will be taken down. PhishGuru is currently being offered as a commercial software as a service (SaaS) solution by Wombat Security [52]. An alternative SaaS implementation of PhishGuru, called PhishSim, is offered under freemium license by SecurityIQ corporation [55].

Embedded email training approach has some drawbacks that need to be addressed. Crafting intervention templates and collecting fake phishing emails is done by the system admins [49]. This might add considerable delays and increase largely the maintenance cost of the system [6]. Additionally, the quality of the embedded training materials depends on the administrator knowledge about phishing and his awareness of the latest trends in phishing scams. All those facts limit the overall effectiveness of the training program. PhishGuru and all similar training approaches reward only the bad user behaviors i.e. click on a link in a phishing email. The user is not rewarded if he/she intentionally ignores a phishing email or identifies a legitimate one. Punishing only bad habits and educating users in so called After seeing many intervention messages users might become oversensitive regarding email classification therefore they could miss some important information. Embedded training might raise some ethical and legal issues. For example, mass mailing of fake phishing emails, even for educational purpose, might not be approved by some organizations, companies or educational institutions. Some ISP providers may blacklist your server IP address and therefore users will never receive your embedded training emails.

2.4.3 Automated Software Detection

Educating and training user to recognize and mitigate phishing attacks is not enough. A study [11] showed that even when the users are trained with the best phishing awareness program they failed to identified 28% of the phishing attacks. A more sophisticated solution to fight against phishing might be an automatic phishing detection using software tools and machine learning algorithms. According to the literature, these phishing detection techniques achieve lower FP and FN rates compared to user education and training approaches [6][12].

2.4.3.1 Phishing Detection by Blacklists

Blacklists are frequently updated collections of malicious IP address and URLs which are detected and reported in the past [12][18]. On the other hand, whitelists are opposite to blacklists and they contain IP address and URLs of legitimate websites. Whitelists are generally used to decrease the FP (false positive) rate of phishing detection algorithms. According to Sheng at al., blacklist does not provide protection against zero-day phishing attacks (attacks not seen before) and they will likely detect only 20% of those attacks [56]. Furthermore, the same study [56] shows that 47-83% of phishing URLs are detected minimum 12 h after they have been activated. This delay is significant since 63% of phishing campaigns ends within 2 hours after they have been initially launched [56]. Popular implementations of phishing blacklists are: Google safe browsing API and PhishTank.

Google safe browsing API is a RESTful API service that enables users and apps to verify if a certain URL is a security threat by consulting URL blacklists regularly updated by Google [57]. The service works in a way that allows clients to send a suspicious URL for verification using GET or POST HTTP request to the API. The current implementation of the API maintains 2 constantly updated blacklists: a phishing URL blacklist and malware

URL blacklist. In the HTTP request the clients have to specify the blacklist that should be consulted by the API. The response of the API is an XML or JSON object which contains the answer if a specific URL is blacklisted, the type of the treat (malware or phish) and the platform type (operating system) for which the treat is intended. In addition, Google safe browsing API provides a RESTful endpoint to download a hashed version of supported blacklists in order to store them in a local database. This service is still in an experimental phase, but it is already implemented as a security protection mechanism in modern web browsers like Google Chrome and Mozilla Firefox. A major drawback of this approach is that the response time of the API endpoints is not limited and sometimes the API feedback might be prolonged which is not very practical [12].

PhishTank is a collaborative blacklist, where people from around the world can submit and verify suspicious phishing URLs by voting [58]. If a certain suspicious URL has enough positive votes it is verified as a phish. On the other hand, a suspected phish might have negative votes which mean that it is legitimate. Similarly to **Google safe browsing API,** PhishTank offers free REST API endpoints to check if a certain suspicious URL it a phish or not. PhishTank data is available in multiple formats: XML, JSON or CSV. Furthermore, PhishTank provides additional endpoints to download the entire database of verified phishes which is updated hourly. This service also provides useful month by month statistics of submitted URLs such as: top 10 phishing domains, top 10 IP addresses, and top 10 phishing targets. According to their statistics median time to verify a suspicious URL is 13.5 hours. Except from the common disadvantages know for blacklists, PhishTank may produce a higher FP rate compared to **Google safe browsing API** because of the user voting. However, they also provide a mechanism to report FP submissions for further review. PhishTank is operated by OpenDNS a company founded in 2005 [59].

PhishNet is a predictive blacklist which addresses the limitations of traditional blacklists to fight against phishing. Traditional blacklist solutions will not be able to detect a phishing attack if the URL which is submitted for checking is not 100% equal to the one stored in the blacklist database. Phishers usually deploy the same phishing page on multiple URLs that might be very similar one to another. To address this limitation PhishNet will produce multiple variation of the submitted URL by following 5 different heuristics such as [60]:

- Replace the TOP level domain (TLD). This will result with creation of 3210 variations of the submitted URL.
- URLs that belong in the same directory folder will be grouped together and from existing filenames in those directories new URL variations will be generated.
- URLs that have a similar directory structure but different domain name are considered equivalent if and only if they point to a same IP address. All combinations of host names and paths of the matched URLs are combined and new URL variations are generated.
- URLs with a same directory structures but different query string paths are combined together by swapping their path part and that way generating new variations of targeted URL.

- URLs which have a brand name in their signature are used to create child URLs by replacing their brand name with different one.

2.4.3.2 Phishing Detection by Heuristics

In general, phishing attacks could be detected by following a set of rules known as heuristics. Not every heuristic rule will be applicable to all types of phishing attacks. The goal of anti-phishing heuristic defense is to extract valuable features from each phishing attack and to create a general set of rules that will be able to predict new forms of phishing. Detecting zero-day phishing attacks is the primary advantage of heuristic based anti-phishing techniques over blacklist (cannot detect attacks which are not seen previously) [12]. A possible drawback of this approach is that generalized heuristics might run in the risk of misclassifying legitimate websites and emails if the heuristic rules are not properly crafted therefore it will produce a higher FP rate. Many popular web browsers and email clients such as: Mozilla Firefox, Internet Explorer, Mozilla Thunderbird and Microsoft Outlook use heuristics to provide real-time filtering of phishing content (emails, websites and URLs) [18]. Furthermore, anti-virus provides like Avast and Norton Antivirus from Symantec provide real-time anti-phishing protection to end-users, implemented using heuristic rules [27][61].

PhishGuard is a browser plugin that applies heuristics rules to detects phishing attacks by examining authentication HTTP requests communicated between the client and suspected website [62]. This plugin follows the idea that phishing website usually do not verify the validity of user credential rather they store them to perform further illegal actions. The limitation of this plugin is that works only in situations when the attacker will not pipe the authentication response from legitimate page to show to the user if his/her login request is successful or not therefore this plugin will not protect users from men-in-the-middle attacks. The detection cycle of this plugin is the following. The user sends his/her login credential to the suspected website by filling a login form. The plugin than intercepts the request, substitutes the correct password with fake one, replicate the request N times and forwards them to the suspected website. If the website responds with success (HTTP 200 OK message) than this website is labeled as phishing and the client is notified. On the other hand, if the websites responds with HTTP 401 code than 2 different outcomes are possible: a) the site is phishing and repeatedly responds with failed authentication message; b) the site is legitimate. To find the right answer PhishGuard will submit the real credentials to the website. If the website responds with HTTP 200 OK status than it is considered to be legitimate if not than two additional outcomes are possible: the website is a phish or the user credentials are wrong. This step is considered as a potential security leak in the algorithm because in case the website is a phish than the phisher will obtain the correct user credentials [6]. To make sure that the user password is correct PhishGuard maintains a list of hashes of correct user passwords and verifies future login requests against it [62]. Finally, if the hash of the entered password is contained in the list of hashes maintained by the plugin than the site is labeled as phish else the user is notified that his/her password is wrong and a correction is required.

Phishwish is an anti-phishing algorithm that is used to detect phishing emails by applying 11 heuristic rules to examine email headers, text and links [63]. This solution is much lighter and more efficient (uses only 11 rules) compared to SpamAssassin (another heuristic solution to detect spam and phishing emails) which applies 795 rules to detect if an email is phish or not. The idea behind this algorithm is to provide protection against zero-day attacks with minimal misclassification of legitimate emails. Some of the most important rules used by the algorithm are: if the URL of the link contained in the email has an IP address instead of a domain name for the host; if the senders organization name is not present in the SMTP header of the received email; if there are some inconsistencies in the WHOIS record for the domain portion of the URL contained in the email body [63]. For more detailed information about the rules applied by the algorithm please look at the referenced paper. The final score of the algorithm is calculated by the weighted mean of these rules. This score is further used to predict the class for the email according to a predefined % than the email is classified as phishing otherwise the email is labeled as legitimate. If some of the rules are not applicable to the email they are excluded from the calculation of the final score.

CANTINA is content based anti-phishing approach, implemented as a toolbar for Internet Explorer, used to classify web pages as phishing or legitimate [64]. CANTINA implementation is based on Term Frequency-Inverse Document Frequency (TF-IDF) information retrieval algorithm, search engine results and a set of simple heuristic rules to reduce false positives. TF-IDF represents a numerical value that shows what is the importance of a word to a document in a given collection of documents [64]. The detection process of CANTINA algorithm can be represented in few simple steps. First, the TF-IDF value for each lexical term (word) shown on a suspected webpage is calculated. Second, top 5 terms with heights TF-IDF value are used to tag the document. Next, those 5 terms are used to query the Google search engine and a predefined number of N search results are stored. Finally, if the URL of suspected webpage is within the stored search results than the website is labeled as legitimate, otherwise the website is classified as phishing. The estimated accuracy rate of the algorithm is around 95% with a false positive rate of 6% [64]. Sometimes the performance of the algorithm might be affected by the delay in receiving feedback from search engines [12].

2.4.3.3 Phishing Detection by Visual Similarity

An alternative way to detect phishing is to look for visual similarities among suspected and targeted webpages. This phishing detection technique is very useful since phishing webpages usually look almost identical to their legitimate counterparts, particularly in layout positioning, images, branding color and text fonts. Fu at al. [65] implemented a technique which uses Earth Distance to calculate visual similarities among webpages. The first step in the algorithm is to collect and convert webpages (phishing and legitimate) into low resolution snapshots with dimensions 100x100 pixels. Next, snapshots signatures are generated where the dominant color category and its corresponding centroid coordinates are used as features. EMD algorithm is used to calculate the visual similarity index between the signatures of suspected and legitimate webpages [65]. A threshold value for each legitimate (protected) webpage in the dataset is calculated upfront. Furthermore, if the value obtained from calculation of the EMD visual similarity of a certain webpage exceeded the threshold of its corresponding legitimate webpage than the page is classified as phishing otherwise the webpage is classified as legitimate [65].

The authors in [66] proposed a technique to visually compare suspected phishing webpage to legitimate one by using 3 different features such as: webpage text excerpts and their style, images embedded in the web page and page visual appearance when it is rendered in a browser. To empirically evaluate their approach, authors used a dataset consisted of 41 pairs of phishing and legitimate webpages obtained from PhishTank database. They also reported that the algorithm performs satisfactory with acceptable false positive and false negative rates. A common drawback of all visual similarly phishing detection approaches is that they perform poorly in detecting phishing webpages which does not look similar to any legitimate webpage.

2.4.3.4 Phishing Detection by Machine Learning

Machine Learning phishing detection is a novel approach to phishing which yields high accuracy and low false positive rates. Each phishing attack is translated into a vector of features that could be feed into a machine learning algorithm able to solve a classification or clustering problems with 2 output classes (phishing or legitimate). Commonly used machine learning classifiers to solve a phishing problem are: Support Vector Machines (SVM), K-nearest neighbors (k-NN), Naïve Bayes, Neural Networks, Decision Trees, Random Forest etc. [6][12].

PhishAri is a browser extension for Google Chrome which uses machine learning to perform real-time phishing detection of tweets [67]. PhishAri makes use of Twitter Streaming API to access tweets in real time. Set of URL based features, Twitter specific features and WHOIS features such as: embedded tweet link, URL length, a

link domain, account ownership, presence of trending hashtags, follower-followees ratio, number of followers etc. are used to classify a tweet as phishing or legitimate. A Random Forest machine learning classifier is used to classify tweets based on proposed features, using 5 fold cross-validation. Training and testing dataset consisted of 1473 phishing and 1500 legitimate tweets, which were partitioned in 5 subsets. Four of the subsets were used for training and 1 subset of data was used for testing the classification accuracy of PhishAri browser extension. Authors reported that the overall accuracy of the algorithm is high 92.52%, however the algorithm misclassified 9.6% of legitimate tweets [67]. They also reported that this happened due to similar behavioral habits of regular users and phishers, where regular Twitter users make extensive use of unrelated hashtags and they also use software for tweeting automation just like phishers do [67]. PhishAri is considered very efficient regarding the time needed to classify single tweet. When the solution has been deployed on Intel Xeon 16 core Ubuntu server with 2.67 GHZ processor and 32 GB RAM the average time to classify a single tweet was 0.425 sec. However, this time may vary because it depends strongly on third party services such as: Twitter Streaming API, WHOIS database lookup and Internet bandwidth. According to creators of PhishAri, this browser extension is the first tool for real-time phishing detection on Twitter [67].

PILFER is a machine learning technique which is very successful in detecting social engineering deception in online communication especially email phishing [13]. Authors in [13] reported that with a minimal modification of the algorithm this method might be applicable in detection of phishing websites. PILFER approach is tested with different machine learning classifies like SVM, rule-based approaches and Bayesian approaches but it performed best with Random Forest classifier [68]. Ten different features which are used as an input to the Random Forest classifier are extracted from within suspected emails and they could be grouped in 2 categories: external and internal features. The first group of features like: IP based URLs, number of links, number of dots in a link, are extracted from the email body itself. On the other hand, external group of feature such as: age of linked domain names and spam filter output are acquired from third party services. Additional features like: site in browser history, TF-IDF and redirected site are proposed by authors when PILFER is used in web page classification. This method has been evaluated on 6950 ham and only 860 phishing emails. Consequently, this method produced an overall accuracy rate of 99.5% with over 96% true positive and only 0.1% false positives. However, this result should be taken with caution because there is a high discrepancy between the numbers of phishing and legitimate emails used in the classification process. As it was stated by the authors, the dataset of phishing emails might not be a representative sample of real world user inbox because it is quite old [13]. They also stated that this phishing dataset is the best one available on the Internet.

2.4.4 Comparison Table of Anti-Phishing Tools and Approaches

Technique Name	Technique Type	Advantages	Drawbacks
Google Safe Browsing API [57]	Blacklist	1. Built-in browser protection 2. False positive rate 0% [69] 3. RESTful API access for free	1. Poor in detecting zero-day attacks 2. FN rate 16%-30% [69]
PhishTank [58]	Blacklist	1.Integrated with major antivirus solutions 2. Community based phishing verification 3. RESTful API access for free	1. Median time to verify a suspicious URL is 13.5 hours 2. Produces higher false positive rates compared to Google Blacklist
PhishNet [60]	Predictive Blacklist	1. False Negative rate of 3% 2. Remove the exact match limitation of traditional blacklists	1. Increased size of blacklist due to URL variations which also increase the bandwidth demands 2. FP rate of 5% is considered too high for practical daily use
PhishGuard [62]	Heuristics	1. Effective against zero-day attacks	1. Manual rule updating is required

		2. Implemented as browser plugin	2. Empirical evaluation of the tool is not presented in [62]
PhishWish [63]	Heuristics	1. Effective against zero-day attacks 2. Good false negative rate of 2.5%	1. High false positive rate 8.3% 2. Not very adaptive to new phishing treats because the rules are hardcoded
CANTINA [64]	Heuristics	1. Accuracy rate of about 95% when additional heuristics is used	1. High false positive rate from 3% to 6% when the TF (term frequency) of specific word is an outlier 2. Performance issues due to delay in querying the search engines 3. TF-IDF does not work well with Eastern Asian Languages
Fu at al. [65]	Visual Similarity	1. Low computational complexity 2. Classification precision of 99.87%	1. Can detect only visual similarities at pixel level but not at text level 2. Low value of phishing recall 88.88%
Authors in [66]	Visual Similarity	1. No misclassification of legitimate websites	2. Very small testing set. Only 41 pairs of phishing and legitimate images
PhishAri [67]	Machine Learning	1. Provides real-time phishing protection (a browser extension) 2. Better performance than -phishing mechanism 3. Accuracy rate over 92% 4. Time efficient: on average 0.425 sec. to classify a tweet	1. High misclassification rate of legitimate tweets 9.6%
PILFER [13]		1. Can be used both for email and website detection 2. Accuracy rate of 99.5%	1. Dataset of phishing emails not very representative, it is too old 2. Large discrepancy between the numbers of phishing and legitimate emails: 860 to 6950

Table 2.1: Comparisons of anti-phishing tools and approaches

2.5 Summary

It has been more than two decades since phishing emerged as number one online security threat. The cost of phishing is exponentially increasing year by year, thus creating many problems to individuals, companies and organizations. A broad range of phishing defense mechanism have been developed by researchers and online community to fight against phishing, however phishers always find a way to break those defenses and to continue with their attacks. In general, phishing attacks are classified in 2 distinct categories: deceptive phishing and malware phishing. In deceptive phishing, social engineering plays a key role and phishers lure their victims to disclose confidential information by using spoofed emails and fraudulent websites. On the other hand, malware phishing is based on technical tricks to steal users credential and credit card details.

Defense against phishing attacks is a hot topic nowadays. Different type of automated phishing detection techniques exist in practice such as: blacklists, heuristics, visual similarity and machine learning, but each of them has some potential drawbacks. Although blacklists produce a very low FP rates, they perform poorly in detection of zero-day phishing attacks and in the same time consume a lot of bandwidth. Heuristics are much better in detecting zero-day phishing attacks compared to blacklists, but if the rules are not properly implemented it yields high FP positives. Many experts agree that systems that misclassify a lot of legitimate websites or emails might cause more harm than good, because users will start to ignore their warnings. Visual similarly is a novel anti-phishing approach, and it is heavily rely on creating and maintain databases of website snapshot silent point descriptors [6]. They are conceptually very similar to blacklists and whitelists and should be frequently updated. Compared to blacklists they are highly accurate in zero-day phishing attack detection. Furthermore, another drawback of visual similarly approach is that it assumes that phishing websites are visually similar to legitimate websites which is not always true. Machine learning approaches are considered to produce the best results in phishing detection. They are able to mitigate zero-day attacks and they achieve very low FP and high accuracy rates sometimes higher than 99% [13]. Drawbacks of these approaches are considered to be the time needed for training the classification model and the resources needed to process input data and perform the actual classification. Additionally, a respectable datasets for training machine learning algorithms are hard to be located on the Internet. Currently available phishing datasets does not reflect the current state of phishing attacks, therefore the need for creating a tool that will preprocess phishing data to generate state of the art phishing datasets will be of a great benefit to the research community.

Since phishing often rely on social engineering and less on technical subterfuge, educating people how to recognize and mitigate phishing attacks is a great benefit for everyone. A company or organization might have installed the best anti-phishing protection software but if an attacker is able to obtain user credential from an employee the whole security infrastructure will be ruined. Many different studies [2][46] [48][53] suggested that raising users awareness by educating and training users will reduce their susceptibility to phishing attacks. Contrariwise, a study [11] showed that even when the users are trained with the best phishing awareness program they failed to identify 28% of the phishing attacks. Stefan Gorling, a professor and researcher at KTH Royal Institute of Technology in Stockholm, argues that a major goal of an employee is to be productive and security is considered to be a secondary goal, therefore it is not advisable to put pressure on employees to study about security, because they are too busy doing their primary obligations [7]. Since automatic phishing detection yields much higher accuracy rates in detecting phishing attacks than user training and education, it could be implemented as a first line of defense against phishing. Consequently, user education and training could be used as secondary defensive countermeasure to complement the automatic phishing detection approaches.

Chapter 3

3. System Requirements and Analysis

3.1 The Goal of the System

Phishing is a pervasive security threat that affects many different individuals, companies and organizations worldwide. The consequences of successful phishing attack are devastating for everyone and are represented by identity theft, data theft, financial loss and reputation damage. Various countermeasures to fight against phishing scam are proposed by the research community nevertheless each one of them has some potential drawbacks. Automatic software anti-phishing techniques are very efficient in detecting phishing attacks, but they are useless in raising end-user awareness about phishing [6]. Traditional automated software based anti-phishing protection might be beneficial to individual6(less)-99(in)-101(ra zero-day phishing attacks will be anticipated. In the same time, by submitting suspicious URLs to the platform and by looking into generated statistics for each website, users are educated and trained to recognize future phishing attacks. Even in a situation when the platform will fail to correctly classify a webpage, by consuming their already acquired anti-phishing knowledge and by looking in the detailed website statistics generated by the platform, users will be able to successfully classify suspected websites on their own.

Secondary goals of PhishEduPro anti-phishing platform are:

- Offer a method to integrate PhishEduPro anti-phishing algorithm into various third-party tools and services to provide them with a real-time anti-phishing protection.

- PhishEduPro platform is envisioned as an open source solution that will open its data to the community. One of the goals is to provide a free access to frequently updated dataset of phishing and legitimate webpages labeled with binary anti-phishing features that could be used as a basis for training and testing new forms of machine learning and heuristics enabled anti-phishing algorithms. Many researchers state that new and reliable datasets for training machine learning algorithms are hard to be located on the Internet [13].

- To find a way to provide an easy tuning of anti-phishing rules utilized by the platform to classify web pages, when new trends in phishing emerge, in order to improve the accuracy rate of the algorithm.

Following the high level goals of the proposed system a detailed set of functional and non-functional requirements is identified and properly documented.

3.2 Functional Requirements of PhishEduPro Platform

By thoroughly reviewing the literature and by having regular discussions and brainstorming sessions with my supervisor, PhishEduPro platform is envisioned to be implemented as a SaaS (Software as a Service) web-based solution, which could be accessed and operated by authenticated end-users. A set of functional requirements for the proposed platform is presented in the following bullet list:

- **FR1:** Authenticated end-users should be able to submit URLs of suspected webpages to the platform for real-time scanning. Therefore the platform should provide a mechanism to classify them as phishing or legitimate.

- **FR2:** PhishEduPro platform should offer a functionality to show detailed classification statistics of each analyzed webpage. This way each end-user will have a continuous insight into the decision process of the anti-phishing algorithm. Hence, by acquiring long-term knowledge users are trained and educated to recognize and mitigate future phishing attacks.

- **FR3:** The platform should provide an integrated search mechanism to facilitate an easy access to its database of evaluated URLs. By providing this functionality the platform could be utilized as frequently updated blacklist and whitelist of suspicious URLs.

- **FR4:** The platform should provide an easy access to a set of evaluation metrics used to measure the performance of its anti-phishing algorithm. By looking into those metrics end-users could decide if further fine-tuning of the algorithm is required.

- **FR5:** PhishEduPro platform should provide an interface to fine-tune the rules and features used to classify suspected and legitimate webpages. By changing the thresholds of existing anti-phishing rules and features, supported by the platform, the accuracy rate of the anti-phishing algorithm could be significantly improved and future trends in phishing might be anticipated.

