DETECTION AND REMEDIATION OF TURKISH PHISHING WEBSITES

by
BARIŞ SERMET

Submitted to the Graduate School of Cyber Security
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
October 2021

DETECTION AND REMEDIATION OF TURKISH PHISHING WEBSITES

Approved by:

Date of Approval: 08/10/2021

ABSTRACT

DETECTION AND REMEDIATION OF TURKISH PHISHING WEBSITES

BARIŞ SERMET

CYBER SECURITY MSc. THESIS, OCTOBER 2021

Thesis Supervisor: Prof. Dr. ERKAY SAVAŞ

Keywords: Turkish phishing website, machine learning, phishing detection, remediation, abuse reporting

The rapid development of cyber technologies has made our life easier. Activities such as online banking, e-commerce and, social media have started to become a part of our lives; however, more threat surface has also emerged for attackers to leverage due to them. Phishing attacks are one of the most preferred and well-known cybercrimes by attackers. Attackers use social engineering techniques to obtain victims' personal information through phishing websites. To remedy this, researchers, on the other hand, have been trying to detect phishing websites by using several approaches. However, after detecting a phishing website, a significant step is to remediate phishing websites as soon as possible to minimize the damage. In this study, we used hybrid CNN+BiLSTM and Random Forest classifiers to detect Turkish phishing websites in real-time. We measured the impact of notifying phishing websites to targeted hosting providers compared to Turkish CERT, USOM. Additionally, we measured phishing website blocking time for three Turkish ISPs. Our work shows no statistically significant difference between sending abuse notifications to USOM and sending abuse notifications to hosting providers. On the other hand, we found that one ISP's blocking time is slower than another.

ÖZET

TÜRK OLTALAMA SİTELERİNİN TESPİTİ VE ENGELLENMESİ

BARIŞ SERMET

SİBER GÜVENLİK YÜKSEK LİSANS TEZİ, EKİM 2021

Tez Danışmanı: Prof. Dr. ERKAY SAVAŞ

Anahtar Kelimeler: Türk oltalama sitesi, makine öğrenmesi, oltalama tespiti, engelleme, kötüye kullanım bildirimi

Siber teknolojilerin hızlı gelişimi hayatımızı kolaylaştırdı. İnternet bankacılığı, e-ticaret ve sosyal medya gibi faaliyetler hayatımızın bir parçası olmaya başladı, ancak saldırganların yararlanabileceği daha fazla tehdit yüzeyi ortaya çıktı. Oltalama saldırıları, saldırganların en çok tercih ettiği ve en bilinen siber suçlardır birisidir. Saldırganlar, sosyal mühendislik teknikleri uygulayarak kurbanların kişisel bilgilerini oltalama web siteleri aracılığı ile ele geçirirler. Araştırmacılar, çeşitli yaklaşımlar kullanarak oltalama sitelerini tespit etmeye çalışıyorlar. Fakat, oltalama sitelerinin tespit edilmesinden sonra zararı en aza indirmek için gerekli olan bir diğer adım ise bu tür sitelerin mümkün olan en kısa sürede engellenmesidir. Bu çalışmada, Türkiye'deki oltalama sitelerini gerçek zamanlı olarak tespit etmek için hibrit Evrişimli Sinir Ağları+Çift Yönlü Uzun Kısa Süreli Bellek ve Rassal Orman sınıflandırıcıları kullandık, ardından tespit edilen oltalama sitelerini hedeflenen barındırma sağlayıcılarına bildirmenin etkisini Türkiye'deki Bilgisayar Acil Müdahale Ekibi, USOM ile karşılaştırdık. Ayrıca üç tane Türk İnternet Servis Sağlayacısının USOM'un kara listesinde görüntülenen oltalama tehdidine karşı engelleme sürelerini ölçtük. Çalışmamız, USOM'a kötüye kullanım bildirimleri göndermek ile barındırma sağlayıcılarına kötüye kullanım bildirimleri göndermek arasında istatistiksel olarak önemli bir fark olmadığını göstermektedir. Öte yandan, bir İnternet Servis Sağlayısının engelleme süresinin bir diğerinden daha yavaş olduğunu bulguladık.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisors, Erkay Savaş and Orçun Çetin, for their continuous support, encouragement, and guidance.

Besides my advisors, I would like to thank my friends who always be there for me.

Finally, I would like to thank my family, Buket, Hande, and Mehmet. Mom, sister, and dad, thank you for always supporting me throughout my life.

to my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Due to the rapid development of cyber technologies, activities such as social media, e-commerce, and online banking have started to take important roles in our daily lives. Although human life has become easier as a result of these developments, more threat surface has also emerged for exploitation by attackers.

Phishing is a form of cyber threat that applies both social engineering and technical fraud to steal consumers' sensitive data such as personal information and financial account credentials (Anti-Phishing Working Group, 2020). Attackers exploit users utilizing social engineering techniques that are frequently employed to manipulate users and retrieve their personal information (almost) voluntarily rather than finding a weakness in the computer systems. Therefore, phishing attacks are one of the most convenient ways preferred by many among other types of cyber-crimes, in which the attacker uses e-mails, SMS, voice calls, or social media to get into contact with victims (Gupta, Singhal & Kapoor, 2016).

As end-users are usually the main target in this type of attack, raising security awareness for phishing attacks can reduce the loss; however, it is not entirely possible to prevent users from falling into such attacks (Greene, Steves & Theofanos, 2018). Moreover, it has been observed that the success rate of phishing attacks is increased when attackers also take into consideration the victims' personality traits such as Machiavellism, narcissism, and psychopathy (Curtis, Rajivan, Jones & Gonzalez, 2018). Due to the COVID-19 pandemic, attackers exploited people's fear and anxiety to deploy phishing attacks (Pranggono & Arabo, 2021) by using covid-related keywords. Internet Crime Complaint Center (2020) reports that, in 2020, phishing was the most common cybercrime and nearly doubled in frequency compared to the previous year, and more than $54 million loss incurred as a result of phishing attacks.

In recent years, researchers have been trying to detect phishing websites by adopting several approaches utilizing machine learning classifiers to detect phishing websites such as those that are URL-based (Sahingoz, Buber, Demir & Diri, 2019), content-

based (Zhang, Liu, Chow & Liu, 2011), and hybrid-based (Zhang, Jiang, Chen & Li, 2017) models. The accuracy of phishing detection models critically depends on the extracted features and data set used in the training phase. Nevertheless, such phishing detection methods for developing countries such as Turkey might not be as effective as they are expected, which may be resulting from the fact that victims in developing countries are deceived with websites that are in the local language. And those websites are unknown to spam filters because they were trained with data sets of international companies.

Detecting phishing is only one of the steps to stop the attacker from luring victims. After detecting a phishing website, a significant and perhaps more vital step is to remediate those websites as soon as possible to prevent end-users from being deceived and minimize their loss. Also, Jhaveri, Cetin, Gañán, Moore & Eeten (2017) emphasize the importance of abuse reporting to mitigate cybercrimes. Hosting providers are responsible for the websites they host. If any malicious activity is found, the hosting provider takes down the website; thus, remediation is completed. On the other hand, ISPs (Internet Service Providers) can block these websites using blacklists to prevent users from visiting malicious websites. In Turkey, USOM[1] (Ulusal Siber Olaylara Müdahale Merkezi), National CERT (Computer Emergency Response Team) of Turkey displays a list of malicious websites[2] so that Turkish ISPs can block these websites and make the material inaccessible to the Internet users. However, hosting providers and ISPs receive a high amount of abuse reports each day. They are receiving not only phishing attacks but also alleged infringements of intellectual property and issues of content regulation (Jhaveri et al., 2017) that may lead to taking late actions by the abuse receivers. Also, a survey[3] shows that 27% of IT professionals are receiving more than 1 million security alerts each day.

This thesis presents a content-based machine learning classifier to detect Turkish phishing websites and conduct a randomized controlled experiment comparing different remediation methods for detected phishing websites. In the detection part, Random Forest and hybrid CNN+BiLSTM algorithms are used to build a classification model. The proposed method uses three content-based features: HTML text, HTML form actions and input names, and HTML tags.

In the remediation part, first, the affected hosting provider will be found and then the provider will be notified about phishing websites via e-mail. This will be compared to

[1] https://www.usom.gov.tr

[2] https://www.usom.gov.tr/url-list.txt

[3] https://www.imperva.com/blog/27-percent-of-it-professionals-receive-more-than-1-million-security-alerts-daily/

another group of phishing websites that were reported to USOM. The effectiveness of notifications made to USOM will be compared to the second treatment group where notifications were sent to affected hosting providers. Additionally, the Turkish ISPs' reaction time to the phishing threat displayed in USOM's list will be measured. To this end, the three largest Internet providers in Turkey were picked, and their blacklisting and censoring time were measured.

To conduct a remediation experiment, unblocked and accessible phishing websites are required. To find unblocked and accessible phishing websites, CZDS (Centralized Zone Data Service)[4] that was provided by ICANN (Internet Corporation for Assigned Names and Numbers)[5] was used. CZDS shares daily updated zone files that contain the mapping of domain names, associated name server names, and IP addresses for those name servers. After receiving zone files from CZDS, files were processed to obtain daily registered domains. Phishing websites that were detected by our detection model were used to conduct remediation experiments.

A hybrid CNN+BiLSTM and a Random Forest classifier were used to detect Turkish phishing websites. We used the Area Under Curve (AUC) score of Receiver Operating Characteristic (ROC) and accuracy to evaluate our approach. Our results show that more than 94% AUC and 91% accuracy were obtained from each model using PRODAFT's dataset. The dataset consists of 1500 Turkish phishing and 1500 legitimate websites. For remediation experiments, daily-registered domains shared by ICANN were used and 210 Turkish phishing websites were detected in the time interval from 25 May 2021 until 17 September 2021. From detected Turkish phishing websites, 104 of them are reported to USOM, while 106 of them are reported to the hosting providers. The log-rank test shows that there are no significant differences between each group. On the other hand, our results indicate that one ISP's blocking time is slower than another.

Real-time detection and remediation of phishing websites have not been studied before. Moreover, phishing websites that target developing countries, such as Turkey, are not very well known for spam filters, and they can easily evade security mechanisms. Therefore, the purpose of this thesis can be summarized as follows:

- We designed and implemented a phishing detection model to detect newly registered Turkish phishing websites in real-time.
- We conducted a remediation experiment on the impact of sending abuse notifications to hosting providers compared to sending them to USOM.

