Nereus: A Proposal for Implementing Anti-phishing Software Using Corporate Branding Color Matching

Benjamin Heald

ABSTRACT

As communication on the internet evolves, existing anti-phishing software is becoming less effective as more users migrate away from Email and into emerging technologies such as Slack, Zoom, and Microsoft Teams. Since effective anti-phishing filters must be created for each new communication platform, developers are fighting an uphill battle to keep users safe. An anti-phishing mechanism that instead positions itself directly between the user and the websites they visit is therefore proposed. This positioning allows the project to protect the user against phishing attacks no matter what communication medium the user received the attack through. This project utilizes a supervised machine learning algorithm with strong features such as corporate branding color matching and URL features. This project design theoretically allows the application be accurate, effective, and adaptable with little to no added overhead, overcoming many of the shortcomings in currently proposed solutions.

ACM Reference format:
Benjamin Heald. 2020. Nereus: A Proposal for Implementing Anti-phishing Software Using Corporate Branding Color Matching. In *Proceedings of NA, NA, 2020,* 4 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Phishing is defined as a criminal activity combining social engineering and technology to access private information without consent [3]. Phishing is one of the most common and easily performed cyber-attacks, costing the world more than $450 billion per year or nearly 90% of the total estimated cost of cyber-crime. [5][6]. Popular tools such as the "Social Engineering Toolkit" make it trivial for cyber-criminals to automate the process of cloning websites and creating custom URLs for their victims to visit [8]. These URLS are then sent either to a great number of users via traditional phishing attack or to a specific person within an organization under spear phishing. Some form of text will usually accompany the URL in order to persuade the victim into clicking on the link and inputting personal data on the following web page. Any data entered by the victim on the phishing site is then usually sent directly to the attacker. The overall flow of a generic phishing attack is given below.

The majority of anti-phishing systems position themselves between the user and their communication platform. From this position, the software can easily intercept and flag incoming messages that are deemed potential phishing attacks before the user ever sees them. Since email is the main online communication platform for both businesses and individuals, most implementations of this approach are created only for use by email service providers. This intercept-and-filter model has been proven to be extremely effective at preventing phishing emails from reaching users, and has been widely implemented in major email systems for many years. Major email service providers such as G-Mail have developed extremely sophisticated systems in which around 100 million phishing messages are intercepted and filtered each day [1]. Over time this approach has begun to be ineffective as the number of available communication platforms skyrockets.

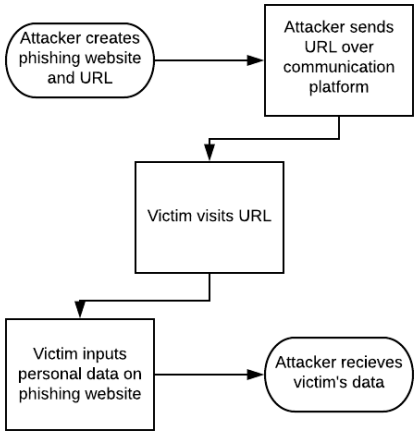

Figure 1: Generic Web Phishing Attack Flow

With the advent of applications such as WhatsApp, Zoom, Facebook Messenger, Slack, Microsoft Teams, and other instant messaging applications, more and more communication on the internet is conducted away from email. These new communication channels have existed for years, yet little work has been done to implement the same kind of intercept-and-filter mechanisms used by email on these platforms. With the fickle nature of consumers and the rapid adoption of new communication platforms, it is presumed a losing battle to try and implement effective anti-phishing filters in every new platform. Software that attempts to prevent phishing attacks once the user has visited a suspicious website is therefore theoretically more adaptable to this environment. These systems attempt to prevent the user from entering sensitive data after they have already been tricked into visiting a URL leading to a phishing site. This identify-and-warn model enables the protection of users against phishing attacks even if the intercept-and-filter protection of the communication platform fail. While this protection is placed

NA, NA

DOI: 10.1145/nnnnnnn.nnnnnnn

much closer to the attacker's end goal, it would be better suited to the rapidly changing landscape of the internet.