- **FR6:** A functionality to export a frequently update dataset of phishing and legitimate webpages, labeled with binary features, should be made accessible for everyone. This dataset can be further utilized by researchers to implement, train and test novel anti-phishing algorithms based on machine learning or rule based heuristics.

- **FR7:** A functionality to login, to create new user account, to delete existing user accounts and to change password details should be offered by the platform. Each user must be authenticated and properly authorized in order to use the functionality provided by the system. This is considered as a precaution step to fight against distributed-denial-of-service (DDoS) attacks and unauthorized update of anti-phishing rules used in web page classification process.

3.3 Use Case Diagram of PhishEduPro Platform

In this section of the report a use case diagram of the PhishEduPro platform will be presented (see Figure 3.1) to visually illustrate the behavior and relationships among various external actors and the system.

Figure 3.1: Use Case diagram of PhishEduPro platform

From the use-case diagram presented in FBT7(i36 0 0 1 99.264a)-911 99.2wse c

aethe ths cadiag a.( )] TJETBT1 0 0 1472.31 23.069 T

3.4 Requirements of the Anti-Phishing Algorithm

One objective of this project is to develop an anti-phishing algorithm that will be capable to accurately classify web pages, submitted by end-users, as phishing or legitimate in real-time. According to the metrics obtained from anti-phishing literature the algorithm should achieve an accuracy rate over 97% to be considered as a reliable alternative to existing anti-phishing solutions. Furthermore, the algorithm should be capable to detect and report zero-day phishing attacks and should perform well in classification of legitimate websites. The performance of the algorithm is another issue that needs to be address. Since it will classify web pages in real-time the results from the classification should be promptly report to end-users. Thus, users should immediately understand that a certain website is a potential phish and they will navigate away from it without providing their confidential information.

There are various anti-phishing related features that could be feed as an input to an anti-phishing algorithm used in web page classification. Mohammad at al. in their work [71] proposed a novel classification of anti-phishing features which is widely accepted by the research community. Four distinct categories of features were identified by the authors such as: Address Bar Based Features, HTML and JavaScript related features, Abnormal Based Features, and Domain Based Features [71]. Address Bar related features could be extracted from the URL of the suspected webpage. Abnormal Based Features are characteristics which are related to anchor tags, script tags, link tags and form tags placed inside the suspected webpages, such as= a
JavaScript related features could also be extracted from the code of the suspected website and they are represented by: IFrame redirects, use of popup windows and right mouse click disabling. Domain Based Features are usually obtained from third party services like WHOIS repositories and Alexa Database and are represented by: Domain Age, Website Ranking and DNS Record information. Many of those features together with some newly proposed one could be utilized by the novel real-time anti-phishing algorithm. More about the design of the anti-phishing features, and their classification impact will be discussed in the chapter 5 of this report.

In the following high-level workflow diagram (see Figure 3.2) an initial idea about a new anti-phishing algorithm is presented. The general workflow of the algorithm is described by showing the initial state, intermediate states, decision points and final state of its novel decision making process. Possible bottlenecks of the algorithm are also discussed. The detailed workflow diagram of the proposed anti-phishing algorithm is presented in chapter 5 (System Architecture and Design), after we design and evaluate the final set of the most promising anti-phishing features with their corresponding rules.

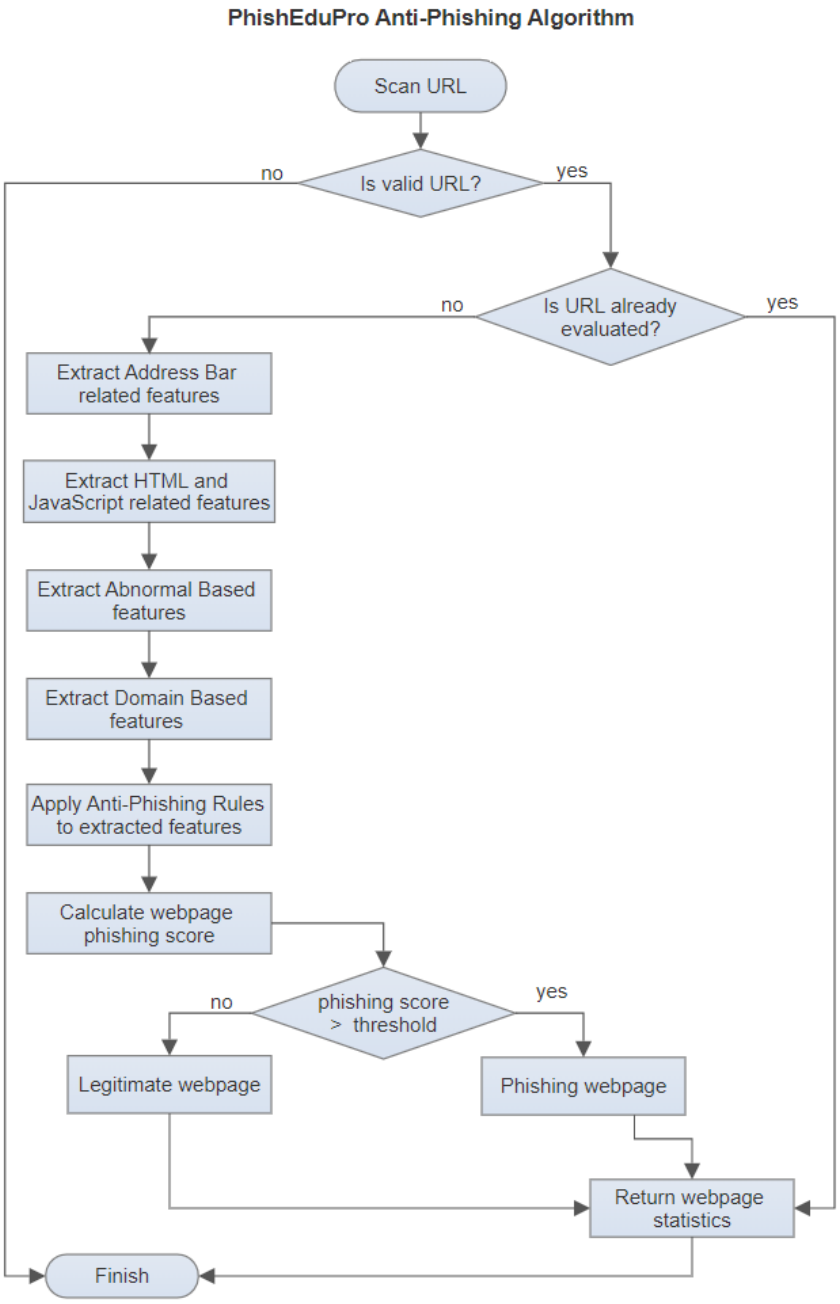

PhishEduPro Anti-Phishing Algorithm
Scan URL
Is valid URL?
no
yes
Is URL already evaluated?
no
yes
Extract Address Bar related features
Extract HTML and JavaScript related features
Extract Abnormal Based features
Extract Domain Based features
Apply Anti-Phishing Rules to extracted features
Calculate webpage phishing score
phishing score > threshold
no
yes
Legitimate webpage
Phishing webpage
Return webpage statistics
Finish

Figure 3.2: High-level workflow diagram of PhishEduPro anti-phishing algorithm

The entry point in the PhishEduPro anti-phishing algorithm is the moment when a specific end-user submits an URL from suspected webpage for scanning. The algorithm first checks if the submitted URL is valid. Valid URL has dual meaning. First the algorithm checks if the URL is properly formatted by flowing URL standardized pattern. Moreover, it will check if the webpage behind this URL is currently online. In case the URL is invalid the algorithm halts, otherwise the algorithm checks if the URL is already evaluated. Evaluated URL is nothing more than a webpage with calculated phishing score and generated anti-phishing features statistics. If the algorithm located such webpage it returns its related data to the end-user and algorithm execution terminates. If the URL is not already evaluated than the evaluation process of the algorithm begins. During its evaluation phase, the algorithm extracts URL related features, HTML and JavaScript related features, Domain Based features and Abnormal Based features already discussed in this section. By applying specific rules bound to those features the phishing score of the suspected webpage is calculated and a set of individual feature scores is generated. Each feature from the set of evaluated webpage features is assigned with a value 1 (legitimate) or -1 (phishing). Some of the features are ternary and can be additionally evaluated with 0 (suspicious). In addition each separate feature has additional score attached to it. For instance, the age of the domain represented in days is a score of the feature Domain Age . In the next step of the algorithm the phishing score of the suspected webpage is compared to predefined threshold. If the value of the phishing score is above the threshold than the webpage is labeled as phishing, otherwise the webpage is labeled as legitimate. In the final step of the algorithm execution the evaluated webpage together with its generated statistics (phishing score, binary features and individual feature scores) are returned to the end-user. A possible bottleneck of the proposed anti-phishing algorithm could be the features extraction process. Each of the proposed features is extracted by using specific tools and third party services. If they are extracted in sequential order than the proposed algorithm evaluation process might be inefficient for real-time phishing detection. A possible solution to this problem could be parallel processing of the features. More about the design of parallel feature extraction process will be discussed in chapter 5.

3.5 The Non-functional Requirements

The proposed system for PhishEduPro platform and its anti-phishing algorithm should satisfy the following non-functional requirements.

- **NFR1:** The accuracy rate achieved -phishing algorithm should be higher than 97%. In addition, the false positive rate (misclassification of legitimate website) must be lower than 3%.

- **NFR2:** PhishEduPro anti-phishing algorithm should scan suspected webpages in real-time, as soon as the webpage is submitted by the authenticated user and should operate exclusively online.

- **NFR3:** Web page statistics generated by the platform cannot be altered or deleted after being stored in the database.

- **NFR4:** The propos
unless a new request to scan a web page is being submitted.

- **NFR5:** The average time required to scan a newly submitted webpage should be between 3 - 4 seconds when the system is deployed on Intel Xeon 16 core Ubuntu production server with 32 GB RAM and 3.0 GHz Quad Core processor. The average time to scan a newly submitted webpage when the system runs on development environment equipped with Intel i7 Dual core 2.4 GHz processor with 4GB RAM should be between 12 14 seconds. If a web page is analyzed within the proposed time frame the user will be promptly notified that the suspected webpage is labeled as phishing. Thus, it is very likely that users will not provide their personal or financial information to the fraudulent website.

- **NFR6:** A reliable broadband internet connection is required. The data throughput at production and development environment should not be less than 6 Mbps. This is required, because the anti-phishing algorithm utilized by the platform should communicate with external APIs to obtain features used in the classification process of suspected websites.

- **NFR7:** An automatic daily backup of platform database should be scheduled to prevent data losses.

- **NFR8:** The search function of the platform should return up to 10 search results (web page statistics) on each search request. With the support of paging (loading small subset of requested web pages and their corresponding features) the performance of the system will be greatly improved.

- **NFR9:** The system should be scalable enough to handle excessive simultaneous web page scans by increasing hardware resources like CPU processing power and RAM memory.

- **NFR10:** System extensibility with new anti-phishing rules and website detection features should be done with minimal programming effort and changes to the core implementation.

- **NFR11:** An authorization and authentication of users should be mandatory for most of the functionalities provided by the PhishEduPro platform. Only the registration of new user and extraction of anti-phishing dataset should not be behind a secure interface.

Chapter 4

4. Project Management

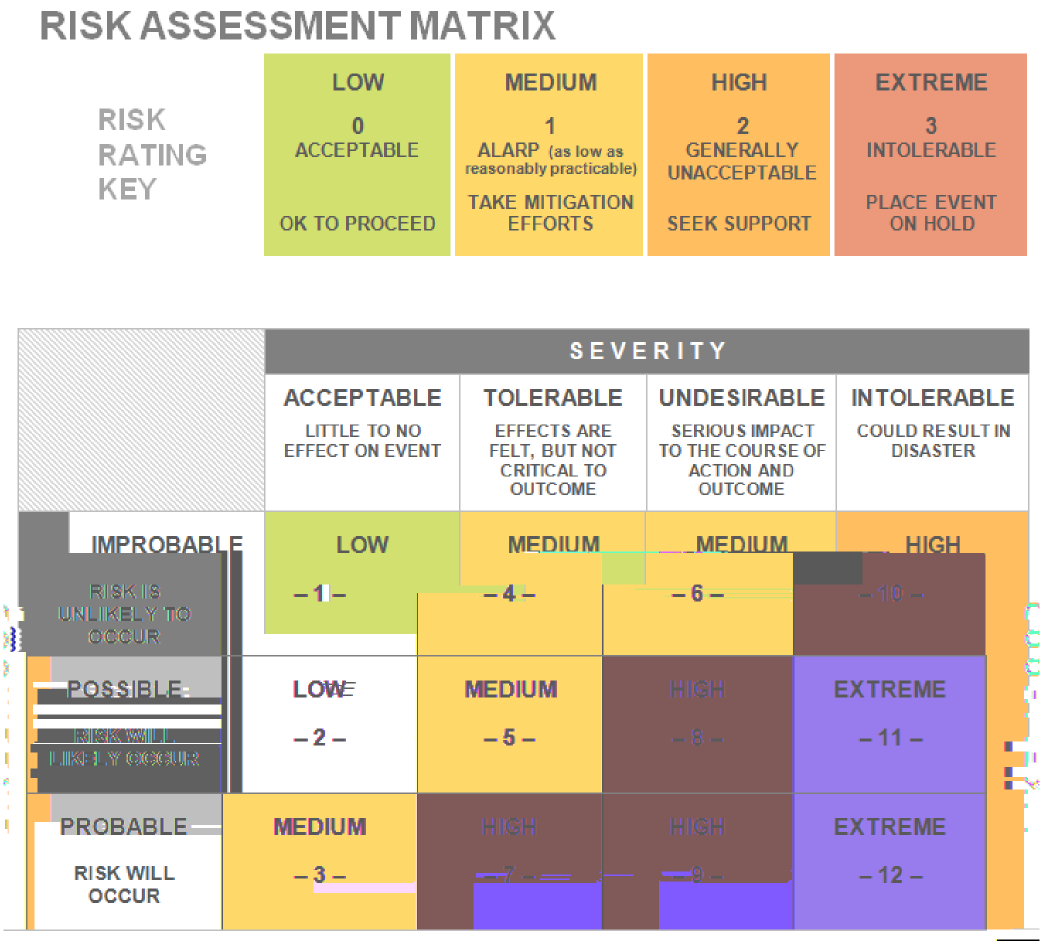
4.1 Software Development Process

By looking into the functional requirements of PhishEduPro platform two separate systems are identified. The first system is represented by a novel anti-phishing algorithm able to classify suspicious webpages as phishing or legitimate in real-time. The second system is the actual platform which will offer different set of functionalities to authenticated end-users and external victors. The platform will provide an interface to end-users for submitting URLs of suspected webpages for scanning. The scanning is performed by an anti-phishing algorithm and the results are returned back and presented to end-users in a way to educate and train them to recognize and mitigate future phishing attacks. Furthermore, the platform will provide an additional interface for users to search for existing web pages, interface for adjusting phishing detection rules and feature thresholds, interface for displaying evaluation metrics to measure algorithm performance, interface for exporting a dataset of phishing and legitimate web page and an interface for managing users.

Our initial idea is to implement those systems as a set of independent services which could be easily reused in third party applications. An early functionality of the PhishEduPro platform is planned, since it should be offered for testing to potential end-users. Thus, we will obtain their immediate feedback to improve the initial requirements and to plan the next releases. An iterative and incremental software development process is most suitable for this type of applications, because it supports iterative development cycles (iterations) involving continuous user feedback and incremental addition of features where early functionality of the product is achieved [72]. Moreover, to implement a robust, reliable and efficient anti-phishing algorithm we need to follow an evolutionary software development process. Therefore we will be able to quickly and continuously prototype each version of the algorithm by receiving continuous user feedback and learning from mistakes done in the previous iterations. Consequently, based on the algorithm performance achieved on a testing dataset, consisted of legitimate and phishing webpages, we could adjust its future prototypes by removing, adding new or updating existing . By incrementally evolving and prototyping the design and structure of the algorithm we can greatly improve its accuracy rate.

4.2 Risk Management Plan

The ability to identify project risks and to propose their mitigation strategy early in the development cycle will greatly decrease projected maintenance cost and will prevent software disasters [73]. Taking into consideration the development of the PhishEduPro platform there are some potential risks that need be addressed early in the development process. Furthermore, their impact to the project and their likelihood of appearance should be estimated by using risk assessment matrix. Figure 4.1 shows the Assessment that might affect the development of PhishEduPro platform. The rows of the matrix measure the risk likelihood of appearing and the columns of the matrix measure the risk impact to the project (severity of the risk). By using this matrix each of the identified risks will be properly ranked and the risks with the highest rank (risk that is very likely to appear and has the highest severity) will become top priorities for resolution. Finally, after all risks are classified and prioritized a risk mitigation strategy for each identified risk will be proposed.



RISK ASSESSMENT MATRIX

RISK RATING KEY	LOW	MEDIUM	HIGH	EXTREME
	0 ACCEPTABLE	1 ALARP (as low as reasonably practicable)	2 GENERALLY UNACCEPTABLE	3 INTOLERABLE
	OK TO PROCEED	TAKE MITIGATION EFFORTS	SEEK SUPPORT	PLACE EVENT ON HOLD

	SEVERITY			
	ACCEPTABLE LITTLE TO NO EFFECT ON EVENT	TOLERABLE EFFECTS ARE FELT, BUT NOT CRITICAL TO OUTCOME	UNDESIRABLE SERIOUS IMPACT TO THE COURSE OF ACTION AND OUTCOME	INTOLERABLE COULD RESULT IN DISASTER
IMPROBABLE RISK IS UNLIKELY TO OCCUR	LOW – 1 –	MEDIUM – 4 –	MEDIUM – 6 –	HIGH – 10 –
POSSIBLE RISK WILL LIKELY OCCUR	LOW – 2 –	MEDIUM – 5 –	HIGH – 8 –	EXTREME – 11 –
PROBABLE RISK WILL OCCUR	MEDIUM – 3 –	HIGH – 7 –	HIGH – 9 –	EXTREME – 12 –

Figure 4.1: Risk Assessment Matrix [74]

No.	Risk	Likelihood	Severity	Risk Score	Priority
R1	Change in requirements	Probable	Acceptable	Medium 3	**7**
R2	Unrealistic project schedule	Probable	Tolerable	High 7	**4**
R3	Problems with product evaluation due to bad time management or unavailable resources	Possible	Tolerable	Medium 5	**6**
R4	Problems with achieving high algorithm accuracy rate and low misclassification of legitimate websites due to inadequate features and rules	Possible	Undesirable	High 8	**3**
R5	Problems with obtaining data from reliable third party services	Possible	Tolerable	Medium 5	**5**
R6	Developing the wrong user interface for PhishEduPro platform	Improbable	Intolerable	High 10	**1**
R7	Problems with anti-phishing algorithm real-time performance	Probable	Undesirable	High 9	**2**
R8	Problem with acquiring sufficient technical knowledge to implement the product	Possible	Acceptable	Low 2	**9**
R9	Problem with deployment in production due to high cost of resources and complexity of the product implementation	Probable	Acceptable	Medium 3	**8**

Table 4.1: Project risks assessment and prioritization

4.2.1 Risk Mitigation Strategy

In this section of the report an effective and efficient risk mitigation strategy is developed. Risks with highest priority (1-highest, 9- lowest) are evaluated first and suitable resolution steps are proposed to lower or completely eliminate their impact to the project.

R6: Developing the wrong user interface for PhishEduPro platform

This risk is very improbable to occur, but if it occurs it might have a devastating impact to the project. Mitigation steps proposed to overcome this risk are:

A) Provide a functional user interface early in the development process, just after the first iteration is completed. A simple user interface that will show basic statistics of evaluated web pages should be offered to end users. By receiving continuous feedback from them we will be able to enhance it and improve it in the upcoming iterations.

B) Consult existing anti-phishing literature to accrue precise knowledge about which features to use and how to express them to effectively and efficiently educate and train users to recognize future phishing attacks.

R7: Problems with anti-phishing algorithm real-time performance

This risk is very likely to occur and its mitigation is crucial for the success of this project. Scanning websites in real-time, extracting features and calculating phishing scores is resource demanding and time consuming process. The website detection and reporting to end users should happen before they are tempted to provide their confidential information to the fraudulent website. By parallelizing the extraction process of anti-phishing features instead of sequentially processing them the execution time of the algorithm will be greatly improved. Additionally, scalability is another issue that should be address. By adding more resources like CPU power and RAM memory to production system, the overall performance of the system should be improved greatly and the time needed by the algorithm to classify a webpage will be within a few seconds.

R4: Problems with achieving high algorithm accuracy rate and low misclassification of legitimate websites due to inadequate features and rules

This risk is possible but very undesirable to occur, because achieving high accuracy rate and low false positive rate of the algorithm is one of the main objectives of this project. To mitigate this risk we should consult the literature and implement only those rules and features which have high impact on classification accuracy of the algorithm. Many features proposed by the literature are good for detecting phishing websites, but in the same time they yield high false positives. Additionally, by implementing early functionality of the algorithm we could measure its classification performance by using existing legitimate and phishing webpages. Thus, we should be able to see the impact of each feature to the classification process. Features and rules which do not yield very high accuracy could be further fine-tuned and features and rules which underperform should be omitted from the classification process.

R2: Unrealistic project schedule

This risk is common for all software development projects. To effectively mitigate it we should further split functional requirements into tasks and estimate and prioritize each task separately. The tasks with highest priority will be added to the early development iterations and the tasks with lower priority will be developed last. After each iteration, the remaining tasks not completed in the current iteration will be re-estimated. In case that the project is behind the schedule its scope should be reduced by removing low priority tasks.

R5: Problems with obtaining data from reliable third party services

This specific risk is probable to occur, but it quite tolerable because there are a few alternatives to consider. Many reliable third party services like WHOIS lookups and website ranking services are under commercial license and are very costly to be considered as potential data provides for PhishEduPro platform. The first step to consider is to contact them and explain to them that we are working on Master Thesis project which is not for commercial use and is of a great interest to online community. If a negative feedback have been received an open source solutions that offer WHOIS lookups and provide website ranking data for free should be considered. There are many open source providers that offer this type of data for free, but they are not as much reliable as commercial services and they enforce specific usage limitations. If open source solutions are the only option to consider we should allocate more time for tasks related to preprocessing and parsing of received data.

R3: Problems with product evaluation due to bad time management or unavailable resources

Product evaluation step is very important for comparing your product to existing solutions with similar approaches. The risk of not having proper evaluation is always present, but with good risk mitigation strategy is quite tolerable. In the worst case scenario when there is not much time left to do an extensive evaluation of your product, by comparing it to other alternatives, a self-evaluation might be a solution to the problem. It is fast and reliable and will add value to your project. Alternatively, if the comparison of your product to similar solutions is crucial for successful evaluation process as it is in our case, obtaining evaluation metrics data from similar products early in the project is considered as a good practice.