[4]https://czds.icann.org

[5]https://www.icann.org

- We measured and compared three Turkish ISP's remediation efficiency for malicious threats included on USOM's blacklist.

The thesis is structured as follows: Chapter II presents background information relevant to the field of this study. Chapter III refers to the studies in the literature related to the subject. Chapter IV describes the methodology of the detection system and remediation experiment. Chapter V shares the results of the experiment. Chapter VI discusses the results and findings of the real-time detection and remediation experiment. Chapter VII concludes the thesis, proposes ideas for future work, and explains the limitations of the experiment.

2. BACKGROUND

2.1 Phishing Attacks

A phishing attack is a form of cybercrime that has become part of our lives since 1995, starting with America Online (AOL), which is used to describe the attacker who "lures" the victims to "fishes" for personal information and credentials using "baits" (Chiew, Yong & Tan, 2018; Lance James, 2006). There may be different types of phishing attacks as briefly described in the following.

Clone phishing is a type of phishing attack, whereby an attacker clones a legitimate website that is usually visited by end-users. The appearance of the mimicked website looks the same as a legitimate website. Still, attackers can receive the users' information on the back-end and record them into a database using their server.

Spear phishing targets a specific individual, company, or business. Attackers use e-mail, social media, instant messaging, or other platforms to get users' personal information, bank credentials, or sensitive information about the businesses.

Whaling is a type of spear phishing, but it targets the C-level (CEO, CTO, CFO, etc.) or high-profiled people who have access to sensitive information about the company or organization. While the success rate of this attack is lower than others, it causes much more harm.

Phone phishing, or **vishing**, is a type of phishing attack where attackers make phone calls to the victims and impersonate themselves as a legitimate authority. Attackers trick victims into giving their personal information on the phone or visiting malicious web pages.

SMS phishing, or **smishing**, is a type of phishing attack in which an attacker sends SMS messages and invites them to click the malicious URL, to call a phone

number, or to contact an e-mail address that is sent to victims.

Email phishing, where an attacker sends a malicious e-mail to victims, to make them click on the malicious URL in the e-mail content.

Social media phishing uses the power of social media and advertisements that showed in social media feeds. Attackers impersonate themselves as someone else or a brand to lure victims and make them click on malicious URLs. Also, attackers give advertisements of their phishing websites to social media companies to get clicked by targeted users.

Phishing attacks, mentioned above, mostly use phishing websites created by malicious actors. Clone phishing is highly popular since there is no need for sophisticated background information. On the other hand, nowadays instead of cloning legitimate websites, attackers clone well-prepared phishing websites from other phishing websites. The well-prepared phishing websites are designed to lure people through their URL and content. Attackers choose specific keywords while registering a domain name and content containing logos and/or images that people think are authentic by a legitimate company. Also, phishing website content and URL may change due to the circumstances of the world or a specific country. Since the beginning of the COVID-19 pandemic, 30.103 covid-related domains have been registered according to Check Point researchers[1], many of which are likely to be used for illegal activities. Besides, there are more than 2000 domain names blacklisted in USOM's blacklist containing covid-related keywords.

Social engineering is frequently used in targeted phishing attacks since the attacker has to know the targeted individual or group of people's background information and their use of particular software such as e-mail services. Phishing webpage is sent to victims, and when the victim inputs personal credentials to the page, the attacker has access to victims' e-mails; consequently, attackers may gain sensitive information. On the other hand, an attacker may install a malicious program on the victim's device to gain control of the device. Malicious programs such as malware need detailed background information about operating systems, and it is more likely to do more damage.

Besides, all given phishing attacks use the power of social engineering techniques that are designed to manipulate people by attackers. In the form of information security, **social engineering** is a manipulation technique that is used by malicious actors to control and divert peoples' actions. Attackers use social engineering techniques

[1] https://blog.checkpoint.com/2020/04/02/coronavirus-update-in-the-cyber-world-the-graph-has-yet-to-flatten/

on people to "phish" them since the weakest chain of the security mechanism is human beings. Rader & Rahman (2015) claimed that social engineers are using the emotions of people to manipulate, and the most common manipulation methods rely on three emotions, curiosity, fear, and empathy.

2.2 Machine Learning for Phishing Website Detection

As discussed in previous sections, phishing attacks are snowballing, and attack methods and vectors change frequently. From past to present, security researchers applied various techniques to detect phishing websites. Although researchers find different ways to detect phishing websites, attackers find even more sophisticated methods to thwart them. In recent years, machine learning and deep learning-based approaches have fulfilled the needs in the detection of phishing attacks due to their capability to learn from data. In this section, we briefly discuss various machine learning and deep learning algorithms that we proposed to detect phishing websites.

Decision Tree is a supervised learning method that is used for classification and regression. It uses a tree-based model, in which each node indicates a test of feature and each descending branch from a given node specifies the outcome of the feature testing (Mitchell, 1997). Also, each leaf node in the tree represents the class label of the instance, and a branch of a node to another node is a decision rule for a particular node.

Random Forest is a supervised learning method that uses multiple decision trees to overcome classification and regression problems. Overfitting may occur in a single decision tree due to noise in the data; however, random forest splits the data into multiple decision trees and aggregates results from each decision tree using the mean of each decision tree (Biau & Scornet, 2016).

Convolutional Neural Network (CNN) is a deep learning algorithm that is mostly used in the image processing field. However, in recent years it has been used to classify texts as well. It consists of a minimum of two layers; namely, the input layer and the output layer. Yet, depending on the problem hidden layers can be added between input and output layers.

Long Short-Term Memory (LSTM) is a deep learning algorithm that uses artificial recurrent neural network (RNN) architecture. LSTM is crafted to classify

time-series data. It consists of a cell, an input gate, an output gate, and a forget gate. Each cell is connected in sequence and the next cell can get the information about the previous cell. And the power of the forget gate is that it can know when and where to throw away or keep the information.

Bidirectional Long Short-Term Memory (Bi-LSTM) consists of two LSTM models. The first LSTM model takes input in a forward direction and the second is backward direction. This approach gives an ability to know both past and future in time-series data. Thus it increases the amount of information available to the network.

3. LITERATURE REVIEW

In this section, existing approaches and proposed methods for detecting phishing websites that use machine learning models are reviewed. Then we discuss existing remediation and mitigation techniques, strategies, and experiments on the effectiveness of security notifications.

3.1 Machine Learning Approaches on Detecting Phishing Websites

Identifying a website, whether it is phishing or not, is a classification problem. In recent years, machine learning approaches have been used to solve the phishing website classification problem. Since machine learning models depend on dataset and features, researchers found different features for identification and used different datasets for training and testing.

The Bayesian approach was used by Zhang et al. (2011) for content-based phishing detection. They proposed an algorithm that combines text and image classifier results that measure the similarity between the protected web pages, i.e., a legitimate web page that can probably be targeted by attackers. Their text classifier uses naive Bayes rules to find the probability that a given website is phishing or not, and the image classifier applies earth mover's distance to calculate the similarity between the input web page image and the protected web page image. Then, they apply both weighted and proposed Bayesian approaches to their combining algorithm. They conclude that the Bayesian approach outperforms the weighted approach. In their experiment, eight web pages were selected as protected web pages to find similarities and they split their dataset into 8 sub-datasets, phishing websites were collected from PhishTank and 10.272 legitimate websites were retrieved from Google. Their CCR (correct classification rate) changes between 97.46% to 100% for 8 protected websites.

Xiang, Hong, Rose & Cranor (2011) found that machine learning approaches in the literature suffer from the paucity of proposed features and high FP (false positive) rate. They proposed a hybrid model with a rich set of features to achieve high TP (true positive) and filtering algorithms to keep FP at a low level. Their features are categorized as URL-based, HTML-based, and Web-based, in the latter of which they use third-party services and search engines in Web-based features. To lower the FP rate and speed up the algorithm, they proposed two filtering algorithms, hash-based similarity for HTML content after preprocessing it and login form detection. They used 8118 phishing and 4883 legitimate webpages and as a result, they achieved over 92% TP on unique testing phishing websites and 04.% FP with 10% training phishing websites.

Mohammad, Thabtah & McCluskey (2014) proposed a model based on self-structuring neural networks. They extracted 17 features along with URL, HTML source, DNS record, age of the domain, and website traffic to train the proposed neural network. A backpropagation algorithm was applied to the system, and they achieved 94.07% accuracy using 600 legitimate and 800 phishing websites. However, their proposed system is a bit complex.

El-Alfy (2017) developed a system that uses the power of PNN (probabilistic neural network) with k-medoids clustering. PNN may require a high amount of time and space for big data; however, using k-medoids reduces their dimensionality of the feature space to 40%. They categorized 30 features into 4 categories; namely, address-bar-based, abnormal-based, HTML and JavaScript-based, and domain-based. They used 4898 legitimate and 6157 phishing websites to train and test their model, as a result, they achieved 96.79% accuracy.

Zhang et al. (2017) proposed a two-staged ELM (extreme learning machine) method that uses hybrid features consisting of URL-based, Web-based, rule-based, and textual content-based. In the first stage, they constructed an ELM model to predict a label of textual content and OCR (Optical Character Recognition) was used to extract text from images. In the second stage, linear combination model-based ensemble ELM models were constructed based on hybrid features. They applied a model on two different datasets; one is in English and the other in Chinese. They achieve 99.04% accuracy on the English dataset and 97.50% accuracy on the Chinese dataset.

URL-based features are used to identify phishing websites. Jain & Gupta (2018a) developed a URL feature-based detection system using an SVM classifier. They used 14 features based on URL and trained their SVM classifier with more than 33.000 phishing and legitimate websites that are taken from PhishTank and Yahoo

Directory. They compare SVM classifiers with Naive Bayes and as a result, they achieve more than 90% accuracy with SVM.

Feng, Zhou, Shen, Yang, Han & Wang (2018) proposed a novel neural network classification method using the Monte Carlo algorithm and designed their model by the risk minimization principle. Their dataset consisted of 11.055 websites collected from PhishTank, MillerMiles, and Google search engine. They adopted the same features as Mohammad et al. (2014). Their model achieved a 97.71% accuracy rate and 1.7% FP rate in the experimental studies.

Jain & Gupta (2018b) proposed language independent model. Their model uses a total of 19 features based on URL and HTML source code. Their training dataset consists of 2141 phishing and 1918 legitimate websites which were collected from PhishTank, Openphish, and Alexa. They achieved a 99.39% true positive rate and 99.09% accuracy.