A piece of anti-phishing software that positions itself between the user and a phishing website needs adhere to the following design principles:

(1) Accurate. The application should be able to accurately and reliably identify if the website visited is a phishing attack.
(2) Adaptable. The application should automatically adapt to new phishing attacks and be effective without requiring major updates from the software maintainers.
(3) Effective. The application should be able to properly notify the user of it findings and ensure the user does not ignore the warning.
(4) Fast. The application must ensure that it's operation does not impact the speed of the user's internet browsing.

Achieving these goals can be extremely difficult however. The transiency of webpages, constantly improving quality of attacks, and speed at which it takes to make an accurate decision all factor into this difficulty. Generic warnings and false positives also threaten to desensitize users, making it unlikely that they will heed the warnings presented by the application. Though many systems have been proposed and are currently in use to prevent users from interacting with potentially dangerous websites, it is believed that none completely match the criteria listed above.

2 REVIEW OF CURRENT EFFORTS

There are a number of current in-browser software packages that attempt to provide in-browser protection against web phishing attacks. All of these packages work to identify if a given website represents a phishing attack in real-time as the user visits the website. Working in real-time gives the application an extremely small window to operate in, as a user of a website expects it to be available for interaction immediately following page load.

Of all the current efforts to implement a effective system in this domain, Google Chrome's "Real-time phishing protection" [4] currently presents the best solution. In this model, when the user visits a new site, the URL is checked against a Google-generated blacklist of known phishing URLS that is updated regularly. If the URL is not present in the blacklist, the application then performs a secondary check wherein the URL is sent to Google to determine if it has the qualities of a phishing URL. If either of these two checks detect a phishing attack, a full-page banner is presented warning the user. While this approach to ant-phishing software achieves both accuracy and speed, it is doubtful that it is either adaptable or effective at ensuring users heed its warnings. Since it depends heavily on the presence of the URL on a blacklist, it would not be useful against an attacker who can quickly generate new domains for their attacks. This blacklist approach would also not be effective at preventing spear phishing attacks, wherein the URL is only sent to one person, as their blacklist is populated using mainly crowd-sourced data. While their phishing detection using URL characteristics is likely accurate, it is unlikely to remain so as attackers constantly adapt and learn what such algorithms identify as phishing attacks. It's effectiveness at keeping users from interacting with the potential phishing site is also in doubt, as the warning it presents is extremely generic. Such warnings are often ignored by users if the perceived reward of visiting the site outweighs the described risks.

Other software packages such as the one created by Mohammad et al [7] propose a machine learning approach to task of identifying phishing attacks in real time. This approach entails learning a model that can accurately predict if a given URL represents a phishing site. This learning is done using certain qualities of the URL as features in a supervised learning algorithm. While this attempt is adaptable to new attacks, it's effectiveness at ensuring users do not interact with the site is in doubt as it uses very poor warnings. This application additionally does not achieve the level of accuracy desired, as it learns solely based upon qualities of the URL. As with Google's model, this will likely not remain accurate in the long run as attackers adapt and modify their URL structure to avoid detection.

From these two examples, it is clear that a more comprehensive software package is needed. An anti-phishing application that is adaptable to new techniques and attacks will need to implement a machine learning algorithm similar to the one proposed by Mohammad et al [7]. To overcome the limitations of that approach however, the speed and accuracy of Google's method [4] will also need to be implemented. A real-time detection method that does not depend solely on qualities of the URL or a blacklist is also needed to overcome the limitations of Google's approach. In addition to these requirements, neither of these current solutions were judged to be effective at ensuring that the user heeds the presented warnings. A solution which optimizes speed, accuracy, and adaptability, while at the same time able to present a detailed and highly engaging warning is therefore required.

3 HYPOTHESIS

It is clear from the review of current anti-phishing software packages that a new solution is needed. This project proposes a new anti-phishing mechanism that involves a supervised learning algorithm similar to the one proposed by Mohammad et al [7], with one major difference. Rather than learning solely from features of the URL, the fundamental hypothesis of this project revolves around using a web-page's branding color as the leading feature of a supervised learning algorithm.

Color is acknowledged as a critical element of the corporate identity, as can be observed in such cases as IBM (otherwise known as "The Big Blue"). It is highly influential given that it is perceived more quickly than symbols or text, and is memorable of the brand. Therefore it is reasonable to say that overhauls of a company's branding color is not common, especially those that are global, though they may often change their HTML, logo, or general design.[2] If a phishing website wishes to accurately impersonate a corporate brand, they will need to directly copy that brand's color scheme. If the branding color is not directly copied, the potential victim is much more likely to notice that the website is a phishing attempt.

