A Logo Identification and Brand Recognition Approach for Phishing Detection (LibraPhish)

Sammy Chien
sammychien@utexas.edu
UT Austin, USA

Abstract

Phishing attacks, like all forms of social engineering, have become increasingly sophisticated over time. Many users and large organizations only rely on browser warnings for phishing attack prevents. These browser warnings are usually powered by semi-frequently updated URL blacklist. However, these warnings are only a stopgap measure: completely new phishing attempts won't be caught by blacklists, and the expected lifespan of these attacks are usually less than a day.

This project will examine properties found in the current landscape of commonly circulating phishing attacks and present LibraPhish (Logo Identification and Brand Recognition Approach for Phishing Detection), a tool that uses image processing on business logos and brands and graph reachability analysis to determine if a website is part of a phishing campaign and what the corresponding legitimate website would be. Unlike many visual similarity based phishing detection approaches, LibraPhish is stateless and does not need to maintain a list of validated, legitimate websites with their metadata.

Depending on the threat model for phishing attacks. LibraPhish can sit on the ingress point and comb through all incoming website links tagged as "suspicious". For each of these suspicious websites, LibraPhish can generate a label and a guaranteed legitimate website correlated to the suspicious website. If the level of suspicion for the website under review is high, expert human review can be requested to make the final determination. An experienced administrator can easily compare the suspected website with a LibraPhish's legitimate website source and make the final determination (either dangerous or safe) for the suspected website.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Sammy Chien. 2022. A Logo Identification and Brand Recognition Approach for Phishing Detection (LibraPhish) . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Phishing is a form of social engineering in which an attacker tricks a victim into giving up their secure credentials, such as login information or credit card information. It is by far the most common form of attack utilized by cyber-criminals, and most tech users have encountered one form of it or another at some point [4].

A fairly common phishing scenario involves an attacker sending a fraudulent message like an email to a victim. The message entices the victim to visit a malicious website - on the surface, the website looks perfectly legitimate. The victim, believing the website is legitimate, may enter sensitive information onto the website, which the attacker will then be able to record and compromise.

The cost of these compromised credentials is extremely high. An attacker can purport to be the identity of the victim to bypass authentication security features. After entering into the secure network, an attacker has many more options for malicious actions: this can range from data exfiltration to completely shutting down a network [3].

It's important to note that phishing attacks usually target more tech-illiterate users. These users often do not follow modern internet security practices, such as avoiding login and password reuse. Additionally, many small-scale business do not have strong IT security capabilities - if many employees are also less well-versed in detecting suspicious websites, then they are even more susceptible to these scams. Once an attacker steals the credentials from a victim for one particular service, they can attempt to use the exact same credentials on another service, and bypass even more authentication checks. Although many systems now have multi-factor authentication, which reduces the negative effects of losing one set of credentials, a sophisticated attacker could target the same victim again to obtain a one-time passcode that bypasses the multi-factor authentication. After all, if a victim were not savvy enough to recognize the first phishing attack, it's unlikely they'll recognize the second as well [6].

This paper explores some current and commonly used techniques that cyber-criminals employ when deploying a

phishing campaign and presents LibraPhish, a system that can detect these attempts and determine whether a site is a phishing attempt or legitimate.

LibraPhish works on the principle that malicious websites attempting to mimic legitimate websites must use images and logos to persuade a victim to believe in their authenticity. Victims who encounter an unfamiliar webpage with an unknown logo are unlikely to trust the website and enter confidential informaiton. Since these logo images must be shown to the user, LibraPhish can take those images and perform image analysis to uncover the actual brand that the logo corresponds to along with a legitimate website for that brand. Then, LibraPhish performs link analysis on the suspected and legitimate websites and uses a set of rules to determine the likelihood of the suspected website as a phishing attack.

2 Motivation

Currently, there are tons of phishing websites out there - and it seems that most phishing prevention tools are reactive rather than active. Browsers like Google Chrome and Firefox employ a blacklist of known social engineering websites, curated by user reports and some basic detection algorithms, like image processing with ML [2].