Chiew, Tan, Wong, Yong & Tiong (2019) proposed a hybrid ensemble of feature selection framework based on machine learning. Their framework consists of two cycles, the data perturbation cycle, and the function perturbation cycle. In the first cycle, a novel Cumulative Distribution Function gradient (CDF-g) algorithm was used to extract the primary feature set, and in the second cycle, extracted feature sets are fed into a data perturbation ensemble to obtain the secondary feature subset. The model achieves a 96.17% accuracy rate when it is integrated with the Random Forest classifier.

Sahingoz et al. (2019) proposed a model that uses 40 NLP-based features that are extracted from the URL. They compared 7 different algorithms along with NLP-based, word vector, and hybrid features using a publicly available dataset containing 37.175 phishing URLs along with 36.400 legitimate URLs. They achieve 97.98% accuracy with Random Forest using NLP-based features.

Jain & Gupta (2019) came up with a client-side solution to detect phishing websites using hyperlinks found in HTML source code. Their proposed method has divided hyperlinks into 12 different categories and uses them as features. Their approach is language and third-party service independent, and results in more than 98.39% TP rate along with a 1.52% FP rate using the Logistic Regression classifier. Dataset consists of 2544 phishing and legitimate websites.

3.2 Remediation and Mitigation of Malicious Websites and Vulnerable

Servers

Remediation and mitigation of malicious websites and vulnerable servers are crucial to minimizing the damage. Naturally, these steps can only be taken after detecting malicious websites. In recent years, security researchers have been trying to find an effective way to remediate and mitigate malicious websites.

Moore & Clayton (2007) analyzed the impact of takedown actions on phishing websites and concluded that this is part of mitigation. However, this issue cannot be completely resolved as it will never be instantaneous.

Nappa, Rafique & Caballero (2013) issued abuse reports of 19 long-lived malware distributing websites. They analyzed the abuse reporting process using interactions for both ISPs and hosting providers. Results show that the probability of the action being taken on the report when a reply is received is significantly higher than no reply is received.

Canali, Balzarotti & Francillon (2013) experiment to test the ability of the web hosting providers to detect compromised websites and react to user complaints. They ran 5 different attacks: a bot-like infection, a drive-by download, the upload of malicious files, an SQL injection stealing credit card numbers, and a phishing kit against 22 shared hosting providers. They start to send abuse notification e-mails to each provider on the 25th day of the experiment. Only 36% of the providers reacted to their abuse notification, and only one hosting provider could take action on time.

Kührer, Hupperich, Rossow & Holz (2014) worked on vulnerability notifications for amplification DDoS attacks. The remediation success was almost %95, however, they found that the connection between non-profit data clearing organizations and both hosting providers and CERTs was inadequate.

Hutchings, Clayton & Anderson (2016) conduct a qualitative study, using interviews of people actively engaged in website takedown (e.g., law enforcement, takedown services). They concluded that law enforcement is far more behind at takedown actions than commercial firms.

Cetin, Hanif Jhaveri, Gañán, van Eeten & Moore (2016) study measuring the role of sender reputation in abuse reporting of compromised websites. They send abuse notifications to hosting providers provided in WHOIS data from e-mail addresses belonging to an individual researcher (low reputation), academic institution (medium reputation), and an anti-malware organization (high reputation). They found that sender reputation does not affect cleanup rates. Also, they reported that sending notifications achieved much higher clean rates than doing nothing.

Li, Durumeric, Czyz, Karami, Bailey, McCoy, Savage & Paxson (2016) analyzed the impact of notification content and notification receiver (WHOIS abuse contact versus national CERTs versus US-CERT) of vulnerable systems. Results show that the most effective method is sending direct notification to WHOIS contacts with detailed information messages. Besides, most of the notification receivers did not take any action or the action taken was only partial.

Vasek, Weeden & Moore (2016) examined the impact of abuse data sharing among web hosting providers. They found, sharing abuse data has an expeditious effect of cleaning the reported malicious URLs, and reducing the probability of compromising again.

Cetin, Ganan, Korczynski & van Eeten (2017) experiment to find effects of the notification content, with the demonstration of the vulnerability on an affected system or just a static notification message, and notification receiver, domain owner, network operator, or name server operator, on remediation. They concluded that reaching out to nameserver operators directly had better results and that demonstration of the vulnerability has no effects.

Jhaveri et al. (2017) constructed an abuse reporting infrastructure to explain how direct and indirect remediation can help to protect online assets. Also, they examine the effectiveness and practicality of security notifications.

Stock, Pellegrino, Li, Backes & Rossow (2018) studied the effectiveness of alternative direct notification channels such as social media and phone for web vulnerability notifications. Yet, they conclude that alternative methods did not yield significant results.

For medium-sized ISP networks, it has been found that quarantining the vulnerable system and sending a notification e-mail has a better impact on remediation rates than sending a notification with no action or not sending a notification (Cetin, Ganán, Altena, Tajalizadehkhoob & van Eeten, 2018; Cetin, Ganán, Altena, Tajalizadehkhoob & van Eeten, 2019; Çetin, Ganán, Altena, Kasama, Inoue, Tamiya, Tie, Yoshioka & van Eeten, 2019). However, quarantining is not the general solution since it is not feasible to deploy it Internet-wide.

4. METHODOLOGY

4.1 Detection

In this study, we built hybrid CNN+BiLSTM and Random Forest classifiers to detect Turkish phishing websites.

In the following, we describe the utilized data set for training and testing our classifier and give brief explanations for feature selection, model creation, data collection for real-time detection, and system architecture.

4.1.1 Dataset

To perform successful phishing detection, a Turkish phishing website dataset was needed to train our model. However, public phishing URL depositories like PhishTank[1] and OpenPhish[2] mostly contain international company datasets, PayPal, Microsoft, Apple, etc. Since we aimed to find Turkish (Turkey-related) phishing websites, public phishing URL repositories and datasets from international sources did not support our experiment. For this study, PRODAFT, a cybersecurity company, supplied the Turkish phishing website dataset, which was collected between 20 August 2020 to 6 February 2021. The data that support the findings of this study are available from PRODAFT. Still, restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Dataset is available from the authors upon reasonable request and with

[1] http://phishtank.org/

[2] https://openphish.com/

permission of PRODAFT. The dataset contains URL, HTML, and screenshot of the 1500 phishing websites. All phishing contents are checked manually by investing screenshots of the websites. Legitimate websites are crawled from Bing Web Search[3] using Microsoft Playwright[4].

To train and test, the phishing dataset is divided into two parts (%75 and %25, respectively).

4.1.2 Feature Selection

For detection, we chose three content-based features: HTML form action and input attribute names, HTML text, and HTML tags. In the following, explanations as to why they have been chosen to identify Turkish phishing websites will be provided.

4.1.2.1 HTML Form Action and Input Name Values

An HTML form is one of the HTML properties designed to send user input when the form is submitted. It contains text fields, password fields, checkboxes and submit buttons, etc. HTML form has action and method fields[5]. The action field represents where to send the form data when the form is submitted using the submit button. Method field represents which HTTP method is to be used while sending that form data. In addition, JavaScript can be used to get user input when the user sends the form, even if the form has no action or method field.

Furthermore, the form input tag is designed to accept data from the user which has different type attributes, such as name, id, required, maxlength, minlength, etc.[6] Text fields and password fields are specified in the form input tag using the type attribute. Yet, when a form is submitted, the user data is sent in name-value pairs to the backend application. Attackers mostly use forms to get the user data, and these forms are different from legitimate website forms.

---

[3] https://www.bing.com/

[4] https://github.com/microsoft/playwright-python

[5] $https://www.w3schools.com/html/html_forms.asp$

[6] https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

HTML form action attribute specifies where to send the form data when the form is submitted. Based on our observations, form action attributes in phishing websites are different from legitimate form action attributes. And, in phishing websites form action attributes have some similarities. Some phishing websites use a form action attribute that contains a PHP file. That PHP file sends user input to the backend, then shows an error message about user input is invalid or shows another form to get more user data from the victim or shows an information message about their process that is still running. However, there are some cases where the form action attribute is null. In that case, an attacker gets the input data using JavaScript. This feature was also used in Jain & Gupta (2018b).

HTML form input is where the user puts information to the website. When a user clicks the submit button in that form, user data is sent to the server along with the name attribute, which is a way to understand which input field the user is typing data into. According to our observation, in forms that are contained in Turkish phishing websites, attackers use keywords or abbreviations for form input attributes, cc for the credit card number, cvv for card validation, tc or tckn for Turkish nationality number, etc. In that way, the attacker saves the user input with related keywords or abbreviations.

BeautifulSoup[7] was used to extract these values from a website form. Since the form action attribute may contain a URL, URL string extracted from the form action attribute value, and just the file name after "/" are kept. After extraction and pre-processing, form action and form input name values are added to a list, which is the first feature of our classifier.

4.1.2.2 HTML Text

To achieve a successful social engineering attack, attackers write similar texts to phishing websites that mimic an original one since the text gives textual information about the website content and has to look legitimate to a user.

BeautifulSoup was used to extract all the strings from HTML source code. Extracted strings were transformed to those with only lowercase letters with punctuation being removed. After that, they are added to a list in the same order, which is the second feature of our classifier.

[7] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

4.1.2.3 HTML DOM

DOM stands for Document Object Model, and it is a programming interface for HTML and XML documents with a tree structure where each node represents an object[8]. It is used to access objects in the document and can change their content, style, and structure. When a user opens a website from the browser, the browser converts an HTML file to a DOM tree. The DOM tree contains HTML tags as a node and the DOM tree is similar to what people see in their browsers. However, the DOM tree can be modified by JavaScript. That is why Playwright was used to obtain the DOM tree of the websites.

As it is mentioned before, attackers use clone phishing to deceive users. These clone phishing websites are either cloned from legitimate websites or existing phishing websites. In either case, newly opened phishing websites have a similar DOM tree structure as cloned ones.

BeautifulSoup was used to get HTML tags and extracted HTML tags added to a list, which is the third feature of our classifier.

4.1.3 Model Construction

To detect Turkish phishing websites, four different machine learning models are used. First, hybrid CNN+BiLSTM models are applied to each feature, resulting in binary output. Then, the Random Forest algorithm is applied to binary output results to get the label of a given website. In the following, adopted models and their parameters are explained.