R1: Change in requirements

Change in requirements is a very common characteristic of modern software development projects. The likelihood that the requirements will change for the PhishEduPro platform is very high, because an evolutionary software development process is being followed. The design of the proposed anti-phishing algorithm will evolve with each iteration and it is not quite sure from the beginning, which anti-phishing features and rules will take part in final decision making process of the algorithm. Thus, the initial requirements to use specific rules and features might be completely eradicated. By receiving continuous feedback from end users and by looking in the evaluation metrics of the current implementation of the algorithm new requirements could be introduced. We embrace these changes in the requirements and we mitigate the risk to impact to the project, by following an agile methodology where all requirements should not be stated and elaborated upfront. Only those requirements which are specific to the current iteration of the product should be further assessed and expanded.

R9: Problem with deployment in production due to high cost of resources and complexity of the product implementation

Since the PhishEduPro platform will perform a real-time classification of suspected webpages it should be deployed on a central server to be accessed by it authorized end-users. This central server (or scalable cloud server) should be powerful enough to efficiently handle simultaneous classification of webpages submitted from various end-users. Various calls to third party APIs will be performed during the webpage evaluation process and additional web scrapers (crawlers) will be run by the platform to obtain HTML and JavaScript features from submitted webpages in real-time. These operations are time and resource consuming and are costly to be implemented and deployed. To mitigate the iteration, both the PhishEduPro platform and the newly proposed anti-phishing algorithm should be operable and should offer new functionalities to end-users.

During the first sprint a simple user interface to display a subset of anti-phishing features, related to websites evaluated in real-time by the anti-phishing algorithm, should be offered to end-user. Consequently, Domain Based Features extraction and evaluation process of the algorithm should be designed, implemented and tested during this phase. The extraction of Domain Based features is considered risky because reliable third party services that offer anti-phishing data for free are hard to be located on Internet.

During the second iteration graphical user interface and its anti-phishing algorithm is implemented by considering user feedback and measurements obtained after the first iteration. Functionality to search and view anti-phishing statistics for web pages which are already evaluated by the platform should be offered to end-user. Furthermore, Abnormal Based Features extraction and evaluation process of the algorithm is implemented. During this sprint a new approach to parallelize the feature extraction process utilized by the anti-phishing algorithm should be designed, implemented and tested in order to improve a s real-time performance.

During the third iteration a mechanism to authorize and authenticate is developed, therefore two user roles are identified: regular user and admin user. The process of extracting HTML and JavaScript related features from suspected web pages together with the process of extracting Address Bar features from webpage URLs should be designed, implemented and tested during this phase. The extraction of those features should be parallelized with the extraction process of the Domain Based and Abnormal Based Features in order to speed up the website evaluation process. The newly extracted website features together with their phishing scores will be shown to end-users by improving the graphical user interface. During this iteration the final set of features used in classification of legitimate and phishing websites is established and anti-phishing algorithm performance is continuously measured.

In the final sprint small improvements will be done to increase PhishEduPro accuracy rate. Moreover, graphical user interface should be finalized based on user feedback obtained after the third iteration. A new module for anti-phishing rules fine-tuning, another one for displaying anti-phishing algorithm evaluation metrics and a functionality to export a binary statistics for evaluated webpages in CSV format are designed, implemented and tested during this phase. After this iteration is completed the final version of PhishEduPro platform should be ready for deployment.

4.3.1 Project Management Tool

To perform an effective and efficient project management, a free project management tool is being used. A tool offered by GitHub is integrated with repository and it is used to track the development progress of PhishEduPro platform. The main dashboard of the tool consists of five custom made swimlanes labeled as . The dashboard is used to track the progress of project tasks. Each project task that is scheduled for development is translated into GitHub project issue Each newly created issue is automatically assigned to the Backlog where all issues are ordered by priority. Before the beginning of each sprint, a finite set of high priority issues are assigned . The swimlane represents a set of features which should be implemented in the current development iteration and just before the start of the sprint this list is frozen. When the implementation of a specific issue is being started the manually by the programmer. As soon as, the implementation of specific issue is finished and the issue is ready for testing it is manually placed to Finally, when a pull request is created by the programmer to merge a development branch into the master branch and in the comment field along with the issue number, the issue is automatically closed and placed into the swimlane. Figure B.1 from Appendix B shown a random intermediate state of the project management dashboard used to track the development progress of the PhishEduPro issues.

4.4 Development Environment

Choosing an optimal development environment setup to implement a brand new and innovative software project is not a straightforward task. Different aspects should be considered such as: the choice of programming language(s), data storage, the right Integrated Development Environment (IDE), third party tools and services, version control system etc. In the flowing section a detailed description of the PhishEduPro platform development environment is presented and a proper justification for each choice is given.

The main programing language used to implement the PhishEduPro platform is JavaScript (JS). More specifically a JavaScript runtime server along with one popular JS framework has been utilized. For implementing the backend functionality of the platform we make use of NodeJS. NodeJS is a JS enabled server-side platform built of top of the Google V8 engine and uses a non-blocking and event-driven I/O model which makes it lightweight and very efficient [75]. Consequently, to implement the graphical user interface of the platform another popular JS framework called AngularJS is being used. AngularJS [76] is a popular open-source JS framework which is implemented with Model View Controller (MVC) architecture in mind. It offers many built-in services and third party directives which facilitate frontend development, reduces development time and minimizes errors in programming.

There are 3 valid reasons to justify our choice to use JS as a primary programing language in the platform development process. First of all, to successfully analyze a webpage in real-time we need to interact heavily with its Domain Object Model (DOM). The DOM elements of the web page are written in HTML and are easily manipulated with JS functions. Thus, to extract HTML and JavaScript related features, webpage links, script tag and link tags from a webpage we need to write and launch a web crawler using JavaScript. Those crawlers should run in real-time and should be able to scrap and process anti-phishing features from suspected webpage. Furthermore, crawler results and results from third party services are usually returned as JSON objects, which notation is almost identical to JS object, and additional object serialization, deserialization and parsing by the platform is not required therefore the performance of the anti-phishing algorithm should not be affected. Additionally, a parallel execution of functions, used to obtain anti-phishing data originating from third party services, should be supported by the platform. This could be achieved by making use of the non-blocking I/O nature of NodeJS runtime where concurrent and asynchronous calls to third party APIs are easily accomplished. The last reason to choose JS for main programing language for the platform is because in the last 3 working on many commercial and open-source projects using this technology and I feel very familiar with it.

Since Ph persistent storage Mongo DB [77] is document oriented and stores objects using BSON format (a specialization of JSON format) the entire process of reading and writing data to the database should be very efficient. By utilizing web crawlers and third party APIs, the format of the data which is returned back to the platform is not unified. Therefore a relational database for storing and manipulating the data would not be an optimal solution. Moreover, the dataset of phishing and legitimate webpages is estimated to grow very fast when the platform is deployed to production. Document oriented databases like Mongo DB are very efficient is storing and manipulating large collections of documents with non-unified format and this is crucial for choosing document persistent storage over a relational database.

To efficiently store, share and manage the source code of PhishEduPro platform a distributed version control system (VCS) is being used. Code repository behind the PhishEduPro platform is GitHub [78], a free and reliable VCS which is based on open source Git protocol. Nowadays, the Git protocol is a standard tool for tracking source code changes and coordinating work among multiple developers. By utilizing a distributed version control system, source code integrity and availability is being achieved and repository changers are easily tracked. ory can be accessed by following this link. Finally, WebStorm a commercial JavaScript IDE [79] is being used for development of the PhishEduPro platform. WebStorm is considered the best

Chapter 5

5. System Architecture and Design

5.1 High-Level System Architecture

Designing extensible, scalable and maintainable systems is a quite challenging task. Many architectural choices should be considered upfront such as: efficient persistent storage, appropriate system architectural styles to achieve low coupling among internal and external modules, the choice of data representation and communication protocol established between the server and the clients and the choice of deployment environment. Figure 5.1 illustrates high-level system architecture of the PhishEduPro platform. In the following section a detail explanation and proper justifications regarding the proposed system architecture is discussed in details.

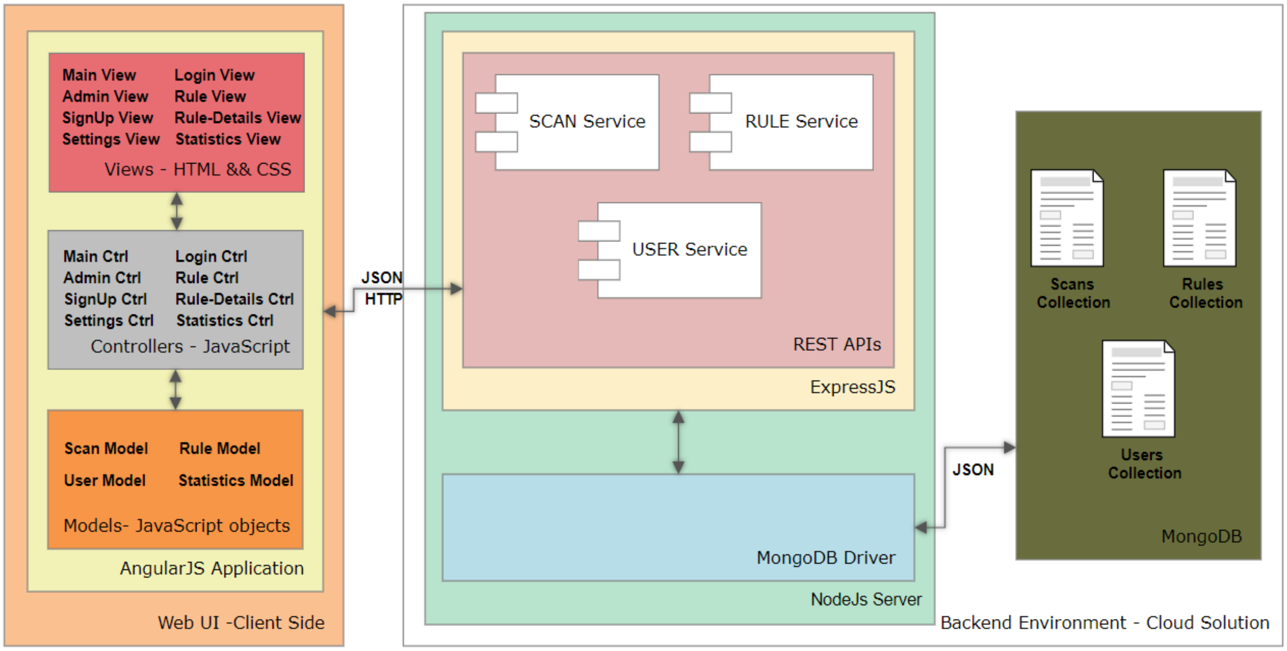


Figure 5.1: High-Level System Architecture Diagram of PhishEduPro Platform

One of the main product objectives stated in the first chapter of the report is to enable a free, efficient and reliable access to data and functions, offered by the platform - phishing algorithm, to third party products and services. Thus, the users which will utilize those product and services could feel the benefits of real-time anti-phishing protection. To accomplish flexibility and to decouple business logic from user interface it is envisioned to implement the main functionality of the platform as RESTful API web services. A RESTful API is a lightweight application program interface (API) that utilizes HTTP network requests and verbs like GET, POST, PUT and DELETE to provide unified access to distributed system resources [80]. A RESTful API design is based on the representational state transfer (REST) architectural style, proposed by Dr. Roy Thomas Fielding in his doctoral dissertation, which is used to guide the design and the development of modern web applications [80]. REST architectural style enforces secure and stateless communication between clients and the REST API interface therefore, it is very useful in cloud applications. Stateless communication means that each request from a client must contain all necessary information that is required by the server to understand the request and to return back the demanded resources. Thus, the server should not keep information from previous transaction to fulfill the current request of the client. Stateless nature of REST API web services promotes availability, scalability and reliability of PhishEduPro platform. Stateless components behave as independent units and they could be easily redeployed if a system failure occurs, or they could be replicated to accommodate increased network traffic. REST API components in PhishEduPro platform are implemented as loosely coupled web services. Each web service is an independent module that supports a set of related business goals and uses REST API interface to communicate with clients.

There are 3 different REST API services currently offered by the PhishEduPro platform. The SCAN service provides an implementation of a novel anti-phishing algorithm which offers real-time classification of suspected webpages. The RULE service provides functionality for fine-tuning anti-phishing rules and assigning weights to anti-phishing features used in the classification process. Finally, the USER service provides user management functionality to the PhishEduPro platform. Details about which RESTful routes (endpoint) are implemented by each of those web services are discussed in one the following chapters. Additionally, we envisioned one more utility service which is not offered via RESTful API web service therefore it is not shown in the system architecture diagram. This utility service is used to provide authentication and authorization of platform users and routes offered by the REST APIs.

The REST API web services shown in the diagram run in the context of ExpressJS app. ExpressJS is to NodeJS as Ruby on Rails is to Ruby. ExpressJS is a minimal and very flexible NodeJS application framework that is used to facilitate the implementation of backend system functionality by providing a robust set of features, utility methods and middleware functions [81]. Middleware functions are utility functions that has access to HTTP request and response objects, thus they are very useful in making changes to those objects each and every time it is required. ExpressJS operates perfectly in NodeJS ecosystem and it is compatible with many third party modules and services like database drivers, object relational mappers (ORM) and many others. In the context of NodeJS runtime environment an additional module Mongo DB driver is used to facilitate the operation of the PhishEduPro platform. The role of the Mongo DB driver is to enable flawless and secure communication with the Mongo DB database. Mongo DB database is a document oriented persistence storage which uses JSON to communicate with NodeJS app and stores those JSON objects using BSON (a specialization of JSON) format in Mongo collections. There are three different document collections contained in our Mongo DB database. The Scan collection is used to store Scan documents (JSON objects containing anti-phishing statistics of unique web pages). The Rules collection stores anti-phishing related information and User collection stores documents (objects) describing users in the system. The reasons why we choose to utilize Mongo DB as data storage over relational database is explained in section 4.4. A Mongoose ORM which runs on top of the ExpressJS app is used to do the object relational mapping from BSON to JSON between DB database. The entire process of reading and writing data to the database is very efficient and additional object serialization and deserialization is not required. By utilizing the REST architectural style we manage to decuple the business logic from pl - phishing algorithm is offered as a service to third party products and tools.

5.1.1 Client-Side App Architecture

The client side of the platform is designed as AngularJS application. AngularJS applications are based on the Module-View-Controller architectural pattern [76]. AngularJS MVC pattern defines the relationships among the components where each component has a predefined role. A Model component contains the data that is displayed by the View. AngularJS supports two-way data binding between the Model and the View. If the model is changed the change is reflected in the View automatically. Consequently, if the View is change the Model is updated with the change in real-time. This is considered as the biggest advantage of AngularJS over JQuery and pure JavaScript. Furthermore, the data to the Model is provided from form input fields or from Angular controllers and services (factories) which interact with external REST APIs, in our case with the API of the PhishEduPro platform. The View is aware of the Model and it is responsible to display the data contained in the Model and to invoke actions (methods) bind to it. The job of the Controller component is to create and populate the Model. Furthermore, the Controller is responsible to hand over the Model to the View and to get back the data from the View and provide it to the Model. Hence, the Controller has to be aware for both the Model and the View. To achieve this, the Controller makes use of various AngularJS services. AngularJS provides a mechanism called Dependency Injection to inject services into Controllers. This is considered as a great advantage of the framework since it helps developers to achieve better functionality with less coding [76]. AngularJS framework extends the HTML syntax, thus it offers better and more functional user interface. A drawback of AngularJS apps is that they perform poorly under heavy data loads. In the case of PhishEduPro platform when the dataset of evaluated webpages increase over 10 000 webpages the graphical user interface might freezes if those webpages are loaded simultaneously. A solution to this problem is to implement a design pattern called lazy loading to defer the initialization of objects until it is needed, thus to load only a limited subset of evaluated web pages on each user request.

The rightmost container of the diagram shown in Figure 5.1 illustrates all components which were design to implement the frontend of PhishEduPro platform. The data exchange HTTP network requests where data is represented is JSON format. This setup allows different technologies to be utilized for implementing the client as soon as the client app uses HTTP and JSON to communicate to the REST API provided by the platform. The client implementation is not limited to AngularJS app it could be Android, IOS app, Desktop app or Windows phone application. The proposed system architecture is considered reliable and efficient, because a unified JSON format of the data is kept from the moment when the user input is provided to the moment when the data is stored at the Mongo DB and vice versa, without a need of additional parsing, serialization and deserialization.

5.2 Assumptions and Limitations of the System

To develop a simple yet functional software product some assumptions and limitation should be stated upfront. These design constraints will facilitate fast product development and should be an important guideline how this product might be improved in the future. The detailed description of the assumptions and limitations specified about PhishEduPro platform is presented in the following bullet list.

- It is envisioned that the platform will support only real-time classification of web pages. Therefore a reliable internet connection both on client and server side is assumed. The limitation of the PhishEduPro platform is that it could not work offline, thus it could not be accessed by users that do not have access to Internet. This is not a big problem, because the users who are offline are not under a threat of webpage oriented phishing attacks.

- It is assumed that the PhishEduPro will manage to classify suspected webpages as phishing or legitimate if and only if the webpage is accessible to the Scan service. If the webpage is not accessible to the anti-phishing algorithm in real-time, because of any reason such as: the webpage is currently offline, the webpage blocks the access by IP address range, the webpage block access of web crawlers etc. than the platform will not be able to classify this webpage. In case that the webpage is not available to the Scan service an error message is shown to the user that the classification is not possible at the moment.

- During the design of the PhishEduPro platform a set of trustworthy third party service provides were selected to obtain anti-phishing data used in the webpage classification process. Services that provide information like: WHOIS lookup data, Page and Domain ranking estimations, domain SSL certificate information, blacklists, libraries for writing and launching web crawlers etc. are called by the Scan service. If any of these services suddenly become unavailable the anti-phishing algorithm will halt and the user will be informed that the specific webpage could not be analyzed at the moment, together with the reason why this has happened. Calculating a phishing score of a suspected webpage if some data is not available may bring more harm than good to end-user, because the estimation will not be based on the complete set of anti-phishing features. For instance, if the SSL certificates check service is unavailable, and its score is not added to the final anti-phishing score calculated for a webpage, a wrong conclusion might be deduced that the currently evaluated webpage is legitimate, but it is not and vice versa. The SSL anti-phishing feature has the biggest impact in calculation of the webpage phishing score, nevertheless most of the features obtain via third party services have a huge impact on the classification process.

- When the design of user roles and roles authorization mechanism has been proposed a few assumption were made. One of the assumptions is that only users with admin roles could see and operate with the module for managing anti-phishing rules and assigning weights to anti-phishing features. Updating the thresholds of anti-phishing rules and weights of anti-phishing features are key ingredients in calculating a phishing score of suspected webpages. Therefore, if a certain change of those thresholds occur it will be reflected in the phishing score of the web pages submitted by all users of the platform. By limiting the accesses to Rules service to admin users only, the correctness of the will be assured and an abuse of the system will be prevented. An alternative approach might be to offer user-centered management of functionality provided by the Rules service, thus each user will be able to set its own feature weights and rules thresholds.

- The change of thresholds of anti-phishing rules and updating the weights of anti-phishing features should not have an effect on already evaluated web pages. Only newly scanned webpages will be evaluated using the updated set of rules and features. Phishing score recalculation is not applied to previously classified webpages, since it is considered as a very resource intensive and fault intolerant operation.

- New anti-phishing rules and features could not be added to the platform dynamically. Only existing rules and features could be updated and fine-tuned using the logic behind the new rules and extracting their related anti-phishing features from web pages. However, our design is very flexible and offers an opportunity to add new rules and features, to improve the classification process of the anti-phishing algorithm, with minimal programming effort. Basically, a few changes in the code should be done to implement a new rule in the system and those steps are identical for adding one or hundred new rules and features.

5.3 Design of the REST APIs Endpoints

The design of the business logic of PhishEduPro platform is envisioned as a RESTful API interface. Three loosely coupled API services are designed: Scan Service, Rule Service and User Service. Each of those REST API services is structured under the same design pattern where its main components are: router, controller and model. The entry point in a REST service is the router. The router consists of endpoints offered by the REST service to route HTTP client requests to the demanded resources. A resource is a key abstraction of information in RESTful architecture and a resource can be anything like text, object, image, sound, video or a file. Moreover, the router is responsible to secure those REST API endpoints by authorizing user roles. The security of each endpoint is enabled by utilizing an Authentication service offered by the PhishEduPro platform. Client authentication and authorization is done by generating a Bearer token on each successful login request. The token is stored in a session cookie of the browser therefore each user can gain access to the REST API during the current login session.

Controllers are responsible for implementing the business logic offered by the REST API Services of PhishEduPro platform. The functionality provided by each REST endpoint is implemented with a specific method defined in the controller. The naming of those methods is unified among all REST API web services unified name of a method which returns an array of JSON resources when a GET HTTPS request is being sent by the client. The last important component used in the design of PhishEduPro platform REST services is the model. The model is used to create, delete, update and get objects to and from a database and to make them accessible to the controller. Each service provided by the platform has its own model with a predefined structure. A model utilizes a model schema to validate the attributes of each object by employing some restrictions to the object. A validation rules such as: required fields, length of string attributes, minimum and maximum value of a number and some custom validation rules are define with the model schema. If a specific JSON object does not correspond to its model schema it is discarded and a validation message is returned back to the client. Thus, the model schema enforces a standard object format and reduces errors raised during reading and writing to the database. A detailed implementation of the endpoints designed for the Scan Service, Rule Service and User Service is presented in chapter 6 with tables 6.1, 6.2 and 6.3 respectively.

5.4 Design of the Anti-Phishing Algorithm

As it is stated in the aim of this project, the core functionality offered by PhishEduPro platform is to perform a real-time detection of phishing and legitimate web pages. To achieve this goal we designed a novel anti-phishing algorithm that is capable to detect zero-day phishing attacks with high accuracy rate. Consequently, the algorithm should perform very well in classification of legitimate web pages.

The design of the anti-phishing algorithms should be elaborated in two stages. In the first stage, a set of anti-phishing features that have a deep impact to a website classification process is isolated. Moreover, a set of rules bind to those features is constructed. Each feature used in the process of website classification might yield different value depending if a page is a phish or it is legitimate. Later, the rules are used to check whether the value bind to a particular feature is above or below a specific threshold. Thus, a feature is evaluated as phishing or legitimate depending on its calculated value. Section 5.4.1 presents all 22 features which are
separately, and its impact on the webpage overall phishing score is weighted. The second stage continues with the actual evaluation process of the anti-phishing algorithm. At this stage, the process of evaluating a webpage in real-time by utilizing the proposed anti-phishing features and rules, is clarified and a function used to calculate the final phishing score of a suspected webpage is presented. A detailed workflow diagram illustrating the classification process is presented in section 5.4.2.