Many businesses' phishing attack mitigation strategies involve educating employees about the risks of phishing scams and a few qualitative examples like misspelled words in an email with malformed links. These examples are not representative of the current state of phishing attack campaigns [3]. A sophisticated and detailed attacker can simply ensure that their phishing attempt doesn't have any typographical or display errors, and many users won't have the technological literacy to deduce whether a website is fake or real.

It's not reasonable to assume that every user will be well-versed in detecting phishing scams, and as attackers get more and more clever with their implementations, it becomes more unlikely that the average user can tell if a website is malicious. For example, if someone is already at the point in which they visit a non-vetted link from an email, it's unlikely that they'll be able to become aware of the phishing attack before they submit their sensitive information to the malicious webpage. This is an high risk and potentially common scenario that comes with incredibly high consequences. Without LibraPhish, preventing users from inadvertently falling victim to a phishing scam requires vetting of every single link that users may encounter. Expecting an IT department to sift through all suspicious webpages is too large a task. LibraPhish can prevent these users from even taking this action - by vetting links at the source, IT teams (or confidence threshold based security systems) can remove all suspicious links from even reaching the user in the first place.

3 Related Work

The phishing detection problem has existed for years and there have been a wide variety of approaches to solve that problem. Perhaps the closest neighbor of LibraPhish are Visual Similarity Based Approaches.

Rosiello et al. [1] presented a system called DOMAntiPhish that compares the Document Object Model (DOM) of different webpages to distinguish between benign and malicious web pages. Specifically, if two different webpages have a similar layout past some threshold, the suspected webpage is determined to be malicious.

Afroz and Greenstadt present PhishZoo [8], which creates a unique profile for a known webpage, based on various visual features such as URL, text content, images such as the website logo, and SSL certificates. These profiles are made for confirmed legitimate websites and stored in a database. Any new suspected website is profiled similarly. If the new website matches the URL or SSL certificate of the previously verified website, then it is determined to be safe. If neither match, then an additional comparison is done between the new website's page content and the previously profiled website. If these are the same, then the new website is inferred to be an attempt to mimic the legitimate website and deemed unsafe.

Mao et al. uses Cascading Style Sheets (CSS) similarity between websites in their approach, BaitAlarm [5]. Since CSS is a language that describes the presentation of a document, the authors claim that it is difficult to replicate websites without copying the website's CSS. Since CSS are highly customizable and visible to any end user, it is quite easy for attackers to copy those settings without any modification. This then makes profiling legitimate websites extremely easy. Websites that employ the same CSS as another legitimate website, but don't match URL or SSL certificates can easily be identified as unsafe.

Pixel-based techniques are also explored in various papers. Here, the main assumption is that websites should not be visually similar - any pair of visually similar websites should indicate the presence of a malicious website. Dunlop et al. proposed GoldPhish [7], a system that extracted logos from suspected websites and converted them into text with optical character recognition (OCR) software. This text is then fed as input into a search engine like Google to obtain generally safe websites. These safe websites are then compared to the website under test. If all don't match, then the suspected website is classified as unsafe.

Although there have been significant research into phishing detection, LibraPhish has various novelties and advantages. For example, in comparison to DOMAntiPhish, PhishZoo, and BaitAlarm, LibraPhish is stateless - it requires no database to conduct its evaluation. GoldPhish is perhaps the closest system to LibraPhish, but LibraPhish can take as abstract logos as input. Additionally, LibraPhish generates a

non-binary level of suspicion for a suspected website, which has higher flexibility for sysadmins, especially when there are a large influx of suspicious websites. Admins can decide that websites above a certain suspicion threshold are automatically all denied, while other suspected websites can be manually reviewed.

4 Our Architecture

LibraPhish is a command-line interface tool implemented in Python 3, using the Selenium library as its main dependency. Selenium is a powerful browser automation tool that allows a computer to mimic an individual's browser connection, which is helpful when attempting to analyze any website from a user's perspective. The following sections explore different parts of LibraPhish and how they determine the possibility of the suspicious website under test as a phishing attack. The code for LibraPhish can be found on Github.

4.1 Logo Image Processing

The vast majority of legitimate websites use digital images and visual icons on their websites. This is especially true for businesses, which have trademarked icons and logos which their users can quickly identify.