4.1.3.0.1 CNN+BiLSTM

Since selected features are content-based, CNN and BiLSTM are used to classify text features, and each feature is trained and tested separately. Three different CNN + BiLSTM models are built from three features, and their results are in binary form, 0 for legitimate and 1 for phishing. Since each feature is text, they are pre-processed, such as removing spaces, punctuation symbols and making them

[8] https://developer.mozilla.org/en-US/docs/Web/API/Document$_{O}bject_{M}odel/Introduction$

lowercase. Tokenization is applied to pre-processed data. Each feature tokenization constants are shown in Table 4.1.

Feature	Max Number of Words	Max Sequence Length
Text	250	100
Form	500	20
DOM	30	450

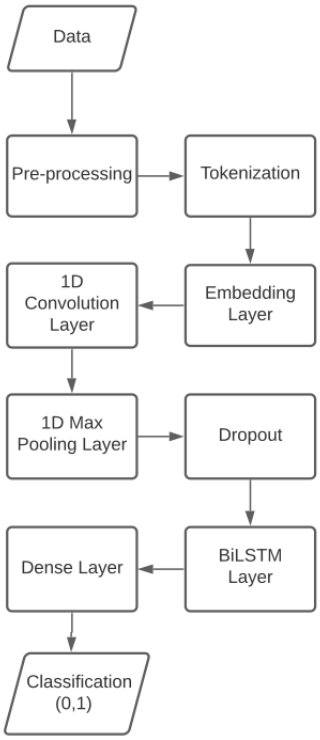
Table 4.1 Tokenization Constant Values

The embedding layer is applied after tokenization, which first assigns random weight to each unique token, then learns the embedding to embed all of the words. Then, a 1D convolution layer is applied to extract features from the text. After that, a max-pooling layer is applied for 1D temporal data to reduce convolutional feature size and retain the information. Max-pooling layers are applied for only form and DOM features. Afterward, a bidirectional LSTM layer is applied. BiLSTM receives the generated features from previous layers and generates features to the next hidden layer. Lastly, a dense output layer with a single neuron is used to make predictions for legitimate or phishing. Figure 4.1 shows the methodology of the hybrid CNN+BiLSTM model, and Table 4.2 shows the hyperparameters of CNN+BiLSTM.

Network	Embedding Size	Dropout	Activation	Optimizer	Batch Size	Max Epochs
Text Network	100	0.2	sigmoid	Adam	64	5
Form Network	100	0.2	sigmoid	Adam	64	5
DOM Network	100	0.4	sigmoid	Adam	64	5

Table 4.2 Hyperparameters for the CNN+BiLSTM Networks

Figure 4.1 Methodology of Hybrid CNN-BiLSTM Model



4.1.3.0.2 Random Forest

Random Forest algorithm is applied to the output of CNN+BiLSTM models. CNN+BiLSTM models result in a binary value; 0 for legitimate and 1 for phishing. Random forest classifies a website as legitimate or phishing, using the results obtained from the three content-based feature models. We used 100 trees in the forest, and the square root of the total number of features is used when looking for the best split. Table 4.3 shows the hyperparameters of the Random Forest classifier.

Number of Trees	Max Features
100	sqrt

Table 4.3 Parameters for the Random Forest Classifier

4.1.4 Data Collection for Real-time Detection

To make a remediation experiment, our system needs a feed of unblocked and accessible websites. Those websites should be classified in real-time since our experiment

needs phishing websites that are not blocked by USOM and not taken down by hosting providers. As a result, daily registered domains are used, which are newly opened from hosting providers, and our phishing detection system can detect phishing attacks before it spreads further.

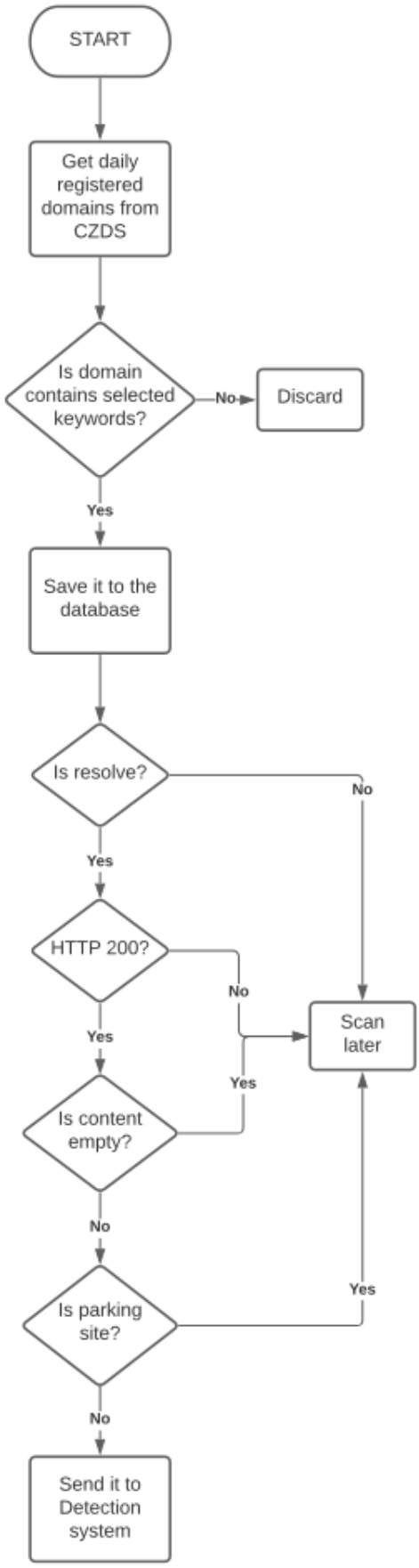
ICANN has a service called CZDS that gives access to zone files. These zone files contain the mapping between domain names, IP addresses, and their nameservers and are updated every 24 hours. After receiving zone files provided by the CZDS service, newly registered domains can be identified using a script. On the other hand, CZDS shares previous days' zone files, in which we can get domains one day after they are registered. Since there are no public services to provide today's daily registered domains, CZDS is the best option to use.

Approximately more than 200,000 domains are registered daily. Since our approach is to identify Turkish phishing websites, 16 keywords from the USOMs blacklist, which are potential keywords contained in Turkish phishing URLs, are extracted to speed up our detection time. Also, 136 parking pages, error and default titles are extracted from daily registered domains before the experiment. Parking pages are displayed to users when an owner of the domain registers a domain name from a hosting provider, but the website is not ready yet[9]. These constraints save tremendous time since parking sites, error and default pages do not contain any illegal activity.

Data collection processes can be summarized as follows: Every 30 minutes, daily registered domains that contain selected 16 keywords were checked from shared zone files. Then, in every hour, requests were made for daily registered domains and if the response is 200, content is not empty, and the title does not match with extracted titles, it is sent to our detection system. A devised model of data collection and preprocessing of newly registered domains is illustrated in Figure 4.2.

[9] https://www.namecheap.com/support/knowledgebase/article.aspx/459/46/what-is-a-domain-parking-page/

Figure 4.2 Flow Chart of Data Collection and Preprocessing of Newly Registered Domains



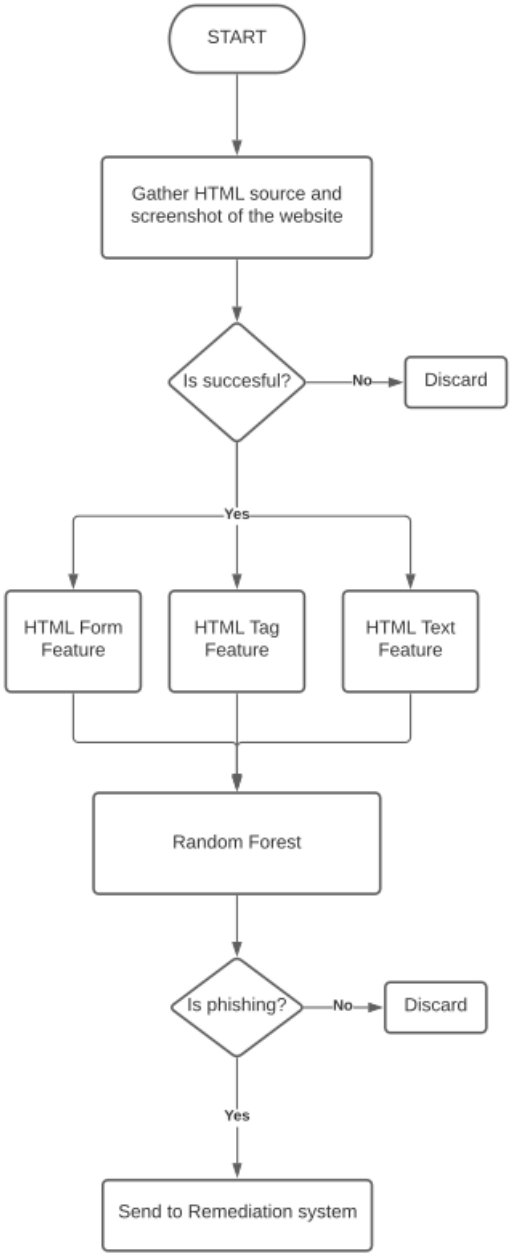
4.1.5 System Architecture

The proposed phishing detection system receives domains from the daily registered domain system and identifies the given domain as phishing or legitimate. In our system, we made requests to given domains using Playwright, a testing and automation library. Playwright automates Chromium browser with a Single API and allows the user to take screenshots and save the source code of the web pages. Since it opens a virtual display screen and real browser, it is slower than console-based HTTP requests. However, phishing websites may be rendered when only JavaScript is enabled. That is why it is needed to render web pages on a real browser.

After taking the screenshot and source of the web page, data was pre-processed considering our features, which are used for the detection model. HTML form action and input names, HTML text, and HTML tags are extracted from the source code using BeautifulSoup. After that, CNN+BiLSTM classifiers are run, and each CNN+BiLSTM classifier produces a binary result. These binary results are given to the Random Forest classifier to produce a single binary result that indicates if the given website is phishing or legitimate.

Websites that are classified as phishing are sent to our remediation system, which we explain in the next chapter. The proposed system architecture for the detection system is shown in Figure 4.3.

Figure 4.3 Flow Chart of Proposed Detection System Architecture



4.2 Remediation

Identifying phishing websites as soon as possible is necessary to minimize the potential loss. However, identifying phishing websites from legitimate ones is not sufficient to stop the attacker from deceiving users. The significant point of preventing this

kind of threat is to remediate those websites urgently, thereby internet users cannot reach phishing sites and as a result, users cannot be lured by attackers.