5.4.1 Design of the Anti-Phishing Features and Rules

There are many different anti-phishing features proposed by the research community. Not all of them have the same impact in web page classification. Few of them perform very well in detection of phishing webpages. Consequently, a few other features yield good results of detecting legitimate webpages. A common drawback of most anti-phishing features is that they might raise outliers while classifying web pages. Basnet at al. conducted a study [82] to measure the impact of a set of anti-phishing features and rules in detecting both legitimate and phishing webpages. They found out that if a webpage and its domain is not indexed by the top search engines, like Google, Yahoo, Bing and Baidu, than it is very probable that it is a phishing webpage. On the other hand, if the website is already indexed than the chances to be legitimate are almost 100%. According to their study this rule was able to classify correctly more than 97% of phishing and legitimate web pages [82]. Another rule that was tested was the presence of [-, _, 0- characters in a webpage URL. Surprisingly, 66% of phishing web pages, but in the same time 20% of legitimate webpage confirm to this rule. This might raise some classification problems if the feature is not properly weighted by the anti-phishing algorithm.

Up until now, the best set of features and rules utilized in detection of phishing webpages is proposed by Mohammad at al. [71]. Their set of 30 anti-phishing features is widely accepted by the research community and it is integrated in many machine learning and heuristic based anti-phishing approaches. The features are organized in 4 distinct categories discussed in section 3.4. Not every feature contained in the proposed set has the same impact to the webpage classification process. An empirical study done by Thabtah and Abdelhamid showed that a small subsets of the proposed features, subset of 2, subset of 5 and subset of 9 features are very promising in classification of phishing and legitimate web pages [83]. They used Information Gain and Chi-square technique to measure the impact of each of the proposed features. Later, they fed those features as an input to PART, RIPPER and C4 machine learning algorithms and they measured their classification accuracy and error rate. By comparing the classification error rates of 2-feature, 5-feature and 9-feature datasets to the 30-feature dataset it was concluded that the error rate rose insignificantly, by 2.42%, 3.49% and 5.5% for each dataset respectively [83]. By leveraging this knowledge we chose wisely the set of features used in our novel anti-phishing algorithm. Consequently, we proposed an additional set of experimental features to further raise its classification accuracy. Each feature was properly weighed, with a number from 0 to 1, depending on its influence to the webpage classification process. The initial weights for already know features were obtained from study presented in [83] and the preliminary weights of the newly proposed features are calculated by finding the ghts. As we were continuously testing and improving our algorithm some of the proposed rules used to evaluate the anti-phishing features were updated. This update was influenced by the classification accuracy of the algorithm achieved during two months evaluation process. In the following subsection 5.4.1.1 are presented and explained the final set of 22 features that were used to build our novel rule-based anti-phishing algorithm.

5.4.1.1 Anti-Phishing Features and Rules Details

<u>Address Bar Based Features</u>

1. Using IP Address

If an IP address is used instead of a domain name there is a great probability that a web page is a phish. Sometimes the IP address is transformed into hexadecimal string to lure the victim [71].

IP Address example: http://125.98.3.123/phish.html
Hexadecimal address example: http://0x23.0xBB.0xCD.0x22/paypal /index.html

$$\textbf{Rule:} \begin{cases} \text{If the domain part of a URL has an IP Address } \rightarrow \text{ Phishing} \\ \text{Otherwise} \rightarrow \text{ Legitimate} \end{cases}$$

2. Long URL

Phishes usually craft long URLs. That way the suspicious part of the URL will stay out of sight to the victim, since it would not be shown in the visible part of the address bar. By reviewing their dataset Mohammad at al. were able to estimate the minimum length of a phishing URL [71].

$$\textbf{Rule:} \begin{cases} \text{URL length} < 54 \rightarrow \text{Legitimate} \\ \text{URL length} \geq 54 \ and \ \leq 75 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

3. Using URL Shortening Service

URL shortening is a common method used nowadays to create a small and tidy URL form long URL. The tiny URL redirects people to the same webpage pointed by the long URL. This method is very popular among Twitter users, because Tweeter restricts the size of a tweet to maximum 140 characters. Phish often use this method to hide the actual URL from the victim [71].

$$\textbf{Rule:} \begin{cases} \text{Shorten URL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

4. URL contains “@” symbol

symbol. Thus,
often use this symbol to lure their victims by using hidden redirects [71].

$$\textbf{Rule:} \begin{cases} \text{Url contains @ Symbol} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5. Adding a Dash (-) as Prefix or Suffix to the Domain

The dash symbol is rarely used in a domain name of a legitimate website. However, phishers often add it as a prefix or suffix to a popular domain name keyword to lure the victim that the presented webpage is legitimate [71]. An example of phishing domain using this feature is the following URL: ***http://confirme-paypal.com/***.

$$\textbf{Rule:} \begin{cases} \text{Dash } (-) \text{ Symbol in Domain Name} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

6. Using Subdomain and Multi Subdomains

Phishers often craft deceptive URLs with many subdomains to lure their victims to visit a phishing webpage. The number of subdomains is count when the top level domain (country code) and the (www.) subdomain are removed from the domain part of the URL. By counting the remaining dots in the domain part of the URL the number of subdomains is calculated [71].

$$\textbf{Rule:} \begin{cases} \text{Remaining Dots In Domain Part} = 0 \rightarrow \text{Legitimate} \\ \text{Remaining Dots In Domain Part} = 1 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

7. HTTPS (Hypertext Transfer Protocol with Secure Socket Layer (SSL))

The existence of HTTPS protocol is very important for securing transfer of data through the network. Many legitimate website are implementing HTTPS to increase the trust in their services. However, phishers follow this trend and they implement HTTPS in their websites too. Mohammad at al. in [71] suggested that the website SSL certificate should be validated by checking the chain of certificate issuers and the certificate age. We have extended this rule by adding a few more checks. Additionally, we check certificate level (Extended Validation EV, Organizational Validation OV and Domain Validation DV), we check if the certificate is self-signed or revoked and finally we check if the domain and its IP address are not blacklisted. Furthermore, by testing web pages with HTTPS using our anti-phishing algorithm we found out that the minimum age of SSL certificate of a legitimate webpage is 6 months if the level of certificate is DV. Google, Facebook, LinkedIn and many other popular webpages use EV certificates, because they provide maximum security. Famous browsers like Chrome, Firefox and Internet Explored show green padlock icon in the left corner of the address bar when the URL is pointing to a website implementing EV certificate.

Rule:

$$\begin{cases} \text{Use HTTPS , Issuer Is Trusted, Certi Chain is completed, Domain and IP is not blacklisted,} \\ \text{Cert is not revoked, Cert is not expired , Cert Level is DV and Age of Certificate} \geq \text{ 6 Months } \rightarrow \textbf{Legitimate}; \\ \text{Use HTTPS , Issuer Is Trusted, Certi Chain is completed, Cert is not revoked, Cert Level is EV or OV,} \\ \text{Domain and IP is not blacklisted, Cert is not expired } \rightarrow \textbf{Legitimate}; \\ \text{Using https and Issuer Is Not Trusted } \rightarrow \textbf{Suspicious} \\ \text{Otherwise} \rightarrow \textbf{Phishing} \end{cases}$$

8. Domain Registration Length

By reviewing their dataset of phishing and legitimate web pages Mohammad at al. reported that the longest living fraudulent domain has been registered for one year only [71]. On the other hand, legitimate domains are registered a few years in advance.

$$\textbf{Rule:} \begin{cases} \text{Expires on} \leq \text{ 364 days} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Abnormal Based Features

9. Request URL

This feature is used to check whether resources like images, videos and sounds are loaded from external domain or not [71]. Legitimate webpages usually load their resources within the same domain. In contrast, phishing webpages usually copy the HTML code of legitimate webpages, thus their resources are loaded from external domain (the domain of the legitimate webpage).

$$\textbf{Rule:}\begin{cases} \text{Request URL external domain } \% < 22\% \rightarrow \text{ Legitimate} \\ \text{Request URL extrnal domain } \% \geq 22\% \text{ and } 61\% \rightarrow \text{ Suspicious} \\ \text{Otherwise} \rightarrow \text{ Phishing} \end{cases}$$

10. Anchor URL

Webpage anchors are elements defined by the <a> tag. They are widely known as webpage links. Generally, the same rules are used as in Request URL, but a few changes are proposed in the process how each anchor URL is examined. We have proposed those changes after evaluating the classification results of the novel anti-phishing algorithm implemented by the PhishEduPro platform.

$$\begin{cases} \text{Links in " < Script > " and " < Link>" from external domain \% < 17\%} \rightarrow \text{Legitimate} \\ \text{Links in " < Script > " and " < Link>" from external domain \% } \geq 17\% \text{ And } \leq 81\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

12. Server Form Handler (SFH)

SFH is the action attribute declared within a HTML form. SFHs that contain an empty string (actio = , because the form destination submission is unknown to the user [71]. Furthermore, if a domain part of a link contained in the SFH is different from a domain of the evaluated webpage than the webpage is considered suspicious. Usually, HTML forms submit their inputs to a form handler within the same domain.

$$\textbf{Rule:} \begin{cases} \text{SFH is "about: blank" or is empty} \rightarrow \text{Phishing} \\ \text{SFH points to a different domain} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

<u>HTML and JavaScript Based Features</u>

13. IFrame redirection

<iframe> is a HTML tag which could be utilized to nest a webpage within the HTML content of another webpage. Phishers often exploit this technique to nest a phishing [71]. Thus, the end-users e that there are 2 different web pages show on the screen, and they are easily lured to provide their confidential information to the phisher.

$$\textbf{Rule:} \begin{cases} \text{Using iframe} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

14. Using Input Fields (Password, Text, Email, Tel)

Phishers usually lure their victims to disclose their personal information like: emails, passwords, credit card numbers, and phone numbers so they can take advantage of it. It is always useful to check if users are asked to provide their personal information to a website that does not run under the HTTPS protocol. Since the transfer of the data in those websites is not encrypted the data could be easily eavesdropped by the phisher. This rule was proposed after looking at the classification statistics of our novel anti-phishing algorithm. More than 98% of the phishing webpages had at least 1 input field displayed on the screen.

$$\textbf{Rule:} \begin{cases} \text{Password Filed and not valid HTTPS} \rightarrow \text{Phishing} \\ \text{Text, Email, Tel Field and not valid HTTPS} \rightarrow \text{Suspicious} \\ \text{Otherwice Legitimate} \end{cases}$$

Domain Based Features

15. Age of Domain

Age of a domain is feature that can be extracted from a WHOIS database. It is a fact that phishing webpages live for a short period of time, from few hours to a few days until they have been taken down by authorities [71]. Mohammad at all reported that a minimum age of legitimate domain in 1 year. However by reviewing the webpage anti-phishing statistics product by PhishEduPro anti-phishing algorithm, we found out that the minimum age of legitimate domains in our dataset is 6 months.

$$\textbf{Rule:} \begin{cases} \text{Age Of Domain} \geq 180 \text{ days} \rightarrow \text{Legitimate} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

16. Website Traffic (Alexa Ranking)

Alexa measures the popularity of a website by assessing the number of visitors that accessed it and the number of webpages they visited under the same domain [84][71]. Phishing websites are short-lived and they might not be index by Alexa crawlers. Mohammad at al. reported that a website is considered legitimate if it is ranked among the top 100 000 by Alexa service. In case that the rank of a website is higher than 100 000 the website is labeled as suspicious. Website is labeled as a phish if it is not ranked at all by Alexa service. We found out that in 2017 there are much more active websites compared to 2013 and that the suspicious website rank threshold should be raised to 500 000 instead of 100 000.

$$\textbf{Rule:} \begin{cases} \text{Alexa Rank} < 500{,}000 \rightarrow \text{Legitimate} \\ \text{Alexa Rank} \geq 500{,}000 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phish} \end{cases}$$

17. MozRank Mozscape

Mozscape is a SEO consulting company that offers both a commercial and a limited free service capable to measure the value and ranking potential of websites, by utilizing a set of intelligent metrics to calculate a diverse set of website scores [85]. Mozscape maintains a database of more than 188 Billion URLs currently operating on the Web. By testing the free web service offered by Mozscape, we have found that some of the scores calculated for webpages and their domains could be utilized as features for our novel anti-phishing algorithm. One of those features is MozRank. MozRank is used to calculate link popularity active. Web pages earn MozRank based on how many different pages on the Web links to them and what is their current MozRank. The higher the MozRank of the linking webpages, the base page will increase its MozRank faster. MozRank is calculated using machine learning model and it is based on logarithmic scale from 1 to 10 [85]. For example, it is quite easy to increase the MozRank from 1 to 2, but it takes much more time and effort to raise it from 4 to 5. Since, MozRank webpage score is scaled identically score we could utilize the same rule proposed by Mohammad at al. in [71].

$$\textbf{Rule:} \begin{cases} \text{MozRank} < 0.2 \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

18. Domain Authority (DA)

Domain Authority (DA) is search engine ranking score that predicts how well a Domain will rank on search engine ranking pages (SERP) offered by top rated search engines like Google, Bing and Yahoo [85]. This score is developed by Mozscape using a machine learning model and it ranges from 1 (lowest) to 100 (highest). It is scaled on logarithmic scale, and similar to MozRank, it is much easier to increase this score from 1 to 10 than from 50 to 60.

$$\textbf{Rule:} \begin{cases} \text{DA} \geq 17 \rightarrow \text{Legitimate} \\ \text{DA} \geq 8 \text{ AND} < 17 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

19. Page Authority (PA)

Similarly to Domain Authority, Page Authority is a score developed my Mozscape that predicts how well a specific webpage will rank on top rated search engines [85]. PA score ranges from 1 to 100 on a logarithmic scale, where higher PA score indicates that a webpage will rank higher on search engine result pages. Compared to DA, PA score is much harder to increase it, because each webpage on a domain is ranked independently.

$$\textbf{Rule:} \begin{cases} \text{PA} \geq 5 \rightarrow \text{Legitimate} \\ \text{DA} \geq 2 \text{ AND} < 5 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

20. External Links Pointing to a Webpage

To obtain this metric we utilize the Mozscape service. This metric indicates how much external links are pointing to a specific webpage. Page is considered more trustful and legitimate if a higher number of external links are pointing toward it. Mohammad at al. reported that 98% of phishing websites contained in their dataset did not have any links pointing towards them [71].

$$\textbf{Rule: }\begin{cases} \text{Number of external links} = 0 \ \rightarrow \ \text{Phishing} \\ \text{Number of external links} > 0 \text{ and} \leq 2 \rightarrow \text{Suspicious} \\ \text{Otherwise} \ \rightarrow \ \text{Legitimate} \end{cases}$$

21. Statistical Report Based Features

The PhishTank website [58], offers numerous statistical reports regarding phishing webpages on monthly and quarterly basis. To improve the classification accuracy of the PhishEduPro anti-phishing algorithm, we utilized 2 forms of the top ten statistics from = , according to statistical reports published during 2017th. From those statistical reports we have designed 2 internal blacklists (blacklist of phishing domains and a blacklist of phishing IP addresses) to be consumed by the PhishEduPro anti-phishing algorithm.

$$\textbf{Rule: }\begin{cases} \text{Host belongs to any of the 2 internal blacklists} \ \rightarrow \ \text{Phishing} \\ \text{Otherwise} \ \rightarrow \ \text{Legitimate} \end{cases}$$

22. Web of Trust (WOT)

Web of Trust (WOT) is a unique and patented crowd sourced website reputation and review system that helps uses to identify different types of malicious websites in real-time [86]. WOT service enables users to rate and comment on websites to express their personal experience about the content therefore WOT service is very efficient in detecting threats that only human eye can register such as phishing scam, fake news and illegal information. This service constantly monitors user behavior to make sure that the website ratings are trustworthy, accurate and constantly updated. Furthermore, WOT service utilize third party services to validate user ratings. An example of such service is a blacklist of phishing websites. WOT service extends its functionality by implementing a patented system based on machine learning to predict fraudulent/phishing websites, despite the fact they are not yet rated. An accurate prediction is achieved by constantly tracking referral sites that drive user to fraudulent web pages. Additionally, WOT service provide ratings and reputation scores on webpage level by tracking and evaluating their URLs. This service is available as a browser plugin which is installed by more than 140 Million end-users worldwide [86]. The plugin shows a traffic light next to each search result link displayed by the top rated search engines. This traffic light estimates the reputation and the rating of those links in real-time while the user is browsing the Web. WOT traffic lights are also visible next to links displayed on social network websites like Facebook, Twitter and LinkedIn. Finally, WOT service offers a free and reliable REST API, to provide access to reputations and ratings of billions webpages in real-time. A web page reputation score ranges from 0 to 100, where a higher score corresponds to higher reputation of the web page [86]. Consuming the REST API provided by WOT service, is a very convenient way to use webpage reputation and rating data as a feature for the PhishEduPro anti-phishing algorithm.

$$\textbf{Rule:} \begin{cases} \text{WOT reputation score} < 40 \rightarrow \text{Phishing} \\ \text{WOT reputation score} \geq 40 \text{ and} < 60 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.4.1.2 Calculation of the Final Phishing Score

Once the features and rules utilized by PhishEduPro anti-phishing algorithms have been designed a strategy to weight their impact to classification process should be proposed. As it was mentioned in section 5.4.1 the initial weights for existing features were obtained from study presented in [83] and the preliminary weights of the newly proposed features are calculated by finding the average of known features weights. To calculate the final phishing score for a suspected webpage, we sum the weights of webpage features evaluated as phishing. If a specific feature is evaluated as suspicious we sum up only 50% of its weight. If any of the features from the set of 22 features utilized by the algorithm is evaluated as legitimate, than we remove it weight from the web page phishing score calculation. Thus, the final phishing score for a suspected webpage is the sum of the weights of features evaluated as phishing or suspicious. Furthermore, the final score is than compared to a phishing threshold and the webpage is assigned with an appropriate label. According to the phishing score percentage value, 5 different labels could be assigned to a suspected webpage such as: very legitimate (score range from 0% to 21%), legitimate (score range from 22% to 41%), fair (score range from 42% to 51%), very suspicious (score range from 52% to 61%) and phishing (score range from 62% to 100%). If a binary classification is used than only two labels are possible: legitimate (score range from 0% to 51%) and phishing (score range from 52% to 100%). Binary scoring is very useful in the evaluation process of the PhishEduPro anti-phishing algorithm where the performance of the algorithm is compared to third party anti-phishing approaches. Table 5.1 sum ups the set of 22 anti-phishing features used in the classification process of PhishEduPro anti-phishing algorithm, it presents their initial estimated weights and shows their origin (existing or new).

Anti-Phishing Feature	Initial Weight	Origin
HTTPS	0.499	Mohammad at al. [71] with updated rules
Anchor URL	0.477	Mohammad at al. [71] with updated rules
Adding a Dash (-) as Prefix or Suffix to the Domain	0.123	Mohammad at al. [71]
Website Traffic (Alexa Ranking)	0.1145	Mohammad at al. [71] with updated rules
Using Subdomain and Multi Sub Domains	0.109	Mohammad at al. [71]
Using Input Fields (Password, Text, Email, Tel)	0.0847	New Feature
Domain Authority (DA)	0.0847	New Feature
Page Authority (PA)	0.0847	New Feature
Web of Trust (WOT)	0.0847	New Feature
Links in <Script> and <Link> tags	0.047	Mohammad at al. [71]
Request URL	0.046	Mohammad at al. [71]
Server Form Handler (SFH)	0.037	Mohammad at al. [71]
Domain Registration Length	0.036	Mohammad at al. [71]
Age of Domain	0.01	Mohammad at al. [71] with updated rules
MozRank Mozscape	0.008	New Features evaluated using old rules, proposed by Mohammad at al. [71]
Using IP Address	0.006	Mohammad at al. [71]
External Links Pointing to a Webpage	0.004	Mohammad at al. [71]
Statistical Report Based Features	0.004	Mohammad at al. [71]
Long URL	0.003	Mohammad at al. [71]
Using URL Shortening Service	0.003	Mohammad at al. [71]
URL contains "@" symbol	0.002	Mohammad at al. [71]
IFrame redirection	0.0001	Mohammad at al. [71]

Table 5.1: Anti-phishing features ordered by their impact to webpage classification process

5.4.2 Detailed Workflow of the Anti-Phishing Algorithm

The high-level workflow diagram of PhishEduPro anti-phishing algorithm is presented in section 3.4. In this section a detailed workflow of the algorithm is presented and the main design choices are elaborated in details. Each design decision is taken to improve the overall performance of the algorithm and to make it suitable for real-time anti-phishing detection.

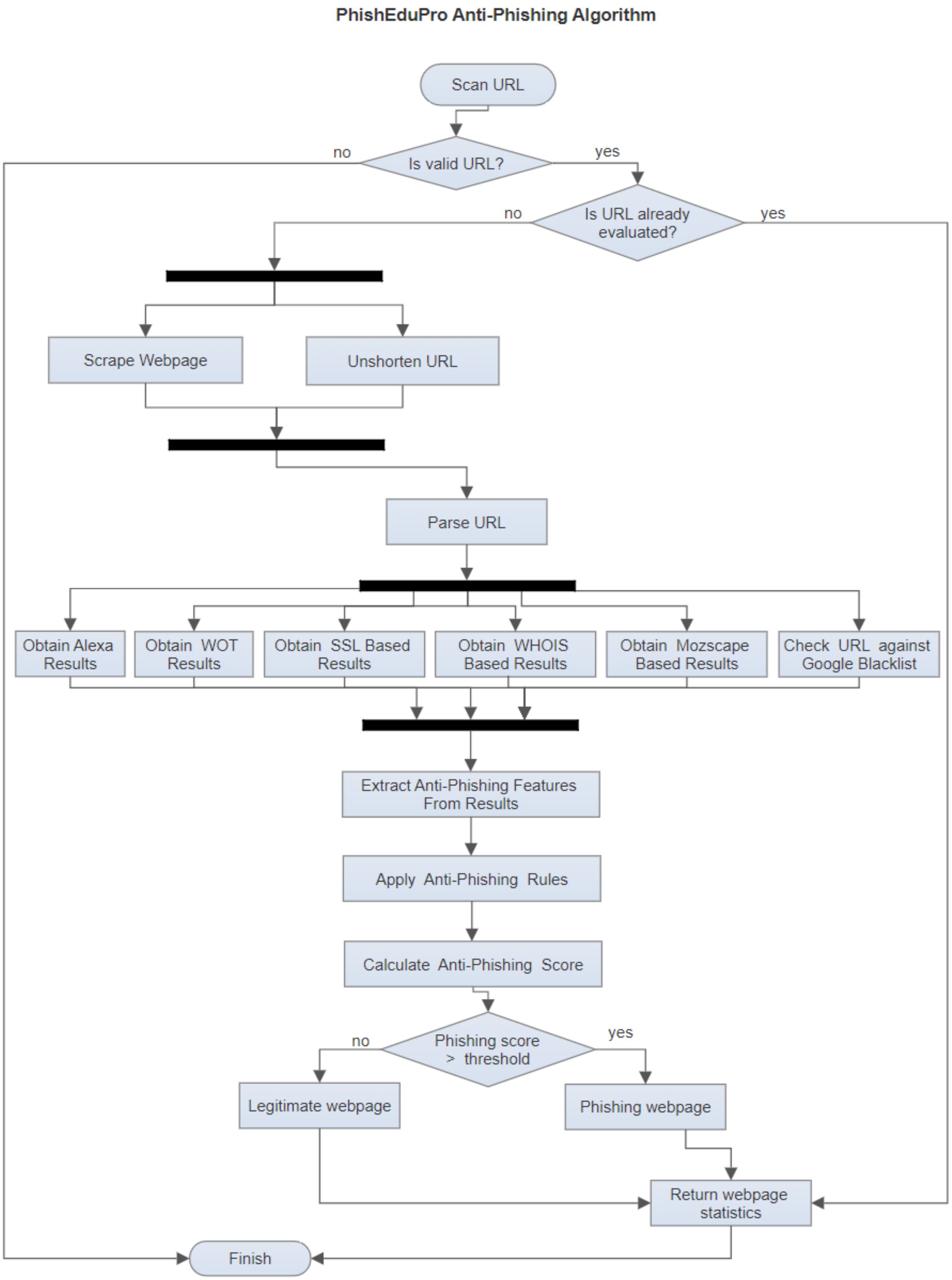

PhishEduPro Anti-Phishing Algorithm
Scan URL
Is valid URL?
no
yes
Is URL already evaluated?
no
yes
Scrape Webpage
Unshorten URL
Parse URL
Obtain Alexa Results
Obtain WOT Results
Obtain SSL Based Results
Obtain WHOIS Based Results
Obtain Mozscape Based Results
Check URL against Google Blacklist
Extract Anti-Phishing Features From Results
Apply Anti-Phishing Rules
Calculate Anti-Phishing Score
Phishing score > threshold
no
yes
Legitimate webpage
Phishing webpage
Return webpage statistics
Finish

Figure 5.2: Detailed workflow of PhishEduPro anti-phishing algorithm

Figure 5.2 illustrates the detail workflow of the PhishEduPro anti-phishing algorithm. Since the basic workflow of the algorithm is elaborated in details in section 3.4, the main focus of this section is to justify some low level design decisions that were made to improve the algorithm performance. The blocks of thick and black horizontal lines shown in the diagram 5.2 represent fork and join nodes. They are used to illustrate concurrent operations performed by the algorithm. By looking at the diagram it is easy to spot the first concurrent block. Actions like webpage scraping and URL unshortening are concurrently executed. Website scraping (crawling) is an action performed by the algorithm to extract HTML objects from webpage content in real-time. Objects like, anchor URLs, script URLs, SFH, iframe, address bar URL, input and password fields are extracted by using this technique. Many websites have a defense mechanism to prevent crawling of their content. Therefore we have envisioned a 2 second delay before the web crawler starts to scrap the content of a suspected webpage. Before we introduced a delay many web scraping attempts were unsuccessful, because web pages blocked the crawlers. By introducing a 2 second delay the website is tricked to view the crawler as a regular user and that is the main reason of having a 98% of crawling success achieved by the PhishEduPro anti-phishing algorithm. Website scraping is a resource intensive and time consuming action. Since this action in performed in the context of the web browser, the main thread of our system is available to perform different action while waiting for the results of the web crawler. Therefore the URL Unshortening action is performed concurrently. Thus, PhishEduPro algorithm performance is improved and its execution time is significantly reduced.