Usually, phishing attacks attempt to copy these logos and attackers embed these images in their mimicked websites. These logo images are usually delivered in an additional network request from the web browser, usually containing "logo" and some form of image format, like png, jpg, etc. Additionally, logos are usually displayed clearly, either in the middle of the screen or near the top left.

Browsers usually have this network information, located in the developer tools program installed in the browser. Figure 1 shows an example of a known phishing website using logo images to mimic the legitimate website; Here, the malicious website is attempting to steal a victim's Instagram login credentials. The attacker has crafted a convincing webpage, with a fairly nondescript URL as well (Warning: the following link routes to a malicious website. Enter with caution! [https://bluetickinstagram.cloud/login.html]). An unsuspicious victim could easily be tricked into entering their credentials.

The logo for Instagram is not generated by the Javascript or HTML on the website itself - it's being downloaded from an external source, under the name `logo.png`. The developer tools program on this browser shows this request and the corresponding Instagram logo image response. LibraPhish uses this information as the first step in its image processing for brand detection. Selenium is used to gather this network information from the browser connection. These network requests are usually very standard: in almost all cases, the network request is an HTML GET request to the endpoint that only contains the image itself, as shown on the right side in figure 1.

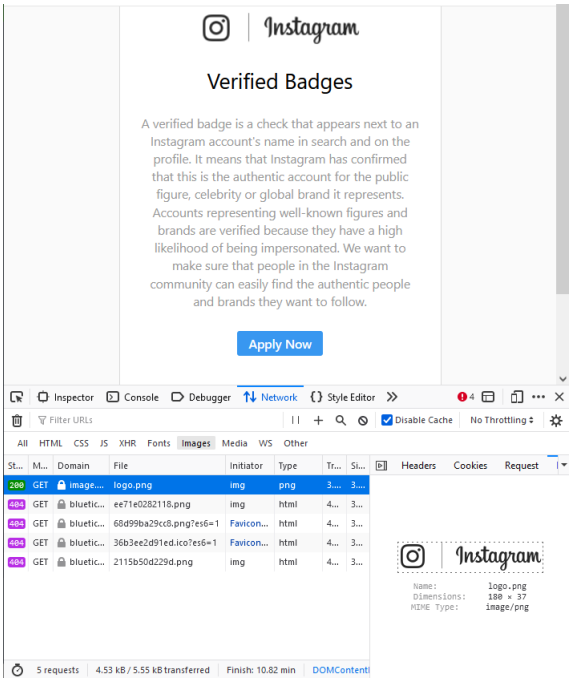

Figure 1. Suspicious Website Mimicking Instagram Using Instagram Logo Images

Sometimes logos aren't easily traced through network information, or their file name isn't easily recognizable. In these cases, it's enough to capture the most essential parts of the user's screen, which will almost certainly showcase the brand logo. This is done via a screenshot feature in Selenium, and the corresponding screen capture can be analyzed for existing logos. Google's Vision AI Logo Detection service is used to determine what logos may exist.

An important part of phishing detection is determining what the corresponding legitimate website is. Since we now have the logo images and their location on the internet, we can easily record those locations and conduct a reverse google image search on each logo image. Google does not have an official reverse-image-search API, but they do have an endpoint with a modifiable query parameter that takes in the endpoint of any image. Again, Selenium is used to access this query-specified reverse image search endpoint and a custom parser is used to collect the search results. These results are aggregated with a bag-of-words model to reduce the search results into a single company's title. Since company names can be more than one word (e.g. American Express), the bag-of-words model is limited to three distinct words, which represent the brand name.

The next step to determine the legitimate website that the logo represents. This is accomplished with another Google search with Selenium, this time with just the brand name. Since Google is a trusted search engine, we can reliably confirm that the suggested links will be safe and accurate.

The output of this module are the top 3 websites listed by Google attributed to the determined brand.

At their core, phishing attacks attempt to imitate legitimate websites. One main assumption that this module relies on is the fact that the image search is completely safe. This is a reasonable assumption, especially with large internet search businesses like Google, whose business model relies on giving users safe and reliable information and links.

4.2 Link Connectivity

The next part of LibraPhish attempts to discern whether the website under investigation is legitimate via attributes of the separate known legitimate websites. First, the logo image processing module must be run in order to generate the set of legitimate websites.