Remediation is completed when a phishing domain is taken down. Takedown events can be done by hosting providers since phishing websites are hosted by them. Hosting providers share abuse contact information to receive notifications from reporters about illegal activities running on their service. After inspection by an analyst at the hosting service, if hosting provider analysts agree that the website contains phishing content the website is taken down.

Also, USOM is the Turkish CERT and is responsible for taking action against cyberthreats in Turkey. As a result, blocking can be facilitated by USOM using a blacklist approach in Turkey. Some Internet Service Providers in Turkey periodically check that blacklist and as a result, they block access to malicious addresses in USOM's blacklist. When an ISP user tries to visit a blacklisted address, the ISP redirects traffic to "88.255.216.16".

In this paper, the impact of reporting to USOM in comparison to reporting to targeted hosting providers directly is measured. Additionally, how fast Turkish ISPs are reacting to phishing threats displayed in the USOM blacklist is also measured. For that, the three largest ISPs in Turkey are selected, and we measured their blacklisting and censoring time. However, not every ISP in Turkey blocks the malicious addresses listed in USOMs blacklist. Therefore, those ISPs are not selected as they do not offer any valuable information to conduct phishing remediation experiments in Turkey.

Once a phishing website is identified, the detection system sends the website to our remediation system. We built a remediation system that has two components. First, we randomly choose where to send an abuse notification e-mail: the hosting provider and USOM. When the notification is sent to the chosen side, websites that are subject to the notifications are tracked periodically whether they are blacklisted by USOM and censored by ISPs or taken down by the hosting provider after our notification.

To find significant differences between notifying hosting providers and USOM, we estimated the probabilities using the Kaplan-Meier method (Kleinbaum & Klein, 2012). Survival functions measure the fraction of phishing websites for a certain amount of time after abuse notification. Also, a log-rank test was used to evaluate differences between treatment groups. In the Log-rank test, p values less than 0.05 were considered significant. These measurements were done for each ISPs to find a significant difference between blocking time as well. Also, for each group, we

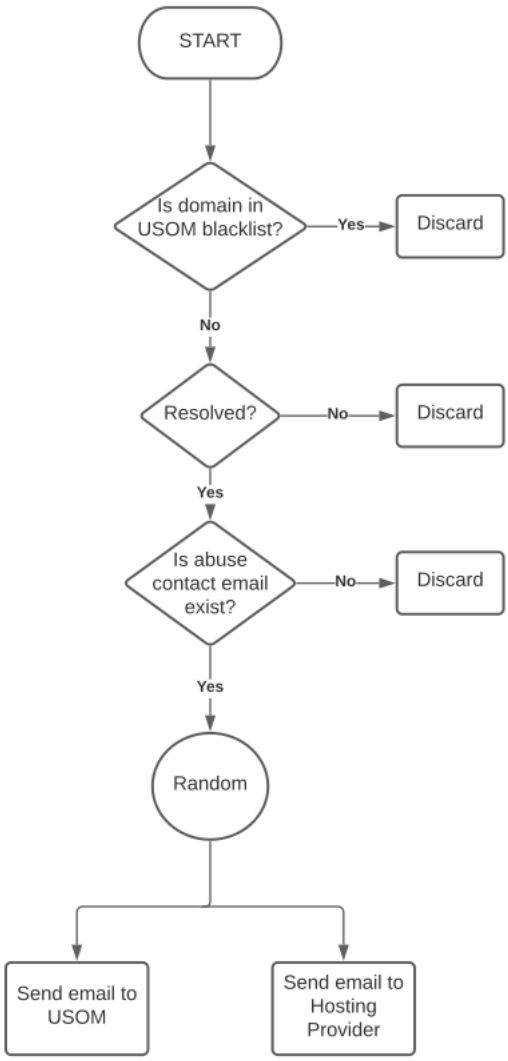
calculated the mean, median, maximum, minimum, and ratio of remediation and blacklisting times in hours. For ISPs, this is done for blocking time after the phishing website is blacklisted.

4.2.1 Identifying the Abuse Recipient

A randomized controlled experiment to identify effective ways to remediate Turkish phishing websites was conducted. For the first group of the control group, we sent an abuse notification e-mail to the hosting provider of the website. For the second group, we sent the same abuse notification e-mail to the USOM. USOM publishes a blacklist that contains malicious IP addresses, phishing and command & control domains, and IP addresses. In this way, Turkish ISPs obtain that blacklist and prevent their users from visiting those malicious addresses.

A devised model as to how to conduct a randomized control experiment to remediate Turkish phishing websites is illustrated in Figure 4.4.

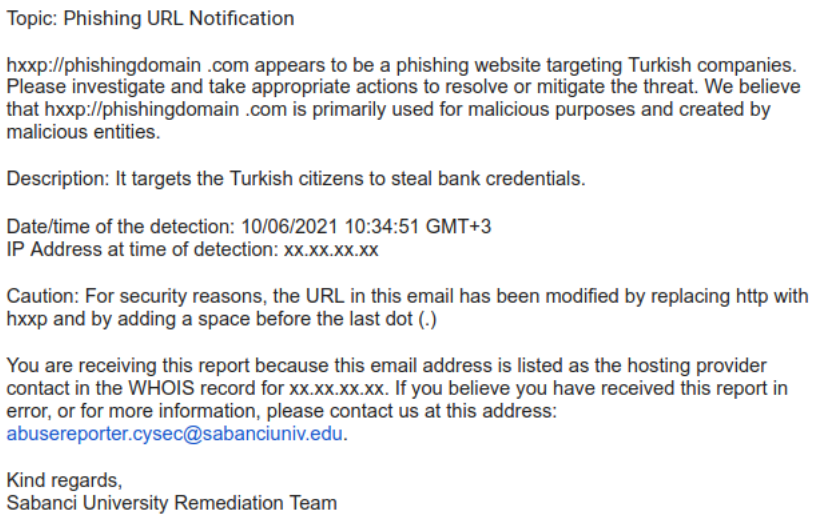
Figure 4.4 Flow Chart of Remediation Experiment



When a phishing website comes to our remediation system, we check whether the domain is on the USOM blacklist or not. If it is in that list, we discard that domain, since it is already blocked in Turkey. If it is not, we resolve its IP address and find the hosting provider's abuse e-mail contact of that domain using the "querycontacts"[10] service. After receiving an abuse contact e-mail, we randomly select the abuse notification receiver and send an e-mail to the chosen one by using the *abusereporter.cysec@sabanciuniv.edu* e-mail address which is under the Sabancı University domain. Abuse e-mail content is shown in Figure 4.5.

[10]https://github.com/abusix/querycontacts

Figure 4.5 Abuse Email Notification Content Example



Topic: Phishing URL Notification

hxxp://phishingdomain .com appears to be a phishing website targeting Turkish companies. Please investigate and take appropriate actions to resolve or mitigate the threat. We believe that hxxp://phishingdomain .com is primarily used for malicious purposes and created by malicious entities.

Description: It targets the Turkish citizens to steal bank credentials.

Date/time of the detection: 10/06/2021 10:34:51 GMT+3
IP Address at time of detection: xx.xx.xx.xx

Caution: For security reasons, the URL in this email has been modified by replacing http with hxxp and by adding a space before the last dot (.)

You are receiving this report because this email address is listed as the hosting provider contact in the WHOIS record for xx.xx.xx.xx. If you believe you have received this report in error, or for more information, please contact us at this address:
abusereporter.cysec@sabanciuniv.edu.

Kind regards,
Sabanci University Remediation Team

4.2.2 Tracking Reported Domains

The experiment requires a set of methodological approaches to track the notified domains acted upon in our abuse report. After sending an abuse notification e-mail, domains were started to track whether they are listed in USOMs blacklist or taken down by a hosting provider. Some Turkish ISPs are taking actions against domains in USOM's blacklist to block access to the pertaining websites. In our paper, we measured how fast Turkish service providers are reacting to the phishing threat displayed in the USOMs list and picked the thee largest ISPs that *do* take action about malicious threats from the USOMs blacklist.

In the following sections, we describe the tracking methodology both in Turkey and globally.

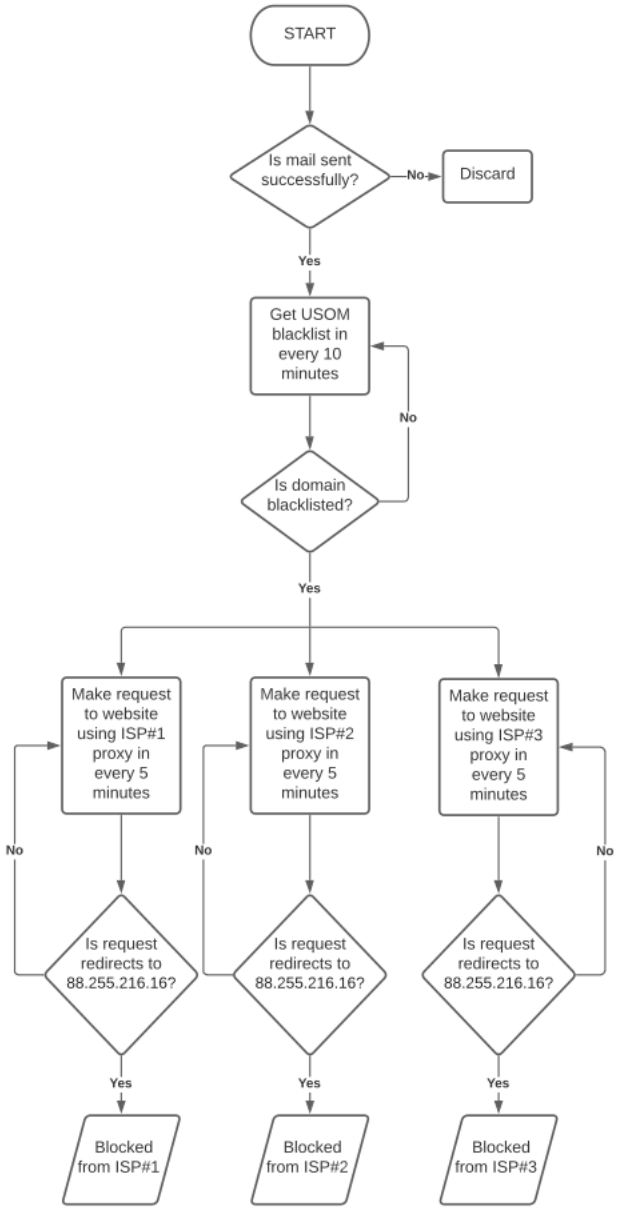
4.2.2.1 Turkey

In Turkey, local remediation can be done by USOM. To check the USOM's blacklist, an RSS feed[11] shared by USOM was used every 10 minutes. RSS feed contains the malicious domains and their detection time. If a website in our remediation system

[11] https://www.usom.gov.tr/rss/zararli-baglanti.rss

is included in that list, requests are made using our three different proxies every 5 minutes. If any response to a blacklisted domain is redirected to “88.255.216.16”, this means that the ISP blocked the traffic to the requested website. As a result, blacklisting time for USOM and 3 ISPs reacting time to block websites listed in USOMs blacklist is found. The tracking process for USOM and three ISPs are shown in Figure 4.6.