Many people could ask themselves why there are two separate concurrent blocks, but not one only. The reason is simple to understand if you know the anatomy of phishing attacks and the behavior of experienced phishers. Phishers rarely expose the final address of their phishing website. They tend to utilize URL shortening service like the one provided by Google (goo.gl) and Bitly (bitly.com) to hide the actual phishing URL behind a tiny URL. When people click on a tiny URL provided by the phisher, they are redirected to the phishing website without noticing. Alternatively, phishers implements different kind of redirects where the user is redirected more than 2 times before landing on the phishing websites. Those redirected are easily caught by web crawlers. Since the web crawler we design to scrap specific HTML content from a webpage, implements 2 seconds delay before crawling the website, the final phishing page URL is always scrap from the browser address bar and provided back to the algorithm. After completion of concurrent block 1 the PhishEduPro algorithm always perform parsing of the primary phishing URL. By obtaining the primary URL of a phishing webpage our algorithm is able to match many secondary URL to a common phishing source. The second concurrent block shown in the diagram 5.2 is designed to run asynchronous HTTP requests to third party API services which provide additional anti-phishing features for the suspected webpage. This is achieved by making use of the non-blocking I/O nature of NodeJS runtime where concurrent and asynchronous calls to third party APIs are easily accomplished by utilizing native NodeJS packages. The reason of doing this is to improve algorithm performance by reducing the total execution time of the web page evaluation process. The rest of the actions in the algorithm workflow are run sequentially using the main thread of the system.

Chapter 6

6. Implementation, Testing and Deployment

6.1 Implementation of the Platform’s REST API Endpoints

As it was mentioned earlier in the report, the implementation of the business logic of PhishEduPro platform is envisioned as a RESTful API. Each of the proposed RESTful API web services the Scan, the Rule and the User are implemented by following the same design pattern. The main components of each web service are the web service endpoints. Consequently, their authorization and authentication and the HTTP request types supported by each of those endpoints are very important aspects of the implementation process. Implementing reliable, easy to use and secure REST API web services to share data with PhishEduPro platform end-users is an imperative. A detailed description of the endpoints implemented for the Scan Service, Rule Service and User Service is presented with tables 6.1, 6.2 and 6.3 respectively.

Endpoint	Request Type	Authorized User Roles	Description
/api/scans	GET	ADMIN, USER	This RESTful endpoint returns an array of already evaluated webpages. Each webpage is represented as a JSON object having as attributes its anti-phishing statistics generated by the anti-phishing algorithm. This route supports filtering of returned results. End-uses are allowed to use query strings additionally filter webpages return with this API call.
/api/scans	POST	ADMIN, USER	This endpoint is used to submit a webpage URL, in the body of the request, to be evaluated by the anti-phishing algorithm. When the algorithm is finished it returns a JSON object containing the anti-phishing statistics of the submitted webpage.
/api/scans/:id	GET	ADMIN, USER	This REST route returns the anti-phishing statistics for a single webpage evaluated by the algorithm. The =id parameter in the URL is replaced by the object id of the suspected webpage.
/api/scans/:id	PUT	ADMIN	This endpoint is used for updating the anti-phishing statistics of a webpage which is already evaluated by the algorithm. It is accessible only to the admin users, but we do not expose it to graphical user interface. A non-functional requirement of the PhishEduPro platform is that each webpage statistics should stay immutable once it was generated.
/api/scans/:id	DELETE	ADMIN	This endpoint is used for deleting a specific webpage statistics. The =id parameter in the URL is replaced by the object id of the suspected webpage. This route is only

			accessible by admin user and it is rarely invoked. This endpoint is not exposed to the fronted.
/api/scans/export	GET	OPEN ACCESS	This endpoint is used to generate and export a CSV file containing a list of all evaluated webpages (phishing and legitimate). Each row in the file contains binary values of all evaluated webpage features. **1: legitimate, -1: phishing,** while some of the features are ternary and has an additional value **0: suspicious.** This route is accessible by everyone therefore having an account on the platform is not required. We opened this route to provide the research community with an unlimited access to a frequently updated anti-phishing dataset. Generated dataset could be used further to train and test an innovative anti-phishing machine learning algorithms. This was one of the main product objectives defined in Chapter 1.
/api/scans/stats	GET	ADMIN, USER	This endpoint is used to generate an evaluation metrics used to measure the performance of the anti-phishing algorithm. It returns a JSON object containing information such as: rate, precision, recall etc. Those metrics are used to inform end- current setup. Additionally, it is a very convenient tool for admin user to understand when further fine-tuning of s required.

Table 6.1: Implementation of SCAN Service API endpoints

Endpoint	Request Type	Authorized User Roles	Description
/api/rules	GET	ADMIN, USER	This route returns JSON array of existing rules utilized in the web page classification process. Each rule is represented as a JSON object with the following attributes: name, description, rule code (a unique short name for each webpage anti-phishing feature), weight (the impact of the rule related anti-phishing webpage feature to the final phishing score), rule phishing threshold, rule suspicious threshold, unit (represented with days, count, length etc. describing the anti-phishing webpage feature measurement unit) and rule status (active or inactive). This endpoint is available both to admin and regular users. The rules are listed on the main screen of the platform to show the evaluation statistics of each webpage classified by the anti-phishing algorithm. By viewing the webpage anti-phishing statistics, end-users are educated how to recognize and mitigate future fishing attacks.
/api/rules	POST	ADMIN	This endpoint is used to create new anti-phishing rules, but it is not available at the graphical user interface. New rules cannot be added to the system dynamically using input forms. More about this limitation of the system is discussed in section 5.2. Nevertheless this API endpoint is utilized in the seed file to populate existing rules into the Mongo database.
/api/rules/:id	GET	ADMIN	This REST route returns a single anti-phishing rule object stored in the database. The d in the URL is replaced by

			the object id of the requested anti-phishing rule.
/api/rules/:id	PUT	ADMIN	This endpoint is used for updating the attributes of an existing anti-phishing rule. It is accessible only to admin users and it is utilized in the implementation of Rule Management module offered to admin users by the AngularJS PhishEduPro app.
/api/rules/:id	DELETE	ADMIN	This endpoint is used for deleting a specific anti-phishing rule. The id parameter in the URL is replaced by the object id of the rule that should be deleted. This route is only accessible by admin users and it is rarely invoked. This endpoint is not exposed to the fronted.

Table 6.2: Implementation of RULE Service API endpoints

Endpoint	Request Type	Authorized User Roles	Description
/api/users	GET	ADMIN	This route returns JSON array of existing . Each user is represented as a JSON object with the following attributes: name, email and user role. Fields like password and salt (used to salt the password) are not provided back for a security reason. This endpoint is called in the frontend app and it is available only to admin users, because they are authorized to do the user management.
/api/users	POST	OPEN ACCESS	This endpoint is used for creating new regular platform users. In the body of the request, data such as: name, email and password for the new user are stored. Before creating a new user a check if this user already exists in the system is being made. This API endpoint is not secured because the creation of new regular users is free for everyone. This endpoint does not support creation of admin users due to security issues. Admin users are created directly in the database.
/api/users/:id	GET	ADMIN, USER	This REST route returns the name and the user role of a single user already registered in the system. Fields like password and salt (used to salt the password) are not provided back for a security reason. The =id parameter in the URL is replaced by the object id of the selected user.
/api/users/:id/password	PUT	ADMIN, USER	This endpoint is used for updating the password of an existing user. Each user could update his/her own password by providing the old password and a new password in the body of the PUT request. A form for platform to all users.
/api/users/:id	DELETE	ADMIN	This endpoint is used for deleting an existing user. The =id parameter in the URL is replaced by the object id of the user. This route is only accessible by admin users. User Management module implemented in the frontend app support deletion of existing users.
/api/users/me	GET	OPEN	This API endpoint is very similar to /api/users/:id

	ACCESS	API endpoint, but returns information specific to the currently logged in user. An additional attribute returned by this endpoint compared to the /api/users/:id endpoint is the user email.

Table 6.3: Implementation of USER Service API endpoints

6.2 External Web Services and Third Party Libraries

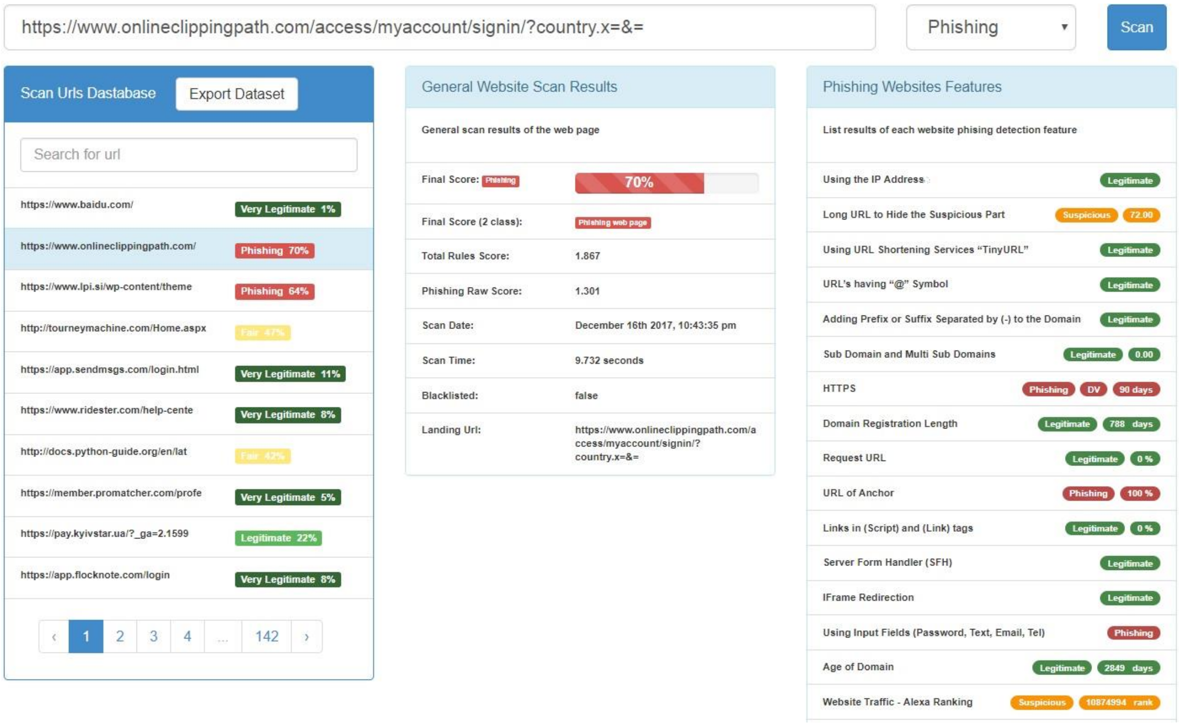
There are many external web services and third party libraries used to facilitate the implementation of PhishEduPro platform. Some of them are implemented as NodeJS packages offered by NPM package manager for JavaScript [87]. All of the NPM packages utilized in this project are popular open source libraries therefore they are frequently maintained by NodeJS community. Additionally, some external web services like WHOIS lockup service, Mozscape web service, WOT web services and SSL checker web service are called using authenticated RESTful API calls which respond with JSON data objects. Table 6.4 lists the most important web services and third party libraries which are used to facilitate the implementation of business logic behind PhishEduPro platform.

Library/ Web Service	Access	Description
Nightmare [88] **/ NPM package**	Free/ No Limitations	Nightmare is a high-level browser automation library which is mainly used for UI testing and web crawling. It exposes a few very simple methods that can be utilized to mimic user action on a webpage. The Nightmare library is used to write a crawler to scrape web page HTML features that are utilized in PhishEduPro anti-phishing detection process.
AlexaRank [89] **/ NPM package**	Free/ No Limitations	This node package is used to obtain the Alexa traffic rank for each scanned webpage by PhishEduPro anti-phishing algorithm.
Tall – URL Unshortner [90]	Free/ No Limitations	A free and reliable URL Unshortner (expander) module for NodeJS.
Google Safe Browsing API [57]	Free/ No Limitations	Google Safe browsing API is used to check if a specific webpage is already blacklisted for hosting a phishing scam. This service is very useful for improving the TP rate of the PhishEduPro algorithm.
Web of Trust (WOT) [86]	Free/No Limitations	WOT service provides a RESTful API for obtaining a reputation score of a suspected webpage. The reputation score for each webpage is calculated based on user voting and rating and a patented machine learning algorithm. The accuracy and reliability of the websites scores are guaranteed by the WOT service. This service is used to decrease the FP rate and to increase the overall accuracy of PhishEduPro anti-phishing algorithm.
SSL Checker [91]	Free/No Limitations	SSL Checks is a tool offered by www.ssl.com to check the SSL certificates of websites running over HTTPS. It checks the duration of the certificate, checks if the certificate is self-signed, check the certificate level, is it expired, and checks if the certificate chain is complete and if the certificate

		authority behind the certificate is trustful. The tool proved this data over RESTful API and the data is returned in JSON format.
WhoisXML_API [92]	Paid/ Sponsorship	WhoisXML_API is the best commercial WHOIS lookup service available on the net. However, we did not pay for it, because we have contact them and explained them that we will use their paid API service for educational purpose only therefore they have offered their services for free. PhishEduPro platform gained access to 5000 WHOIS lookup queries per month, in a period of 6 months, from September 2017 to March 2018. The regular price for this package is 50$ per month.
Mozscape API [85]	Freemium/Usage and Data Limits	Mozscape is a SEO consulting company that offers both a commercial and a limited free service capable to measure the value and ranking potential of websites, by utilizing a set of intelligent metrics to calculate a diverse set of website scores. Using their RESTful API web service we obtain features for improving PhishEduPro anti-phishing algorithm. However no more than 1 API call on every 10 seconds is allowed with their free package.

Table 6.4: External web services and third party libraries utilized in the implementation of PhishEduPro platform

6.3 Graphical User Interface (GUI) of PhishEduPro Platform



https://www.onlineclippingpath.com/access/myaccount/signin/?country.x=&=

https://www.baidu.com/	Very Legitimate 1%
https://www.onlineclippingpath.com/	Phishing 70%
https://www.lpi.si/wp-content/theme	Phishing 64%
http://tourneymachine.com/Home.aspx	Fair 47%
https://app.sendmsgs.com/login.html	Very Legitimate 11%
https://www.ridester.com/help-cente	Very Legitimate 8%
http://docs.python-guide.org/en/lat	Fair 42%
https://member.promatcher.com/profe	Very Legitimate 5%
https://pay.kyivstar.ua/?_ga=2.1599	Legitimate 22%
https://app.flocknote.com/login	Very Legitimate 8%

General Website Scan Results

General scan results of the web page

Final Score: Phishing	70%
Final Score (2 class):	Phishing web page
Total Rules Score:	1.867
Phishing Raw Score:	1.301
Scan Date:	December 16th 2017, 10:43:35 pm
Scan Time:	9.732 seconds
Blacklisted:	false
Landing Url:	https://www.onlineclippingpath.com/access/myaccount/signin/?country.x=&=

Phishing Websites Features

List results of each website phising detection feature

Using the IP Address	Legitimate
Long URL to Hide the Suspicious Part	Suspicious 72.00
Using URL Shortening Services “TinyURL”	Legitimate
URL’s having “@” Symbol	Legitimate
Adding Prefix or Suffix Separated by (-) to the Domain	Legitimate
Sub Domain and Multi Sub Domains	Legitimate 0.00
HTTPS	Phishing DV 90 days
Domain Registration Length	Legitimate 788 days
Request URL	Legitimate 0 %
URL of Anchor	Phishing 100 %
Links in (Script) and (Link) tags	Legitimate 0 %
Server Form Handler (SFH)	Legitimate
IFrame Redirection	Legitimate
Using Input Fields (Password, Text, Email, Tel)	Phishing
Age of Domain	Legitimate 2849 days
Website Traffic - Alexa Ranking	Suspicious 10874994 rank

Figure 6.1: Main screen of PhishEduPro platform

The Graphical User Interface (GUI) of PhishEduPro platform is an important aspect in educating and training end-user to recognize and mitigate future phishing attacks. Upon login each user is redirected to the Scan screen. The Scan screen is considered as a main screen of the platform. The Scan screen is multifunctional, but its main purpose is to engage users with the anti-phishing content generated in real-time for each suspected webpage. By looking in Figure 6.1 it could be noticed that the central place of the Scan screen is took by a big search input field and a search button. As soon as an end-user lands upon a suspected webpage, he/she can copy and paste the webpage URL in the presented search box. By clicking on the big blue Scan button the scanning process of the webpage is initiated. During the webpage scanning users are actively engaged into the anti-phishing evaluation process, where the suspected webpage is shown in a small nested browser frame positioned centrally on the screen. In this browser frame the suspected webpage is loaded in real-time to be scrapped by the PhishEduPro platform web crawler. At this stage the end-user can see the content of the suspected webpage without the risk that his confidential data will be obtained by the phisher. As soon as the crawling is done the browser frame is closed and the phishing score for the suspected webpage is calculated.

The phishing score is displayed to end-users using 3 vertical panels shown at the Figure 6.1. The first panel shows a list of already evaluated webpages ordered by scan date. By clicking on each webpage URL shown in this panel, panels 2 and 3 are filled with anti-phishing statistics for the selected webpage. Panel located in the middle of the screen shows the general webpage scan results such as: Final phishing score %, Final phishing score reported as a binary value (phishing or legitimate), The total rule score (sum of the weights of all anti-phishing features), Phishing Raw score (the sum of phishing and suspicious feature weights for the selected webpage), scan date, scan time in seconds, blacklisting status of the URL and the URL address of the scanned webpage. Most important metric shown in this panel are the Final Phishing Score % and the Final phishing score reported as a binary value. The Final Phishing Score % metric is represented with a progress bar used to show the phishing rate % for each webpage. There are 5 different colors to illustrate the phishing status of a webpage. The dark green indicates a very legitimate webpage, light green indicates a legitimate webpage, yellow progress bar indicates a fair webpage (webpage that has some phishing characteristics, but is more likely to be legitimate), the orange progress bar indicates a very suspected webpage (page that is very likely to host a phishing attack) and red progress bar indicates a webpage which is confirmed as an active phish. Consequently, the Final Phishing Score reported as a binary value is a metric displayed using only 2 colors. Dark green is used to label a legitimate webpage and a red color is used to label a phishing webpage. The offset for a legitimate webpage is from 0% to 51% and the range for a phishing web page is from 52% to 100%. The cutoff point at 51% is calculated by evaluating the dataset of scanned webpages. Using colors to indicate a status of a webpage is considered very productive, because colors are easily remembered by end users.

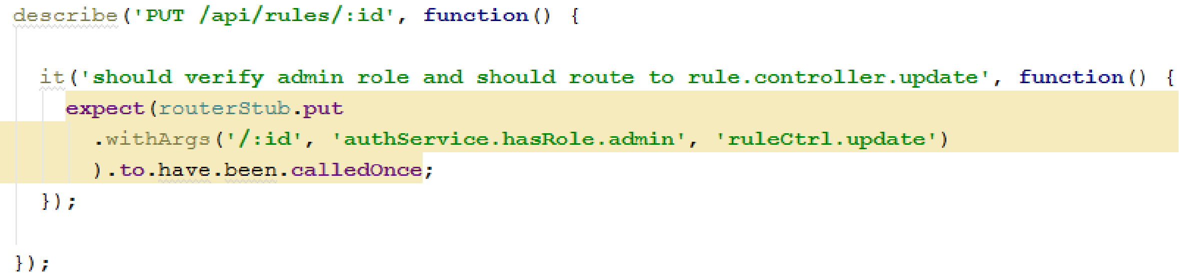
The last panel illustrated in Figure 6.1 displays the value of each anti-phishing feature used in the web page classification process. PhishEduPro platform uses 3 different colors green (legitimate), orange (suspicions) and red (phishing) to indicate the status of each feature separately. By continuously looking into anti-phishing statistics and by observing the entire webpage scanning process users are trained to label a specific page on their own, without the need to look at the final score produced by the anti-phishing algorithm. Thus, end-user will have an insight in the evaluation process of PhishEduPro anti-phishing algorithm and they will be educated and trained to recognize common patterns in phishing and legitimate web pages. The screen shown in Figure 6.1 provides additional functionalities like: searching for existing URLs scanned by the platform and exporting a dataset of webpages represented with binary features which are already evaluated by the algorithm. The screens developed for Statistics and for Rule tuning modules are shown in Appendix C.