Then, Selenium is used to determine the set of reachable websites from each legitimate website. All neighboring websites are located at a depth of 1. Neighboring websites at a depth of 2 can be determined by finding all reachable non-explored websites from the websites with depth 1. This neighboring link exploration can be done multiple times, depending on the depth. The default depth is set to 3, but can be specified by the user. A set of reachable websites is also determined for the suspicious website, with a user-specified depth defaulted at 3.

Once LibraPhish finishes aggregating the reachable website sets for the legitimate and suspicious website, a domain filtering is used to limit the size of these sets. Each fully qualified domain name is reduced to its second-level domain and top-level domain, while ignoring all sub-directories (for example, "mail.google.com/mail/u/0/" would be reduced to "google.com").

Now, suppose the legitimate website is known as A and the suspicious website under test is known as B. Let the websites reachable from A be known as A^*, and the websites reachable from B be known as B^*. Then, the following rules can be created:

If $B \equiv A$, then B is a legitimate website. This is trivial, since this means that the logo that B embedded in its website actually corresponds to the legitimate website, which is itself, as well.

If $B \in A^*$, then B is likely to be a legitimate website. In most cases, a reachable website from a legitimate website is legitimate as well. There's no good incentive for legitimate websites to link to other malicious websites that attempt to phish their own users.

If there exists an $a \in (A \cup A^*)$ $s.t.$ $a \in B^*$, then B is highly likely to be a phishing website. Most malicious websites conducting phishing campaigns will attempt to mimic a legitimate website. This can be done with stolen logos, icons, and trademarks. Legitimate websites can sometimes contain other logos, such as describing integrations with other businesses or advertising partnerships. However, these legitimate websites would usually be recognized as legitimate or likely legitimate from the first two rules, since their own logo should have been attributed to their own website.

For all other cases, B is still considered likely to be a phishing website, since it had contained a logo associated with A, and its domain is not reachable from A.

The output of this section is a label for the suspicious website, along with the suspected brand associated with it. Table 1 summarizes the possible labels, when the suspicious website is S and the brand associated with a logo that S contains is X.

Label	Brand	Description
Legitimate	X	S is the same as X's website.
Likely Legitimate	X	S's domain can be reached from X's website.
Inconclusive	N/A	A logo cannot be found from S. Additional inquiry needs to be done.
Suspicious	X	S's logo has been traced to X's website, but S does not link to X.
Highly Suspicious	X	S's logo has been traced to X's website, and S links to X.

Table 1. Summary of Website Labels

5 Experimental Results

Experimentation was performed on an a Windows computer with a Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz with 16GB of RAM. The experimental setup involved acquiring a dataset of malicious websites and a dataset of safe, legitimate websites.

To acquire a list of known phishing websites, I used the OpenPhish Database and Mitchell Krog's Phishing Database. These websites are good sources of malicious websites and are frequently updated daily or even more frequently, and old phishing sites that are no longer in use are taken down accordingly.

To acquire a list of known safe websites, I chose businesses by randomly selecting a subset of the Fortune 1000 Company list. Corresponding known legitimate websites for each of these businesses were obtained by getting the top result from a google search API with the brand name.

LibraPhish was run on a test suite with an input of 100 malicious/phishing websites and 100 known legitimate websites. The table 2 below shows the classification rates for the test suite.

LibraPhish performs quite well on malicious websites, with no false negatives. However, a significant amount of malicious websites are labelled "Inconclusive" due to limitations in the logo image processing system or a complete

	Website Type	
	Malicious	Safe
Legitimate	0	34
Likely Legitimate	0	22
Inconclusive	22	40
Suspicious	41	3
Highly Suspicious	37	1
Total	100	100

Table 2. LibraPhish Output on Malicious and Safe Websites

lack of logo on the phishing website. Still, 78 out of the 100 malicious websites were correctly identified automatically.

For known, safe websites, Libraphish performs well, but not as well as it does with malicious websites. The majority of safe websites are correctly classified, but 40% are inconclusive and there are a few false positives. These errors likely stem from a lack of tuning in the link connectivity system and limitations in the logo image processing system.