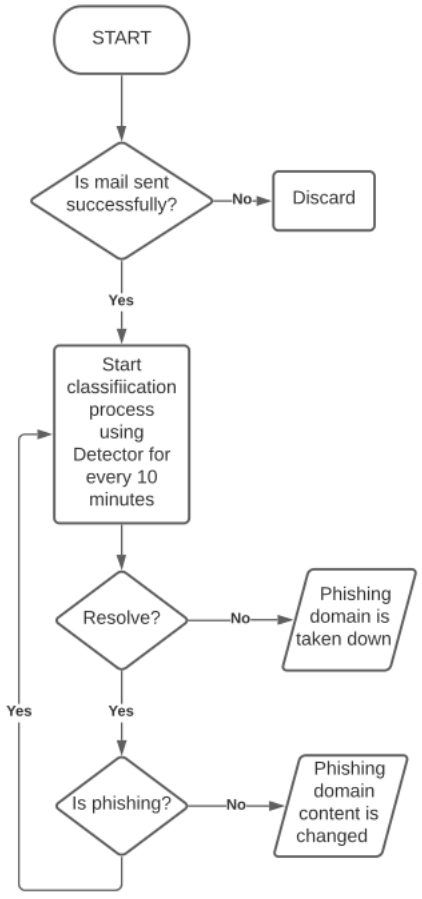
Figure 4.6 Flow Chart of Tracking Process in Turkey



4.2.2.2 Global

Our detection system was used periodically to track phishing web page content globally. If a given domain is not resolved, the phishing website is considered as taken down. On the other hand, phishing websites may be resolved, yet content is changed. Content change in phishing websites can be the result of hosting providers' actions or attackers' actions to hide malicious content. In either case, a web page, which was detected as phishing in the previous findings, is not considered phishing. Yet, the globally tracking process continues if phishing websites still resolve because attackers may put malicious content again. The tracking process in global is shown in Figure 4.7.

Figure 4.7 Flow Chart of Tracking Process in Global



5. RESULTS

5.0.1 Experiment Setup

Five different VPS (Virtual Private Server) are used in our experiments. Each system works separately on a single machine and runs on Ubuntu 20.04.2 LTS. The detection system has 64 GB RAM along with 10 CPUs. Also, our phishing detection models are kept in memory to decrease the detection time using joblib[1]. The tracker and daily registered domain collector systems have 4 GB RAM and 2 CPUs, and each proxy server using different ISPs has 1 GB RAM and 1 CPU.

5.0.2 Detection

For Turkish phishing website detection, four classifiers were trained and tested. To train and test, the phishing data set, which is provided from PRODAFT, is divided into two parts (%75 and %25, respectively). For each proposed model in Section 4.1 precision, recall, F1 score, AUC score, and accuracy are shown in Table 5.1.

A detection system is used to detect real-time daily registered Turkish phishing domains. From 25 May 2021 until 17 September 2021, 28734 domains were registered that contained the selected keywords mentioned in Section 4.1.4. Among 28734 registered domains, 8204 domains were scanned by the detection model. For the remaining domains, either the domain was not resolved, or the content was empty, or parking site content was blocked by USOM. Since ICANN gives new domains one day after their registrations, Turkish phishing websites could also have been blocked by USOM before we scan them.

[1] https://joblib.readthedocs.io/en/latest/

Model	Precision	Recall	F1	AUC	Accuracy
CNN+BiLSTM (Text Feature)	0.972	0.948	0.960	0.990	0.961
CNN+BiLSTM (Form Feature)	0.985	0.950	0.967	0.986	0.969
CNN+BiLSTM (DOM Feature)	0.963	0.857	0.907	0.946	0.914
Random Forest	0.970	0.970	0.970	0.981	0.970

Table 5.1 Precision, Recall, F1 Score, AUC score and Accuracy of Proposed Detection Models, which are trained and tested using PRODAFT's Dataset

Our approach is to remediate Turkish phishing websites by sending abuse notifications on real-time data. In order not to send a false notification, we checked every true labeled result manually. Among 8204 domains, 210 Turkish phishing websites were classified correctly, and another 163 websites were incorrectly classified as phishing as well. However, the latter set of websites were legitimate websites. We were not able to detect true negative and false negative ratios for labeled domains, since there were too many websites and they needed manual check as well. Table 5.2 shows the registered domain count, scanned domain count, true positive domain count, and false-positive domain count. Finally, Figure 5.1 shows the flow chart of our real-time detection system used to obtain results in the time interval from 25 May 2021 until 17 September 2021.

Registered Domains	28734
Scanned Domains	8204
TP Domains	210
FP Domains	163

Table 5.2 Registered, Scanned, True Positive and False Positive Domain Counts From 25 May 2021 until 17 September 2021

Figure 5.1 Flow Chart of Real-time Detection System Results From 25 May 2021 until 17 September 2021

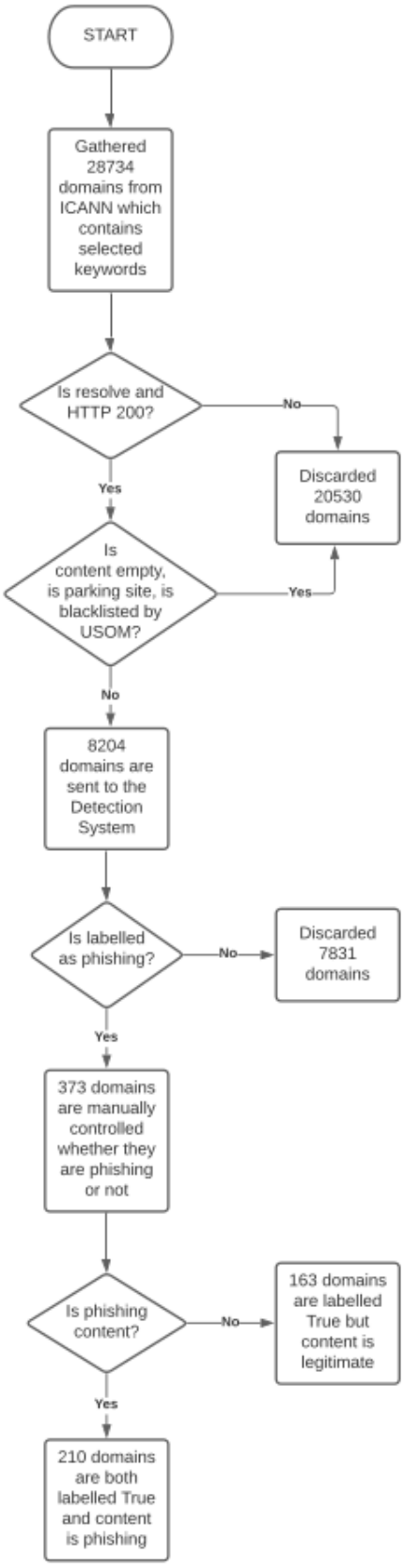


Table 5.3 shows the domain statistics in between 25 May 2021 and 17 September 2021. USOM blacklisted 9571 domains, and from these, around 30% of the domains (2938) contains the sensitive keywords that we were looking for while gathering daily-registered domains from ICANN. And, in 2938 domains, 5.28% of the domains (155) were both detected by our detection system and blacklisted by USOM. On the other hand, 2664 domains were both registered and blacklisted between 25 May 2021 and 17 September 2021. If we subtract the matched keyword domain count (2938) from that number, it shows that 274 domains were registered before 25 May 2021 and blacklisted between 25 May 2021 and 17 September 2021. Lastly, 1761 domains were blacklisted before ICANN shares them. More than 66% of the domains were both registered and blacklisted between 25 May 2021 and 17 September 2021.

Blacklisted Domains	9571
Contains Sensitive Keywords in Blacklisted Domains	2938
Blacklisted & Registered Domains	2664
Blacklisted Before Shared by ICANN	1761
Blacklisted & Detected by Our Detection System	155

Table 5.3 Statistics of Domains Between 25 May 2021 and 17 September 2021

5.0.3 Remediation

From 25 May 2021 until 17 September 2021, a total of 210 Turkish phishing websites were detected. From these, 106 of them were reported to the associated hosting providers and the rest (104) of them were reported to USOM. Most days, two phishing websites were detected. However, there were some days that there was no detection reported by our system. Table 5.4 shows the TLD (Top-Level Domain) counts for detected phishing websites. Among 210 Turkish phishing websites, ".com" was the most used TLD. Scammers used 17 other TLDs to register their phishing domains.

We extracted hosting provider and contact information for each phishing website by querying Abusix's Abuse Contact DB[2]. Table 5.5 provides some basic statistics of hosting providers in terms of the number of phishing websites detected and their phishing remediation times. One can observe sharp differences both in terms of the number of phishing websites detected and the median time to remediate a phishing website. For example, 'Namecheap' hosting provider remediated 80 phishing

[2]https://abusix.com/contact-db/

TLD	Count
com	103
xyz	70
Others (email, info, istanbul, website, space, store, live, tech, one)	9
org	6
online	6
club	4
digital	4
shop	3
net	3
fun	2
Total	**210**

Table 5.4 TLD Count for Detected Phishing Domains

websites 7 hours in median times, while for 8 phishing websites Microsoft's hosting platform took nearly 50 hours. Also, important to note that Cloudflare is not a hosting provider. However, Abusix's Abuse Contact DB and WHOIS databases only display Cloudflare contact details, not the real hosting provider that hosts the phishing website. To enable the remediation process, abuse notifications must be sent to Cloudflare's abuse department. To start the remediation process, Cloudflare forwards the abuse notification to the real hosting provider that hosts the websites. We followed the same approach for 21 phishing websites that were using Cloudflare's CDN service.

In the following, empirical estimation of the survival probabilities using the Kaplan–Meier method is performed. Then, a log-rank test is used to evaluate differences between the treatment groups.