6.4 Testing

Testing the functionality of the PhishEduPro platform is an important step in the software development process. The testing phase is scheduled at the end of each iteration cycle to validate the user requirements and to verify that the product is of a high quality and it is error-free. Three different techniques are used to test the PhishEduPro platform such as: unit testing, integration testing and manual system testing. A different configuration file in the project environment named **test.js** is created to define the testing setup. A new instance of MongoDB is used for the tests so that the production and development databases will not be affected by the testing inputs. Unit and Integration tests are run in a batch with the following command: ***grunt test: server***. The name of the testing database is ***phishedupro-test*** and it is deleted after all tests are completed, in order to be prepared for the next testing round. Manual testing is done after all unit and integration tests are completed, in order to test the overall performance of the system. Manual testing is continuously performed during all iteration cycles. If a change in requirements or in the implementation of the PhishEduPro platform is made at any stage of the development process all tests are repeated. If some of the tests failed, those tests are updated to confirm to the new requirements and implementation. The testing suite used to test PhishEduPro platform consists of 3 independent testing tools. Mocha [93] is a testing framework for JavaScript running on NodeJS which makes asynchronous testing simple and fun. Mocha tests run in a sequential fashion which allows flexible and accurate reporting of the testing results. Chai [94] is an assertion library for NodeJS which could be paired with any JavaScript testing framework. Chai provides several interfaces to write test assertions in a readable style. = **Should”** **Expect”** **Assert** **Should** **Expect** styles use a natural language like syntax very close to the English language, which make tests very readable and understandable for everybody. Sinon [95] is a library that allows developers to replace difficult parts in tests with something simpler and it works with any JavaScript testing framework. For example testing network Ajax calls, database operations, and function callbacks are very hard testing tasks, but by using Sinon library and its test spies, stubs and mocks they are easily achievable. Sinon helps developers to eliminate testing complexity by allowing them to create test-doubles (replacements for pieces of code used in the test).

6.4.1 Unit Testing

Unit testing is a white box testing technique where individual units of source code are tested to identify, analyze and fix defects. For this project we focused on testing the most important functions defined for the Scan, Rule and User RESTful web services. The setup of the unit tests is the same for each web service. Each web service is organized as a module having a separate working directory therefore unit testing setup file named **index.spec.js** is created and placed in this directory. At the top of **index.spec.js** file a mock object is created, containing the signatures of all methods defined in the controller of the web service. Methods like index, show, create, update, delete etc. are set as attributes to this mock object.
object methods
are used to check if the route is only accessible for users having the right permissions. RouterStub is a mock object which uses Sinon library to create spies (offer information about function calls without affecting their behavior) for mocking the get, put, post and delete routes to eliminate the testing complexity via the network. Spies could be utilized to verify specific behavior of a function. For example they can give information to the developer if a function has been called successfully zero or more times during the test run. ScanIndex object is used to mock the structure of the index.js file which is used to define the Scan RESTful API
used to override the require statements of the module that is currently under test in this case the index.js file. The authServiceStub, RouterStub and mock object containing the controller methods are passed as modules to the ScanIndex object to mock the behavior of the index.js router file.

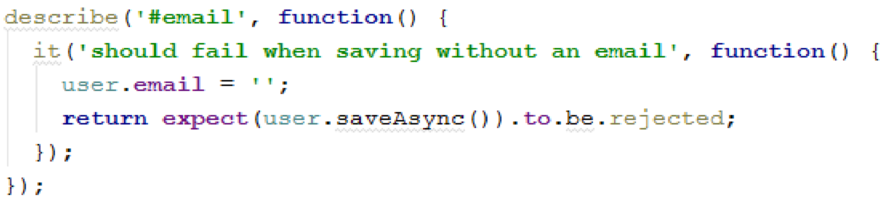


```
describe('PUT /api/rules/:id', function() {

  it('should verify admin role and should route to rule.controller.update', function() {
    expect(routerStub.put
      .withArgs('/:id', 'authService.hasRole.admin', 'ruleCtrl.update')
      ).to.have.been.calledOnce;
  });

});
```

Figure 6.2: Unit test of the update Rule method

Figure 6.2 illustrates a unit test for the Update Rule method which is accessed via the Rules RESTful API interface. This is a very basic test which is used to confirm that only admin user can update rules in the system and that the call to PUT /api/rules/:id will route to the update method defined in the controller. Furthermore this test ensures that the call to the route will trigger the update method only once.

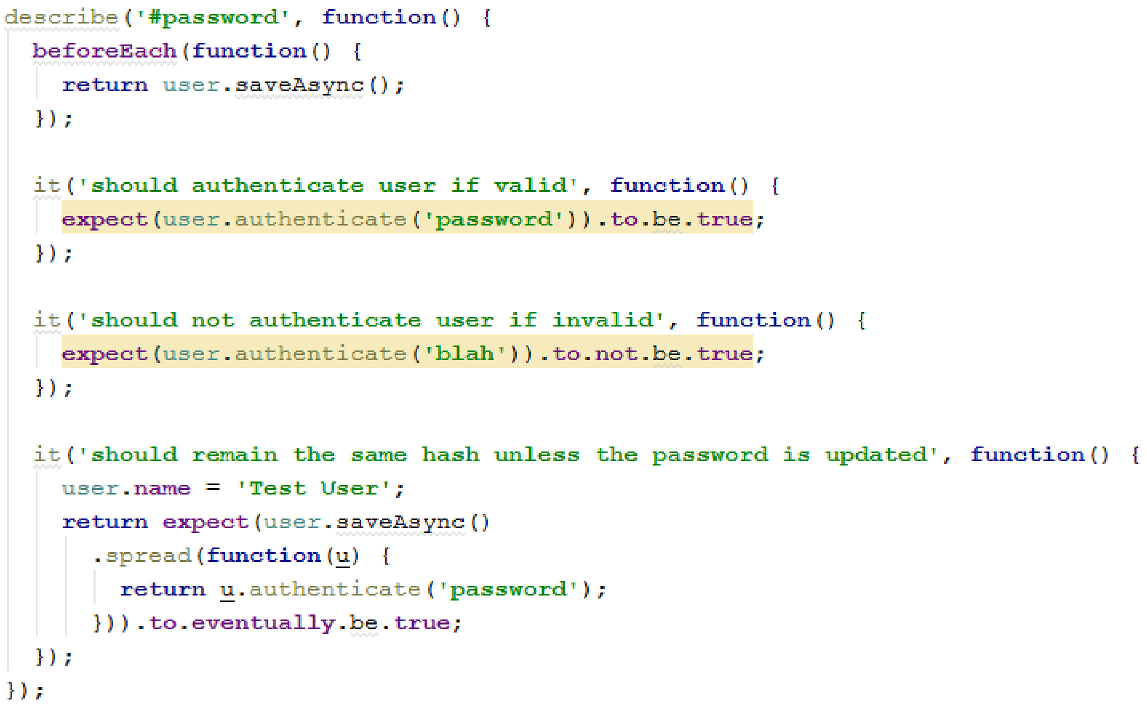
There are more complex unit tests which are used to test specific methods attached to the model while performing database operations. The file **user.model.spec.js** placed in the web service directory contains unit tests which are used to test functions linked to the User model.



```
describe('#email', function() {
  it('should fail when saving without an email', function() {
    user.email = '';
    return expect(user.saveAsync()).to.be.rejected;
  });
});
```

Figure 6.3: Save User without an email unit test

The unit test illustrated in Figure 6.3 is very descriptive and can be understand well just by looking at of related unit tests. Each unit test is written in the body of the it() function. The it function has 2 parameters. The first parameter is a user friendly textual description of the test and the second parameter is an anonymous function where the unit test is executed. T Chai assertion is used to test if the user creation will be rejected if the user email is set to an



```
describe('#password', function() {
  beforeEach(function() {
    return user.saveAsync();
  });

  it('should authenticate user if valid', function() {
    expect(user.authenticate('password')).to.be.true;
  });

  it('should not authenticate user if invalid', function() {
    expect(user.authenticate('blah')).to.not.be.true;
  });

  it('should remain the same hash unless the password is updated', function() {
    user.name = 'Test User';
    return expect(user.saveAsync()
      .spread(function(u) {
        return u.authenticate('password');
      })).to.eventually.be.true;
  });
});
```

Figure 6.4: Unit tests related to user password management

Figure 6.4 illustrates a set of related unit tests that are run to ensure proper user password management. The first test tries to authenticate a user when the password is valid. The second test is the opposite of the first and should pass if the password is invalid. The last test in this set, checks if the change of the name of the user will affect the password hash.

6.4.2 Integration Testing

After completion of unit testing, related methods of each web service are combined and tested as a group. Integration tests could expose problems with the interface among independent units (methods) which are combined to achieve a higher system and business goal. In the testing process of PhishEduPro application we follow up the bottom-up integration testing approach. The characteristic of this approach is that it starts with unit testing, and then gradually unit tested methods are combined in a module with higher complexity where the functionality of this advanced module is tested by utilizing a black box or white box testing techniques. In the case of PhishEduPro platform, integration tests are done using black box testing technique, because many different methods are blindly invoked to achieve a proper operation of a specific web service. This means that any integration test should accept specific input and should produce an expected output without revealing the logic and processing behind its internal components.

For our system we have designed 17 integration tests which are covering all functionalities offered by the platform. Each web service the Scan, the Rule and the User has a specific set of integration tests defined within the **scan.integraion.js**, **rule.inegration.js** and **user.integration.js** files respectively. The structure of those files is very similar and each integration test module contains a reference to a user object and a valid authentication token, because most of the API routes in the PhishEduPro platform are authenticated and authorized. The execution of integration tests written for a specific web service starts with a

argument is the name of the web service API. Furthermore, its second argument is an anonymous function invocation call which holds the set of integration tests written for testing the functionality offered by this web service. Usually

The

once, just before any integration test is started. On the other ha

specific functions after all integration tests are being completed. Creating a new admin user and obtaining an authentication token for this user, is a usual action performed within the

test file. Consequently, removing all users created during the integration testing process and removing objects like Scans and Rules from the

located in the integration testing file. Thus, it is ensured that each newly created testing round would not be affected by earlier integration test runs and that the testing database will be clean at the beginning of each testing cycle.

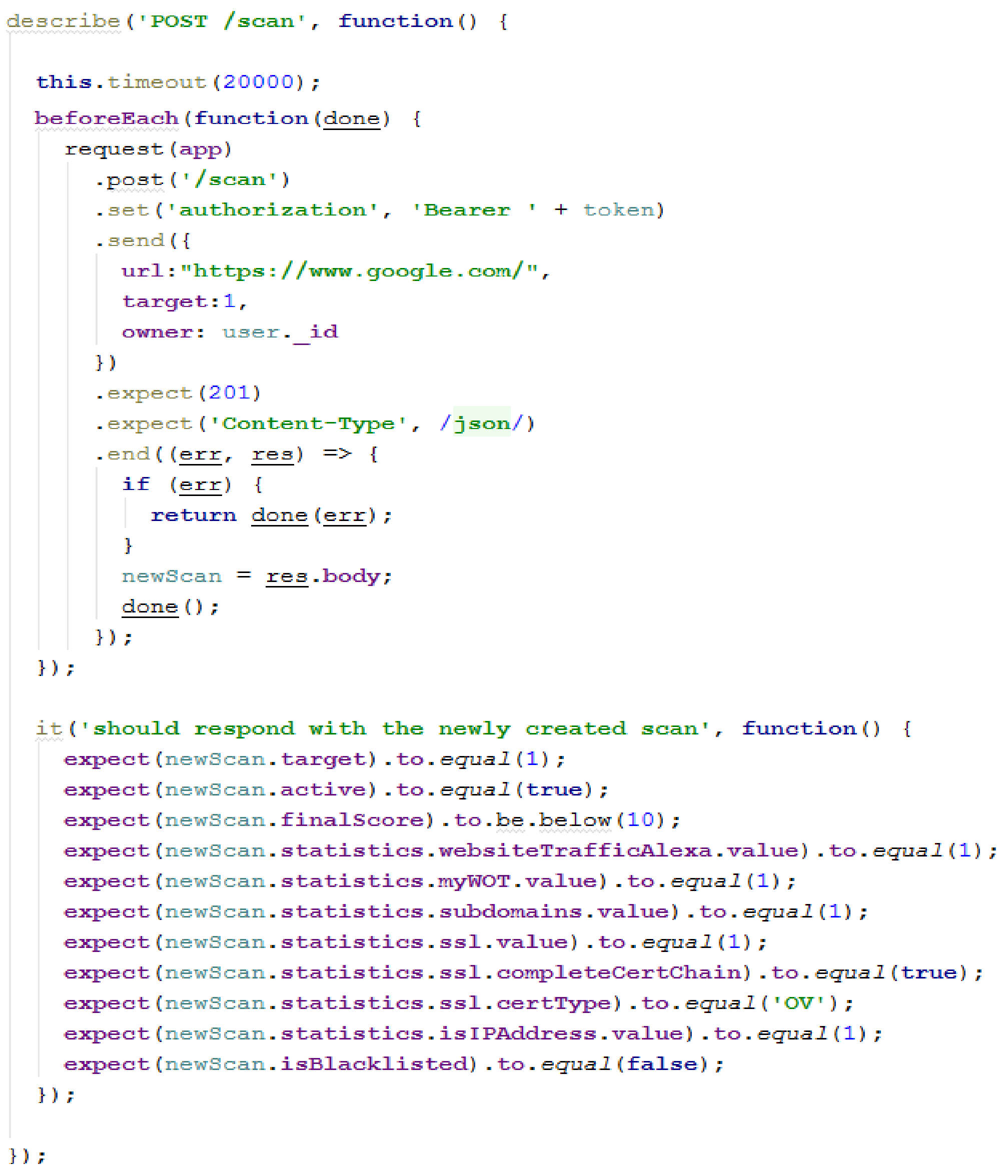
compared to

block of the Scan web service integration testing file, the system should create a list of 22 anti-phishing rules and should store them in the test database. Those Rule objects are essential for scanning suspected websites and must be present in the testing database for successfully run of integration tests.

all integration tests are being completed. Figure 6.5 illustrates an integration test written for

legitimate.

Furthermore, all other integration tests could be found in the GitHub repository of PhishEduPro web application.



```
describe('POST /scan', function() {

  this.timeout(20000);
  beforeEach(function(done) {
    request(app)
      .post('/scan')
      .set('authorization', 'Bearer ' + token)
      .send({
        url:"https://www.google.com/",
        target:1,
        owner: user._id
      })
      .expect(201)
      .expect('Content-Type', /json/)
      .end((err, res) => {
        if (err) {
          return done(err);
        }
        newScan = res.body;
        done();
      });
  });

  it('should respond with the newly created scan', function() {
    expect(newScan.target).to.equal(1);
    expect(newScan.active).to.equal(true);
    expect(newScan.finalScore).to.be.below(10);
    expect(newScan.statistics.websiteTrafficAlexa.value).to.equal(1);
    expect(newScan.statistics.myWOT.value).to.equal(1);
    expect(newScan.statistics.subdomains.value).to.equal(1);
    expect(newScan.statistics.ssl.value).to.equal(1);
    expect(newScan.statistics.ssl.completeCertChain).to.equal(true);
    expect(newScan.statistics.ssl.certType).to.equal('OV');
    expect(newScan.statistics.isIPAddress.value).to.equal(1);
    expect(newScan.isBlacklisted).to.equal(false);
  });

});
```

Figure 6.5: Integration tests f

max timeout of 20 seconds (20000 .ms). The average time needed to scan a suspected webpage is around 12.5 seconds, but the default timeout max

block implemented for this set of related tests is executed before each independent test run

beforeEach block from Figure 6.5 is used to trigger a POST request call to scan a suspected URL by providing URL address (https://google.com/), a target value where 1 means that the website is legitimate and an owner (the user who initiated the scan request). Chained .set() method invoked

is used to authorize and authenticate the user who initiated this call by providing an authentication token to the request. When the request is finished and the evaluation of the web page is completed the result from the evaluation is assigned to a
an it() the
Chai assertions to check if the returned object conforms to the predefined test specifications.

6.5 Deployment

The PhishEduPro platform is envisioned to be deployed on a scalable cloud server which will efficiently perform a real-time classification of suspected web pages. This cloud solution should be scalable enough to effectively handle simultaneous classification of websites submitted from various end-users. A suitable deployment solution for PhishEduPro web application is Heroku [96]. Heroku is a cloud platform based on a managed container system with integrated data system for deploying and running modern apps. It also supports continuous delivery and continuous deployment plans and it is very popular deployment environment for NodeJS applications.

According to nonfunctional requirements the average time required to scan a newly submitted webpage should be between 3 - 4 seconds when the system is deployed on Intel Xeon 16 core Ubuntu production server with 32 GB RAM and 3.0 GHz Quad Core processor. The free version of Heroku deployment plan offers only 512 MB of RAM memory and a shared processing power where autoscaling is not supported. This means that the offered solution will be at least 8 times slower than our development environment where the average scan time for a single web page is around 13 seconds. On the other hand, the most powerful commercial solution offered by Heroku has 14 GB of RAM memory with dedicated processing power and auto scaling enabled. The system auto scaling option will ensure that the processing power of the cloud server will scale, when the traffic is increased, therefore end-users experience will not be affected. The price for the proposed commercial package is 500$ per month not including the database storage. This is powerful, yet very expensive solution and currently it is not an option for us.

R9 risk mitigation strategy described in section 4.2.1 should be consulted to resolve the issue related to PhishEduPro platform deployment. R9 risk mitigation strategy recommends publishing the entire code of the platform to GitHub, thus we will share the code with all potential users for free. Each of the users will be able to download the code and run it on their local environment. Additionally, if the work on this project is being continued and a necessary financial support is acquired from third party investors PhishEduPro platform will be deployed to a reliable production server.

Chapter 7

7. Evaluation

7.1 Evaluation Methodology

A real life experiment has been conducted to evaluate the performance of PhishEduPro anti-phishing algorithm. The experiment was carried out on a laptop running Windows 8.1 equipped with Intel i7 Dual Core 2.4 GHz processor, 4GB RAM and 6 Mbps Internet link. The experiment started on 13th October 2017 and last until 16th December 2017. A dataset of 1420 URLs 710 phishing and 710 legitimate has been used in the evaluation process. The phishing URLs were obtained from PhishTank database [58]. The selection process has been very precise and only newly reported URLs were selected for evaluation. Since the evaluation of URLs was an ongoing process performed in real-time, as soon as a new URL was reported to PhishTank it was collected and fed to PhishEduPro platform for analysis. By using this approach the chances that the URL is pointing to a webpage recognized as a zero-day phishing attack were greatly increased. Moreover, the chances that this webpage is not already blacklisted were improved as well. Webpages that have been accessible online by the time they were fed to PhishEduPro platform were evaluated only.

Legitimate URLs were collected from 2 different sources. The first set of legitimate URLs was obtained from Quantcast an innovative digital marketing company that uses Big Data and machine learning to solve crucial marketing challenges for their clients. They have analyzed more than 150 million of legitimate websites, and they process around 20 petabytes of data every day [97]. Additionally, they provide a free online access to a list of millions of legitimate websites ranked by popularity. A random dataset of 500 webpages was selected from this list. The rest 210 legitimate URLs were randomly selected from a dataset provided by a security researcher Faizan Ahmad, in one of his GitHub repositories [98]. PhishEduPro algorithm results were compared to 5 anti-phishing approaches. One of the approaches is implemented as a blacklist, other 2 use rule-based heuristics and the last 2 approaches are based on machine learning. More details about the evaluation strategy are discussed in section 7.2.

7.2 Evaluation Strategy

During the 2 months evaluation process, PhishEduPro platform continuously tracked the performance of the anti-phishing algorithm, by utilizing the Statistics module which is accessible -user. The Statistics module implemented a well-known set of anti-phishing evaluation metrics. Metrics that were used in the evaluation of the algorithm are discussed in details in section 2.4.1 from the literature review. Later, when the final set of anti-phishing statistics were calculated by the platform, the performance of the PhishEduPro anti-phishing algorithms was compared to 5 different anti-phishing approaches published in reputable information security journals. A screenshot of the final state of the Statistics module is shown in Appendix D.

First of all, the performance of PhishEduPro algorithm was compared to Google blacklist output on the same dataset used in the evaluation process. In each scan request initiated by an end-user our algorithm utilized Google Safe Browsing API [57] to check if a suspected webpage is blacklisted. Later, the result obtained from the blacklist is bounded to suspected webpage anti-phishing statistics and it is stored in Mongo DB. Thus, each webpage has been evaluated by PhishEduPro algorithm and by Google Safe Browsing API and the anti-phishing evaluation metrics were calculated for both approaches. Evaluation metrics which are utilized in the comparison among the proposed anti-phishing approaches are the following: Accuracy Rate of the algorithm, False Positive Rate (FP), False Negative Rate (FN), True Positive Rate (TP) and True Negative Rate (TN). Moreover, the scan time performance of the PhishEduPro algorithm is measured and compared to one of the approaches that provide this type of data.

The next anti-phishing approach used in the comparison is CANTINA. CANTINA is content based anti-phishing approach, implemented as a toolbar for Internet Explorer and its anti-phishing detection process is based on Term Frequency-Inverse Document Frequency (TF-IDF) information retrieval algorithm, search engine results and a set of simple heuristic rules to reduce false positives [64]. The evaluation metrics of this approach are obtained from a study [64] published as a conference paper . This approach uses 100 phishing URLs collected0 1 164.54 558.1 Tm[(-)](4.851 0 1 364.vd)-10(a)4(pproankist)-(me)-3b 9usesm newly proposed features. Total of 27 anti-phishing features are fed into Support Vector Machine (SVM) classifier to obtain the class of each webpage. Dataset used in this approach is composed of 500 legitimate webpages obtained from Alexa Service and 500 phishing webpages obtained from PhishTank database. According to the study in [100] 70% of the dataset is used for training and validation of the classifier and the rest 30% are used for testing.

The last approach used in the comparison is proposed by Ramesh at al. [101] and it is recognized as one of the most accurate anti-phishing methods by the research community. It is published in the same journal, Decision Support Systems, like the neural network approach discussed earlier. Their Target Identification (TID) algorithm is based on phishing detection via target identification, where the researchers are promoting the idea that the target will always be a legitimate website. The classification process of this algorithm is based on extracting a set of direct links S1 and a set of the most influential keyword form suspected webpage S2. Later, the keywords set S2 is fed to Google search engine and the top results (links returned from the search engine) are cross checked to S1. A subset S3 of matching links is evaluated by the TID algorithm and the target website is identified. The last step of the algorithm is to match the IP address of suspected website to the IP address of the target. If they are identical the page is legitimate, otherwise the suspected webpage is phishing. This approach is implemented using Java 7 standard edition and it is very accurate in detecting zero-day phishing attacks. The dataset used to test TDI algorithm consists of 4574 webpages where 1200 are legitimate and 3374 are phishing [101]. The legitimate webpages are obtained from Alexa top sites and Google Top 1000 visited sites and the phishing webpages are obtained from PhishTank database. In the following section, evaluation results obtained by doing a comparison among the proposed anti-phishing approaches are presented using bar charts.