Overall, these outcomes were expected. In the sphere of security, false negatives such as misclassifying a malicious website as safe are exponentially worse than false positives. With this in mind, LibraPhish does a decent job at detecting phishing websites, but should not be relied on as a complete solution to completely prevent phishing attacks.

6 Further Work

LibraPhish is not at all a completed work. There are a considerable amount of additional work that can be done to enhance LibraPhish's effectiveness.

6.1 Scalable Vector Graphics

Scalable Vector Graphics (SVG) is a vector-based image format, which means that SVG images are easily scaled without loss in quality, unlike raster graphics, which encounter quality decrease upon magnification.

Because of raster graphic limitations, some business logos are designed as in the SVG format. These SVG logos are not difficult to find - most are still available at some endpoint. However, Google Images does not allow SVG formatted images in its reverse image search, so the image must first be converted to a compatible format such as .png or .jpg with sufficient quality in order to be used by Google Images.

Additionally, developers can directly write SVG images into the website HTML, which would allow the logo to surface on the website without a separate network call. In these cases, LibraPhish would need to identify SVG tags within the HTML and convert that image to a compatible format and search with Google Images.

6.2 Input HTML Elements

In almost all phishing campaigns, a bad actor attempts to gain some sort of secretive information from the user, like a username and password. These forms of information are usually input in some input or password HTML element. Websites without any input elements are less likely to be a phishing website, but more work should be done to verify this assumption.

7 Conclusions

The problem of detecting phishing campaigns is incredibly difficult and extremely important, given how common the attacks are. LibraPhish is a tool that uses Logo Image Processing and Link Connectivity techniques to determine the likelihood that a website is safe or malicious/part of a phishing campaign. Many standard anti-phishing tools use a frequently updated blacklist or database that contain reported phishing websites. However, these tools lack an ability to independently recognize when a website is malicious. LibraPhish does not require any state to determine whether a website is malicious - even new phishing campaigns are easily tracked and caught. To accomplish this task, LibraPhish uses Logo Image Processing to identify brands and Link Connectivity techniques to determine the similarity between a suspicious input website and a known legitimate brand. Logo Image Processing is implemented by combing through the network tab for downloaded logo images or using Google Vision AI to find logos within the webpage. For Link Connectivity, LibraPhish explores the reachable sites from the suspicious and the safe websites and uses various set and subset related rules to determine a likelihood score for phishing.

In experimentation, LibraPhish performs reasonably well - there have been no false negatives, and the false positives are extremely rare. Libraphish has a decently high success rate, between 55 to 75 percent, but it does fail to categorize the rest of the inputs. These results indicate that LibraPhish should not be an end-all solution. LibraPhish will work great in tandem with existing off-the-shelf antiphishing tools and blacklisting tools.

References

[1] A. P. E. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi. A layout-similarity-based approach for detecting phishing pages. In *Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm '07)*, pages 454–463, 2007.

[2] Chromium Blog. Faster and more efficient phishing detection in M92 . https://blog.chromium.org/2021/07/m92-faster-and-more-efficient-phishing-detection.html.

[3] Cisco. Cisco: How Phishing Works. https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html#~how-phishing-works.

[4] FBI, United States Department of Justice. FBI Internet Crime Report 2020. https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf.

[5] J. Mao, P. Li, K. Li, T. Wei, and Z. Liang. BaitAlarm: detecting phishing sites using similarity in fundamental visual features. In *Proceedings of the 5th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS '13)*, pages 790–795, 2013.

[6] M. Kan. Google: Phishing Attacks That Can Beat Two-Factor Are on the Rise. https://www.pcmag.com/news/google-phishing-attacks-that-can-beat-two-factor-are-on-the-rise.

[7] M. Dunlop, S. Groat, and D. Shelly. GoldPhish: using images for content-based phishing analysis. In *Proceedings of the 5th International Conference on Internet Monitoring and Protection (ICIMP '10)*, pages 123–128, 2010.

[8] S. Afroz and R. Greenstadt. PhishZoo: detecting phishing websites by looking at them. In *Proceedings of the 5th Annual IEEE International Conference on Semantic Computing (ICSC '11)*, pages 368–375, 2011.