The semi-permanent nature of branding colors provides a supervised learning algorithm with the opportunity to more accurately identify both the presence of a phishing website and the exact corporate identity the website is trying to mimic. Measuring how "close" a given website's color scheme is to a list of known corporate

branding colors will be an extremely powerful feature in a supervised learning algorithm. Working in tandem with URL features, this will produce a model that is able to identify a phishing website with high accuracy.

This project is additionally hypothesized to be effective at reducing the click-through rate of its users. It is theorized that the low user retention and high click-through rate of most anti-phishing tools is due to the presentation of generic warnings. These non-descriptive warnings likely do little to properly warn the user about the level of danger that they are in when a phishing attack has been detected. The ability of the model to identify the name of the corporate service that will be compromised by the phishing attack will likely drive up user retention and reduce the warning click-through rate.

By implementing corporate color matching as a feature in a supervised machine learning model, this project aims to achieve all four of the design principles described in section one.

(1) Accurate. The project will use color matching along with URL features in order to accurately identify phishing attacks.
(2) Adaptable. The project will be able to automatically adapt to new phishing attacks in the long term through constant re-learning of the model as well as depending on the semi-permanent nature of branding colors.
(3) Effective. The ability of the project to identify the name of the corporate entity being used in the phishing attack will likely drive up user retention.
(4) Fast. The project will take in an extremely small amount of data from the website on page load and transmit this to a server where the model will determine the phishing status. This distribution of machine duties will allow the project to operate with little noticeable overhead.

4 IMPLEMENTATION

Named for the ancient Greek god of fishermen, Nereus will be implemented in two separate pieces of technology. The first piece of this project will be a supervised learning model that learns on a multitude of features about a potential phishing site in order to produce a prediction. These features will include the qualities of the URL as well as branding color comparison. The model will take in a large vector of boolean features for the URL qualities, as well as float values ranging from 0 to 1 for the color matching. These values will represent the euclidean distance between the dominant RGB value of the given website and the RGB values of the top 50 most popular sites on the internet. A large curated data-set of phishing websites will be generated in order to both train this model and test the statistical relevance of the features. This training and feature evaluation will be done using the Python SciPy package. A study of known phishing sites and how often they use corporate branding will need to be done in order to validate the usefulness of the branding color comparison.

The secondary piece of this project will be a Google Chrome browser extension. This piece of software will act as the main GUI for the application. The user will install this extension in order to use the project. The extension will position itself between the user and every website they visit, collect the necessary data on the URL and color branding, and pass it along to the trained machine learning model. The color branding will be accomplished by taking a screenshot of the page, analyzing its dominant color, and then comparing that RGB value to a database of known brand RGB values. In order to improve running time, the trained machine learning model will live on a separate cloud interface contacted through an API by the Chrome extension. If the model determines the website is a phishing attack, the model will send this information back to the Chrome extension, along with the identity of the corporate entity that the website is attempting to mimic, if the identity was determined. The Chrome extension will then generate a warning regarding the attack, and the user may click through this warning. The proposed flow of this project is shown below.