One hundred six phishing websites were reported to the associated hosting providers, while the rest were reported to USOM. Table 5.6 shows the summary statistics on the time to remediate per treatment group. Only one phishing website, which was was assigned to the USOM treatment group, lived long enough to see another week. From Table 5.6, the mean remediation time for Turkish phishing websites was more than a day after an abuse notification was sent to the chosen recipient. The mean average remediation time for the hosting providers' treatment group was around 27 hours, while for the USOM treatment group it was almost 32 hours. On average, phishing websites assigned to the hosting provider treatment group remediated 5 hours faster than the ones assigned to the USOM treatment group. Within 5 hours, the attacker can still deceive people and steal their private information. On the other hand, the difference between the median and mean remediation times for both recipients was excessive. It shows that each remediation time data is right-

Hosting Provider	Count	Median remediation time (hours)
Namecheap	80	7.23
Google Cloud	47	7.74
GoDaddy	30	9.16
Cloudflare	21	7.59
Microsoft	8	49.96
Lacnic	4	22.95
Beanfield	3	13.41
Digital Ocean	2	19.82
Güzel	2	18.46
Hostinger	2	12.27
Vultr	2	5.50
Aerotek	1	4.82
Bluehost	1	0.36
MskHost	1	21.73
Cloudwm	1	20.95
Hostwinds	1	299.33
Leaseweb	1	2.18
JustHost	1	17.53
Timeweb	1	98.87
Veridyen	1	2.15

Table 5.5 Detected Phishing Website's Hosting Provider Count and Median Remediation Times

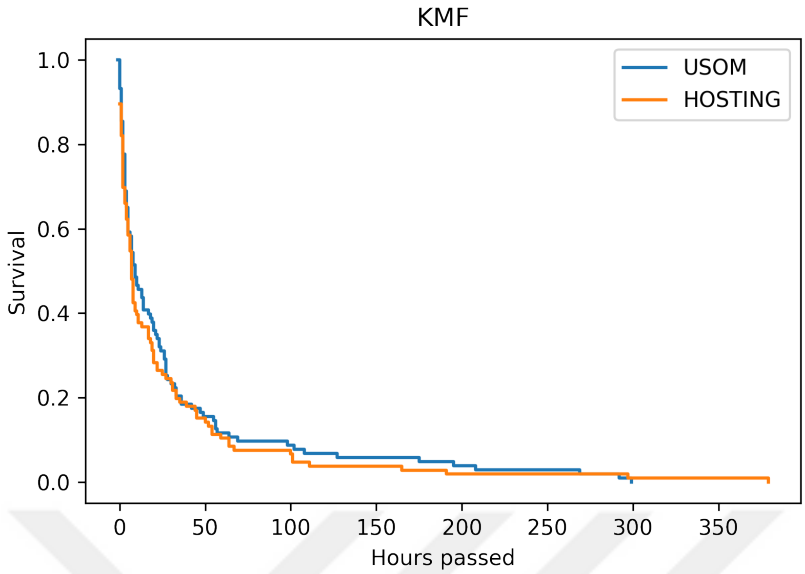
skewed. Also, the median average remediation time for hosting providers was less than that of USOM by nearly one and a half hours.

To further explore whether differences between sending abuse notifications to hosting providers and USOM are significant, we compute survival probabilities for each group. Figure 5.2 plots the Kaplan–Meier estimates. This figure shows that the similarity among the treatment groups was high. And, the log-rank test confirms that there was no statistically significant difference between the treatment groups ($\chi^2 = 0.54$, p=0.46).

Recipient	Count	Median remediation time (hours)	Mean remediation time (hours)	Max remediation time (hours)	Min remediation time (hours)	% of remediation
Hosting Provider	106	7.53	26.97	379.55	0.20	100.00%
USOM	104	9.25	32.00	299.33	0.36	96.15%

Table 5.6 Summary of Remediation Time According to Treatment Group

Figure 5.2 Survival Probabilities For Each Notification Recipient



USOM adds websites to their blacklist when these domains were either found by their operators or reported by external reporters. Table 5.7 provides a summary of blacklisting time according to the treatment group. The results show that the median blacklisting time was shorter when USOM was notified directly. The median blacklisting time was more than 3 hours for the hosting provider treatment group while, for the USOM treatment group, the median blacklisting time was less than an hour. Also, the average blacklisting time for USOM was nearly 3 hours where the mean blacklisting time for hosting providers was more than 8 hours. Surprisingly, five phishing that belonged to the USOM treatment group were not blacklisted. Moreover, 50 phishing websites that were assigned to the hosting provider treatment group were not blacklisted. The results demonstrate that those 50 phishing websites, which USOM did not blacklist, could be found by neither USOM nor external reporters; or they were taken down faster than the initial CERT phishing investigation.

Recipient	**Count**	**Median USOM blacklisting time (hours)**	**Mean USOM blacklisting time (hours)**	**Max blacklisting time (hours)**	**Min blacklisting time (hours)**	**% USOM blacklisting**
Hosting Provider	106	3.44	8.05	76.69	0.09	52.83%
USOM	104	0.82	2.97	145.44	0.06	95.19%

Table 5.7 Summary of Blacklisting Time According to Treatment Group

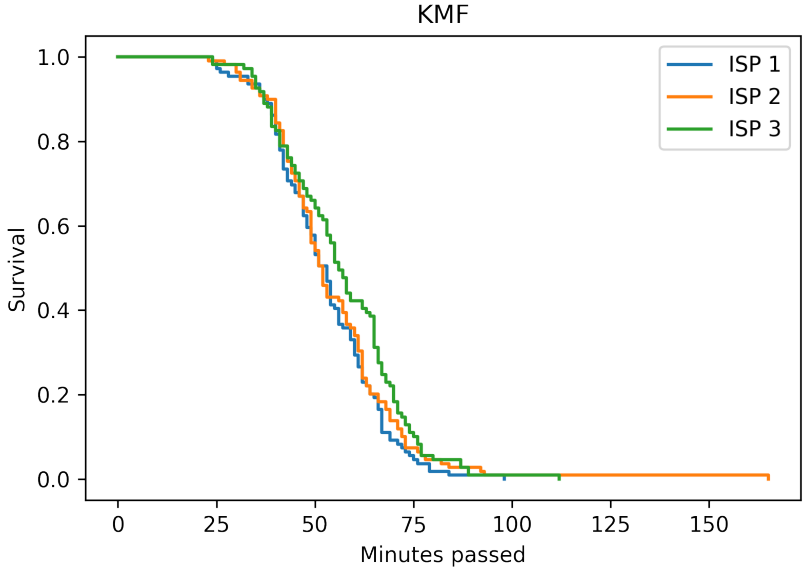
Some ISPs in Turkey periodically check USOM's blacklist and block malicious con-

tent for their users. Table 5.8 shows the summary of blocking times for three chosen Turkish ISPs. Since blocking methodology is different and can be automated, their blocking times are reported in minutes rather than hours. However, as seen in Table 5.8, median (and mean) blocking times are almost an hour for all three Turkish ISPs. However, minimum blocking times were under 30 minutes for all three ISPs. Maximum blocking times, on the other hand, were between more than one and half hours and nearly 3 hours for the same ISPs. To further explore whether differences in blocking times of the three ISPs are significant, we compute survival probabilities for each group. Figure 5.2 plots the Kaplan–Meier estimates. This figure shows that the similarity between ISP 1 and ISP 2 was high while the difference between ISP 1 and ISP 3 was high. The log-rank test confirms that there was a statistically significant difference between ISP 1 and ISP 3 ($\chi^2 = 5.39$, p=0.02), while there was no statistically significant difference between ISP 1 and ISP 2 ($\chi^2 = 0.61$, p=0.44), and ISP 2 and ISP 3 ($\chi^2 = 1.65$, p=0.2). Note that all three ISPs blocked 81.95% of the websites after they were added to USOM's blacklist. However, we found that the remaining websites (18.05%) were remediated by hosting providers before ISPs block them.

ISP No.	**Median blocking time (minutes)**	**Mean blocking time (minutes)**	**Max blocking time (minutes)**	**Min blocking time (minutes)**	**% blocked after seen in USOM blacklist**
ISP 1	53	52.77	98	24	81.95%
ISP 2	52	54.68	165	23	81.95%
ISP 3	56	56.8	112	24	81.95%

Table 5.8 Summary of Blocking Time According to the ISPs

Figure 5.3 Survival Probabilities For ISPs



After the remediation experiment, we sent abuse notifications of detected 24 Turkish phishing websites between 18 September 2021 and 3 October 2021 to only each ISP's abuse email addresses. Neither of the three ISPs' take any action to block these websites after our abuse notifications. Those websites were remediated by hosting providers, or they were blacklisted from USOM first, then blocked by each ISP.

6. DISCUSSION

In this section, first, we will discuss the results and issues of the proposed real-time detection system, then discuss the findings and issues of the remediation study.

6.0.1 Detection

The real-time phishing detection model labeled 373 domains as phishing out of 8204 scanned domains. From 373 domains, 210 of them were correctly labeled, while 163 of them were falsely labeled as phishing. The false-positive rate of the detection system is high because of the insufficient number of features used for detection and the small labeling sample size for training the models. It would result in fewer false-positive rates if more distinguishing content-based features were used and models trained with more both phishing and legitimate data. Yet, the detection system is successfully detected 210 Turkish phishing websites. The true positive rate would be higher if ICANN shares real-time registered domain names. Since ICANN share daily-registered domains the day after, most of the Turkish phishing websites were already blacklisted by USOM. On the other hand, if the attacker uses a path or sub-domain to perform a phishing attack, our system cannot detect it, because ICANN shares registered domains without a path. Lastly, 16 sensitive keywords are used to get daily-registered domains. Results show that only around 30% of the blacklisted domains contain selected keywords. If the detection system runs without hardware limitations, it is possible to scan every domain that ICANN shared. This would increase the true positive rate of our system.