7.3 Evaluation Results

Figure 7.1 illustrates a bar chart showing the accuracy rates of each of the proposed anti-phishing approaches. The vertical axis shows the accuracy rate estimated in percentage (%) where the accuracy rate value ranges from 0% to 100%. The horizontal axis displays each of the proposed anti-phishing approaches by their name or its main author name if the name of the approach is not available.

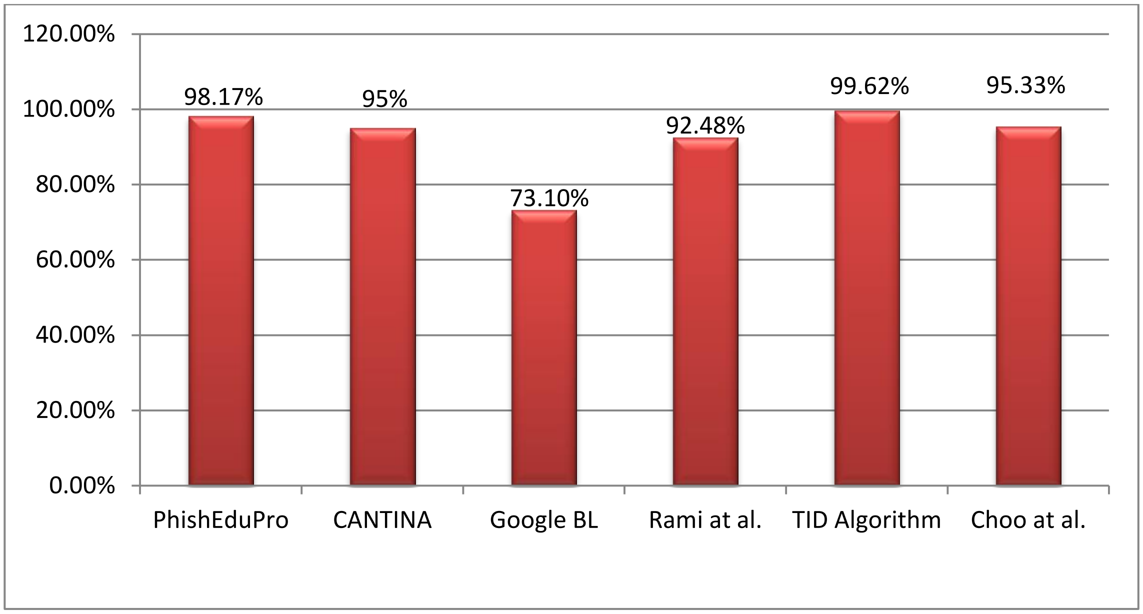
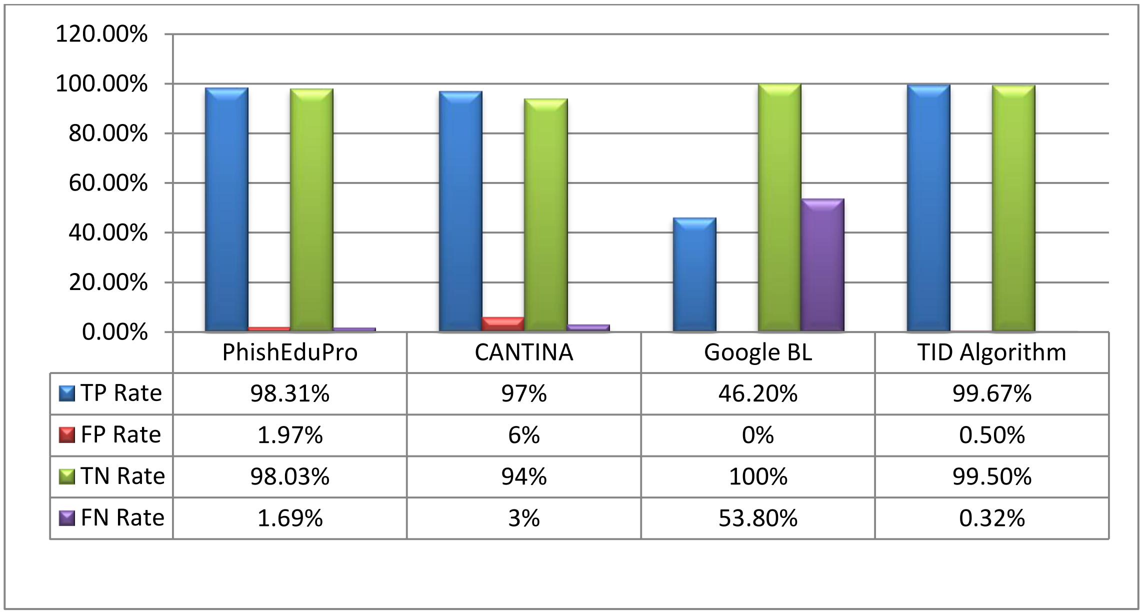


Figure 7.1: Accuracy rates achieved by different anti-phishing approaches

Figure 7.2 illustrates a chart showing the most important evaluation metrics used in the comparison among the proposed anti-phishing approaches such as: True Positive Rate (TP), False Positive Rate (FP), True Negative Rate (TN) and False Negative Rate. Only 4 of the proposed approaches have reported those metrics. The approaches developed by Mohammad at al. [99] and by Choo at al. [100] did not reported these metrics in their studies. Each algorithm is represented with a sequence of 4 bars in different colors where each color represents a specific anti-phishing metric. The correct value of each anti-phishing metric is reported using a legend table displayed below the chart.



	PhishEduPro	CANTINA	Google BL	TID Algorithm
🟦 TP Rate	98.31%	97%	46.20%	99.67%
🟥 FP Rate	1.97%	6%	0%	0.50%
🟩 TN Rate	98.03%	94%	100%	99.50%
🟪 FN Rate	1.69%	3%	53.80%	0.32%

Figure 7.2: Anti-phishing evaluation metrics for different anti-phishing approaches

In the chart shown in Figure 7.3, the average scan time of PhishEduPro anti-phishing algorithm is reported. The first bar shows the average time need to scan a phishing URL. The second bar shows the average time needed to scan a legitimate URL. Finally, the last bar displays the average scan time measured for both phishing and legitimate URLs.

Figure 7.3: PhishEduPro algorithm average scan times measured by URL type

Figure 7.4 illustrates the PhishEduPro anti-phishing algorithm minimum and maximum scan time measured on the whole dataset

7.4 Discussion

features, the updates applied to anti-phishing rules of the most influential features and the introduction of new features which were proven as an excellent choice in decreasing the FP rate and increasing the overall accuracy of the PhishEduPro anti-phishing algorithm.

Since, PhishEduPro anti-phishing approach performs better than the above mentioned approaches a suitable candidate to compare our method with is the TID algorithm. TID algorithm yields the highest accuracy rate 99.62% among all evaluated candidates. Moreover, TID algorithm achieved the lowest FP and FN rates of 0.50% and 0.32% respectively. This could be seen in Figure 7.2. On the other hand, PhishEduPro algorithm achieved an accuracy of 98.17% with FP rate of 1.97% and FN rate of 1.69%. However, our observations showed that these results should not be taken for granted when comparing both approaches. First of all, in the study [101] it is reported that TID algorithm is not able to detect a phishing attack if a suspected webpage does not contain links to internal or external web pages or textual content to extract keywords from it. Many time phishers create fraudulent web pages composed only of images. The PhishEduPro algorithm is able to detect those attacks successfully which is not the case with TID algorithm. From the dataset used to evaluate PhishEduPro anti-phishing algorithm, 143 out of 710 phishing webpages or in percentage 20.15% were based only on images. Those webpages did not have any links or keywords. Only 7 out 710 (0.99%) legitimate webpages did not include links in their HTML content. Thus, we devise a new rule for assigning a class to Anchor URL feature, where the feature is labeled as phishing if the suspected webpage does not include any internal or external links. Furthermore, in the study [101] it was reported that TID algorithm is not able to detect phishing attacks hosted on compromised domains, because in this case the target and suspected web page will point to the same host and IP address. PhishEduPro is able to detect most of these attacks, however with some difficulties that are discussed in the next section of this chapter. Since, TID algorithm require an efficient language independent keyword extraction this might raise some difficulties in detecting web pages where the content is not in English[101]. Quite the reverse, PhishEduPro algorithm is not content based anti-phishing approach and it is language independent. Therefore, it is able to detect phishing websites written in many different languages like Russian, Chinese, Spanish, and English etc. Despite the fact that PhishEduPro platform operates in real-time and TID algorithm is implemented as offline desktop solution, PhishEduPro algorithm is much faster in scanning and classifying webpages as phishing or legitimate. According to the findings reported in [101] an average scan time of TID algorithm when it is run on a computer with a 2.4 GHz processor and 4GB RAM is 20.8 seconds. Accordingly, the average run time of PhishEduPro algorithm run under similar environment setup (Intel i7 2.4 GHz processor with 4GB RAM) is 12.75 seconds. More specifically the average scan time for phishing webpages is 11.09 seconds and for legitimate web pages is 14.4 seconds. This shows 38.7% better performance of PhishEduPro algorithm compared to the performance of TID algorithm.

The main advantage of PhishEduPro platform over all of the evaluated anti-phishing approaches is that it raises user awareness about phishing by educating and training users to recognize and mitigate future phishing attacks. None of the evaluated approaches gives an insight into their website classification process to end-users. They operate like black-boxes that reports only if a specific webpage is phishing or legitimate, but they do not offer extra guidelines why that is the case. By presenting anti-phishing statistics to end-users for each evaluated webpage, PhishEduPro platform educates and trains users to recognize future phishing attacks. Even in a situation when the platform will fail to correctly classify a webpage, by consuming their already acquired anti-phishing knowledge and by looking in the detailed website statistics generated by the platform, users will be able to successfully classify suspected websites on their own. Additionally, PhishEduPro platform offers a module for fine-tuning rules and feature weights used in the classification process. None of the evaluated approaches offer a similar functionality. By fine-tuning the initial weights our algorithm could be adapted to recognize future trends in phishing, or it could be made more sensitive in detecting legitimate webpages. Finally, the PhishEduPro platform successfully resolved the last product objectives stated in Chapter 1. The platform provides a free access to a frequently updated set of phishing and legitimate webpages represented with binary features. Therefore security researcher will utilize them for training and testing novel anti-phishing approaches.

7.5 Limitations of Our Approach

Despite the benefits offered by PhishEduPro anti-phishing approach there are few limitations that should be considered. PhishEduPro anti-phishing algorithm does not perform very well when the phishing webpage is hosted on a compromised legitimate domain, but only if the legitimate domain is behind a HTTPS protocol. Since the weight assigned to HTTPS anti-phishing feature has the biggest impact to classification process compared to all other features, it is very probable that PhishEduPro algorithm will yield a legitimate value for a phishing webpage hosted on a compromised legitimate domain using HTTPS. Furthermore, if a legitimate webpage is implemented using old fashioned HTML style, and the <frame> tag is used to build different section of this webpage, PhishEduPro algorithm might misclassifies it as phishing and vice versa. This limitation is due to NightmareJS package that is used for scraping the content of webpages. The webpage crawler which has been developed using this package is not able to scrape HTML features contained within <frame> tags. Finally, PhishEduPro algorithm is likely to misclassify a legitimate web page, which has an excessive number of links pointing to external domains. Examples of such web pages are personal blogs, or community forums websites. However, this could be prevented by utilizing the Rules module of the platform to decrease the weight associated with Anchor URL feature.

Chapter 8

8. Conclusion

8.1 Future Work

There is always space for improvement no matter how good the system looks at a given moment of time. Without continual growth and progress the PhishEduPro platform will become just another forgotten idea at the cemetery of academic achievements. In this section of the report guidelines for improving the PhishEduPro platform will be proposed so that some prospective students could continue its journey.

Create anti-phishing browser plugin or anti-virus extension

PhishEduPro platform is currently operating as a standalone web application. To make use of it each user should navigate to its web address and login with his/her credentials. This is an easy step if the user is a security researcher or the user is aware of the danger of phishing attacks. However, the cyber security is a secondary task for majority of people which are currently online. They do not care much about phishing or any other type of online attacks. A better approach is to protect them with a mechanism which will be triggered automatically in case of potential attack. Since PhishEduPro platform is implemented as set of RESTful API web services, a browser plugin or an anti-virus extension is easy to be implemented and offered to end-users as a service. By downloading and installing this plugin in the browser, or installing it as a service for the anti-virus software, end-users will be notified about the phishing attack as soon as they step on a suspect webpage. This approach will significantly reduce the success of phishing attacks and will raise user awareness about phishing.

Implement educational RESTful API web service

Currently, PhishEduPro platform is implemented as a set of distinct and scalable web services. The Scan service is the most important web service and its main functionality is to evaluate a web page and to inform end-users if the suspected web page is phishing or legitimate. To calculate the anti-phishing score of a web page, the Scan web service utilizes 22 different features and rules. Many end-users will not have an opportunity to use an automated tool to protect themselves from phishing attacks therefore educating and training them about phishing is an alternative that should be considered seriously. Some of the 22 features used in phishing detection process of the PhishEduPro platform are not good candidates for education and training user to recognize and mitigate phishing attacks, because they are not easily noticeable at a first glance. However, some of them especially **Address Bar Based Features** are perfect for educating and training users to recognize potential phishing attacks. Implementing an additional RESTful API web service, which will be able to analyze a suspected web page and then return a subset of features good for anti-phishing education should be an imperative. Some prospective students could use this web service to implement a real-time educational tool as a gamification platform that will educate end-user about phishing by utilizing real life examples of phishing and legitimate web pages. The educational RESTful API endpoint is very easy to be implemented by following the design and the implementation steps of the existing RESTful web services.

Improve the accuracy of PhishEduPro anti-phishing algorithm

The evaluated accuracy of PhishEduPro algorithm is 98.17% which is considered as a very high number by the research community. Nevertheless, this value could be further enhanced by introducing some small platform improvements. The first proposed improvement is to replace the rule based phishing detection of PhishEduPro platform with a machine learning approach. The major disadvantage of rule based heuristics is that its yields a higher false positive rate compared to machine learning approaches. Moreover, rule based heuristics is static and if a change in phishing tactics occurs the change will be missed by the currently implemented anti-phishing algorithm. On the other hand, machine learning approaches are better in detecting changes in phishing strategy therefore they yield lower FP rates. Additional alternative approach would be to implement dynamic addition of new anti-phishing features and rules. Those features and rules could be created by end-users using drag and drop components or web forms and could be saved in the database as soon as a new phishing strategy emerges. This way the PhishEduPro platform will be easily adaptable to changes and its FP rate will be kept at minimum.

8.2 Final Words

Phishing is a pervasive threat that affects and will continue to affect many individuals, companies and organizations worldwide. Fighting against phishing attacks is not an easy task to accomplish, because phishing acts against different aspects of human behavior, such as: technical knowledge, and their online trust. PhishEduPro platform has managed to fight back phishing attacks in many different ways. First of all, it offered an automated phishing detection tool that accurately classified 98.17% of suspected websites therefore reducing the potential risk of phishing attacks to just 1.83%. Furthermore, it has raised user awareness about phishing by offering end-users a complete set of anti-phishing statistics of each evaluated web page. By utilizing PhishEduPro platform users have a detailed insight into phishing detection process therefore they are educated and trained to recognize common patterns in phishing and legitimate web pages. By continuously looking into anti-
observing the entire web page scanning process, users are trained to label a specific page on their own. Finally, the entire process of developing this Master Thesis, starting from a raw idea and finishing with a working product, was an amazing, yet challenging journey. We feel pleased that we have created something for the community without asking anything for a return. We hope that this project will continue to evolve, and will become something that will make us all proud in the near future.

References

[1] -line trust: Concepts, evolving *Int. J. Hum. Comput. Stud.*, vol. 58, no. 6, pp. 737 758, 2003.

[2] = *Diss. Abstr. Int. Sect. B Sci. Eng.*, vol. 70, no. 5 B, p. 3000, 2009.

[3] = *Crypto-Gram Newsletter*, 2000. [Online]. Available: https://www.schneier.com/crypto-gram/archives/2000/1015.html. [Accessed: 14-Nov-2017].

[4] *ACM Comput. Surv.*, vol. 48, no. 3, pp. 1 39, 2015.

[5] M. Jakobsson and S. Myers, *T e h g e h e h increasing problem of electronic identity theft*. Wiley-Interscience, 2007.

[6] = *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 2091 2121, 2013.

[7] *Proc. 16th Virus Bull. Int. Conf.*, vol. 73, no. 1992, pp. 1 2, 2006.

[8] -Phishing Working Group Phishing Attack Trends Report Anti-Phishing Working Group,

[9] *APWG*, no. December, p. 14, 2016.

[10] -phishing phil: the design and evaluation of a game *Proc. SOUPS 2007*, pp. 88 99, 2007.

[11] phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of *Proc. 28th Int. Conf. Hum. factors Comput. Syst. - GL M 4*, pp. 373 382, 2010.

[12] = *Neural Comput. Appl.*, pp. 1 26, 2016.

[13] *Proc. 16th Int. Conf. World Wide Web - 4* , p. 649, 2007.

[14] =

BMJ Br. Med. J., p. 368, 2003.

[15] *SANS Inst.*, no. GSEC Option 1 version 1.4b, pp. 1 39, 2003.

[16] = [Online]. Available: https://www.symantec.com/connect/articles/social-engineering-fundamentals-part-i-hacker-tactics. [Accessed: 19-Nov-2017].

[17] *Proc. CONF-IRM*, 2011.

[18] Attacks: Taxonomy *CoRR*, vol. abs/1705.0, pp. 1 32, 2017.

[19] = https://www.fbi.gov/news/stories/romance-scams. [Accessed: 20-Nov-2017].

[20] *Security*, pp. 1 42, 2008.

[21] L. James, *Phishing exposed*. Syngress, 2005.

[22] *Cent. Strateg. Int. Stud.*, no. July, pp. 1 20, 2013.

[23] *Comput. Fraud Secur.*, vol. 2010, no. 6, pp. 5 8, 2010.

[24] phished? Testing individual differences in phishing vulnerability within an *Decis. Support Syst.*, vol. 51, no. 3, pp. 576 586, 2011.

[25] = *J. Pers. Soc. Psychol.*, vol. 52, no. 3, pp. 639 644, 1987.

[26] G. Krieg and T. Kop - *CNN*, 2016. [Online]. Available: http://edition.cnn.com/2016/10/28/politics/phishing-email-hack-john-podesta-hillary-clinton-wikileaks/. [Accessed: 21-Nov-2017].

[27] 77, 2017.

[28] *The Guardian*, 2016. [Online]. Available: https://www.theguardian.com/us-news/2016/dec/14/dnc-hillary-clinton-emails-

hacked-russia-aide-typo-investigation-finds. [Accessed: 25-Nov-2017].

[29]
=
https://www.forbes.com/sites/leemathews/2017/05/05/phishing-scams-cost-american-businesses-half-a-billion-dollars-a-year/#74bff6a53fa1. [Accessed: 22-Nov-2017].

[30]
=
https://www.gartner.com/newsroom/id/492157. [Accessed: 22-Nov-2017].

[31]

[32] Anti-
no. June, pp. 1 13, 2017.

[33]
= = -and-
phishing-in-q3-2017/82901/. [Accessed: 23-Nov-2017].

[34] -phishing emai =

[35]
Commun. ACM, vol. 50, no. 10, pp. 94 100, 2007.

[36] =
Beyo *J. Digit. Forensic Pract.*, vol. 1, no. 3, pp. 245 260, 2006.

[37] =
Proc. Int. Conf. Dependable Syst. Networks, pp. 3 12, 2009.

[38]
poisoning in file sharing peer-to- *Proc. 6th ACM Conf. Electron. Commer.*, pp. 68 77, 2005.

[39] = = rg/phishing-techniques. [Accessed: 21-Nov-2017].

[40] N. Adhikary, R. Shrivastava, A. Kumar, S. K. Verma, M. Bag, and V. Singh,
Int. J. Comput. Netw. Inf. Secur., vol. 4, no. 5, pp. 13 21, 2012.

[41]

[42]

Conf. Res. Pract. Inf. Technol. Ser., vol. 54, pp. 221 232, 2006.

[43] = = - techniques. [Accessed: 26-Nov-2017].

[44] = https://www.pcworld.com/article/135293/article.html. [Accessed: 27-Nov-2017].

[45] S. Abu- *Proc. anti-phishing Work. groups 2nd Annu. eCrime Res. summit - G 4* , pp. 60 69, 2007.

[46] A. Berghol Detection using Model- *Ceas*, no. January 2008, 2008.

[47] *Proc. anti-phishing Work. groups 2nd Annu. eCrime Res. summit - G 4* , pp. 37 44, 2007.

[48] *ACM Trans. Internet Technol.*, vol. 10, no. 2, pp. 1 31, 2010.

[49] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, = *Proc. ACM CHI 2007 Conf. Hum. Factors Comput. Syst.*, vol. 1, pp. 905 914, 2007.

[50] J. S. *Proc. Second Symp. Usable Priv. Secur. - WS TW 4:* , p. 79, 2006.

[51] *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, no. November 2005, pp. 581 590.

[52] Available: https://www.wombatsecurity.com/. [Accessed: 29-May-2017].

[53] A. A. at al. Kumaragu *Carnegie Mellon Univ.*, p. 2.

[54] Available: http://phish-

[56] Empirical An *6th Conf. Email Anti-Spam, ser. GI EW 4* , 2009.

[57] = = - browsing/. [Accessed: 28-Nov-2017].

[58] = = tank.com/index.php. [Accessed: 28-Nov-2017].

[59] = https://www.opendns.com/. [Accessed: 01-Dec-2017].

[60] = blac *Proc. - IEEE INFOCOM*, 2010.

[61] - = = - phishing. [Accessed: 29-Nov-2017].

[62] = ug-in for *M WEE 4 - 2nd Int. Conf. Internet Multimed. Serv. Archit. Appl.*, pp. 1 6, 2008.

[63] = *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5143 LNCS, pp. 182 186, 2008.

[64] = -based approach to detecting *G 2 h h f*, pp. 639 648, 2007.

[65] *IEEE Trans. Dependable Secur. Comput.*, vol. 3, no. 4, pp. 301 311, 2006.

[66] E. Medvet, E. Kirda, and C. Kruegel, -similarity- *Proc. 4th Int. Conf. Secur. Priv. Commun. netowrks - W g 2 4* , p. 1, 2008.

[67] = *eCrime Res. Summit, eCrime*, pp. 1 12, 2012.

[68] *Mach. Learn.*, vol. 45, no. 1, pp. 5 32, 2001.

[69] = - *HumanComputer Interact. Inst.*, no. Paper 76, 2006.

[70] [Online]. Available: http://investors.proofpoint.com/releasedetail.cfm?releaseid=819799. [Accessed: 03-

Dec-2017].