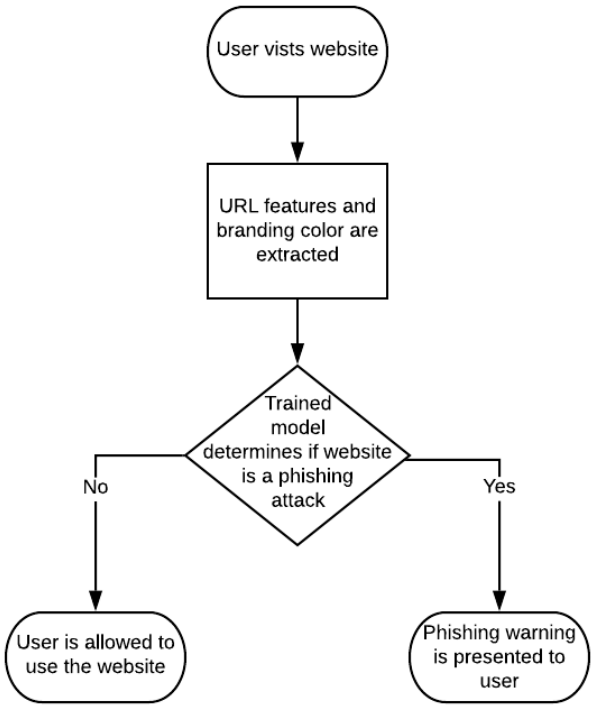

Figure 2: Proposed Project Design

In order to validate the hypothesis that corporate branding color matching will be a strong feature, the model will be trained both with and without this feature present. The hypothesis will be supported if the accuracy of the model increases significantly when the color matching feature is included. The adaptability of the project will be validated by testing the completed model on new phishing attack data over a period of time and seeing if the accuracy of the model. In order to assess the project's impact on website load speed, measurements of the website response time will be taken. If the average response time increase is limited to around 0.5 seconds, this added overhead will be considered acceptable. If time permits, a study on the effectiveness of generic versus specific warnings

to reduce click-through rates and user retention will also be performed. This type of study will serve to validate the secondary hypothesis that the project will be effective at persuading users to heed the warnings presented by the application.

5 PROJECT TIMELINE

A general project timeline is presented below in accordance with the milestones from the CS graduate handbook.

- Week 4: Complete construction and review of a curated data-set for the supervised learning model. Statistical analysis of the effectiveness of the URL and color branding features completed.
- Week 8: Initial model designed, trained, and documented, with the Google Chrome extension also completed.
- Week 12: Validation and improvements of the model finalized, with the project as whole in a working state. This includes the Google Chrome extension being able to reliably interact with the model through a secure API.
- Finals week: Final validation of project and hypothesis completed as described in the implementation section. Validation results included within both the project report and poster, which are also at this point completed and reviewed.

Deliverables for this project include the following, as required by the CS graduate handbook.

- Project Report: At the end of this project, a complete report will be created that includes all results of the project.
- Project Software: At the conclusion of the project, a complete software package will be implemented that includes all pieces described in the implementation section and described in figure two.
- Project Poster: A poster will be produced that summarizes all project results in an interesting and engaging format.

6 CONCLUSION

The shortcomings of current anti-phishing mechanisms necessitate the development of new systems. As communication on the internet evolves, intercept-and-filter software becomes increasingly hard to implement across all platforms. This exacerbates the need for a reliable and accurate anti-phishing screening system that exists between the user and the websites. This system will prevent users from inputting personal data on phishing sites that they have already visited. By utilizing the feature of corporate color matching in a supervised machine learning model, this project is theorized to be able to overcome all shortcoming of previous work conducted in this domain. High accuracy, strong user retention, and low overhead speed are all expected in this approach.

REFERENCES

[1] Elie Bursztein. 2019. Understanding why phishing attacks are so effective and how to mitigate them. (2019). https://security.googleblog.com/2019/08/understanding-why-phishing-attacks-are.html

[2] Jose Luis Caivano and Mabel Amanda Lopez. 2007. Chromatic identity in global and local markets: analysis of colours in branding. *Journal of the International Colour Association* 1, 3 (2007), 1–14.

[3] A. Carella, M. Kotsoev, and T. M. Truta. 2017. Impact of security awareness training on phishing click-through rates. In *2017 IEEE International Conference on Big Data (Big Data)*. 4458–4466.

[4] Google. Phishing Protection. (????). https://cloud.google.com/phishing-protection

[5] Christian Konradt, Andreas Schilling, and Brigitte Werners. 2016. Phishing: An economic analysis of cybercrime perpetrators. *Computers & Security* 58 (2016), 39–46.

[6] James Moar. 2015. The future of cybercrime & security: Financial and corporate threats & mitigation. *Juniper, Dec* (2015).

[7] Rami M. Mohammad, Fadi Thabtah, and Lee McCluskey. 2014. Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications* 25, 2 (01 Aug 2014), 443–458. DOI: http://dx.doi.org/10.1007/s00521-013-1490-z

[8] N. Pavković and L. Perkov. 2011. Social Engineering Toolkit — A systematic approach to social engineering. In *2011 Proceedings of the 34th International Convention MIPRO*. 1485–1489.