6.0.2 Remediation

6.0.2.1 Efficacy of the USOM's blacklist in terms of phishing website remediation

In this study, we investigated the efficacy of USOM's public malicious resource blacklist compared to another standard approach where hosting providers were contacted directly about malicious resources in their network. Our results showed that there is no statistically significant difference between sending abuse notifications to USOM and sending abuse notifications to hosting providers. USOM is applying a blacklisting approach to mitigate malicious threats that target Turkish citizens. On the other hand, hosting providers take down malicious threats on their networks when notified. The results show that USOM's reaction time for abuse notifications is faster than hosting providers. Both treatment groups use operators to check the authenticity of the received abuse notifications. Since hosting providers may receive more abuse notifications than USOM, a faster action rate from USOM was expected because more abuse notifications take much more time to process. To increase the effectiveness of phishing website remediation, USOM and other CERTs might consider moving towards automated feeds and API-based abuse data sharing solutions. By using an API, USOM can send blacklisted IP addresses and URLs to associated hosting providers to achieve faster and automated malicious resource remediation. The problem with this approach is to promote API-based solutions to hosting providers. Not all hosting providers might be willing to use these API-based solutions. That is why it is important to provide incentives for early adoption.

6.0.2.2 Hosting providers' abuse remediation policies

The results show that notified hosting providers have varying phishing remediation times. Differences between remediation times are expected since each hosting provider's abuse remediation policies are different. As a result of this, some hosting providers can remediate phishing websites in less than an hour, and others remediate the phishing website after a week. To overcome this problem, hosting providers who take action for a very long time should be warned by authorities. Also, a study needs to be done to make the abuse remediation process occur in a more automated fashion. Hosting providers receive a lot of abuse notifications via e-mail. If the e-mail-based abuse notification content becomes a standard, it will be easier to parse the e-mail content and the remediation process will be more automated. However, abuse remediation processes will continue to differ for each hosting provider, unless

there is an authority that oversees all hosting providers' abuse remediation processes.

Additionally, the security community can also work towards creating more automated remediation and abuse notification parsing tools so that hosting providers can easily automate their remediation process.

6.0.2.3 ISPs' abuse blocking policies

Our results indicated that ISP blocking time can be much faster. Also, we noticed that chosen three ISPs do not take further actions towards mitigating phishing sites when they were notified directly by e-mail based abuse notifications. Since USOM operators already manually check blacklisted malicious threats, ISPs' can fetch the USOM's blacklist and block malicious threats immediately without validation. To make the blocking process faster, ISPs can request real-time blacklist feeds from USOM. Moreover, ISPs can invest more resources into validating and processing notifications that were sent by reporters. This could help ISPs mitigate more malicious threats that target their clients.

7. CONCLUSION AND FUTURE WORK

Phishing is one of the most serious cyber threats for Internet users. Machine learning approaches have been increasingly used to detect phishing websites in recent years. Following the machine learning approach, in this study, a hybrid CNN+BiLSTM and Random Forest classifier is used to detect Turkish phishing websites. The existing solutions do not use Turkish phishing data set, which makes our study unique in the literature. Also, the proposed approach uses a different approach in the detection of phishing websites. Our system can result in higher than 97% accuracy on test data; however, on real-time detection, the false positive rate turns to be high while the true positive rate is low partially due to the limitations discussed in the following.

Remediation of phishing websites is a significant step to stop the phishing threat and minimize the loss. Remediation can be done by hosting providers by taking down phishing websites. Also, the local CERT, USOM in the case of Turkey, publishes a blacklist of phishing websites that target Turkish citizens to protect them against malicious cyber activities. This study measured the impact of reporting Turkish phishing websites to USOM as opposed to reporting them to hosting providers. Our results show that there is no statistically significant difference between reporting to USOM as opposed to reporting to hosting providers.

In addition to that, we experimented with the blocking times of phishing websites of three Turkish ISPs. We found that one ISP's blocking time is better than another ISP's blocking time. Each ISP's median blocking time is nearly 1 hour, which is sufficiently long for attackers to achieve their malicious goals. Therefore, we believe ISPs can automate their processes by checking USOM's blacklist more frequently for faster blocking of malicious websites to protect their users.

In the future, more distinguishing content-based features can be added to our classifier to achieve lower false-positive rates in the detection of phishing websites. Also, the dataset for both legitimate and phishing can be increased. Since the hardware used in our experiments has limited computation and communication resources, we scanned daily-registered domains which contain only a small set of selected key-

words, which contains only 16 keywords. With more hardware resources we can scan more daily-registered domains and potentially find more phishing websites. This will definitely increase our true positive rate, which would give more insights about the differences in our treatment groups. The proposed methodology used in this study can be profitably utilized in future works to this end.

The proposed system only scans registered domains shared by ICANN. And, if the attacker puts the malicious content in a different location than the root directory or root sub-domain, our system cannot detect it. Lastly, ICANN shares daily-registered domains the day after, and if USOM blacklisted a Turkish phishing website on the day before ICANN shares, we are not able to detect it and use it in our remediation study.

BIBLIOGRAPHY

Anti-Phishing Working Group (2020). Phishing Activity Trends Report. (June), 1–11.

Biau, G. & Scornet, E. (2016). A random forest guided tour. *Test*, *25*(2), 197–227.

Canali, D., Balzarotti, D., & Francillon, A. (2013). The role of web hosting providers in detecting compromised websites. In *Proceedings of the 22nd international conference on World Wide Web*, (pp. 177–188).

Çetin, O., Ganán, C., Altena, L., Kasama, T., Inoue, D., Tamiya, K., Tie, Y., Yoshioka, K., & van Eeten, M. (2019). Cleaning up the internet of evil things: Real-world evidence on isp and consumer efforts to remove mirai. In *NDSS*.

Cetin, O., Ganán, C., Altena, L., Tajalizadehkhoob, S., & van Eeten, M. (2018). Let me out! evaluating the effectiveness of quarantining compromised users in walled gardens. In *Fourteenth Symposium on Usable Privacy and Security ({SOUPS} 2018)*, (pp. 251–263).

Cetin, O., Ganán, C., Altena, L., Tajalizadehkhoob, S., & van Eeten, M. (2019). Tell me you fixed it: Evaluating vulnerability notifications via quarantine networks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, (pp. 326–339). IEEE.

Cetin, O., Ganan, C., Korczynski, M., & van Eeten, M. (2017). Make notifications great again: learning how to notify in the age of large-scale vulnerability scanning. In *Workshop on the Economics of Information Security (WEIS)*.

Cetin, O., Hanif Jhaveri, M., Gañán, C., van Eeten, M., & Moore, T. (2016). Understanding the role of sender reputation in abuse reporting and cleanup. *Journal of Cybersecurity*, *2*(1), 83–98.

Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, *484*, 153–166.

Chiew, K. L., Yong, K. S. C., & Tan, C. L. (2018). A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, *106*, 1–20.

Curtis, S. R., Rajivan, P., Jones, D. N., & Gonzalez, C. (2018). Phishing attempts among the dark triad: Patterns of attack and vulnerability. *Computers in Human Behavior*, *87*, 174–182.

El-Alfy, E.-S. M. (2017). Detection of phishing websites based on probabilistic neural networks and k-medoids clustering. *The Computer Journal*, *60*(12), 1745–1759.

Feng, F., Zhou, Q., Shen, Z., Yang, X., Han, L., & Wang, J. (2018). The application of a novel neural network in the detection of phishing websites. *Journal of Ambient Intelligence and Humanized Computing*, 1–15.

Greene, K., Steves, M., & Theofanos, M. (2018). No phishing beyond this point. *Computer*, *51*(6), 86–89.

Gupta, S., Singhal, A., & Kapoor, A. (2016). A literature survey on social engineering attacks: Phishing attack. In *2016 international conference on computing, communication and automation (ICCCA)*, (pp. 537–540). IEEE.

Hutchings, A., Clayton, R., & Anderson, R. (2016). Taking down websites to prevent

crime. In *2016 APWG symposium on electronic crime research (eCrime)*, (pp. 1–10). IEEE.

Internet Crime Complaint Center (2020). 2020 Internet Crime Report, 1–28.

Jain, A. K. & Gupta, B. (2018a). Phish-safe: Url features-based phishing detection system using machine learning. In *Cyber Security* (pp. 467–474). Springer.

Jain, A. K. & Gupta, B. B. (2018b). Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems*, *68*(4), 687–700.

Jain, A. K. & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing*, *10*(5), 2015–2028.

Jhaveri, M. H., Cetin, O., Gañán, C., Moore, T., & Eeten, M. V. (2017). Abuse reporting and the fight against cybercrime. *ACM Computing Surveys (CSUR)*, *49*(4), 1–27.

Kleinbaum, D. G. & Klein, M. (2012). Kaplan-meier survival curves and the log-rank test. In *Survival analysis* (pp. 55–96). Springer.

Kührer, M., Hupperich, T., Rossow, C., & Holz, T. (2014). Exit from hell? reducing the impact of amplification ddos attacks. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, (pp. 111–125).

Lance James (2006). Chapter 1 - Banking on phishing. In Lance James (Ed.), *Phishing Exposed* (pp. 1–35). Burlington: Syngress.

Li, F., Durumeric, Z., Czyz, J., Karami, M., Bailey, M., McCoy, D., Savage, S., & Paxson, V. (2016). You've got vulnerability: Exploring effective vulnerability notifications. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, (pp. 1033–1050).

Mitchell, T. (1997). Machine learning.

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, *25*(2), 443–458.

Moore, T. & Clayton, R. (2007). Examining the impact of website take-down on phishing. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, (pp. 1–13).

Nappa, A., Rafique, M. Z., & Caballero, J. (2013). Driving in the cloud: An analysis of drive-by download operations and abuse reporting. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, (pp. 1–20). Springer.

Pranggono, B. & Arabo, A. (2021). Covid-19 pandemic cybersecurity issues. *Internet Technology Letters*, *4*(2), e247.

Rader, M. & Rahman, S. (2015). Exploring historical and emerging phishing techniques and mitigating the associated security risks. *arXiv preprint arXiv:1512.00082*.

Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from urls. *Expert Systems with Applications*, *117*, 345–357.

Stock, B., Pellegrino, G., Li, F., Backes, M., & Rossow, C. (2018). Didn't you hear me?—towards more successful web vulnerability notifications.

Vasek, M., Weeden, M., & Moore, T. (2016). Measuring the impact of sharing abuse data with web hosting providers. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, (pp. 71–80).

Xiang, G., Hong, J., Rose, C. P., & Cranor, L. (2011). Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, *14*(2), 1–28.

Zhang, H., Liu, G., Chow, T. W., & Liu, W. (2011). Textual and visual content-based anti-phishing: a bayesian approach. *IEEE transactions on neural networks*, *22*(10), 1532–1546.

Zhang, W., Jiang, Q., Chen, L., & Li, C. (2017). Two-stage elm for phishing web pages detection using hybrid features. *World Wide Web*, *20*(4), 797–813.