[71]
IEEE, no. March, pp. 1 7, 2013.

[72]
Computer (Long. Beach. Calif)., vol. 36, no. 6, pp. 47 56, 2003.

[73] = *IEEE Softw.*, vol. 8, no. 1, pp. 32 41, 1991.

[74] =
https://www.smartsheet.com/all-risk-assessment-matrix-templates-you-need.
[Accessed: 06-Dec-2017].

[75] = = = -Dec-2017].

[76] =
https://angularjs.org/. [Accessed: 09-Dec-2017].

[77] vailable:
https://www.mongodb.com/. [Accessed: 09-Dec-2017].

[78] = = = -Dec-2017].

[79] = =
https://www.jetbrains.com/webstorm/. [Accessed: 09-Dec-2017].

[80] -based Software

[81] - =
https://expressjs.com/. [Accessed: 10-Dec-2017].

[82] - *Int. Conf.*
, no. July, 2011.

[83]
Phishing Detection: A Computational I *J. Inf. Knowl. Manag.*, vol. 15, no. 4, p. 1650042, 2016.

[84] - =
https://www.alexa.com/siteinfo. [Accessed: 14-Dec-2017].

[85]
Available: https://moz.com/products/api. [Accessed: 14-Dec-2017].

[86]
Available: https://www.mywot.com/en/reputation-api. [Accessed: 14-Dec-2017].

[87] = https://www.npmjs.com/. [Accessed: 20-Dec-2017].

[88] = = [Accessed: 20-Dec-2017].

[89] Online]. Available: https://www.npmjs.com/package/alexarank. [Accessed: 20-Dec-2017].

[90] - Available: https://www.npmjs.com/package/tall. [Accessed: 21-Dec-2017].

[91] icate &amp; Digital Certificate Authority - Available: https://www.ssl.com/. [Accessed: 20-Dec-2017].

[92] - Whois Lookup - = https://www.whoisxmlapi.com/. [Accessed: 20-Dec-2017].

[93] - = https://mochajs.org/. [Accessed: 09-Jan-2018].

[94] = = = -Jan-2018].

[95] - Standalone test spies, stubs and mocks for JavaScript. Works with any = = = -Jan-2018].

[96] = https://www.heroku.com/. [Accessed: 10-Jan-2018].

[97] = https://www.quantcast.com/top-sites. [Accessed: 17-Dec-2017].

[98] Available: https://github.com/faizann24/Using-machine-learning-to-detect-malicious-URLs. [Accessed: 17-Dec-2017].

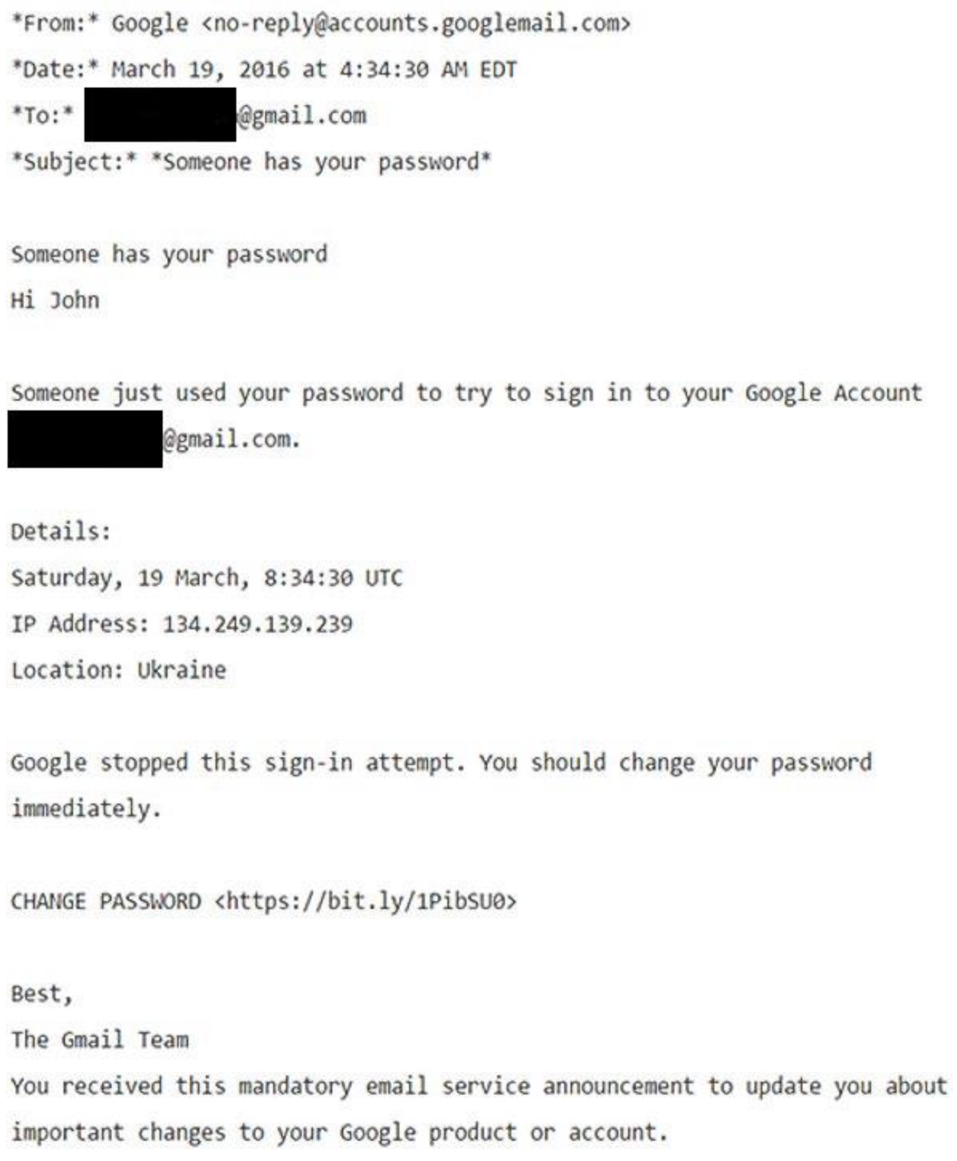
[99] based on self- *Neural Comput. Appl.*, vol. 25, no. 2, pp. 443 458, 2014.

[100] X. M. Choo, K. L. Chiew, D. H. A. Ibrahim, N. Musa, S. N. Sze, and W. K. Tiong, - *J. Theor. Appl. Inf. Technol.*, vol. 91, no. 1, pp. 101 106, 2016.

[101] G. Ramesh, I. Krishnamurthi, and K. S. S. Kumar *Decis. Support Syst.*, vol. 61, no. 1, pp. 12 22, 2014.

Appendix A

Spare Phishing Email Example



From: Google <no-reply@accounts.googlemail.com>
Date: March 19, 2016 at 4:34:30 AM EDT
To: [redacted]@gmail.com
Subject: *Someone has your password*

Someone has your password
Hi John

Someone just used your password to try to sign in to your Google Account
[redacted]@gmail.com.

Details:
Saturday, 19 March, 8:34:30 UTC
IP Address: 134.249.139.239
Location: Ukraine

Google stopped this sign-in attempt. You should change your password
immediately.

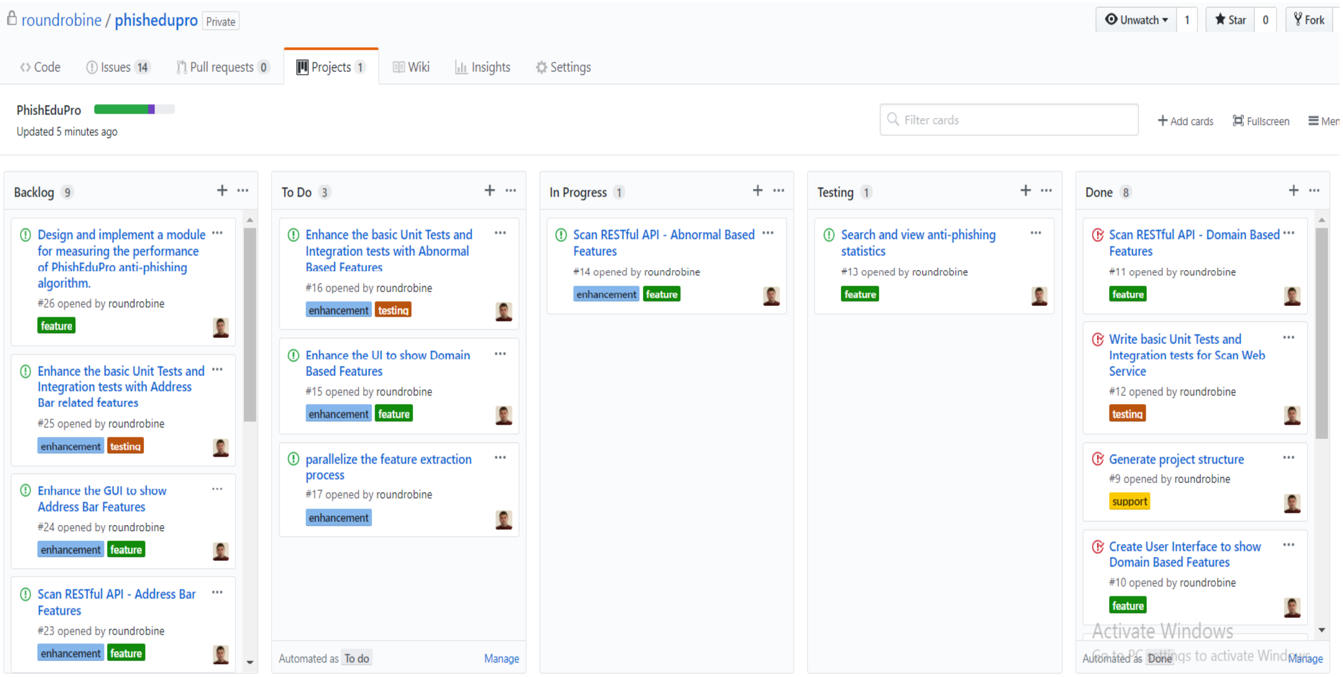
CHANGE PASSWORD <https://bit.ly/1PibSU0>

Best,
The Gmail Team
You received this mandatory email service announcement to update you about
important changes to your Google product or account.

Figure A.1: Text of spear-phishing email sent to John Podesta, the chairman of the 2016 Clinton presidential campaign

Appendix B

GitHub Project Management Tool



PhishEduPro

Updated 5 minutes ago

Backlog 9

- Design and implement a module for measuring the performance of PhishEduPro anti-phishing algorithm.
  #26 opened by roundrobine
  feature
- Enhance the basic Unit Tests and Integration tests with Address Bar related features
  #25 opened by roundrobine
  enhancement testing
- Enhance the GUI to show Address Bar Features
  #24 opened by roundrobine
  enhancement feature
- Scan RESTful API - Address Bar Features
  #23 opened by roundrobine
  enhancement feature

To Do 3

- Enhance the basic Unit Tests and Integration tests with Abnormal Based Features
  #16 opened by roundrobine
  enhancement testing
- Enhance the UI to show Domain Based Features
  #15 opened by roundrobine
  enhancement feature
- parallelize the feature extraction process
  #17 opened by roundrobine
  enhancement

Automated as To do Manage

In Progress 1

- Scan RESTful API - Abnormal Based Features
  #14 opened by roundrobine
  enhancement feature

Testing 1

- Search and view anti-phishing statistics
  #13 opened by roundrobine
  feature

Done 8

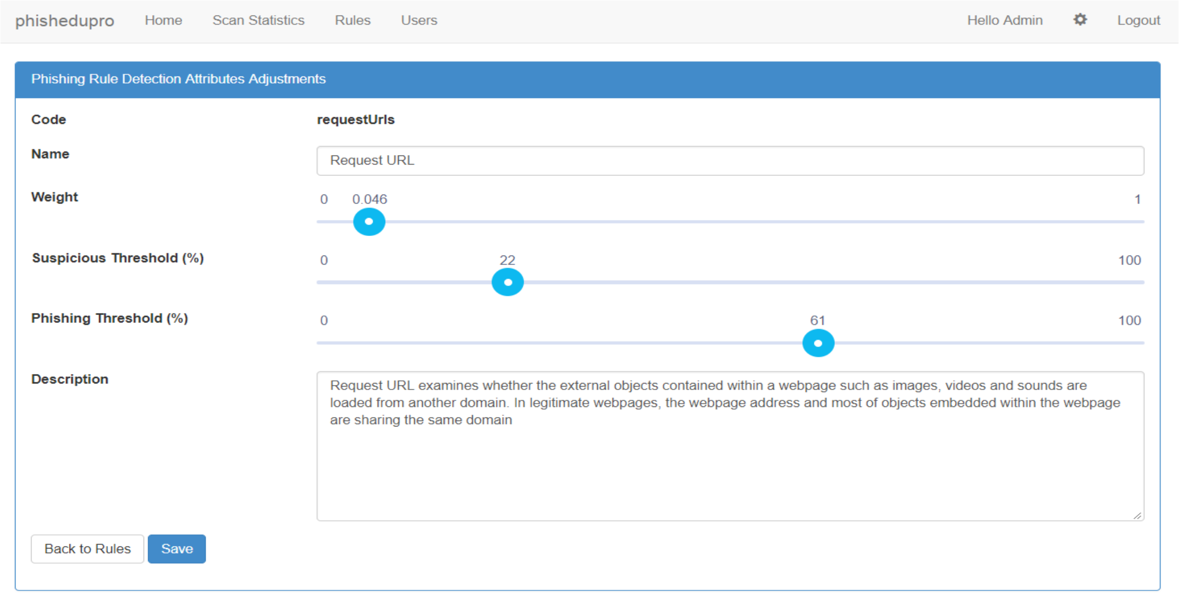
- Scan RESTful API - Domain Based Features
  #11 opened by roundrobine
  feature
- Write basic Unit Tests and Integration tests for Scan Web Service
  #12 opened by roundrobine
  testing
- Generate project structure
  #9 opened by roundrobine
  support
- Create User Interface to show Domain Based Features
  #10 opened by roundrobine
  feature

Automated as Done Manage

Figure B.1: A random state of the project management dashboard used to track the development progress of PhishEduPro platform

Appendix C

Additional Screenshots of PhishEduPro Platform



Phishing Rule Detection Attributes Adjustments

Code requestUrls

Name Request URL

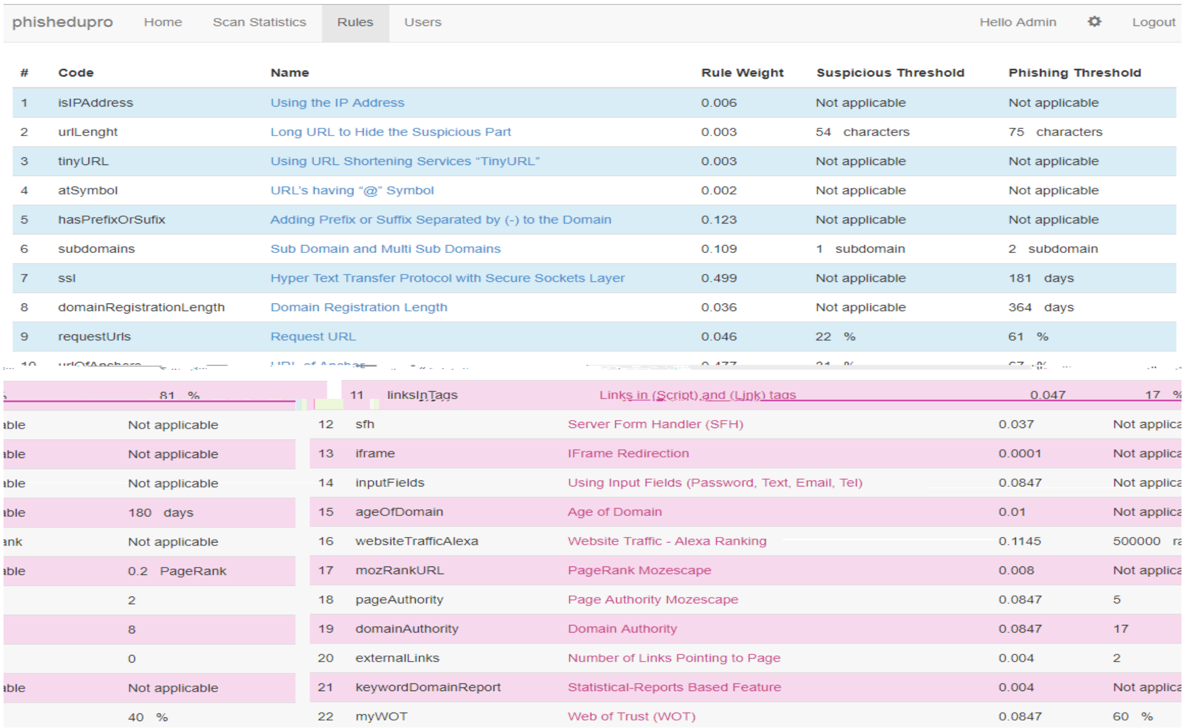
Weight 0 0.046 1

Suspicious Threshold (%) 0 22 100

Phishing Threshold (%) 0 61 100

Description Request URL examines whether the external objects contained within a webpage such as images, videos and sounds are loaded from another domain. In legitimate webpages, the webpage address and most of objects embedded within the webpage are sharing the same domain

Figure C.2: Screenshot from Rule adjustment module illustrating thresholds tuning of Request URL feature

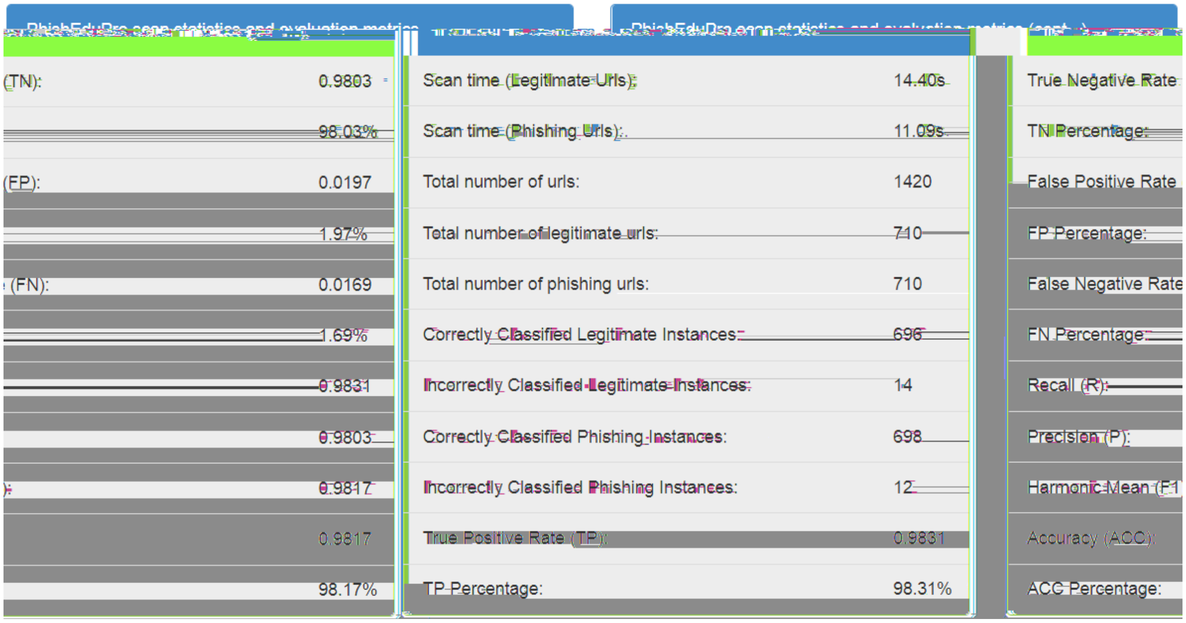


#	Code	Name	Rule Weight	Suspicious Threshold	Phishing Threshold
1	isIPAddress	Using the IP Address	0.006	Not applicable	Not applicable
2	urlLenght	Long URL to Hide the Suspicious Part	0.003	54 characters	75 characters
3	tinyURL	Using URL Shortening Services "TinyURL"	0.003	Not applicable	Not applicable
4	atSymbol	URL's having "@" Symbol	0.002	Not applicable	Not applicable
5	hasPrefixOrSufix	Adding Prefix or Suffix Separated by (-) to the Domain	0.123	Not applicable	Not applicable
6	subdomains	Sub Domain and Multi Sub Domains	0.109	1 subdomain	2 subdomain
7	ssl	Hyper Text Transfer Protocol with Secure Sockets Layer	0.499	Not applicable	181 days
8	domainRegistrationLength	Domain Registration Length	0.036	Not applicable	364 days
9	requestUrls	Request URL	0.046	22 %	61 %
10	urlOfAnchors	URL of Anchor	0.477	31 %	67 %
11	linksInTags	Links in (Script) and (Link) tags	0.047	17 %	81 %
12	sfh	Server Form Handler (SFH)	0.037	Not applicable	Not applicable
13	iframe	IFrame Redirection	0.0001	Not applicable	Not applicable
14	inputFields	Using Input Fields (Password, Text, Email, Tel)	0.0847	Not applicable	Not applicable
15	ageOfDomain	Age of Domain	0.01	Not applicable	180 days
16	websiteTrafficAlexa	Website Traffic - Alexa Ranking	0.1145	500000 rank	Not applicable
17	mozRankURL	PageRank Mozescape	0.008	Not applicable	0.2 PageRank
18	pageAuthority	Page Authority Mozescape	0.0847	5	2
19	domainAuthority	Domain Authority	0.0847	17	8
20	externalLinks	Number of Links Pointing to Page	0.004	2	0
21	keywordDomainReport	Statistical-Reports Based Feature	0.004	Not applicable	Not applicable
22	myWOT	Web of Trust (WOT)	0.0847	60 %	40 %

Figure C.3: Screenshot of the 22 rules and features which are used in the anti-phishing classification process of the PhishEduPro algorithm

Appendix D

Screenshot of Statistics Module



PhishEduPro scan statistics and evaluation metrics

PhishEduPro scan statistics and evaluation metrics (cont...)

(TN):	0.9803	Scan time (Legitimate Urls):	14.40s	True Negative Rate
	98.03%	Scan time (Phishing Urls):	11.09s	TN Percentage:
(FP):	0.0197	Total number of urls:	1420	False Positive Rate
	1.97%	Total number of legitimate urls:	710	FP Percentage:
(FN):	0.0169	Total number of phishing urls:	710	False Negative Rate
	1.69%	Correctly Classified Legitimate Instances:	696	FN Percentage:
	0.9831	Incorrectly Classified Legitimate Instances:	14	Recall (R):
	0.9803	Correctly Classified Phishing Instances:	698	Precision (P):
):	0.9817	Incorrectly Classified Phishing Instances:	12	Harmonic Mean (F1
	0.9817	True Positive Rate (TP):	0.9831	Accuracy (ACC):
	98.17%	TP Percentage:	98.31%	ACC Percentage:

Figure C.1: Screenshot from the final state of the statistics module showing PhishEduPro platform evaluation metrics in real-time